



*Research article*

## **Research on a lightweight electronic component detection method based on knowledge distillation**

**Zilin Xia, Jinan Gu\*, Wenbo Wang and Zedong Huang**

School of Mechanical Engineering, Jiangsu University, Zhenjiang 212013, China

\* **Correspondence:** Email: [gujinan@tsinghua.org.cn](mailto:gujinan@tsinghua.org.cn).

**Abstract:** As an essential part of electronic component assembly, it is crucial to rapidly and accurately detect electronic components. Therefore, a lightweight electronic component detection method based on knowledge distillation is proposed in this study. First, a lightweight student model was constructed. Then, we consider issues like the teacher and student's differing expressions. A knowledge distillation method based on the combination of feature and channel is proposed to learn the teacher's rich class-related and inter-class difference features. Finally, comparative experiments were analyzed for the dataset. The results show that the student model Params (13.32 M) are reduced by 55%, and FLOPs (28.7 GMac) are reduced by 35% compared to the teacher model. The knowledge distillation method based on the combination of feature and channel improves the student model's mAP by 3.91% and 1.13% on the Pascal VOC and electronic components detection datasets, respectively. As a result of the knowledge distillation, the constructed student model strikes a superior balance between model precision and complexity, allowing for fast and accurate detection of electronic components with a detection precision (mAP) of 97.81% and a speed of 79 FPS.

**Keywords:** object detection; electronic component; knowledge distillation; lightweight

---

### **1. Introduction**

The rapid progress in artificial intelligence and intelligent manufacturing has established a solid theoretical and technical foundation for implementing automation and intelligence across various industries [1,2]. Currently, various electronic products have become an important part of people's routine life. As the core components, it is critical to insert and assemble them efficiently and accurately

onto the PCB circuit board. However, electronic components come in various types, with small sizes and high similarity, making the current manual insertion methods inefficient and prone to assembly errors. The automatic assembly of electronic components is of great significance. Electronic component detection provides information about the category and two-dimensional position of the targets, which can assist robotic arms grasping the required electronic component. Besides, the detection method can provide real-time feedback about the assembled and unassembled electronic components, facilitating the next assembly steps. Therefore, electronic component detection is a crucial step in electronic component assembly. Research on fast and accurate detection methods of electronic components based on machine vision is the prerequisite and foundation for realizing electronic component assembly.

Object detection techniques in machine vision classify and localize objects by analyzing their features in the image [3]. Hand-designed feature approaches and deep learning-based methods are the two major categories of object detection techniques. Hand-designed feature methods involve selecting regions of interest using a sliding window approach, extracting handcrafted features from these regions and finally feeding them into a classifier for object classification [4]. This approach relies on manually designed features to represent the objects in the image. In contrast, deep learning-based methods are end-to-end approaches that integrate feature extractors and classifiers into a unified model [5]. They use gradient descent methods to train the model, simultaneously learning feature representations and object classifications. In recent years, the emergence of convolutional neural networks has greatly advanced the development of object detection techniques [6]. Deep learning-based approaches offer higher accuracy, faster processing speed and improved robustness compared to Hand-designed feature object detection methods. There are two major categories that these methods fall into two-stage and one-stage. Two-stage object detection techniques create candidate regions of interest using an RPN (Region Proposal Network) and then classify and regress these regions to get the final detections, such as RCNN [7], Fast-RCNN [8] and Faster-RCNN [9]. These methods produce excellent accuracy in object detection tasks, but their real-time performance is typically low due to their numerous parameters and high computing complexity. One-stage object detection methods utilize predefined anchor boxes of various scales to directly predict the positions and categories of objects in the image, such as SSD [10] and YOLO series [11–13]. The one-stage method's performance has significantly increased, attributable to the FPN (Feature Pyramid Network) [14] and Focal Loss [15] proposals, which are more efficient in computation and have fewer parameters than the two-stage method. It also performs better in scenarios where high real-time performance is needed.

As a result, significant research has been carried out on deep learning methods for electronic component detection. Sun et al. [16] proposed an enhanced SSD method, in which they added a feature fusion module to the SSD framework. This module fuses shallow detail features with deep semantic features to enhance the detection of electronic components of different sizes. Researchers have proposed several improved algorithms to detect stacked scenes' electronic components. Huang et al. [17] introduced a method based on YOLOv3, replacing the backbone feature extraction network from DarkNet53 with MobileNet [18]. This replacement reduced the model's parameter and computation complexity, improving detection speed. Dong et al. [19] presented a method of enhanced Masked R-CNN [20], which strengthened the feature extraction network of Mask R-CNN, resulting in improved overall network performance and a slight increase in speed and accuracy. For the specific scene of PCB assembly, Li et al. [21] developed an enhanced YOLOv3. They analyzed the network's effective receptive field and, based on this, designed a prediction head suitable for the size of the electronic

components. Furthermore, Xia et al. [22] presented a high-precision electronic component detection method, which proposed an adaptive positive and negative sample matching approach based on K-Means to balance positive and negative samples during training. The model they constructed demonstrated outstanding performance in electronic component detection. Remote sensing targets are similar to electronic components, with many small targets [23]. Lei et al. [24] proposed an improved detection method based on YOLOX-Nano for remote sensing targets, which resulted in a lightweight model. In addition, other different fields also make lightweight improvements to the model to adapt to practical applications [25–27]. However, the existing detection methods based on improved SSD and YOLOv3 achieve fast detection speed with low detection accuracy. The electronic component detection method based on Mask-RCNN is relatively complex and has poor real-time performance. Electronic component detection methods based on deep learning face some challenges. For instance, they often have large model parameters, high computational complexity and limited real-time performance. Although model lightweight can reduce the number of parameters and computational complexity, it is usually accompanied by a loss of accuracy. Thus, targeted optimization measures are needed to balance detection performance and computational efficiency.

As a key technique for model lightweight, knowledge distillation typically involves a large-capacity teacher model that provides excellent performance and a student model that needs performance improvement [28]. The student model will acquire more profound knowledge and representational skills from the teacher model through knowledge distillation, which could improve its performance. Output feature distillation, intermediate feature distillation [29], structured feature distillation [30] and channel distillation [31] are several types of knowledge distillation. For output feature distillation, Hinton et al. [32] minimized the KL (Kullback-Leibler) divergence of probability distributions output of the teacher and student model classifiers. Li et al. [33] utilized L2 loss to constrain the feature maps output by the student model's RPN with those from the teacher model, achieving knowledge distillation for intermediate features. They discovered that the model's performance would suffer if the pixel-level loss were applied directly to each position of the feature maps. Liu et al. [34] constructed spatial attention maps separately for the teacher and student to achieve structured feature distillation through these attention maps. Wang et al. [35] considered that current CNN models learn the same features for the same class of pixels. To address this issue, they proposed using IFV (Inter-class Feature Variation) as a structured feature for knowledge distillation. Shu et al. [36] normalized the feature maps in each channel to obtain channel soft-label activation maps. They then performed channel distillation by minimizing the KL divergence between the channel activation maps of the teacher and student model. However, due to issues such as differences in representation between teacher and student models. Existing methods of knowledge distillation let the student model directly learn the features of the teacher model in a sub-optimal way.

In summary, current deep learning-based electronic component detection methods suffer from large model parameters and computational complexity, which makes it challenging to deploy on marginal devices and embedded systems. Even if using a lightweight model helps to reduce the parameters and computational complexity, it also leads to a drop in accuracy. Therefore, we focus on researching a lightweight electronic component detection method based on knowledge distillation. This approach aims to strike a better balance between accuracy and model complexity, achieving rapid and accurate detection of electronic components. The paper's primary contributions are as follows:

- 1) A lightweight student model for electronic component detection is constructed, and a training method based on knowledge distillation is proposed, which deals with finding a balance between

model accuracy and complexity.

2) Based on the problems of expression differences between teacher and student, and to learn the rich class-related and inter-class difference feature of the teacher. A knowledge distillation method based on the combination of feature and channel is proposed, which noticeably enhances the student model's performance.

3) Experiments on the publicly available Pascal VOC dataset and the electronic component detection dataset are performed, demonstrating the validity and robustness of the proposed approach.

