Automatic Recognition And Classification Of Product Packaging Images Based On Convolutional Neural Network

You Li[1] and Chen Jiang[2*]

[1] College of Art and Design, Wuhan Technology and Business University, Wuhan, Hubei 430065, China
[2] School of art and media, Wuhan College, Wuhan, Hubei 430065, China
[*] Corresponding author. E-mail: chen1jiang2@outlook.com

Received: Feb. 12, 2026; Accepted: Apr. 04, 2026

Traditional packaging image recognition methods rely on manual detection or manual feature extraction, which has the problems of low efficiency and poor adaptability, while existing deep learning models still face challenges when dealing with complex backgrounds, small target detection, and category imbalance. The purpose of this paper is to design an efficient and reliable product packaging image recognition system. By proposing a multi-scale attention fusion network (MSAFNet), this paper integrates a lightweight hybrid backbone network (combined with EfficientNet and ResNet), a dual attention module (DAM) to enhance feature focusing, an adaptive multi-scale feature pyramid (AFPN) to optimize multi-scale fusion, and a multi-task learning framework to jointly optimize detection and classification. The proposed model achieves an mAP@0.5 of 89.6% on the COCO dataset, outperforming the baseline by 4.4% while maintaining real-time performance at 40 FPS with only 8.9 M parameters. It demonstrates strong robustness with a low performance degradation rate of 10.8% and good generalization with only 6.1% cross-dataset degradation. These results highlight its effectiveness in balancing accuracy, efficiency, and scalability for industrial applications. The architecture design and system validation confirm its suitability for automated packaging detection tasks. However, performance under extreme occlusion remains a limitation and can be improved through future self-supervised learning approaches.

Keywords: convolutional neural network; products; packaging images; automatic identification; classification

http://dx.doi.org/10.6180/jase.202609_32.037

1. Introduction

With the growth of industrial automation, automatic recognition of product packaging images has become vital for food safety, quality control, and logistics. Traditional manual or feature-based methods are inefficient and subjective, while CNNs face challenges with complex backgrounds, small targets, class imbalance, and real-time constraints, limiting practical deployment. The contribution of this paper is to achieve the collaborative improvement of accuracy and efficiency (for example, mAP@0.5 in the test reaches 89.6%, and the number of parameters is only 8.9 M), which provides a scalable solution for industrial applications and verifies the generalization and practicability of the model through system experiments.

2. Materials, methods and related works

Convolutional Neural Networks (CNNs) have advanced image recognition, enabling industrial automation and intelligent detection.

1. Evolution and optimization of core algorithms for image classification

   Image classification, a core task in computer vision, focuses on improving accuracy and efficiency through model optimization. Early work by Chaganti et al. [1]

fused SVM with CNNs for stability but struggled with complex textures. Liu et al. [2] used multi-level feature aggregation, improving accuracy by ∼5% at the cost of high computation. Greeshma and Gripsy [3] combined HOG and LBP with CNNs for low-resolution images, though generalization suffered. Alom et al. [4] leveraged Inception-Residual networks to reduce vanishing gradients, achieving 83.1% top-5 accuracy on ImageNet with higher training complexity. EvoD-CNN by Hassanzadeh et al. [5] optimized layers via genetic algorithms, reaching 96.2% on CIFAR-10 but with longer search time. Tripathi [6] noted excessive parameters limit real-time performance, while Hasan et al. [7] simplified kernels for edge devices, losing only 2.1% accuracy.

2. Research on the adaptation of industry to specific application scenarios

   In practical applications, CNNs need customized designs to handle domain-specific challenges like complex backgrounds and small targets. In agriculture, Yang et al. [8] achieved 98% apple recognition via transfer learning, but performance dropped under changing illumination. Ren et al. [9] highlighted multi-scale feature fusion for defect detection, though limited labeled data hampers deployment. In food analysis, Liu et al. [10] detected fat content with $< 5\%$ error, but struggled in noisy environments. Zhang et al. [11] reached an F1 of 0.89 for waste classification, yet rare classes were missed. Adi et al. [12] achieved $> 90\%$ accuracy in foreign object detection, but low-contrast targets remained challenging. Masood et al. [13] improved scene recognition using contextual features.

