Salient Object Detection Based on Improved PoolNet

X. X. Yuan, Y. Xu

Abstract—Salient object detection (SOD) aids in accurately detecting and segmenting the most distinctive objects from visual scenes. However, existing SOD algorithms exhibit specific problems such as redundant model parameters and slow reasoning speed. To address these, we propose an efficient lightweight SOD algorithm. The number of parameters in the model is reduced using the PoolNet algorithm, wherein the original residual neural network structure is replaced by the DenseNet structure. Additionally, the PoolNet feature pyramid network (FPN) is improved into a weighted bi-directional FPN to enhance the model accuracy. The problem of the unclear boundary of salient objects is solved by fusing binary cross-entropy, structural similarity, and intersection ratio loss functions to replace the traditional cross-entropy loss function. The obtained results indicate that the accuracy of the improved PoolNet algorithm is 1% higher than that of the original algorithm. The number of model parameters is reduced by 80%, and the clarity of the salient object boundaries is enhanced.

***Index Terms*—Computer vision, Edge detection, Object detection, PoolNet, Salient object detection**

I. Introduction

Salient object detection (SOD) refers to the detection of the most prominent objects in a visual scene. The recent increase in the studies on SOD has resulted in its extensive utilization in various applications such as image classification [1], semantic segmentation[2], object detection, automated driving, visual tracking, and medical image processing.

Typically, SOD methods can be divided into traditional and deep learning-based methods. Traditional SOD methods primarily use human intuitive sense or heuristics, such as chromaticity comparison, background comparison, and prior boundary points, to detect the target through manual feature extraction; however, manual feature extraction is time-consuming. Further study of SOD, increasing the development of deep-learning-based SOD methods [3]. Unlike traditional methods, SOD based on deep learning does not require the manual extraction of features; it automatically obtains multi-scale features with significant improvement in the detection performance. For instance, the detection accuracy and timeliness are improved substantially. Moreover, the model is lightweight, and a single network can simultaneously detect salient objects and localize boundaries. However, the existing deep-learning-based SOD methods exhibit specific shortcomings. The SOD performance requires further improvement under complex backgrounds [4]. The real-time detection should be enhanced, and the model complexity must be reduced.

By 2019, PoolNet [5] was identified as the optimal model in terms of quantitative analysis and visualization in SOD algorithms. In comparison with the SOD algorithms developed during the same period, PoolNet exhibited clear advantages with respect to speed and accuracy. However, in comparison with the SOD algorithms that have emerged since 2020, PoolNet exhibits specific disadvantages with respect to edge detection and accuracy of salient graphs. The main drawback is the insufficient discrimination between the saliency graph and complex background, which leads to fuzzy edges in the detected saliency graph. However, the PoolNet algorithm is not inferior to the existing mainstream algorithm in terms of model structure, design, and operating speed; therefore, improving the PoolNet algorithm is highly meaningful. Based on the U-shaped structure, PoolNet initially constructs the global guidance module (GGM) in the bottom-up path to provide potentially salient object location information for layers at different feature levels. Additionally, the feature aggregation modules (FAMs) are added after the top-down path fusion operation seamlessly integrate the coarse features of GGM with features of different scales. Consequently, the extraction of advanced semantic information is deepened, and the location of salient objects can be determined in a better manner. However, the PoolNet algorithm can be significantly improved in terms of the extraction of low-level features. This is because the low-level features in the multi-layer and multi-scale features extracted by deep networks, such as visual geometry group (VGG) network (VGGNet) and residual neural network (ResNet), contain adequate spatial information and can be positioned appropriately. Therefore, the detection ability of salient object boundaries should be improved.

In this study, we propose an improved PoolNet SOD algorithm using DenseNet [6], which has a narrow network and requires only a few parameters for feature extraction. The original backbone network of the PoolNet algorithm, namely ResNet, was replaced by DenseNet; this significantly

Manuscript received February 20, 2022; revised November 10, 2022. This work was supported by the Liaoning Education Department Project (LJKZ0310), 2021 Outstanding Talents Project of University of Science and Technology Liaoning.

X. X. Yuan is a postgraduate student at the School of Computer Science and Software Engineering, University of Science and Technology LiaoNing, Anshan 114051, China (e-mail: yxx10262022@163.com).

Y. Xu is a Professor at the School of Computer Science and Software Engineering, University of Science and Technology LiaoNing, Anshan 114051, China (corresponding author, e-mail: xuyang_1981@aliyun.com).

reduced parameters and operating cost of the algorithm. To improve the accuracy of model calculation, we used a novel bidirectional weighted feature pyramid network (BiFPN) [7] to replace the original traditional feature pyramid network (FPN), which further strengthened the capability of feature fusion. Finally, the PoolNet edge detection method was improved, and the original traditional cross-entropy loss function of the PoolNet algorithm was adjusted to enhance the edge definition of salient objects.

II. Principle of the PoolNet Algorithm

The PoolNet algorithm designed two new modules, based on the U-shaped structure. The GGM in the bottom-up path provides potentially salient object location information for different feature layers, whereas the FAMs added after the top-down path fusion operation ensure the seamless fusion of coarse features of GGM with the features of different scales. Fig. 1 depicts the basic structure of the PoolNet algorithm.

