Perceptual loss guided Generative adversarial network for saliency detection

Xiaoxu Cai [a,b], Gaige Wang [c], Jianwen Lou [b,*,1], Muwei Jian [a,d], Junyu Dong [c], Rung-Ching Chen [e], Brett Stevens [a], Hui Yu [a,*,1]

[a] School of Creative Technologies, University of Portsmouth, Portsmouth PO1 2DJ, United Kingdom
[b] College of Computer Science and Technology , Zhejiang University, Hangzhou 310000, China
[c] College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China
[d] School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan 250000, China
[e] Department of Information Management, Chaoyang University of Technology, Taiwan 41349, China

Keywords:
Saliency detection
Deep learning
Perceptual loss
Generative Adversarial Network

A B S T R A C T

In this work, we introduce a novel approach for saliency detection through the utilization of a generative adversarial network guided by perceptual loss. Achieving effective saliency detection through deep learning entails intricate challenges influenced by a multitude of factors, with the choice of loss function playing a pivotal role. Previous studies usually formulate loss functions based on pixel-level distances between predicted and ground-truth saliency maps. However, these formulations don't explicitly exploit the perceptual attributes of objects, such as their shapes and textures, which serve as critical indicators of saliency. To tackle this deficiency, we propose an innovative loss function that capitalizes on perceptual features derived from the saliency map. Our approach has been rigorously evaluated on six benchmark datasets, demonstrating competitive performance when compared against the forefront methods in terms of both Mean Absolute Error (MAE) and F-measure. Remarkably, our experiments reveal consistent outcomes when assessing the perceptual loss using either grayscale saliency maps or saliency-masked colour images. This observation underscores the significance of shape information in shaping the perceptual saliency cues.

The code is available at https://github.com/XiaoxuCai/PerGAN.

1. Introduction

Saliency detection, a cornerstone challenge within the realms of psychology and computer vision, strives to replicate the human visual and cognitive systems' capacity to identify prominent objects in a scene comprehensively [1]. This pursuit holds significant implications for a wide range of computer vision tasks, such as image segmentation [2] and object recognition. A paramount pre-requisite for successful saliency detection is the provision of a robust feature representation of the target object. Traditional approaches [3–5] have mainly concentrated on crafting low-level features such as color contrast and edges. However, when salient objects are

* Corresponding authors.
E-mail addresses: jianwen.lou@zju.edu.cn (J. Lou), hui.yu@port.ac.uk (H. Yu).
[1] The corresponding authors Dr. Jianwen Lou and Prof. Hui Yu is responsible for ensuring that the descriptions are accurate and agreed by all authors.

embedded within complex scenes, from the hand-crafted low-level features the detection model may struggle to yield discernible distinctions between the background and the salient regions. An effective remedy for this problem is learning high-level feature representations using deep neural networks, exemplified by Convolutional Neural Networks (CNNs) [6,7] and Generative Adversarial Networks (GANs) [8].

The performance of the deep learning-based method is affected by a multitude of factors, in which the loss function is a critical one. The loss function plays a key role in expediting the training convergence and improving the network's inference accuracy. For instance, the Wasserstein GAN (WGAN) [9] proposes to use the Earth Mover's Distance (EMD) instead of the conventional Jensen–Shannon divergence as the training loss, engendering greater training stability and heightened GAN performance. Despite its importance, the loss function adopted in deep saliency detection model is mostly confined to the measurement on the pixel-level distance between the predicted saliency map and its ground-truth counterpart. This loss function is insensitive to perceptual saliency cues such as the object's shape and texture. In the meantime, recent studies in neural image generation have highlighted the effectiveness of a perceptual loss function rooted in feature maps extracted from multiple layers of pre-trained deep models. Such a perceptual loss function exhibits unprecedented ability in capturing high-level semantic information and pushes the quality of the generated image to a new level. With these observations, we envisage a promising and largely-unexplored space for enhancing saliency detection with a perception-aware loss.

In this paper, we focus on generating superior-quality saliency maps with a Generative Adversarial Network (GAN) [10] by leveraging the perceptual loss [11]. Within the GAN framework, training unfolds as an intricate interplay between two neural networks: the generator, which is responsible for synthesizing samples that align with the training dataset, and the discriminator, which is tasked on distinguishing real samples from those fake produced by the generator. In this study, a real sample consists of the input RGB image and its ground-truth saliency map, a fake sample consists of the RGB image and the predicted saliency map. Instead of using the intuitive pixel-level loss functions like the L_1 loss, we propose to intensify GAN training with a novel perceptual loss function, which measures the disparity between the predicted saliency-masked object and its real counterpart in multiple latent spaces of a pre-trained deep neural network. The perceptual loss not only exploits the salient object's low-level visual cues, but also capitalizes on its high-level semantic information such as shape and texture. To further strengthen the network's generation ability, we introduce a coarse-to-fine structure: a preliminary saliency map is generated from the latent feature vector, it then undergoes refinement through two fully convolutional layers. We also apply a multi-scale discriminator that penalizes the generator's prediction error on multiple image resolutions, which shows superiority over using the conventional discriminator.

Our main contributions can be concluded as follows:

- We propose a novel Generative Adversarial Network (GAN) framework named Perceptual Loss Guided GAN (PerGAN) for salient object detection on images. The cornerstone of our approach is a novel perceptual loss function, which encompasses both the salient object's low-level visual cues and its high-level semantic information. This amalgamation ensures the salient object's edge localization accuracy while bolstering the comprehensive portrayal of salient object in terms of its structural completeness.
- We introduce a simple yet effective coarse-to-fine structure into the generator to convert intermediate feature map to saliency map smoothly.
- We employ a multi-scale discriminator to enhance the generator's fitting capacity by discerning between real and fake samples across varying resolutions.

The rest of this paper is structured as follows: Section 2 recaps the related work, Section 3 elaborates the proposed method, Section 4 reports and analyses the experimental results, and Section 5 provides a comprehensive summary of this paper.

