Research on Multi-Scene Electronic Component Detection Algorithm with Anchor Assignment Based on K-Means

Zilin Xia, Jinan Gu*, Ke Zhang, Wenbo Wang and Jing Li

School of Mechanical Engineering, Jiangsu University, Zhenjiang 212013, China; xiazilin@stmail.ujs.edu.cn (Z.X.); zhangke@stmail.ujs.edu.cn (K.Z.); wangwb@ujs.edu.cn (W.W.); aygyxyl163.com (J.L.)
* Correspondence: gujinan@tsinghua.org.cn

Abstract: Achieving multi-scene electronic component detection is the key to automatic electronic component assembly. The study of a deep-learning-based multi-scene electronic component object detection method is an important research focus. There are many anchors in the current object detection methods, which often leads to extremely unbalanced positive and negative samples during training and requires manual adjustment of thresholds to divide positive and negative samples. Besides, the existing methods often bring a complex model with many parameters and large computation complexity. To meet these issues, a new method was proposed for the detection of electronic components in multiple scenes. Firstly, a new dataset was constructed to describe the multi-scene electronic component scene. Secondly, a K-Means-based two-stage adaptive division strategy was used to solve the imbalance of positive and negative samples. Thirdly, the EfficientNetV2 was selected as the backbone feature extraction network to make the method simpler and more efficient. Finally, the proposed algorithm was evaluated on both the public dataset and the constructed multi-scene electronic component dataset. The performance was outstanding compared to the current mainstream object detection algorithms, and the proposed method achieved the highest mAP (83.20% and 98.59%), lower FLOPs (44.26GMAC) and smaller Params (29.3 M).

Keywords: multi-scene electronic component detection; K-Means-based anchor assignment algorithm; EfficientNetV2

1. Introduction

With the rapid development of artificial intelligence and intelligent manufacturing, it has become an inevitable trend to achieve automation and intelligence in various industries [1]. To prepare for automation and intelligence, the real-time and accurate extraction of the state information of objects is the core and basis. Machine vision is one of the important technologies that can obtain the state information of objects from the sensed image [2]. At present, the electronic products are the important parts in smart systems, and they are developing rapidly. Various electronic products exist in a complex system, but the assembly of electronic components (especially the insertion of electronic components on the PCB circuit board) is executed by skilled workers. Therefore, the application of intelligent assembly robots in the electronic component assembly is one of the important research hotspots in the era of Industry 4.0. The object detection of electronic components is not only an important basis of electronic component assembly but can also inspect the quality of electronic components after assembly autonomously.

In recent years, object detection, which is based on deep learning, has developed rapidly and can be generally divided into two categories: anchor-based methods and anchor-free methods. The anchor-based object detection method has two branches: one-stage methods and two-stage methods. Both of them pre-determine a large number of anchors on the image first, then predict the class of these anchors and update their coordinates.
several times. Finally, the final detection results after filtering and adjusting the anchors are obtained. The two-stage method is more accurate because of the finer adjustment of the anchor compared to the one-stage method. Faster RCNN [3], as the first two-stage object detection method, is improved based on R-CNN [4] and Fast RCNN [5]. Faster RCNN proposes an RPN network to obtain candidate frames and send them to the subsequent network for classification and regression, and achieves end-to-end detection for the first time. However, its speed is slow. On the basis of Faster RCNN, SSD [6] proposes a one-stage object detection method and introduces multi-scale in its network. Its speed is greatly improved, but the accuracy is inferior to Faster RCNN. YOLOV1 [7] gets rid of the reliance on an anchor, and its direct prediction of the width and height of each position on the final feature map yields the final prediction instead of adjusting the anchor. Although this approach is fast, the disadvantage of not having an anchor also leads to the model’s bad detection of some objects with uncommon aspect ratios. The subsequent YOLOV2 [8], YOLOV3 [9] and YOLOV4 [10] shift to rely on the anchor, and the ability to detect objects is gradually improved. However, with the proposal of FPN [11] and Focal Loss [12], the disadvantages of anchors come to the fore. A large number of anchors will not only cause a huge increase in the number of parameters but also lead to an extreme imbalance in the number of positive and negative samples. Besides, a large number of redundant anchors need to be processed in the post-processing process. As a result, the final detection results are affected.

The anchor-free detection method no longer depends on the anchor, and it is mainly divided into two kinds; one is the keypoint-based detection method, which gets the coordinates and width and height of the target center point by pre-setting the key point to the GTBOX, such as CenterNet [13] or CornerNet [14]. The other is the center-based anchor-free object detection method, which is based on the GTBOX center point and the region near the center point. It divides positive and negative samples in the region, and finally completes the detection of the target by regressing the distance from the center point to the four edges, such as FCOS [15]. Among these two anchor-free methods, the center-based anchor-free object detection method is very similar to the anchor-based object detection method. The main difference between them is how to handle positive and negative samples. ATSS [16] argues that the difference between the center-based anchor-free detection method and anchor-based detection method is how to divide positive and negative samples at training time, and it is not necessary to approach the object with a large number of anchors. Therefore, the paper proposes an adaptive training sample selection to automatically select positive and negative training samples according to the statistical characteristics of the object. In PAA [17], the matching strategy considering anchors should be efficient and flexible; it should not only calculate the IoU (Intersection over Union) between the GTBOX (ground truth box) and anchor to determine whether it is a positive or negative sample but should let the model classify the positive and negative samples by itself and have a basis to explain the assignment strategy. Even when there is no anchor having a high IoU with GTBOX, the model should assign positive samples to it. This requires the model to be able to adjust its strategy adaptively during the training process and to rely on more than just one factor, the IoU, to determine whether it is a positive or negative sample.