The GGM module of the PoolNet algorithm is composed of two parts, the improved pyramidalization module (PPM) and the global guided flow (GGF). The model places PPM at the top of the entire network to extract global startup information, and then introduces GGF into the network, thus transmitting the semantic feature information collected by PPM directly to all pyramid level feature maps. In general, during the network down-sampling process, low-level semantic information is diluted, the receptive field will be small; therefore, global information is not captured. However, the design of GGM effectively neutralizes the gradual dilution of U-shaped network signals from top to bottom.

As indicated in Fig. 2, the module incorporates four different pyramid features. The first line represents the roughest feature, generating a single output feature under global pooling, whereas the subsequent three lines denote the pooling features at different scales.

Aiming at the fusion problem of GGF corresponding rough feature mapping and pyramid feature mapping of different scales, PoolNet algorithm designed a feature aggregation module (FAM) and took the fused features as input information. FAM converts input information into multiple dimensions to obtain rich local context information, which is then fused to better calculate the composition of input features after fusion.

Although the PoolNet algorithm has advantages in network structure and detection accuracy, it is still insufficient for the detection of significant object edges. Moreover, the parameters in the PoolNet training method is too much, and the inference speed of the algorithm has room for improvement. In this study, PoolNet was comprehensively optimized to significantly reduce the parameters of the model and improve the detection effect of the model, while enhancing the edge clarity of the detection results.

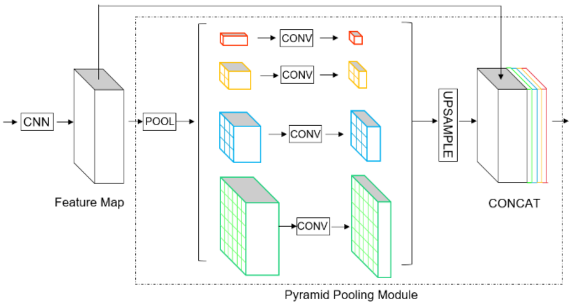


Fig. 2. Global guidance module.

III. Improved Strategy

Based on the SOD indicators and PoolNet algorithm, we performed three aspects of optimization, namely the optimization of the backbone network, enhancement of feature extraction, and adjustment of the loss function. The first is the optimization of the backbone network. The primary function of the backbone network was to extract the salient features of the input image and output them. Typically, the backbone network adopted by the PoolNet model is ResNet, which is deep and densely connected and can excellently extract deep features. However, the numerous

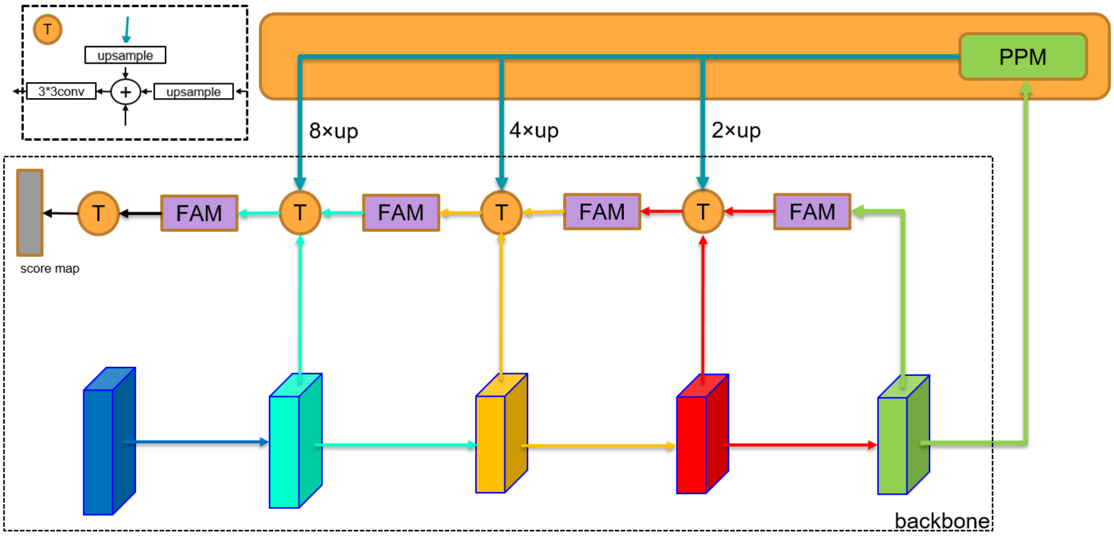


Fig. 1. Schematic of the basic structure of the PoolNet algorithm.

network parameters lead to the overstaffing of the model. Therefore, DenseNet with a narrow network and few parameters was used as the backbone network to reduce the model parameters. The second part was the enhancement of feature extraction. A typical PoolNet algorithm uses a network model based on an FPN. To further enhance the fusion of feature information of different levels, we used a weighted BiFPN. The third part was the adjustment of the loss function, which improved the traditional standard loss function of the original model is improved into a hybrid loss function which is weighted by multiple losses. This adjustment effectively improved the boundary definition of signed graphs.

A. *Optimization of Backbone Networks*

The hierarchy of the network crucially affects the model's effect of detection. When the number of network layers is increased, more complex feature patterns can be extracted from the network; this theoretically implies that the deeper the model, the better the results. With the increase in network depth, network accuracy may saturate or even decline. Therefore, He *et al*. proposed a ResNet structure based on the VGGNet and added residual units through a short-circuit mechanism to form residual learning and solve the degradation of network depth. However, the DenseNet structure has a denser and more radical connection mechanism than ResNet. It connects each layer in the network to ensure that each layer accepts all the subsequent layers as its additional input. Figs. 3 and 4 depict the connection mechanisms of the ResNet and DenseNet, respectively.