2. Related work

2.1. Saliency detection

Image saliency detection is a long-standing research topic in the field of computer vision. Initial approaches rely on hand-crafted feature representations such as edge, color contrast, and texture pattern to estimate salient objects. While these manual features excel in depicting object's low-level attributes, they fall short in capturing the semantic essence of the object. Recent advances in salient object detection have witnessed the emergence of deep learning methods [12–21]. We categorize these methods into three main groups. Firstly, there are methods founded on local image cues, which predict saliency for segmented regions. For example, Wang et al. [18] first divide images into distinct segments, then deduce segment-wise saliency labels using multilayer perceptron (MLP) from deep features extracted via Convolutional Neural Networks (CNNs). While this kind of method demonstrate substantial performance improvements, they overlook global image context due to their reliance on segmented regions. To surmount this limitation, the second category entails fully convolutional network (FCN) [15,16,19,20], capable of directly estimating saliency map from the input image in an end-to-end manner. Noteworthy contributions within this method stream include the fundamental work by Zhang et al. [15]. Furthermore, extensions exploiting skip connections [16], attention mechanisms [14,17], multi-scale operations [19,20] have been devised to enhance salient object detection. Zhang et al. [21] introduce a distinctive FCN-based saliency detection framework that requires no human annotations for training, by leveraging a novel supervision synthesis scheme. The third category is a combination of the first two ones. Wang et al. [20] pioneer this category by proposing a hybrid network-based model to concurrently forecast local and global saliency cues. However, it's worth pointing out that this hybrid model incurs high computational costs.

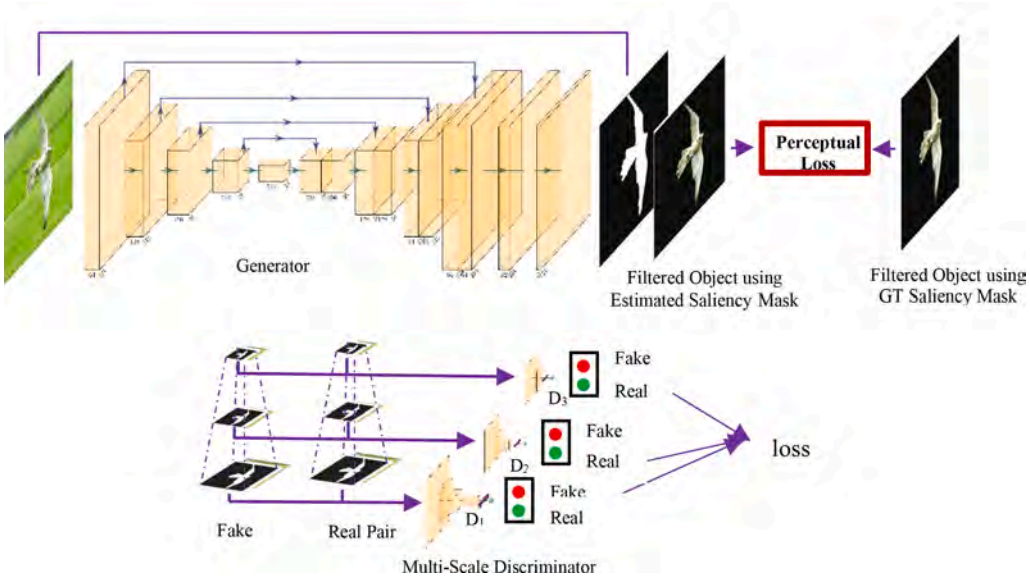

Fig. 1. The framework of PerGAN.

2.2. Generative adversarial networks

Generative adversarial networks (GANs) [10] has demonstrated promising efficacy across diverse visual analysis tasks, particularly in image generation [9,22,23]. Their utility has extended to saliency detection as well. Cai et al. [24] employ U-net and short connections within the generator to learn saliency maps. Distinctively, Zhu et al. [25] devise a GAN featuring a multi-scale end-to-end generator to yield pixel-wise saliency labels. Notably, CBGAN [8] integrates capsule blocks into both the generator and discriminator to estimate salient objects. However, despite these advancements, the abovementioned approaches primarily update networks through minimizing conventional adversarial losses rooted in low-level feature representations. The integration of perceptual loss into the optimization framework remains unexplored.

2.3. Perceptual loss function

The impact of loss function on neural network's performance is well-recognized. Typically, a loss function encompasses a distance measurement for quantifying dissimilarities between two samples' feature representations. A comprehensive assessment of distance measurement techniques, including KL divergence, EMD, MSE, SIM, etc., can be found in Ref. [26]. Among these, Bhattacharyya distance has exhibited the best performance in saliency detection [27]. However, the selection of appropriate feature representations remains an open challenge. Recent endeavors [11,22] have based loss functions on feature representations extracted from pre-trained neural networks, culminating in the generation of high-quality images. This approach imbues the loss with semantic information, such as the target object's shape, texture and category. Such a kind of loss is commonly known as perceptual loss. However, using perceptual loss to train GANs for saliency detection is uncharted territory. In this paper, we pioneer the combination of GANs and perceptual loss for robust saliency estimation.

3. Perceptual loss guided GAN

We devise a perceptual loss guided GAN – PerGAN for detecting salient objects from images. As depicted in Fig. 1, PerGAN comprises two convolutional neural networks: a generator and a multi-scale discriminator. Training PerGAN entails the contest between the two neural networks, in which the generator tries to synthesize saliency maps that can fool the discriminator, while the discriminator aims at distinguishing real saliency maps from those produced by the generator. The training procedure is guided by a perceptual loss that measures the distance between the estimated salient region and its ground-truth counterpart in deep feature spaces. In this section, we expound the structure of PerGAN, the perceptual loss function and the training details.

3.1. Generator

The generator has an auto-encoder architecture with encoder-decoder skip connections. The encoder is mainly composed of five convolutional layers, each with a filter size of 3x3 and a stride of 1×1. Following each convolutional layer, max pooling with 2×2 filters is applied to down-sample the feature maps. Following through the encoder, the feature map shrinks from 224×224 to 14×14, while the number of channels increases from 64 to 512. For a better convergence during training, the encoder is initialized with a pre-

trained VGG-16 model, which excels in image generation tasks. Inspired by Ref. [17], the encoder's output, bypassing linear transformation for latent space conversion, is directly fed to the decoder. The decoder is basically symmetric to the encoder, which features five transposed convolutional layers and applies up-sampling for saliency map generation. At the end of the decoder, we propose to attach two fully convolutional layers with 1×1 filters to finetune the saliency map from coarse to fine, thus enabling a smooth generation. Noteworthy, the filter size for decoding is increased to 5×5. for a wider receptive field that is supposed to be more sensitive to global features of the salient object. Both encoder and decoder employ ReLU activation, except for the final layer that uses logistic regression for squashing the output into [0, 1]. Last but not least, we apply skip-connections to allow local structure propagation between the encoder and the decoder.

