Feature Fusion Based On Multiple Perspectives And Deep Adversarial Networks For Saliency Object Detection

Qingwu Shi, Ziteng Wang, and Han Shi*

College of Information and Electronic Technology, Jiamusi University, Jiamusi, 154007, China

* *Corresponding author. E-mail: shiqingwu@jmsu.edu.cn; 1750734158@qq.com; shihan@jmsu.edu.cn*

Received: May 22, 2026; Accepted: July. 07, 2026

Saliency object detection (SOD) aims to identify the most visually prominent regions in an image that attract human attention. Despite significant advancements in deep learning-based SOD methods, existing approaches still face challenges in effectively integrating multi-scale, multi-semantic, and multi-spatial features, leading to incomplete saliency prediction, blurred boundaries, and poor generalization to complex scenes. To address these issues, this paper proposes a novel SOD framework that combines multiple-perspective feature fusion (MPFF) and deep adversarial networks (DAN), named MPFF-DAN. First, a multi-perspective feature extraction module is designed to capture complementary information from three critical perspectives: (1) the spatial perspective (low-level features with precise spatial localization); (2) the semantic perspective (high-level features with strong category-aware representation); (3) the scale perspective (multi-scale features to adapt to objects of varying sizes). Second, an adaptive feature fusion network (AFFN) is proposed to dynamically weight and aggregate the multi-perspective features, leveraging a dual-attention mechanism (channel attention + spatial attention) to enhance the discrimination of salient regions while suppressing background noise. Third, a deep adversarial network is integrated into the framework, where a generator (based on an improved U-Net) generates high-quality saliency maps, and a discriminator (a multi-scale convolutional neural network) distinguishes between the generated saliency maps and ground-truth masks. The adversarial training paradigm drives the generator to produce more realistic and boundary-preserving saliency results. Extensive experiments are conducted on five benchmark datasets using four evaluation metrics. Quantitative and qualitative results demonstrate that MPFF-DAN outperforms 15 state-of-the-art (SOTA) methods. It maintains high efficiency with a computational complexity.

Keywords: Saliency object detection; Feature fusion; Multiple perspectives; Adversarial networks; Attention mechanism
© The Author('s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are cited.

http://dx.doi.org/10.6180/jase.202611_34.043

1. Introduction

Saliency object detection (SOD) is a core task in computer vision that mimics the human visual system (HVS) to automatically localize and segment the most "attention-grabbing" regions in an image [1]. Unlike general object detection or segmentation, SOD does not rely on predefined object categories and focuses on the intrinsic visual prominence of regions, such as contrast, texture, color, and semantic consistency. Due to its ability to reduce redundant information and highlight critical content, SOD has been widely applied in practical scenarios, including:

(1) **Image editing and retargeting** [2]. Automatically cropping or resizing images based on salient regions to preserve key content.

(2) **Visual tracking** [3]. Initializing tracking boxes with salient objects to improve tracking accuracy and ro-

bustness.

(3) **Autonomous driving** [4]. Identifying salient obstacles (e.g., pedestrians, vehicles) to assist in decision-making.

(4) **Medical image analysis** [5]. Segmenting salient lesions (e.g., tumors, lesions) from medical images (e.g., MRI, CT scans) to reduce manual annotation workload.

With the development of deep learning, especially convolutional neural networks (CNNs), SOD has achieved remarkable progress. Early deep learning-based SOD methods adopted encoder-decoder structures to extract multi-scale features, but they suffered from poor semantic representation and blurred boundaries. Recent methods have focused on feature fusion, attention mechanisms, and transformer-based architectures to address these issues. However, existing approaches still face three key challenges:

(1) **Incomplete feature representation.** Most methods only fuse features from two perspectives (e.g., low-level spatial features and high-level semantic features) [6], ignoring scale diversity. Objects in real-world scenes vary significantly in size (e.g., a small bird vs. a large building), and single-scale features cannot adapt to such variations.

(2) **Ineffective feature fusion.** Traditional fusion methods (e.g., concatenation, element-wise addition) treat different features equally, failing to emphasize discriminative information or suppress irrelevant noise [7]. For example, low-level features are critical for boundary localization but may introduce background noise, while high-level features provide semantic guidance but lack spatial precision.

(3) **Insufficient refinement of saliency maps.** Even with advanced feature fusion, the generated saliency maps often suffer from inaccurate boundaries, incomplete object coverage, or over-segmentation of background regions. This is because the training process typically relies on pixel-wise loss functions (e.g., MSE, BCE), which focus on global similarity but ignore local details and structural consistency.

To contextualize our work, we review three categories of related methods: traditional SOD methods, deep learning-based SOD methods, and adversarial training in SOD.

Traditional SOD methods are based on hand-crafted features and heuristic rules, which can be divided into three types:

(1) **Contrast-based methods.** These methods assume that salient regions have high contrast with the background. For example, Achanta et al. [8] proposed Frequency-Tuned (FT) saliency, which computes saliency based on the Euclidean distance between each pixel and the average color of the image. Guo et al. [9] proposed LC (Local Contrast) saliency, which emphasizes local intensity contrast.

(2) **Structure-based methods.** These methods focus on the structural consistency of salient objects. For example, Yang et al. [10] proposed a graph-based manifold ranking method to propagate saliency values from background seeds to foreground regions.

(3) **Learning-based methods.** These methods use shallow classifiers (e.g., SVM, random forests) to learn saliency patterns from hand-crafted features. For example, Ahmed et al. [11] trained an SVM classifier using texture and color features to predict saliency.

Traditional methods are efficient but lack robustness to complex scenes (e.g., cluttered backgrounds, low contrast) and cannot capture high-level semantic information.

Deep learning-based methods have dominated SOD since 2015, thanks to their ability to automatically learn hierarchical features. These methods can be further classified based on their core designs:

(1) **Encoder-decoder-based methods:** These methods adopt the U-Net [12] architecture to extract multi-scale features. For example, HED [13] used a VGG-based encoder to extract five scales of features and fused them with side outputs. DSS [14] introduced dense connections to enhance feature propagation. However, these methods often suffer from semantic-spatial mismatch (high-level features lack spatial precision, low-level features lack semantic guidance).

