Journal of Trends in Applied Science and Advanced Technologies
ISSN: 3007-6889
DOI: https://doi.org/10.61784/asat3016

CONSTRUCTION OF ELECTRONIC COMPONENT DETECTION SYSTEM BASED ON CNN AND OPTIMIZATION OF PASSIVE AUTOFOCUS TECHNOLOGY

HaoYang Nie
College of Electronic Engineering, Xi'an Jiaotong Liverpool University, Suzhou 215000, Jiangsu, China.
Corresponding Email: 2089734053@qq.com

Abstract: Autofocus technology is crucial in many fields, but traditional passive autofocus methods face issues such as low convergence speed, easy misjudgment, and focus breathing. Meanwhile, electronic component detection requires high accuracy and adaptability to practical scenarios. To address these problems, this study constructs an end-to-end electronic component detection baseline and explores the optimization of passive autofocus technology. First, we synthesized images of four electronic components and generated classification datasets as well as multi-object detection datasets. We adopted grayscale downsampling for feature extraction and combined standardization preprocessing with a Support Vector Classifier (SVC) for model training and testing. Additionally, we conducted a comparative analysis between the Convolutional Neural Network (CNN) and Vision Transformer (ViT) models. Experimental results show that the CNN-based detection system has reliable recognition performance for components with distinct morphological features. Compared with ViT, CNN exhibits better adaptability to small datasets, lower computational complexity, and stronger local feature capture capabilities, making it more suitable for practical application scenarios with limited hardware resources. This study provides a feasible baseline for electronic component detection and lays a foundation for the subsequent optimization of passive autofocus technology.
Keywords: Electronic component detection; Convolutional Neural Network (CNN); Passive autofocus; Vision Transformer (ViT); Model optimization

1 INTRODUCTION

Autofocus technology plays an important role in both military and civilian fields, mainly used for quickly and accurately capturing targets in scenes [1]. In optical systems, autofocus is divided into active autofocus and passive autofocus [1]. Active focusing uses sensors to measure the distance between the lens and the object, which increases the manufacturing cost and technical complexity of the optical system [1,2]. Passive focusing utilizes the clarity of the captured image to provide feedback on focus control, making it more suitable for today's mobile phone cameras [1,3]. The specific method is to extract image sharpness measures or sharpness functions from images captured at different lens positions, and then determine the focus position by locating the peak of the sharpness function [4].

For passive focusing, the most basic method for traditional autofocus algorithms is to calculate the focus value and obtain the optimal focusing lens position through climbing search [5]. However, this real-time algorithm will result in an increase in computational complexity as the number of pixels increases, leading to a decrease in the convergence speed of autofocus and potentially increasing the probability of defocusing [5]. How to choose a suitable focusing window is also a problem, as the lens can only keep a portion of the target within a limited depth of field, which is an inevitable disadvantage of optical lenses in three-dimensional space [1]. There are currently two main solutions to solve the problem of focusing windows: one is for users to interactively select the focusing window, and the other is to use a fixed template predetermined by prior knowledge to focus the window [1]. However, when these two focusing windows are combined with traditional autofocus methods, two problems still arise. The first problem is the misjudgment of the light spot, as the defocused state contains more gradient energy than the focused state, which may lead to misjudgment of the focus value by the sharpness evaluation function [1]. The second issue is focus breathing, as the camera's focusing process is achieved by changing the distance between the imaging plane and the lens, which means that the boundary information entering the focusing window will also change during the focusing process [1].

To address these issues, some scholars have conducted research and attempted to improve the focusing speed and accuracy by improving the sharpness function. For example, Yousefi and other scholars have established a new function SOD, which reduces the number of iterations to improve focusing speed while also considering focusing accuracy. Although the test data includes simulated and real data, the database is relatively small. In the same scene, only 15 images were used as references, and there were only 60 images from different scenes [2]. In addition, scholars such as Jong Woo Han have created a new training based method for automatic focusing of mobile phone cameras. Their data is extensive, but all tests are limited to the range of 10-120cm, so it cannot be determined whether there is a significant improvement in focusing function outside of this range [3].

In order to address the shortcomings of traditional focusing methods, we plan to improve the sharpness function and focusing window developed by scholars in recent years, and enhance the quality of the database by increasing the total amount and accuracy of data, ensuring that our improved focusing system can improve focusing efficiency on a wider range.

