Hindawi
Journal of Sensors
Volume 2022, Article ID 9518910, 9 pages
https://doi.org/10.1155/2022/9518910

Research Article

Fast Method of Detecting Packaging Bottle Defects Based on ECA-EfficientDet

Zhenwen Sheng and Guiyun Wang

Shandong Xiehe University, Jinan, China

Correspondence should be addressed to Guiyun Wang; xiehewsq@sdxiehe.edu.cn

Received 29 September 2021; Revised 6 January 2022; Accepted 19 January 2022; Published 23 February 2022

Academic Editor: Chao Wang

Conventional methods of detecting packaging defects face challenges with multiobject simultaneous detection for automatic filling and packaging of food. Targeting this issue, we propose a packaging defect detection method based on the ECA-EfficientDet transfer learning algorithm. First, we increased the complexity in the sampled data using the mosaic data augmentation technique. Then, we introduced a channel-importance prediction mechanism and the Mish activation function and designed ECA-Convblock to improve the specificity in the feature extractions of the backbone network. Heterogeneous data transfer learning was then carried out on the optimized network to improve the generalization capability of the model on a small population of training data. We conducted performance testing and a comparative analysis of the trained model with defect data. The results indicate that, compared with other algorithms, our method achieves higher accuracy of 99.16% with good stability.

1. Introduction

To guarantee high-quality products from automatic packaging production lines, defect inspections are indispensable. Usually, such inspections look for defects in the caps, labels, packaging, spraying code, etc. In particular, the cap and label significantly affect product quality and their appearance and inspection are therefore of great importance.

Conventional packaging defect inspections are mostly made using equipment based on image processing techniques. For instance, Toxqui-Quitl et al. [1] proposed a PET bottle defect inspection method with self-adaptive gamma adjustments to bottle images through a frequency filtering technique for highly accurate inspections of the face, wall, and bottom of bottles. Zheng et al. [2] proposed a texture area defect detection algorithm based solely on phase change. Zhou et al. [3] combined mean squared cyclic detection and entropy partition and proposed an improved random cyclic detection method to determine defective regions on the bottoms of bottles. The above-mentioned works use image processing for packaging defect inspection. However, they require many experiments to determine the judgment rules. In addition, they require complex inspection environments and face challenges with simultaneous detections in multiple categories. Therefore, their advantages for practical inspection are limited.

Simultaneously detecting defects in multiple categories thus remains an issue to be solved. Deep learning has recently shown potential in the field of classification and detection and is a promising solution in the field of packaging defect inspection.

The development of deep learning techniques began with the representative convolution neural network architecture, LeNet-5 [4], followed by the iconic deep learning algorithm, AlexNet [5]. Later algorithms included VGG [6], ResNet [7], MobileNet [8], and EfficientNet [9]. Zhou et al. [10] improved the size and number of kernels on the standard LeNet5 model and integrated the PSO optimization algorithm and proposed an improved LeNet5 model for fault detection of liquid plunger pumps. Wang et al. [11] proposed an AlexNet-based convolutional neural network method to deal with defects from a data-driven perspective. Zhang et al. [12] proposed a cost-sensitive residual convolutional neural network for unbalanced data defect detection.

Marques [13] proposed a convolutional neural network diagnosis system using EfficientNet, which effectively improved the accuracy of the medical decision-making system. Michele et al. [14] explored the applicability of the MobileNet V2 deep convolutional network by fine-tuning the pretrained MobileNet neural network. The results show that 100% classification accuracy can be achieved based on MobileNet V2. As network architectures became more complex, their feature extraction capabilities increased as well. Deep learning networks have demonstrated outstanding performance in the field of image identification. In particular, feature extraction networks based on deep learning have made great progress in object detection. Object detection falls into two categories: one- stage object detection and two-stage object detection. One-stage object detection has a broader range of applications, due to its superior balancing of speed and accuracy. EfficientDet, proposed by Tan et al. [15] from the Google Brain Team, and YOLOv4, proposed by Bochkovskiy et al. [16], are the best-performing object detection models. However, there is little research on the use of object detection for packaging defect inspection. We used EfficientDet in this study because it is more accurate than YOLOv4.

In addition, the object detection algorithm requires a large amount of data to build the model. However, it is difficult to obtain packaging defect samples. Therefore, to improve the model's generalization capability, the model must be specifically designed and the data must be processed. Regarding the latter, data augmentation and transfer learning are effective techniques to improve model accuracy with small datasets. Regarding the model design, effective channel feature importance prediction can improve the generalization capability. For example, Hu et al. [17] proposed the squeeze-and-excitation network (SE-Net), which makes predictions on channel importance during convolution to improve the overall accuracy of the model. Subsequently, a large number of scholars applied SE to some networks [18, 19]. Wang et al. [20] showed that SE-Net can also be used for dimension reduction. To prevent any unnecessary loss in accuracy due to dimension reduction, they designed the effective channel attention network (ECA-Net) for improved model accuracy.