The backbone feature extraction network is one of the most important parts of the object detection algorithm. It aims to extract image features and is the basis for the subsequent classification and positioning of objects in the image. VGGNet [18], the first widely used backbone feature extraction network, largely reduces the number of model parameters and improves the speed by stacking multiple $3 \times 3$ convolutional kernels in place of larger convolutional kernels. GoogLeNet [19] further enhances the depth of the network to seek higher accuracy, but the gradient disappears and the accuracy weakens instead when the number of layers of the network becomes too high. ResNet [20] solves this problem well by using shortcut branching, which allows the network layers to be designed
Deeper without gradient disappearance, such as Resnet50 and Resnet101. SENet [21] proposes an attention mechanism to further enhance the capability of the backbone feature extraction network and thus is widely used in object detection networks. As accuracy improves, speed is also critical. The MobileNet series [22–24] optimizes the structure of convolution and proposes a deep separable convolution and group convolution module, which greatly reduces the number of parameters of the model and improves the speed of the model. The ShuffleNet family [25,26] proposes a channel shuffle to optimize the problem that channel information cannot be shared between group convolutions. As the optimization of convolution has been nearly extreme, the study of the backbone feature extraction network shifts to the optimization of the overall network structure and the selection of related modules. RegNet [27] and EfficientNet [28] both use NAS technology to search different parts of the network so as to select the optimal combination in the search space, thereby further improving the ability to extract features. However, the training of EfficientNet is more difficult and slower. EfficientNetV2 [29] solves this problem by replacing the shallow MBConv module with the Fused-MBCONV module, and the accuracy is further improved.

On the other hand, a great deal of research has been conducted on electronic component detection based on deep learning. Kuo et al. [30] constructed the PCB board generic electronic component dataset and proposed a deep-learning-based detection method. Sun et al. [31] proposed an improved SSD algorithm to implement electronic component detection in stacked scenes. Huang et al. [32] proposed an improved YOLOV3 algorithm to detect electronic components in stacked scenes. Li et al. [33] proposed an improved YOLOV3 algorithm to detect electronic components on PCB boards. Dong et al. [34] proposed an improved Mask R-CNN method to achieve the segmentation of electronic components in the stacked state. Li et al. [35] studied the relationship between the perceptual field and the anchor size in the object detection algorithm and made improvements based on YOLOV3 to achieve the detection of electronic components on PCB boards. However, since the characteristics of electronic components in different scenes vary largely and it is hard to fulfill object detection using one algorithm, there is a lack of research regarding an object detection algorithm that can be applied simultaneously in multiple scenes. Besides, the existing electronic component detection algorithms are improved based on the fundamental detection network without proposing a completely new algorithm for electronic component detection.

The main problems are as follows:
1. In the training process of the existing object detection network, unbalanced positive and negative samples and the need to manually set the threshold of positive and negative samples division;
2. How to make the target detection algorithm simpler and more efficient to avoid the high computational complexity of the model;
3. The detection of electronic components in multiple scenes is not researched in depth.

Therefore, based on the above ideas, an adaptive two-stage positive and negative sample matching strategy is proposed in this paper. The model can be adjusted adaptively as its parameters change during the training process and can be embedded in any current anchor-based object detection method. EfficientNetV2 is chosen as the backbone feature extraction network for our object detection algorithm, which has an efficient parameter design and is simpler and more efficient overall. A multi-scene electronic component dataset for multi-scene electronic component detection is constructed, and an object detection algorithm for multi-scene electronic component detection is proposed.

The following arrangements for the article are as follows: Section 2 introduces the proposed multi-scene electronic component dataset and the public dataset required for the experiments. The proposed object detection algorithm is presented in Section 3. Section 4 validates the algorithm on different datasets and obtains the final experimental results. Section 5 concludes the whole paper and provides an outlook for the future.
2. Dataset

2.1. Multi-Scene Electronic Component Dataset

The detection of electronic components is the basis of the assembly process of electronic components. Electronic components need to be identified and positioned prior to assembly. After assembly, it is necessary to check whether the assembly is completed and whether it is correct. At the same time, the characteristics of electronic components before and after assembly are different. The purpose of constructing this dataset is to make the proposed object detection algorithm, which can adapt to different scenes in real-life production, even if the characteristics of electronic components are very different in various scenes.

Three main kinds of scenes are contained: (1) a messy flat scene of electronic components before assembly, as shown in Figure 1a,b; (2) a mixed scene of two scenes during assembly, as shown in Figure 1c; (3) a PCB board scene after assembly, where the electronic components have been inserted and assembled on the PCB board, as shown in Figure 1d. Table 1 shows the concrete types of electronic components in the multi-scene dataset, which contains a total of 14 categories, including common electronic components in circuit boards, such as capacitors, inductors, resistors, chips, diodes, etc. The 1040 images in the constructed multi-scene electronic components were acquired using an industrial camera with a resolution of $4092 \times 3000$. The pre-assembly scene contains 494 images, the in-assembly scene contains 256 images and the post-assembly scene contains 290 images.