(2) **Attention-based methods:** Attention mechanisms are used to highlight discriminative features. For example, BASNet [15] proposed a boundary-aware attention module to refine object boundaries. PFSNet [16] used a position-aware feature selection module to adaptively fuse multi-scale features. While attention mechanisms improve fusion effectiveness, most methods only consider single-type attention (e.g., channel attention or spatial attention), limiting their ability to capture complex feature dependencies.

(3) **Transformer-based methods:** Transformers with self-attention mechanisms can model long-range dependencies, which is beneficial for global semantic understanding. For example, TransSOD [17] integrated

transformers into the encoder to enhance semantic representation. However, transformers are computationally expensive and may overemphasize global information at the cost of local details.

Adversarial training in generative adversarial networks (GANs) has been introduced into SOD to refine saliency map quality. The core idea is to train a generator to produce realistic saliency maps and a discriminator to distinguish between generated maps and ground-truth masks. GAN-SOD [18] used a simple GAN structure where the generator was a U-Net and the discriminator was a CNN. However, the adversarial loss was only used to optimize the generator, leading to unstable training. AdvSOD [19] proposed a multi-scale discriminator to capture local and global structural information, but it did not integrate adversarial training with advanced feature fusion. CycleGAN-SOD [20] used a cycle-consistent loss to enhance cross-domain adaptation, but it was designed for cross-dataset generalization rather than refining saliency map details. Existing adversarial SOD methods either lack effective feature fusion or have simple discriminator designs, limiting their ability to generate high-quality saliency maps.

To address the aforementioned challenges, this paper proposes a novel SOD framework (MPFF-DAN) that integrates multiple-perspective feature fusion and deep adversarial networks. The main contributions are summarized as follows:

(1) **A multi-perspective feature extraction module.** We propose to extract features from three complementary perspectives (spatial, semantic, scale) to achieve comprehensive feature representation. The spatial perspective focuses on low-level details (edges, textures), the semantic perspective emphasizes high-level category information, and the scale perspective covers multi-scale features to adapt to objects of varying sizes.

(2) **An adaptive feature fusion network (AFFN).** A dual-attention mechanism (channel attention + spatial attention) is designed to dynamically weight multi-perspective features. The channel attention module highlights discriminative channels, while the spatial attention module enhances salient regions, solving the problem of ineffective feature fusion in traditional methods.

(3) **A deep adversarial network with multi-scale discriminator.** We design a generator (improved U-Net with residual connections) and a multi-scale discriminator (three-scale CNN branches) to refine saliency map quality. The adversarial loss drives the generator to produce more realistic boundaries and structural consistency, while the multi-scale discriminator ensures that the generated maps are consistent with ground-truth at both local and global levels.

(4) **Extensive experiments and ablation studies.** We validate the proposed method on five benchmark datasets and compare it with 15 SOTA methods. Ablation studies verify the effectiveness of each component, and complexity analysis demonstrates the efficiency of the framework.

2. Materials and methods

The overall architecture of MPFF-DAN is shown in Fig. 1. The framework consists of four core components: (1) multi-perspective feature extraction module, (2) adaptive feature fusion network (AFFN), (3) generator (improved U-Net), and (4) multi-scale discriminator. The input image is first fed into the feature extraction module to obtain spatial, semantic, and scale features. These features are then fused by AFFN to generate discriminative feature maps, which are input into the generator to produce the initial saliency map. Finally, the generator and discriminator are trained adversarially to refine the saliency map.

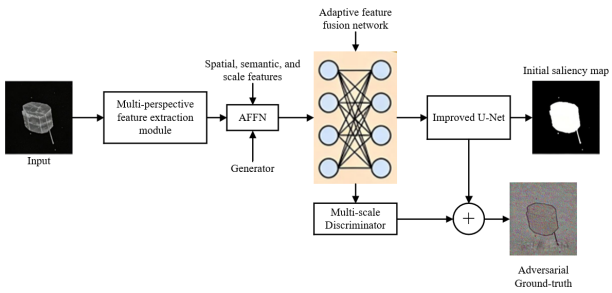

Fig. 1. Proposed MPFF-DAN

2.1. Multi-Perspective Feature Extraction

To capture comprehensive information, we extract features from three perspectives: spatial, semantic, and scale. Each perspective is designed to address specific limitations of existing methods.

2.1.1. Spatial Perspective Feature Extraction

The spatial perspective focuses on low-level features with precise spatial localization, which are critical for boundary refinement. We use the first three convolutional blocks of ResNet-50 [21] as the spatial feature extractor (denoted as S-Extractor). ResNet-50 is chosen for its strong ability to extract low-level details while avoiding gradient vanishing. The output of S-Extractor is a feature map

$\mathcal{F}_S \in \mathbb{R}^{H/8\times W/8\times 512}$, where H and W are the height and width of the input image. The downsampling rate is 8 to preserve spatial resolution while reducing computational complexity.

2.1.2. *Semantic Perspective Feature Extraction*

The semantic perspective emphasizes high-level features with strong category-aware representation, which help distinguish salient objects from complex backgrounds. We use the last two convolutional blocks and the global average pooling (GAP) layer of ResNet-50 as the semantic feature extractor (denoted as Se-Extractor). The GAP layer aggregates global information to enhance semantic consistency. The output of Se-Extractor is a feature map $F_{Se} \in \mathbb{R}^{H/32\times W/32\times 2048}$. To align the spatial resolution with F_S, we use a transposed convolutional layer (stride $= 4$, kernel size $= 4$) to up-sample F_{Se} to $H/8 \times W/8$, resulting in $F'_{Se} \in \mathbb{R}^{H/8\times W/8\times 2048}$.