In summary, conventional imaging processing methods have difficulties simultaneously detecting defects in multiple categories. Moreover, the scarcity of defect samples renders trained models weak in terms of generalization. Thus, we propose a fast packaging defect detection method based on the ECA-EfficientDet transfer learning algorithm. The contributions of this study are twofold. First, we incorporated the ECA mechanism in a backbone feature extraction network and designed an ECA-Convblock convolution block that is capable of predicting the channel importance during convolution. This suppresses channels that carry no information, to make specific representations of object features and improve the defect detection accuracy. Second, we used the mosaic augmentation technique on our sample data. This effectively enhanced the sample complexity and therefore improved the generalization capability. In addition, we adopted heterogeneous data transfer learning during the training process and we utilized the Mish activation function to improve the robustness of our model in complex environments.

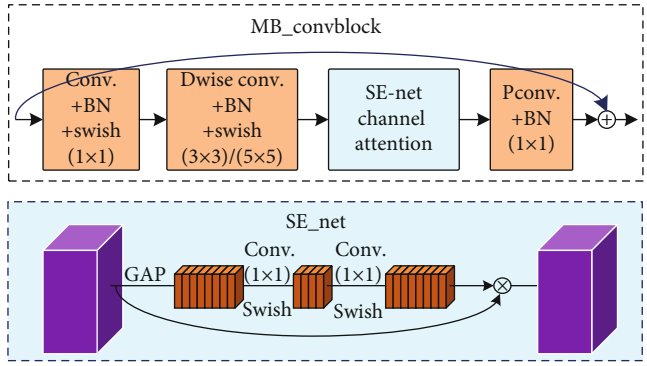

Figure 1: MB_Convblock convolution block.

The remainder of this paper is organized as follows: Section 2 provides an overview of the structure of the EfficientDet algorithm. Section 3 describes the framework and design of the packaging bottle defect detection model. In Section 4, we describe our evaluation of the proposed method through the analysis and comparison of simulation results. Section 5 offers conclusions.

2. EfficientDet Model Architecture

EfficientDet [21] is an object detection algorithm proposed by the Google Brain Team at CVPR2020. It can be viewed as an extension of EfficientNet [9], extending from classification to object detection. EfficientDet balances efficiency and accuracy. Its overall architecture consists of the backbone feature extraction network, an enhanced feature extraction network, and box/class determination.

The backbone feature extraction network for EfficientNet contains eight convolution layers. Except for the ordinary convolution (Conv. + BN + Swish) operation in the first layer, all the other layers are formed by piling up the convolution block (MB_Convblock), as illustrated in Figure 1. The overall structure is illustrated in Figure 2(c1, I). First, the dimensions are increased through an ordinary 1 1 convolution. Then, a 3 3 or 5 5 deep separable convolution is carried out with the SE-Net channel attention mechanism. The remaining seven layers of the network can be described as follows: MB_Convblock piles once in the second and eighth layers, piles twice in the third and fourth layers, piles three times in the fifth and sixth layers, and piles four times in the seventh layer. The fourth, sixth, and seventh layers all adopt 5 5 separable convolution (whereas the other layers use 3 3). Therefore, the depth of the feature extraction network increases. Finally, the fourth, sixth, and eighth layers (P3–P7) of the backbone network are outputted as effective feature layers for the next stage of feature fusion and reuse.

The enhanced feature extraction network repeatedly performs upsampling and downsampling on the features extracted by the backbone network to achieve effective fusion and high utility of the extracted features, as illustrated

