

Metric Learning-based Incorrect Component Detection Framework

Chia-Yu Lin*, Yu-Chiao Kuo†, Sheng-Che Lin‡, and Ted T. Kuo§

* Department of Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan

† Department of Computer Science and Engineering, Yuan Ze University, Taoyuan, Taiwan

‡ Department of Creative Convergence Design, Korea University, Seoul, South Korea

§ College of Artificial Intelligence, National Yang Ming Chiao Tung University, Tainan, Taiwan

Corresponding Author: Chia-Yu Lin (sallylin0121@ncu.edu.tw)

Abstract—Artificial intelligence is applied in surface mount technology (SMT) to reduce the false alarm rate of automatic optical inspection systems. Among SMT defects, incorrect component placement—where a component is positioned incorrectly—is one of the most critical and damaging issues. Many factories rely on optical character recognition (OCR) to detect incorrect components; however, OCR struggles with blurry, obstructed, or dirt-covered characters. To address these challenges, this paper proposes a Metric Learning-based Incorrect Component Detection (MICD) framework, designed to handle new component and text detection tasks in low-quality images. The MICD framework is evaluated against the OCR-based method, demonstrating a leakage rate below 7% and a false alarm rate of only 10% of that of OCR. Overall, the proposed framework effectively detects incorrect components across various production lines while significantly reducing false alarm rates.

Index Terms—Surface mount technology, incorrect component placement, metric learning

I. INTRODUCTION

Defect detection is crucial in surface mount technology (SMT) production, where defects like solder balls, bridging, and tombstones frequently occur. Automated optical inspection (AOI) systems help identify incorrect components, but their strict screening often results in high false alarm rates [1], leading to costly manual inspections. To mitigate this, AI is being explored to enhance defect detection.

Among SMT defects, incorrect component placement—where a machine mispositions a component—can be particularly detrimental, often triggering cascading errors. Traditionally, manufacturers rely on optical character recognition (OCR) techniques to read product serial numbers on printed circuit boards (PCBs) and detect incorrect components [2]. However, OCR performance deteriorates significantly when text elements are blurry or obscured by dust, as shown in Fig. 1, and it frequently misinterprets component layouts when new products are introduced.

Therefore, this paper proposes a Metric Learning-based Incorrect Component Detection (MICD) framework, which replaces text recognition with feature similarity learning. Unlike OCR and existing deep learning methods that require extensive retraining, MICD leverages metric learning to generalize across variations without text parsing. Its golden sample selection further enhances robustness in diverse production conditions. Unlike conventional metric learning in

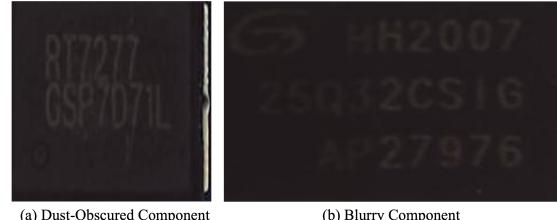


Fig. 1: The examples of blurry and dust-obscured components.

image retrieval, MICD is specifically designed for SMT defect detection, addressing layout and component variations challenges. The framework consists of three modules: image preprocessing, feature extraction, and feature comparison. In the image preprocessing module, component images are converted to grayscale and sharpened to improve clarity. The feature extraction module utilizes prototypical and relation networks to extract robust features from components. The feature comparison module identifies a golden sample and computes its similarity to the target component, classifying it as correct or incorrect based on a predefined threshold.

In the experiments, we evaluate the MICD framework on a real-world SMT dataset and compare its performance with Google Cloud OCR [3]. In addition, we conduct an ablation study to analyze the contributions of the prototypical and relation networks, and we use Grad-CAM methods [4] to provide interpretability. The proposed MICD framework significantly outperforms the OCR method, with a 54% reduction in false alarm rate in one category and a 90% reduction in another, demonstrating its superior accuracy across diverse component types. The MICD framework introduces several key contributions:

- Effectively detects incorrect components across diverse production lines, significantly reducing false alarm rates.
- Demonstrates superior accuracy compared to OCR methods, MICD demonstrates superior accuracy in recognizing components, even in dusty conditions.
- Employs a similarity comparison design, enabling adaptation to products with varying layouts.

