International Journal of Robotics and Control Systems
Vol. 5, No. 6, 2025, pp. 3300-3317
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

Lightweight Deep Learning for Real-Time Defect Detection in SMT Component Placement

Trung Nhan Nguyen [a,1,*], Phan Van Trung Tin [a,2], Thanh Quyen Ngo [a,3], Van Sy Nguyen [b,4]

[a] Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Vietnam
[b] Faculty of Automotive Engineering Technology, Industrial University of Ho Chi Minh City, Vietnam
[1] nguyentrungnhan@iuh.edu.vn; [2] trungtin8947@gmail.com; [3] ngothanhquyen@iuh.edu.vn; [4] nguyenvansy@iuh.edu.vn
[*] Corresponding Author

ARTICLE INFO

Article history
Received September 13, 2025
Revised November 21, 2025
Accepted December 27, 2025

Keywords
Surface-Mount Technology (SMT);
Pre-placement Inspection;
Lightweight CNN;
Ghost Convolution;
Knowledge Distillation;
Edge Deployment;
Real-Time Defect Detection

ABSTRACT

This study presents a lightweight deep learning approach for real-time defect detection in Surface-Mount Technology (SMT) systems, addressing key challenges in industrial quality control. A dedicated data acquisition system was developed to collect diverse component images directly from the production line, with both offline and online augmentation applied to enhance dataset robustness. Building on this foundation, a modified ResNet-18 architecture was proposed, incorporating Ghost Convolution and Knowledge Distillation to balance accuracy with computational efficiency. Experimental results demonstrate that the optimized model achieves high accuracy (96.5%) while significantly reducing model size and inference latency compared with the baseline ResNet-18. Additional optimization techniques, including quantization and weight pruning, further improved efficiency, with comparisons against MobileNetV2 confirming the competitiveness of the proposed approach. These results highlight the potential of lightweight CNN architectures for SMT component inspection under constrained resources. However, the study remains limited to a specific dataset and experimental setup, and real-world deployment on embedded platforms such as the Raspberry Pi 5 or direct integration into pick-and-place control loops requires further validation.

1. Introduction

In the Industry 4.0 era, smart systems, the Internet of Things (IoT), and artificial intelligence (AI) have become the foundation of technological innovation. In this context, **electronic circuits** play a crucial role as the core components of smart devices, ranging from phones [1]–[4], medical equipment [5]–[8], to self-driving cars [9], [10] and industrial robots [11], [12]. To meet the increasing demand for performance, reliability, and stability, Surface Mount Technology (SMT) has been widely applied to produce Printed Circuit Boards (PCBs) with high component density and absolute precision [13].

In real-world industry, producing a high-quality PCB requires the SMT production line to go through multiple complex and demanding stages, as shown in Fig. 1. These stages include Solder Paste Printing, Solder Paste Inspection (SPI), Pick & Place, Reflow Soldering, Automated Optical Inspection (AOI), X-ray Inspection (AXI), and Electrical Inspection (ICT, FCT). Errors can occur at each stage; therefore, a quality inspection system plays a crucial role in ensuring the accuracy and

reliability of the final product. However, traditional Automated Optical Inspection (AOI) has many limitations [14], as it relies on hand-crafted features and stable lighting conditions. The diversity in component size, shape, and color makes it difficult for traditional systems to maintain high accuracy in real production environments.

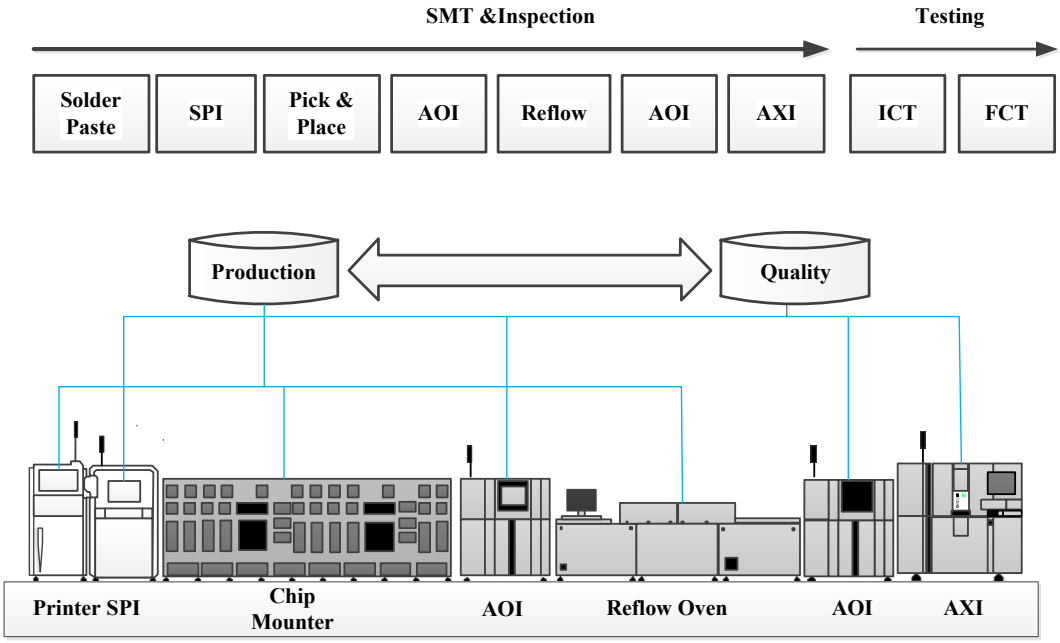


Fig. 1. Manufacturing and inspection process in the SMT production line

The outstanding development of deep learning [15] in the past decade, especially convolutional neural networks (CNNs), has opened up many new opportunities in industrial computer vision. CNNs are capable of automatically extracting features and achieving superior accuracy compared to solutions based on hand-crafted features. Thanks to this, in recent years, several studies have applied CNNs to computer vision tasks such as object detection [16]–[20], image segmentation [21]–[26], image classification [27], [28], and industrial defect detection [29].

In the context of digital production, especially in SMT production lines, learning-based methods have been applied to classify defective components. Many studies have been published in this field. For example, Kim and his collaborators [30], [31] proposed a dual-stream CNN model that simultaneously exploits the welding area, achieving superior performance compared to traditional AOI systems. In addition to AOI, another approach is pre-placement checking, which inspects components directly on the suction head of the chip mounter before placing them on the PCB. This method helps detect and correct errors such as eccentricity, angular misalignment, or component type mismatches in real time, thereby preventing defect propagation to subsequent stages. The combination of pre-placement inspection and post-placement AOI creates a complementary mechanism to ensure the overall quality of the SMT production line, establishing a dual chain of custody that covers both pre- and post-assembly stages. However, most existing AOI and pre-placement checking systems rely either on heavy deep learning models or on hardware-specific integration [32], which limits their applicability to real-time SMT production. Furthermore, deep learning methods in this domain have primarily focused on post-placement inspection, leaving a gap in lightweight solutions that can be deployed earlier in the assembly pipeline.

To address these challenges, this paper introduces a lightweight deep learning for real-time defect detection in SMT component placement. Unlike conventional AOI systems, the proposed method integrates directly with the pick-and-place process, enabling immediate detection of defects such as misalignment, polarity errors, and component mismatches before soldering. Our contributions are threefold: This research designs a compact CNN architecture optimized for high accuracy with low computational cost, making it suitable for real-time deployment. The proposed model is seamlessly

integrated into the SMT pick-and-place workflow, thereby reducing reliance on additional inspection hardware. The system is validated on real SMT production data, and the results demonstrate improved defect detection efficiency while maintaining production throughput.