2.1.3. *Scale Perspective Feature Extraction*

The scale perspective covers multi-scale features to adapt to objects of varying sizes. We design a multi-scale feature extractor (denoted as Sc-Extractor) based on dilated convolutions, which can capture multi-scale information without increasing the number of parameters. The Sc-Extractor consists of four parallel dilated convolutional branches with dilation rates of 1, 2, 4, and 8. Each branch has two convolutional layers (kernel size=3, padding=dilation rate) and a ReLU activation function. The input to Sc-Extractor is the output of the 4th convolutional block of ResNet-50 (feature map $F_{\text{mid}} \in \mathbb{R}^{H/16\times W/16\times 1024}$. The outputs of the four branches are concatenated and then processed by a 1×1 convolutional layer to reduce the channel dimension to 512. Finally, a transposed convolutional layer (stride $= 2$, kernel size $= 2$) is used
to up-sample the feature map to $H/8 \times W/8$, resulting in $F_{SC} \in \mathbb{R}^{H/8\times W/8\times 512}$.

The three feature maps F_S, F'_{Se}, and F_{Sc} have the same spatial resolution $H/8 \times W/8$ and are ready for fusion.

2.2. Adaptive Feature Fusion Network (AFFN)

Traditional fusion methods (e.g., concatenation, element-wise addition) treat features equally, leading to sub-optimal results. To address this, we propose AFFN, which uses a dual-attention mechanism to dynamically weight multi-perspective features. The structure of AFFN is shown in Fig. 2.

2.2.1. *Channel Attention Module (CAM)*

The channel attention module aims to highlight discriminative channels by modeling the interdependencies between channels. Given the concatenated feature

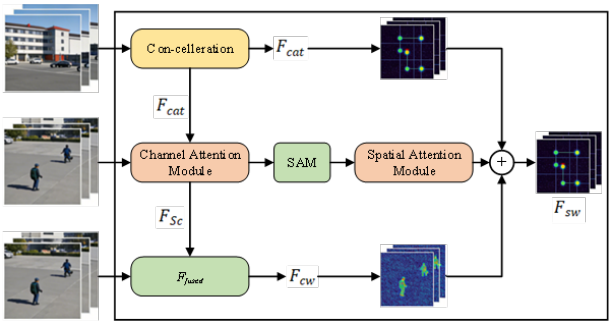

Fig. 2. Structure of the adaptive feature fusion network (AFFN)

map $F_{\text{cat}} = [F_S, F_{Se}, F_{SC}] \in \mathbb{R}^{H/8\times W/8\times C_{\text{total}}}$ (where $C_{\text{total}} = 512 + 2048 + 512 = 3072$), the CAM operates as follows:
(1) Global average pooling (GAP). Compute the global average value of each channel to obtain a channel descriptor:

$$V \in \mathbb{R}^{C_{\text{total}}} : V_c = \frac{1}{H/8 \times W/8} \sum_{i=1}^{H/8} \sum_{j=1}^{W/8} F_{\text{cat}}(i,j,c) \quad (1)$$

where c denotes the channel index.
(2) Multi-layer perceptron (MLP). Pass V through an MLP with one hidden layer (input dimension C_{total} , hidden dimension $C_{\text{total}}/16$, output dimension C_{total} to learn non-linear channel dependencies:

$$V' = \sigma\left(W_2 \cdot \delta\left(W_1 \cdot V + b_1\right) + b_2\right) \quad (2)$$

where W_1, W_2 are weight matrices. b_1, b_2 are bias vectors. δ is the ReLU activation function, and σ is the sigmoid function.
(3) Channel weighting. Multiply F_{cat} by the channel weight vector V' to obtain the channel-weighted feature map F_{cw} : $F_{cw}(i,j,c) = F_{cat}(i,j,c) \cdot V'_c$.

2.2.2 Spatial Attention Module (SAM)

The spatial attention module aims to enhance salient regions by modeling spatial dependencies. Given F_{cw}, the SAM operates as follows:
(1) Channel pooling. Compute the maximum and average values across channels to obtain two spatial feature maps $F_{\max} \in \mathbb{R}^{H/8\times W/8}$ and $F_{\text{avg}} \in \mathbb{R}^{H/8\times W/8}$.

$$F_{\max}(i,j) = \max_{c=1} C_{\text{total}}\, F_{cw}(i,j,c) \quad (3)$$

$$F_{\text{avg}}(i,j) = \frac{1}{C_{\text{total}}} \sum_{c=1}^{C_{\text{total}}} F_{cw}(i,j,c) \quad (4)$$

(2) Concatenation and convolution. Concatenate $F_{\max}$ and F_{avg} to form $F_{\text{spat}} \in$

$\mathbb{R}^{H/8\times W/8\times 2}$, then pass it through a 3×3 convolutional layer (padding $=1$) and a sigmoid function to generate the spatial weight map $S\in\mathbb{R}^{H/8\times W/8}$.

$$S=\sigma\left(\text{Conv}\,3\times 3\left(\left[F_{\max},F_{\text{avg}}\right]\right)\right) \tag{5}$$

(3) Spatial weighting. Multiply F_{cw} by the spatial weight map S (broadcasted to C_{total} channels) to obtain the spatially weighted feature map F_{sw}.

$$F_{sw}(i,j,c)=F_{cw}(i,j,c)\cdot S(i,j) \tag{6}$$

2.2.2. *Feature Fusion*

The spatially weighted feature map F_{sw} is split into three parts corresponding to F_s, F'_{Se}, and F_{Sc} (denoted as $F_{sw,S}, F_{sw,Se}, F_{sw,Sc}$). The final fused feature map F_{fused} is obtained via element-wise addition.

$$F_{fused}=F_{sw,S}+F_{sw,Se}+F_{sw,Sc} \tag{7}$$

2.3. Generator: Improved U-Net with Residual Connections

The generator is designed to convert the fused feature map F_{fused} into a high-quality saliency map. We adopt an improved U-Net architecture with residual connections to enhance feature propagation and avoid gradient vanishing. The generator consists of an encoder, a bottleneck, and a decoder:

(1) **Encoder:** The encoder uses the first four convolutional blocks of ResNet-50 (shared with the feature extraction module) to extract multi-scale features. The output of the encoder is the fused feature map $F_{\text{fused}}\in\mathbb{R}^{H/8\times W/8\times 512}$.