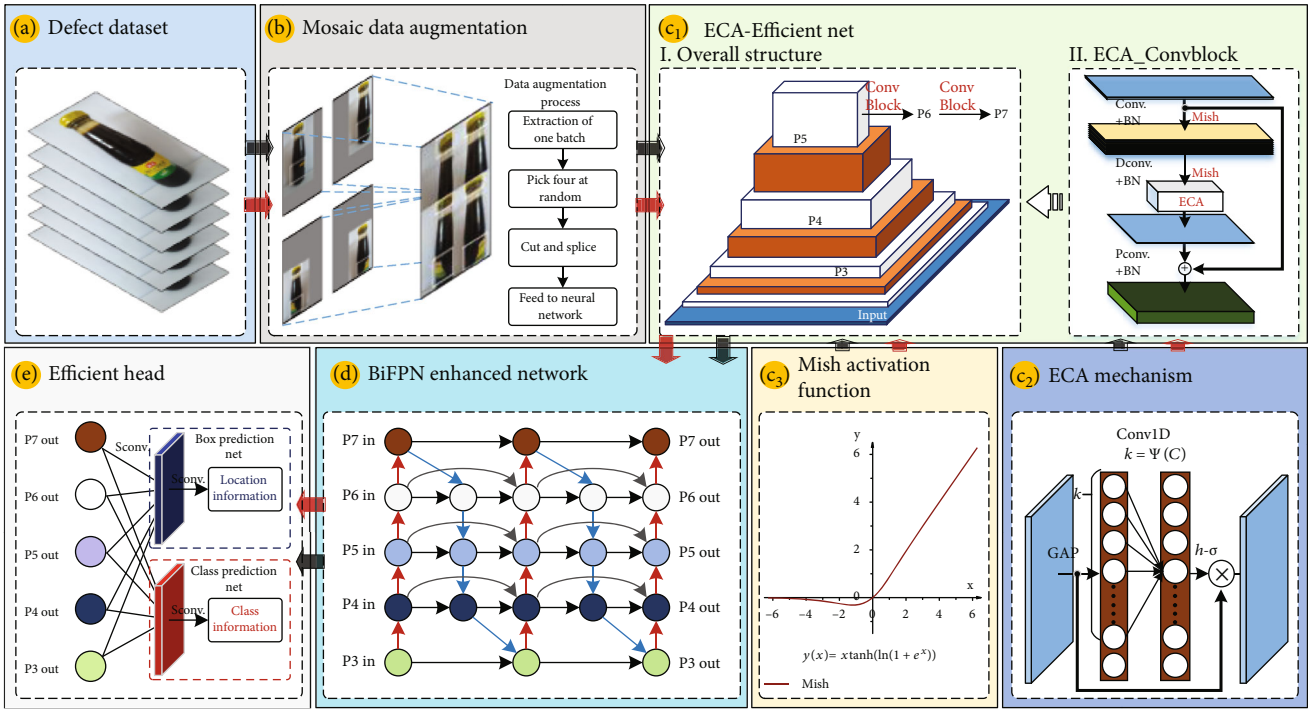

Figure 2: Framework for the defect inspection method based on the ECA-EfficientDet algorithm.

in Figure 3. The enhanced feature extraction network is formed by three serially connected BiFPNs. Feature layers P6 and P7 are obtained by downsampling from P5. After fusing the features, five feature maps with rich semantic information are outputted for the final class and box information detection. Two deep, separable convolution operations on the fused five feature maps are thus used to predict the final class and box information.

3. Research on EfficientDet Optimization

The subject of our study is the detection of packaging defects. Since there are many categories of defects but samples of defects are few, directly implementing the current algorithms cannot meet the accuracy requirements of practical production. Since the backbone feature extraction network is directly related to the feature extraction capability, we optimize three aspects of the backbone network. First, we utilize data augmentation to improve as much as possible the complexity of the sample data to enhance the model's generalization capability. Second, we modify the structure of the network to enhance the feature extraction capability. And third, we conduct transfer learning on the optimized model to improve the overall robustness of the model. Our design of the bottle packaging defect detection model is illustrated in Figure 2. It consists of five parts: sample input (a), mosaic data augmentation (b), the ECA-EfficientNet backbone feature extraction network (c1–c3), the BiFPN enhanced feature extraction network (d), and the efficient head box/class result output (e).

The detection flow can be generally described as follows. First, before training, we perform random mosaic data augmentation. Then, we feed the augmented data into the optimized ECA-EfficientNet backbone network for multiscale feature extraction. Five feature maps (P3–P7) of different scales are obtained as a result. Next, we input the acquired

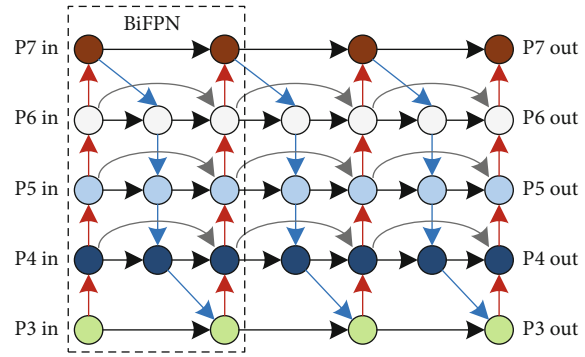

Figure 3: BiFPN-enhanced feature extraction network.

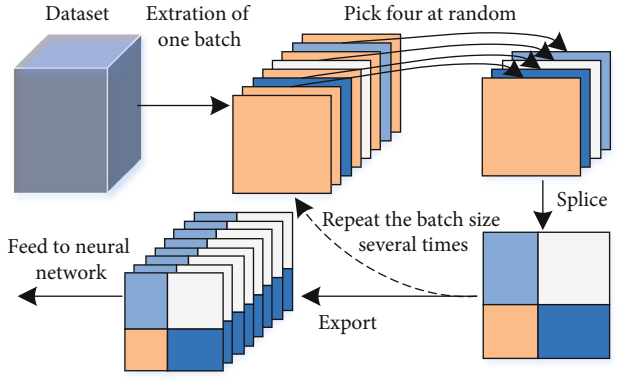

Figure 4: Mosaic data augmentation flow.