![Figure 1](image-url)

**Figure 1.** Different scenes in a multi-scene electronic component dataset: (a,b) pre-assembly scene; (c) in-assembly scene; (d) post-assembly scene.
Table 1. Multi-scene electronic components dataset categories and corresponding images.

| Name              | Image       | Name         | Image       |
|-------------------|-------------|--------------|-------------|
| Cap22uF           | ![Cap22uF](image) | Cap220uF     | ![Cap220uF](image) |
| Inductance_1      | ![Inductance_1](image) | Cap470uF     | ![Cap470uF](image) |
| Resistance        | ![Resistance](image) | Capacitor    | ![Capacitor](image) |
| Red pin           | ![Red pin](image) | Varistor     | ![Varistor](image) |
| Chip              | ![Chip](image) | Rectifier diode | ![Rectifier diode](image) |
| Inductance        | ![Inductance](image) | Yellow capacitor | ![Yellow capacitor](image) |
| White pin         | ![White pin](image) | Relay        | ![Relay](image) |

LabelImg is used to annotate the images in the dataset to generate labels. The labels in the dataset are uniformly converted to PASCAL VOC2012 format since the proposed algorithm is not only validated on the constructed multi-scene electronic components dataset but also further validated on the public datasets PASCAL VOC2007 and 2012. The number of labels corresponding to each category is shown in Figure 2 below. The multi-scene electronic component dataset is divided into three parts: training set, validation set and test set. The training set is used to train the model, the validation set is used to adjust the hyperparameters of the model (learning rate, epoch etc.) to prevent overfitting, the test set is used to test the effect of the model and, finally, the algorithm is verified on the test set for its strengths and weaknesses. The training and validation set account for 70% of the data set, while the test set accounts for 30%.

2.2. Public Datasets

In order to verify the performance of the proposed algorithm, it is necessary to perform validation on a public dataset. Public datasets with a larger amount of data can better reflect the robustness of the algorithm and its ability to detect objects. The PASCAL VOC dataset [36] is a classical public dataset and authoritative in the field of target detection. The two most-used versions are 2007 and 2012, including a total of 21 categories. The PASCAL VOC dataset is widely used to evaluate models for classification, localization, detection, segmentation, etc. We mixed the PASCAL VOC2007 training set, validation set and PASCAL VOC2012 training set as our training set and validation set, which included a total of 21,380 images, and the PASCAL VOC2007 test set as our test set, which included a total of 4952 images. The purpose of this is to make the training set contain more images and to test the performance of the proposed algorithm when the dataset is large.
Figure 2. Categories and numbers on the multi-scene electronic components dataset.

3. Method

An object detection method for multi-scene electronic component detection is proposed, which uses EfficientNetV2 as the backbone network, FPN as the feature fusion module and uses the proposed two-stage adaptive positive and negative sample matching strategy in the training part.

3.1. EfficientNetV2

EfficientNetV2 is improved on the basis of EfficientNetV1, replacing the shallow MBConv module with the Fused-MBConv module, and using NAS (Neural Architecture Search) technology to search for the best combination of the two modules. As shown in Figure 3, the MBConv module consists of two $1 \times 1$ Conv, $3 \times 3$ Depthwise Conv, and the attention module SE. As shown in Figure 4, the difference between the Fused-MBConv module and the MBConv module is the replacement of $1 \times 1$ Conv and $3 \times 3$ Depthwise Conv with a $3 \times 3$ convolution, which can significantly improve the speed of extracting features. EfficientNetV2 improves both network training speed and parameter efficiency and has excellent performance in accuracy on ImageNet dataset. Its main features are as follows: (1) Fused-MBConv module is used in the shallow layer of the network and MBConv module is used in the deep layer to improve the training speed and parameter efficiency of the network; (2) the MBConv module uses a smaller expansion ratio to reduce the memory consumption; (3) uses smaller $3 \times 3$ convolutional kernels and stacks more network layers to avoid the reduction in the perceptual domain associated with the use of small convolutional kernels.

Figure 3. MBConv.

Figure 4. Fused-MBConv.
3.2. Adaptive Positive and Negative Sample Matching Strategy

The goal of the network is to adaptively learn matching strategies for positive and negative samples. As the loss of positive samples becomes smaller and smaller during the training process of the network, the loss of negative samples becomes larger and larger. Based on this idea, we design a two-stage matching strategy for positive and negative samples. Its main features are as follows: (1) it can find effective evidence according to the model to measure the quality of the anchor relative to GroundTruth; (2) it can adaptively realize the matching of positive and negative samples without introducing hyperparameters.

First, it is necessary to find out the relationship between the good and bad stratification of anchor and GTBOX. During the training process of the network, the loss scores of anchors with higher correlation with GTBOX will gradually decrease with back propagation. On the contrary, the loss scores of anchors with lower correlation with GTBOX will become larger and larger. Therefore, in the first stage, the loss score of the anchor is obtained, which reflects the fit between its prediction frame and the GTBOX. The specific calculation step is done by calculating the classification loss score of the anchor and the regression loss score of the prediction box and adding them together using the following equations.

\[
S(f_a(x), g) = \lambda_1 S_{cls}(f_a(x), g) + \lambda_2 S_{loc}(f_a(x), g),
\]

where \(S(f_a(x), g)\) refers to \(S_{cls}(f_a(x), g)\) and \(S_{loc}(f_a(x), g)\) denotes the classification loss score and regression loss score. \(f_a\) denotes the whole network model and \(\alpha\) is the model parameter. Further, \(a\) and \(g\) represent the anchor and GTBOX, respectively. \(\lambda_1\) and \(\lambda_2\) indicate the proportion of the two scores to the total score, respectively. The values of \(\lambda_1\) and \(\lambda_2\) are 0.5, respectively. The category prediction for each anchor can be obtained from the classification branch and calculated with the real GTlabel (ground truth label) to obtain the classification loss score. The offset value can be obtained in the regression branch, and the offset value can be decoded to obtain the prediction bbox corresponding to the anchor, and then the prediction BBOX and the GTBOX can be calculated by IoU to obtain the regression loss score corresponding to the anchor.