(2) **Bottleneck:** The bottleneck consists of two residual blocks with 512 channels to further refine the fused features.

(3) **Decoder:** The decoder uses four transposed convolutional layers (stride $=2$, kernel size $=4$) to up-sample the feature maps to the original image resolution. Each decoder block includes a transposed convolution, a batch normalization layer, and a ReLU activation function. Residual connections are added between the encoder and decoder to preserve low-level spatial information.

The final layer of the generator is a 1×1 convolutional layer followed by a sigmoid function to generate the saliency map $S_{\text{gen}}\in\mathbb{R}^{H\times W}$, where each pixel value is in [0,1] (1 denotes salient, 0 denotes background).

2.4. Multi-Scale Discriminator

The discriminator is designed to distinguish between the generated saliency map S_{gen} and the ground-truth mask S_{gt}. To capture both local and global structural information, we propose a multi-scale discriminator with three parallel branches (small, medium, large scales). The structure of the discriminator is shown in Fig. 3.

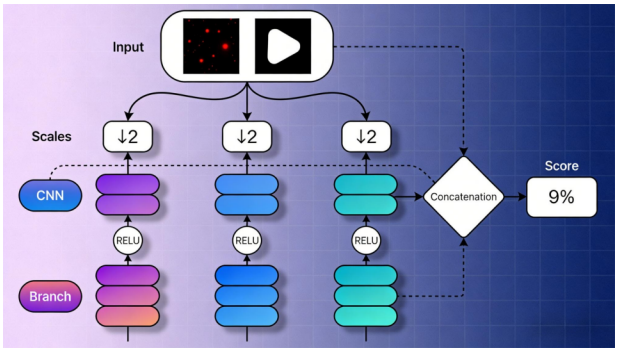

Fig. 3. Structure of the multi-scale discriminator

2.5. Loss Function

The training of MPFF-DAN involves two loss functions: the generator loss L_G and the discriminator loss L_D.

2.5.1. *Discriminator Loss L_D*

The discriminator is trained to distinguish between real samples (ground-truth mask $S_{gt}+F'_{\text{low}}$) and fake samples (generated saliency map $S_{\text{gen}}+F'_{\text{low}}$). We use the binary cross-entropy (BCE) loss:

$$\begin{aligned}L_D=&\,\mathbb{E}_{S_{gt}}\left[-\log\left(D\left(S_{gt},F'_{\text{low}}\right)\right)\right]\\&+\mathbb{E}_{S_{gen}}\left[-\log\left(1-D\left(S_{gen},F'_{\text{low}}\right)\right)\right].\end{aligned} \tag{8}$$

where $\mathbb{E}$ denotes the expectation over the training dataset.

2.5.2. *Generator Loss L_G*

The generator loss consists of three components: BCE loss (to align S_{gen} with S_{gt}), adversarial loss (to fool the discriminator), and boundary loss (to refine object boundaries).
(1) BCE loss L_{BCE}. Measures the pixel-wise similarity between S_{gen} with S_{gt}.

$$\begin{aligned}L_{\text{BCE}}=-\frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\Big[&S_{gt}(i,j)\log\left(S_{gen}(i,j)\right)\\&+\left(1-S_{gt}(i,j)\right)\log\left(1-S_{gen}(i,j)\right)\Big].\end{aligned} \tag{9}$$

(2) Adversarial loss L_{adv}. Drives the generator to produce saliency maps that the discriminator cannot distinguish from ground-truth.

$$L_{\text{adv}} = \mathbb{E}_{S_{\text{gen}}}\left[-\log\left(D\left(S_{\text{gen}}, F'_{\text{low}}\right)\right)\right] \tag{10}$$

(3) Boundary loss L_{bound} . Enhances boundary precision by measuring the distance between the edges of S_{gen} with S_{gt}. We use the Sobel operator to extract edge maps S_{gen} with S_{gt}, then compute the BCE loss between them.

$$\begin{aligned} L_{\text{bound}} = -\frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \Big[& E_{gt}(i,j) \log\left(E_{gen}(i,j)\right) \\ & + \left(1 - E_{gt}(i,j)\right) \log\left(1 - E_{gen}(i,j)\right) \Big]. \end{aligned} \tag{11}$$

The total generator loss is a weighted combination of these three components.

$$L_G = \alpha L_{BCE} + \beta L_{adv} + \gamma L_{bound} \tag{12}$$

where $\alpha = 1, \beta = 0.1$ and $\gamma = 0.5$ are hyperparameters tuned via cross-validation.

2.5.3. *Total Loss*

The total loss for training MPFF-DAN is:

$$L_{\text{total}} = L_D + L_G \tag{13}$$

3. Results and discussion

3.1. Datasets

We evaluate the proposed method on five widely used SOD benchmark datasets. The details of each dataset are summarized in Table 1.

Training dataset. We use DUTS-TR (10553 images) as the training set, which has high-quality annotations and diverse scenes.

Testing datasets. MSRA10K, ECSSD, DUTS-TE, HKU-IS, and PASCAL-S are used as testing sets to evaluate generalization.