2. Related work

2.1. Image classification

Image classification is one of the most important applications of computer vision, playing a key role in identification systems, automated inspection, and monitoring. Over the past decade, deep learning – particularly convolutional neural networks (CNNs) – has revolutionized image analysis by enabling automatic feature extraction with minimal manual preprocessing. Beginning with LeNet by LeCun [33], CNNs have continued to evolve across generations: AlexNet [34] marked the beginning of the deep learning era in computer vision with its breakthrough performance at ImageNet 2012, leveraging GPU acceleration; VGGNet [35] introduced a simple architecture based on 3x3 convolution layers, making the model easier to scale; GoogleNet/Inception [36] optimized parameter efficiency through the Inception modulel; ResNet [37] introduced skip connections to enable training of very deep networks; MobileNet [38] and ShuffleNet [39] targeted lightweight, high-speed models for mobile devices; EfficientNet [40] proposed a balanced scaling method for network depth, width, and resolution. Thanks to there advancements, CNNs have achieved impressive results accuracy, in some cases even surpassing human-level performance on specific tasks.

In recent years, Transformer architectures, originally introduced in natural language processing (Vaswani et al., 2017) [41], have demonstrated their effectiveness through the self-attention mechanism, which enables the model to capture global relationships among data elements. When applied to computer vision, the Vision Transformer (ViT) and its variants (such as Swin Transformer, DeiT, etc.) divide images into patches and process them similarly to word sequences, allowing the model to learn powerful global representations. Compared to CNNs, ViTs typically require larger training datasets but offer advantages in terms of generalization and scalability across diverse tasks [42]. A recent trend is the integration of CNNs and Transformers to simultaneously leverage the strong local feature learning of CNNs and the global feature representation of Transformers.

2.2. Applications in SMT Technology

In the manufacturing domain, particularly in Surface Mount Technology (SMT)—the process of mounting electronic components onto the surface of printed circuit boards (PCBs)—ensuring quality at every stage is a critical factor. Deep learning models such as Convolutional Neural Networks (CNNs) have demonstrated outstanding capabilities in industrial image processing, including product classification, surface defect detection, positional deviation inspection, and object recognition (LeCun et al. [43]; Simonyan & Zisserman [44]).

Deep learning, particularly CNNs and Transformers, has been effectively applied in inspection tasks at the Pick & Place stage. One application is missing component detection [45], where images captured by a camera on the suction head are classified as "present" or "missing" immediately after pickup. Another is component type classification [46], which ensures that the correct component is picked, avoiding confusion between visually similar parts. In addition, position and orientation estimation [47] provides correction data before placing components onto the PCB. A high-speed camera captures images of components while the suction head is in motion, and these images are processed by CNN- or Transformer-based models for classification and defect detection. When errors are detected, the machine can automatically discard the component, trigger an alert, or halt the production line.

The model training process consists of the following steps:

- Collecting image data under various lighting and speed conditions.
- Labeling the data (complete, missing, wrong type, incorrect orientation).
- Training and evaluating the model using training and testing datasets.

- Deploying the model in real time on production machines.

The integration of computer vision and deep learning not only optimizes the current inspection process but also enables real-time production monitoring and data analysis, thereby supporting automated decision-making for production line optimization (Lee & Shin [48]).

3. Methodology

In this study, image data were collected directly from industrial cameras on the SMT production line, covering cases of correct components, incorrect components, missing components, and rotational misalignment. To enhance diversity and robustness, data augmentation techniques were applied both offline and online, enabling the model to adapt to image noise, lighting variations, and component distortions. Building on this foundation, the study proposes a lightweight deep learning model based on ResNet-18, optimized for component recognition with high accuracy, fast processing speed, and practical deployability.

3.1. Data Collection

The application of deep learning in component classification and inspection has increasingly become a mainstream trend in electronics manufacturing. However, the greatest challenge in building an effective defect detection system lies in ensuring that the input data are both high-quality and diverse. In this study, image data were collected directly from the SMT production line, covering scenarios such as correct components, wrong components, missing components, positional deviations, and rotational misalignments. These cases reflect variations in size, shape, color, and environmental conditions, thereby enabling deep learning models to achieve higher recognition accuracy. Nevertheless, collecting a comprehensive dataset remains a complex task that requires carefully designed systems to ensure representativeness and reliability [49]–[52].

To address the challenge of constructing diverse and representative datasets for training deep learning systems aimed at classifying components and detecting missing parts on the suction head, a data acquisition system was deployed directly on the automatic component placement line, as illustrated in Fig. 2.

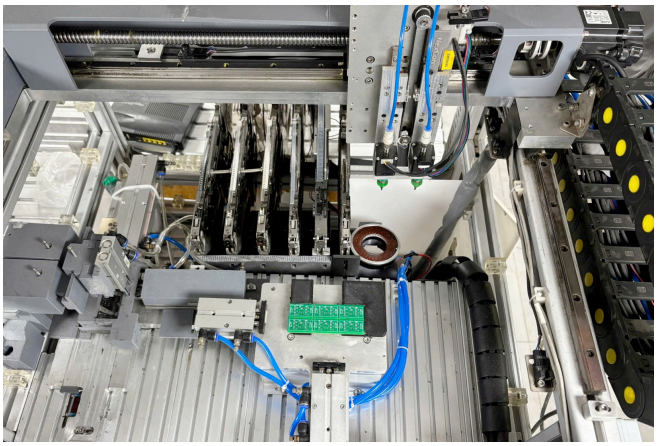


Fig. 2. Data collection system

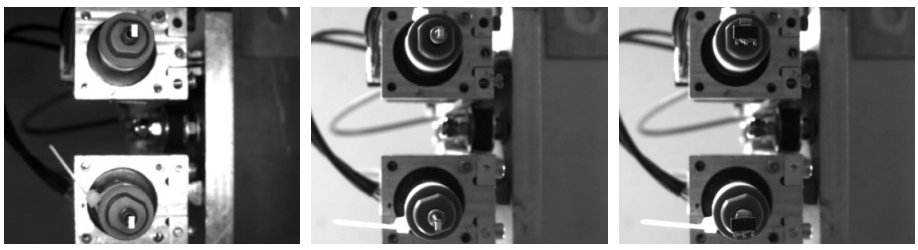
This system employs a Cognex IS8401 camera [53], mounted beneath the suction head to capture the underside of components. A C-Mount adapter with a 6 mm lens is used to ensure a wide field of view while maintaining sufficient detail for detecting anomalies such as wrong component types or positional deviations. Illumination is provided by an industrial white ring LED, optimized in both intensity and angle to maintain consistent image quality under varying production conditions. The entire system is connected to a computer equipped with an Intel Core i7-12700H CPU and an NVIDIA RTX 3050 Ti GPU, enabling real-time processing of large-scale image data. In addition, data augmentation techniques—including rotation, blurring, brightness–contrast adjustment, and Gaussian

noise injection—are applied to simulate diverse abnormal scenarios, thereby significantly enhancing the quality and diversity of the training dataset. The detailed technical specifications of the system are presented in Table 1, serving as a reference for replication in future studies.