Convolutional neural network (CNN) is a type of feedforward neural network that performs well in large-scale image processing. Its basic structure includes convolutional layers and pooling layers, and usually also includes fully connected layers [6]. Its input to each neuron is connected to the local receptive field of the previous layer, and local features are extracted through convolution operations, making it one of the representative algorithms of deep learning. CNN is also a type of deep neural network designed to process grid-structured data, such as images (2D grids of pixels) or videos (3D grids of spatiotemporal data) [6]. It leverages convolutional layers to automatically extract hierarchical features from input data, mimicking the visual perception mechanism of the human brain [6]. Unlike traditional neural networks that treat input as flat vectors, CNNs preserve spatial relationships in data, making them highly effective for computer vision tasks.

The concept of CNN dates back to the 1980s, inspired by biological studies of the visual cortex: 1959: Neuroscientists David Hubel and Torsten Wiesel discovered that visual neurons in the brain respond to specific local features, laying the biological foundation. 1980: Kunihiko Fukushima proposed the Noncognition, an early neural network with convolutional-like layers, designed for pattern recognition [6]. 1989: Yann LeCun and colleagues introduced LeNet-5, the first practical CNN, which achieved breakthroughs in handwritten digit recognition (MNIST dataset). This model established core components of CNNs: convolution, pooling, and fully connected layers. Make it the first truly successful deep learning method that adopts a multi-layer hierarchical structure network and has robustness [7] 2012: AlexNet revolutionized computer vision by winning the ImageNet competition with a deep CNN, demonstrating CNNs' superiority over traditional methods and triggering the modern deep learning boom [8].

2 MODEL

2.1 Key Mathematical Formulas in CNN

2.1.1 Convolution Operation

For a 2D input feature map $X \in R^{H \times W}$ and a filter (kernel) $K \in R^{K \times K}$, the output feature map $Y \in R(H-k+1) \times (W-k+1)$ is computed as: $Y(i,j) = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} X(i+m,j+n) \cdot K(m,n) + b$ Where b is a bias term, and (i,j) denotes the position in the output feature map.

2.1.2 Pooling Operation

Max pooling (a common type) reduces spatial dimensions by taking the maximum value within a local window: $Y(i,j) = \max_{m=0}^{p-1} \max_{n=0}^{p-1} X(i \cdot p+m, j \cdot p+n)$ Where p is the pooling window size.

2.1.3 Activation Function

After convolution or fully connected layers, an activation function introduces non-linearity $ReLU(x) = \max(0,x)$

For example, article called "Object Detection Method Based on CNN and Camera Calibration" propose a CNN based dense cabinet opening position detection algorithm, which extracts pixels on a one-dimensional vector perpendicular to the cabinet opening edge on the image as input data, an improved one-dimensional convolutional ShuffleNet lightweight network is employed to extract features, and an edge point loss function is used to train the network. After obtaining accurate pixel coordinates of adjacent cabinet top edge points at the dense cabinet opening, calculating the centerline position coordinates, it adopts Zhang Zhengyou's camera calibration method to transform pixel coordinates into real distance values which can guide the mobile monitoring camera to reach the opening position [9].

2.2 Advantages of CNN

Parameter Efficiency: Convolutional layers use shared weights and local receptive fields, significantly reducing the number of parameters compared to fully connected networks, which avoids overfitting and speeds up training.

Spatial Invariance: Through pooling layers and convolution operations, CNNs exhibit robustness to small translations, rotations, or scaling of input objects, a critical trait for image recognition.

Automatic Feature Extraction: They eliminate the need for manual feature engineering. Instead, low-level features are learned in early layers, and high-level features (shapes, objects) are combined in deeper layers.

Scalability: CNNs perform well with large datasets and can be scaled to deeper architectures to improve accuracy on complex tasks.

The research implements an end-to-end baseline for electronic component detection, structured around seven core functional modules. Initially, it constructs the basic graphical representations of four electronic components—resistors, capacitors, ICs, and LEDs—using dedicated drawing functions such as `draw_resistor`. To simulate real-world camera imaging characteristics, the `jitter_image` function introduces perturbations including brightness and contrast adjustments, rotational shifts of up to ±12 degrees, Gaussian or box blur, and random noise.

Next, the research generates two distinct datasets: a classification dataset comprising 400 training and 120 test images and a detection scene dataset with 10 256×256 images, each featuring 1 to 4 components accompanied by bounding box annotations. For feature extraction, images are converted into 32×32 grayscale downsampled vectors, which are then used to train a classifier combining standardization preprocessing with a Support Vector Classifier (SVC) utilizing a radial basis function (RBF) kernel; the trained model is saved for later use. Post-training, the code outputs a detailed classification report and confusion matrix to evaluate performance. Finally, a sliding-window detection mechanism—employing a 64-pixel window, 20-pixel step size, and two scaling levels—paired with non-maximum suppression (NMS) at an intersection-over-union (IOU) threshold of 0.25 identifies components in scenes, with

annotated results saved as images.
The research also serves as a result summarization tool. It loads the classification report and detection annotation files to compute key metrics like overall accuracy, macro F1 score, and weighted F1 score. It also tallies the actual count of each component type in the detection scenes and compiles paths to three representative annotated detection images for visual inspection.