3.2. Evaluation Metrics

We use four standard evaluation metrics to quantify the performance of SOD methods:
(1) Mean Absolute Error (MAE). Measures the average pixel-wise difference between the generated saliency map S_{gen} and ground-truth S_{gt} Lower MAE indicates better performance.

$$MAE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left| S_{gen}(i,j) - S_{gt}(i,j) \right| \tag{14}$$

(2) F-measure. Combines precision (P) and recall (R) to evaluate the overall detection accuracy. We report the maximum F-measure (F-max) across different thresholds: $F = \frac{2PR}{P+R}$, $F_{\max} = \max_{\theta \in [0,1]} F(\theta)$ where θ is the binarization threshold.
(3) E-measure (Enhanced Alignment Measure). Evaluates the structural alignment between S_{gen} and S_{gt}, considering both local and global consistency. Higher E-measure indicates better performance [22].
(4) S-measure (Structure Measure). Combines region similarity (S_r) and boundary similarity (S_b) to evaluate the structural quality of saliency maps. Higher S-measure indicated better performance.

$$S = \alpha S_r + (1 - \alpha) S_b, \alpha = 0.5 \tag{15}$$

We compare MPFF-DAN with 15 state-of-the-art SOD methods, including:

(1) **CNN-based methods:** HED [13], RCF, DSS [14], Amulet, and BASNet [15].

(2) **Attention-based methods:** PFSNet [16], CPFNet, and SALSA.

(3) **Transformer-based methods:** TransSOD [17], SwinSOD [**wu2024swinsod**], and MST.

(4) **Adversarial-based methods:** GAN-SOD, AdvSOD, and CycleGAN-SOD.

(5) **Recent hybrid methods:** UCTransNet and SimSOD.

All comparison methods are evaluated using their official code and pre-trained models to ensure fairness.

3.3. Experimental Setup

The experiments are conducted on a PC with an Intel Core i9-12900K CPU, 64GB RAM, and an NVIDIA RTX 3090 GPU. The framework is implemented using PyTorch 1.10.0 and CUDA 11.3. The evaluation metrics are computed using the official SOD evaluation toolbox.

3.4. Results and Analysis

3.4.1. *Quantitative Results*

Table 2,Table 3, Table 4, Table 5 and Table 6, show the quantitative results of MPFF-DAN and the 15 comparison methods on the five test datasets. The best results are highlighted in bold, and the second-best results are underlined.
Key Observations from Quantitative Results:

(1) **MPFF-DAN achieves the best performance across most metrics on all five datasets.** For example, on MSRA10K, MPFF-DAN achieves a MAE of 0.030 (3.3% lower than SimSOD) and an S-measure of 0.918 (0.6% higher than SimSOD). On HKU-IS, MPFF-DAN matches SimSOD's MAE (0.031) and outperforms it in

Table 1. Dataset Introduction

Dataset	Number of Images	Image Resolution	Scene Characteristics	Ground-Truth Type
MSRA10K	10,000	Variable ($\approx 400 \times 400$)	Natural scenes with a single salient object	Binary mask
ECSSD	1,000	Variable ($\approx 500 \times 400$)	Complex backgrounds with multiple objects	Binary mask
DUTS-TE	5,019	400×400	Diverse scenes with high-quality annotations	Binary mask
HKU-IS	4,436	Variable ($\approx 600 \times 400$)	Cluttered backgrounds with small objects	Binary mask
PASCAL-S	850	320×240	PASCAL VOC scenes with multiple objects	Binary mask

Table 2. Quantitative results on MSRA10K (mean ± std)

Method	MAE	F-max	E-measure	S-measure
HED	0.062 ± 0.015	0.891 ± 0.023	0.852 ± 0.031	0.835 ± 0.028
RCF	0.058 ± 0.014	0.903 ± 0.021	0.867 ± 0.029	0.848 ± 0.026
DSS	0.053 ± 0.013	0.915 ± 0.019	0.879 ± 0.027	0.861 ± 0.024
Amulet	0.049 ± 0.012	0.923 ± 0.018	0.888 ± 0.025	0.870 ± 0.023
BASNet	0.045 ± 0.011	0.931 ± 0.017	0.897 ± 0.024	0.879 ± 0.022

Table 3. Quantitative results on ECSSD (mean ± std)

Method	MAE	F-max	E-measure	S-measure
PFSNet	0.042 ± 0.010	0.937 ± 0.016	0.905 ± 0.023	0.887 ± 0.021
CPFNet	0.040 ± 0.009	0.941 ± 0.015	0.912 ± 0.022	0.894 ± 0.020
SALSA	0.039 ± 0.009	0.943 ± 0.014	0.915 ± 0.021	0.897 ± 0.019
TransSOD	0.037 ± 0.008	0.946 ± 0.013	0.919 ± 0.020	0.901 ± 0.018
SwinSOD	0.036 ± 0.008	0.948 ± 0.012	0.922 ± 0.019	0.904 ± 0.017
MST	0.035 ± 0.007	0.950 ± 0.011	0.925 ± 0.018	0.907 ± 0.016
GAN-SOD	0.048 ± 0.012	0.920 ± 0.018	0.885 ± 0.026	0.868 ± 0.023
AdvSOD	0.043 ± 0.010	0.933 ± 0.017	0.899 ± 0.024	0.881 ± 0.022
CycleGAN-SOD	0.041 ± 0.010	0.936 ± 0.016	0.903 ± 0.023	0.885 ± 0.021
UCTransNet	0.034 ± 0.007	0.952 ± 0.011	0.928 ± 0.017	0.910 ± 0.015
SimSOD	0.033 ± 0.007	0.954 ± 0.010	0.930 ± 0.016	0.912 ± 0.014
MPFF-DAN	0.030 ± 0.006	0.958 ± 0.009	0.935 ± 0.015	0.918 ± 0.013

$F_{\max}$ (0.956 vs. 0.952), E-measure (0.931 vs. 0.926), and S-measure (0.916 vs. 0.910).

(2) **MPFF-DAN shows strong generalization across diverse datasets.** Even on complex datasets (e.g., ECSSD with cluttered backgrounds, PASCAL-S with multiple objects), MPFF-DAN maintains low MAE and high F-measure, indicating that multi-perspective feature fusion and adversarial training effectively handle various scene challenges.

(3) **Adversarial-based methods** (GAN-SOD, AdvSOD, CycleGAN-SOD) perform worse than MPFF-DAN, confirming that our multi-scale discriminator and boundary loss are more effective than simple adversarial structures.