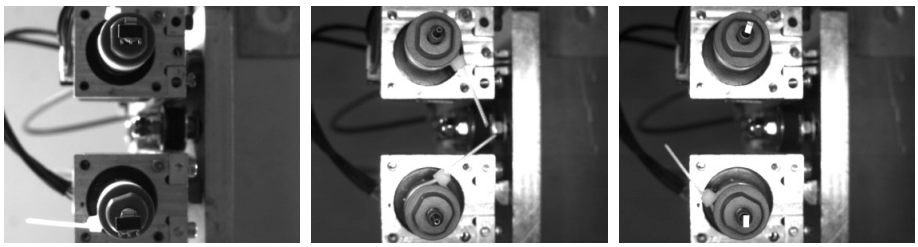
Table 1. Hardware specifications for data acquisition

Hardware Component	Specification
Camera	Cognex IS8401
Lens	C-Mount, 6 mm focal length
Lighting System	Industrial white LED ring light
Central Processing Unit (CPU)	Intel Core i7-12700H
GPU Image Processing	NVIDIA RTX 3050 Ti
Data Communication	Industrial Gigabit Ethernet
Processing Software	In-Sight Explorer 6.5.1 + Python 3.13

To address the issue that abnormal component cases on the suction head rarely occur in SMT production lines, the system was designed to automatically detect, label, and store images of such anomalies when the suction head either mispicks or fails to pick up a component. Through this mechanism, a specialized dataset of component-picking defects was constructed. Although the number of defective samples is smaller than that of correct pickups, the dataset is highly diverse in terms of error types. This dataset is crucial for training deep learning models to accurately classify and detect errors under real production conditions. At the same time, to mitigate the data imbalance between the majority class (correct pickups) and the minority class (pickup errors), the system applies data augmentation techniques such as image rotation, slight blurring, brightness/contrast adjustment, and Gaussian noise injection. These methods have been shown to improve the generalization ability of deep learning models for SMT defect detection [54], [55]. During the data collection process, the system captured and analyzed images of components for use in training and evaluating the defect detection model. Clear categorization of these cases enhances the accuracy of defect recognition and ensures the feasibility of deployment in real production environments. Fig. 3 illustrates several types of suction head errors considered in this study. Fig. 3(a) shows a correctly picked component. Fig. 3(b) presents a case where no component is picked due to pick up failure. Fig. 3(c) depicts a misaligned component that is not centered on the suction head. Fig. 3(d) shows a component that is either rotated incorrectly or is of the wrong type. Explicit classification of these states improves the accuracy of defect detection models and strengthens their practical applicability in SMT production, including missing components and type-related errors, thereby increasing the system's overall reliability in manufacturing processes.



(a) Suction head with component



(b) Misaligned component (c) Suction head without component (d) Suction head with wrong component type

Fig. 3. Typical component pickup errors on the suction head

3.2. Preprocessing and Online Data Augmentation Procedure in CNN Model Training

In SMT production, the quality of training data directly determines the accuracy of defect detection models. However, real-world defect samples are often limited. Therefore, data augmentation becomes a crucial solution to expand and diversify the training set. Transformations such as rotation, flipping, variations in lighting and color, or adding noise help simulate real fluctuations (e.g., conveyor vibration, changes in illumination), thereby enhancing the model's generalization capability. In general, there are two main approaches:

- **Offline augmentation**: preprocessing and storing augmented data in advance (e.g., rotation of ±90°, contrast adjustment).
- **Online augmentation**: applying transformations on-the-fly during training, generating new samples in each iteration.

By effectively combining these two methods, the SMT defect detection model can accurately recognize even unseen image variations, thereby improving the reliability of the entire visual inspection system and supporting AI-driven decision making (Fig. 4).

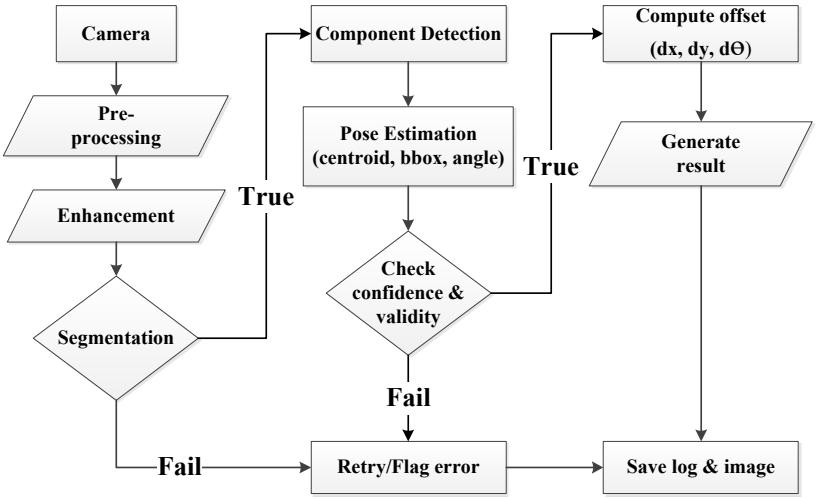


Fig. 4. AI decision-making workflow in Pick & Place (Pre-placement check)

The initial dataset (including defective and non-defective images) was collected from the existing inspection system, then augmented and used to train the CNN model. Online augmentation techniques were applied during training, allowing the model to be continuously exposed to new variants that closely resemble real defect patterns, thereby enhancing generalization capability and reducing the false positive rate. Instead of manually verifying all defects detected by the camera, the AI model trained on augmented data performs a re-evaluation step, ensuring more accurate classification of component types (Fig. 5).

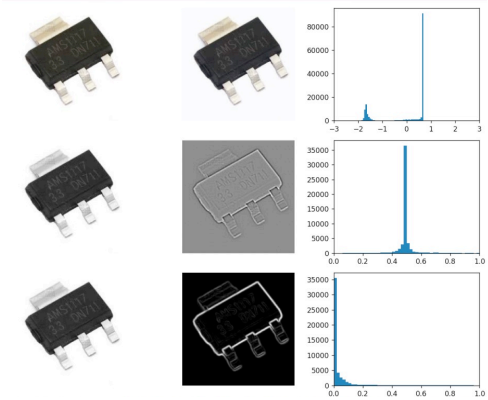


Fig. 5. Image data preprocessing

Image processing techniques include Gaussian blur, edge detection using the Laplacian transform, image intensity normalization, as well as histogram analysis of both the original and the normalized images. The application of online augmentation techniques ensures that the model continuously encounters diverse data variations, thereby significantly enhancing its generalization ability to real-world production scenarios.

3.3. CNN Model Architecture and Training Configuration

After completing data preprocessing and augmentation, the next step is to design and optimize the deep learning model. A major challenge in the SMT production environment is the variation in lighting conditions and camera configurations, which can alter color and image features, thereby affecting model performance.

To meet real-time processing requirements under limited hardware resources, we selected ResNet-18 due to its balance between accuracy and computational cost. Compared to ResNet-34 or ResNet-50, ResNet-18 is lighter with fewer parameters, reducing latency and energy consumption—critical factors for practical deployment. Our research team enhanced ResNet-18 in two main ways: (i) replacing the initial convolutional layer with Ghost Convolution to reduce computations while preserving feature extraction capability; and (ii) applying knowledge distillation so that the lightweight model can achieve accuracy close to that of the original model. These improvements enable the system to maintain high accuracy while meeting the real-time requirements of the SMT production line (Fig. 6).

