
11785 Final Project Report for Team 28

Frank Hu

Carnegie Mellon University
Pittsburgh, PA 15213
zhemingh@andrew.cmu.edu

Sungmyeon Park

Carnegie Mellon University
Pittsburgh, PA 15213
sungmyep@andrew.cmu.edu

Lehong Wang

Carnegie Mellon University
Pittsburgh, PA 15213
lehongw@andrew.cmu.edu

Zepeng Zhao

Carnegie Mellon University
Pittsburgh, PA 15213
zepengz@andrew.cmu.edu

Abstract

Automated defect detection in Printed Circuit Boards (PCBs) is essential for ensuring the quality and reliability of electronic components. This project leverages a dataset from Peking University’s Open Lab on Human Robot Interaction, containing six types of PCB defects: missing holes, mouse bites, open circuits, shorts, spurs, and spurious copper. High-resolution images from this dataset were augmented and processed into 600×600 sub-images, resulting in a training set of over 10,000 samples.

We employ Squeeze-and-Excitation and cardinality to improve the baseline ResNet50. The final model achieved an overall accuracy of 96.58%, demonstrating robust performance across defect types. Class-specific analysis revealed that the model performed best on missing holes and faced challenges with mouse bites. These results indicate strong generalization with minimal overfitting, as reflected in balanced training and validation metrics.

Future improvements will focus on enhancing the model’s precision for challenging classes through advanced feature extraction and additional data augmentation. This work underscores the potential of deep learning in PCB defect detection, contributing to efficient quality control in electronics manufacturing.

1 Introduction

This project aims to enhance Printed Circuit Board (PCB) defect detection using advanced image-based techniques. Given that PCBs are the backbone of modern electronics, defects like open circuits and scratches can lead to significant performance issues. Current research emphasizes the need for more accurate and reliable inspection systems to reduce such defects.

Traditional methods for PCB defect detection have largely relied on manual visual inspection and rule-based machine vision systems. Rule-based systems employ pre-defined algorithms, such as edge detection and template matching, to identify anomalies. While these methods are useful for simple and repetitive patterns, they struggle with the variability and complexity of real-world defects, such as tiny solder bridges or inconsistent shapes, which require more nuanced understanding and adaptability.

With the rapid advancements in deep learning, a variety of sophisticated models have emerged that are highly suitable for PCB defect classification tasks. Convolutional Neural Networks (CNNs), such as ResNet[8] and DenseNet, have demonstrated exceptional performance in image-based tasks due to their ability to capture spatial hierarchies and extract meaningful features from complex patterns.

Additionally, Transformer-based models like Vision Transformers (ViTs)[5] have shown promise in handling large-scale image data and modeling global relationships within an image, which can be particularly useful for identifying contextual information in PCBs with a variety of components. Furthermore, advanced techniques like object detection models (e.g., YOLO[17], Faster R-CNN[19]) can be employed to localize and classify multiple defects simultaneously, while attention mechanisms and Squeeze-and-Excitation (SE) modules enhance the network’s ability to focus on critical defect regions.

Inspired by previous work and recent advancements, this project aims to explore and experiment with various deep learning model architectures to identify the most effective and accessible solution for robust PCB defect classification. By evaluating these architectures, we seek to establish a strong baseline model that balances accuracy, efficiency, and practicality. Building on this foundation, we will implement targeted improvements to enhance the baseline model’s classification accuracy, particularly for subtle and challenging defect types. Ultimately, our goal is to provide a practical and high-performance solution that can support the electronics industry in achieving better productivity, improved quality control, and more efficient manufacturing processes.

2 Literature Review

In this section, we will discuss three types of PCB defect detection solutions [3]: one-stage, two-stage, and transformer-based algorithms. One-stage algorithms, known for their high detection accuracy and speed, are particularly suited to real-time PCB defect detection. In contrast, two-stage algorithms, although typically slower, excel in detecting intricate defects due to their superior accuracy, making them popular in applications where speed is less critical. Transformer-based algorithms, using transformer models as the backbone, are also rapidly evolving. Additionally, transformer models have shown impressive performance in other fields, suggesting strong potential for PCB defect detection.

2.1 One-Stage Algorithms

Single-stage algorithms, notably represented by the single shot multibox detector (SSD) [14] and You Only Look Once (YOLO) [17] [18]. The YOLO algorithm offers a unique approach by framing target detection as a regression problem. It does this by dividing the input image into a grid and simultaneously predicting the class and bounding box attributes for multiple targets within each grid cell. YOLO performs target detection in a single-shot forward propagation pass, making it exceptionally fast, though it may be less effective for small target detection. Similarly, SSD predicts target classes and bounding boxes on feature maps at different scales, accommodating various target sizes through multiple anchor boxes of different dimensions. This design enables SSD to perform well with small and multi-scale targets.

To address stability and accuracy issues in PCB defect detection models, Xin et al. [25] introduced an improved YOLOv4 model. This model uses mosaic data augmentation during input processing and replaces the leaky rectified linear unit (Leaky-ReLU) activation in the network backbone with the Mish activation function. Similarly, to tackle the challenge of detecting small defects against complex PCB backgrounds, Zhang et al. [27] proposed a lightweight single-stage defect detection network. This network incorporates a dual attention mechanism and a path-aggregation feature pyramid network (PAFPN) to improve the detection of small defects. MobileNetV2, a lightweight backbone neural network, replaces ResNet101, significantly reducing model parameters, while the dual attention mechanism ensures efficient feature extraction. Li et al. [11] developed a dataset for PCB assembly scene object detection to address detection challenges related to anchor box sizes.