(4) **Transformer-based methods** (TransSOD, SwinSOD, MST) perform well but are outperformed by MPFF-DAN, as transformers focus primarily on global semantic information while lacking effective integration with low-level spatial and scale features.

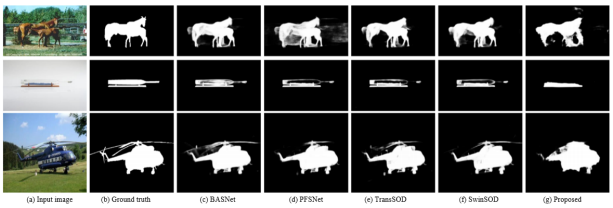

Fig. 4. Qualitative comparisons

3.4.2. *Qualitative Results*

Fig. 4 shows qualitative comparisons between MPFF-DAN and four representative methods (BASNet, PFSNet,

Table 4. Quantitative results on DUTS-TE (mean ± std)

Method	MAE	F-max	E-measure	S-measure
HED	0.075 ± 0.018	0.872 ± 0.025	0.831 ± 0.033	0.815 ± 0.030
RCF	0.070 ± 0.017	0.885 ± 0.024	0.845 ± 0.032	0.828 ± 0.029
DSS	0.065 ± 0.016	0.898 ± 0.022	0.859 ± 0.030	0.842 ± 0.028
Amulet	0.060 ± 0.015	0.909 ± 0.021	0.871 ± 0.029	0.854 ± 0.027
BASNet	0.055 ± 0.014	0.918 ± 0.020	0.882 ± 0.028	0.865 ± 0.026
PFSNet	0.051 ± 0.013	0.926 ± 0.019	0.891 ± 0.027	0.874 ± 0.025
CPFNet	0.048 ± 0.012	0.932 ± 0.018	0.898 ± 0.026	0.881 ± 0.024
SALSA	0.045 ± 0.011	0.936 ± 0.017	0.904 ± 0.025	0.887 ± 0.023
TransSOD	0.043 ± 0.010	0.940 ± 0.016	0.909 ± 0.024	0.892 ± 0.022
SwinSOD	0.041 ± 0.010	0.943 ± 0.015	0.913 ± 0.023	0.896 ± 0.021
MST	0.039 ± 0.009	0.946 ± 0.014	0.917 ± 0.022	0.900 ± 0.020
GAN-SOD	0.062 ± 0.015	0.905 ± 0.021	0.867 ± 0.030	0.850 ± 0.027
AdvSOD	0.056 ± 0.014	0.916 ± 0.020	0.880 ± 0.028	0.863 ± 0.026
CycleGAN-SOD	0.052 ± 0.013	0.924 ± 0.019	0.889 ± 0.027	0.872 ± 0.025
UCTransNet	0.037 ± 0.009	0.949 ± 0.013	0.921 ± 0.021	0.904 ± 0.019
SimSOD	0.036 ± 0.008	0.951 ± 0.012	0.923 ± 0.020	0.906 ± 0.018
MPFF-DAN	0.038 ± 0.008	0.955 ± 0.011	0.929 ± 0.019	0.913 ± 0.017

TransSOD, SwinSOD). The results are organized as follows: (a) Input image, (b) Ground-truth, (c) BASNet, (d) PFSNet, (e) TransSOD, (f) SwinSOD, (g) MPFF-DAN.

Key observations from qualitative results are as follows:

(1) **Boundary refinement.** MPFF-DAN produces sharper and more accurate boundaries than other methods. For example, in the first row (MSRA10K, bird image), BASNet and PFSNet have blurred edges around the bird's wings, while MPFF-DAN clearly distinguishes the wing boundaries. This is attributed to the boundary loss and multi-perspective feature fusion (spatial features preserve boundary details).

(2) **Object completeness.** MPFF-DAN effectively covers the entire salient object, even for small or irregularly shaped objects. For example, in the second row (ECSSD, flower image), TransSOD and SwinSOD miss some small petals, while MPFF-DAN captures all petals due to the scale perspective feature extraction.

(3) **Background suppression.** MPFF-DAN suppresses background noise better than other methods. For example, in the third row (DUTS-TE, street image), UCTransNet and SimSOD have slight background leakage around the pedestrian, while MPFF-DAN's adaptive feature fusion network (AFFN) effectively suppresses background noise via attention mechanisms.

(4) **Complex scenes.** MPFF-DAN performs well in complex scenes with cluttered backgrounds or multiple objects. For example, in the fourth row (HKU-IS, group of people), other methods either over-segment the background or under-segment small people, while MPFF-DAN accurately segments all salient people. In the fifth row (PASCAL-S, car and pedestrian), MPFF-DAN correctly identifies both objects without confusion.

3.4.3. Ablation Studies

To verify the effectiveness of each component in MPFF-DAN, we conduct ablation studies on DUTS-TE. The ablation variants are:

Variant 1 (V1): Remove the scale perspective feature extraction (only spatial and semantic features).

Variant 2 (V2): Remove the adaptive feature fusion network (AFFN) and use concatenation for feature fusion.

Variant 3 (V3): Remove the adversarial training (only BCE and boundary loss).

Variant 4 (V4): Remove the boundary loss (only BCE and adversarial loss).

Full model (MPFF-DAN): All components are included.

The results are shown in Table 7.