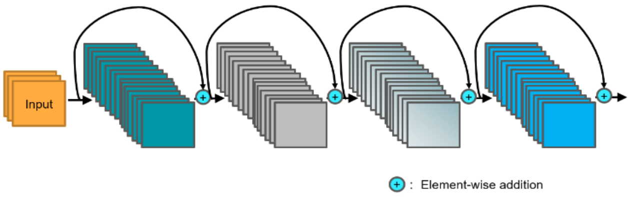


Fig. 3. Short-circuit connection mechanism of ResNet (+ represents element-level addition operation).

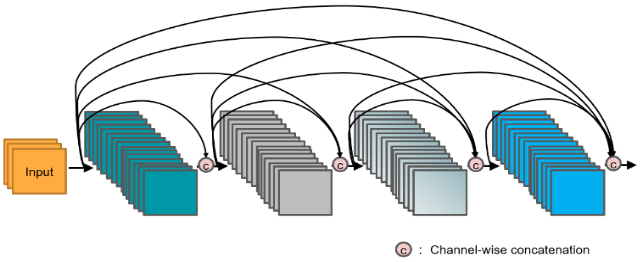


Fig. 4. Connection mode of the DenseNet (C denotes Cascade of channels).

The connection mode can be expressed mathematically as follows. The output of the traditional network in the L layer is $x_l = H_l(x_{l-1})$. In the case of ResNet, the identity function input from the upper layer is added as follows: $x_l = H_l(x_{l-1}) + x_{l-1}$. In DenseNet, the data of all preceding layers are connected as input data as $x_l = H_l(x_0, x_1, ..., x_{l-1})$. Here, $H_l(.)$ represents the nonlinear transformation function. Fig. 5 illustrates the forward process of DenseNet, and its connection mode can be intuitively understood. The input data of H3 include X2 from H2 and X1 and X0 from the initial two layers.

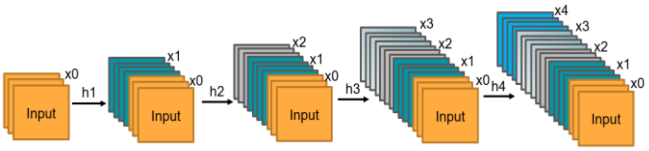


Fig. 5. Forward process of DenseNet.

The proposed DenseNet has a dense block design, wherein the output feature of each convolutional layer is less, rather than hundreds or thousands of widths, as observed in other networks. Moreover, this connection transmits the features and gradients more efficiently compared to other connection mechanisms, and the network is easier to train.

B. *Enhancement of Multi-scale Feature Fusion*

The PoolNet algorithm is based on the feature fusion pyramid FPN. The function of the FPN pyramid network is to solve the multi-scale problem of detecting objects, redesign the link of the network, and optimize the detection effect of small targets while maintaining the computational amount of the original model. Upper-level features are sampled, lower-level features are joined in a top-down manner, and predictions are made at each level. The structure is shown in Fig. 6.

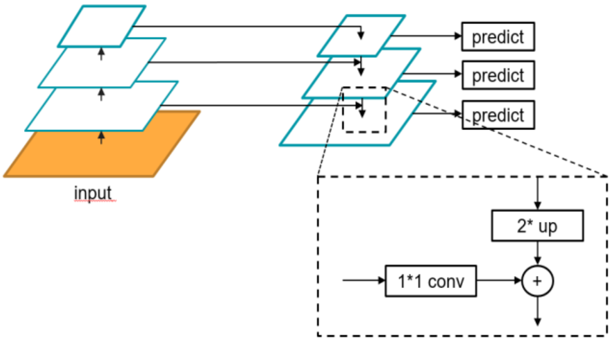
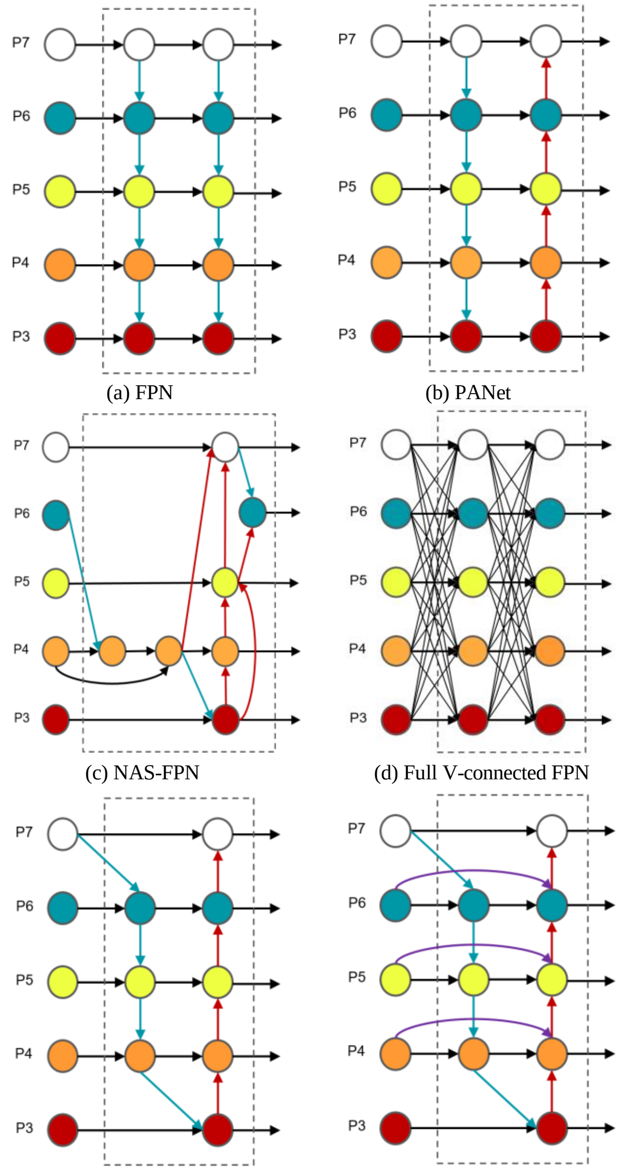