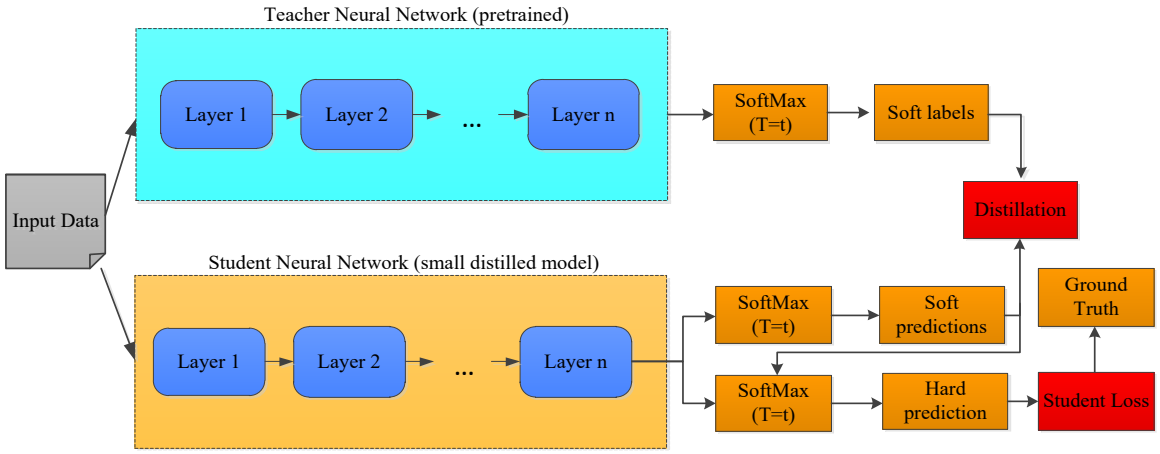


Fig. 6. Knowledge distillation technique

– Reduce the number of channels in certain layers to optimize model size and processing speed.

+ The student model applies a cross-entropy–based loss function as follows:

$$CE(q,p) = H(q,p) = -\sum_{i=1}^{C} q_i \, log(\, q_i) \tag{1}$$

To reinforce the student's predictions, the teacher's predictions are used to replace the ground-truth in the loss function. In other words, we train the model so that the probability distribution of the student's predictions is as close as possible to that of the teacher. This is similar to how a student gets better at a multiple-choice test by aligning their answers with the correct ones provided by the teacher.

Therefore, the distillation loss for observation xi will have the form:

$$Ld_{dl}(x_i; W) = H(q_{it}, q_{is}) \tag{2}$$

The combination of Ghost Convolution and knowledge distillation allows the modified ResNet-18 model to achieve competitive performance compared to deeper models, while significantly

reducing computational costs and power consumption. This enables the system to operate continuously, stably, and efficiently in an industrial environment, meeting the real-time requirements for detecting component defects on SMT camera nozzle data.

3.3.1. ResNet-18 Architecture

ResNet-18 is a convolutional neural network (CNN) architecture from the ResNet family, recognized as one of the fastest and lightest models, making it a popular choice for research, deployment, and rapid educational purposes. In this study, the authors adopted ResNet-18 as the backbone architecture for the defect detection model. The primary reason is that ResNet-18 achieves a favorable balance between accuracy and computational cost, which is particularly critical in production environments that demand real-time processing. ResNet-18 utilizes residual blocks with skip connections, which effectively mitigate the vanishing gradient problem in deep networks. This mechanism enables signals to propagate directly through the layers without degradation, thereby enhancing the training efficiency on large and complex datasets. The ResNet-18 architecture (Fig. 7) is composed of 16 convolutional layers organized into 8 residual blocks, culminating in a global average pooling layer and a fully connected layer. The first convolutional layer employs a 7×7 kernel with 64 filters and a stride of 2, followed by a 3×3 max pooling layer (stride = 2) to reduce spatial dimensions while preserving essential information. After the input stage, the network is divided into four stages, each containing two residual blocks, with each block comprising two convolutional layers using 3×3 kernels. The number of feature channels increases progressively as 64, 128, 256, and 512, allowing the model to learn features hierarchically—from low-level patterns such as edges and contours to high-level representations such as overall shapes and microstructures. Finally, the global average pooling layer reduces each feature map to a single value, which is then passed to the fully connected layer for classification. In the component classification task—covering resistors, capacitors, LEDs, ICs, and others—the final layer contains the same number of nodes as the target classes, followed by a softmax function to output prediction probabilities.

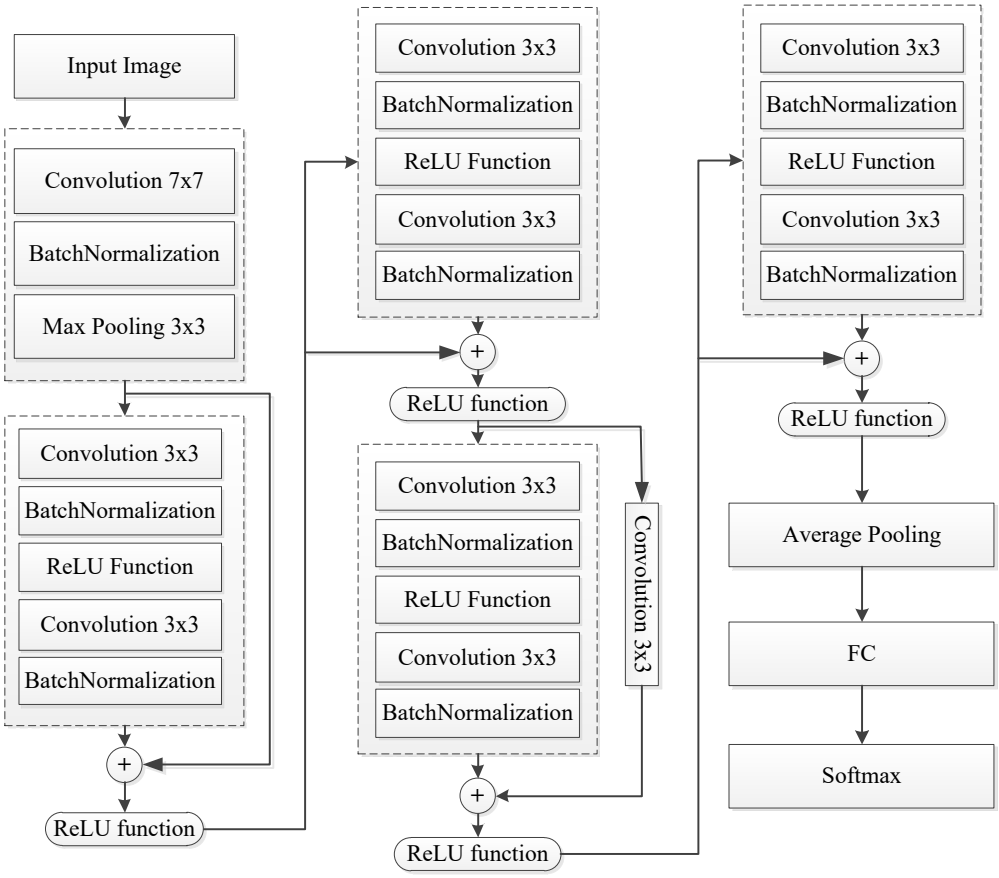


Fig. 7. ResNet-18 architecture

3.3.2. Improving the Initial Convolution Layer

To further optimize the model, this study applies Ghost Convolution in place of standard convolution at the initial layer. Unlike traditional convolution, which computes the entire set of feature maps directly, Ghost Convolution first generates a subset of intrinsic feature maps and then produces additional *ghost features* through inexpensive linear transformations. This approach offers several advantages:

- Reduces the number of convolutional operations (FLOPs).
- Preserves or even enhances the feature extraction capability.
- Decreases model size, facilitating deployment on embedded devices.

