Advances in Computer, Signals and Systems (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2025.090320
ISSN 2371-8838 Vol. 9 Num. 3

Research on SSD-Mobilenet-Based Electronic Component Object Detection Algorithm

Xianshuang Zhao

Guangzhou College of Applied Science and Technology, Zhaoqing, Guangdong, China

Keywords: SSD; NPU; Deep Separable Convolution; GIoU

Abstract: This paper designs an electronic component detection system for physics teaching experiments in educational courses. The system model has been successfully deployed on NPU-embedded mobile devices. The SSD (Single Shot Multi-Box Detector) network serves as the foundation for model training. To adapt the trained model to NPU-based embedded devices, we replace the basic convolutional layer with lightweight depth-separable convolutions. To improve object detection accuracy under high overlap conditions, we employ GIoU (Generalized Intersection-Union) to mitigate missed detection caused by excessive overlap.

1. Introduction

In recent years, with the rapid advancement of chip technology, computing power has significantly increased, enabling artificial intelligence (AI) to flourish across various disciplines. In the field of computer vision [1], object detection remains a key research focus. The demand for deploying computer vision in mobile education has grown substantially [2]. Current physical circuit teaching methods face challenges in providing timely guidance due to faculty limitations. Therefore, developing a real-time circuit education system to address timeliness issues is essential. To tackle the problem of overlapping electronic components that often escape detection, this paper proposes using Gross Intersection Over Union (GIoU) instead of Intersection Over Union (IoU) for object detection, thereby improving accuracy.

In traditional object recognition systems, sliding windows are randomly selected from images to identify target regions. Convolutional neural networks are then employed to extract features, followed by classification and localization using conventional algorithms like Support Vector Pursuit (SVP) and DPM. However, real-world scenarios often involve multi-scale challenges for target objects, making traditional algorithms prone to false positives and missed detections. This fundamental limitation has driven the development of two distinct research directions in object detection algorithms:

1) How to ensure the scale characteristics of the detected object and improve the detection efficiency and accuracy.

2) How to improve the detection efficiency while ensuring the accuracy of detection.

In the target detection algorithm, it is divided into two categories, single-stage detection and double-stage detection.

One-stage[3] object detection algorithm: This single-stage detection algorithm employs an

end-to-end training approach, eliminating the need for a region proposal generation phase. Instead, it directly produces object confidence probabilities and positions within the training network.

Two-stage[4] object detection algorithm: This algorithm divides the detection process into two distinct phases.

1) Generate region proposals, which mainly search for the approximate location of objects on the map, and then use the region proposal to generate the possible location of objects.

2) After generating a large number of candidate regions through the region candidate generation process, these regions are rough locations generated by the candidate region algorithm that may contain object information. To further obtain the precise location of the target object, confidence calculation and position regression are performed on each candidate box generated by the region recommendation system to determine the final exact position of the object.

Currently, the Two-Stage algorithm demonstrates certain accuracy advantages, but its dual-phase approach sacrifices detection speed. In contrast, the One-Stage algorithm employs an end-to-end training method during object detection, which reduces detection accuracy while enhancing speed.

To balance detection speed and accuracy for seamless mobile integration, this paper employs the lightweight YOLOv5s model.

This paper addresses real-time rapid detection on mobile devices by replacing traditional convolutional methods with depth separable convolution (DSC) for model deployment. The GIoU algorithm is employed to mitigate false detection or missed detection in scenarios with high object overlap.