Fig. 6. Basic structure diagram of FPN.

In the figure, the region where the transmission is connected to the top-down process is enlarged. It can be seen that the 1*1 convolution operation primarily reduces the number of convolution cores, that is, reduces the feature graphs without changing the size of feature graphs.

However, FPN exhibits several problems, such as a semantic gap between different layers before fusion and a reduction in the representation ability of multi-scale features because of direct fusion. Typically, the feature information of the highest pyramid is lost during down-sampling. Therefore, several improved methods have been proposed as alternatives for FPN in recent years. In this study, the original FPN was replaced with a weighted BiFPN, which enhanced the capability of feature representation by adding residual links and removing the single side node. As the single node input edge do not carry out feature fusion, they contain less semantic feature and do not significantly impact the final feature fusion. The computation load on the model is reduced after the removal of these nodes. Finally, a weight is added for each scale feature to adjust their contribution degree. Fig. 7 illustrates a structural comparison of BiFPN, FPN, and other improved methods.



(a) FPN (b) PANet

(c) NAS-FPN (d) Full V-connected FPN

(e) Simplified PANet (f) BiFPN

Fig. 7. Structural comparison of BiFPN, FPN, and other improved methods.

C. Optimization of the Loss Function

At present, most SOD algorithms use the traditional loss function; they cannot adequately distinguish the boundary pixels during the training process, which results in the boundary blur phenomenon. As the PoolNet model also uses a similar method, we modified the original traditional loss function into a mixed loss function. This included the BCE, SSIM, and IOU loss functions to obtain the saliency graph with high confidence and a clear boundary.

BCE loss is widely used in binary classification and segmentation, and can be expressed as

$$Loss = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(p(y_i)) + (1-y_i)\log(1-p(y_i)) \tag{1}$$

where *y* denotes a binary label with a value of 0 or 1, and *P(y)* indicates the predicted probability of significance. It can be seen from the above formula that the loss is related to every pixel; hence, this loss function can focus on the saliency feature at the pixel level of the image.

SSIM was first proposed for image quality assessment. As the pixels of the image are strongly correlated, they contain important information regarding the structure of the object in a visual scene. SSIM considers the brightness and contrast associated with object structure as the definition of structural information in the image to effectively capture the structural information in the input image.

SSIM measurement system comprises three contrast modules, namely brightness, contrast, and structure.

The brightness contrast function calculates the average gray level of the image as an estimation of brightness measurement; it can be expressed as

$$\mu_X = \frac{1}{H \times M}\sum_{i=1}^{H}\sum_{j=1}^{M} X(i,j) \tag{2}$$

Therefore, the brightness contrast function of the two images can be obtained as:

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{3}$$

The contrast function calculates the standard deviation of the image using the estimation formula of the contrast measurement.

$$\sigma_X = \left(\frac{1}{H+W-1}\sum_{i=1}^{H}\sum_{j=1}^{M}(X(i,j)-\mu_X)^2\right)^{\frac{1}{2}} \tag{4}$$

Thus, the contrast function of the two images can be determined as

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{5}$$

The structural comparison function can be represented as

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{6}$$

Finally, the SSIM function formula is obtained by integrating the aforementioned three comparison functions:

$$\begin{aligned} SSIM(x,y) &= f\left(l(x,y), c(x,y), s(x,y)\right) \\ &= \left[l(x,y)\right]^{\alpha}\left[c(x,y)\right]^{\beta}\left[s(x,y)\right]^{\gamma} \end{aligned} \tag{7}$$

where α, β, and γ > 0 are used to integrate these three modules. If α, β, and γ are all 1, then $C_3 = \frac{C_2}{2}$.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{8}$$

By analyzing this loss function, it can be found that the loss generated by each pixel is related to its nearby local patch. Therefore, in the process of training, the loss value of the edges of the object will be strengthened, and the non-edges will be suppressed. Because of the existence of this loss, the algorithm can pay attention to more edge details of target saliency.

IOU was initially used to measure the similarity of two sets; subsequently, it is used for image processing orientation.

$$IoULoss = 1 - IoU = 1 - \frac{(A \cap B)}{A \cup B} \tag{9}$$

IV. Analysis of Experiments And Results

A. Dataset Selection

To satisfy the different research requirements of SOD, several SOD datasets have been proposed; the DUTS dataset [8] was used in our experiment. DUTS-TR and DUTS-TE of the dataset were considered as the training and test sets, respectively. The DUTS dataset consists of two parts; one has 10,553 images for training, and the other has 5,019 images used for testing. PASCAL-S is composed of 850 images with complex backgrounds [9].