Wan et al. [21] proposed a defect detection method with a data-expanding strategy (DE-SSD) and evaluated it using YOLOv5 on both labeled and unlabeled samples. This approach reduces dependency on labeled data by utilizing both types of samples, while the data-expanding strategy helps mitigate the challenges posed by unlabeled data. The improvement is particularly noticeable with smaller data volumes, although the effect decreases as data size increases. In a related study, Wu et al. [23] introduced GSC YOLOv5, a deep learning detection method combining a lightweight network with a dual-attention mechanism. This modified algorithm integrates Ghost Conv and Ghost Bottleneck structures to significantly reduce model parameters and floating-point operations. The inclusion of SE and CBAM modules further enhances accuracy and detection speed. Zhao et al. [22] extended YOLOv5 by incorporating adaptive spatial feature fusion (ASFF) [13], allowing adaptive

fusion of feature information across different spatial levels. Additionally, they introduced the global attention mechanism (GAM) to boost the model's information extraction capabilities. Lim and colleagues [12] developed a novel multi-scale feature pyramid network using YOLOv5, specifically targeting the detection of tiny PCB defects by leveraging contextual insights. This network also integrates the CIoU loss function to improve the precision of spatial parameter estimation, effectively pinpointing defect locations.

In summary, these studies have made notable advancements in PCB defect detection using one-stage algorithms. Innovations include attention mechanisms, data enhancement strategies, and advanced backbone networks. However, challenges remain, including reduced performance in complex defect scenarios, high dependence on limited sample data, and the need for further improvements to handle varied angles and lighting conditions effectively.

2.2 Two-Stage Algorithms

While single-stage algorithms offer faster processing, two-stage algorithms are known for their superior detection accuracy. Prominent two-stage algorithms include Region CNN (R-CNN) [7], Fast R-CNN [6], Faster R-CNN [19], and Mask R-CNN [9]. These approaches divide PCB defect detection into two phases: first, region proposal (RP), where pre-selected boxes are generated to identify areas likely containing defects; second, classification is performed on these regions using CNNs.

The R-CNN algorithm begins by creating a series of candidate regions (Region Proposals) within the input image. Each region undergoes CNN feature extraction, and the resulting features are passed to classifiers and bounding box regressors for target detection. While R-CNN achieves high accuracy, its speed is limited due to the need for separate CNN feature extraction for each candidate region. Faster R-CNN, an improved version, introduces the region proposal network (RPN), a learnable network designed to quickly generate candidate regions. By seamlessly integrating the RPN with classifiers and bounding box regressors, Faster R-CNN forms an end-to-end detection network, enabling candidate region generation and feature extraction within a single network and substantially improving detection speed.

In summary, while fewer studies focus on two-stage algorithms compared to one-stage approaches, two-stage methods clearly offer advantages in detection accuracy. Recently, one-stage algorithms have become popular due to their ability to meet real-time detection speed requirements, with only a slight difference in accuracy compared to two-stage algorithms. Consequently, there is relatively less research centered on two-stage methods.

2.3 Transformer-Based Algorithms

Although transformers have shown strong performance in computer vision (CV) and natural language processing (NLP), they face limitations in visual inspection tasks, particularly due to time constraints. These challenges have led to limited research on transformer applications in PCB defect detection. However, despite these barriers, recent studies have begun exploring ways to overcome these limitations, aiming to fully leverage transformer capabilities for PCB defect detection.

Despite the limited use of transformers in PCB defect detection, advancements in computer vision and computational power have driven the development of numerous transformer models. In 2021, Liu et al. [16] trained a Swin Transformer v2 model with three billion parameters, incorporating post-normalization and scaled cosine attention techniques. This model achieved state-of-the-art (SOTA) results across various visual tasks. As of 2023, the Swin Transformer v2 backbone network remains an active research area, consistently demonstrating exceptional performance.

An et al. [1] introduced a label-robust and patch-correlation-enhanced ViT (LPViT). Their ViT model, based on LPViT principles, emphasizes robustness and leverages relationships among distinct regions of PCB images. Additionally, random masking or substitution of certain blocks enhances the mutual understanding of different image regions. Chen [2] employed an enhanced clustering algorithm to generate optimized anchor frames tailored to the PCB defect dataset. In this approach, CNNs were replaced with a shifted window transformer (Swin Transformer) for network feature extraction, and channel ordering in the feature map was adjusted to allow the network to prioritize more valuable

information. Yang et al. [26] proposed an enhanced YOLOv7 model with the SwinV2_TDD module, which includes an added convolutional layer to facilitate local PCB information extraction.

These innovative transformer models, designed for visual macro-models, capture multi-scale features and excel in dense prediction tasks. Such capabilities are intrinsic to transformer-based PCB defect detection, highlighting transformers' potential in this field.

2.4 Summary

In comparing two-stage and single-stage algorithms, the two-stage algorithm offers higher accuracy but is time-consuming, making it unsuitable for real-time detection tasks. In contrast, the single-stage algorithm is faster but has relatively lower accuracy. Transformer-based PCB target detection algorithms, built on the Transformer architecture, achieve excellent detection accuracy; however, they have slower detection speeds, require higher computational resources for training, and depend on large amounts of data.



Figure 1: Dataset samples after pre-processing

3 Dataset

3.1 Dataset Overview

The PCB defect dataset provides annotated images for defect detection in printed circuit boards (PCBs). This dataset is essential for benchmarking defect detection models and includes six types of defects: missing hole, mouse bite, open circuit, short, spur, and spurious copper.

3.2 Augmented Dataset

To address the small size of the original dataset, data augmentation techniques were applied. High-resolution images were cropped into 600×600 sub-images to form the training and testing datasets.

The augmented dataset contains 19,202 images with corresponding annotations. The augmented dataset can be accessed from:

[https://www.dropbox.com/scl/fi/xfg0mwmpf8jp8q81u5577/VOC_PCB.zip?rlkey=0v5v83usldw1vuu0ul3u54jyw&e=1&d=0\[4\]](https://www.dropbox.com/scl/fi/xfg0mwmpf8jp8q81u5577/VOC_PCB.zip?rlkey=0v5v83usldw1vuu0ul3u54jyw&e=1&d=0[4]). Samples from the dataset is in figure 1.

Type of Defect	Number of Images
Short	3117
Missing Hole	3297
Mouse Bite	3302
Open Circuit	3150
Spurious Copper	3186
Spur	3150

Table 1: Dataset defect statistics.

Dataset Split	Number of Images
Train	13,654
Validation	3,414
Test	2,134

Table 2: Dataset split overview.

The training split contains the majority of the images (13,654), allowing the model to learn from a large sample of defects. The validation split (3,414 images) is used to tune model parameters and prevent overfitting. The test split (2,134 images) is reserved for evaluating the final performance of the trained model, ensuring unbiased metrics on unseen data.