II. RELATED WORKS

Placing incorrect components on a PCB in SMT production lines can lead to significant losses. Manufacturers typically use two methods to verify the correctness of components.

The first approach utilizes Optical Character Recognition (OCR) to identify text elements on the components. A previous study [5] designed a set of low-cost image processing techniques for PCBs, focusing on contour defect analysis, pixel subtraction, and OCR. For OCR training, they used NI OCR training in LabVIEW to build dataset-specific parameters. However, OCR struggles with obscured or dirty components, making text unreadable. To overcome these limitations, deep learning methods are increasingly explored for incorrect component detection.

The second approach directly recognizes component images. Chang et al. [6] presented a detection method for SMT production, leveraging MobileNet with two fully connected layers (FCLs). One layer acts as a feature extractor, comparing sample images against golden samples to reduce false alarms in AOI-based component marking detection.

Other deep learning-based methods have also been applied to PCB defect detection. Tan et al. [7] incorporated transfer learning with AlexNet, ResNet-50, and GoogleNet to develop a fault detection method for PCB components, achieving 99% accuracy in component identification. Chen et al. [8] proposed an end-to-end model for IC marking inspection, employing weakly supervised learning for character recognition and a direction recognition module to handle multidirectional markings. Their model achieved a 96.34% recall rate on real SMT chip datasets. However, these methods are limited to clear PCB components. When text is obscured by dirt, damage, or blurring, they struggle to recognize characters and fail to detect incorrect components.

To address unclear text issues, Nava et al. [9] extracted corrupt or unreadable numeric characters from PCB images and manually segmented them to create a dataset of 500 matrix images containing digits 0–9. They used a Bayesian classifier to match input image features with stored templates. However, this method was limited to digit recognition only. Lin [10] proposed a character verification and image classification method for IC components on PCBs. They first applied contour border detection to locate IC regions, then used a modified LeNet-based CNN with a refinement mechanism for character verification. To handle blurred or non-character components, they modified AlexNet, removed the SoftMax layer, and recorded classification scores. A threshold was set per class, and images scoring below it were labeled as incorrect components. Chen et al. [11] developed an adaptive deep learning framework consisting of chip region segmentation, orientation correction, character extraction, and character recognition. The system removed image noise, corrected character orientation, extracted individual characters, and assembled them into complete sequences. Although effective, these methods do not generalize well to new fonts or layout changes in updated production lines and thus require frequent

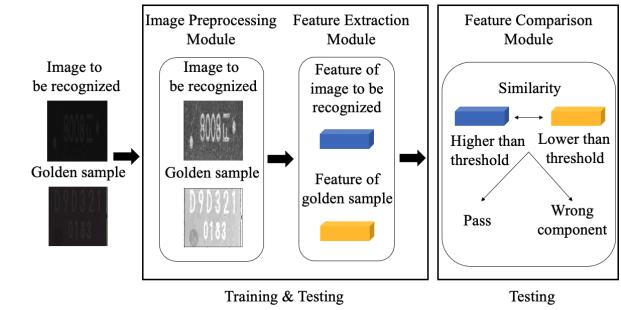


Fig. 2: Meta learning-based incorrect component detection framework.

retraining. To address font variability, Gang et al. [12] and NDAYISHIMIYE et al. [13] constructed ResNet50 and SENet models using coresets that captured features like illumination and shape variance. However, their results showed reduced accuracy when tested on unfamiliar font styles, again requiring additional training for font adaptation. However, their results showed reduced accuracy when tested on unfamiliar font styles, again requiring additional training for font adaptation.