The articles are arranged in the following manner. Section 2 presents the proposed lightweight electronic component detection method based on knowledge distillation, Section 3 introduces the experimental design and results and Section 4 presents the conclusions.

## 2. The proposed method

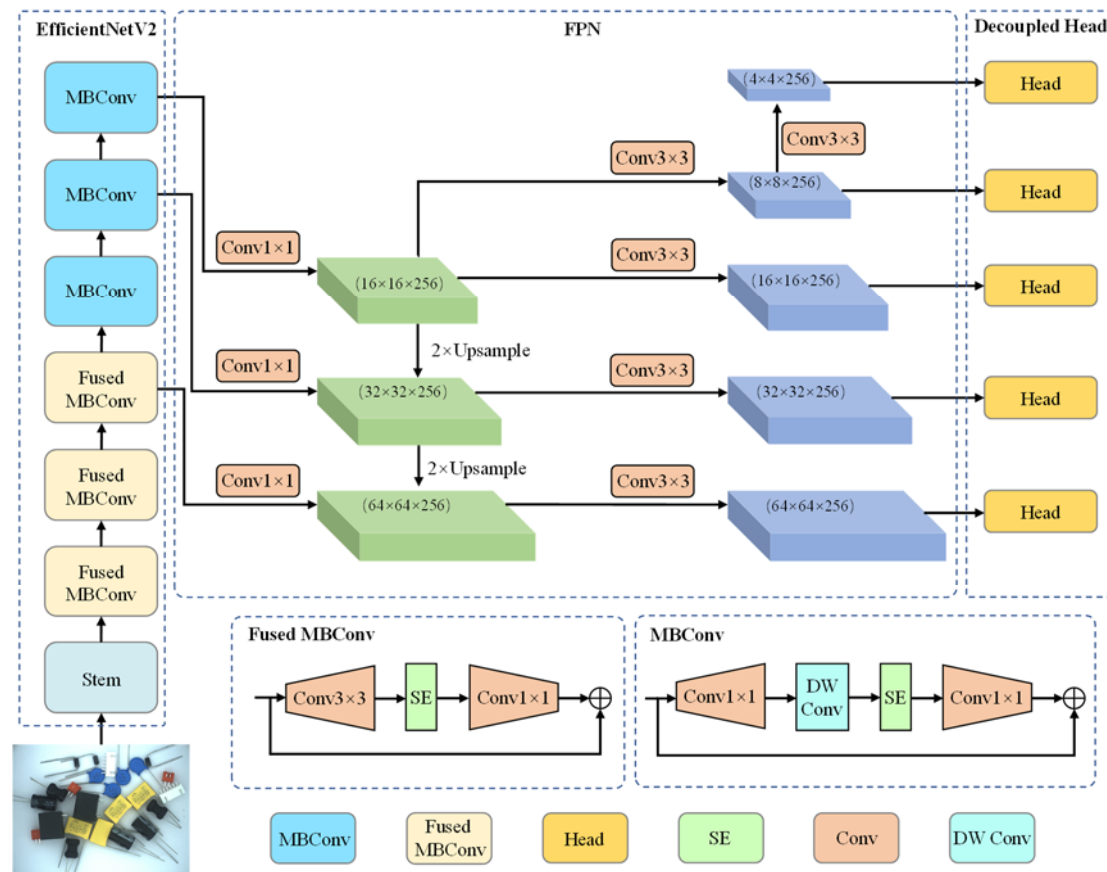
This section is divided into four subsections. Subsection 2.1 introduces the teacher model, Subsection 2.2 presents the student model, Subsection 2.3 discusses the knowledge distillation method and Subsection 2.4 describes the overall model's loss function. All specialized terms and symbols used in this paper are listed in Table 1.

**Table 1.** Explanation of specialized terms and symbols used in this paper.

Name	Description
FD	Feature knowledge distillation
FCD	Feature center distillation
FDD	Feature difference distillation
CD	Channel knowledge distillation
$F$	Output of the feature fusion network
$FC$	Feature center
$FD$	Feature difference
$MSE$	Mean Square Error
GAP	Global Average Pooling

### 2.1. Teacher model

As a crucial component in knowledge distillation, the teacher model significantly influences the quality of the results. The performance of the teacher model should be excellent, thereby providing rich semantic features. Additionally, it should demonstrate good robustness, ensuring accurate outputs for different input data. In this study, we utilize the high-precision teacher model developed by Xia et al. [22], which exhibits remarkable performance in electronic component detection and demonstrates strong generalization capabilities on the public dataset. The teacher model's overall structure, is shown in Figure 1, incorporates EfficientNetV2 [37] as the primary feature extraction network, FPN as the feature fusion network and a decoupled prediction network.



**Figure 1.** Structure of the teacher model.

## 2.2. Student model

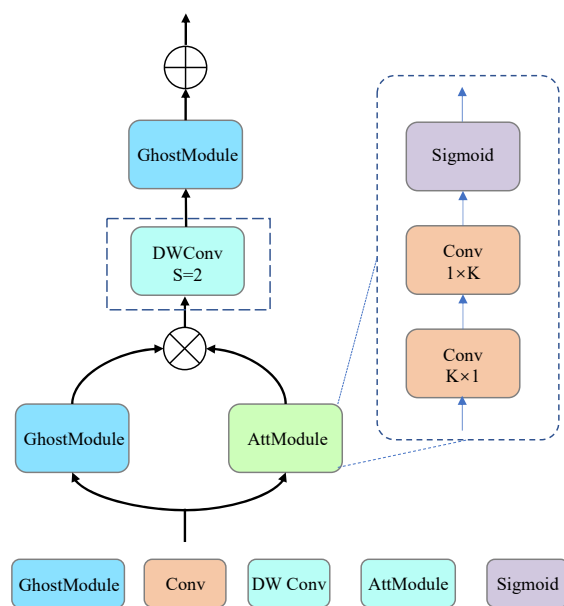
The teacher model exhibits high accuracy but lacks real-time performance, making it unsuitable for edge devices and embedded systems. The student model should be designed with minimum parameters and computational complexity since it will serve as the ultimate target model. After analyzing the model complexity of the teacher model, its main computational parameters come from the backbone and the fusion network. Therefore, we follow the lightweight idea of the Ghost module and chooses GhostNetV2 [38] serves as the backbone feature extraction network, and will be introduced in detail later. The fusion module GhostPAN is the integration of the Ghost module into the PAN (Path Aggregation Network). The prediction module remains consistent with the teacher model, using a decoupled prediction module. Figure 2 shows the general organization of the student model.

GhostNetV2 is improved based on GhostNetV1 [39] by adding a lightweight spatial attention module to GhostBlock. It slightly increases the parameter number, which can make the feature extraction capability more excellent. The overall structure of GhostNetV2 is more efficient, making it well-suited for resource-constrained devices such as edge devices and embedded systems. In Figure 3, the GhostBlock structure is displayed, where the Ghost module is a lightweight convolution module proposed in GhostNet, which can maintain the performance of the whole feature extraction module with fewer parameters. When the quantity of feature map channels in the network is extensive, many feature maps are more similar and can be obtained by a simple linear transformation. Therefore, the Ghost module divides the output channel into two parts. The first portion is created using conventional



convolutional kernel to perform convolution on the feature map generated in step 1). The computational complexity of this step is  $HW^2$ .

After completing steps 1) and 2) to obtain the spatial relationships across the entire feature map, the spatial attention map is created using a sigmoid function. The overall computational complexity for these steps is  $H^2W + HW^2$ . In contrast, directly utilizing a fully connected layer to obtain the spatial attention map would result in a computational complexity of  $H^2W^2$ . When the feature maps  $W$  and  $H$  are larger, the advantages of the lightweight attention module are more prominent, and its computational complexity is lower. Therefore, it is more suitable for capturing the spatial relationships of feature maps in lightweight networks.