3. Integration of emerging architectures and interdisciplinary technologies

   The integration of CNNs with quantum computing and spiking neural networks has opened new directions. Li et al. [14] proposed a quantum deep CNN achieving 99.2% on MNIST, but requiring quantum hardware. Chen et al. [15] optimized quantum kernels, reducing error by 3.5%, though classical-quantum interfaces remain inefficient. Basha et al. [16] used quantum dilated CNNs (Dice 0.91) with weak noise resistance. Ngu and Lee [17] converted CNNs to SNNs, cutting energy use by 60% with $\sim$ 8% accuracy loss. Guizilini et al. [18] applied 3D packing for self-supervised depth estimation, reducing KITTI errors by 11%. Jacob and Darney [19] developed lightweight CNNs for IoT, enabling real-time recognition but losing detail. Zhang et al. [20] improved FCNs for packaging segmentation (IoU 0.85) but struggled with extreme deformations. Aburass et al. [21] enhanced rotation robustness by 15% via geometric invariance, yet non-rigid deformations remain challenging. These works expand CNN designs but hardware-algorithm integration remains difficult.

4. Enhancing model interpretability and robustness

   Enhancing model transparency and robustness against interference are crucial for ensuring reliable deployment. Dai et al. [22] critically analyzed the feature bias of CNNs and found that the model overly relies on texture features rather than shape features. After mitigating the bias through adversarial training, the classification consistency improved by 12%. Nauta et al. [23] proposed a neural prototype tree structure that visualizes the decision-making process as prototype matching.

   Cao [24] investigates modern packaging design through the lens of big data, highlighting how designers optimize product appearance and enrich packaging connotations to enhance consumer perception.

3. Model construction

3.1. Model overview

This paper proposes MSAFNet, a CNN-based model for automatic packaging image recognition, integrating a lightweight backbone, attention mechanism, multi-scale feature pyramid, and adaptive loss to handle small objects, complex backgrounds, and class imbalance. Its overall architecture is shown in Fig. 1, highlighting the core innovations.

1. Lightweight hybrid backbone network: Combining the lightweight design of EfficientNet and the residual connection of ResNet, it ensures accuracy and reduces computational costs.

2. Dual Attention Module (DAM): It integrates channel and spatial attention to enhance feature representation capabilities.

3. Adaptive Feature Pyramid Network (AFPN): It dynamically integrates features of different scales to improve small target detection performance.

4. Multi-task learning framework: It simultaneously completes target detection (such as packaging defect localization) and image classification (such as packaging type identification), and improves generalization capabilities through weight sharing.

5. Improved loss function: It combines Focal Loss and PolyLoss to solve the problem of category imbalance.

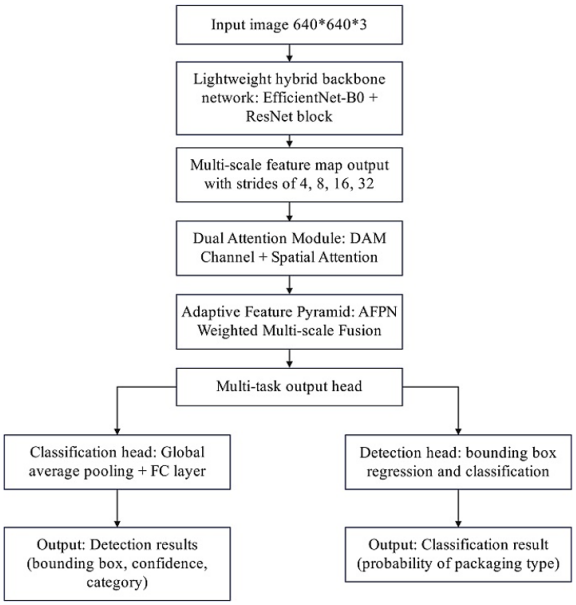

Fig. 1. Overall architecture of MSAFNet. The framework integrates a hybrid backbone network, attention modules, and a multi-scale feature pyramid to improve feature extraction and object detection performance for product packaging images.

The proposed MSAFNet framework integrates multiple advanced components to achieve efficient and accurate packaging image recognition.

3.2. Lightweight hybrid backbone network

In order to balance the computational efficiency and feature extraction ability, this paper uses EfficientNet-B0 as the basis and introduces the residual block of ResNet to build a hybrid backbone network. EfficientNet-B0 and ResNet are selected due to their complementary advantages in feature extraction.

$$Y = F(X, \{W_i\}) + X \tag{1}$$

The hybridization of EfficientNet-B0 and ResNet is expected to outperform single-backbone approaches due to their complementary characteristics. The selection of EfficientNet-B0 and ResNet as the hybrid backbone is motivated by their complementary feature extraction capabilities and efficiency trade-offs.