\[
S_{cls}(f_a(x), g) = L_{cls}(f_a(x), g),
\]

\[
S_{loc}(f_a(x), g) = L_{iou}(f_a(x), g),
\]

The \(L_{cls}\) and \(L_{iou}\) represent the Focal loss and Glou loss. The loss score of each anchor can be calculated as the input of the second stage, which is a good measure of the fit between the anchor and the GTBOX.

In the second stage, the positive and negative samples are divided adaptively based on the anchor scores obtained in the first stage. Since the anchor scores will change with the training process, two characteristics are certain: (1) the loss scores of the anchors corresponding to positive samples will be smaller and smaller; (2) the loss score of the anchors corresponding to negative samples will become larger and larger. Therefore, the clustering method is used to cluster the anchors into two classes using their loss scores, one for positive samples and the other for negative samples. Since the number of anchors is particularly large, and the number of anchors with high correlation with GTBOX is only a small fraction, the \(K\) anchors with the smallest loss score per GTBOX are selected for subsequent clustering on each feature layer.

\[
S(f_a(x), g) = \sum_{i=1}^{level} \text{FindKthSmallest}(S(f_a(x), g), K),
\]

where \(level\) denotes the number of feature layers and \(K\) denotes the number of anchors selected for each feature layer. Next, the clustering algorithm is used to cluster them into two classes, and the corresponding loss scores of the centers of the two classes are found as the basis for the subsequent classification of positive and negative samples. In this way, it is possible to make the model adaptively classify positive and negative samples during
the training process instead of setting implicit hyperparameters. Thus, it can avoid the problem that the adjustment of hyperparameters brings about the bad effect of the model.

\[
\text{cluster centers} = K\text{Means}(S(f_u(a, x), g), n\_clusters = 2),
\]

where \(\text{cluster centers}\) denotes the scores corresponding to the two types of clustering centers, \(K\text{Means}\) is used clustering algorithm, and \(n\_clusters\) denotes the number of clustering centers. The smaller of the scores corresponding to the two types of clustering centers is chosen, which is the one corresponding to the positive samples. Next, this is used as a basis to adaptively divide the positive and negative samples during the training of the network.

\[
m_s = \min(\text{cluster centers}),
\]

where \(m_s\) is the score corresponding to the positive sample center.

As shown in Figure 5, the two clustering centers obtained by \(K\text{Means}\) are the positive sample center and the negative sample center, and the center \(m_s\) of the positive sample center is selected as the basis for dividing the positive and negative samples. When the anchor score is less than \(m_s\), it is a positive sample, and, otherwise, it is a negative sample.

![Figure 5. The criteria for dividing positive and negative samples.](image)

\[
P_{pos} = \text{Scores} < m_s,
\]

\[
N_{neg} = \text{Scores} > m_s,
\]

where \(P_{pos}\) represents the anchor corresponding to the positive sample, and \(N_{neg}\) is the anchor corresponding to the negative sample. So far, the adaptive division of positive and negative samples during the network training has been completed. These positive and negative samples will be used to calculate the final loss of the network and perform back propagation and update parameters. It should be noted that the previously calculated loss score will not be backpropagated but only used as a reliable basis for distinguishing between positive and negative samples.

The whole process of adaptive positive and negative sample matching strategy is shown in Algorithm 1. Firstly, find out the IoU between GTBOX and anchor, which is larger as \(A_g\) (Line 2). Secondly, for each feature layer, calculate the score corresponding to anchor, and find out the anchor of each predicted feature layer \(\text{KthSmallest as} C_g\) (Line 3–8). Finally, the scores corresponding to \(C_g\) are then used for cluster analysis, and the positive and negative samples are divided according to the scores corresponding to the cluster centers to obtain \(P_g\) and \(N_g\) (Line 9–13).
Algorithm 1. K-Means-based anchor assignment algorithm.

**Input**: \( G, A, A_i, L, K \)

- \( G \) is a set of ground truth boxes on the image
- \( A \) is a set of all anchor boxes
- \( A_i \) is a set of all anchor boxes from the \( i \text{th} \) pyramid levels
- \( L \) is the number of feature pyramid levels
- \( K \) is the number of candidate anchors for each pyramid

**Output**: \( P_{pos}, N_{neg} \)

- \( P_{pos} \) is a set of positive samples
- \( N_{neg} \) is a set of negative samples

1: for each ground-truth \( g \in G \) do
2: \( A_g \leftarrow \) Get the best ground truth with \( g \) among all anchors by IoU
3: for each level \( i \in [1, L] \) do
4: \( A_i^g \leftarrow A_i \cap A_g \)
5: \( S_i \leftarrow \) Calculate the loss score of \( A_i^g \)
6: \( C_i^l \leftarrow \) FindKthSmallest\((S_i, K)\)
7: \( C_g \leftarrow C_g \cup C_i^l \)
8: end for
9: \( \text{cluster} \_\text{centers} \leftarrow \text{KMeans}(C_g, 2) \)
10: \( m_s \leftarrow \min(\text{cluster} \_\text{centers}) \)
11: \( P_g \leftarrow \) The score in \( C_g \) is less than or equal \( m_s \)
12: \( N_g \leftarrow \) The score in \( C_g \) is greater than \( m_s \)
13: \( P_{pos} \leftarrow P \cup P_g, N_{neg} \leftarrow N \cup N_g \)
14: end for
15: return \( P_{pos}, N_{neg} \)