B. Experimental Evaluation Criteria

We used Pytorch1.8 to implement the improved model and used the GTX 3070 GPU. Adam optimizer was used in the experiment, and the learning rate was 5e-5. The batch size was set to 8 and epoch to 32. We compared the PoolNet algorithm before and after improvement; the improved algorithm is referred to as DBLPoolNet. The proposed approach was evaluated using three widely-used metrics.

F-measure F_β [10] is used to comprehensively evaluate both precision and recall as follows:

$$F_\beta = \frac{(1+\beta^2) \times \mathrm{Pr}ecision \times \mathrm{Re}call}{\beta^2 \times \mathrm{Pr}ecision + \mathrm{Re}call} \tag{10}$$

MAE score [11] indicates the similarity between the saliency graph *S* and true value *G*; it can be expressed as follows:

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} \left| S(x,y) - G(x,y) \right| \tag{11}$$

where *W* and *H* represent the width and height of *S*, respectively.

The **weighted F-measure** (F_β^ω) is utilized as a complementary measure to *max* F_β for reducing the possible unfair comparison caused by interpolation flaw, dependency flaw and equal-importance flaw.

$$F_\beta^\omega = (1+\beta^2) \frac{Precision^\omega \cdot Recall^\omega}{\beta^2 \cdot Precision^\omega + Recall^\omega} \tag{12}$$

C. Experimental Analysis

Initially, we analyzed the effectiveness of the weighted BiFPN and adjusted the loss function after optimizing the network structure. Then, we compared the experimental results considering the optimized number of parameters and network model size with those reported for existing models. Finally, the experimental results and saliency graphs were compared with those of other models.

To validate the effectiveness of the proposed optimization scheme, we split the optimization process and experimentally tested each optimized scheme. Fig. 8 depicts the comparison of the obtained results.

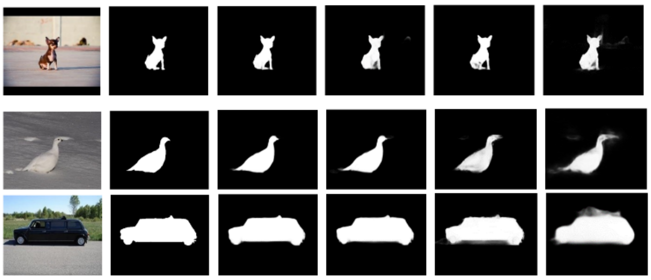

(a) Image (b) GT (c) Ours (d) D + B (e) D +B (f) D

Fig. 8. Comparison of the obtained experimental results.

Figs. 8(a), 8(b), and 8(c) depict the original image, the real value of the image, and the final result obtained after optimizing the model, respectively. Fig. 8(d) indicates the result obtained using BiFPN after replacing the network. Fig. 8(e) depicts the result obtained by adjusting the loss function after replacing the network. Fig. 8(f) illustrates the resulting graph obtained by replacing the network. As indicated in Fig. 8(f), the salient graph obtained after replacing ResNet was not adequate because the DenseNet structure did not fit well in the original model. However, Figs. 8(c), 8(d), and 8(e) confirm that the proposed optimization technique effectively addresses this problem. Table I compares the number of model parameters; here, we replaced the network of the original model with the lightweight network MobileNetv3 for simplifying the comparison.

TABLE I
Comparison of Network Parameters

Model	Parameters
PoolNet-ResNet	68,261,057
PoolNet-MobileNetv3	19,395,637
DBLPoolNet (Ours)	13,732,219

As indicated in the table, the proposed improved model has the least number of parameters compared to the original and MobileNetv3 models. Additionally, although replacing the network with MobileNetv3 considerably reduced the parameters, the experiments indicated that its detection effect was also significantly reduced (Fig. 9).

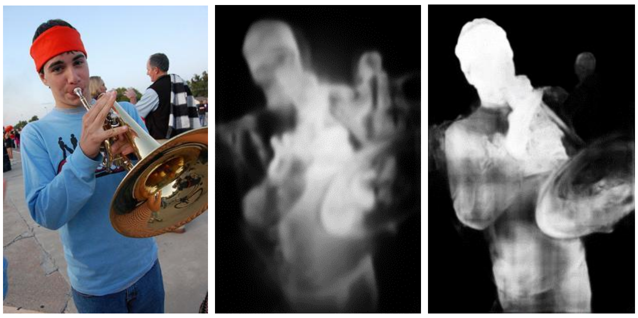

(a) Original map (b) MobileNetv3 result map (c) DenseNet result map

Fig. 9. Comparison of detection effect.

Table II summarizes the comparison of error rates between ResNet and DenseNet considering CIFAR-10 and CIFAR-100 datasets ("+" represents an enhanced data set). The table indicates that DenseNet achieved lower error rates with fewer parameters when compared to ResNet. Even without data augmentation, DenseNet performed better by a significant margin; it performed significantly better than ResNet after using data enhancement.