3.3. Dual attention module (DAM)

Inspired by CBAM (Convolutional Block Attention Module), this paper designs a dual attention module (DAM) to deal with the importance of features in channel and spatial dimensions in turn. DAM is embedded between the backbone network and the feature pyramid, and its structure is shown in Fig. 2.

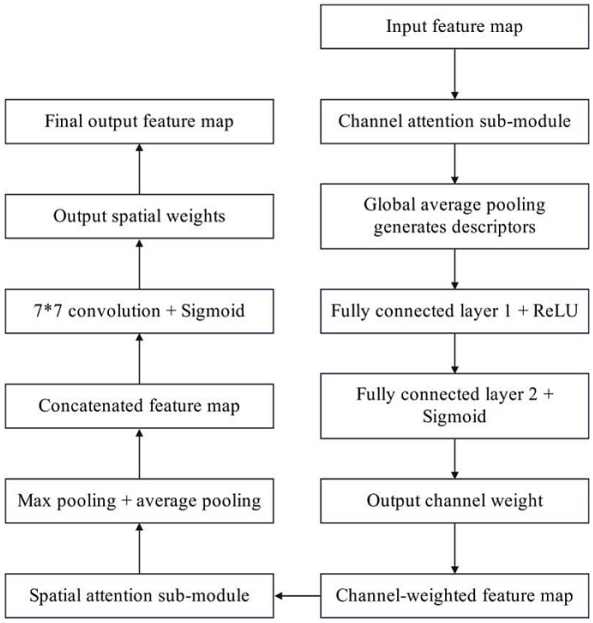

Fig. 2. Structure of the Dual Attention Module (DAM). The module captures both channel and spatial attention information to enhance feature representation and highlight important regions in packaging images.

Channel attention submodule: It employs a squeeze-and-excitation mechanism. The input feature map $U \in \mathbb{R}^{H\times W\times C}$ is first processed by global average pooling to compress spatial information and generate channel descriptors $z \in \mathbb{R}^{C}$:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} U_c(i, j) \tag{2}$$

Then, the dependency relationship between channels is learned through the two fully connected layers, and the channel weights s_c are output:

$$s_c = \sigma(W_2 \delta(W_1 z)) \tag{3}$$

Among them, δ is the ReLU activation function, σ is the Sigmoid function, and W_1 and W_2 are learnable parameters. The weighted feature map U' is:

$$U' = s_c \cdot U \tag{4}$$

$$M_s = \sigma\left(f^{7\times 7}\left(\left[F\,\mathrm{avg}_{\max}[]\right]()\right)()\right) \tag{5}$$

The final output feature map U'' is:

$$U'' = M_s \cdot U' \tag{6}$$

where U, U′, and U″ denote the input feature map, the channel-refined feature map, and the final attention-enhanced feature map, respectively.

3.4. Adaptive multiscale feature pyramid network (AFPN)

This paper improves BiFPN and proposes AFPN, which adaptively fuses multi-scale features using learnable weights (Fig. 3). Given input feature maps $\{P_2, P_3, P_4, P_5\}$, AFPN applies the adaptive fusion formula:

$$P_{\text{out}} = \sum_i \frac{\omega_i}{\varepsilon + \sum_j \omega_j} \cdot P_i \tag{7}$$

where P_i represents the input feature maps at different pyramid levels, ω_i denotes the learnable fusion weight, and $\varepsilon = 0.0001$ is used to avoid numerical instability.

The improvement of AFPN over BiFPN is primarily attributed to its adaptive weighting mechanism, which dynamically adjusts the contribution of feature maps from different pyramid levels. Unlike BiFPN, where feature fusion weights are fixed after normalization, AFPN learns the importance of each scale during training based on the input data distribution.