An improved solution is proposed to enhance computational efficiency while still preserving essential feature extraction capability, addressing the issues of high latency and computational overhead associated with the initial convolution in ResNet-18—commonly implemented with a 7×7 filter and stride of 2. Specifically, the initial convolution layer is modified by reducing the kernel size from 7×7 to 5×5 and increasing the stride to 4. This adjustment significantly lowers the computational burden, thereby shortening processing time. However, reducing the kernel size and increasing the stride may result in information loss at the early stage. To mitigate this, the study proposes inserting a Ghost Convolution layer [56] immediately after the adjusted 5×5 convolution. Ghost Convolution is a modern neural network optimization technique that generates additional feature maps from fewer primary maps through low-cost transformations. This approach not only reduces computational demands and model size but also maintains—or even improves—feature recognition performance (Fig. 8).

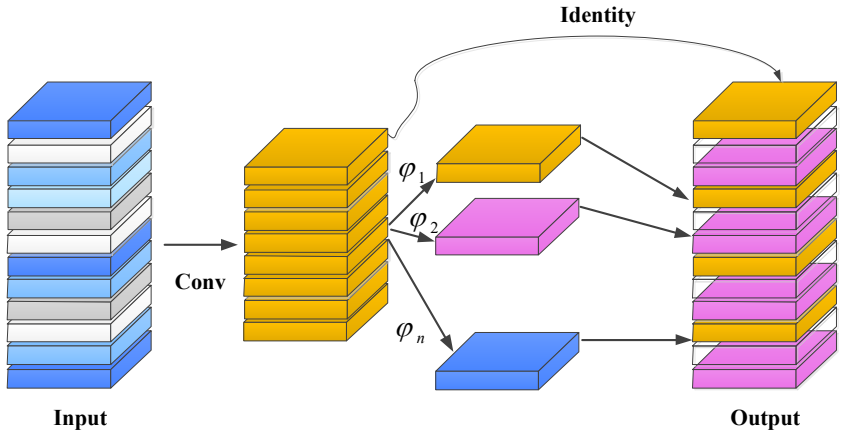


Fig. 8. Ghost convolution structure

3.3.3. Knowledge Distillation

Knowledge distillation is a training technique in deep learning where a large, powerful model (referred to as the *Teacher*) transfers its knowledge to a smaller, lighter model (the *Student*), enabling the Student to achieve performance close to that of the Teacher but with significantly lower computational cost. In this study, the authors apply Knowledge Distillation (KD) to enhance the performance of a lightweight model, making it suitable for deployment in industrial production environments.

- Teacher model: A full ResNet-18, thoroughly trained on a large-scale dataset.
- Student model: A compressed ResNet-18 variant, integrated with Ghost Convolution and reduced channel dimensions in selected layers to minimize computational cost.

Instead of learning solely from ground-truth labels (*hard labels*), the Student also learns from the probability distribution (*soft labels*) provided by the Teacher. These soft labels encode information about the Teacher's confidence and the inter-class relationships. For example, a capacitor may share

subtle similarities with a resistor (e.g., 5% probability), and this relational knowledge helps the Student capture fine-grained distinctions among electronic components.

The loss function is formulated as a weighted combination:

$$L = \alpha \cdot L_{CE}(y, p) + \beta \cdot L_{KD}(p_t, p_s, T) \quad (3)$$

where:

- L_{CE}: Cross-entropy loss between the ground-truth labels y and the Student predictions p_s.
- L_{KD}: Kullback–Leibler divergence between the Teacher predictions p_t and the Student predictions p_s, with a temperature parameter T to soften the probability distribution.
- α, β: Balance coefficients controlling the contribution of each loss component.

The advantage of this approach is that the Student model learns faster, remains lightweight, and retains most of the knowledge embedded in the Teacher. This is crucial for practical deployment in industrial scenarios, where the Student can run directly on CPUs or embedded devices (e.g., Raspberry Pi or PLC) with very low latency, meeting real-time requirements

4. Results and Discussion

In this section, the study provides a detailed description of the setup and the evaluation metrics used to assess system performance. Subsequently, the analysis delves into a comprehensive examination of the results, comparing the proposed model with both classical CNNs and other convolutional architectures. This comparison highlights the strength and efficiency of the model. Furthermore, the research team draws inferences to evaluate the impact of different modules and architectural adjustments, thereby offering additional insights into the contribution of individual components to the overall performance of the model. Finally, the team discusses the optimal deployment of the model on PCs within an automated SMT system, demonstrating its practical applicability and positive effectiveness both domestically and internationally.

4.1. Experimental Setup

For training our neural network model, designed to classify components on the suction head in the SMT system, we employed a high-performance computing setup equipped with an Intel i7-12700H CPU, 16 GB RAM, and an NVIDIA RTX 3050 Ti GPU. This hardware configuration was supported by the CUDA 12.9 Toolkit, which provides the necessary computational functions to enable efficient model training using TensorFlow.

4.2. Evaluation metrics

To accurately evaluate the performance of the model—an essential factor in ensuring its effectiveness in practical applications—we employ mean Average Precision (mAP) [57]–[59], which encompasses accuracy, precision, and recall. These metrics can be readily derived from the confusion matrix associated with the applied detection algorithm [60]. The algorithm is illustrated in Fig. 9.

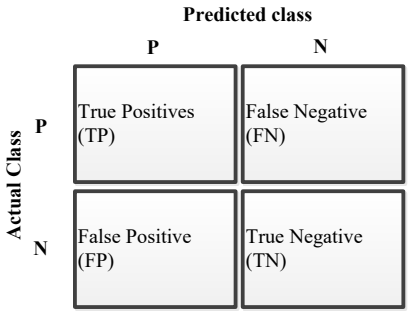


Fig. 9. Confusion matrix

In this study, measurements of accuracy, precision, and recall were conducted. For the component classification task, the results are typically labeled as positive (P) or negative (N). Four outcomes can be derived from a binary classifier. If the predicted result is P and the ground truth is also P, it is considered a true positive (TP). However, if the ground truth is N, it is considered a false positive (FP). Conversely, a true negative (TN) occurs when both the predicted and ground truth results are N, and a false negative (FN) occurs when the predicted result is N while the ground truth is P.

Based on these definitions, accuracy, precision, and recall are defined as follows:

Accuracy: $$(TP + TN)/(TP + FN + FP + TN) \quad (4)$$

Precision: $$TP/(TP + FP) \quad (5)$$

Recall: $$TP/(TP + FN) \quad (6)$$

While these metrics assess the recognition capability of the system, we also need to evaluate how precisely the components are localized on the PCB. To estimate the positional and rotational deviations of the components, we employ an affine transformation matrix [61], which allows simultaneous representation of translation along the X and Y axes as well as rotation by an angle θ.

$$\begin{bmatrix} X'_i \\ y'_i \end{bmatrix} \approx \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (7)$$

where

- θ: rotation angle
- t_x, t_y: positional translations

The rotation matrix is defined as:

$$R(\theta) = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \quad (8)$$

After obtaining the rotation matrix $R(\theta)$, we have:

$$\theta = \arctan2(R21, R11) \quad (9)$$

where

- R21 = sinθ
- R11 = cosθ
- arctan2: This ensures the correct angle estimation even when θ is negative or exceeds 90°.

The deviations of components for five samples are presented in Table 2. For instance, when the suction head picks up a component rotated by 5° from the reference orientation, the affine matrix captures this deviation through the rotation elements R11 and R21. Consequently, the model can accurately compute the deviation and provide either an alert or a corrective adjustment. Including an illustrative figure showing a correctly positioned component alongside a slightly rotated one would help readers better visualize the relationship between the mathematical formulation and the actual production errors.