TABLE II
ERROR RATE COMPARISON OF RESNET AND DENSENET

Model	Depth	Parameters	CI10	CI10+	CI100	CI100+
ResNet	110	1.7M	13.63	6.41	44.74	27.22
DenseNet	100	0.8M	5.92	4.51	24.15	22.27

Table III presents the comparison of the size of the improved model with that of the original and other models that use ResNet. The size of the optimized model was significantly reduced to only 53M. This enabled its embedding in small devices with reduced memory usage.

TABLE III
COMPARISON OF NETWORK MODEL SIZES

Model	Parameters
DGRL	646M
PiCANet	197M
CPD	183M
BASNet	348.5M
U2-Net	176.3M
PoolNet	260M
DBLPoolNet (Ours)	53M

The data in Table III verifies that the size of the optimized DBLPoolNet model is reduced by approximately 80% in comparison with the PoolNet model. Moreover, the accuracy of the proposed model is also improved in comparison with that of the original one.

Quantitative Analysis

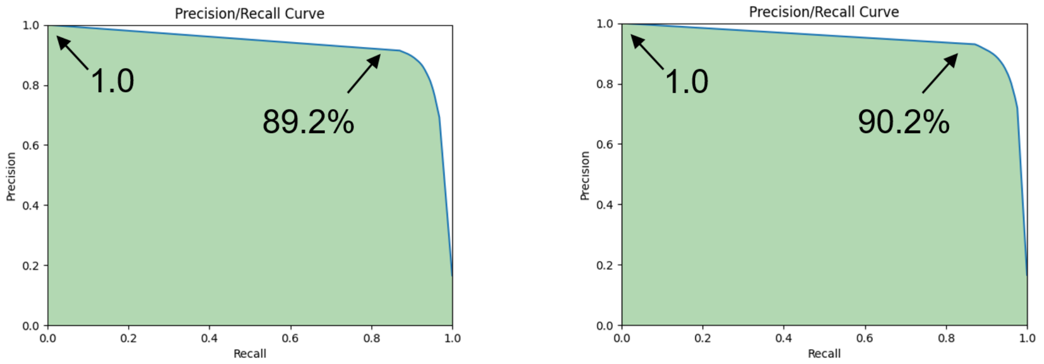
To evaluate the quality of detecting salient objects, we analyzed the PR and F-measure curves of the model considering the DUTS-TE dataset. The recall rate and accuracy of the two networks were calculated, and the P-R curve was established with the horizontal axis as the recall ratio and the vertical axis as accuracy. The size of the area under the curve represents the effectiveness of salient object detection. Recall rate *R* and accuracy rate *P* can be expressed as

$$R = \frac{X_{TP}}{X_{TP} + X_{FN}} \tag{13}$$

$$P = \frac{X_{TP}}{X_{TP} + X_{FP}} \tag{14}$$

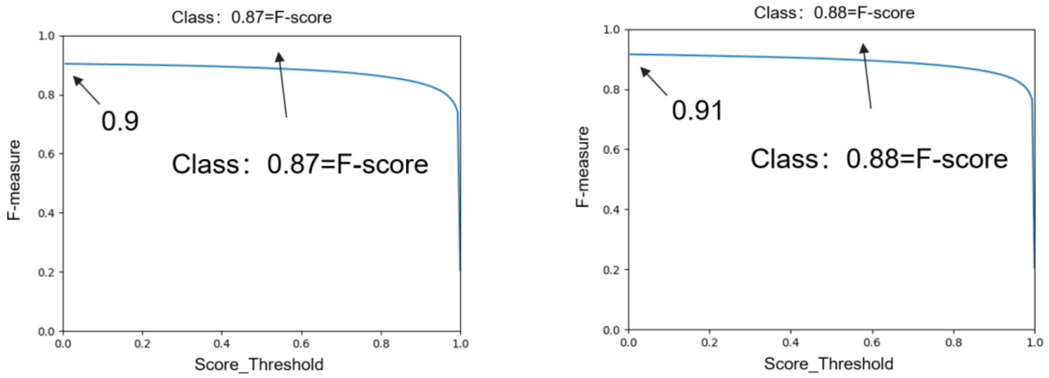
where X_{TP} represents the correctly detected part, X_{FN} denotes the part that is not detected, and X_{FP} indicates the part that detects errors. Both the PoolNet algorithm before improvement and the improved algorithm were tested on the DUTS-TE dataset; Fig. 10 depicts the P–R curve of the obtained results.

The F-measure score indicates the precision and recall weighted harmonic mean, and is calculated as



(a) PoolNet (b) DBLPoolNet

Fig. 10. Comparison of precision–recall (P–R) curves before and after model improvement.



(a) PoolNet (b) DBLPoolNet

Fig. 11. Comparison of F-Measure curves before and after model improvement.