**Figure 3.** GhostBlock structure diagram.

Since the student network is relatively lightweight, in order to maximize the capability of the student network, a PAN feature fusion network is chosen. PAN is a top-down and bottom-up bilateral feature fusion network, which first transfers the rich semantic information from the deeper layer to the shallower layer and fuses it with the shallow features. Then, the bottom-up layer transfers the rich location information from the shallow layer to the deep layer and fuses it with the deep feature layer. This bidirectional feature fusion network can better improve the performance of the network. To avoid introducing too many parameters, the convolutional modules in PAN are replaced with Ghost modules. This replacement reduces the parameter count and computational complexity while preserving the original feature extraction capability. Therefore, the final feature fusion network is named GhostPAN, which maintains computational efficiency and model lightweight while achieving effective fusion of rich semantic information from deep layers and rich spatial information from shallow layers.

### 2.3. Knowledge distillation method

Knowledge distillation is a training strategy that involves constructing a complex deep teacher network along with a relatively simple and shallow student network. During the training process of the

student network, the teacher network is used to guide and enhance the performance of the student network, as shown in Figure 4.

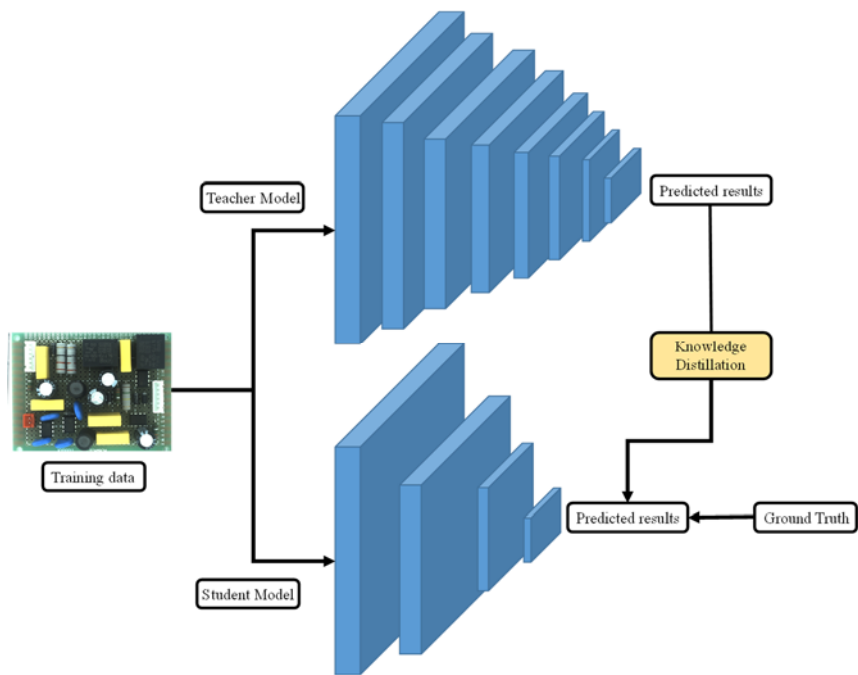


Figure 4. Knowledge distillation schematic diagram.

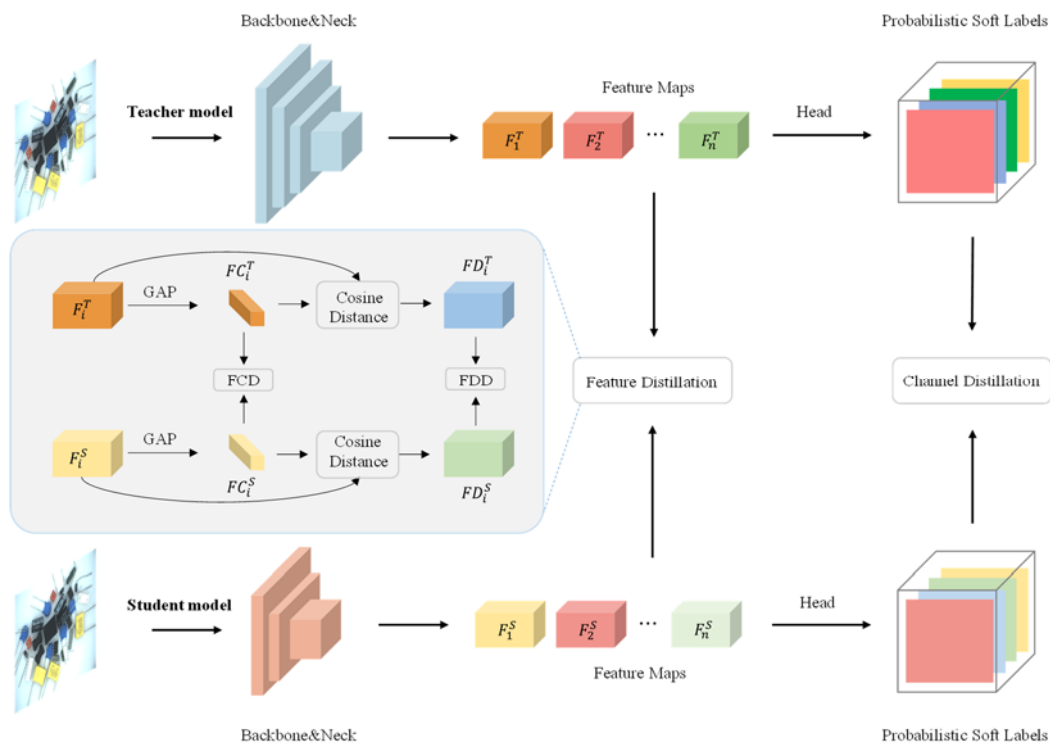


Figure 5. Structure of the combined feature and channel knowledge distillation method.



The knowledge distillation method based on the combination of feature and channel proposed in this study consists of two parts. The first part is feature knowledge distillation (FD) based on the output features of the feature fusion network, which includes feature center distillation (FCD) and feature difference distillation (FDD). The second part is channel-related knowledge distillation (CD) based on the final class predictions output by the overall network, which includes channel soft label knowledge distillation. The overall knowledge distillation structure is illustrated in Figure 5.

### 2.3.1. Feature knowledge distillation (FD)

As the teacher model is more complex and the number of network layers is deeper, its feature fusion network output features contain rich semantic information and location information. Using these outputs as supervision for the student enables it to learn deeper representations, thereby enhancing its expressive capabilities. However, letting the student directly get deeper expressions of the teacher will be counterproductive because of the significant difference between the number of layers of the two. Therefore, we propose a method to decouple the direct learning of feature-level knowledge into two parts. One part is feature center knowledge distillation, which first calculates the feature center about the teacher model and the student model output feature. Afterward, minimize the distance between their feature center. Another part is to calculate the feature differences between the output features of the teacher and student networks and their respective feature centers. Then minimize the feature difference between the two. This decoupling method is an excellent way to avoid the problem of “difference” in the expression of networks. Furthermore, it helps the student better understand the characteristics of the teacher.

#### 1) Feature center knowledge distillation (FCD)

The feature centers of the teacher model and student models' feature centers must first be obtained for feature center knowledge distillation. GAP (Global Average Pooling) is used to get the feature centers to retain the feature information better. As shown in the following Eq (1).

$$FC_i^T = GAP(F_i^T) \quad (1)$$

$F_i^T$  is the i-th feature output from the teacher model feature fusion network, where  $FC_i^T$  denotes its feature center.  $GAP(\cdot)$  is global average pooling.

After obtaining the feature centers of the teacher model as well as the student model. Then, they are constrained using the MSE (Mean Square Error). The aim is to reduce the distance between the student and teacher's feature centers. As shown in the following Eq (2).

$$MSE(FC_i^S, FC_i^T) \quad (2)$$

#### 2) Feature differences knowledge distillation (FDD)

The feature difference is expressed as the difference between the feature and its corresponding feature center. The teacher and student models feature centers have been obtained in Eq (1). The cosine distance is used to calculate the difference between the output features with their feature centers. It can better characterize the similarity between high-dimensional vectors.

$$FD_i^T = CosineSimilarity(F_i^T, FC_i^T) \quad (3)$$

As in Eq (3),  $FD_i^T$  is the feature difference between the i-th feature output from the feature fusion

module of the teacher model and its feature center.  $CosineSimilarity(\cdot)$  is the cosine distance calculation function. The  $MSE$  is then used to restrict the feature differences between the student and teacher models.

$$MSE(FD_i^S, FD_i^T) \quad (4)$$

Feature center knowledge distillation ensures that the feature centers learned by the student network are not too far away from the teacher's feature center, similar to "aligning centroids as much as possible." Feature difference distillation ensures that the feature difference learned by the student is more similar to the feature difference of the teacher, similar to "making the radii as equal as possible." These two steps make it possible to make the features learned by the student closer to those of the teacher's, thus enhancing the expressive power of the student model.