Comparing component features is an effective approach to detect incorrect components without relying on text recognition. Metric learning-based models, e.g., siamese, matching, prototypical, and relational networks—extracted features from training data and compared similarity. In the Siamese network, triplet loss was used to train the feature extractor. The matching network employed episodic training and negative cosine distance. Prototypical network [14] extended the matching network by averaging feature vectors as category prototypes in high-dimensional space, where similar images cluster linearly. These prototypes were then used for training. The relational network [15] used a non-linear similarity metric capable of learning deep embeddings, offering greater flexibility in similarity assessment and replacing traditional end-to-end architectures. Liu et al. [16] proposed a feature extraction framework addressing data imbalance in electroluminescence image defect detection. They used metric learning to optimize the CNN feature space and added regularization terms to enhance discriminative power. Li et al. [17] implemented a classification-based metric learning model combined with scene text recognition to identify open-set logos. However, these methods were trained on clean datasets without contamination. In real-world factory settings, dirt and debris are common during manufacturing, posing challenges to model robustness.

III. METRIC LEARNING-BASED INCORRECT COMPONENT DETECTION FRAMEWORK

In this paper, we propose the MICD framework, consisting of three modules: image preprocessing, feature extraction, and feature comparison, as shown in Fig. 2. The image preprocessing module converts component images to grayscale and sharpens them to improve clarity. The feature extraction

TABLE I: Comparison of metric learning methods.

Dataset / Accuracy	Siamese Network	Matching Network	Prototypical Network	Relation Network
Omniglot miniImageNet	96.50% x	98.50% 55.31%	98.90% 68.20%	99.10% 65.32%

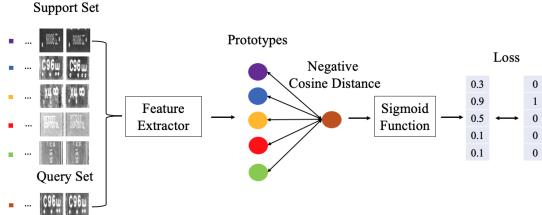


Fig. 3: Prototypical network.

module captures essential features, while the feature comparison module defines the golden sample and detects incorrect components.

A. Feature Extraction Module

A feature extractor can detect blurred or disturbed components; however, it may struggle to identify components from new products. To address this limitation, we adopt metric learning to train the feature extractor, improving its generalization to unseen components. Among the four metric learning methods—siamese, matching, prototypical, and relation networks—we select the prototypical network and relation network for training the feature extractor, as they achieve the highest accuracy on the Omniglot and miniImageNet datasets, respectively, as shown in Table I.

The feature extractor in our framework is based on an untrained DenseNet model with approximately 20 million parameters, which is sufficient for extracting features from AOI images.

In the prototypical network, as shown in Fig. 3, we randomly select five components and sample five images from each to form the support set. An additional five images from one of these components are selected as the query set. We extract features from both sets and average the vectors of each class to obtain the prototypes. The negative cosine distance between the query and support prototypes is computed and scaled to the range [0,1] using the sigmoid function, resulting in a 1×5 vector. Finally, we compute the cross entropy loss between this vector and the corresponding one-hot label to train the feature extractor, as in Eq. (1).

$$c_k = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f_\phi(x_i), \quad (1)$$

where c_k is an average of x .

In the relation network, the same procedure as in the prototypical network is used to select the support and query sets, as shown in Fig. 4. Features extracted from both sets are concatenated and passed through the relation module to

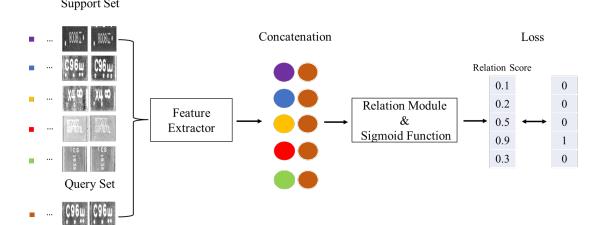


Fig. 4: Relation network.