2. Related Work

2.1 Mobilenet Convolutional Neural Network

The MobileNet network employs deep separable convolution to construct a lightweight deep convolutional neural network, replacing traditional convolution methods. This architecture replaces conventional convolution with two stages: deep convolution and pointwise convolution. The $D_K \times D_K$ $D_K \times D_K$ convolution process operates as follows: Given an input feature map with M channels and an output feature map with N channels, where the network's convolution kernel size is unspecified, the computational parameters of the deep separable convolution model are specified in Equation (1).

$$D_K \times D_K \times D_F \times D_F \times M + M \times N \times D_F \times D_F \tag{1}$$

The computational cost of standard convolution in the network model is shown in Equation (2).

$$D_K \times D_K \times M \times N \times D_F \times D_F \tag{2}$$

The ratio of computational complexity between deep separable network model and traditional convolutional network model is calculated as shown in Equation (3).

$$\frac{D_K \times D_K \times D_F \times D_F \times M + M \times N \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_K^2} \tag{3}$$

The formula demonstrates that deeper network layers enhance compression efficiency while drastically reducing computational demands, making inference operations more feasible on devices with limited processing power. The MobileNet convolutional neural network comprises 28 layers, including an input layer, 13 convolutional layers, an average pooling layer, and a fully connected layer.

Deep convolution is shown in Figure 1:

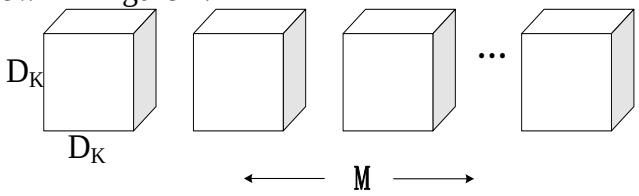


Figure 1: Deep convolution

The pointwise convolution is shown in Figure 2:

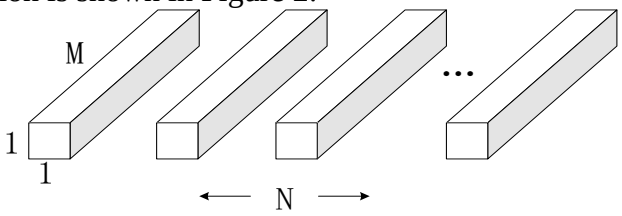


Figure 2: Pointwise convolution

The standard convolution is shown in Figure 3:

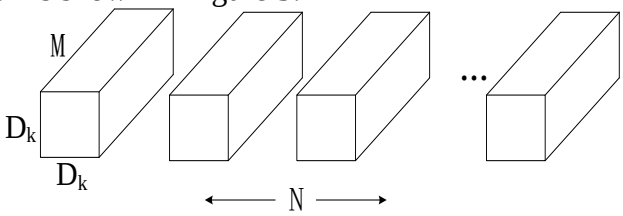


Figure 3: Standard convolution

2.2 SSD (Single Shot Multi-Box Detector) Convolutional Neural Network

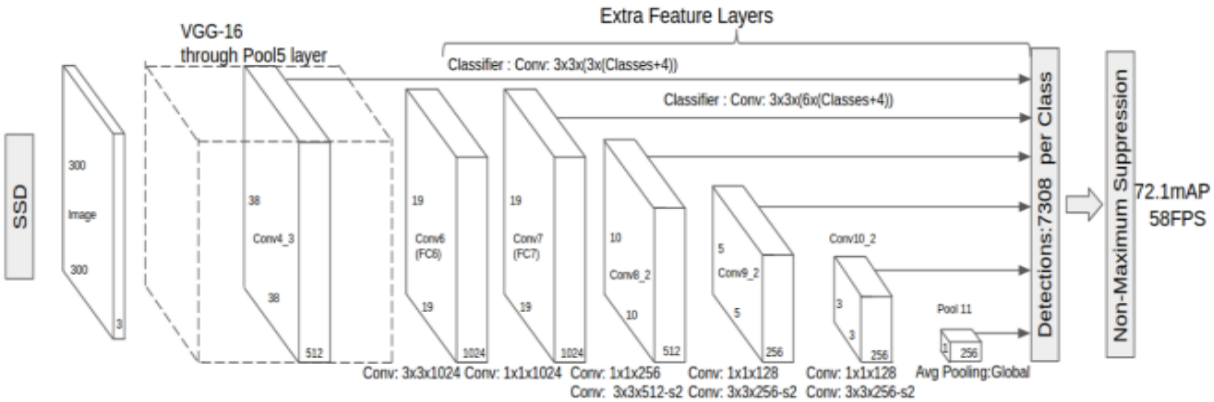


Figure 4: Single Shot MultiBox Detector

The Single Shot Multibox Detector (SSD), a single-stage object detection algorithm proposed by Wei Liu at ECCV 2016, features a network architecture as illustrated in Figure 4. Its backbone network consists of truncated VGG16 convolutional neural networks, where the two fully connected layers are replaced with convolutional layers. Four additional convolutional layers are appended to extract deeper feature information. The SSD loss function combines a location loss and a confidence loss through a weighted average, with the total loss function expressed in Equation 4.