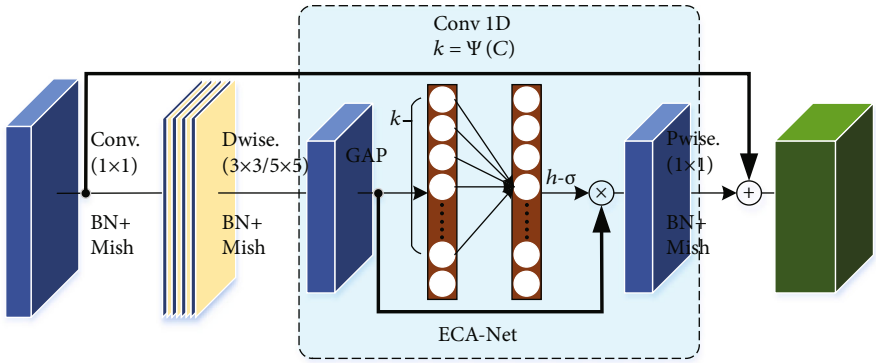

Figure 5: ECA_Convblock convolution block.

feature maps into the enhanced feature extraction network, which is formed by three BiFPN modules concatenated serially for feature fusion and reuse. Finally, we conduct deep separable convolution on the five fused feature maps to detect the box and class information of the defect samples.

3.1. Mosaic Data Augmentation Structure. Mosaic data augmentation [16] is a new type of data augmentation scheme proposed at the same time as the YOLOv4 algorithm. It was developed from CutMix augmentation [22] and is capable of improving the image background complexity with a small population of samples to improve the generalization capability of models. Therefore, we adopted mosaic augmentation in this study. Considering that EfficientDet is an end-to-end network, we also adopted an end-to-end network and embedded data augmentation into model training. The procedures are illustrated in Figure 4.

First, we select a batch from the defect sample set. Next, we randomly pick four images from the batch and generate a new image by stitching the top-left, top-right, bottom-left, and bottom-right sections of the four selected images. This step is repeated until a new batch is created. Finally, the newly generated batch and previous samples are fed to the model for training. Notice that the total number of samples used for training does not change, which enables the network to perform data augmentation before training in an end-to-end manner.

3.2. Reconstruction of the Backbone Feature Extraction Network

3.2.1. Design of the ECA-Convblock Convolution Block. The backbone network of EfficientDet is EfficientNet, which is described above. The backbone network is all piled from the MB_Convblock convolution block, as illustrated in Figure 1. In terms of algorithm design, EfficientNet adopts deep, separable convolution to decrease the number of parameters in the model. To avoid insufficiency in feature extraction, the authors of the algorithm utilized the SE-Net [17] channel attention mechanism after layer-by-layer convolution to improve the specificity in the extracted features by predicting channel importance during the convolution. However, according to the structural principles of SE-Net,

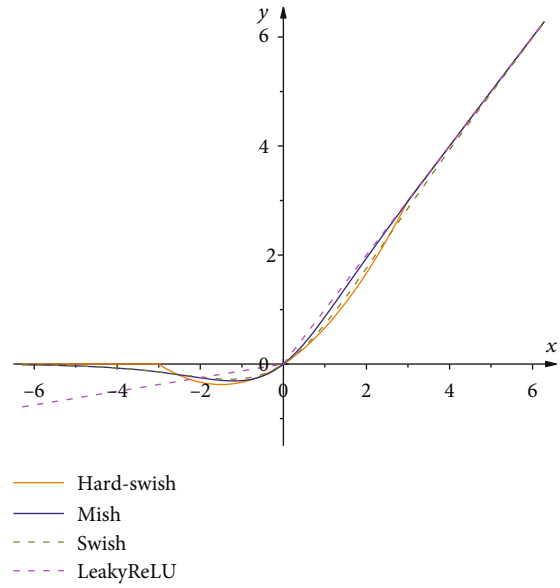

Figure 6: Various activation functions.

there is a reduction in feature dimensions when applying the attention mechanism, resulting in a loss in model accuracy. To avoid this, we introduce an ECA_Convblock convolution block as illustrated in Figure 5 to prevent accuracy loss due to the dimension reduction when applying SE-Net.

The structure of the ECA_Convblock convolution block is very similar to that of MB_Convblock. The major difference is that there is no dimension reduction in the ECA channel importance prediction. Instead, the weighted channel is obtained from a 1D convolution of size k, avoiding loss in accuracy due to dimension reduction. Here, k is obtained self-adaptively from the number of input feature map channels. The equation can be written as follows [20].

$$k = \psi(C) = \left| \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right|_{\text{odd}}, \tag{1}$$

where C is the number of input sample channels, odd means taking the odd integer number of the input, and b and γ are heuristic parameters, which are set to $b = 2$ and $\gamma = 1$, per the

Figure 7: Sample of the packaging bottle defect dataset; the red frame is the ground-truth box.

values provided by the ECA authors. We maintain the outmost residual edge, which many researchers have shown is beneficial for stability when updating weights.

3.2.2. Introduction of the Mish Activation Function. After showing the ECA's improvement in model's accuracy, we proceed to discuss the impact of the activation function on model accuracy. EfficientNet adopts Swish as the activation function [23]. Based on the backbone feature extraction network, we experimented with three other activation functions: the hard-Swish function [24], the LeakyReLU function [25], and the Mish function [26], as illustrated in Figure 6.