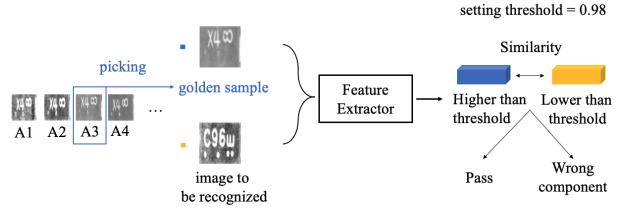


Fig. 5: Feature comparison module.

compute a relation score. The sigmoid function scales the scores to the $[0, 1]$ range, producing a 1×5 vector. Finally, the mean squared error between this vector and the corresponding one-hot label is used as the loss function to train both the feature extractor and the relation module, as defined in Eq.(2).

$$\varphi, \phi \leftarrow \underset{\varphi, \phi}{\operatorname{argmin}} \sum_{i=1}^m \sum_{j=1}^n (r_{i,j} - 1(y_i == y_j))^2 \quad (2)$$

B. Feature Comparison Module

This module detects incorrect components by comparing a test image with a golden sample. As shown in Fig. 5, a similarity threshold is used to balance the false alarm and leakage rates. The cosine similarity between the test image and the golden sample is calculated. If the similarity exceeds the threshold, the component is classified as correct; otherwise, it is marked as incorrect.

The threshold is set based on the leakage rate tolerance specified by our industrial partner. While techniques like cross-validation could further optimize it, we prioritize practical deployment to meet industry requirements while maintaining strong detection performance.

As shown in Table II, the choice of the golden sample selection method significantly affects the results. To ensure consistency and effectiveness, we adopt a total similarity-based selection approach, where the image of a component with the highest total similarity to all other images is selected as the golden sample for that component. This method enhances representativeness while maintaining selection consistency, as illustrated in Fig. 6.

IV. EXPERIMENT

Two main experiments are conducted to evaluate the performance of the proposed MICD framework. First, we assess its

TABLE II: Golden sample picking comparison.

	First Image	Second Image	Third Image
Leakage Rate	0.5%	0.5%	1%
False Alarm Rate	42.3%	51.92%	82.21
Accuracy	96.82%	96.27%	93.90%

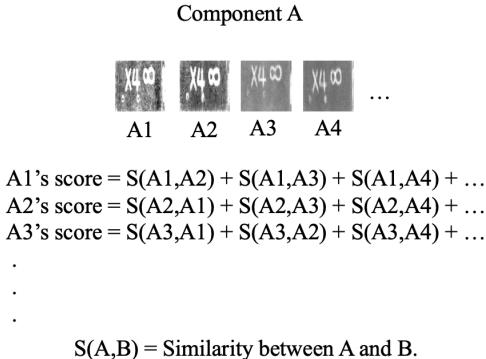


Fig. 6: Golden sample selection method.

ability to detect incorrect components across four categories and compare the results with the OCR method. Second, we apply Grad-CAM to analyze and interpret the model’s predictions.

A. Experimental Settings

In these experiments, we use a dataset provided by an SMT company, categorized into four classes: Class 1 (C1), Class 2 (C2), Class 3 (C3), and Class 4 (C4), as shown in Fig. 7. Each class contains multiple components, with up to 100 images per component: C1 including 14 components, C2 including 15, C3 including 7, and C4 including 18.

- **C3:** Most components have vague model names.
- **C2:** Many components are partially obscured by dust.
- **C1 and C4:** Components are clean and have clearly printed model names.

This categorization reflects varying levels of text clarity and cleanliness, providing a robust testbed for evaluating recognition methods. We assess the performance of MICD and OCR in identifying both existing and new components. To ensure fair evaluation, 90% of the components are randomly assigned to the training set and 10% to the testing set.

B. Comparison with OCR

As mentioned earlier, many factories adopt OCR methods to support AOI systems in detecting incorrect components. In this study, we compare the proposed MICD framework with Google Cloud OCR [3] in terms of the false alarm rate. For the OCR method, components are classified as correct if the

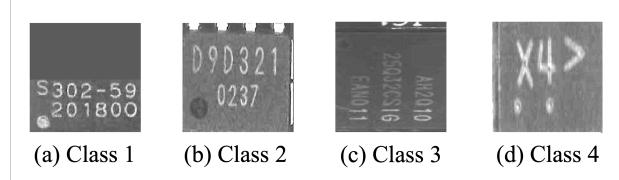


Fig. 7: Four categories of components.