4 Baseline Model

4.1 Baseline Selection

The 2023 study by Sharma et al. [20] establishes SSD ResNet50 as the baseline model and serves as the baseline paper for PCB defect detection, highlighting the effectiveness of ResNet50's deep residual network for this task. With its 50-layer architecture, ResNet50 excels in feature extraction by enabling the propagation of gradients through residual connections, which mitigates issues such as vanishing gradients. This structure is particularly valuable in PCB defect detection, where subtle variations in defect patterns, like minor scratches or PSR peel-off, require nuanced and detailed feature extraction to achieve high classification accuracy [9].

While the SSD framework adds efficiency by combining localization and classification in a single-stage approach, the ResNet50 backbone is the core of this model's strength. Its robust feature representation and deep layer structure allow it to capture high-level semantic features, making it well-suited for distinguishing complex defect types. This depth enables precise localization of defect

regions, even in high-resolution images, supporting applications where timely and accurate defect identification is essential to maintaining manufacturing quality and throughput [15].

As the baseline model validated by Sharma et al. [20], SSD ResNet50 leverages ResNet50’s powerful feature extraction capabilities, making it a reliable solution for real-time PCB quality control. Its adoption as a baseline in defect detection research underscores ResNet50’s advanced capacity to handle high-dimensional, complex data—meeting the stringent demands of industrial-scale PCB inspection.

4.2 Baseline Model Description

ResNet50 is a 50-layer deep residual network known for its powerful feature extraction capabilities, making it well-suited for complex image classification tasks. With residual connections that facilitate efficient gradient flow, ResNet50 enables deep and nuanced feature representation, capturing intricate details essential for distinguishing subtle variations, such as those found in PCB defects. This depth and robustness in feature extraction allow ResNet50 to handle high-dimensional data with precision, making it an ideal choice for tasks requiring high accuracy and reliability. When paired with detection frameworks, such as SSD, ResNet50 becomes highly effective for real-time inspection applications.

4.3 Mathematical Description

Given an input printed circuit board (PCB) image $X \in \mathbb{R}^{H \times W \times C}$, we perform feature extraction using a ResNet-50 network. Let $f_{\text{ResNet}} : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{h \times w \times d}$ denote the feature extraction process, yielding the feature map $F \in \mathbb{R}^{h \times w \times d}$ such that:

$$F = f_{\text{ResNet}}(X)$$

where h and w are the spatial dimensions of the feature map, and d is the feature depth.

To detect objects within the PCB image, the base model apply the Single Shot MultiBox Detector (SSD)[15] approach on the feature map F . The SSD process involves the following steps:

1. Anchor Boxes: We define a set of anchor boxes, also known as “default boxes”, at each spatial location of F . Let $\mathcal{A} = \{(x_i, y_i, w_i, h_i)\}_{i=1}^M$ denote the set of anchor boxes, where (x_i, y_i) are the center coordinates, and w_i and h_i are the width and height of the i -th anchor box. Each anchor box has a predefined aspect ratio and scale.

2. Offset and Class Prediction: For each anchor box $(x_i, y_i, w_i, h_i) \in \mathcal{A}$, SSD predicts:

- Class probabilities $\mathbf{p}_i = (p_{i,1}, p_{i,2}, \dots, p_{i,C+1}) \in \mathbb{R}^{C+1}$, representing the probability distribution over C object classes and an additional background class.
- Bounding box offsets $\mathbf{d}_i = (t_{x,i}, t_{y,i}, t_{w,i}, t_{h,i}) \in \mathbb{R}^4$, which are adjustments to the center coordinates and dimensions of the anchor box, refining it to fit detected objects.

3. Loss Function: The objective function for SSD is a weighted sum of the classification and localization losses, applied across all anchor boxes:

$$L = \frac{1}{N} (L_{\text{conf}} + \alpha L_{\text{loc}})$$

where N is the number of matched anchor boxes, and α is a weighting factor. The components of the loss function are defined as follows:

- Classification Loss L_{conf} : This is a cross-entropy loss applied to the predicted class probabilities \mathbf{p}_i for each anchor box, defined as:

$$L_{\text{conf}} = - \sum_{i=1}^M \sum_{c=1}^{C+1} y_{i,c} \log(p_{i,c})$$

where $y_{i,c}$ is the ground truth label (one-hot encoded) for class c of anchor box i .

- Localization Loss L_{loc} : This is a Smooth L1 loss applied to the bounding box offsets for the matched anchor boxes:

$$L_{\text{loc}} = \sum_{i=1}^M \sum_{m \in \{x,y,w,h\}} \text{SmoothL1}(\hat{d}_{i,m} - d_{i,m})$$

where $\hat{d}_{i,m}$ is the predicted offset and $d_{i,m}$ is the ground truth offset for anchor box i along dimension m , and the Smooth L1 loss is defined as:

$$\text{SmoothL1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

4. Final Detection: After computing the class probabilities and bounding box offsets, Non-Maximum Suppression (NMS) is applied to eliminate redundant detections, retaining only the highest-confidence predictions for each detected object. This completes the SSD-based object detection process for the PCB image.

5 Baseline Implementation

The proposed approach for printed circuit board (PCB) defect classification leverages a transfer learning strategy using the ResNet-50 architecture. The primary objective is to detect and classify six common defect types: mouse bites, shorts, open circuits, spurs, missing holes, and spurious copper. The implementation consists of data preprocessing, model training, evaluation, and visualization components.

5.1 Data Preparation and Preprocessing

The dataset was sourced from PCB images formatted following the PASCAL VOC structure. Image annotations were stored in XML format, containing detailed defect class information. The PCB Dataset class was created to load images and their respective labels by parsing the XML annotation files. The images were transformed using standard normalization and resizing operations to ensure compatibility with ResNet-50, which requires 224×224 pixel inputs.

Key transformations included:

- **Resize**: Images were resized to 300×300 pixels.
- **Normalization**: Mean and standard deviation values specific to ImageNet were applied for data normalization.