TABLE IV
COMPARISON OF PERFORMANCE

	Training		PASCAL-S			DUTS-TE		
Model	Image	Dataset	MaxF	MAE	F_β^ω	Max F	MAE	F_β^ω
SRM[12]	10,553	DUTS	0.838	0.084	0.758	0.826	0.058	0.722
DGRL	10,553	DUTS	0.844	0.072	0.787	0.828	0.049	0.760
PiCANet[13]	10,553	DUTS	0.864	0.075	0.777	0.863	0.050	0.755
CPD-R[14]	10,553	DUTS	0.864	0.072	0.800	0.865	0.043	0.795
BASNet[15]	10,553	DUTS	0.854	0.076	0.798	0.860	0.047	0.803
U2-Net[16]	10,553	DUTS	0.859	0.074	0.797	0.873	0.044	0.804
PoolNet	10,553	DUTS	0.880	0.065	0.798	0.892	0.036	0.807
DBLPoolNet	10,553	DUTS	0.884	0.055	0.802	0.902	0.033	0.815

$$F-measure=\left(\frac{\mathrm{Pr}\,ecision^{-1}+\mathrm{Re}\,call^{-1}}{2}\right)^{-1} \qquad (15)$$

Combining the reasoning of (12) and (13), we concluded that

$$F-Measure=\left(\frac{\frac{X_{TP}+X_{FP}}{X_{TP}}+\frac{X_{TP}+X_{FN}}{X_{TP}}}{2}\right)^{-1}=\frac{2X_{TP}}{2X_{TP}+X_{FP}+X_{FN}} \qquad (16)$$

With the threshold as the horizontal axis, F-measure as the vertical axis, and the size of the area under the curve as the evaluation standard of the model, we obtained the salient figure F-Measure curve (Fig. 11).

The experimental indicated that the accuracy of the improved DBLPoolNet model with the DUTS-TE dataset increased from 89.2 to 90.2% in comparison with the original PoolNet model. Table IV compares the maximum F-measure, MAE, and weighted F-measure values obtained using the improved model with the results of other models.

Qualitative analysis

To clearly demonstrate the effectiveness of the improved model, Fig. 12 shows a visual comparison of experimental results with other models. As indicated in the figure, improved model has good robustness and can be applied to various scenarios. Moreover, it performed better than other algorithms with respect to the distinction between the saliency graph.

Ablation Studies

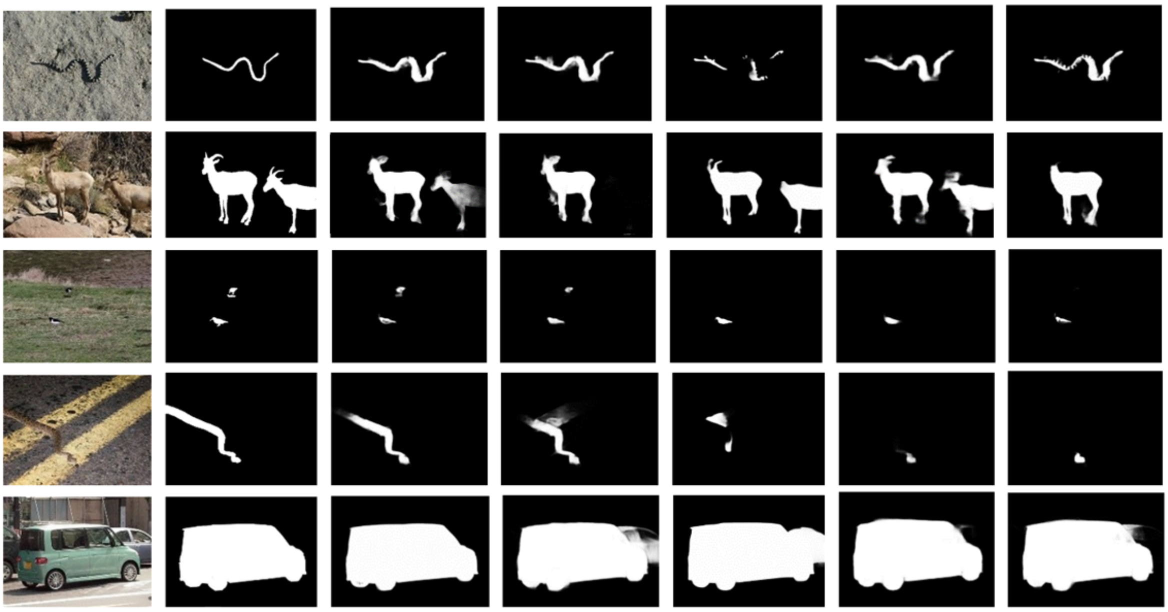
To validate the effectiveness of the components of the proposed method, we performed a series of experiments on two datasets with different settings under the DenseNet backbone. Particularly, we verified the effectiveness of BiFPN in the model and the mixing loss function.

Effectiveness of BiFPN and Mixing Loss

To demonstrate the effectiveness of using BiFPN and the mixing loss in the model, we performed ablation experiments based on the original model. Except for combinations of BiFPN and mixing loss, all other configurations were retained. Table V summarizes the performance of the model considering two datasets, namely DUTS-TE and PASCAL-S; Fig. 8 depicts the corresponding visual comparisons.