Figure 6 shows that all four activation functions make adjustments in the negative value region. These are beneficial for preventing some of the weights from losing gradients during model training. However, the excessive dispersion of LeakyReLU in the negative value region makes updating the weights unstable. The hard-Swish function, on the other hand, has poor gradient flow during weight updating, since it utilizes a hard zero boundary. Only the Mish function, similar to the Swish function, avoids saturation by clipping the positive value region and smoothly processing the negative value region, stabilizing weight updating during the iterations to achieve good generalization capability. Therefore, we adopted the Mish activation function in our model. Through comparison, we found that the Mish function's performance is excellent.

3.2.3. Heterogeneous Data Transfer Learning. It is well known that transfer learning is capable of applying model parameters and weights learned from a large amount of data to a small dataset to improve the generalization capability when the relevant dataset is small. Therefore, to further improve the overall robustness of the model after structural optimization and because the number of defect samples is small, we adopted transfer learning to improve the model's generalization capability.

When training the optimized model (ECA-EfficientDet), in the early stage, we utilized the VOC2007 dataset. VOC2007 is a comparatively large dataset for object detection. It contains rich data types and objects. However, due to the differences between VOC2007 and the defect sample data, transfer learning is heterogeneous in nature. After the first stage, we transferred the obtained weights to the second stage of training with defect sample data, after which we obtained the model for defect detection.

4. Experiments and Results

4.1. Source of Data. For practical packaging defect inspection, we obtained defect samples from a common production line packaging product, namely, bottles. A total of 1200 samples were collected with cap and label defects. Cap defects included mislocated caps, absent caps, and normal caps. Label defects included mislocated labels, absent label, damaged labels, and normal labels. The sample of the packaging bottle defect image is shown in Figure 7, and the number details are summarized in Table 1.

We utilized professional labeling software, LabelImg, to build the ground-truth from the sample data and automatically generated XML files to store practical information about the defect data, which were used together with the samples during model training and testing.

4.2. Model Training and Evaluation Metrics. Our experimental environment was Windows 10 OS, Anaconda Python 3.7, NVIDIA RTX 2070 GPU, CUDA 10.1, and cuDNN 7.6.5.32. We used the Keras library for programming. For data allocation, we selected 1000 samples from the dataset for training, among which 900 samples were used for model training and the other 100 for crossvalidation. The remaining 200 samples were used for performance testing after model training. In terms of the training methodology, we adopted transfer learning for overall model training. Then, we conducted model optimization using our own defect sample data. We

Table 1: Defect sample number by category.

Category	Count
Cap defect	
No cap	263
Misloc cap	612
Normal cap	325
Label defect	
No label	293
Misloc label	335
Damaged label	345
Normal label	227

configured hyperparameters with 512×512 input data. We enabled early-stop, such that the iterations would not continue after the model had optimally converged. To speed up the training, we froze the backbone network during the first 50 epochs, during which time only the enhanced feature extraction network's weights were updated. The batch size was set to 8, and the initial learning rate was 0.001. In the next 50 epochs, we unfroze the backbone network for weight updating to optimize the global network. During this time, the batch size was set to 4 and the initial learning rate was 0.00005.

We used two commonly used metrics for object detection model evaluation, precision (P) and recall (R) [27].

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{2}$$

where TP is the number of samples that the model correctly detected, FN is the number of samples that the model failed to detect, and FP is the number of samples that the model falsely detected. Considering that the precision and recall can vary for different confidence thresholds, we also recorded the P-R (precision-recall) to obtain the average precision (AP) [27].

$$\mathrm{AP} = \int_0^1 P(R)dR. \tag{3}$$

The AP value is the area under the P-R curve, where P represents precision and R represents recall. For multicategory defects, it is common to use the mean average precision (mAP) of all categories for overall evaluation.

4.3. Model Evaluation and Performance Comparison and Analysis

4.3.1. Evaluation of Model Training. It can be observed in Figure 8 that, in general, the loss gradually decreases as the number of epochs increase until convergence, indicating that the model's prediction error gradually decreases. In addition, there is a sudden drop in the loss curves for the crossvalidation set after epoch 43. This is because the model freezes the backbone network in the early stage of the training. In later iterations, after unfreezing the network, the model is trained globally and the loss of the crossvalidation set approaches