Table 2. Component deviation results on the suction head

Digital Image	X Deviation (mm)	Y Deviation (mm)	Rotation Angle Deviation	Deviation (Yes/No)
1	0.12	0.08	3	No
2	0.35	0.22	5	Yes
3	0.10	0.05	0.5	No
4	0.14	0.12	2	Yes
5	0.28	0.15	5	No

4.3. Experiments

The data were collected using an industrial camera with four component types: 1206 resistors (R), capacitors (C), and isolation ICs (ISO223). Images were captured under controlled conditions, with LED ring illumination, fixed shutter speed, and component positioning. Each component type was collected under various surface conditions to increase dataset diversity. Image preprocessing included Z-score color normalization, histogram equalization, ROI cropping according to pad contours, and resizing to 224×224 px. The dataset was split into 70% training, 15% validation, and 15% testing, ensuring no sample overlap. Data augmentation was applied, including rotations of ±15°, translations up to 10 px, horizontal flipping, and brightness/contrast adjustments. The model was built on a ResNet-18 pretrained on ImageNet and fine-tuned using the Adam optimizer (learning rate = 1×10^{-3} with cosine decay, batch size = 64, up to 100 epochs, early stopping based on validation loss). Evaluation metrics included Accuracy, Precision, Recall, F1-score, FLOPs, number of parameters, and mAP@0.5 (IoU threshold = 0.5). Test set evaluation results are presented in Table 3. Beyond accuracy, to verify practical deployment on embedded environments, we measured computational cost, memory footprint, and processing speed of the model on the target hardware (Raspberry Pi 5, 224×224 images, batch = 1). These results are presented in Table 4, allowing performance comparison between the original FP32 model and optimized variants, including quantization, weight pruning, as well as a reference lightweight architecture (MobileNetV2).

Table 3. Model evaluation results on the test set

Component Type	Precision (%)	Recall (%)	F1-score (%)	AP@0.5 (%)
Resistor (R)	98.5	97.9	98.2	98.0
Capacity (C)	97.3	96.5	96.9	96.7
IC ISO223	95.5	94.8	95.1	95.0
Average	**97.1**	**96.4**	**96.73**	**96.57**

The training process employed the Adam optimizer with an initial learning rate of 1×10^{-3}, combined with a cosine decay schedule to dynamically adapt the learning rate during convergence. The batch size was set to 64, and training was run for a maximum of 100 epochs. In addition, an early stopping mechanism was applied based on the validation loss to prevent overfitting. The dataset was split into three subsets: 70% for training, 15% for validation, and 15% for testing, ensuring no overlap between samples. To improve generalization and increase data diversity, both offline augmentation (rotation of ±90°, brightness adjustment, contrast adjustment) and online augmentation (translation, Gaussian noise injection, image flipping) were applied.

Table 4 presents a detailed comparison of the baseline ResNet-18, the improved ResNet-18 (integrating Ghost Convolution and Knowledge Distillation, followed by quantization and weight pruning), and MobileNetV2.

Table 4. Comparison of computational cost and inference speed (sample)

Item	Configuration (Used in Experiments)	Parameter (M)	FLOPs (GFLOPs)	Model File Size (MB)	RAM Usage (MB)	Inference Time (ms/img)	Throughput (img/s)
1	ResNet-18 (FP32, PyTorch) — baseline	11.7	1.8	46.8	120	600	1.7
2	ResNet-18 → TFLite (int8 quantized)	11.7	~1.8	11.8	60	40	7.1
3	ResNet-18 (pruned 40%) → TFLite int8	7.0	~1.1	7.2	45	95	10.5
4	MobileNetV2 (baseline,as alternative)	3.4	0.35	13.6	70	180	5.6

The experimental results show that converting the ResNet-18 model from FP32 to TFLite int8 reduced the file size from 46.8 MB to 11.8 MB (a reduction of approximately 74.8%), while increasing the inference speed from 1.7 images/s to 7.1 images/s (about 4.2× faster). When combining 40% weight pruning with int8 quantization, the model size was further reduced to 7.2 MB, achieving an average inference time of 95 ms per image (equivalent to 10.5 images/s), while still maintaining an accuracy of 96.5%. In comparison with MobileNetV2, although MobileNetV2 has fewer parameters and lower FLOPs, its inference speed did not surpass that of the optimized ResNet-18. This demonstrates that the integration of Ghost Convolution, Knowledge Distillation, and optimization techniques such as quantization and weight pruning is an effective strategy for deploying deep learning models in industrial environments, where strict constraints on computational resources and real-time latency must be satisfied.

4.4. Results and Discussion

The machine vision inspection results presented in Fig. 10 clearly demonstrate the effectiveness of the proposed recognition system. Specifically, the system was able to accurately localize the two designated target positions on the test samples, achieving a high matching rate of 89.2% for both cases. The detected inspection regions are visually highlighted on the interface with green bounding boxes and a "+" marker precisely placed at the center of each identified position. These visual indicators not only confirm the correctness of the detection process but also provide evidence that the localization procedure was consistent and stable across repeated trials. Furthermore, the inspection interface shows that both verification checks returned a Present status, while the string verification module yielded an OK outcome. Taken together, these results indicate that the inspected products satisfied all required geometric and positional criteria, thereby meeting the established quality standards. The successful detection and verification steps further confirm that the system is reliable in differentiating acceptable products from defective ones. Another important observation is that the high recognition rate and consistent OK outcomes were achieved under actual production conditions, where environmental factors such as illumination changes typically affect image quality. The near-perfect alignment of both inspection points validates that the proposed recognition model, in combination with the optimized camera configuration, was able to maintain strong robustness against lighting variations. This demonstrates that the system is suitable for practical deployment in manufacturing environments. Moreover, the integration of the previously optimized deep learning model into the recognition pipeline contributed significantly to improving stability and minimizing false detections, ensuring that the inspection process is both accurate and efficient.

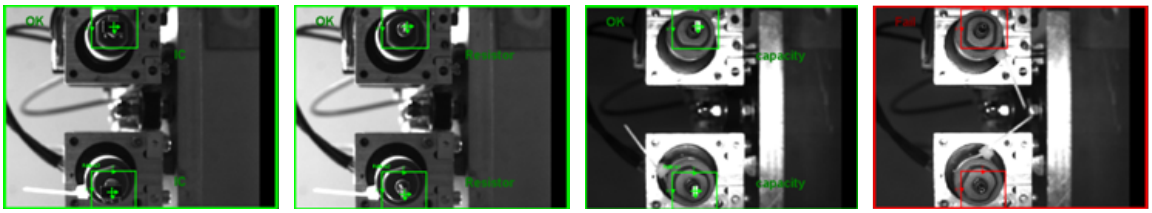


Fig. 10. Experimental results

This study demonstrates the feasibility of applying an improved ResNet-18 architecture, enhanced with Ghost Convolution and Knowledge Distillation, for component defect detection in SMT production lines. Experimental results show that the proposed model maintains high accuracy (96.5%) while significantly reducing both model size and inference time compared with the original ResNet-18. With the integration of quantization and weight pruning, the model size decreased from 46.8 MB to 7.2 MB, and inference speed improved from 1.7 images/s to 10.5 images/s, fully satisfying real-time processing requirements in industrial environments. When compared with MobileNetV2, a lightweight model widely adopted in embedded applications, the improved ResNet-18 achieved both higher accuracy (96.5% vs. 95.0%) and faster inference speed after optimization. This confirms that enhancing conventional CNN architectures with Ghost Convolution and Knowledge Distillation can be more effective than relying solely on pre-designed lightweight models. Previous studies, such as,

[62], [63], have shown that MobileNet and EfficientNet are often preferred for embedded devices due to their reduced parameter counts. However, the findings of this work demonstrate that an optimized ResNet-based model, when combined with advanced optimization techniques, can achieve comparable or even superior performance while maintaining high accuracy.