3.2. Discriminator

A discriminator that excels at discerning between real and fake samples is essential for learning a strong generator. To this end, we propose to employ a multi-scale discriminator for adversarial training. The discriminator comprises three sub-networks (denoted as D_1, D_2, and D_3), operating across three different image resolutions. The sub-network has a similar structure to the generator's encoder, which principally consists of convolutional layers, but down-samples feature map with 2×2 striding convolution rather than max pooling. The number of convolutional layers depends on the input image's resolution, varying from 4, 3 to 2. All convolutional layers except for the last one are followed with a ReLU activation. To achieve overall consistency between the estimated saliency map and the real one, D_1 and D_2, which are fed with low-resolution samples, employ 5×5. filters for a relatively large receptive field. D_3 that accepts higher-resolution samples as input instead employs 3x3 filters to encourage more a local match between the two saliency maps.

3.3. Loss functions

The objective function to optimize PerGAN includes three weighted components: an adversarial loss, a pixel-level L_1 loss, and a perceptual loss.

3.3.1. Adversarial loss

The adversarial loss indicates how well the generator can deceive the discriminator and the discriminator can identify real and fake samples. It is formulated as:

$$L_{adv} = \sum_{k=1}^{3} E_{(x,y)}[\log(D_k(x,y))] + E_{(x,y')}[\log(1 - D_k(x, G(x)))] \tag{1}$$

As mentioned in the previous section, D_k refers to the k^{th} discriminator and G stands for the generator. The input image is defined as x, and its corresponding saliency map is defined as y. Following the GAN training routine, we update the generator twice and the discriminator once in each iteration.

3.3.2. Smooth L_1 loss

Pixel-wise losses such as L_2 loss and L_1 loss are popular choices in image generation tasks [11]. Compared with L_2 loss, L_1 loss has shown superiority on preserving high-frequency details in the generated images. In PerGAN, we apply L_1 loss between the real sample T and the generated sample $G(x)$:

$$L_{l1} = |G(x) - T| \tag{2}$$

3.3.3. Perceptual loss

Perceptual loss gauges the discrepancy between two images' high-level representations extracted from a well-trained deep neural network, which has been validated in numerous image generation tasks [28,29]. Inspired by this observation, we propose to exploit the perceptual loss for high-quality saliency map generation. The perceptual loss applied in PerGAN contains two parts. One is the content loss for measuring the difference of the object's content, such as shape, the other is the style loss for measuring the difference of the object's style, such as texture and colour. Both losses are built upon the intermediate feature spaces embedded in a pre-trained VGG-16 model.

Content loss: Similar as [11], we encourage the salient objects extracted from the input RGB image using the predicted saliency map S (or $G(x)$) or the ground truth saliency map T to have similar feature representations. The content loss between two masked images after passing through the VGG-16's Conv1_2 layer is defined as follows:

$$L_{content} = \sum_{i=1}^{c^*l^*w} \frac{(V_i(T^*x) - V_i(S^*x))^2}{c^*l^*w} \tag{3}$$

here, c, l and w represent the output's channels (channel amount), length and width separately. $V()$ is the non-linear transformation, performed in VGG-16. * represents point-wise multiplication.

Style loss: It can be found that the content loss is defined on the shallow feature spaces of VGG-16, which mainly accounts for low-level visual information such as the object's edges (see Fig. 2). To encourage a semantic-level consistency between the predicted

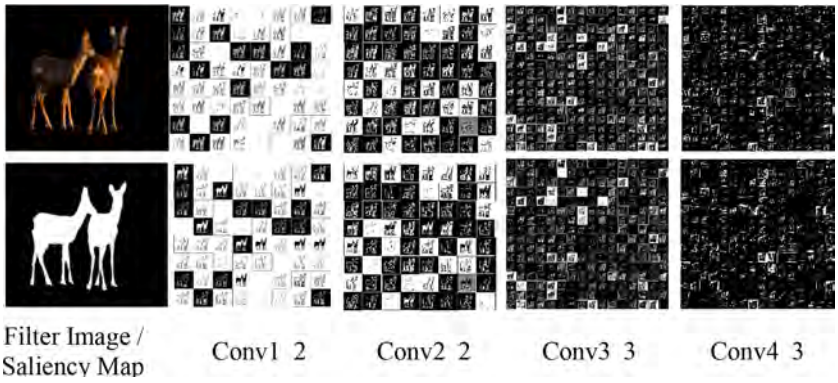

Fig. 2. The feature map examples extracted from VGG-16. The output of Conv1_2 was adapted for building content loss. All feature maps were used to reconstruct the style loss. Obviously, the shallow layer usually extracts the obvious edge and shape information. While the deep layer records abstract semantic information.

saliency map and the ground-truth map, we further enhance the perceptual loss function with a style loss item. The style loss is originally proposed to measure the differences between the style features of two different images for high-quality neural style transfer. The style feature is supposed to depict the image from some abstract and semantic views, which has shown close connections to the image's texture and color. The style feature is derived from a few specific convolutional layers of a pre-trained neural network like VGG-16, and defined as the correlation between two different groups of feature maps, where the correlation is usually formulated into a Gram matrix. For an input x, the output $\varphi_j(x)$ of the VGG-16's j^{th} layer is with size $C_j \times H_j \times W_j$. The elements of Gram matrix are described by:

$$GM^{\varphi}(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h}^{H_j} \sum_{w}^{W_j} \varphi_j(x)_{h,w,c} \varphi_j(x)_{h,w,c'} \tag{4}$$

The size of Gram matrix is$C_j \times C_i$. We use the outputs of VGG-16's Conv1_2, Conv2_2, Conv3_3 and Conv4_3 to construct 4 correlation matrices. Example feature maps across those layers can be seen in Fig. 2. The style loss is then defined as the distance between the Gram matrices of the predicted salient regions and the ground-truth salient regions:

$$L_{style} = \sum_{n=1}^{4} \| GM(V_n(S^*x)) - GM(V_n(T^*x)) \|_{\mathbb{F}}^2 \tag{5}$$

The perceptual loss is a weighted sum of the style loss and content loss:

$$L_{percep} = L_{content} + \lambda L_{style} \tag{6}$$

The overall objective function to train PerGAN is:

$$L_{total} = \omega_{adv} * L_{adv} + \omega_{l1} * L_{l1} + \omega_{percep} * L_{percep} \tag{7}$$

where ω_{adv}, ω_{l1} and ω_{percep} are weights for balancing different loss items.