### 2.3.2. Channel knowledge distillation (CD)

The various channels in the class prediction results signify different categories of prediction data. These categories are then converted into soft labels with probabilistic values. By learning the soft labels, the student network is not only able to learn richer class-related features but also to learn the inter-class differences. Therefore, it consists of two steps, first converting class prediction information into probabilistic soft labels. Second, let the student get probabilistic soft labels from the teacher.

For each channel of the class prediction result, the Softmax function generates the soft label of the corresponding category so that the sum of the probability of each channel is 1. The response is large where there is a high correlation with the class, and small where there is a low correlation. As shown in the following Eq (5).

$$\phi(y_c) = \frac{\exp\left(\frac{y_{c,i}}{\tau}\right)}{\sum_{i=1}^{W \cdot H} \exp\left(\frac{y_{c,i}}{\tau}\right)} \quad (5)$$

where  $\phi(y_c)$  is the transformation function that converts the category prediction information into probabilistic soft labels,  $W$  and  $H$  are the corresponding feature maps' width and height.  $\tau$  is a hyperparameter. By adjusting  $\tau$ , the label can be made softer and the learning range wider.

Then KL divergence is used to reduce the gap between the probability soft label distribution of the teacher model and the student model to realize knowledge distillation. As shown in the following Eq (6).

$$KL(y^T, y^S) = \frac{\tau^2}{C} \sum_{c=1}^C \sum_{i=1}^{W \cdot H} \phi(y_{c,i}^T) \cdot \log \left[ \frac{\phi(y_{c,i}^T)}{\phi(y_{c,i}^S)} \right] \quad (6)$$

where  $y^T$  and  $y^S$  are the teacher and student model classification outputs, respectively. When  $\phi(y_{c,i}^T)$  is large,  $\phi(y_{c,i}^S)$  will correspondingly increase, and when  $\phi(y_{c,i}^T)$  is small,  $\phi(y_{c,i}^S)$  will correspondingly decrease. Therefore, KL divergence can enable the student to learn the probability distribution of the teacher, thereby improving the student's performance.

## 2.4. Loss function

---

**Algorithm 1** The lightweight electronic component detection method based on knowledge distillation

---

**Input:** Image:  $x$ , hyper-parameter:  $\lambda_1, \lambda_2, \lambda_3$ , Student:  $S$ , Teacher:  $T$ , label:  $y$

- 1: Getting the feature  $F_i^S$  and output  $\hat{P}^S$  of Image  $x$  utilizing  $S$
- 2: Getting the feature  $F_i^T$  and output  $\hat{P}^T$  of Image  $x$  utilizing  $T$
- 3: Get the class prediction results  $y^S, y^T$  in  $\hat{P}^S$  and  $\hat{P}^T$
- 4: Calculating the feature center  $FC_i^S$  and  $FC_i^T$  for  $F_i^S$  and  $F_i^T$  using Eq (1)
- 5: Calculating feature difference  $FD_i^S$  and  $FD_i^T$  using Eq (3)
- 6: Calculating the loss of student model:  $L_S(\hat{P}^S, y)$
- 7: Calculate the channel distillation loss in Eq (6):  $L_{CD}(y^S, y^T)$
- 8: Calculate the feature distillation loss in Eq (10):

$$L_{FD}(FC_i^S, FC_i^T, FD_i^S, FD_i^T, \lambda_1, \lambda_2)$$

- 9: Total knowledge distillation loss:  $L_{KD} = L_{FD} + \lambda_3 L_{CD}$
- 10: Using  $L_{total} = L_S + L_{KD}$  to update  $S$

**Output:**  $S$

---

The constructed lightweight electronic component detection method based on knowledge distillation involves only the student in the backpropagation process. The teacher, which is a trained model, does not participate in the backpropagation and is used only to provide supervisory information for the student model. Thus, the overall model contains two parts of loss. The first part is the student model loss, specifically including regression loss, classification loss and center-ness loss. The other part is knowledge distillation loss, which specifically includes feature and channel distillation loss.

$$L_{total} = L_S + L_{KD} \quad (7)$$

where  $L_S$  is the student model loss as shown in Eq (8), and  $L_{KD}$  is the knowledge distillation loss as shown in Eq (9).

$$L_S = L_{cls} + L_{loc} + L_{center-ness} \quad (8)$$

where  $L_{cls}$  is the classification loss, specifically the Focal Loss.  $L_{loc}$  is the regression loss, specifically the GIoU.  $L_{center-ness}$  is the center-ness loss [40].

$$L_{KD} = L_{FD} + L_{CD} \quad (9)$$

where  $L_{CD}$  stands for channel distillation loss, described in Eq (6), and  $L_{FD}$  stands for feature distillation loss, shown in Eq (10).

$$L_{FD} = \sum_{i=1}^Q \lambda_1 MSE(FC_i^S, FC_i^T) + \lambda_2 MSE(FD_i^S, FD_i^T) \quad (10)$$

where the first half is the feature center distillation loss and the second half is the feature difference distillation loss.  $Q$  is the number of effective feature layers output by the feature fusion module, and in this paper, it is 5 effective feature layers.  $\lambda_1$  and  $\lambda_2$  are the hyper-parameters for balancing the two parts of the loss.

The lightweight electronic component detection method based on knowledge distillation includes

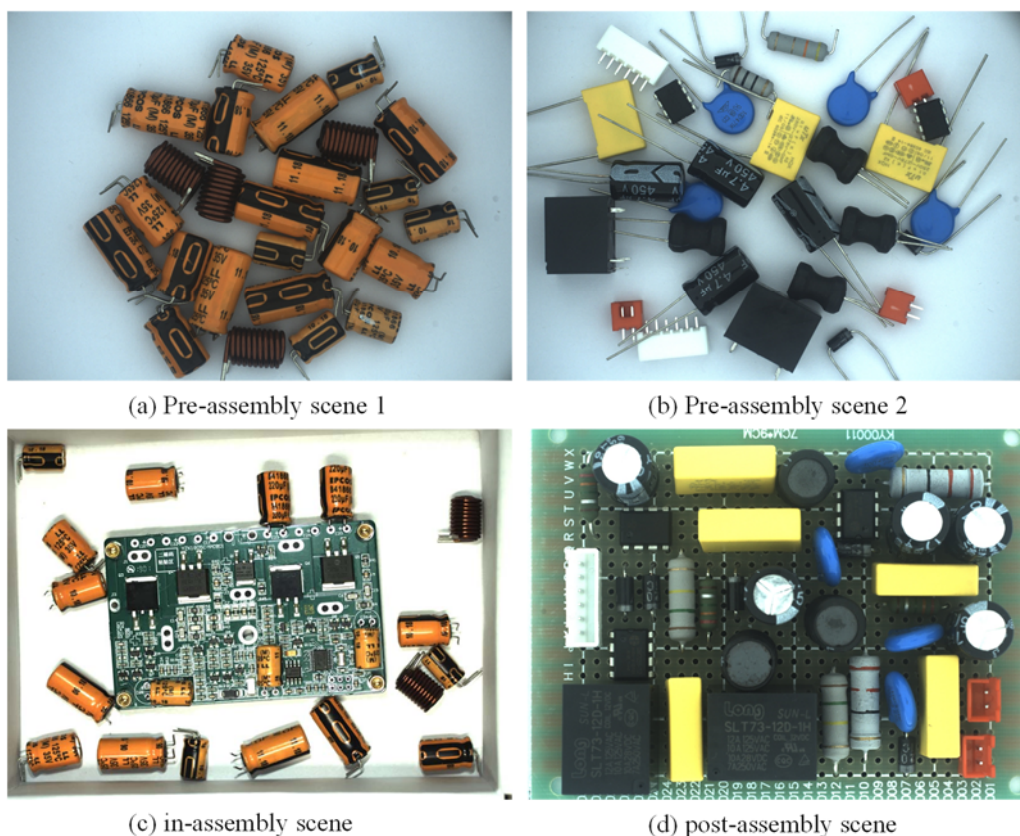
a teacher, student, knowledge distillation method and loss function. The general procedure is displayed in Algorithm 1.