$$L(x,c,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)) \quad (4)$$

In the formula, N denotes the number of matching default boxes, x indicates whether an object is present in the matched box (taking values {0,1}), g represents the ground truth box, and c denotes the confidence level that the selected box belongs to category p.

2.3 SSD-Mobilenet Network Model

The SSD-Mobilenet architecture replaces the traditional convolutional layers in the VGG network with deep separable convolutions. It adds eight convolutional layers after the conv13 layer, selecting six of these for object detection. As shown in Figure 5, the modified convolutional structure replaces standard convolutions with deep separable ones. By combining the speed and high precision of the SSD network with the low computational cost of Mobilenet, this model significantly reduces training time while maintaining high detection accuracy and minimizing parameter size.

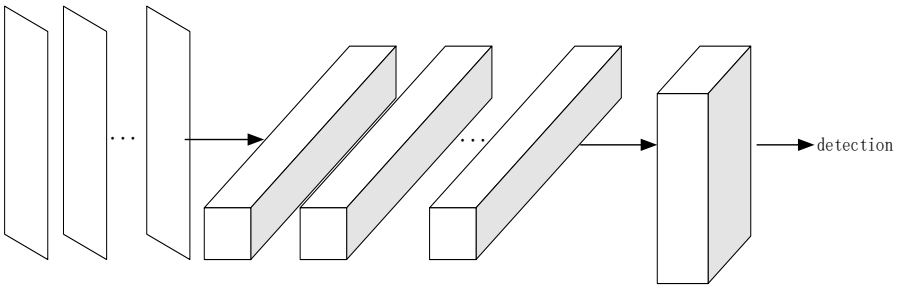


Figure 5: SSD-Mobilenet Network Model

2.4 GIoU

GIoU (Generalized Intersection over Union) is a novel regression loss function for object detection bounding boxes. Unlike the traditional IoU (Intersection over Union), GIoU incorporates the concept of minimum enclosing region, which generates more effective gradient signals and enhances model training performance.

3. Algorithm Improvement and System Implementation

3.1 Model Compression Design

In traditional object detection algorithms, representative models like Fast-RCNN, Faster-RCNN, and YOLOV3 utilize conventional convolutional neural networks for training. These models achieve high precision and low latency on PCs, delivering excellent detection performance. However, embedded devices face computational challenges due to processors with significantly lower processing power compared to PCs. Models trained on PCs often exhibit excessive memory consumption and computational demands on mobile devices due to their large size, high precision, and substantial parameter counts. This results in high latency and memory consumption during inference on mobile platforms. To address this, deep separable convolutional networks are employed during training to reduce parameter counts and achieve model compression.

3.2 SSD-Mobilenet with GIoU

The conventional NMS[5] algorithm employs the IOU metric during post-processing to eliminate

redundant bounding boxes generated by model predictions, thereby identifying and highlighting the most probable object locations in images. The pseudocode of this traditional NMS algorithm is as follows:

Input: $B=\{b_1,...,b_N\}$, $S=\{s_1,...,s_N\}$, N_t

Box B is the initial detection box

S is the score for each box in cross-validation.