4. Experiments

4.1. Datasets

In line with previous study [17], we evaluate the proposed PerGAN on the following saliency benchmark datasets.

MSRA-10K [3]: an extension of MSRA-A and ASD dataset, comprising 10 K images with pixel-wise saliency annotations.

ECSSD [30]: an extended version of CSSD dataset, consisting of 1 K natural images collected from the internet. The saliency labels were annotated by 5 human subjects.

PASCAL-S [31]: containing 850 natural images with saliency segmentation masks annotated by 12 human subjects. Binary saliency maps were derived using a threshold of 0.5 [32]. The subject's eye fixation information during labelling is also provided.

HKU-IS [32]: having 4,447 images, each is with pixel-level saliency annotations. This dataset is notably challenging since the images often contain multiple salient objects and have low colour contrast. The dataset is normally partitioned into a training set (2,450 samples), a validation set (500 samples), and a testing set (1,447 samples).

DUTS [33]: encompassing 15,572 complex images along with per-pixel saliency annotations. It covers a wide range of scenes, including indoor, outdoor, human, animals, and vehicles. The dataset is commonly divided into a training subset (10,552 images) and a testing subset (5,019 images).

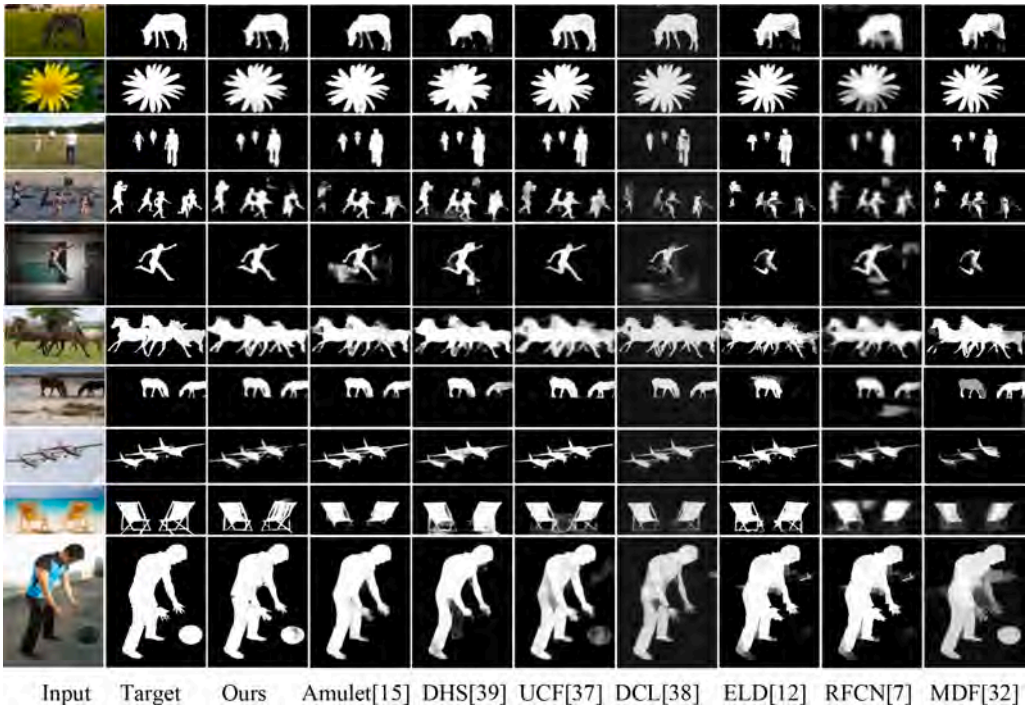

Fig. 3. Saliency maps generated by state-of-the -art methods and our (PerGAN) method.

DUT-OMRON [34]: consisting of 5,167 natural images and offering both bounding box and pixel-wise saliency annotations. It provides multiple saliency annotations from different human labellers, making it a demanding benchmark for saliency detection.

THUR15K [35]: providing 6 K images with accurate pixel-wise saliency annotations. The images cover five object categories: butterfly, coffee mug, dog, giraffe, and plane. As the scene background in the image is quite complex, to achieve high saliency detection accuracy on this dataset is a formidable task.

4.2. Implement details

Following previous studies [12,16,36,37], we use the whole MSRA-10K [3] dataset to train PerGAN. The initial data is then augmented with random image flip and rotation, resulting in a training set of ~ 80 K images, with each image in 224x224. We follow [11,22] to set ω_{adv}, ω_{l1} and ω_{percep} as 1, 100, and 100 respectively. The λ for balancing the content and style items in the perceptual loss function is empirically set to 10. The batch size is set to 1. We employ Adam optimizer which has a momentum of 0.5 and a step size of 0.0002 to train PerGAN. The training converges at about 180 K iterations, which takes ~50 h using a 4.0 GHz Intel i7 processor and a GTX1080 GPU with 8 GB RAM.

4.3. Results

Metrics. We employ four widely-accepted metrics, namely precision (P), recall (R), F-measure score (F_β), and Mean Absolute Error (MAE), to quantitatively evaluate the proposed PerGAN. For each metric, its final measurement is obtained by averaging results across 256 different thresholds (ranging from 0 to 255). Specifically, precision (P) measures the accuracy of positive predictions, while recall (R) measures the completeness of positive predictions, and are formulated as follows:

$$P = \frac{PSR \cap GTR}{PSR} \tag{8}$$

$$R = \frac{PSR \cap GTR}{GTR} \tag{9}$$

where *PSR* represents the predicted salient region and *GTR* represents the ground-truth salient region. Mean Absolute Error (MAE) instead accounts for the true negative salient score. It calculates the average pixel-wise difference between the predicted saliency map S and the ground-truth saliency map T:

$$MAE = \frac{1}{I^*w}\sum_{i=1}^{I^*w}|S(x_i) - T(x_i)| \tag{10}$$

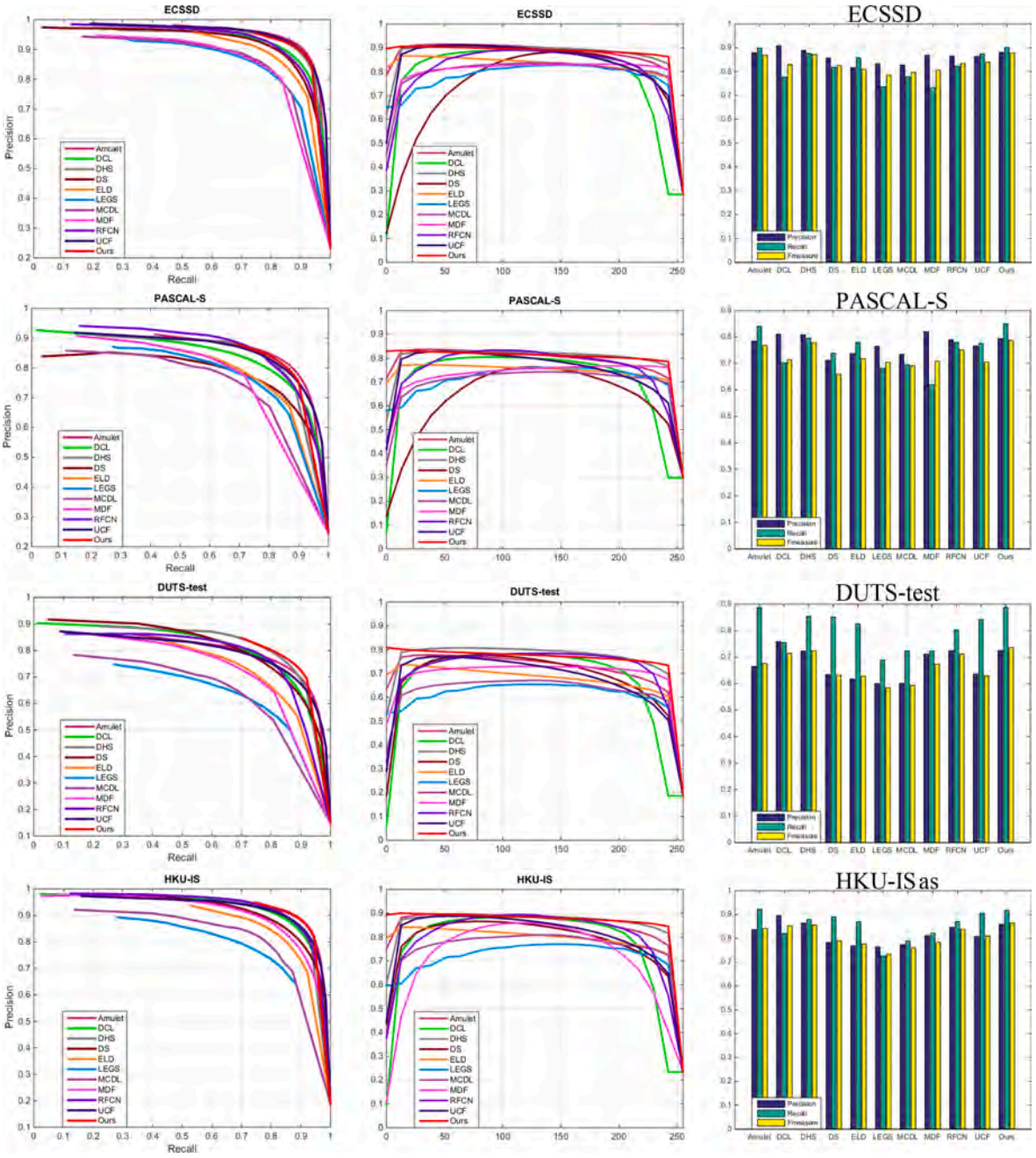

Fig. 4. Comparisons of eleven deep learning based salient object detection approaches covering ours on four publicly datasets. For each dataset, the left and middle columns are the precision-recall curves and F-measure-threshold curves of different methods respectively. The right column shows the average precision, recall and F-measure scores on four challenging databases. Our (PerGAN) proposed method is comparable to all databases under all measurements.

Here, l and w are the length and width of the saliency map, respectively. Both S and T are normalized to [0, 1]. F-measure, considering both precision and recall, is for assessing the overall saliency detection performance:

$$F_\beta = \frac{(1+\beta^2)*P*R}{\beta^2*P+R} \tag{11}$$

As in Ref. [32], we set$\beta^2 = 0.3$ to highlight the precision more.

Table 1
Comparison of quantitative MAE (closer to zero is better), F-measure (large is better) and model size. The top three results are marked with red, green and blue.

Methods	ECSSD		PASCAL-S		DUTS-test		DUT-OMRON		HKU-IS		THUR15K		Model size(MB)
	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	
Amulet [15]	0.061	0.869	0.100	0.763	0.085	0.678	0.098	0.647	0.052	0.839	0.094	0.670	132.6
BL[47]	0.217	0.684	0.249	0.574	0.238	0.490	0.239	0.499	0.207	0.660	0.219	0.530	-
CBGAN[8]	0.089	0.858	0.176	0.686	-	-	0.113	0.689	0.092	0.859	-	-	-
CF [48]	0.115	0.692	0.192	0.037			0.112	0.606					-
DCL[38]	0.151	0.827	0.181	0.714	0.149	0.714	0.097	0.657	0.136	0.853	0.161	0.676	265
DHS [39]	0.059	0.871	0.095	0.773	0.067	0.724	-	-	0.054	0.852	0.082	0.673	376.2
DMCN [40]	0.054	0.867	0.223	0.634			0.091	0.686	0.057	0.858			-
DRFI [49]	0.166	0.733	0.207	0.618	0.175	0.541	0.138	0.550	0.145	0.722	0.150	0.576	-
DS [41]	0.124	0.826	0.176	0.659	0.091	0.632	0.120	0.603	0.078	0.785	0.116	0.626	537.1
ELD [12]	0.082	0.810	0.123	0.718	0.093	0.628	0.092	0.611	0.074	0.769	0.098	0.634	667.2
KSR[42]	0.135	0.782	0.157	0.704	0.121	0.602	0.131	0.591	0.120	0.747	0.123	0.604	-
LAWS[43]	0.088	0.831	0.119	0.741	0.084	0.628	0.093	0.634	0.067	0.821	0.088	0.684	-
LEGS [44]	0.119	0.785	0.155	0.697	0.138	0.585	0.133	0.592	0.119	0.723	0.125	0.607	73.6
MCDL [36]	0.102	0.796	0.145	0.691	0.105	0.594	0.089	0.625	0.092	0.757	0.103	0.620	233.1
MDF [32]	0.108	0.805	0.146	0.709	0.100	0.673	0.092	0.644	-	-	0.109	0.636	330.8
MSNSD [45]	0.171	0.777	0.151	0.792			0.109	0.688	0.071	0.837			-
RFCN1 [7]	0.109	0.834	0.133	0.751	0.090	0.712	0.111	0.627	0.089	0.835	0.100	0.695	1126.4
RFCN2[16]	0.067	0.871	0.105	0.778	-	-	-	-	0.055	0.856	-	-	1126.4
SCNN [46]	0.118	0.797							0.079	0.826			-
UCF [37]	0.080	0.841	0.127	0.701	0.117	0.629	0.132	0.613	0.074	0.808	0.112	0.645	117.9
Ours	0.052	0.878	0.091	0.782	0.064	0.732	0.086	0.677	0.041	0.864	0.081	0.687	88