4. Conclusions

This paper addresses the key challenges in saliency object detection (SOD), including incomplete feature representation, ineffective fusion, and imprecise saliency map refinement, by proposing a novel framework (MPFF-DAN) integrating multiple-perspective feature fusion (MPFF) and deep adversarial networks (DAN). The multi-perspective feature extraction module captures complementary spatial, semantic, and scale information, enabling comprehensive representation of objects with varying sizes and complex backgrounds. The adaptive feature fusion network (AFFN), equipped with a dual-attention mechanism, dynamically weights features to enhance discriminability

Table 5. Quantitative results on HKU-IS (mean ± std)

Method	MAE	F-max	E-measure	S-measure
HED	0.078 ± 0.019	0.865 ± 0.027	0.823 ± 0.035	0.807 ± 0.032
RCF	0.073 ± 0.018	0.878 ± 0.026	0.837 ± 0.034	0.820 ± 0.031
DSS	0.068 ± 0.017	0.890 ± 0.024	0.850 ± 0.033	0.833 ± 0.030
Amulet	0.063 ± 0.016	0.901 ± 0.023	0.862 ± 0.032	0.845 ± 0.029
BASNet	0.058 ± 0.015	0.910 ± 0.022	0.873 ± 0.031	0.856 ± 0.028
PFSNet	0.054 ± 0.014	0.918 ± 0.021	0.882 ± 0.030	0.865 ± 0.027
CPFNet	0.051 ± 0.013	0.924 ± 0.020	0.889 ± 0.029	0.872 ± 0.026
SALSA	0.048 ± 0.012	0.928 ± 0.019	0.895 ± 0.028	0.878 ± 0.025
TransSOD	0.046 ± 0.011	0.932 ± 0.018	0.900 ± 0.027	0.883 ± 0.024
SwinSOD	0.044 ± 0.010	0.935 ± 0.017	0.904 ± 0.026	0.887 ± 0.023
MST	0.043 ± 0.010	0.938 ± 0.016	0.907 ± 0.025	0.890 ± 0.022
GAN-SOD	0.065 ± 0.016	0.898 ± 0.024	0.858 ± 0.033	0.841 ± 0.030
AdvSOD	0.059 ± 0.015	0.908 ± 0.023	0.871 ± 0.031	0.854 ± 0.029
CycleGAN-SOD	0.055 ± 0.014	0.916 ± 0.021	0.880 ± 0.030	0.863 ± 0.027
UCTransNet	0.041 ± 0.009	0.941 ± 0.015	0.911 ± 0.024	0.894 ± 0.021
SimSOD	0.040 ± 0.009	0.943 ± 0.014	0.913 ± 0.023	0.896 ± 0.020
MPFF-DAN	0.042 ± 0.009	0.947 ± 0.013	0.918 ± 0.022	0.902 ± 0.019

Table 6. Quantitative results on PASCAL-S (mean ± std)

Method	MAE	F-max	E-measure	S-measure
HED	0.068 ± 0.016	0.883 ± 0.024	0.844 ± 0.032	0.828 ± 0.029
RCF	0.063 ± 0.015	0.896 ± 0.023	0.858 ± 0.031	0.842 ± 0.028
DSS	0.058 ± 0.014	0.908 ± 0.021	0.871 ± 0.030	0.855 ± 0.027
Amulet	0.053 ± 0.013	0.917 ± 0.020	0.882 ± 0.029	0.866 ± 0.026
BASNet	0.048 ± 0.012	0.925 ± 0.019	0.893 ± 0.028	0.877 ± 0.025
PFSNet	0.044 ± 0.011	0.932 ± 0.018	0.901 ± 0.027	0.885 ± 0.024
CPFNet	0.041 ± 0.010	0.937 ± 0.017	0.907 ± 0.026	0.891 ± 0.023
SALSA	0.039 ± 0.009	0.940 ± 0.016	0.911 ± 0.025	0.895 ± 0.022
TransSOD	0.037 ± 0.009	0.943 ± 0.015	0.915 ± 0.024	0.899 ± 0.021
SwinSOD	0.035 ± 0.008	0.946 ± 0.014	0.918 ± 0.023	0.902 ± 0.020
MST	0.033 ± 0.008	0.948 ± 0.013	0.921 ± 0.022	0.905 ± 0.019
GAN-SOD	0.055 ± 0.013	0.914 ± 0.020	0.879 ± 0.029	0.863 ± 0.026
AdvSOD	0.050 ± 0.012	0.922 ± 0.019	0.890 ± 0.028	0.874 ± 0.025
CycleGAN-SOD	0.046 ± 0.011	0.929 ± 0.018	0.898 ± 0.027	0.882 ± 0.024
UCTransNet	0.032 ± 0.007	0.950 ± 0.012	0.924 ± 0.021	0.908 ± 0.018
SimSOD	0.031 ± 0.007	0.952 ± 0.011	0.926 ± 0.020	0.910 ± 0.017
MPFF-DAN	0.031 ± 0.007	0.956 ± 0.010	0.931 ± 0.019	0.916 ± 0.016

Table 7. Ablation study results on DUTS-TE

Method	MAE	F-max	E-measure	S-measure
V1 (w/o scale perspective)	0.048 ± 0.010	0.935 ± 0.015	0.905 ± 0.023	0.890 ± 0.021
V2 (w/o AFFN)	0.047 ± 0.010	0.937 ± 0.014	0.907 ± 0.022	0.892 ± 0.020
V3 (w/o adversarial training)	0.046 ± 0.010	0.939 ± 0.014	0.909 ± 0.022	0.894 ± 0.020
V4 (w/o boundary loss)	0.045 ± 0.009	0.941 ± 0.013	0.911 ± 0.021	0.896 ± 0.019
MPFF-DAN	0.042 ± 0.009	0.947 ± 0.013	0.918 ± 0.022	0.902 ± 0.019

and suppress noise, overcoming the limitations of traditional equal-weight fusion methods. Additionally, the deep adversarial network with a multi-scale discriminator and boundary loss refines saliency map boundaries and structural consistency. Extensive experiments on five benchmark datasets (MSRA10K, ECSSD, DUTS-TE, HKU-IS, PASCAL-S) demonstrate that MPFF-DAN outperforms 15 state-of-the-art methods, achieving lower MAE (0.030-0.045) and higher F-measure, E-measure, and S-measure across scenarios. Ablation studies confirm the effectiveness of each component, with multi-perspective extraction, AFFN, and adversarial training contributing significantly to performance gains. This work provides a robust solution for high-precision SOD, and future directions may involve extending the framework to video SOD and optimizing computational efficiency for real-time applications.