In summary, a lightweight student model is proposed. It utilizes GhostNetV2 as the feature extraction network and introduces GhostPAN, a feature fusion network incorporating Ghost modules. Additionally, in order to transfer the knowledge from the teacher to the student without changing the student model structure to improve its accuracy, a knowledge distillation method based on the combination of features and channels is proposed. It performs knowledge distillation from feature centers, feature differences and feature channels.

### 3. Experiments

#### 3.1. Dataset

This study conducted extensive experiments on the electronic component detection dataset and the widely used Pascal VOC dataset to assess the presented approach's efficacy and reliability. Experiments on the electronic component detection dataset validate the performance of electronic component detection. Furthermore, the experiments on Pascal VOC allowed us to evaluate the approach's generalization ability and universality.



**Figure 6.** Samples of electronic component detection dataset. (a) and (b) are pre-assembly scenes, (c) is an in-assembly scene and (d) is a post-assembly scene.

### 1) Public dataset Pascal VOC [41]

The utilization of the public dataset offers a substantial volume of data, which effectively evaluates the model's robustness and object detection capabilities. The Pascal VOC dataset, a renowned and authoritative public dataset in object detection, is widely adopted for assessing the performance of various models, including classification, detection and segmentation. The commonly employed versions of this dataset are 2007 and 2012, containing 21 categories for comprehensive analysis.

For the training and validation sets in this work, a total of 21,380 images from the Pascal VOC 2007 training set, validation set and Pascal VOC 2012 training set are combined. Pascal VOC 2007 test set is used as the testing set, consisting of 4952 images. This approach ensures more sufficient training data and allows us to evaluate the method's performance on a larger dataset.

### 2) Electronic component detection dataset [22]

The electronic component dataset is used to evaluate the model's performance on electronic component detection. Which used in this study consists of a total of 3 assembly scenes, 14 types of electronic components and 1040 images. Figure 6 displays a few of the dataset's samples, where Figure 6(a),(b) are pre-assembly scenes, Figure 6(c) is an assembly scene and Figure 6(d) is a post-assembly scene. The specific electronic component categories can be seen in Figure 7. The different categories and their corresponding numbers in the dataset are shown in Table 2.



**Figure 7.** categories of electronic components and corresponding pictures.

**Table 2.** Different categories and numbers in the electronic components dataset.

Categories	Number	Categories	Number
Cap22uF	2121	Cap220uF	3037
Cap470uF	2226	Yellow capacitor	1781
Inductance	2341	Inductance_1	551
Relay	993	Rectifier diode	2618
Varistor	2797	White pin	1051
Resistance	4309	Chip	2328
Capacitor	2298	Red pin	1420

To guarantee the model's strong performance and generalization capability, a 7:3 data partitioning ratio was adopted in this research. Precisely, 70% of images were allocated to the training and validation, comprising 600 images for training and 140 images for validation. 30% of the images

formed the test set of 300 images. This data partitioning approach facilitates optimal data utilization during training and mitigates the risk of overfitting.

### 3.2. Experimental environment

#### 1) Experimental hardware platform

To ensure fairness in the experiments, we conducted comparative evaluations of the presented and other methods on the same hardware platform. The hardware setup included an E5-2678 V3 CPU, 16 GB of RAM and an NVIDIA 3090 graphics card with 24 GB of VRAM.

#### 2) Training parameter

The experiment is based on the PyTorch framework, and the Adam optimizer is chosen, where  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ . The learning rate was adjusted using the StepLR scheduler, where the learning rate was decreased by gamma for every “step” number of epochs. Gamma was altered in this case to 0.92 and step to 1, meaning the learning rate dropped by 0.92 after each epoch. All methods were trained for 100 episodes with an initial learning rate 0.001. Hyperparameters for the loss function were set as follows:  $\lambda_1 = 0.01$ ,  $\lambda_2 = 200$  and  $\lambda_3 = 5$ .

### 3.3. Evaluation metrics

The AP (Average Precision) and mAP (mean Average Precision) are used to evaluate accuracy. The speed is measured in terms of Params (Parameters), FLOPs (Floating Point Operations) and FPS (Frames Per Second).

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

$$Precision = \frac{TP}{TP + FP} \quad (12)$$

$$AP = \int_0^1 p(r) dr \quad (13)$$

$$mAP = \frac{\sum AP}{N} \quad (14)$$

$TP$ ,  $FN$  and  $FP$  represent the corresponding true positives, false negatives and false positives. AP is a measure of the accuracy of a single category,  $N$  stands for total categories and mAP is a measure of the average accuracy of all categories in the dataset.  $p(r)$  denotes the precision-recall curve.

Essential metrics for judging a model’s complexity and speed are FLOPs and Params. FLOPs measure the network’s computational load, while Params represents the number of parameters that can be learned from the model. Typically, higher values of FLOPs and Params indicate a more complex model, which can result in slower detection speed.

In order to ensure the reliability and accuracy of the experimental results, we adopt the strategy of conducting multiple experiments. The average value is then calculated to avoid the influence of chance, particularly for the mAP and FPS metrics.

### 3.4. Results and discussion

#### 3.4.1. Performance analysis of the student model

First, to verify the effectiveness of the proposed method in the field of general object detection, comparative experiments on the public dataset Pascal VOC are made, including mainstream object detection methods SSD, Faster-RCNN and YOLO series methods. Second, comparative experiments with other electronic component detection methods are added to the electronic component data set to verify the specificity and advancement of the proposed method.

##### 1) The public dataset

Table 3 presents a comparison of the student model with the teacher and other object detection methods on the public dataset. It can be seen that the constructed student network has a lighter structure, with 35% less computation and 55% fewer parameters than the teacher model. Compared to the computationally intensive Faster-RCNN, the student model requires approximately 16 times fewer computations. Additionally, compared to the parameter-heavy YOLOv3 and YOLOv4, the student model has approximately five times fewer parameters. However, it's important to note that while the network is lightweight, it may cause a drop in precision, with the student model achieving an accuracy of 70.16%.

**Table 3.** Comparing the performance of the teacher model and the student model with other object detection models on the public dataset.

Model	FLOPs	Params	mAP
SSD	90.54 GMac	26.42 M	71.44%
Faster-RCNN	461.76 GMac	28.48 M	76.84%
YOLOv3	49.7 GMac	61.59 M	77.73%
YOLOv4	45.37 GMac	64.05 M	80.49%
Teacher [22]	44.26 GMac	29.3 M	<b>83.44%</b>
Student	<b>28.7 GMac</b>	<b>13.32 M</b>	70.16%

##### 2) Electronic component detection dataset

Table 4 compares the student model with the teacher and other object detection methods on the electronic components detection dataset. Compared to the teacher, the student is more lightweight in structure but achieves a slightly lower accuracy of 2.15%. Compared to Huang's proposed lightweight electronic components detection method, the student model reduces the parameter count by 47%. Although the computational complexity is slightly higher, it achieves a 0.92% improvement in accuracy. Furthermore, compared to other object detection methods on the electronic components detection dataset, the student model is more lightweight while maintaining a high accuracy of 96.68%. This accuracy is higher than classical object detection methods such as SSD, Faster-RCNN and YOLOv4. The student model achieves the highest FPS, with improvements of 10 frames per second compared to SSD, 32 frames per second compared to YOLOv3 and 34 frames per second compared to YOLOv4. Compared to Faster-RCNN, there is a substantial improvement of 57 frames per second. Compared to the methods proposed by Huang and Li, there are improvements of 10 and 37 frames per second, respectively.