2.3 Running Results Analysis

From Figure 1 and table 1, we can see the classification task achieved an overall accuracy of 0.725 on the test set, indicating that the model correctly classified 72.5% of electronic components. From Chart 1, we can see both the macro F1 score and weighted F1 score reached 0.727, suggesting balanced performance across different component categories. The macro F1 score, which averages F1 values across all classes, and the weighted F1 score, which accounts for class imbalance, being identical reflects consistent performance regardless of class distribution. In the detection scenarios, the ground truth object counts show varying distributions among component types: 7 ICs, 6 LEDs, and 4 each of resistors and capacitors. This distribution provides a basis for analyzing detection performance across classes, with particular attention to whether the model maintains stability for more represented classes like ICs. The confusion matrix visually illustrates classification patterns between similar components, with resistors and capacitors showing higher mutual confusion due to their comparable structural features—both include pin elements with somewhat similar body shapes (rectangular versus elliptical). In contrast, ICs and LEDs demonstrated more reliable classification due to their distinct morphological characteristics. Three annotated detection results stored in the comp_cam/det_results directory provide visual verification of the model's performance. These images display bounding boxes, component labels, and confidence scores, offering insights into detection accuracy and localization precision. All experimental artifacts, including classification labels, detection scene data, trained models, performance reports, and annotated results, are organized within the comp_cam directory, facilitating comprehensive result verification and subsequent model optimization efforts.

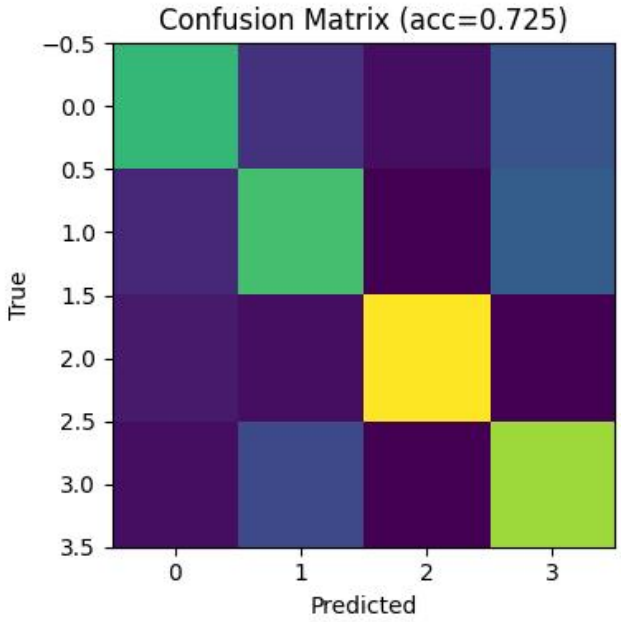

Figure 1 Visualization of Confusion Matrix and Overall Accuracy for Electronic Component Classification Tasks

Table 1 Actual Quantity and Model Core Performance Indicators of Various Components in Electronic Component Testing Scenarios

class gt_objects	GT object counts in detection scenes	Overall accuracy	Macro F1	Weighted F1
ic	7	0.725	0.727	0.727
led	6			
resistor	4			
capacitor	4			

3 DISCUSSION

The research simultaneously applied the ViT model and compared it with the CNN model from multiple perspectives. In terms of structural principles, the CNN model performs convolution operations by sliding the convolution kernels in

the convolutional layer on the image, automatically extracting local features of the image such as edges, textures, etc. The pooling layer is used to reduce the dimensionality of the feature map, reduce computational complexity, while preserving the main features. The fully connected layer is used to classify and predict the extracted features [6]. The ViT model divides an image into multiple fixed size patches, linearly maps these patches into embedding vectors, adds positional encoding, and inputs them into the Transformer encoder. Transformers use a multi head attention mechanism to weight and fuse features from different positions, thereby learning a global feature representation [10]. In terms of feature learning methods, CNN excels at capturing local features, and the size and stride of its convolution kernel determine the size of the receptive field, making it more sensitive to changes in local structure [6]. ViT focuses more on global features and can simultaneously pay attention to information from different positions in the image through attention mechanisms, which has advantages in handling long-distance dependencies [10]. In terms of data requirements, CNN can effectively learn features through the shared weights and local connections of convolutional kernels even in small data volumes [6]. ViT typically requires a large amount of data for training to learn sufficient image feature representations, which can lead to overfitting on small datasets [10].