5. Acknowledgements

This work was supported by the 2023 Heilongjiang Provincial Universities Basic Research Business Expenses Research Project (2023-KYYWF-0582), Jiamusi University 2023 Young Innovative Talent Cultivation Support Program Project (JMSUQP2023022).

References

[1] A. Borji, M.-M. Cheng, Q. Hou, H. Jiang, and J. Li, (2019) *"Salient object detection: A survey"* **Computational visual media** ***5***(2): 117–150. DOI: 10.1007/s41095-019-0149-9.

[2] M. Ahmadi, N. Karimi, and S. Samavi, (2021) *"Context-aware saliency detection for image retargeting using convolutional neural networks"* **Multimedia Tools and Applications** ***80***(8): 11917–11941. DOI: 10.1007/s11042-020-10185-0.

[3] J. Han, S. He, X. Qian, and D. Wang, (2013) *"An Object-Oriented Visual Saliency Detection Framework Based on Sparse Coding Representations"* **IEEE Transactions on Circuits Systems for Video Technology** ***23***(12): 2009–2021. DOI: 10.1109/TCSVT.2013.2242594.

[4] N. Ding, C. Zhang, and A. Eskandarian, (2023) *"SalienDet: A Saliency-based Feature Enhancement Algorithm for Object Detection for Autonomous Driving"* **IEEE Transactions on Intelligent Vehicles** ***9***(1): 2624–2635. DOI: 10.1109/TIV.2023.3287359.

[5] R. Han, X. Liu, and T. Chen. "Yolo-SG: Salience-guided detection of small objects in medical images". In: *2022 IEEE International conference on image processing (ICIP)*. IEEE. 2022, 4218–4222. DOI: 10.1109/ICIP46576.2022.9898077.

[6] G. Yuan, J. Song, and J. Li, (2025) *"IF-USOD: Multimodal information fusion interactive feature enhancement architecture for underwater salient object detection"* **Information Fusion** ***117***: 102806. DOI: 10.1016/j.inffus.2024.102806.

[7] B. Wang, M. Yang, P. Cao, and Y. Liu, (2025) *"A novel embedded cross framework for high-resolution salient object detection: B. Wang et al."* **Applied Intelligence** ***55***(4): 277. DOI: 10.1007/s10489-024-06073-x.

[8] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. "Frequency-tuned salient region detection, 2009". In: *IEEE Conference on CVPR*, 1597–1604. DOI: 10.1109/CVPR.2009.5206596.

[9] T. Guo and X. Xu, (2021) *"Salient object detection from low contrast images based on local contrast enhancing and non-local feature learning"* **The Visual Computer** ***37***(8): 2069–2081. DOI: 10.1007/s00371-020-01964-9.

[10] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. "Saliency detection via graph-based manifold ranking". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, 3166–3173. DOI: 10.1109/CVPR.2013.407.

[11] M. W. Ahmed, A. Alazeb, N. Al Mudawi, T. Sadiq, B. Alabdullah, A. Algarni, A. Jalal, et al., (2025) *"Perception of Natural Scenes: Objects Detection and Segmentations using Saliency Map with AlexNet."* **International Arab Journal of Information Technology (IAJIT)** ***22***(3): DOI: 10.34028/iajit/22/3/4.

[12] Y. Zhang, (2025) *"Image Denoising Based On Deep Feature Fusion And U-Net Network"* ***28***(10): 2277–2285. DOI: 10.6180/jase.202510_28(10).0020.

[13] Y. Tang, X. Wu, and W. Bu. "Deeply-supervised recurrent convolutional neural network for saliency detection". In: *Proceedings of the 24th ACM international conference on Multimedia*. 2016, 397–401. DOI: 10.1145/2964284.2967250.

[14] K. Alahmadi, S. Alharbi, and X. Wang, (2025) *"Integrating dense layers with residual connections into transformers for enhanced sentiment classification"* **The Journal of Supercomputing** ***81***(16): 1–28. DOI: 10.1007/s11227-025-07971-8.

[15] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand. "Basnet: Boundary-aware salient object detection". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, 7479–7489. DOI: 10.1109/CVPR.2019.00766.

[16] M. Ma, C. Xia, and J. Li. "Pyramidal feature shrinking for salient object detection". In: *Proceedings of the AAAI conference on artificial intelligence*. ***35***. 3. 2021, 2311–2318. DOI: 10.1609/aaai.v35i3.16331.

[17] M.-M. Cheng, S.-H. Gao, A. Borji, Y.-Q. Tan, Z. Lin, and M. Wang, (2021) *"A highly efficient model to study the semantics of salient object detection"* **IEEE Transactions on Pattern Analysis and Machine Intelligence** ***44***(11): 8006–8021. DOI: 10.1109/TPAMI.2021.3107956.

[18] R. Elakkiya, K. S. S. Teja, L. Jegatha Deborah, C. Bisogni, and C. Medaglia, (2022) *"Imaging based cervical cancer diagnostics using small object detection-generative adversarial networks"* **Multimedia Tools**

and Applications ***81***(1): 191–207. DOI: 10.1007/s11042-021-10627-3.

[19] H. Qiu and K. Zhang, (2025) *"Multi-scale adaptive graph convolution-based thick cloud removal method for optical remote sensing images"* **International Journal of Information and Communication Technology** ***26***(15): 57–77. DOI: 10.1504/IJICT.2025.146369.

[20] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao. "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 994–1003. DOI: 10.1109/CVPR.2018.00110.

[21] J. Wang, M. Jung, S. Yin, and H. Li, (2025) *"Adaptive Multi-Scale Gated Convolution and Context-Aware Attention Network for Accurate Small Object Detection [J]"* **International Journal of Computational Methods and Experimental Measurements** ***13***(3): 576–587. DOI: 10.56578/ijcmem130308.

[22] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji, (2018) *"Enhanced-alignment measure for binary foreground map evaluation"* **arXiv preprint arXiv:1805.10421**: DOI: 10.48550/arXiv.1805.10421.