4.3.1. *Comparison with state-of-the-arts*

Utilizing the aforementioned datasets and metrics, we compare our PerGAN against more than ten state-of-the-art salient object detection methods, including VGG-16 based methods such as Amulet [15], DCL [38], DHS [39], DMCN [40], DS [41], ELD [12], KSR [42], LAWS [43], RFCN1 [7], RFCN2 [16], MSNSD [45], UCF [37], SSCN [46], deep multi-scale methods like BL [47], CF [48], DRFI [49], MCDL [36], MDF [32], and a GAN-based method – CBGAN [8]. To ensure fairness, the saliency maps for comparison are directly acquired from the corresponding author or generated by the officially-released code using the same running setup as reported in the paper. Fig. 3 exhibits some typical visual comparisons. As shown in the figure, PerGAN achieves a much higher completeness on both the salient object's internal region and its boundary, while having lower misdetection rates on the background area. For instance, in the fifth row of Fig. 3, PerGAN extracts the jumping man and the ball in front of the white wall more intactly. We guess this is mainly due to the use of a pre-trained VGG-16 for feature extraction. Recalling Fig. 2, the feature maps extracted from the specified layers of VGG- 16 include both low-level visual cues such as edges and high-level semantic information, which is supposed to facilitate the subsequent saliency map generation.

For quantitative comparison, we plot the methods' P-R curves, F-measure curves, and mean precision, recall, F-measure scores in Fig. 4. The figure indicates an intuitive superiority of the proposed PerGAN against the other methods on almost all four datasets and all evaluation metrics. To gain an insight on how much PerGAN quantitatively outperforms the forefront, we report the methods' MAE and mean F-measure values in Table 1. It shows that PerGAN ranks first on most (9/12) dataset-measurement combinations. Specifically, in terms of the F-measure score, PerGAN outperforms the second-best method by 0.8%, 1.1%, 5.8% on ECSSD, DUTS-test and HKU-IS respectively. For MAE which is the lower the better, it has a decline of 3.7%, 4.2%, 4.5%, 6.5%, 21.1%, 1.2% on ECSSD, PASCAL-S, DUTS-test, DUT-OMRON, HKU-IS and THUR15K respectively when comparing to the runner-up. What's more, our detection accuracy improvements are not accompanied by an increased model complexity as we often see in many previous approaches. In contrary, the model of PerGAN is quite lightweight among those of the SOTA methods reported in this paper. For example, its size (88 Mb) is only about a quarter of that (376.2 Mb) of the DHS which has shown competitive performance on almost all six benchmark datasets.

4.3.2. *The effectiveness of perceptual loss*

Theoretical analysis and empirical evidence indicate that high-level features extracted from a proficiently-trained image classification network possess the potential to encapsulate perceptual information present in real-world images [28]. By minimizing differences between these high-level features, the performance of diverse image-to-image transformation tasks is notably improved

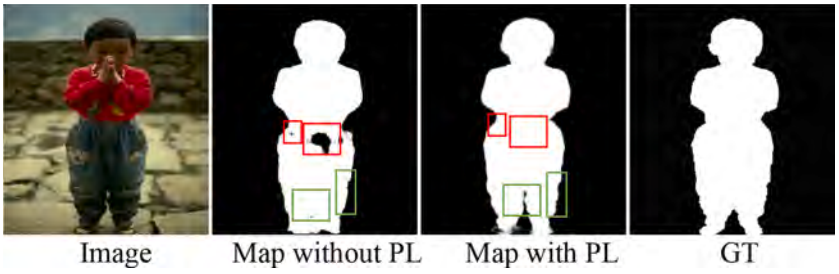

Fig. 5. Comparison between visual results generated by guiding with and without the perceptual loss (PL).

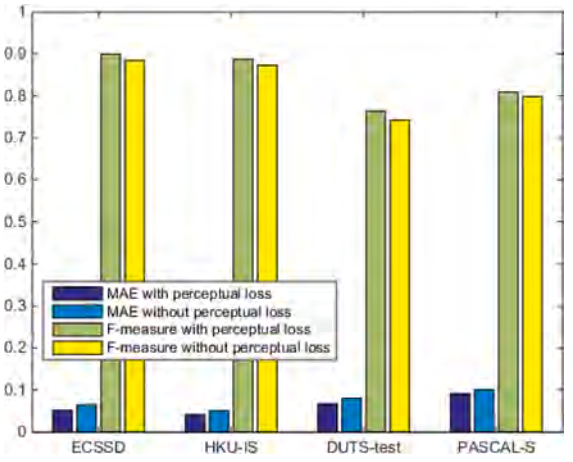

Fig. 6. The bar chart of MAE (closer to zero is better) and F-measure (closer to one is better) with and without perception loss calculated from our (PerGAN) method.

Table 2
The F-measure and MAE on salient RGB images and binary saliency maps (SM).

Dataset		ECSSD	PSACAL-S	DUTS-test	DUT-OMRON	HKU-IS
F_β	SM	0.8991	0.8094	0.7633	0.8871	0.7108
	RGB	0.9024	0.8111	0.7767	0.8904	0.7228
MAE	SM	0.0517	0.0909	0.0673	0.0410	0.0908
	RGB	0.0519	0.0909	0.0638	0.0408	0.0855

Table 3
Ablation Study Using Different Components Combinations on ECSSD Dataset.

CGAN	Multi-Discriminator	Fine-Tune	Perceptual loss	MAE	F_β
×				0.108	0.778
×	×			0.0762	0.823
×	×	×		0.0611	0.851
×	×	×	×	**0.0519**	**0.878**

Table 4
The Comparison between multi-scale generator and multi-scale discriminator on two public datasets.