Among them, ω_i is the weight of each feature map, and numerical instability is avoided by fast normalization $(\varepsilon = 0.0001)$. AFPN takes a bi-directional (top-down and bottom-up) path to enhance the feature flow by jumping connections with the following formula:

$$P_6^{\text{td}} = \text{Conv}\left(\frac{\omega_1 \cdot P_6 + \omega_2 \cdot \text{Re size}\,(P_7)}{\omega_1 + \omega_2 + \varepsilon}\right) \tag{8}$$

$$P_6^{\text{out}} = \text{conv}\left(\frac{\omega_1' \cdot P_6 + \omega_2' \cdot P_6^{td} + \omega_3' \cdot \text{Re size}\left(P_5^{\text{out}}\right)}{\omega_1' + \omega_2' + \omega_3' + \varepsilon}\right) \tag{9}$$

AFPN outputs enhanced multiscale features $\{N_2, N_3, N_4\}$ for subsequent multi-task learning heads.

During feature aggregation, the adaptive weights dynamically regulate the contribution of each pyramid level according to the spatial characteristics of the input features.

The weights are initialized uniformly and updated through backpropagation based on their contribution to multi-scale feature fusion. In scenarios with significant scale variation, shallow feature maps (high-resolution) receive increased weights for small objects, while deeper feature maps (low-resolution) are emphasized for larger objects. The feature fusion process is defined in a structured manner, where weighted feature maps from different pyramid levels are normalized and aggregated to produce enhanced multi-scale representations.

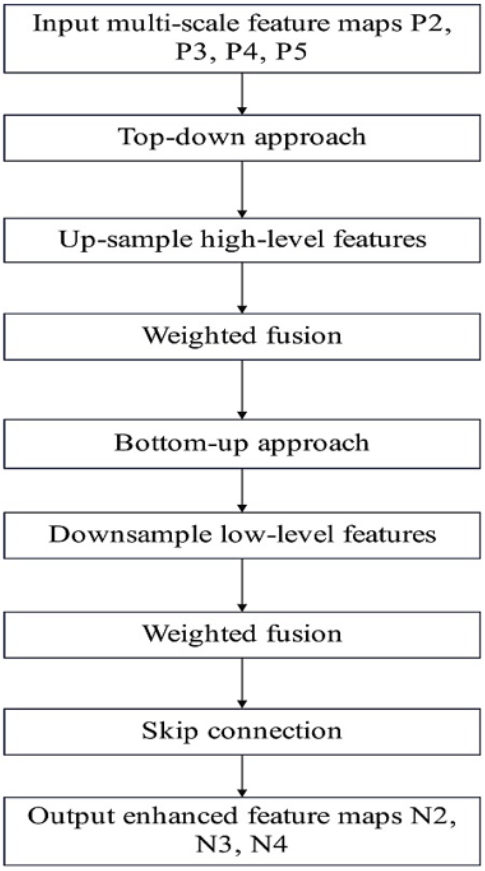

Fig. 3. Structure of the Adaptive Feature Pyramid Network (AFPN). The module performs multi-scale feature fusion to effectively detect objects of different sizes and improve detection accuracy.

3.5. multi-tasking learning header

MSAFNet adopts a multi-task learning framework that handles object detection and image classification simultaneously:

Multi-task optimization, gradients from the detection head (regression and classification) and the parallel classification branch are computed independently and scaled by λ_1 and λ_2. During training, gradients from the detection and classification branches are computed independently and scaled using predefined weighting factors to control their relative influence.

Classification header: Global average pooling followed by a fully connected layer, which outputs the probability distribution of packaging types (such as carton, plastic,

metal, etc.).

$$L_{\text{total}} = \lambda_1 L_{\text{det}} + \lambda_2 L_{\text{cls}} \tag{10}$$

where L_{det} represents the detection loss, L_{cls} denotes the classification loss, and λ_1 and λ_2 are balancing weights used to control the contribution of each task, which are set to 1.0 and 0.5 in the experiments. This decoupling reduces feature competition and improves spatial precision when multiple objects are closely packed.

The weighting coefficients for detection and classification tasks are selected to ensure balanced optimization within the multi-task learning framework

3.6. Improved loss function

Bounding box loss: EIoU Loss (Enhanced Intersection over Union) is used to solve the width-height coupling problem of CIoU:

$$L_{\text{box}} = 1 - \text{IoU} + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \frac{\rho^2\left(\omega, \omega^{gt}\right)}{c_\omega^2} + \frac{\rho^2\left(h, h^{gt}\right)}{c_h^2} \tag{11}$$

Classification loss: Using PolyLoss, the polynomial form of Focal Loss is generalized to improve learning on hard samples:

$$L_{\text{cls-det}} = -\sum_{c=1}^{C}(1-p_c)^{\gamma}\log\left(p_c\right) + \varepsilon_1 p_c^2 \tag{12}$$

Among them, γ is the focusing parameter and ε_1 is the polynomial coefficient.

$$L_{\text{cls}} = -\sum_{i=1}^{N} y_i \log\left(\hat{y}_i\right) + \varepsilon_2\left(1-\hat{y}_i\right)^2 \tag{13}$$

The proposed loss formulation integrates the advantages of Focal Loss and PolyLoss to effectively address class imbalance and enhance learning from hard samples in packaging classification tasks.