3.3. The Proposed Algorithm

The overall structure of the proposed network is shown in Figure 6, where the input network images are first uniformly resized to 512 × 512 and then passed through the backbone feature extraction network EfficientNetV2 to extract features. Three valid feature layers, C3, C4 and C5, are selected as the results of feature extraction, and then they are passed into the FPN for feature fusion to generate the final five predicted feature layers. Finally, the prediction results are generated by the classification prediction branch and the regression prediction branch of the Head section. The specific parameters of the network are shown in Figure 7. Stem corresponds to the structure of Conv-BN-SiLU, Conv 1 to Conv 3 and Conv 4 to Conv 6 correspond to the structure of Fused-Conv module and MB-Conv module described above, where Conv 3, Conv 5 and Conv 6 correspond to the three effective feature layers C3, C4 and C5.

![Figure 6](image-url)  
*Figure 6.* The overall network structure.
The loss function in the proposed algorithm mainly consists of classification loss, regression loss and Center-ness loss.

\[ \text{Loss} = \text{Error}_{\text{class}} + \text{Error}_{\text{regression}} + \text{Error}_{\text{centerness}} \]  \hspace{1cm} (9)

where \( \text{Error}_{\text{class}} \) denotes classification loss, \( \text{Error}_{\text{regression}} \) denotes boundary regression loss and \( \text{Error}_{\text{centerness}} \) denotes Center-ness loss.

\[ \text{Error}_{\text{class}} = \frac{1}{N_{\text{pos}}} \sum_{x,y} L_{\text{cls}} (p_{x,y}, p_{x,y}^*) = 1, \hspace{1cm} (10) \]

where \( L_{\text{cls}} \) denotes the classification loss and the loss function used is Focal loss; \( p_{x,y} \) denotes the predicted value of the category in the classification branch, \( p_{x,y}^* \) denotes the true value and \( N_{\text{pos}} \) denotes the number of positive samples. When calculating the classification loss, we calculate the loss of all samples, including positive and negative samples.

\[ \text{Error}_{\text{regression}} = \frac{\lambda}{N_{\text{pos}}} \sum_{x,y} \Phi(c_{x,y}^* > 0) L_{\text{reg}} (t_{x,y}, t_{x,y}^*), \hspace{1cm} (11) \]

where \( L_{\text{reg}} \) represents regression loss, and the loss function used is GIoU. As shown in Figure 8, the specific is the distance from the center point of the regression boundaries to the top, bottom, left and right four edges, denoted as \( (l,b,l,r); \Phi(c_{x,y}^* > 0) \) is a representation function with its function value of 1 when \( c_{x,y}^* > 0 \) and 0 in the rest of cases, which is to indicate that only positive samples are involved in the process of regression loss; \( t_{x,y} \) is the predicted value of the regression branch, \( t_{x,y}^* \) is the true value of the corresponding GTBox and \( \lambda \) is taken as 1 here in order to balance the loss of each component.

\[ \text{Error}_{\text{centerness}} = \frac{1}{N_{\text{pos}}} \sum_{x,y} \Phi(c_{x,y}^* > 0) L_{\text{cen}} (c_{x,y}, c_{x,y}^*), \hspace{1cm} (12) \]

where \( L_{\text{cen}} \) is the Center-ness loss, and the loss function used is BCE. Its function is to make the quality of the prediction box higher, and it can suppress some low-quality prediction boxes, where \( c_{x,y} \) is the predicted value of the centerness branch and \( c_{x,y}^* \) is the true value whose specific is calculated by Equation (13) [15]. Note that the Center-ness loss also has only positive samples involved.

\[ c_{x,y}^* = \sqrt{\frac{\min(t_{x,y}^*, r_{x,y}^*)}{\max(t_{x,y}^*, r_{x,y}^*)}} \times \frac{\min(t_{x,y}^*, b_{x,y}^*)}{\max(t_{x,y}^*, b_{x,y}^*)} \hspace{1cm} (13) \]
4. Experiments and Discussion

In order to validate the effectiveness and robustness of the proposed algorithm, it was verified on both the constructed dataset of electronic components and the public dataset PASCAL VOC. Both the proposed algorithm and the other compared algorithms for backbone feature extraction network weights were used pre-trained on ImageNet, and all the algorithms were validated on the same device with an E5-2678 V3 CPU, 16 G RAM and 3090 graphics card with 24 G video memory. All the algorithms were trained for 100 epochs, the first 50 epochs had a learning rate of 0.001 and the weights of the backbone feature extraction network were frozen. The second 50 epochs had a learning rate of 0.0001, the weights of the backbone feature extraction network were unfrozen and the weights of the whole network were updated.

4.1. Evaluation Criteria

In order to evaluate the performance of the proposed algorithm, the following evaluation metrics will be used.