5.2 Model Architecture

The ResNet-50 model, pre-trained on ImageNet, was adapted for multi-class classification by modifying its final fully connected layer. The original output layer was replaced with a new linear layer to output predictions for six classes. This adaptation retained the feature extraction capabilities of ResNet-50 while fine-tuning the network for PCB defect classification.

5.3 Training Process

The model was trained using the following settings:

- **Optimizer**: AdamW optimizer with a learning rate of 0.0001.
- **Loss Function**: Cross-entropy loss.
- **Scheduler**: A ReduceLROnPlateau scheduler was employed to reduce the learning rate when the validation loss plateaued, ensuring stable convergence.
- **Batch Size**: 256 for training and testing, with 8 for validation.

Training and evaluation metrics were tracked for each epoch, including overall and per-class accuracy, and loss.

6 Baseline Model Evaluation

The classification model was trained and evaluated on a dataset consisting of six defect classes: Mouse Bites, Shorts, Open Circuits, Spurs, Missing Holes, and Spurious Coppers. The evaluation metrics include per-class accuracy and loss, overall accuracy, and loss trends during training and validation.

6.1 Per-Class Test Performance

The performance across the different classes was evaluated using accuracy and loss. The results show a variance in model effectiveness across the defect types:

- **Mouse Bites:** The model achieved an accuracy of 89.55% with a test loss of 0.2928. This indicates a relatively high loss compared to other classes, which suggests that Mouse Bites were among the more challenging classes for the model to classify accurately.
- **Shorts:** The accuracy was 91.93% with a loss of 0.2679. While the accuracy is above 90%, the associated loss highlights some classification uncertainty.
- **Open Circuits:** This class performed well with an accuracy of 93.64% and a test loss of 0.2229, showing that the model handled these defects effectively.
- **Spurs:** Achieved an accuracy of 90.68% and a test loss of 0.2706, comparable to the Shorts class.
- **Missing Holes:** This was the best-performing class with a high accuracy of 98.64% and the lowest test loss of 0.0334, indicating that the model had a strong ability to correctly identify these defects.
- **Spurious Coppers:** An accuracy of 94.01% and a loss of 0.2412, showing robust performance with moderate confidence.

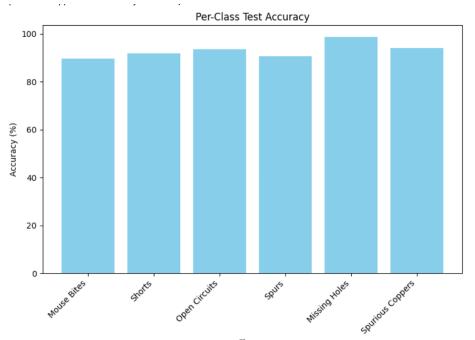


Figure 2: Per-Class Test Accuracy.

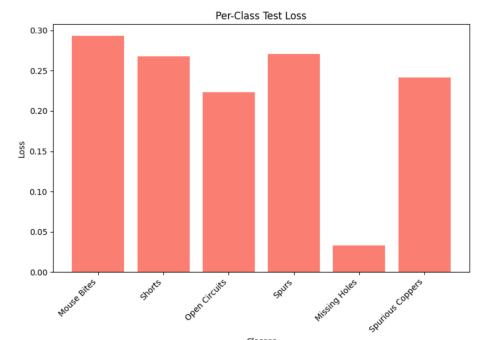


Figure 3: Per-Class Test Loss.

6.2 Overall Baseline Model Performance

The overall test accuracy reached 93.02%, demonstrating that the model generalizes well across the dataset. The final overall test loss was 0.2216, reflecting the model's effective minimization of classification errors.

6.3 Training and Validation Analysis

The training and validation curves (Figures 3 and 4) illustrate the learning behavior of the model across epochs:

- **Training and Validation Accuracy:** The model showed a steady increase in both training and validation accuracy, reaching convergence around epoch 20. This consistent improvement highlights the effectiveness of the model's learning capacity without significant overfitting.



Figure 4: Training and Validation Accuracy over Epoch

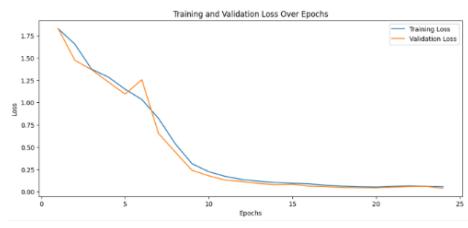


Figure 5: Training and Validation Loss over Epoch

- **Training and Validation Loss:** Both training and validation loss curves decreased progressively, indicating effective learning. After epoch 20, the loss values plateaued, signifying that further training beyond this point offered diminishing returns.

6.4 Baseline Result Discussion

The per-class performance revealed variability in the model’s ability to classify certain defect types, with Missing Holes standing out as the most accurately classified defect, while Mouse Bites posed a greater challenge. The balanced trend between training and validation performance suggests that the model maintained generalization without overfitting, which was further supported by the absence of divergence between training and validation metrics over epochs.

The promising performance across various metrics positions this model as a reliable tool for defect classification. However, further improvements may focus on enhancing the precision for challenging classes such as Mouse Bites through additional data augmentation, feature extraction, or algorithmic adjustments.

7 Implemented Extensions

The final model for PCB defect detection was developed through s with different neural architectures. As mentioned above, we used ResNet50 as our baseline. Building on this, we explored EfficientNet, Vision Transformer (ViT), and SE-ResNext, each offering unique benefits, such as improved computational efficiency or more effective representation learning. After thoroughly evaluating these models on our PCB defect dataset, we selected SE-ResNext as our final model as it achieves a strong accuracy, while balancing efficiency, and generalization.

7.1 EfficientNet Experimentation

We first experimented with the EfficientNet-B0 design. The EfficientNet-B0 model achieved an overall test loss of 0.5094 and an overall test accuracy of 84.40%. Examining each defect class, it achieved 74.63% accuracy (loss: 0.7806) on Mouse Bites, 86.74% accuracy (loss: 0.4826) on Shorts, 86.36% accuracy (loss: 0.4525) on Open Circuits, 80.79% accuracy (loss: 0.6279) on Spurs, 96.46% accuracy (loss: 0.0971) on Missing Holes, and 82.34% accuracy (loss: 0.5946) on Spurious Coppers. Although the model demonstrated strong performance in most categories, certain classes, such as Mouse Bites, presented greater challenges. This follows the trend we identified in the baseline model where small and irregular defects like Mouse Bites are harder to classify.