TABLE III: Comparison of MICDs and OCR method.

	C1		C2		C3		C4	
	FAR ¹	LR ²						
Google Cloud OCR	48.52	0	99.11	0	94.53	0	31.74	0
PN-MICD	37.5	7.14	6.25	7	59.50	6.9	33.8	6.81
RN-MICD	39.64	6.88	8.88	6.9	40.51	7.07	49.33	7

¹ False Alarm Rate (%)

² Leakage Rate (%)

OCR result matches the default model name; otherwise, they are marked as incorrect.

As shown in Table III, the OCR method achieves a 0% leakage rate across all categories, as strict matching eliminates leakage entirely. To meet the leakage tolerance specified by our industrial partner, we set the threshold for MICD to maintain a 5%–7% leakage rate and compare its false alarm rate with that of the OCR method. In Category C3, where model names are highly blurred, the relation network-based MICD (RN-MICD) achieves a 54% lower false alarm rate than OCR. This improvement stems from OCR’s frequent misreading of blurred text, while MICD bypasses text recognition by comparing visual similarity to a golden sample. In Category C2, where components are obscured by dust, the OCR method results in a 99% false alarm rate, whereas prototypical network-based MICD (PN-MICD) reduces false alarms by 90%. This gain is attributed to MICD’s feature extractor, which is trained to recognize components under noisy visual conditions. In Categories C1 and C4, where components are clean with clearly printed text, both the OCR method and MICD perform similarly. OCR slightly outperforms MICD in C4, while MICD shows marginally better performance in C1.

C. Ablation Study

The feature extractor is trained with prototypical and relation networks. In the following experiments, we compare their performance and apply Grad-CAM for result interpretation.

1) *Architecture Comparison:* In this experiment, we analyze the false alarm rate of PN-MICD and RN-MICD across the four categories. Based on the leakage tolerance specified by the cooperating company, the threshold is set between 5%–7%.

As shown in Fig.8 and Fig.9, at a 7% leakage rate, RN-MICD achieves a 19% lower false alarm rate than PN-MICD in C3. In contrast, PN-MICD outperforms RN-MICD in C4, reducing the false alarm rate by 15%. In C1 and C2, both models show comparable performance.

2) *Grad-CAM Analysis:* We use heatmaps to verify that the feature extractor focuses on distinguishing similar components

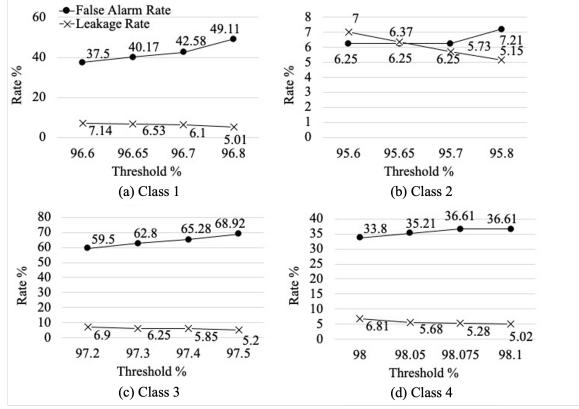


Fig. 8: Results of PN-MICD.

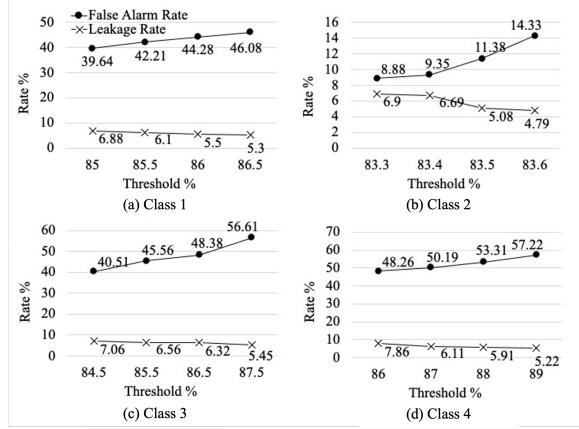


Fig. 9: Results of RN-MICD.

rather than relying solely on text elements like trademarks. Grad-CAM [4] generates heatmaps by computing gradients from the final convolutional layer, highlighting regions that influence the model's decisions. Higher color intensity indicates greater attention. Each image is divided into nine blocks, and heat percentages are calculated to identify key distinguishing regions, supporting our analysis. PN-MICD and RN-MICD show similar performance in C1 and C2. Therefore, we focus our analysis on C4, where PN-MICD outperforms RN-MICD, and C3, where RN-MICD performs better than PN-MICD.