4.1.1. Precision and Recall

Precision is the percentage of detected objects that are correctly detected, and recall is the percentage of correct detections to the total number of detections required in the test set.

\[
\text{Precision} = \frac{TP}{all \ detections} \tag{14}
\]

\[
\text{Recall} = \frac{TP}{all \ GTBox} \tag{15}
\]

TP denotes correctly detected targets in the detected targets. When different Recall values are selected, numerous sets of Precision and Recall values can be obtained. With Recall as the horizontal axis and Precision as the vertical axis, the P–R curve can be drawn for subsequent calculation of AP (average precision) and mAP (mean average precision).

4.1.2. AP and mAP

The AP value corresponding to this category can be calculated based on the P–R curve drawn by Precision and Recall. Specifically, the AP value of the species can be obtained by integrating the P–R curve of the species, which is calculated as follows Equation (16) [37].

\[
\int_0^1 p(r) dr, \tag{16}
\]

where \( p \) represents Precision, \( r \) represents Recall and \( p(r) \) is a function with \( r \) as the parameter. In the actual calculation process, for each kind of threshold (Recall value), the corresponding Precision value is obtained and then multiplied by the change of Recall value, and then the product value obtained under all thresholds is accumulated to obtain the AP value, which is calculated by the following Equation (17) [37].


\[
AP = \sum_{k=1}^{N} \max_{k\geq k} P(k) \Delta r(k),
\]

where \( N \) denotes the number of images, \( \max_{k\geq k} P(k) \) indicates the maximum Precision value for recognizing \( K \) images. \( \Delta r(k) \) indicates the change of Recall value when the number of recognized images changes from \( k-1 \) to \( k \).

Furthermore, mAP is the accumulation of all kinds of AP values and then takes the average value. It is used to measure the performance of the whole model. In general, the AP value is used to measure the performance of a category on the model, while mAP measures the performance of the network based on all categories.

4.1.3. Params and Flops

FLOPs (floating point of operations) and Params are important reference factors to measure the complexity of the model. FLOPs is the calculation amount of the network, and Params is the number of parameters of the network. The larger their value, the more complex the network. With the development of target detection, the development trend of the model is bound to be low FLOPs and small Params.

4.2. Comparison of Different Algorithms

4.2.1. Comparison of Public Datasets

To validate the performance of the proposed algorithm, the public dataset PASCAL VOC dataset was used, which has a larger amount of data to better reflect the robustness of the algorithm and its ability to detect objects. The PASCAL VOC2007 training set was mixed with the PASCAL VOC2012 training set as the training set, and the PASCAL VOC2007 test set was used as the test set for experiments. By comparing other typical object detection algorithms and reproducing them on our own device, we obtained their mAP values on PASCAL VOC2007, as shown in the following table.

The mAP comparisons of the proposed algorithm with other methods on the PASCAL VOC2007 test set are shown in Table 2, along with the FLOPs and Params of the proposed algorithm compared with other algorithms. Figure 9 shows the P–R curves for the two categories from the PASCAL VOC2007 test set.

**Table 2.** mAP values, FLOPs and Params of the proposed model and other models on the PASCAL VOC2007 test set.

| Model     | FLOPs  | Params | mAP   |
|-----------|--------|--------|-------|
| SSD       | 90.54  | 26.42 M | 71.44% |
| Faster RCNN| 461.76 | 28.48 M | 76.84% |
| YOLOV3    | 49.7   | 61.59 M | 77.73% |
| YOLOV4    | 45.37  | 64.05 M | 80.49% |
| FCOS      | 51.69  | 31.92 M | 76.18% |
| Proposed model | 44.26  | 29.3 M   | 83.44% |

As shown in Table 2, by comparing with the classical two-stage Faster RCNN, the one-stage SSD and the popular YOLOV3 and YOLOV4, the proposed algorithm has a relatively large improvement in mAP by 6.6%, 12%, 5.71% and 2.95%, respectively. At the same time, compared to the anchor-free object detection algorithm FCOS, the method also has a large improvement, with a 7.26% improvement in mAP. The proposed algorithm has the lowest FLOPs, and the parameter amount is only 2.88 M more than the smallest SSD, indicating that the proposed algorithm parameter design is more efficient and the model is simpler and more efficient.

The P–R curve in Figure 9 shows that the proposed algorithm has a relatively great advantage over other algorithms, and the Precision can still be maintained at a high level when the Recall value is high. Pottedplant is a small and dense type in the public dataset,
and it can be seen from Figure 9b that the proposed algorithm performs better for this type compared to other algorithms, which proves that the method is more robust and has better results for small and dense scenes.

![Figure 9. P–R curves for two categories from the PASCAL VOC2007 test set: (a) Cat; (b) Pottedplant.](image)

### 4.2.2. Comparison of Multi-Scene Electronic Component Datasets

After the validation of the public dataset, the next step is the further discussion of the constructed multi-scene electronic component dataset. The model is trained on the multi-scene electronic component training set to make the model converge, and then tested on the multi-scene electronic component test set to obtain the mAP values of the proposed algorithm and other algorithms.

The mAP comparison of the proposed algorithm with other methods on the multi-scene electronic component test set is shown in Table 3, along with the FLOPs and Params of the proposed algorithm compared with other algorithms. Figure 10 shows the P–R curves for the four categories from the multi-scene electronic component test set. Figure 11 shows the detection results of the proposed algorithm on the test set of the multi-scene electronic component dataset.