7.2 Vision Transformer (ViT) Experimentation

The Vision Transformer (ViT) model only reached an overall test loss of 1.4288 and an overall test accuracy of 29.71%. Breaking down the performance by defect class, it achieved 13.18% accuracy (loss: 1.6144) on Mouse Bites, 20.17% accuracy (loss: 1.6013) on Shorts, 0.00% accuracy (loss: 1.7604) on Open Circuits, 22.60% accuracy (loss: 1.6626) on Spurs, 91.83% accuracy (loss: 0.4198) on Missing Holes, and 28.14% accuracy (loss: 1.5592) on Spurious Coppers. Although it performed relatively well on Missing Holes, its accuracy was notably lower across most other defect classes, indicating that the ViT model struggled to consistently identify defect types. Overall the result for ViT was much worse compared to our baseline and other models.

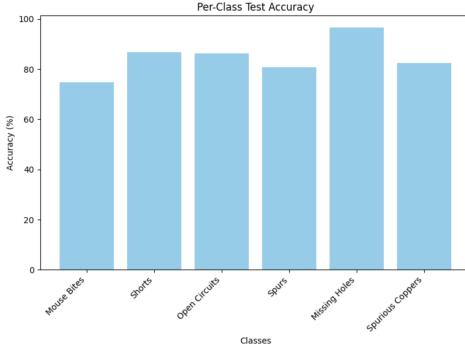


Figure 6: EfficientNet Per-Class Test Accuracy.

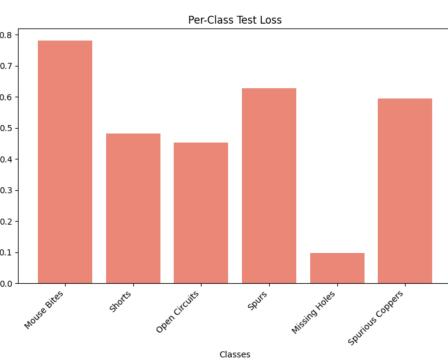


Figure 7: EfficientNet Per-Class Test Loss.



Figure 8: EfficientNet Training and Validation Accuracy over Epoch

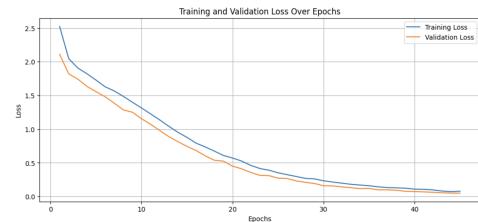


Figure 9: EfficientNet Training and Validation Loss over Epoch

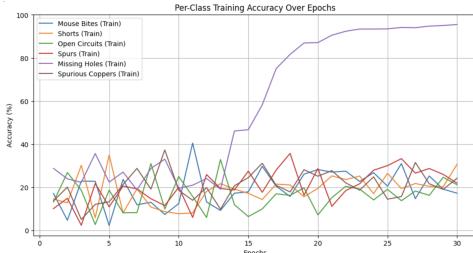


Figure 14: Per class ViT Training and Validation Accuracy over Epoch

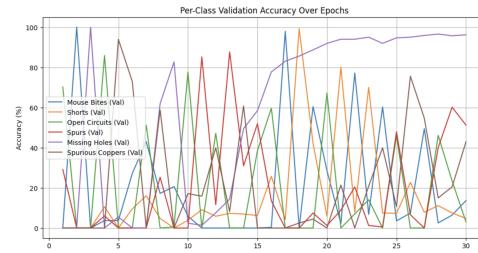


Figure 15: Per classViT Training and Validation Loss over Epoch

8 Final Model Description

8.1 Implemented Extensions

To improve our model performance, based on the proposed model extensions from literature review, we added Squeeze-and-Excitation technique[10] and Cardinality technique[24].

The Squeeze-and-Excitation (SE) module is a neural network component designed to improve the representational power of convolutional neural networks (CNNs). It adaptively recalibrates channel-wise feature responses by explicitly modeling interdependencies between channels. An SE block consists of 2 steps: a) Squeeze, where the spatial dimensions of the feature maps are reduced by performing global average pooling across each channel. This operation squeezes the spatial information into a single value per channel, producing a global descriptor; b) Excitation, where the squeezed descriptors are passed through two fully connected (FC) layers with non-linearities to model the dependencies between channels. SE modules enable the network to focus on the most important feature channels while suppressing less useful ones. This enhances the discriminative power of the model.

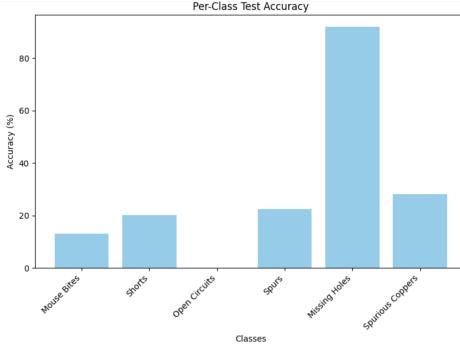


Figure 10: ViT Per-Class Test Accuracy.

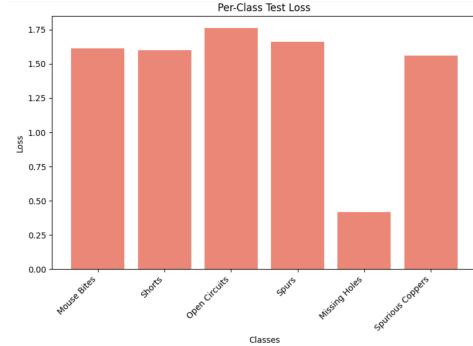


Figure 11: ViT Per-Class Test Loss.