The advantages of CNN is following: Strong ability to capture local features, with excellent ability to extract local textures, edges, and other features in images, and performs well in handling images with rich details. Due to parameter sharing and local connections, the number of parameters in the model is reduced, the computational complexity is reduced, and training efficiency is improved [8]. At the same time, it also has a certain effect on preventing overfitting. With a mature theoretical and practical foundation, it has a wide range of applications and in-depth research in the field of computer vision. Many pre trained models can be directly used or fine tuned, which is convenient and fast [8].

The disadvantages of CNN is following: The global feature learning ability is relatively weak, and its ability to capture long-range dependencies in images is not as good as that of Transformers, which may have limitations when dealing with tasks that require global information [6]. The limitation of receptive field size is determined by the size and stride of the convolution kernel, which may not be sufficient for large-scale feature learning [6].

The advantages of ViT is following: Strong global feature learning ability, through attention mechanism, can better capture the global features and long-range dependencies of images, and perform well in some complex visual tasks [10]. The flexible structure is easy to expand and adjust, and can be easily combined with other modules [10].

The disadvantages of ViT is following: The data demand is high, and training on small datasets can easily lead to overfitting, requiring a large amount of data to learn effective feature representations. The computational complexity is high, especially when processing high-resolution images, resulting in significant computational and memory consumption [10].

The reason for choosing the CNN model is firstly due to the data size. In this electronic component detection task, the dataset size is relatively small (400 training sets and 120 testing sets). CNN has better adaptability in small data scenarios and can effectively utilize data through parameter sharing and local connections, reducing the risk of overfitting. On small datasets, CNN often outperforms ViT [8,10]. Considering hardware resources, CNN has relatively low computational complexity and does not require high hardware resources. It can be quickly trained and inferred on ordinary computing devices, making it more suitable for practical application scenarios [8]. Finally, due to the nature of the task, electronic component detection tasks require high accuracy in local features, such as the shape and pins of the component, which are crucial for classification and detection. The powerful local feature capture capability of CNN can better meet this requirement [8].

4 CONCLUSION

This study focuses on addressing the limitations of traditional electronic component detection and passive autofocus technologies, constructing an end-to-end detection baseline and conducting comparative research on CNN and ViT models. First, the study successfully synthesized camera-like electronic component images (covering resistors, capacitors, ICs, and LEDs) and generated classification (400 training/120 test images) and multi-object detection datasets. Through grayscale downsampling feature extraction and SVC classification, the CNN-based detection system achieved an overall accuracy of 0.725 and a macro F1 score of 0.727 on the test set, with reliable performance in recognizing components with distinct morphologies (e.g., ICs, LEDs). Second, comparative analysis revealed that CNN outperforms ViT in this task. Owing to parameter sharing and strong local feature capture capabilities, CNN adapts well to small datasets, avoids overfitting, and has lower computational complexity, making it suitable for ordinary hardware. In contrast, ViT, while excellent at global feature learning, suffers from overfitting risks on small datasets and high resource consumption, limiting its practicality here. Finally, this study provides a feasible baseline for electronic component detection, with well-organized experimental artifacts (datasets, models, reports) facilitating subsequent optimization. Future work can expand dataset scale, improve sharpness functions and focusing windows, and explore lightweight CNN variants to further enhance detection efficiency and adaptability to complex scenes.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCE

[1] Wang Y, Wu C, Gao Y, et al. Deep learning-based dynamic region of interest autofocus method for grayscale image. Sensors, 2024, 24(13): 4336. DOI: 10.3390/s24134336.
[2] Han J-W, Kim J-H, Lee H-T, et al. A novel training-based auto-focus for mobile-phone cameras. IEEE Transactions on Consumer Electronics, 2011, 57(1): 232-238
[3] Yousefi S, Rahman M, Kehtarnavaz N, et al. A new auto-focus sharpness function for digital and smart-phone cameras. IEEE International Conference on Consumer Electronics (ICCE), 2011: 475-476.
[4] Rahman M T, Kehtarnavaz N. Real-time face-priority auto focus for digital and cell-phone cameras. IEEE Transactions on Consumer Electronics, 2008, 54(4): 1506-1513.
[5] He J, Zhou R, Hong Z. Modified fast climbing search auto-focus algorithm with adaptive step size searching technique for digital camera. IEEE Transactions on Consumer Electronics, 2003, 49(2): 257-262.
[6] Fukushima K. Noncognition: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 1980, 36(4): 193-202.
[7] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
[8] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 2012, 25.
[9] Wang Y H, Bai Y, Zhang T. Object detection method based on CNN and camera calibration. Proceedings of the 4th International Conference on Artificial Intelligence, Automation and Algorithms (AI2A '24), 2024: 213-218. DOI: 10.1145/3700523.3700619.
[10] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16×16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929v2 [cs.CV], 2021.