**Table 4.** Comparing the performance of the teacher model and the student model with other object detection models on the electronic components detection dataset.

Model	FLOPs	Params	FPS	mAP
SSD	90.54 GMac	26.42 M	69	95.67%
Faster RCNN	461.76 GMac	28.48 M	22	96.19%
YOLOv3	49.7 GMac	61.59 M	47	97.98%
YOLOv4	45.37 GMac	64.05 M	35	88.40%
Huang [17]	<b>15.46 GMac</b>	25.24 M	70	95.76%
Li [21]	53.97 GMac	61.86 M	42	88.56%
Teacher [22]	44.26 GMac	29.3 M	41	<b>98.83%</b>
Student	28.7 GMac	<b>13.32 M</b>	<b>79</b>	96.68%

Based on the comprehensive analysis and discussion, the teacher model demonstrates high accuracy and outstanding performance in the electronic components detection task. Furthermore, it can extract rich feature information, but its Params and FLOPs are high, making it unsuitable for edge and embedded devices. On the other hand, the constructed student model is lightweight, significantly reducing the Params and FLOPs compared to the teacher model, making it suitable for devices with limited computational resources. However, as compared to the teacher model, it is less accurate. Thus to increase accuracy, it needs to use the knowledge distillation approach to allow the student model to learn the extensive feature information of the teacher model.

### 3.4.2. Performance analysis and discussion of knowledge distillation methods

To validate the effectiveness of the proposed knowledge distillation method and assess whether it can improve the accuracy while keeping the student model structure unchanged. This section conducts comparative experiments on the Pascal VOC and electronic component datasets. These experiments provide a visual understanding of the changes in the student model's accuracy before and after knowledge distillation.

**Table 5.** Performance of the proposed knowledge distillation method on the public dataset Pascal VOC.

Model	FLOPs	Params	mAP
Teacher	44.26 GMac	29.3 M	83.44%
Student	28.7 GMac	13.32 M	70.16%
After distillation	28.7 GMac	13.32 M	74.07%

Tables 5 and 6 present the proposed knowledge distillation method's performance on the public and electronic components detection datasets, respectively. As observed, the proposed knowledge distillation method based on the combination of feature and channel operates superbly. It enhances the mAP of the student by 3.91% on the public dataset and by 1.13% on the electronic components detection dataset. The final accuracy of the student model on the electronic components detection



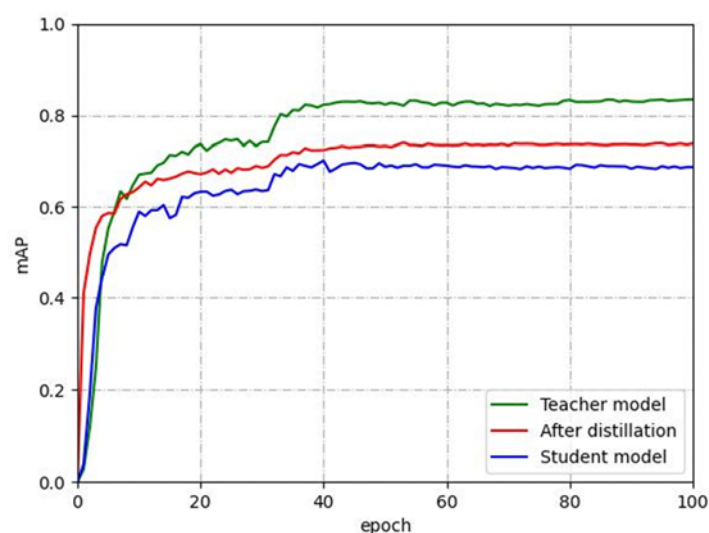
dataset reaches 97.81%, demonstrating its capability to fulfill the need for fast and accurate detection of electronic components. The constructed student model has the highest FPS, reaching 79 frames per second, which can meet real-time detection requirements. Compared to the teacher network FPS, it has significantly improved by 38, indicating that the student network is more lightweight. Table 7 shows the detection accuracy of different categories of the student model after knowledge distillation. Except for Cap22uF, Cap470uF and Cap220uF, the detection accuracy of the remaining 11 categories exceeds 98%. Therefore, after knowledge distillation, the student model demonstrates strong detection performance.

**Table 6.** Performance of the proposed knowledge distillation method on the electronic components detection dataset.

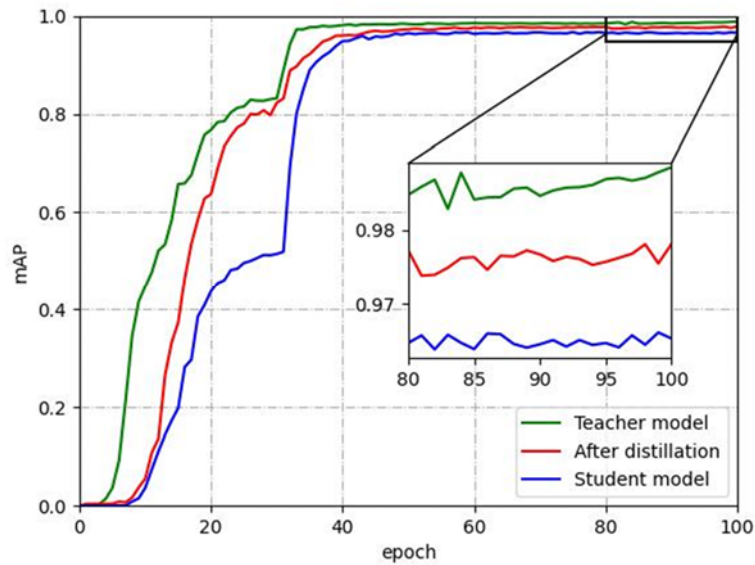
Model	FLOPs	Params	FPS	mAP
Teacher	44.26 GMac	29.3 M	41	98.83%
Student	28.7 GMac	13.32 M	79	96.68%
After distillation	28.7 GMac	13.32 M	79	97.81%

**Table 7.** Different categories and AP of student model after knowledge distillation.

Categories	AP	Categories	AP
Cap22uF	0.94	Cap220uF	0.95
Cap470uF	0.95	Yellow capacitor	1.0
Inductance	0.99	Inductance_1	0.98
Relay	0.99	Rectifier diode	0.98
Varistor	0.98	White pin	0.99
Resistance	0.98	Chip	1.0
Capacitor	0.98	Red pin	0.98



**Figure 8.** Comparison of mAP on the public dataset Pascal VOC.



**Figure 9.** Comparison of mAP on the electronic components detection dataset.

Figures 8 and 9 show the mAP comparison of the teacher, student and student models trained with the proposed knowledge distillation method on the public and electronic components detection datasets, respectively. From the figures, it is evident that the presented knowledge distillation approach not only improves the performance of the student model but also accelerates the convergence of the model.

### 3.4.3. Ablation experiments of the proposed knowledge distillation method

To further validate the effectiveness of each part of the proposed knowledge distillation method, this section conducts ablation experiments to explore them separately, which allows for a visual understanding of the contributions of each part to the overall accuracy improvement.

The teacher model's accuracy is increased through the proposed knowledge distillation method based on feature and channel fusion by transferring high-precision knowledge to the student model. This part conducts ablation experiments for analysis and discussion to verify the efficacy of each element of the proposed knowledge distillation approach based on the combination of feature and channel. First, feature knowledge distillation method experiments are conducted, and then channel knowledge distillation is added to verify the performance of feature distillation and channel distillation, respectively. The results of the ablation experiments on the public dataset are displayed in Table 8; the results of the experiments on the electronic component detection dataset are displayed in Table 9.

According to Table 8, on the public dataset, the feature knowledge distillation method improves the mAP of the student model by 1.26%. When combined with channel knowledge distillation, it further increases by 2.65%. The overall knowledge distillation method based on the combination of feature and channel improves the mAP of the student model by 3.91%, significantly improving the precision of the student model. It effectively compensates for the accuracy loss caused by the lightweight model, achieving a better balance between speed and precision.