N_t is the NMS threshold N_t

Begin

D={}

While $B \neq \varnothing$ do

m $\leftarrow$ argmax S

M $\leftarrow$ b_m

D $\leftarrow$ D $\cup$ M;B $\leftarrow$ B-M

For b_i in B do

If iou(M, b_i) $\geq$ N_t then

B $\leftarrow$ B- b_i ;S $\leftarrow$ S- s_i

End

End

Return D,S

End

Traditional Non-Maximum Suppression (NMS) suffers from computational inefficiency due to its sequential processing mode, where calculating IoU (Intersection Over Union) hinders efficiency. The conventional rejection mechanism mechanically selects thresholds, leading to false negatives [6]. Since these thresholds are empirically determined, they cannot accurately capture optimal values. The evaluation criterion of IoU, which only considers the overlapping area between two boxes, fails to comprehensively describe the relationships between predicted boxes. The traditional NMS rejection mechanism operates as follows:

$$S_i=\begin{cases}0 & \text{IoU(M,B}_i\text{)} \geq \text{threshold} \\ s_i & \text{IoU(M,B}_i\text{)} < \text{threshold}\end{cases} \tag{5}$$

For adjacent boxes $IoU \geq N_t$ $IoU \geq N_t$ of the threshold, the traditional NMS approach is to brute-force their scores to zero, which is prone to missed detection in cases with occlusion.

Traditional NMS algorithms rely solely on box overlap area information while neglecting scale information in images [7]. When two similar objects overlap extensively, edge suppression occurs, leading to missed detections of lower-scoring objects within the same category. To address this, we adopt the GIoU method for overlapping object calculation. This approach provides effective gradients through the enclosed region when no intersection exists [8-9], enabling continuous model optimization. Moreover, it demonstrates scale insensitivity and better discriminates prediction boxes with different overlap patterns.

The GiOU algorithm is as follows:

$$\text{GIoU}=\text{IoU}-\frac{\text{A}^c-\text{U}}{\text{A}^c} \tag{6}$$

The formula A^c above defines the minimum bounding box area, where U represents the intersection area between the predicted and ground truth boxes. As per this formula, GIoU equals

IoU when the predicted and ground truth boxes overlap, and GIoU=-1 when they do not overlap. Therefore, when GIoU falls within the range [-1,1], it indicates an overlapping region. Leveraging this characteristic of GIoU, we can further distinguish object categories and improve detection accuracy.

4. Experimental Results and Data Analysis

4.1 Training Environment

The Caffe deep learning framework, developed by Jia Yangqing, is a neural network learning framework that supports command-line interfaces, Python, and MATLAB. It enables seamless switching between CPU and GPU. This experiment was conducted on Linux 16.04 with Caffe as the deep learning framework, and the experimental environment is detailed in Table 1.

Table 1: Configuration Table

Configuration	Version
Python	2.7
Caffe	Caffe2
Sublim	Sublimtext3
Cuda	10.0
Cudnn	7.0.5
Graphics card	Nvidia GeForce 1050Ti
System	Linux16.04

4.2 Dataset Creation

The dataset for this experiment was compiled by professional physics educators who captured teaching equipment in real classroom environments and manually annotated the data. To adapt to evolving teaching conditions and enhance the model's robustness against interference during training, we implemented multi-angle and multi-scene data collection from practical teaching scenarios. After acquisition, images underwent manual filtering to eliminate those with excessive exposure, overly complex scenes, or distortion. For annotation, we utilized LabelImage, a widely adopted tool based on industry-standard VOC and COCO datasets.

The number of categories and their names in the dataset are shown in Table 2.