4. Result and discussion

This experiment evaluates MSAFNet's performance in packaging image recognition across accuracy, robustness, efficiency, interpretability, generalization, and real-time capability. Controlled tests and ablation analyses assess its reliability in complex industrial scenarios.

4.1. Test methods

The datasets include: COCO (Common Objects in Context) dataset (URL: http://cocodataset.org/), which contains 80 object categories covering daily packaged items; ImageNet dataset (URL: http://image-net.org/), which includes 1000 categories and contains packaging-related images; Open Images Dataset v6 (URL: https://storage.googleapis.com/openimages/web/index.html), which contains 1.9 million images covering diverse packaging scenarios.

Table 1 presents all evaluation metrics used in this study to provide a clear and standardized description of the experimental setup. Data augmentation techniques are employed to simulate real-world variations in packaging environments. These include random horizontal flipping with a probability of 0.5 , random rotation within $\pm 15^{\circ}$, brightness and contrast adjustments with scaling factors ranging from 0.8 to 1.2, and Mosaic augmentation to enhance contextual diversity and improve small object detection.

To further verify the effectiveness of the model, two additional experiments were conducted:

1. Cross-dataset generalization is evaluated by pre-training MSAFNet on COCO train2017 and fine-tuning on three external datasets: Open Images Dataset v6 (packaging subset), PackNet (https://github.com/packnet-dataset), and Google Open Images (packaging categories).
2. Real-time test: Evaluate the inference efficiency of the model on different hardware platforms, including high-end GPU (NVIDIA RTX 3080), edge device GPU (NVIDIA Jetson Xavier NX), and CPU (Intel i7-10700K).

The selection process involved filtering dataset annotations using predefined keywords and category labels corresponding to packaging-related objects. Categories that do not directly contribute to packaging analysis, such as animals, natural scenes, or unrelated objects, were excluded.

4.2. Test results

1. Performance comparison test

   The performance test uses the COCO dataset, comparing MSAFNet with YOLOv5s, Faster R-CNN, and EfficientDet-D0 under a standard evaluation protocol. Training uses COCO train2017 (118k images), validation uses val2017 (5k images), and metrics include mAP@0.5.

   At the per-class level, metrics such as precision, recall, and mAP are evaluated for each packaging category to identify class-wise performance variations, particularly for small and less frequent object classes.

   mAP@0.5:0.95, and F1 score. Models are trained for 300 epochs with batch size 16 using the Adam optimizer (learning rate 0.001). The test results are shown in Table 2. The test results are shown in Table 2.

   A statistical significance analysis using a paired t-test was conducted on MSAFNet's mAP@0.5 versus YOLOv5s, Faster R-CNN, and EfficientDet-D0. Results show p-values below 0.05, confirming that MSAFNet's performance gains are statistically significant and reliable.

   Table 3 presents the statistical significance analysis comparing MSAFNet with YOLOv5s, Faster R-CNN, and EfficientDet-D0 using a paired t-test.

2. Robustness test

   The robustness test evaluates model performance on the ImageNet dataset under degraded conditions by adding Gaussian noise, motion blur, and random occlusion. The mAP@0.5 decrease rate (difference between original and degraded performance) is calculated, with results shown in Table 4.

Table 1. Evaluation Metrics Used for Model Performance Assessment.

Metric	Description
mAP@0.5	Mean Average Precision at IoU threshold 0.5
mAP @0.5:0.95	Mean Average Precision averaged across IoU thresholds 0.5 to 0.95
F1-score	Harmonic mean of precision and recall
FPS	Frames per second (inference speed)
Parameters	Number of Learnable Parameters in millions (M)
FLOPs	Computational complexity in GFLOPs

Table 2. Performance comparison test results (on COCO dataset).