### Table 3. mAP values, FLOPs and Params of the proposed model and other models on the multi-scene electronic component test set.

| Model   | FLOPs  | Params  | mAP   |
|---------|--------|---------|-------|
| SSD     | 90.54  | 26.42 M | 95.67%|
| Faster RCNN | 461.76 | 28.48 M | 96.19%|
| YOLOV3  | 49.7   | 61.59 M | 97.98%|
| YOLOV4  | 45.37  | 64.05 M | 88.40%|
| FCOS    | 51.69  | 31.92 M | 98.24%|
| Huang   | 15.46  | 25.24   | 95.76%|
| Li      | 53.97  | 61.86   | 88.56%|
| Proposed model | 44.26  | 29.3 M  | 98.83%|

![Figure 10. P–R curves for the four categories from the multi-scene electronic component test set.](image)

![Figure 11. Detection results of the proposed algorithm on the test set of the multi-scene electronic component dataset.](image)
Figure 10. P–R curves of four electronic components from the multi-scene electronic component dataset (a) Cap22uF; (b) Cap220uF; (c) Cap470uF; (d) Rectifier diode.

Figure 11. Detection results on multi-scene electronic component dataset.

As shown in Table 3, compared with Faster RCNN and SSD, the proposed algorithm has significant improvements: the mAP can be improved by 2.64% and 3.16%, respectively. Compared with the FCOS, the method also has a certain improvement: the mAP improved 0.59%. Compared with the popular YOLOV3 and YOLOV4, the method also has some improvement: the mAP can be improved by 0.85% and 10.43%, respectively. Compared with other detection methods for electronic components, the proposed method improves the map by 3.07% compared to the improved YOLOV3 method proposed by Huang. Compared to the improved YOLOV3 method proposed by Li, the mAP improved by 10.27%. It can be seen that the proposed algorithm performed outstandingly on the multi-scene electronic components dataset with the highest mAP. Moreover, it can be seen that the proposed algorithm has lower FLOPs and a smaller number of parameters, which indicates that the proposed algorithm has a more effective parameter design and simpler and more efficient model.
Figure 10 shows that the curves of the proposed algorithm have a clear advantage over other algorithms, with P–R values closer to the coordinates (1,1), and can maintain a higher Precision at higher Recall values. For example, YOLOV4 performs better on Rectifier diode and worse on Cap22uF, Cap220uF and Cap470uF; Faster RCNN performs better on Cap220uF, Cap470uF and worse on Rectifier diode. In contrast, the method we proposed is stable and has better results on different kinds of electronic components. The specific detection results are shown in Figure 11, which shows that the electronic components in different scenes can be accurately detected even if the characteristics are different. Figure 11a,b shows the results of scene detection before assembly, Figure 11c shows the results of scene detection in assembly, and Figure 11d shows the results of scene detection in post assembly.

4.3. The Influence of Positive and Negative Sample Matching Strategies

In this section, we compare the adaptive positive and negative sample matching strategies proposed in ATSS in order to highlight the advantages of the proposed K-Means-based two-stage adaptive positive and negative sample matching strategy. The approach is to keep the rest of the proposed network unchanged and only switch the matching strategy for positive and negative samples to the one proposed in ATSS. Finally, the final experimental results were validated on the public dataset and the multi-scene electronic component dataset.

Table 4 shows the mAP values of the proposed matching strategies compared with ATSS on the public dataset, and Table 5 shows the mAP values of the proposed matching strategies compared with ATSS on the multi-scene electronic component dataset. Figure 12 shows the mAP curves of the two matching strategies on the public dataset, and Figure 13 shows the mAP curves of the two matching strategies on the multi-scene electronic component dataset.

![Figure 12](image1.png)

**Figure 12.** mAP curves of two matching strategies on public dataset.

![Figure 13](image2.png)

**Figure 13.** mAP curves of two matching strategies on multi-scene electronic component dataset.
Table 4. Comparison of mAP values of the proposed matching strategy with ATSS on public dataset.

| Model                        | mAP  |
|------------------------------|------|
| Proposed model               | 82.89% |
| Proposed model + ATSS        | 81.62% |

From Table 4, it can be seen that, compared to the adaptive positive and negative sample matching strategy proposed by ATSS, the proposed K-Means-based two-stage positive and negative sample matching strategy has better results, with a significant improvement of 1.27% in mAP.

Table 5. Comparison of mAP values of the proposed matching strategy with ATSS on the multi-scene electronic component dataset.

| Model                        | mAP |
|------------------------------|-----|
| Proposed model               | 98.83% |
| Proposed model + ATSS        | 98.67% |

From Table 5, it can be seen that the proposed two-stage K-Means-based positive and negative sample matching strategy works better with a 0.16% increase in mAP. It is limited by the complexity of the dataset, so the improvement on the constructed dataset is not as large as the public dataset, but the performance is also more outstanding.

As can be seen in Figure 12, the proposed matching strategy on the public dataset is always more effective with higher mAP values as the epoch of training increases. As can be seen in Figure 13, in the multi-scene electronic component dataset, although the proposed matching strategy mAP is slightly lower than ATSS in the first half, as the epoch increases, our matching strategy is still better in the end.

5. Conclusions

In this paper, an electronic component detection method in multiple scenes was proposed. Since the characteristics presented by electronic components in different scenes may be different, in order to achieve accurate detection of the same electronic component in different scenes, we constructed a multi-scene electronic component dataset. It includes a total of 1040 pictures and 14 categories. In order to solve the problem of imbalance between positive and negative samples during training due to the number of anchor points in the object detection method, a two-stage adaptive positive and negative sample matching strategy based on K-Means was proposed, and the number of anchor points was set to 1, which greatly reduces the number of parameters and solves the problem of imbalance between positive and negative samples. EfficientNetV2 was chosen as the backbone of the network, which makes the proposed method simpler and more efficient in parameter design, and the network is less computationally intensive compared with other methods.