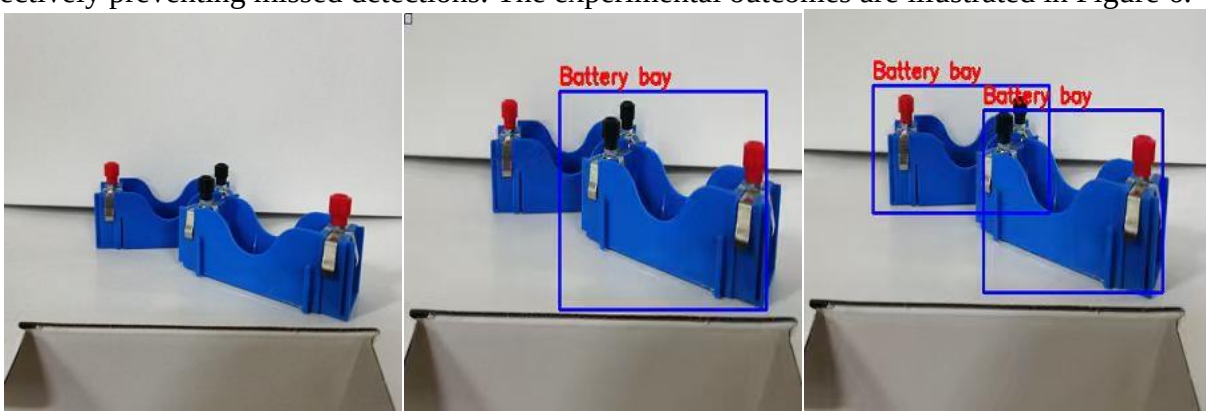
Table 2: Data Table

Category Name	Number
Switch	1765
Light bulb	2246
Resistance	2363
Battery	2246
Battery bay	1534
Voltmeter	4151
Ammeter	1524

4.3 Training and Result Analysis

Experimental results demonstrate that our SSD-Mobilenet algorithm, which employs GIoU as the loss function instead of IoU, significantly improves the detection of circuit components by

effectively preventing missed detections. The experimental outcomes are illustrated in Figure 6.



(a) Original Image (b) Original Network Detection Image (c) The Net add GIoU Detection image

Figure 6: Experimental outcomes

The training process in this study was conducted on a GPU. As the number of training iterations increased, the loss value gradually converged and continued to decrease. After reaching a certain number of iterations, the loss value exhibited gradual fluctuations while stabilizing, with a very slow decline, indicating the training was essentially complete. The total training steps amounted to 120,000, achieving an accuracy rate of 86.3% for detection. When the quantized model was deployed on an NPU device, it delivered 34.1FPS detection performance on mobile devices with 72.3% accuracy.

5. Conclusion

This study utilizes the SSD-Mobilenet-GIoU hybrid object detection algorithm. By integrating deep separable convolutional networks with GIoU optimization, the model achieves enhanced detection efficiency even when objects exhibit high overlap, effectively preventing false positives and missed detections. Furthermore, the model's deployment on NPU devices significantly expands its practical applications across diverse scenarios.

References

[1] Gao Wen, Chen Xilin. Computer Vision: Principles of Algorithms and Systems [M]. Tsinghua University Press, 1999.

[2]Zhang H, Wang K F, Wang F Y. Advances and Perspectives on Applications of Deep Learning in Visual Object Detection [J]. Acta Automatica Sinica, 2017, 43(8):1289-1305.

[3] Feng Jingchuan, Hu Xiaolong, Li Bin. Research on Feature Fusion-Based Object Detection Algorithms [J]. Digital Technology and Applications, 2018,36(12):124-125.

[4] Wang Caiyun. Research Progress in Object Detection [C]// Proceedings of the 23rd Annual Conference on New Network Technologies and Applications, 2019, organized by the Network Application Branch of China Computer Users Association. 2019.

[5] Neubeck A, Gool L J V. Efficient Non-Maximum Suppression [C]// International Conference on Pattern Recognition. IEEE Computer Society, 2006.

[6] Li X, Di X, Liu M, et al. Feature-aligned distillation for dense object detection via refined semantic guidance and distribution consistency [J]. Computer Vision and Image Understanding, 2025, 262104519.

[7] Liu H, Wan Y, Zhang M, et al. LGRDet: A Light Object Detection Network for Gesture Recognition [J]. Concurrency and Computation: Practice and Experience, 2025, 37(25-26):e70350.

[8] Qiao W, Yin C, Huang W .Improved method for small target detection based on infrared sensing images and wearable IoT [J]. Journal of Radiation Research and Applied Sciences,2025,18(4):102002.

[9] Ha K C, Than M P, Nguyen H. Adaptive query allocation for dense object detection in deformable transformers [J]. Results in Engineering,2025,28107571.