Multi-scale Generator	Multi-scale Discriminator	ECSSD	PASCAL-S	MAE	F_β
×		×		0.0626	0.9104
×			×	0.1010	0.8126
	×	×		0.0519	0.8783
	×		×	0.0907	0.7821

[11,28,29].

In this part, we deeply investigate the effectiveness of the perceptual loss within the proposed saliency detection framework. We train two saliency detection models with or without the perceptual loss, and compare the two models' performance. As illustrated in Fig. 5, using perceptual loss yields a saliency map that preserves the object's shape more completely and captures intricate edge details. Fig. 6 shows consistent outcomes when evaluating the two models with MAE and F-measure score. Our experiment also reveals that using the RGB image masked with the binary saliency map rather than the binary saliency map itself for perceptual loss calculation leads to an improved saliency detection model. The comparisons can be seen in Table 2, which shows the model trained with the perceptual loss on RGB saliency maps outputs better F-measure and MAE scores in most testing cases, although the improvement looks unexceptional. There are two-level implications here: 1) the perceptual loss exploits not just the object's shape information but its texture and color cues which are not available in binary maps for more accurate saliency detection; 2) the object's shape is pivotal in determining its saliency.

Table 5
The complexity analysis of PerGAN. (For the first Formulation, L, M, K and C means layers, the size of feature map, the size of kernel and the channel separately. For the second one, E, D, B and T stands for the epochs, the size of training dataset, batch size and the time computation complexity of an iteration).

Element	Formulation		Computation Complexity			
			VGG16-based		VGG19-based	
	Time	Space	Time	Space	Time	Space
Generator	$O(\sum_{l=1}^{L} M_l^2 \bullet K_l^2 \bullet C_{l-1} \bullet C_l)$	$O(\sum_{l=1}^{L} K_l^2 \bullet C_{l-1} \bullet C_l)$	O(2.2*1e10)	O(29 M)	O(2.6*1e10)	O(34 M)
Discriminator			O(2.3*1e9)	O(2.4 M)	O(2.3*1e9)	O(2.4 M)
Perceptual Loss			O(1.1*1e10)	–	O(1.5*1e10)	–
PerGAN	$O(E*D/B*T)$		O(4.0*1e15)	O(31.4 M)	O(4.9*1e15)	O(36.4 M)

4.3.3. Ablation study

We conduct an ablation study to investigate the importance of each principal component in the proposed saliency detection method. As shown in Table 3, optimal performance is achieved after we incorporate all components, namely the multi-scale discriminator, saliency map fine-tune and the perceptual loss. The results underline the component's indispensable role in the proposed method. From Table 3, we also notice that the multi-scale discriminator makes a significant contribution to the performance improvement. Considering the difficulty in GAN training which involves learning two competing neural networks, we raise a question that whether we can discard the multi-scale discriminator and train only a multi-scale generator instead. We thus train saliency detection models using only a multi-scale generator and evaluate them on two public datasets - ECSSD and PASCAL-S. The results (see Table 4) show that the multi-scale generator model outperforms the multi-scale discriminator model on F-measure, however is inferior in terms of MAE. It suggests that, without the supervision of a discriminator, the multi-scale generator is prone to be overconfident and erroneously identify the background as salient objects, hence obtaining a large MAE whereas the F-measure is high. This indicates an interesting direction of leveraging a multi-scale generator for accurate saliency detection, while preventing the generator from overconfident.

4.4. Complexity analysis

We provide an informative view on the computational complexity of the proposed PerGAN in Table 5. For a rigorous complexity analysis, we take all the factors including the number of convolutional layers, the size of kernel, and the number of feature channels into calculation. As shown in the table, compared with the general GAN, PerGAN has a bigger model size and a higher computational cost, which is mainly due to the introduction of the multi-scale discriminator and the perceptual loss during training. Although training the model incurs a moderate time and space investment, the inference speed of a VGG-16-based model is about 25fps, which is commendable. The choice of taking VGG16 as the backbone of PerGAN is for balancing between the model's accuracy and complexity, since employing a more competent backbone such as VGG-19 normally at the cost of efficiency (see Table 5).

5. Conclusion

In this paper, we propose a perceptual loss guided GAN – PerGAN for robust salient object detection in images. PerGAN is mainly featured with a perceptual loss function that utilizes high-level semantic cues such as the object's shape and texture to facilitate saliency learning. At the end of PerGAN's generator, there are two fully convolutional layers using 1×1 kernels, enabling a coarse-to-fine generation of saliency map. To boost the training of the generator, we apply a multi-scale discriminator across different image resolutions for a stronger penalization on saliency misdetections. After benchmarking on several mainstream saliency datasets and metrics, PerGAN exhibits highly competitive performance against the-state-of-the-art. We also provide an in-depth ablation study and a complexity analysis on PerGAN, to give readers an insight into the method's key components and its efficiency.

CRediT authorship contribution statement

Xiaoxu Cai: Writing – original draft, Conceptualization, Methodology, Writing – review & editing. **Gaige Wang:** Writing – review & editing. **Jianwen Lou:** Writing – review & editing, Methodology. **Muwei Jian:** Writing – review & editing. **Junyu Dong:** Formal analysis, Validation. **Rung-Ching Chen:** Writing – review & editing. **Brett Stevens:** Writing – review & editing. **Hui Yu:** Supervision, Writing – review & editing, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by Royal Society-K. C. Wong International Fellowship (NIF\R1\180909), EPSRC Grant (EP/N025849/1), National Natural Science Foundation of China (61601427, 61976123), Mount Tai Youth Talents Plan and MOST-107-2221-E-324-018-MY2.

Appendix A

The detail architecture of generator and discriminator networks.