**Table 8.** Results of ablation experiments for the public dataset Pascal VOC.

Knowledge distillation method	mAP
Teacher: FLOPs (44.26 GMac), Params (29.3 M), mAP (83.44%)	
Student: FLOPs (28.7 GMac), Params (13.32 M), mAP (70.16%)	
Feature distillation	71.42% (+1.26%)
Feature distillation + Channel distillation	74.07% (+3.91%)

According to Table 9, on the electronic components detection dataset, the feature knowledge distillation method improves the mAP of the student model by 0.74%. When combined with channel knowledge distillation, it further increases by 0.39%. The overall knowledge distillation method based on the combination of feature and channel improves the mAP of the student model by 1.13%, ultimately achieving a mAP of 97.81% on the electronic components detection dataset. Therefore, the knowledge distillation method based on the combination of feature and channel can significantly enhance the precision of the student, resulting in an outstanding performance on electronic component detection. It enables the student model to achieve fast and accurate detection, making it highly effective in electronic component detection.

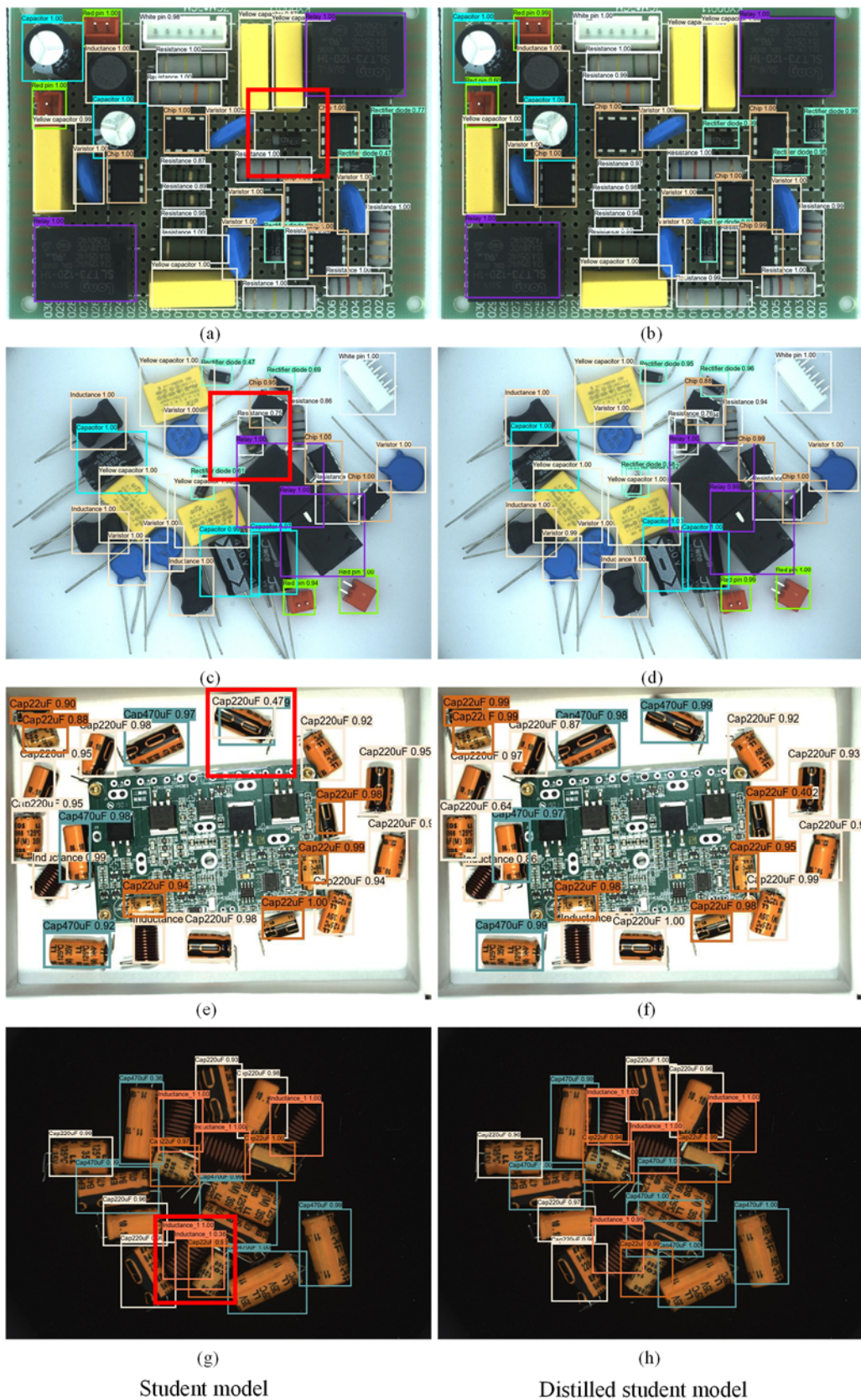
In conclusion, the proposed knowledge distillation method demonstrates excellent performance. It enables the student to effectively learn the feature representation from the teacher, significantly improving its precision.

**Table 9.** Results of ablation experiments for the electronic component detection dataset.

Knowledge distillation method	mAP
Teacher: FLOPs (44.26 GMac), Params (29.3 M), mAP (98.83%)	
Student: FLOPs (28.7 GMac), Params (13.32 M), mAP (96.68%)	
Feature distillation	97.42% (+0.74%)
Feature distillation + Channel distillation	97.81% (+1.13%)

#### 3.4.4. Visual analysis and discussion

Figure 10 compares the partial detection results of the student model before and after knowledge distillation on the electronic components dataset. From Figure 10, it can be observed that knowledge distillation effectively mitigates issues such as missing detections and redundant detections. In Figure 10(a), the rectifier diode is undetected, while in Figure 10(c), two resistances are mistakenly detected as one. Moreover, Figure 10(e),(g) show instances of redundant detections. After knowledge distillation, these problems are significantly reduced, proving the validity of the proposed method in achieving fast and accurate detection of electronic components.



**Figure 10.** Comparison of detection results before and after knowledge distillation. (a), (c), (e) and (g) represent the detection results of the student model before knowledge.

### 3.4.5. Discussion of limitations

While we have constructed a lightweight student model and proposed knowledge distillation to enhance its accuracy, a performance gap exists between the student and teacher models. As shown in Figure 7, for classes like Cap22uF, Cap220uF and Cap470uF, their inter-class features are relatively similar, resulting in detection accuracies below the average, specifically 94%, 95% and 95%, respectively. Therefore, in the future, we will further explore the internal mechanisms of knowledge distillation and consider ways to address the issue of low detection accuracy caused by inter-class similarities and intra-class differences in the dataset.

## 4. Conclusions

This study introduces a novel lightweight object detection method based on knowledge distillation, enabling swift and precise detection of electronic components. By utilizing the knowledge distillation method based on the combination of feature and channel, the student model effectively learns the feature representation from the teacher model, thus improving its performance. The following are the key research conclusions.

1) A lightweight student model is constructed. Compared with the teacher model, its Params are reduced by 55%, FLOPs are reduced by 35% and the detection accuracy on the electronic component detection dataset reaches 96.68%.

2) A knowledge distillation method based on the combination of feature and channel is proposed. It can improve the mAP of the student model by 3.91% and 1.13% on the publicly available dataset Pascal VOC and the electronic components detection dataset, respectively. The student model's ultimate detection accuracy is 97.81%, making detecting electronic components quickly and precisely possible.

In the future, we plan to conduct more in-depth research into the internal mechanisms of knowledge distillation, with the aim of further improving the accuracy of the student model. Given the limited computational resources available in the manufacturing process, we will also work on reducing the model's complexity through pruning and quantization, ensuring its suitability for edge and embedded devices. Moreover, in the future, we will apply this method to optoelectronic chip defect detection tasks to verify its effectiveness and advancement.

### Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

### Acknowledgments

This project is financially supported by the National Natural Science Foundation of China (No. 52375499, 52105516).