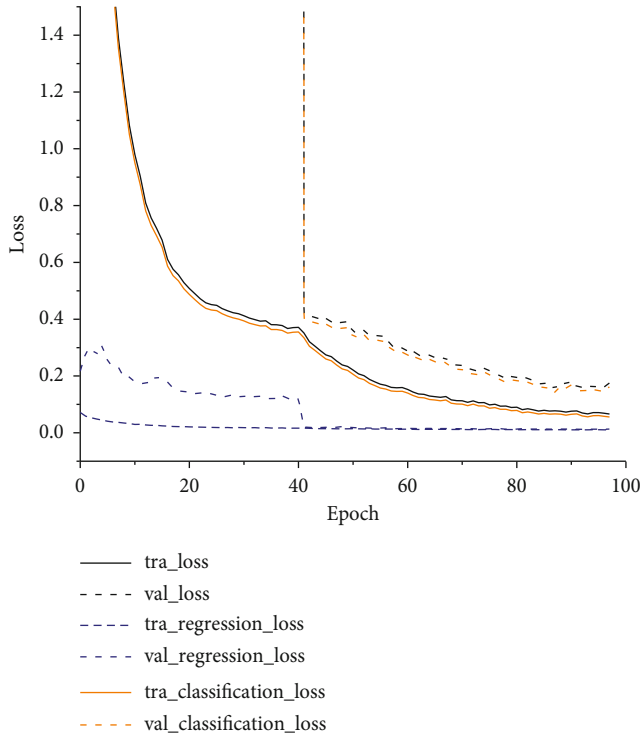

Figure 8: Model loss function.

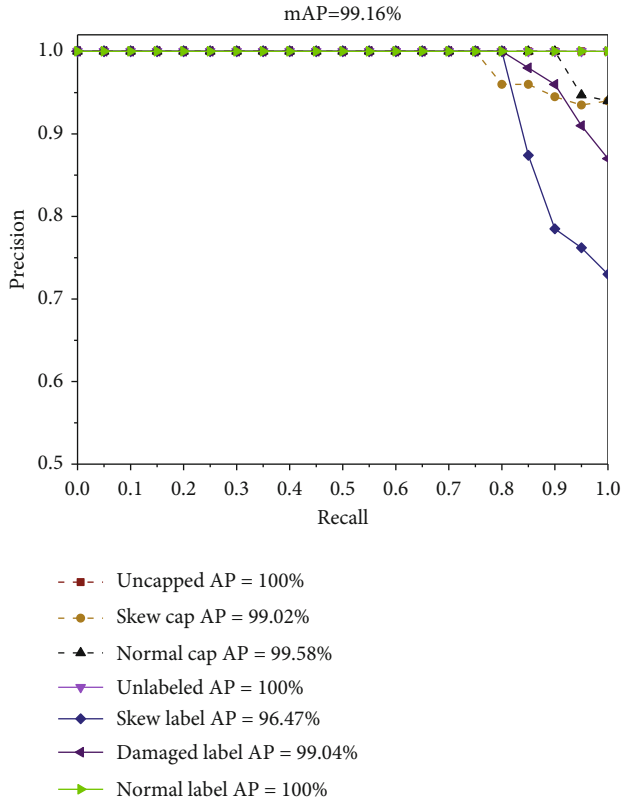

Figure 9: P-R curve.

Table 2: Impact of transfer learning and ECA on model performance.

Method	Parameters (10^7)	FPS	Transfer learning?	mAP (/%)	Standard deviation (mAP)
EfficientDet	0.389	45.62	Yes	97.75	0.62
			No	23.12	0.65
(Mish)ECA-EfficientDet	**0.325**	**41.72**	**Yes**	**99.16**	**0.54**
			No	30.12	0.61

Table 3: Impact of mosaic trick on model performance.

Method	Parameters (10^7)	FPS	Mosaic trick?	mAP (/%)	Standard deviation (mAP)
EfficientDet	0.389	45.62	Yes	97.75	0.62
			No	95.45	0.59
(Mish)ECA-EfficientDet	**0.325**	**41.72**	**Yes**	**99.16**	**0.54**
			No	97.88	0.63

that of the training set, indicating a progressive improvement in the model's generalization capability.

4.3.2. Performance Testing. After model training, we imported the files containing the learned weights into our program and tested the model with 200 samples. The results are illustrated in Figure 9 as *P-R* curves. The solid lines denote label defects and the dashed lines denote cap defects.

From the *P-R* curve, the AP value for each type of defect can be readily obtained. Overall, our method achieved a mAP of 99.16%. For each individual defect category, except for a slightly lower precision for mislocated labels, our method achieved accuracy above 99% for all the other categories. This indicates that our method is capable of meeting the practical needs of applications.

4.3.3. Performance Comparison and Analysis. In our investigation, we conducted several experimental simulations to validate the optimized model and demonstrate the advantages of our method. In Tables 2–5, the results of our method or strategy are presented in bold.

First, we investigated the impact of transfer learning and the ECA mechanism on model performance. We compared ECA-EfficientDet and EfficientDet with and without transfer learning. The simulation results are summarized in Table 2.

It can be observed in Table 2 that, compared with the model without transfer learning, the precision of the model with transfer learning was substantially higher. This demonstrates the effectiveness of transfer learning for small-sized sample data and its ability to avoid hindering the model's generalization capability due to limited samples. In addition, the ECA mechanism improved the model's precision as well, albeit at the cost of some loss in detection speed.

Next, in order to prove the necessity of the mosaic method. We compared ECA-EfficientDet and EfficientDet (transfer learning) with and without mosaic trick. The results are shown in Table 3.