Through this study, the proposed method was able to achieve the detection of electronic components in multiple scenes. Its precision is higher compared to other mainstream detection algorithms, and it is simpler and more efficient in parameter design.

Our future research direction is to further expand our multi-scene electronic component dataset and to transplant the method to embedded devices to achieve real-time detection of electronic components.

Author Contributions: Conceptualization, Z.X.; Data curation, K.Z.; Formal analysis, Z.X. and K.Z.; Investigation, K.Z.; Methodology, Z.X.; Resources, Z.X. and K.Z.; Software, Z.X.; Supervision, J.G. and W.W.; Writing—original draft, Z.X.; Writing—review & editing, Z.X., J.G., K.Z., W.W. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 51875266, 52105516).

Institutional Review Board Statement: Not applicable.
Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Wang, W.; Zhang, Y.; Gu, J.; Wang, J. A Proactive Manufacturing Resources Assignment Method Based on Production Performance Prediction for the Smart Factory. *IEEE Trans. Ind. Inform.* 2022, 18, 46–55. https://doi.org/10.1109/TII.2021.3073404.

2. Fu, L.; Zhang, Y.; Huang, Q.; Chen, X. Research and Application of Machine Vision in Intelligent Manufacturing. In Proceedings of the 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, China, 28–30 May 2016; pp. 1126–1131.

3. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 2017, 39, 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031.

4. Girshick, R.; Jeff, D.; Trevor, D.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.

5. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Las Vegas, NV, USA, 27–30 June 2015; pp. 1440–1448.

6. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot Multibox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Volume 9905 LNCS, pp. 21–37.

7. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.

8. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, 22–25 July 2017; pp. 6517–6525.

9. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* 2018, arXiv:1804.02767.

10. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* 2020, arXiv:2004.10934.

11. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.

12. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 2020, 42, 318–327. https://doi.org/10.1109/TPAMI.2018.2858826.

13. Zhou, X.; Austin, U.T.; Wang, D.; Berkeley, U.C.; Austin, U.T. Objects as Points. *arXiv* 2019, arXiv:1904.07850.

14. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Volume 128, pp. 734–750.

15. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9627–9636.

16. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S.Z. Bridging the Gap between Anchor-Based and Anchor-Free Detection via Adaptive Training Sample Selection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 9759–9768.

17. Kim, K.; Lee, H.S. Probabilistic Anchor Assignment with IoU Prediction for Object Detection. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Volume 12370 LNCS, pp. 355–371.

18. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* 2014, arXiv:1409.1556.

19. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.

20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.

21. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 2020, 42, 2011–2023. https://doi.org/10.1109/TPAMI.2019.2913372.

22. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* 2017, arXiv:1704.04861, preprint.

23. Mark Sandler, Howard, A.; Zhu, M.; Zhmoginov, A.; Liang-Chieh Chen MobileNetV2: Inverted Residuals and Linear Bottlenecks Mark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.

24. Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.C.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 1314–1324.
25. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
26. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Volume 11218 LNCS, pp. 116–131.
27. Radosavovic, I.; Kosaraju, R.P.; Girshick, R.; He, K.; Dollár, P. Designing Network Design Spaces. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10428–10436.
28. Tan, M.; Le, Q. v. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 6105–6114.
29. Tan, M.; Le, Q. v. EfficientNetV2: Smaller Models and Faster Training. In Proceedings of the International Conference on Machine Learning, Jeju, Korea, 23–25 April 2021.
30. Kuo, C.-W.; Ashmore, J.; Huggins, D.; Kira, Z. Data-Efficient Graph Embedding Learning for PCB Component Detection. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Village, HI, USA, 7–11 January 2019; pp. 551–560.
31. Sun, X.; Gu, J.; Huang, R. A Modified SSD Method for Electronic Components Fast Recognition. Optik 2020, 205, 163767. https://doi.org/10.1016/j.ijleo.2019.163767.
32. Huang, R.; Gu, J.; Sun, X.; Hou, Y.; Uddin, S. A Rapid Recognition Method for Electronic Components Based on the Improved YOLO-V3 Network. Electronics 2019, 8, 825. https://doi.org/10.3390/electronics8080825.
33. Li, J.; Gu, J.; Huang, Z.; Wen, J. Application Research of Improved YOLO V3 Algorithm in PCB Electronic Component Detection. Appl. Sci. 2019, 9, 3750. https://doi.org/10.3390/app9183750.
34. Yang, Z.; Dong, R.; Xu, H.; Gu, J. Instance Segmentation Method Based on Improved Mask R-Cnn for the Stacked Electronic Components. Electronics 2020, 9, 886. https://doi.org/10.3390/electronics9060886.
35. Li, J.; Li, W.; Chen, Y.; Gu, J. A PCB Electronic Components Detection Network Design Based on Effective Receptive Field Size and Anchor Size Matching. Comput. Intell. Neurosci. 2021, 2021, 6682710. https://doi.org/10.1155/2021/6682710.
36. Everingham, M.; van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. Int. J. Comput. Vis. 2010, 88, 303–338. https://doi.org/10.1007/s11263-009-0275-4.
37. Padilla, R.; Passos, W.L.; Dias, T.L.B.; Netto, S.L.; da Silva, E.A.B. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. Electronics 2021, 10, 279. https://doi.org/10.3390/electronics10030279.