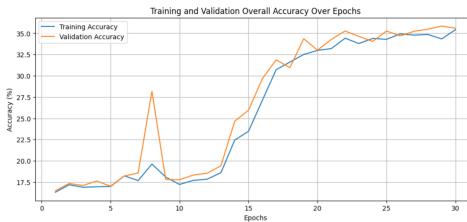


Figure 12: ViT Training and Validation Accuracy over Epoch

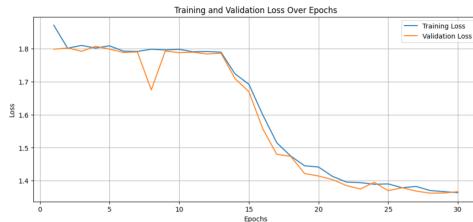


Figure 13: ViT Training and Validation Loss over Epoch

ResNeXt introduces cardinality as a new dimension for scaling networks, alongside depth and width. By using grouped convolutions and increasing the number of groups, ResNeXt achieves a better trade-off between performance and efficiency. This feature is supported by Pytorch, therefore it's easy to implement.

8.2 Architecture

In figure 16, we show the architecture of our final model, which is built following the design philosophy of ResNet.

At the beginning, we use Conv2d and MaxPool2d to compress the images size while promoting the channel dimmension, making input ready for Bootleneck to process. There are 4 following layers consisting of different numbers of Bottlenecks. Within each layer, there is 1 Bottleneck with downsampling feature and some Bottlenecks with this feature. Downsampling Bottleneck is used for compressing the image size while promoting channel dimension. Downsampling is used for ensuring correct shape for residual connection. The architecure and modules of Bottleneck with or without Downsample is also drawn next to the overall architecture. Finally, after flowing through all Bottleneck layers, intermediates are sent to AdaptiveAvgPool2d to compress them to [1,1], which will be flattened and used as input for fully connected layer. The last two layers serve as the classifier for our classification problem.

The shape of input, output, and intermediates are shown in the diagram. It clearly shows how the tensors flow through the model and how their shape is changed along the way. This is how the most ResNet works.

9 Evaluation Metric

The evaluation of our model is based on two key aspects: the loss function and the evaluation metric.

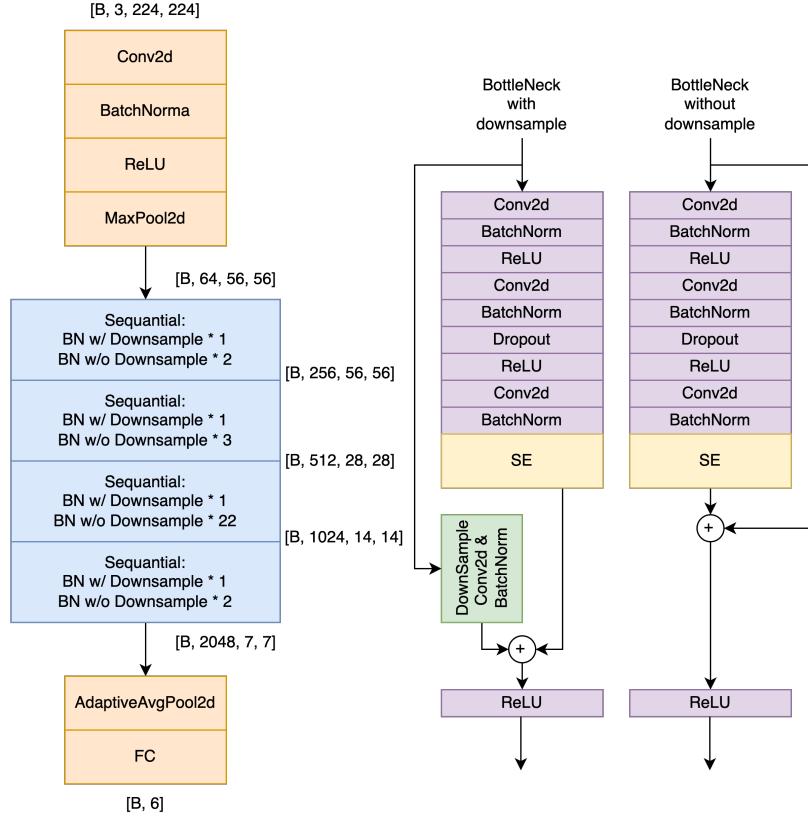


Figure 16: Architecture diagram of final model

9.1 Loss Function: Cross-Entropy Loss

To train the model, we utilize the Cross-Entropy Loss, which is a widely used loss function for classification tasks. It measures the difference between the predicted probability distribution and the true class labels. For a given input sample, the cross-entropy loss is defined as:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log(\hat{y}_{i,c}), \quad (1)$$

where:

- N is the total number of samples,
- C is the number of classes,
- $y_{i,c}$ is a binary indicator (1 if the class label c is the correct class for sample i , 0 otherwise),
- $\hat{y}_{i,c}$ is the predicted probability of class c for sample i .

This loss function ensures that the predicted probability for the correct class is maximized during training.

9.2 Evaluation Metric: Accuracy Score

The primary evaluation metric used to assess the performance of our model is the Accuracy Score. Accuracy is defined as the proportion of correctly classified samples out of the total number of samples, expressed mathematically as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples}}. \quad (2)$$

For a dataset with N samples, this can be written as:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N \mathbb{1}(\hat{y}_i = y_i), \quad (3)$$

where:

- \hat{y}_i is the predicted label for sample i ,
- y_i is the true label for sample i ,
- $\mathbb{1}(\cdot)$ is the indicator function, which equals 1 if its argument is true and 0 otherwise.

This metric provides an intuitive measure of the model's performance by quantifying how often the predicted labels match the true labels.

10 Final model Result

Again, the final model was implemented using a SE-ResNext architecture, was trained and tested on a dataset comprising six defect categories: Mouse Bites, Shorts, Open Circuits, Spurs, Missing Holes, and Spurious Coppers. We assessed its performance using per-class accuracy and loss, along with overall accuracy metrics and observed training and validation loss trends over time same as the baseline model.