Models	mAP@0.5 (%)	mAP @ 0.5: 0.95 (%)	F1 score	Inference Speed (FPS)
Faster R-CNN	78.5	56.3	0.81	12
YOLOv5s	85.2	62.1	0.87	45
EfficientDet-D0	82.7	59.8	0.84	38
MSAFNet (This article)	89.6	65.9	0.91	40

Table 3. Statistical significance test results.

Model Comparison	Test Method	p-value	Significance
MSAFNet vs YOLOv5s	Paired t-test	< 0.05	Significant
MSAFNet vs Faster R-CNN	Paired t-test	< 0.05	Significant
MSAFNet vs EfficientDet-D0	Paired t-test	< 0.05	Significant

Table 4. Robustness test results (mAP @ 0.5% decrease rate).

Models	Gaussian Noise Condition	Motion Blur Condition	Occlusion Condition	Average Drop Rate
Faster R-CNN	15.3	18.7	22.1	18.7
YOLOv5s	10.5	13.2	16.8	13.5
EfficientDet-D0	12.1	14.9	18.3	15.1
MSAFNet (This article)	8.2	10.5	13.7	10.8

Table 5. Practicality test results (Resource consumption and efficiency).

Models	Parameter Quantity (M)	FLOPs (G)	FPS
Faster R-CNN	137.2	240.5	12
YOLOv5s	7.5	16.5	45
EfficientDet-D0	5.3	12.8	38
MSAFNet (This article)	8.9	18.2	40

Table 6. Ablation test results of different modules.

Model Variant	Precision	Recall	F1-score	mAP@0.5
w/o DAM	0.87	0.86	0.86	85.1
Channel Attention Only	0.89	0.87	0.88	86.7
Spatial Attention Only	0.88	0.88	0.88	86.2
MSAFNet (Complete DAM)	0.92	0.90	0.91	89.6

3. Practicability Test

   Indicators include the number of parameters (Params), the number of calculations (FLOPs), and the number of frames per second (FPS). The test uses a batch size of 1 and an input resolution of 640×640. The results are shown in Table 5.

4. Ablation test

   The ablation test evaluates the contribution of each key component of MSAFNet on the COCO dataset is given in Table 6.

5. Interpretability test

   The test uses 100 packaging images from the Open Images dataset, and the results are shown in Table 7.

   Grad-CAM heatmaps are generated to visualize the regions that contribute most to the model's classification decisions.

6. Generalization experiments across datasets

   Cross-dataset generalization tests validate the generalization ability of MSAFNet on unseen packaging image datasets. The

Table 7. Interpretability test results (mean expert scores).

Models	Attention Area Accuracy	Degree of Coincidence with Real Box	Comprehensive Score
MSAFNet (This article)	4.5	4.3	4.4
YOLOv5s	3.8	3.5	3.7
EfficientDet-D0	4	3.8	3.9

Table 8. Generalization test results across datasets (mAP @ 0.5%).

Models	COCO val2017	Open Images v6	PackNet	Google Open Images	Average Drop Rate
YOLOv5s	85.2	78.3	76.5	79.1	8.90%
EfficientDetD0	82.7	75.8	74.2	76.9	9.80%
MSAFNet (This study)	89.6	83.7	81.9	84.5	6.10%

Table 9. Real-time test results (reasoning efficiency and energy consumption).

Models	Hardware Platform	FPS	Delay (ms)	Energy Consumption (Wh)
YOLOv5s	RTX 3080	45	22.2	1.2
	Jetson Xavier NX	28	35.7	0.8
	Intel i7 CPU	5	200	3.5
EfficientDet-D0	RTX 3080	38	26.3	1.1
	Jetson Xavier NX	22	45.5	0.7
	Intel i7 CPU	4	250	3.8
MSAFNet (This study)	RTX 3080	40	25.0	1.0
	Jetson Xavier NX	25	40.0	0.6
	Intel i7 CPU	4.5	222.2	3.2

Table 10. Quantitative Interpretability Evaluation Results.

Model	IoU (%)	Pointing Accuracy (%)	Attention Precision	Attention Recall
YOLOv5s	52.3	68.5	0.64	0.59
EfficientDetD0	55.1	71.2	0.67	0.62
MSAFNet	61.8	78.6	0.74	0.69

results are shown in Table 8.

7. Real-time test

   Real-time tests evaluate the inference efficiency of the model on different hardware platforms, including high-end GPUs, edge device GPUs and CPUs. The results are shown in Table 9.