Overall, the proposed model strikes a balance between accuracy and computational efficiency, is easy to deploy thanks to improvements on the well-known ResNet-18 backbone, and effectively leverages multiple optimization techniques. Nevertheless, the dataset used remains limited in terms of component diversity, the system has not yet been fully evaluated under continuous production conditions, and further validation is required to optimize deployment on embedded devices.

5. Conclusion

This study has demonstrated the effectiveness of a deep learning system combined with computer vision for analyzing component quality on an SMT (Surface Mount Technology) production line. By integrating an advanced data acquisition system directly on the production line, we constructed a rich dataset that accurately reflects the inherent variability of real-world manufacturing processes. The application of both offline and online data augmentation techniques significantly enhanced the robustness of the dataset, enabling the model to handle diverse operational scenarios effectively. From a theoretical perspective, this research contributes by providing empirical evidence that the combination of Ghost Convolution and Knowledge Distillation can be effectively applied in practical SMT production environments. This highlights the potential of optimized CNN architectures as competitive alternatives to pre-designed lightweight models in industrial applications. Nevertheless, several limitations remain. The dataset lacked sufficient representation of rare defect types, and experiments were conducted on a single production line, which restricts the generalizability of the findings.

Future research will aim to expand the dataset across multiple factories and diverse lighting conditions, integrate Transformer–CNN hybrid architectures to enhance global feature representation, and extend the application to post-soldering AOI inspection, thereby establishing a dual-stage (pre-placement and post-placement) inspection pipeline for greater reliability in SMT production.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding.

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

References

[1] H. Matyi, P. Veres, T. Bányai, P. Tamás, and V. Demin, "Digitalization in Industry 4.0: the role of mobile devices," *Journal of Production Engineering*, vol. 23, no. 1, pp. 75–78, 2020, https://jpe.ftn.uns.ac.rs/index.php/jpe/article/view/888.

[2] A. S. Kurhade, G. D. Siraskar, P. S. Bhambare, D. K. Kaithari, S. M. Dixit, and S. Y. Waware, "Enhancing smartphone circuit cooling: A computational study of PCM integration," *Journal of Advanced Research in Numerical Heat Transfer*, vol. 27, no. 1, pp. 132–145, 2024, https://pdfs.semanticscholar.org/1f65/64f508bddea03d0b4fe332de93cccb1e248b.pdf.

[3] V. Blahnik and O. Schindelbeck, "Smartphone imaging technology and its applications," *Advanced Optical Technologies*, vol. 10, no. 3, pp. 145–232, 2021, https://www.degruyterbrill.com/document/doi/10.1515/aot-2021-0023/html.

[4] M. J. Rotheram-Borus, M. Tomlinson, D. Swendeman, A. Lee, and E. Jones, "Standardized functions for smartphone applications: Examples from maternal and child health," *International Journal of Telemedicine and Applications*, vol. 2012, no. 1, p. 973237, 2012, https://doi.org/10.1155/2012/973237.

[5] M. Javaid and A. Haleem, "Industry 4.0 applications in medical field: A brief review," *Current Medicine Research and Practice*, vol. 9, no. 3, pp. 102–109, 2019, https://doi.org/10.1016/j.cmrp.2019.04.001.

[6] P. Maresova, M. Penhaker, A. Selamat, and K. Kuca, "The potential of medical device industry in technological and economical context," *Therapeutics and Clinical Risk Management*, vol. 11, pp. 1505–1514, 2015, https://doi.org/10.2147/TCRM.S88574.

[7] A. V. Kaplan, D. S. Baim, J. J. Smith, D. A. Feigal, M. Simons, D. Jefferys, T. J. Fogarty, R. E. Kuntz, and M. B. Leon, "Medical device development," *Circulation*, vol. 109, no. 25, pp. 3068–3072, 2004, https://doi.org/10.1161/01.CIR.0000134695.65733.64.

[8] X. Xu, A. Awad, P. R.- Martinez, S. Gaisford, A. Goyanes, and A. W. Basit, "Vat photopolymerization 3D printing for advanced drug delivery and medical device applications," *Journal of Controlled Release*, vol. 329, pp. 743–757, 2021, https://doi.org/10.1016/j.jconrel.2020.10.008.

[9] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixão, F. Mutz, L. de Paula Veronese, T. Oliveira-Santos, and A. F. De Souza, "Self-driving cars: A survey," *Expert Systems with Applications*, vol. 165, p. 113816, 2021, https://doi.org/10.1016/j.eswa.2020.113816.

[10] J. Zhao, B. Liang, and Q. Chen, "The key technology toward the self-driving car," *International Journal of Intelligent Unmanned Systems*, vol. 6, no. 1, pp. 2–20, 2018, https://doi.org/10.1108/IJIUS-08-2017-0008.

[11] E. Appleton and D. J. Williams, *Industrial Robot Applications*, Open University Press Robotics Series. Dordrecht, Netherlands: Springer Netherlands, 2012, https://books.google.co.id/books?id=d1j-CAAAQBAJ.

[12] J. Wallén, *The history of the industrial robot*. Linköping, Sweden: Division of Automatic Control, Department of Electrical Engineering, Linköping University, Research Report 2853, 2008, https://rt.isy.liu.se/research/reports/2008/2853.pdf.

[13] V. Yevsieiev, S. Maksymova, and N. Starodubcev, "An automatic assembly SMT production line operation technological process simulation model development," *International Science Journal of Engineering & Agriculture*, vol. 2, no. 2, pp. 1–9, 2023, https://doi.org/10.46299/j.isjea.20230202.01.

[14] R. Ajax, "Standardizing Automated Optical Inspection (AOI) systems using deep learning for IPC Class 3 PCB compliance," *ResearchGate*, May 2025, https://www.researchgate.net/publication/392094522_Standardizing_Automated_Optical_Inspection_AOI_Systems_Using_Deep_Learning_for_IPC_Class_3_PCB_Compliance.

[15] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019, https://doi.org/10.1109/ACCESS.2019.2912200.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788, https://doi.org/10.1109/CVPR.2016.91.

[17] R. Girshick, "Fast R-CNN," *arXiv preprint*, arXiv:1504.08083, 2015, https://arxiv.org/abs/1504.08083.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587, https://doi.org/10.1109/CVPR.2014.81.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017, https://doi.org/10.1109/TPAMI.2016.2577031.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Computer Vision – European Conference on Computer Vision 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer International Publishing, 2016, pp. 21–37, https://doi.org/10.1007/978-3-319-46448-0_2.

[21] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the 2015 IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528, https://doi.org/10.1109/ICCV.2015.178.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham, Switzerland: Springer International Publishing, 2015, pp. 234–241, https://doi.org/10.1007/978-3-319-24574-4_28.

[23] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, 2017, pp. 2481–2495, https://doi.org/10.1109/TPAMI.2016.2644615.

[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the 2017 IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988, https://doi.org/10.1109/ICCV.2017.322.

[25] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6402–6411, https://doi.org/10.1109/CVPR.2019.00657.

[26] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9156–9165, https://doi.org/10.1109/ICCV.2019.00925.

[27] T. Nakazawa and D. Kulkarni, "Wafer map defect pattern classification and image retrieval using convolutional neural network," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 2, pp. 309–314, 2018, https://doi.org/10.1109/TSM.2018.2795466.

[28] H. Yang, S. Mei, K. Song, B. Tao, and Z. Yin, "Transfer-learning-based online mura defect classification," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 1, pp. 116–123, 2018, https://doi.org/10.1109/TSM.2017.2777499.