It can be seen from the results that the mosaic trick can effectively improve the performance of ECA-EfficientDet and EfficientDet. In addition, we also compared various activation functions on the ECA-EfficientDet with transfer learning. The results are summarized in Table 4.

Table 4: Impact of activation functions on model performance.

Method	Activation function	mAP (/%)	Standard deviation (mAP)	FPS
ECA-EfficientDet	Swish [23]	98.63	0.58	45.98
	Hard-Swish [24]	95.64	0.61	44.08
	Hard-Swish [24]	97.12	0.59	49.36
	Mish [19]	**99.16**	**0.54**	**41.72**

Table 5: Performance of the proposed method compared with other single-stage object detection algorithms.

Method	Parameters (10^7)	mAP (/%)	Standard deviation (mAP)	FPS
YOLOv3 [28]	6.200	92.60	0.66	31.08
YOLOv4 [16]	6.443	97.22	0.58	22.64
YOLOv4-tiny [16]	0.606	91.23	0.63	157.47
SSD [29]	2.628	97.55	0.71	62.56
EfficientDet [21]	0.389	97.75	0.62	45.62
Our method	**0.325**	**99.16**	**0.54**	**41.72**

It can be observed in Table 4 that the Mish function has higher precision than EfficientDet's default activation function, Swish. However, Mish function's detection speed is lower than that of the others.

Because our method performs end-to-end single-stage object detection, we compared our ECA-EfficientDet (with Mish as the activation function) with transfer learning to other object detection algorithms. The results are summarized in Table 5.

It can be observed in Table 5 that our method achieved higher precision (mAP = 99.16%) than the other five

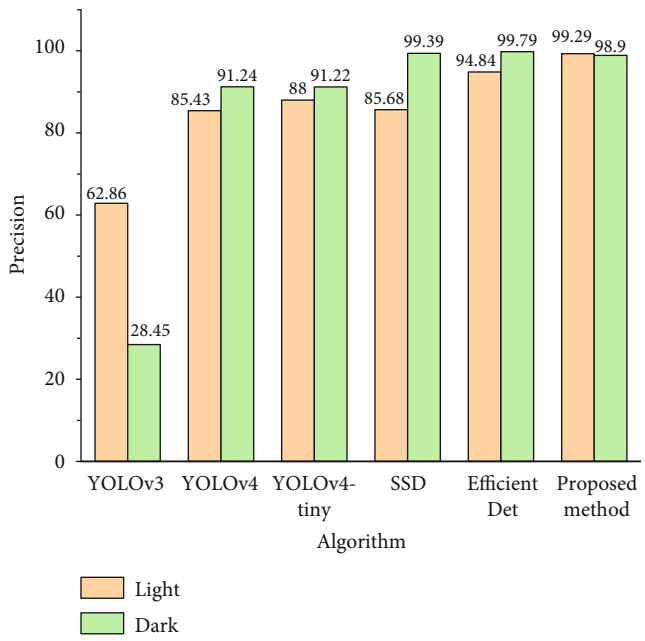

Figure 10: Comparison of our method to other object detection algorithms in light and dark environments.

single-stage object detection algorithms. In terms of detection speed, our method was faster than YOLOv3 and YOLOv4 and close to EfficientDet. In addition, although YOLOv4-tiny had the fastest detection speed, its accuracy was the worst, rendering it disadvantageous in practical applications.

To test the model's stability in complex environments, we lightened and darkened the 200 test samples using the OpenCV library, thus generating two subsets that simulated detection in light and dark environments. We tested our object detection algorithm with these two subsets. The results are illustrated in Figure 10. The dark color represents testing in a dark environment, and the light color represents testing in a light environment.

It can be observed in Figure 10 that, compared with other object detection algorithms, our method was more precise in both environments and there were few differences between the two subsets, indicating good model stability.

In summary, although our method is slightly inferior in terms of detection speed, it shows good performance in terms of precision and stability. Our detection speeds can meet the needs for practical inspection. Therefore, we expect that our method will be advantageous for applications of packaging defect detection.

5. Conclusions

We proposed a packaging defect detection method based on the ECA-EfficientDet object detection algorithm. We validated its effectiveness and advantages on a sample dataset of defects. Our results show the following: (1) The proposed method solves a challenging problem in conventional machine vision algorithms. It can simultaneously detect multiple defect objects, effectively reducing inspection costs on production lines. (2) Our design of ECA-Convblock in the backbone feature extraction network prevents dimension reduction in model channel importance prediction and enhances high-quality expressions of object features, effectively improving the defect detection accuracy. (3) The incorporation of Mosaic data augmentation and the Mish activation function into the model and the adoption of heterogeneous-data-based transfer learning for model training effectively enhance the model's generalization capability and improve its robustness in complex environments. It should be pointed out that the algorithm has obvious advantage in accuracy performance and algorithm stability, but it is slightly insufficient in detection speed. Therefore, combining light weight with the model in this paper to improve the model's actual generation and detection will be carried out in the follow-up research.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] C. Toxqui-Quitl, J. Cardenas-Franco, A. Padilla-vivanco, and J. Valdiviezo-Navarro, "Bottle Inspector Based on Machine Vision, Proc," in *Image Processing: Machine Vision Applications VI*, pp. 1–10, International Society for Optics and Photonics, 2013.