The results indicate that MSAFNet achieves superior alignment between attention regions and target objects, demonstrating improved interpretability compared to baseline models is given in Table 10.

4.3. Analysis and discussion

Motion blur simulates camera movement or conveyor belt speed in industrial environments, leading to loss of edge sharpness and boundary information Eq. (1) Analysis of experimental results The performance comparison (Table 2) shows MSAFNet achieves an mAP@0.5 of 89.6% on COCO, improving 4.4% over YOLOv5s.

The robustness test (Table 4) shows that the average performance degradation rate of MSAFNet under Gaussian noise, motion blur, and occlusion conditions is only 10.8%, significantly lower than that of the baseline model.

In the practicality test (Table 5), MSAFNet achieved an inference speed of 40 FPS with 8.9M parameters and 18.2G FLOPs, achieving the best balance between accuracy and efficiency.

The interpretability experiment (Table 7) demonstrates through Grad-CAM visualization that MSAFNet achieves an attention region accuracy score of 4.5, which is higher than that of the baseline model.

In the cross-dataset generalization experiment (Table 8), MSAFNet achieved an average decrease rate of only 6.1% on the external dataset, outperforming the baseline models with a decrease rate ranging from 8.9% to 9.8%. The real-time performance test (Table 9) shows that MSAFNet consumes only 0.6 Wh of energy on the Jetson Xavier NX edge device and has a latency of 222.2 ms on the CPU, balancing efficiency and resource consumption.

5. Conclusion

This study proposes MSAFNet, a CNN-based model for automatic packaging image recognition, integrating a lightweight backbone, Dual Attention Module (DAM), and Adaptive Feature Pyramid Network (AFPN) for enhanced feature representation and multi-scale detection. Experiments on public datasets show that MSAFNet outperforms baseline models in accuracy, robustness, and generalization. Future research will focus on introducing self-supervised learning strategies to reduce data dependence, designing dynamic inference mechanisms to further improve com-

putational efficiency, and enhancing cross-domain generalization ability to support more complex real-world packaging scenarios.

References

[1] S. Y. Chaganti, I. Nanda, K. R. Pandi, T. G. Prudhvith, and N. Kumar, (2020) *"Image classification using SVM and CNN"* **Proceedings of the 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)**: 1–5. DOI: 10.1109/ICCSEA49143.2020.9132851.

[2] Z. Liu, L. Sun, and Q. Zhang, (2022) *"High similarity image recognition and classification algorithm based on convolutional neural network"* **Computational Intelligence and Neuroscience *2022***: 2836486. DOI: 10.1155/2022/2836486.

[3] K. V. Greeshma and J. V. Gripsy, (2020) *"Image classification using HOG and LBP feature descriptors with SVM and CNN"* **International Journal of Engineering Research and Technology** *8*(4): 1–4. URL: https://d1wqtxts1xzle7.cloudfront.net/110347554.

[4] M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, (2020) *"Improved inception-residual convolutional neural network for object recognition"* **Neural Computing and Applications** *32*(1): 279–293. DOI: 10.1007/s00521-018-3627-6.

[5] T. Hassanzadeh, D. Essam, and R. Sarker, (2022) *"EvoDCNN: An evolutionary deep convolutional neural network for image classification"* **Neurocomputing** *488*: 271–283. DOI: 10.1016/j.neucom.2022.02.003.

[6] M. Tripathi, (2021) *"Analysis of convolutional neural network-based image classification techniques"* **Journal of Innovative Image Processing** *3*(2): 100–117. DOI: 10.36548/jiip.2021.2.003.

[7] M. A. Hasan, T. Bhargav, V. Sandeep, V. S. Reddy, and R. Ajay, (2024) *"Image classification using convolutional neural networks"* **International Journal of Mechanical Engineering Research and Technology** *16*(2): 173–181. DOI: 10.1109/ICEEICT53079.2022.9768622.

[8] M. Yang, P. Kumar, J. Bhola, and M. Shabaz, (2022) *"Development of image recognition software based on artificial intelligence algorithm for the efficient sorting of apple fruit"* **International Journal of System Assurance Engineering and Management** *13*(Suppl. 1): 322–330. DOI: 10.1007/s13198-021-01415-1.

[9] Z. Ren, F. Fang, N. Yan, and Y. Wu, (2022) *"State of the art in defect detection based on machine vision"* **International Journal of Precision Engineering and Manufacturing–Green Technology** *9*(2): 661–691. DOI: 10.1007/s40684-021-00343-6.