[29] Q. Xie, D. Li, J. Xu, Z. Yu, and J. Wang, "Automatic detection and classification of sewer defects via hierarchical deep learning," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1836–1847, 2019, https://doi.org/10.1109/TASE.2019.2900170.

[30] Y.-G. Kim and T.-H. Park, "SMT assembly inspection using dual-stream convolutional networks and two solder regions," *Applied Sciences*, vol. 10, no. 13, p. 4598, 2020, https://doi.org/10.3390/app10134598.

[31] Y.-G. Kim, D.-U. Lim, J.-H. Ryu, and T.-H. Park, "SMD defect classification by convolution neural network and PCB image transform," in *Proceedings of the 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, 2018, pp. 180–183, https://doi.org/10.1109/CCCS.2018.8586818.

[32] Q. Zhang, K. Zhang, K. Pan, and W. Huang, "Image defect classification of surface mount technology welding based on the improved ResNet model," *Journal of Engineering Research*, vol. 12, no. 2, pp. 154–162, 2024, https://doi.org/10.1016/j.jer.2024.02.007.

[33] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, https://doi.org/10.1109/5.726791.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems 25 (NeurIPS 2012)*, 2012, pp. 1097–1105, https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[35] S. Zhou, W. Liang, J. Li, and J.-U. Kim, "Improved VGG model for road traffic sign recognition," *Computers, Materials & Continua*, vol. 57, no. 1, pp. 11–25, 2018, https://doi.org/10.32604/cmc.2018.02617.

[36] L. Ma, H. Wu, and P. Samundeeswari, "GoogLeNet-AL: A fully automated adaptive model for lung cancer detection," *Pattern Recognition*, vol. 155, p. 110657, 2024, https://doi.org/10.1016/j.patcog.2024.110657.

[37] S. Bharati, P. Podder, M. R. H. Mondal, and V. S. Prasath, "CO-ResNet: Optimized ResNet model for COVID-19 diagnosis from X-ray images," *International Journal of Hybrid Intelligent Systems*, vol. 17, no. 1–2, pp. 71–85, 2021, https://doi.org/10.3233/HIS-210008.

[38] H. Pan, Z. Pang, Y. Wang, Y. Wang, and L. Chen, "A new image recognition and classification method combining transfer learning algorithm and MobileNet model for welding defects," *IEEE Access*, vol. 8, pp. 119951–119960, 2020, https://doi.org/10.1109/ACCESS.2020.3005450.

[39] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6848–6856, https://doi.org/10.1109/CVPR.2018.00716.

[40] G. Marques, D. Agarwal, and I. de la Torre Díez, "Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network," *Applied Soft Computing*, vol. 96, p. 106691, 2020, https://doi.org/10.1016/j.asoc.2020.106691.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS 2017)*, 2017, pp. 5998–6008, https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[42] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint*, arXiv:2010.11929, 2020, https://doi.org/10.48550/arXiv.2010.11929.

[43] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal Brain Damage," in *Proceedings of the Advances in Neural Information Processing Systems 2 (NIPS 1989)*, 1989, pp. 598–605, https://proceedings.neurips.cc/paper_files/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint*, arXiv:1409.1556, 2015, https://arxiv.org/abs/1409.1556.

[45] I. A. E. Kader, G. Xu, S. Zhang, and S. Saminu, "Brain tumor detection and classification by hybrid CNN-DWA model using MR images," *Current Medical Imaging*, vol. 17, no. 10, pp. 1248–1255, 2021, https://doi.org/10.2174/1573405617666210224113315.

[46] W. Liu, H. Sun, Z. Jia, and X. Yu, "Surface mounted devices classification using a mixture network of DCNN and DFCN," *Neurocomputing*, vol. 465, pp. 428–436, 2021, https://doi.org/10.1016/j.neucom.2021.09.011.

[47] Y. Chen, W. Gong, C. Chen, and W. Li, "Learning orientation-estimation convolutional neural network for building detection in optical remote sensing image," in *2018 Digital Image Computing: Techniques and Applications (DICTA)*, 2018, pp. 1–8, https://doi.org/10.1109/DICTA.2018.8615859.

[48] A. A. Santos, C. Schreurs, A. F. da Silva, F. Pereira, C. Felgueiras, A. M. Lopes, and J. Machado, "Integration of artificial vision and image processing into a pick and place collaborative robotic system," *Journal of Intelligent & Robotic Systems*, vol. 110, no. 4, pp. 159, 2024, https://doi.org/10.1007/s10846-024-02195-z.

[49] L. F. Tian and D. C. Slaughter, "Environmentally adaptive segmentation algorithm for outdoor image segmentation," *Computers and Electronics in Agriculture*, vol. 21, no. 3, pp. 153–168, 1998, https://doi.org/10.1016/S0168-1699(98)00037-4.

[50] S. Zhu, X. Xia, Q. Zhang, and K. Belloulata, "An image segmentation algorithm in image processing based on threshold segmentation," in *2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System*, 2007, pp. 673–678, https://doi.org/10.1109/SITIS.2007.116.

[51] Y. Song and H. Yan, "Image segmentation algorithms overview," *arXiv preprint*, arXiv:1707.02051, 2017, https://arxiv.org/abs/1707.02051.

[52] W.-X. Kang, Q.-Q. Yang, and R.-P. Liang, "The comparative research on image segmentation algorithms," in *2009 First International Workshop on Education Technology and Computer Science*, vol. 2, pp. 703–707, 2009, https://doi.org/10.1109/ETCS.2009.417.

[53] G. Tsigaras and V. Dolga, "Robot soldering positions correction with Cognex vision camera," in *New Advances in Mechanisms, Mechanical Transmissions and Robotics*, E.-C. Lovasz, I. Maniu, I. Doroftei, M. Ivanescu, and C.-M. Gruescu, Eds., Cham, Switzerland: Springer International Publishing, 2021, pp. 444–456, https://doi.org/10.1007/978-3-030-60076-1_40.

[54] J. Jang, Q. Tang, and H. Jung, "PCB defect classification with data augmentation-based ensemble method for sustainable smart manufacturing," *Sustainability*, vol. 16, no. 23, pp. 10417, 2024, https://doi.org/10.3390/su162310417.

[55] H. Wu, R. Lei, and Y. Peng, "PCBNet: A lightweight convolutional neural network for defect inspection in surface mount technology," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–14, 2022, https://doi.org/10.1109/TIM.2022.3193183.

[56] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1577–1586, https://doi.org/10.1109/CVPR42600.2020.00165.

[57] "Key Performance Indicators (KPIs)," 2021.

[58] B. Wang, "A parallel implementation of computing mean average precision," *arXiv preprint*, arXiv:2206.09504, 2022, https://arxiv.org/abs/2206.09504.

[59] P. Henderson and V. Ferrari, "End-to-end training of object class detectors for mean average precision," in *Computer Vision – ACCV 2016*, S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, Eds., Cham, Switzerland: Springer International Publishing, 2017, pp. 198–213, https://doi.org/10.1007/978-3-319-54193-8_13.

[60] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006, https://doi.org/10.1016/j.patrec.2005.10.010.

[61] A. Ghosh, A. Pananjady, A. Guntuboyina, and K. Ramchandran, "Max-affine regression: Parameter estimation for Gaussian designs," *IEEE Transactions on Information Theory*, vol. 68, no. 3, pp. 1851–1885, 2022, https://doi.org/10.1109/TIT.2021.3130717.

[62] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint*, arXiv:1704.04861, 2017, https://arxiv.org/abs/1704.04861.

[63] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *arXiv preprint*, arXiv:1905.11946, 2020, https://arxiv.org/abs/1905.11946.