[2] Y. Zheng, Y. Wang, X. Zhou, X. Jiang, Y. Peng, and Y. Liu, "Empty bottle texture area defect detection based on POBT," *Journal of Electronic Measurement and Instrumentation*, vol. 31, no. 4, pp. 549–558, 2017.

[3] X. Zhou, Y. Wang, Q. Zhu et al., "A surface defect detection framework for glass bottle bottom using visual attention model and wavelet transform," *IEEE T. Ind. Inform.*, vol. 16, no. 4, pp. 2189–2201, 2020.

[4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[6] Z. J. Gao, Y. J. Zhang, and Y. Li, "Extracting features from infrared images using convolutional neural networks and transfer learning," *Infrared Physics & Technology*, vol. 105, p. 103237, 2020.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[8] A. G. Howard, "MobileNets: efficient convolutional neural networks for mobile vision applications," *Computer Vision and Pattern Recognition*, 2017, arXiv: 1704.04861v1.

[9] M. Tan and Q. V. Le, "EfficientNet: rethinking model scaling for convolutional neural networks, Proc," in *2019*

International Conference on Machine Learning(ICML), pp. 1–11, 2019.

[10] Y. Zhou and G. P. Li, "Intelligent fault diagnosis of hydraulic piston pump combining improved LeNet-5 and PSO hyperparameter optimization," *Applied Acoustics.*, vol. 183, p. 108336, 2021.

[11] Z. J. Wang, W. L. Zhao, and W. H. Du, "Data-driven fault diagnosis method based on the conversion of erosion operation signals into images and convolutional neural network," *Process Safety and Environmental Protection*, vol. 149, pp. 591–601, 2021.

[12] H. Zhang, L. X. Jiang, and C. Li, "CS-ResNet: cost-sensitive residual convolutional neural network for PCB cosmetic defect detection," *Expert Systems With Applications*, vol. 185, p. 115673, 2021.

[13] G. Marques, "Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network," *Applied Soft Computing Journal*, vol. 96, 2020.

[14] A. Michele, V. Colin, and D. D. Santika, "MobileNet convolutional neural networks and support vector machines for palmprint recognition," in *4th International Conference on Computer Science and Computational Intelligence*, vol. 157, pp. 110–117, 2019.

[15] T M P R, "Efficient Det: scalable and efficient object detection," in *Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10778–10787, 2020.

[16] A. Bochkovskiy, C. Wang, and H. Liao, "YOLOv4: optimal speed and accuracy of object detection," 2020, arXiv preprint, arXiv: 2004.10934v1.

[17] J. Hu, L. Shen, G. Sun, and S. Albanie, "Squeeze-and-excitation networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 42, no. 8, 2020.

[18] X. Li, X. Shen, Y. Zhou, X. Wang, and T. Q. Li, "Classification of breast cancer histopathological images using interleaved DenseNet with SENet (IDSNet)," *PloS one*, vol. 15, no. 5, 2020.

[19] Z. Li, K. Jiang, S. Qin, Y. Zhong, and A. Elofsson, "GCSENet: A GCN, CNN and SENet ensemble model for microRNA-disease association prediction," *PLOS Computational Biology*, vol. 17, no. 6, p. e1009048, 2021.

[20] Q. Wang, B. Wu, P. Zhu, P. Li, and Q. Hu, "ECA-net: efficient channel attention for deep convolutional neural networks," in *Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11531–11539, 2020.

[21] M. T. R. Pang and Q. V. Le, "EfficientDet: scalable and efficient object detection," in *Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10778–10787, 2020.

[22] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: regularization strategy to train strong classifiers with localizable features," 2019, arXiv preprint, arXiv: 1905.04899v1.

[23] P. Ramachandran, B. Zoph, and Q. V. Le, "Swish: a self-gated activation function," 2017, arXiv preprint arXiv:1710.05941v1.

[24] A. Howard, "Searching for MobileNetV3," in *Proc. 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324, 2020.

[25] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models, proc," in *Proceedings of the 30th International Conference on Machine Learning*, pp. 1–6, 2013.

[26] D. Misra, "Mish: a self regularized non-monotonic neural activation function," 2019, arXiv preprint arXiv:1908.08681v1.

[27] K. Li and L. Cao, "A review of object detection techniques," in *Proc. 2020 5th International Conference on Electromechanical Control Technology and Transportation (ICECTT)*, pp. 385–390, 2020.

[28] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," *Computer Vision and Pattern Recognition*, 2018, arXiv: 1804.02767v1.

[29] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multi box detector," in *Proc. 14th European Conference on Computer Vision (ECCV)*, pp. 21–37, Springer International Publishing, 2016.