10.1 Final Model Per-Class Test Performance

The performance across the different classes was evaluated using accuracy and loss. The results show promising performance across the different defect types:

- **Mouse Bites:** The model achieved an accuracy of 92.04% with a test loss of 0.3406. Although the accuracy is high, the loss value suggests that the model faced moderate difficulty in distinguishing Mouse Bites.
- **Shorts:** With an accuracy of 97.41% and a loss of 0.0926, the model showed strong proficiency. The low loss indicates a high degree of confidence in classifying Shorts.
- **Open Circuits:** The model performed very well, attaining 97.88% accuracy and a test loss of 0.1002. These metrics indicate that Open Circuits were correctly identified with relative ease and consistency.
- **Spurs:** Spurs were accurately identified with a 96.61% accuracy and a 0.2200 loss. While performance remained robust, the slightly higher loss compared to Shorts and Open Circuits suggests a bit more classification uncertainty.
- **Missing Holes:** This was the best-performing class, achieving a flawless 100.00% accuracy and a minimal test loss of 0.0020. The model displayed an exceptional ability to recognize Missing Holes.
- **Spurious Coppers:** With a 96.11% accuracy and a loss of 0.1423, the model demonstrated strong performance. The relatively low loss underscores a confident classification for Spurious Coppers.

10.2 Overall Final Model Performance

Overall, the final model demonstrated strong classification capabilities in PCB defects, achieving an overall test accuracy of 96.58%. This high accuracy, accompanied by a relatively low overall test loss of 0.1538, indicates that the model consistently identified defects suggesting good results.

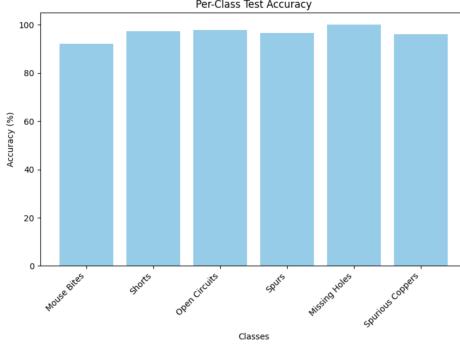


Figure 17: Final Model Per-Class Test Accuracy.

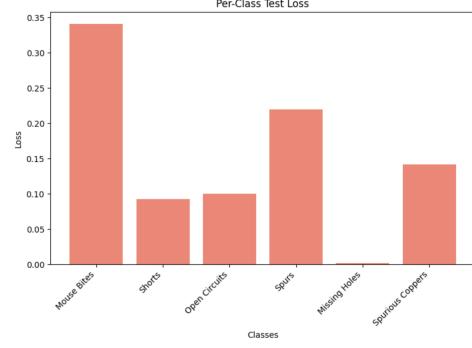


Figure 18: Final Model Per-Class Test Loss.



Figure 19: Final Model Training and Validation Accuracy over Epoch

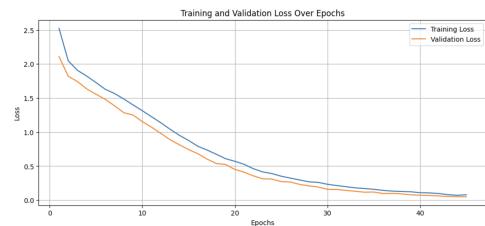


Figure 20: Final Model Training and Validation Loss over Epoch

10.3 Training and Validation Analysis

The training and validation curves (Figures 15 and 16) illustrate the learning behavior of the model across epochs:

- **Training and Validation Accuracy:** The final model with SE-ResNext showed a steady increase in both training and validation accuracy, reaching convergence around epoch 35. This trend shows that our final model was able to learn without significant overfitting.
- **Training and Validation Loss:** Both training and validation loss curves decreased progressively, indicating effective learning. After about epoch 35, the decrease in loss values started to slow down, which can show that further training beyond this point offered diminishing returns.

11 Discussion

In this study, we explored multiple deep learning architectures and techniques to improve automated defect detection in PCB images. Our work began with a baseline model employing an SSD framework with a ResNet50 backbone, which achieved a robust overall accuracy of 93.02%. Notably, the model excelled in identifying certain defects such as Missing Holes, achieving near-perfect accuracy, while encountering more difficulty in distinguishing subtler defects like Mouse Bites. The balanced trends in training and validation metrics indicated limited overfitting and good generalization, demonstrating that even a relatively straightforward model can perform well on PCB defect data given careful preprocessing, appropriate hyperparameters, and sufficient training.

Building on this baseline, we experimented with more advanced architectures, including EfficientNet and Vision Transformer (ViT). Although EfficientNet offered a balance between efficiency and accuracy, it did not outperform the baseline model. ViT, while promising in other computer vision tasks, struggled significantly on our PCB dataset, potentially due to the model's limited ability to handle localized defect patterns within a complex background.

The most substantial improvement was observed with our final chosen architecture: SE-ResNext. By integrating Squeeze-and-Excitation (SE) modules and advanced residual connections, the SE-ResNext

model achieved an overall accuracy of 96.58%, outperforming both the baseline and other tested models. This gain can be attributed to the model's ability to focus on critical feature maps and enhance subtle defect patterns, leading to more confident predictions. The final model succeeded in narrowing the performance gap between easy-to-detect and challenging defects, maintaining strong accuracy across classes and improving upon the baseline's weakest points. The near-flawless classification of Missing Holes and the improved accuracy on Mouse Bites highlight the effectiveness of advanced attention mechanisms and feature recalibration.

Throughout these experiments, we confirmed several key insights. First, the careful choice of architecture and attention mechanisms is crucial for capturing subtle defects in PCB images. Second, more complex architectures do not necessarily yield better results if not well-suited to the data characteristics. Finally, addressing noisy, imbalanced data remains an ongoing challenge, influencing model robustness and performance.

12 Conclusion

This project demonstrates the effectiveness of deep learning models in detecting PCB defects, highlighting both the strengths and limitations of various architectures. Starting from a robust SSD ResNet50 baseline, we integrated different strategies and advanced models to achieve significant performance improvements. The final SE-ResNext model, achieving a test accuracy of 96.58%, underscores the potential of carefully designed networks and attention mechanisms in industrial inspection tasks.

The outcomes emphasize that automated defect detection systems can substantially improve manufacturing quality, reduce the reliance on manual inspection, and enhance throughput. Our findings encourage the broader adoption of advanced deep learning approaches in real-world electronic manufacturing quality control, paving the way for more reliable, efficient, and scalable inspection solutions.