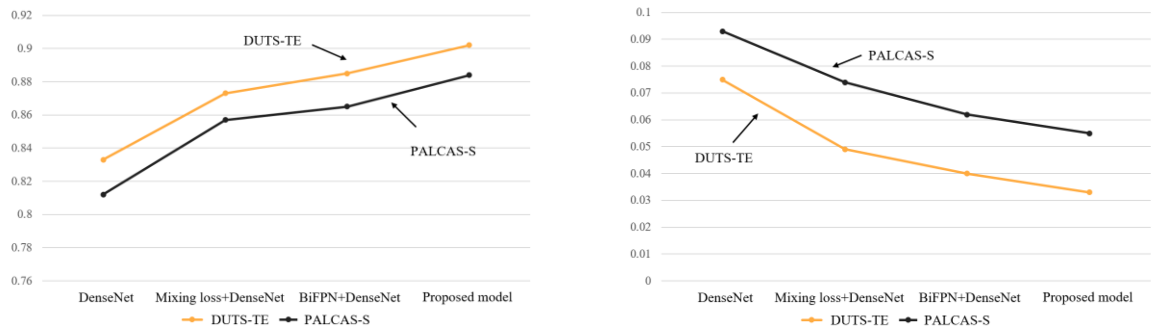
TABLE V
ABLATION STUDIES OF BIFPN AND MIXING LOSS FUNCTION

No.	Method			DUTS-TE		PASCAL-S	
	D	B	L	MaxF↑	MAE↓	MaxF↑	MAE↓
1	✓			0.833	0.075	0.812	0.093
2	✓		✓	0.873	0.049	0.857	0.074
3	✓	✓		0.885	0.040	0.865	0.062
4	✓	✓	✓	0.902	0.033	0.884	0.055



(a) Image (b) GT (c) Proposed model (d) PoolNet (e) BASNet (f) CPD (g) U2-Net

Fig. 12. Qualitative comparison between the improved model and other saliency detection models.



（a）MaxF↑ （b）MAE↓

Fig. 13. Bar chart of ablation experiment.

BiFPN

The addition of BiFPN improved the performance in terms of both F-measure and MAE in the case of both datasets. BiFPN effectively enhanced the extraction of detailed features, significantly improving the quality of the resulting saliency maps.

Mixing Loss Function

Embedding mixing loss into the original model also improved the performance with respect to both F-measure and MAE scores on the same two datasets. This was because the mixing loss function effectively determined the boundary feature information of salient objects; therefore, it could distinguish the boundary of salient objects and complex backgrounds appropriately.

To exhibit the superiority of each improvement method in the improvement process, Fig. 13 visually compares the differences in each improvement phase.

V. Conclusion

In this study, we proposed an improved PoolNet-based SOD algorithm and applied it to lightweight object detection. The PoolNet algorithm was improved using the weighted BiFPN, which was optimized and upgraded based on the traditional FPN. The developed design achieved simple and fast multi-scale fusion. Furthermore, an efficient loss function optimization scheme enabled the simultaneous capture of fine salient object structures, generating saliency maps with clear boundaries. In comparison with the original PoolNet algorithm, the accuracy of the improved DBLPoolNet algorithm was increased by 1%, and the model size was reduced by 80%. Therefore, the proposed model can theoretically be applied to embedded platforms.

References

[1] K. Zhang et al., “Channel-wise and feature-points reweights densenet for image classification,” in *Proc. 2019 IEEE Int. Conf. Image Processing*, Taiwan, China, Sep 22, 2019, pp. 410–414.

[2] Z. Zhang et al., “Pattern-affinitive propagation across depth, surface normal and semantic segmentation,” in *Proc. 2019 IEEE Conf. on Comput. Vis. Pattern Recognit.*, Long Beach, Jun 16, 2019, pp. 4106–4115.

[3] X. Chen and M. Wang, “Condition monitoring based on an improved deep learning method,” *IAENG International Journal of Computer Science*, vol. 48, no. 4, pp. 899–905, 2021.

[4] Q. Zhang, M. Yang, K. Kpalma, Q. Zheng, and X. Zhang, “Segmentation of hand posture against complex backgrounds based on saliency and skin colour detection,” *IAENG International Journal of Computer Science*, vol. 45, no. 3, pp. 435–444, 2018.

[5] J. J. Liu, Q. Hou, M. M. Cheng, J. Feng, and J. Jiang, “A simple pooling-based design for real-time salient object detection,” in *Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.*, 2019, pp. 3917–3926.

[6] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” In *Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[7] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and efficient object detection,” in *Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.*, 2020, pp. 10781—10790.

[8] L. Wang et al., “Learning to detect salient objects with image-level supervision,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 136–145.

[9] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, “The secrets of salient object segmentation,” in *Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.*, 2014, pp. 280–287.

[10] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, “Frequency-tuned salient region detection,” in *2009 IEEE Conf. on Comput. Vis. Pattern Recognit.*, 2009, pp. 1597–1604.

[11] F. Perazzi, P. Kr ähenb ühl, Y. Pritch, and A. Hornung, “Saliency filters: Contrast based filtering for salient region detection,” in *Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.*, Jun. 2012 pp. 733–740.

[12] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu, “A stagewise refinement model for detecting salient objects in images,” in *IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct 22– 29, 2017, pp. 4039–4048.

[13] N. Liu, J. Han, and M. H. Yang, “Picanet: Learning pixel-wise contextual attention for saliency detection,” in *Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.*, 2018, pp. 3089–3098.

[14] Z. Wu, L. Su, Q. M. Huang, “Cascaded partial decoder for fast and accurate salient object detection,” in *Proc. 2019 IEEE Conf. on Comput. Vis. Pattern Recognit.*, Long Beach, Jun 16–20, 2019, pp. 3907–3016.

[15] X. Qin et al., “BASNet: Boundary-aware salient object detection,” in *Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.*, 2019, pp. 7479–7489.

[16] X. Qin et al., “U2 -Net: Going deeper with nested U-structure for salient object detection,” *Pattern Recognit.*, vol. 106, p. 107404, Oct. 2020.