[10] Y. Liu, H. Pu, and D. W. Sun, (2021) *"Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices"* **Trends in Food Science and Technology** *113*: 193–204. DOI: 10.1016/j.tifs.2021.04.042.

[11] Q. Zhang, Q. Yang, X. Zhang, Q. Bao, J. Su, and X. Liu, (2021) *"Waste image classification based on transfer learning and convolutional neural network"* **Waste Management** *135*: 150–157. DOI: 10.1016/j.wasman.2021.08.038.

[12] K. U. S. W. O. R. O. Adi, C. E. Widodo, A. P. Widodo, and U. S. Margiati, (2022) *"Detection of foreign object debris (FOD) using convolutional neural network (CNN)"* **Journal of Theoretical and Applied Information Technology** *100*(1): 184–191. URL: https://www.jatit.org/volumes/Vol100No1/16Vol100No1.pdf.

[13] S. Masood, U. Ahsan, F. Munawwar, D. R. Rizvi, and M. Ahmed, (2020) *"Scene recognition from image using convolutional neural network"* **Procedia Computer Science** *167*: 1005–1012. DOI: 10.1016/j.procs.2020.03.400.

[14] Y. Li, R. G. Zhou, R. Xu, J. Luo, and W. Hu, (2020) *"A quantum deep convolutional neural network for image recognition"* **Quantum Science and Technology** *5*(4): 044003. DOI: 10.1088/2058-9565/ab9f93.

[15] G. Chen, Q. Chen, S. Long, W. Zhu, Z. Yuan, and Y. Wu, (2023) *"Quantum convolutional neural network for image classification"* **Pattern Analysis and Applications** *26*(2): 655–667. DOI: 10.1007/s10044-022-01113-z.

[16] R. Basha, P. Pathak, M. Sudha, K. V. Soumya, and J. Arockia Venice, (2025) *"Optimization of quantum dilated convolutional neural networks: Image recognition with quantum computing"* **Internet Technology Letters** *8*(3): e70027. DOI: 10.1002/itl2.70027.

[17] H. C. V. Ngu and K. M. Lee, (2022) *"Effective conversion of a convolutional neural network into a spiking neural network for image recognition tasks"* **Applied Sciences** *12*(11): 5749. DOI: 10.3390/app12115749.

[18] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, (2020) *"3D packing for self-supervised monocular depth estimation"* **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**: 2485–2494. URL: https://openaccess.thecvf.com/content_CVPR_2020/html/Guizilini_3D_Packing_for_Self-Supervised_Monocular_Depth_Estimation_CVPR_2020_paper.html.

[19] I. J. Jacob and P. E. Darney, (2021) *"Design of deep learning algorithm for IoT application by image-based recognition"* **Journal of ISMAC *3***(3): 276–290. DOI: 10.36548/jismac.2021.3.008.

[20] C. Zhang, M. Han, J. Jia, and C. Kim, (2024) *"Packaging design image segmentation based on improved full convolutional networks"* **Applied Sciences *14***(22): 10742. DOI: 10.3390/app142210742.

[21] S. Aburass, A. Huneiti, and M. B. Al-Zoubi, (2022) *"Classification of transformed and geometrically distorted images using convolutional neural network"* **Journal of Computer Science *18***(8): 757–769. DOI: 10.3844/jcssp.2022.757.769.

[22] D. Dai, Y. Li, Y. Wang, H. Bao, and G. Wang, (2022) *"Rethinking the image feature biases exhibited by deep convolutional neural network models in image recognition"* **CAAI Transactions on Intelligence Technology** *7*(4): 721–731. DOI: 10.1049/cit2.12097.

[23] M. Nauta, R. Van Bree, and C. Seifert, (2021) *"Neural prototype trees for interpretable fine-grained image recognition"* **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**: 14933–14943. URL: https://openaccess.thecvf.com/content/CVPR2021/html/Nauta_Neural_Prototype_Trees_for_Interpretable_Fine-Grained_Image_Recognition_CVPR_2021_paper.html.

[24] Q. Cao, (2021) *"The art of packaging: an investigation on modern packaging design and artistic thinking under the background of big data"* **Journal of Applied Science and Engineering *24***(4): 807–812. DOI: 10.6180/jase.202110_24(5).0017.