13 Future Work

Several avenues for future research and enhancements are apparent:

- **Data Expansion and Augmentation:** As training is highly influenced by unlabeled samples, employing advanced data-expanding strategies and semi-supervised approaches can increase dataset diversity, thus improving model robustness. Techniques like mosaic augmentation—common in models such as YOLOv5—cut-mix, or even GAN-based synthetic data generation can mitigate overfitting and handle data scarcity. Additionally, selectively masking or substituting regions of the PCB image, inspired by Vision Transformer (ViT) training methods, may help the model better understand context and maintain resilience to occlusion and noise.
- **Advanced Loss Functions:** Replacing the standard loss function with more advanced ones, such as the Complete Intersection over Union (CIoU) loss used in YOLOv4 and YOLOv5, can lead to more accurate localization of defects. Such refined loss metrics can significantly improve training outcomes and model precision.
- **Architectural Enhancements and Attention Modules:** Introducing multi-scale detection capabilities (e.g., adding an FPN layer to ResNet50) can improve the model's ability to identify defects of varying sizes. Furthermore, replacing parts of ResNet50 with MobileNetV2 layers can reduce the model size and accelerate inference, making it more suitable for resource-constrained environments.
- **Advanced Attention and Transformer Models:** Although initial experiments with ViT were not conclusive, next-generation transformer-based backbones, such as Swin Transformer, or hybrid CNN-transformer architectures may better capture complex defect patterns. These architectures can improve both large and small defect detection tasks and show increased robustness. Future work should also consider combining CNNs with ViT-inspired masking or feature substitution strategies to enhance performance.
- **Real-Time Inference and Hardware Optimization:** For integration into manufacturing lines, optimizing models for low-latency, on-device inference is crucial. Model pruning,

quantization, and lightweight backbones can be explored to maintain accuracy while improving efficiency and scalability in real-time applications.

By pursuing these directions—refining data augmentation, exploring advanced losses, leveraging attention mechanisms, adopting hybrid architectures, and optimizing for real-world deployment—we can continue to enhance PCB defect detection solutions. Ultimately, these future improvements will lead to more reliable detection, lower production costs, and compliance with stringent quality standards in electronics manufacturing.

14 Code Availability

Our code is all open-source and available at <https://github.com/MiRaCLeXeoN/PCB-Defect-Classification>. Please refer to README file for environment setup and how to run the code.

15 Division of work

1. Zepeng Zhao: Literature review, baseline implementation, final model description, final model implementation, final model training, report review, and Github repository organization
2. Frank Hu: Introduction, baseline model selection and description, model experiments analysis, final result analysis, visualization analysis
3. Lehong Wang: Mathematical description, baseline model selection, evaluation metrics, discussion, conclusion and future work, and report formatting
4. Sungmyeon Park: Baseline model implementation, final and other model implementations, managing dataset, and model evaluations

References

- [1] Kang An and Yanping Zhang. Lpvit: a transformer based model for pcb image classification and defect detection. *IEEE Access*, 10:42542–42553, 2022.
- [2] Wei Chen, Zhongtian Huang, Qian Mu, and Yi Sun. Pcb defect detection method based on transformer-yolo. *IEEE Access*, 10:129480–129489, 2022.
- [3] Xing Chen, Yonglei Wu, Xingyou He, and Wuyi Ming. A comprehensive review of deep learning-based pcb defect detection. *IEEE Access*, 2023.
- [4] Runwei Ding, Linhui Dai, Guangpeng Li, and Hong Liu. Tdd-net: a tiny defect detection network for printed circuit boards. *CAAI Transactions on Intelligence Technology*, 4(2):110–116, 2019.
- [5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.
- [6] R Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.
- [7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

- [10] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.
- [11] Jing Li, Weiye Li, Yingqian Chen, and Jinan Gu. Research on object detection of pcb assembly scene based on effective receptive field anchor allocation. *Computational Intelligence and Neuroscience*, 2022(1):7536711, 2022.
- [12] JiaYou Lim, JunYi Lim, Vishnu Monn Baskaran, and Xin Wang. A deep context learning based pcb defect detection model with anomalous trend alarming system. *Results in Engineering*, 17:100968, 2023.
- [13] Songtao Liu, Di Huang, and Yunhong Wang. Learning spatial fusion for single-shot object detection. *arXiv preprint arXiv:1911.09516*, 2019.
- [14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I* 14, pages 21–37. Springer, 2016.
- [15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. *SSD: Single Shot MultiBox Detector*, page 21–37. Springer International Publishing, 2016.
- [16] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022.
- [17] J Redmon. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [18] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.
- [19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.
- [20] Sachin Sharma, Dhruv Patel, and Dharmesh Shah. Automated detection and classification of defects in pcb using deep learning techniques: A comparison of resnet50v1 and efficientdet d1. *2023 9th IEEE India International Conference on Power Electronics (IICPE)*, pages 1–6, 2023.
- [21] Yusen Wan, Liang Gao, Xinyu Li, and Yiping Gao. Semi-supervised defect detection method with data-expanding strategy for pcb quality inspection. *Sensors*, 22(20):7971, 2022.
- [22] Xiaoqi Wang, Xiangyu Zhang, and Ning Zhou. Improved yolov5 with bifpn on pcb defect detection. In *2021 2nd International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, pages 196–199. IEEE, 2021.
- [23] Ligang Wu, Liang Zhang, and Qian Zhou. Printed circuit board quality detection method integrating lightweight network and dual attention mechanism. *IEEE Access*, 10:87617–87629, 2022.
- [24] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks, 2017.
- [25] Haojia Xin, Zibo Chen, and Boyuan Wang. Pcb electronic component defect detection method based on improved yolov4 algorithm. In *Journal of Physics: Conference Series*, volume 1827, page 012167. IOP Publishing, 2021.
- [26] Yujie Yang and Haiyan Kang. An enhanced detection method of pcb defect based on improved yolov7. *Electronics*, 12(9):2120, 2023.
- [27] Yue Zhang, Fei Xie, Lei Huang, Jianjun Shi, Jiale Yang, and Zongan Li. A lightweight one-stage defect detection network for small object based on dual attention mechanism and pafpn. *Frontiers in Physics*, 9:708097, 2021.