### Conflict of interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

1. W. Wang, Y. Zhang, J. Gu, J. Wang, A proactive manufacturing resources assignment method based on production performance prediction for the smart factory, *IEEE Trans. Ind. Inf.*, **18** (2018), 46–55. <https://doi.org/10.1109/TII.2021.3073404>
2. W. Wang, T. Hu, J. Gu, Edge-cloud cooperation driven self-adaptive exception control method for the smart factory, *Adv. Eng. Inf.*, **51** (2022), 101493. <https://doi.org/10.1016/j.aei.2021.101493>
3. Z. Zou, K. Chen, Z. Shi, Y. Guo, J. Ye, Object detection in 20 years: A survey, *Proc. IEEE*, **111** (2023), 257–276. <https://doi.org/10.1109/JPROC.2023.3238524>
4. Y. Xu, G. Yu, X. Wu, Y. Wang, Y. Ma, An enhanced Viola-Jones vehicle detection method from unmanned aerial vehicles imagery, *IEEE Trans. Intell. Transp. Syst.*, **18** (2016), 1845–1856. <https://doi.org/10.1109/TITS.2016.2617202>
5. L. Liu, J. Liang, J. Wang, P. Hu, L. Wan, Q. Zheng, An improved YOLOv5-based approach to soybean phenotype information perception, *Comput. Electr. Eng.*, **106** (2023), 108582. <https://doi.org/10.1016/j.compeleceng.2023.108582>
6. A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM*, **60** (2017), 84–90. <https://doi.org/10.1145/3065386>
7. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (2014), 580–587. <https://doi.org/10.1109/CVPR.2014.81>
8. R. Girshick, Fast R-CNN, in *Proceedings of the IEEE International Conference on Computer Vision*, (2015), 1440–1448. <http://dx.doi.org/10.1109/ICCV.2015.169>
9. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Trans. Pattern Anal. Mach. Intell.*, **39** (2017), 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
10. F. Zeng, Y. Liu, Y. Ye, J. Zhou, X. Liu, A detection method of edge coherent mode based on improved SSD, *Fusion Eng. Design*, **179** (2022), 113141. <https://doi.org/10.1016/j.fusengdes.2022.113141>
11. J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, (2017), 7263–7271. <https://doi.org/10.1109/CVPR.2017.690>
12. J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, preprint, arXiv:1804.02767.
13. A. Bochkovskiy, C. Y. Wang, H. Y. M. Liao, YOLOv4: Optimal speed and accuracy of object detection, preprint, arXiv:2004.10934
14. T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, (2017), 2117–2125. <https://doi.org/10.1109/CVPR.2017.106>
15. T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, *IEEE Trans. Pattern Anal. Mach. Intell.*, **42** (2020), 318–327. <https://doi.org/10.1109/TPAMI.2018.2858826>
16. X. Sun, J. Gu, R. Huang, A modified SSD method for electronic components fast recognition, *Optik*, **205** (2020), 163767. <https://doi.org/10.1016/j.ijleo.2019.163767>

17. R. Huang, J. Gu, X. Sun, Y. Hou, S. Uddin, A rapid recognition method for electronic components based on the improved YOLOv3 network, *Electronics*, **8** (2019), 825. <https://doi.org/10.3390/electronics8080825>
18. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (2018), 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>
19. Z. Yang, R. Dong, H. Xu, J. Gu, Instance segmentation method based on improved mask R-CNN for the stacked electronic components, *Electronics*, **9** (2020), 886. <https://doi.org/10.3390/electronics9060886>
20. K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, *IEEE Trans. Pattern Anal. Mach. Intell.*, **42** (2020), 386–397. <http://dx.doi.org/10.1109/TPAMI.2018.2844175>
21. J. Li, J. Gu, Z. Huang, J. Wen, Application research of improved YOLOv3 algorithm in PCB electronic component detection, *Appl. Sci.*, **9** (2019), 3750. <https://doi.org/10.3390/app9183750>
22. Z. Xia, J. Gu, K. Zhang, W. Wang, J. Li, Research on Multi-scene electronic component detection algorithm with anchor assignment based on K-means, *Electronics*, **11** (2022), 514. <https://doi.org/10.3390/electronics11040514>
23. L. Yang, G. Yuan, H. Zhou, H. Liu, J. Chen, H. Wu, RS-YOLOx: A high-precision detector for object detection in satellite remote sensing images, *Appl. Sci.*, **12** (2022), 8707. <https://doi.org/10.3390/app12178707>
24. L. Yang, G. Yuan, H. Wu, W. Qian, An ultra-lightweight detector with high accuracy and speed for aerial images, *Math. Biosci. Eng.*, **20** (2023), 13947–13973. <https://doi.org/10.3934/mbe.2023621>
25. W. Wang, Z. Han, T. R. Gadekallu, S. Raza, J. Tanveer, C. Su, Lightweight blockchain-enhanced mutual authentication protocol for UAVs, *IEEE Internet Things J.*, **2023** (2023). <https://doi.org/10.1109/JIOT.2023.3324543>
26. J. Zong, C. Wang, J. Shen, C. Su, W. Wang, ReLAC: Revocable and lightweight access control with blockchain for smart consumer electronics, *IEEE Trans. Consum. Electron.*, **2023** (2023). <https://doi.org/10.1109/TCE.2023.3279652>
27. L. Zhao, H. Huang, W. Wang, Z. Zheng, An accurate approach of device-free localization with attention empowered residual network, *Appl. Soft Comput.*, **137** (2023), 110164. <https://doi.org/10.1016/j.asoc.2023.110164>
28. J. Chen, Y. Liu, J. Hou, A lightweight deep learning network based on knowledge distillation for applications of efficient crack segmentation on embedded devices, *Struct. Health Monit.*, **2023** (2023), 107200. <https://doi.org/10.1177/14759217221139730>
29. A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, Y. Bengio, Fitnets: Hints for thin deep nets, preprint, arXiv:1412.6550.
30. Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, J. Wang, Structured knowledge distillation for semantic segmentation, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (2019), 2604–2613. <https://doi.org/10.1109/CVPR.2019.00271>
31. Z. Zhou, C. Zhuge, X. Guan, W. Liu, Channel distillation: Channel-wise attention for knowledge distillation, preprint, arXiv:2006.01683.
32. G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, preprint, arXiv:1503.02531.

33. Q. Li, S. Jin, J. Yan, Mimicking very efficient network for object detection, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, (2017), 6356–6364. <https://doi.org/10.1109/CVPR.2017.776>
34. Y. Liu, C. Shu, J. Wang, C. Shen, Structured knowledge distillation for dense prediction, *IEEE Trans. Pattern Anal. Mach. Intell.*, **45** (2023), 7035–7049. <https://doi.org/10.1109/TPAMI.2020.3001940>
35. Y. Wang, W. Zhou, T. Jiang, X. Bai, Y. Xu, Intra-class feature variation distillation for semantic segmentation, in *European Conference on Computer Vision*, (2020), 346–362. [https://doi.org/10.1007/978-3-030-58571-6\\_21](https://doi.org/10.1007/978-3-030-58571-6_21)
36. C. Shu, Y. Liu, J. Gao, Z. Yan, C. Shen, Channel-wise knowledge distillation for dense prediction, in *Proceedings of the IEEE International Conference on Computer Vision*, (2021), 5311–5320. <https://doi.org/10.1109/ICCV48922.2021.00526>
37. M. Tan, Q. Le, Efficientnetv2: Smaller models and faster training, in *International Conference on Machine Learning*, (2021), 10096–10106
38. Y. Tang, K. Han, J. Guo, C. Xu, C. Xu, Y. Wang, GhostNetv2: enhance cheap operation with long-range attention, in *Advances in Neural Information Processing Systems*, **35** (2022), 9969–9982.
39. K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, C. Xu, GhostNet: More features from cheap operations, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (2020), 1580–1589. <https://doi.org/10.1109/CVPR42600.2020.00165>
40. Z. Tian, C. Shen, H. Chen, T. He, FCOS: Fully convolutional one-stage object detection, in *Proceedings of the IEEE International Conference on Computer Vision*, (2019), 9627–9636. <https://doi.org/10.1109/ICCV.2019.00972>
41. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, *Int. J. Comput. Vis.*, **88** (2010), 303–338. <https://doi.org/10.1007/s11263-009-0275-4>



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)