Ideally, the heatmap should concentrate on blocks containing text and highlight distinguishing features between components. In C4, the key difference is located in block 6, which should exhibit the highest concentration. As shown in Fig. 10, the prototypical network effectively focuses on block 6 when differences occur, while the relation network incorrectly emphasizes block 5, where the text remains largely unchanged. This misdirected attention contributes to misclassification and higher leakage. Statistical analysis supports this observation: in block 6, the prototypical network accounts for 0.52%

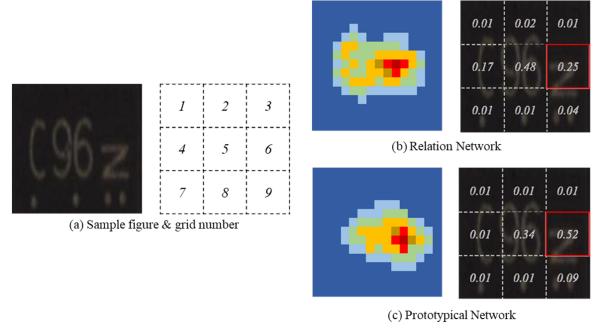


Fig. 10: MICD Grad-CAM results of C4.

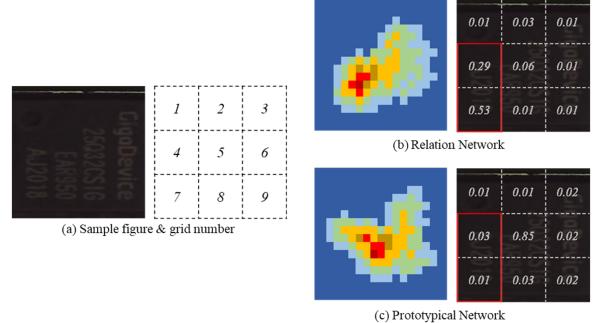


Fig. 11: MICD Grad-CAM results of C3.

of the total heat, compared to only 0.25% for the relation network. This precise focus on critical regions explains why the prototypical network outperforms the relation network in reducing the false alarm rate in C4.

In Category C3, components contain model numbers and trademarks with significant textual differences and ambiguities, primarily in blocks 4 and 7. As such, heat distribution is expected to be concentrated in these regions. As shown in Fig. 11, the relation network demonstrates ideal heat distribution, effectively focusing on blocks 4 and 7. In contrast, the prototypical network misplaces attention on blocks 5 and 8, where the text remains unchanged. Statistical analysis confirms this observation: blocks 4 and 7 account for 0.83% of total heat in the relation network, but only 0.04% in the prototypical network, highlighting the relation network's superiority in handling textual ambiguities in C3.

V. CONCLUSION

In this paper, we proposed a Metric Learning-based Incorrect Component Detection (MICD) framework to identify incorrect components without relying on text recognition, thereby mitigating misreading issues. MICD framework consists of three modules: image preprocessing, feature extraction, and feature comparison. Prototypical and relation networks are used to extract discriminative features, and a golden sample selection process further enhances detection accuracy. In challenging cases involving blurry or stained images, MICD

reduced false alarms by 54% in C3 and 90% in C2 compared to OCR. Grad-CAM analysis revealed that the prototypical network focused on critical blocks, capturing 0.52% of the total heat in the C4 category, while the relation network captured 0.82% in C3. Overall, MICD outperformed OCR methods in detecting incorrect components, particularly under blurred or dust-obscured conditions.

To integrate seamlessly with AOI systems, MICD can serve as a verification stage in the SMT pipeline, running in parallel to reduce false alarms. It is compatible with existing inspection cameras and requires minimal adjustments. Future work will focus on enhancing efficiency through lightweight feature extractors and model pruning.

ACKNOWLEDGEMENTS

This work is sponsored by the National Science and Technology Council (NSTC) under the projects NSTC 113-2222-E-008-002 and NSTC 113-2218-E-A49-027.

REFERENCES

- [1] Chia-Yu Lin, Yan-Hung Chou, and Yun-Chiao Cheng, “A deep learning-based general defect detection framework for automated optical inspection,” in *IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology*, 2023, pp. 332–337.
- [2] Mitchel M Hsu, Ming-Hsien Wu, Yun-Chieh Cheng, and Chia-Yu Lin, “An efficient industrial product serial number recognition framework,” in *IEEE International Conference on Consumer Electronics-Taiwan*, 2022, pp. 263–264.
- [3] Google Cloud, “Cloud vision api,” <https://cloud.google.com/vision>, 2025.
- [4] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 618–626.
- [5] DB Anitha and Mahesh Rao, “Smt component inspection in pcba’s using image processing techniques,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 12, pp. 541–547, 2019.
- [6] Yi-Ming Chang, Ti-Li Lin, Hung-Chun Chi, and Wei-Kai Lin, “Deep learning-based aoi system for detecting component marks,” in *2023 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2023, pp. 243–247.
- [7] Fu-Gui Tan, Fu-Min Zhou, Li-Sang Liu, Jian-Xing Li, Kan Luo, Ying Ma, Ling-Xiao Su, Hua-Liang Lin, and Zhi-Gang He, “Detection of wrong components in patch component based on transfer learning,” *Netw. Intell.*, vol. 5, no. 1, pp. 1–9, 2020.
- [8] Zhongshu Chen, Lin Zuo, Feng Guo, Changhua Zhang, and Yu Liu, “Weakly supervised end-to-end learning for inspection on multidirectional integrated circuit markings in surface mount technology,” *IEEE Transactions on Industrial Informatics*, 2023.
- [9] Carlos F Nava-Dueñas and Felix F Gonzalez-Navarro, “Ocr for unreadable damaged characters on pcbs using principal component analysis and bayesian discriminant functions,” in *IEEE International Conference on Computational Science and Computational Intelligence*, 2015, pp. 535–538.
- [10] Chun-Hui Lin, Shyh-Hau Wang, and Cheng-Jian Lin, “Using convolutional neural networks for character verification on integrated circuit components of printed circuit boards,” *Applied Intelligence*, vol. 49, no. 11, pp. 4022–4032, 2019.
- [11] Zhongshu Chen, Changhua Zhang, Lin Zuo, Tangfan Xiahou, and Yu Liu, “An adaptive deep learning framework for fast recognition of integrated circuit markings,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2486–2496, 2021.
- [12] Sumyung Gang, Ndayishimiye Fabrice, and JoonJae Lee, “Coresets for pcb character recognition based on deep learning,” in *IEEE International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2020, pp. 637–642.
- [13] Fabrice NDAYISHIMIYE, Sumyung Gang, and Joon Jae Lee, “Training data sets construction from large data set for pcb character recognition,” *Journal of Multimedia Information System*, vol. 6, no. 4, pp. 225–234, 2019.
- [14] Jake Snell, Kevin Swersky, and Richard S Zemel, “Prototypical networks for few-shot learning,” *arXiv preprint arXiv:1703.05175*, 2017.
- [15] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales, “Learning to compare: Relation network for few-shot learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.
- [16] Kun Liu, Jiangrui Han, Haiyong Chen, Haowei Yan, and Peng Yang, “Defect detection on el images based on deep feature optimized by metric learning for imbalanced data,” in *IEEE International Conference on Automation and Computing (ICAC)*, 2019.
- [17] Chenge Li, István Fehérvári, Xiaonan Zhao, Ives Macedo, and Srikanth Appalaraju, “Seetek: Very large-scale open-set logo recognition with text-aware metric learning,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2544–2553.