Generator Network	
Input	RGB Image(224,224,3)
[Layer 1]	Conv.(3, 3, 64); Stride = 1; Batchnorm;
[Layer 2]	Conv.(3, 3, 128); Stride = 1; Batchnorm;
[Layer 3]	Conv.(3, 3, 256); Stride = 1; Batchnorm;
[Layer 4]	Conv.(3, 3, 512); Stride = 1; Batchnorm;
[Layer 5]	MaxPool();
[Layer 6]	*ReLU*; Conv.(5, 5, 512); Stride = 2; Batchnorm;
Concat (Layer 4, Layer 6)	
[Layer 7]	*ReLU*; Conv.(5, 5, 256); Stride = 2; Batchnorm;
Concat (Layer 3, Layer 7)	
[Layer 8]	*ReLU*; Conv.(5, 5, 128); Stride = 2; Batchnorm;
Concat(Layer 2, Layer 8)	
[Layer 9]	*ReLU*; Conv.(5, 5, 64); Stride = 2; Batchnorm;
Concat(Layer 1, Layer 9)	
[Layer10]	*ReLU*; Conv.(5, 5, 64); Stride = 2; Batchnorm;
[Layer 11]	*ReLU*; Conv.(1, 1, 32); Stride = 2; Batchnorm;
[Layer 12]	*ReLU*; Conv.(1, 1, 1); Stride = 1; Batchnorm; *Sigmoid;*
Output	Saliency Map224, 224, 1
Discriminator Network	
Discriminator 1	
Input	Image Pair (224, 224, 4)
[Layer 1]	Conv.(5, 5, 64); Stride = 2; *LReLU;*
[Layer 2]	Conv.(5, 5, 128); Stride = 2; Batchnorm; *LReLU;*
[Layer 3]	Conv.(5, 5, 256); Stride = 2; Batchnorm; *LReLU;*
[Layer 4]	Conv.(5, 5, 512); Stride = 1; Batchnorm; *LReLU;*
[Layer 5]	Fully Connected(1); *Tanh*;
Output	Fake/Real
Discriminator 2	
Input	Image Pair (112, 112, 4)
[Layer 1]	Conv.(5, 5, 64); Stride = 2; *LReLU;*
[Layer 2]	Conv.(5, 5, 128); Stride = 2; Batchnorm; *LReLU;*
[Layer 3]	Conv.(3, 3, 256); Stride = 2; Batchnorm; *LReLU;*
[Layer 4]	Fully Connected(1); *Tanh*;
Output	Fake/Real
Discriminator 3	
Input	Image Pair (56, 56, 4)
[Layer 1]	Conv.(5, 5, 64); Stride = 2; *LReLU*
[Layer 2]	Conv.(5, 5, 128); Stride = 2; Batchnorm; *LReLU;*
[Layer 3]	Fully Connected(1); *Tanh;*
Output	Fake/Real

Abbreviation	Full Name
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
EMD	Earth Mover Distance
KL	Kullback–Leibler (KL) divergence
MSE	Mean Square Error
SIM	Similarity Metric
PerGAN	perceptual loss guided GAN
MLP	multi-layer perceptron

(*continued on next page*)

(*continued*)

Abbreviation	Full Name
FCN	fully convolutional network
PSR	predicted salient regions
GTR	groundtruth salient regions

Variable	Explanations
D_k	k^{th} discriminator
G	Generator
x	feeding/RGB image
y	salient object map
T	target saliency map
S	predicted saliency map
c	the output's channels
l	the output's length
w	the output's width
$V()$	represents the non-linear transformation
GM	Gram matrice
L_{name}	A kind of loss, for example, L_{adv} means the adversarial loss
ω_{loss}	The weight of loss, for example,ω_{percep} means the weight of perceptual loss

References

[1] W. Gao, S. Fan, G. Li, W. Lin, A Thorough benchmark and a new model for light field saliency detection, IEEE Transactions on Pattern Analysis and Machine Intelligence (2023).

[2] Y. Wang, W. Zhang, L. Wang, T. Liu, and H. Lu. Multi-source uncertainty mining for deep unsupervised saliency detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11727–11736.

[3] M.M. Cheng, N.J. Mitra, X. Huang, P.H.S. Torr, S.M. Hu, Global contrast based salient region detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (3) (2015) 569–582.

[4] M. Jian, H. Yu, Towards reliable object representation via sparse directional patches and spatial center cues, Fundamental Research (2023).

[5] Q. Liu, X. Hong, B. Zou, J. Chen, Z. Chen, G. Zhao, Hierarchical contour closure-based holistic salient object detection, IEEE Transactions on Image Processing 26 (9) (2017) 4537–4552.

[6] P. Zhang, D. Wang, H. Lu, H. Wang, B. Yin, Learning uncertain convolutional features for accurate saliency detection, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 212–221.

[7] L. Wang, L. Wang, H. Lu, P. Zhang, X. Ruan, Saliency detection with recurrent fully convolutional networks, in: European Conference on Computer Vision (ECCV), 2016, pp. 825–841.

[8] C. Zhang, F. Yang, G. Qiu, Q. Zhang, Salient object detection with capsule-based conditional generative adversarial network, in: 2019 IEEE International Conference on Image Processing (ICIP), 2019.

[9] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: Proceedings of International Conference on Machine Learning (ICML), 2017, pp. 214–223.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, X.u. Bing, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Communications of the ACM 63 (11) (2020) 139–144.

[11] J. Johnson, A. Alahi, F.F. Li, Perceptual losses for real-time style transfer and super-resolution, in: European conference on computer vision (ECCV), 2016, pp. 694–711.

[12] G. Lee, Y.W. Tai, J. Ki, ELD-net: An efficient deep learning architecture for accurate saliency detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (7) (2017) 1599–1610.

[13] X. Zhou, W. Cao, H. Gao, Z. Ming, J. Zhang, STI-Net: Spatiotemporal integration network for video saliency detection, Information Sciences 628 (2023) 134–147.

[14] L. Wei, G. Zong, EGA-Net: Edge feature enhancement and global information attention network for RGB-D salient object detection, Information Sciences 626 (2023) 223–248.

[15] P. Zhang, D. Wang, H. Lu, H. Wang, X. Ruan, Amulet: Aggregating multi-level convolutional features for salient object detection, in: Proceedings of the IEEE International Conference on Computer Vision (CVPR), 2017, pp. 202–211.

[16] L. Wang, L. Wang, H. Lu, P. Zhang, X. Ruan, Salient object detection with recurrent fully convolutional networks, IEEE Transactions on Pattern Analysis and Machine Intelligence. 41 (7) (2018) 1734–1746.

[17] T. Zhao, X. Wu, Pyramid Feature Attention Network for Saliency Detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3085–3094.

[18] Q. Wang, J. Lin, Y. Yuan, Salient band selection for hyperspectral image classification via manifold ranking, IEEE Transactions on Neural Networks and Learning Systems. 27 (6) (2016) 1279–1289.

[19] Y. Liu, Q. Zhang, D. Zhang, J. Han. Employing deep part-object relationships for salient object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision(ICCV), 2019, pp. 1232–1241.

[20] S. Song, Z. Jia, J. Yang, N. Kasabov, Salient detection via the fusion of background-based and multiscale frequency-domain features, Information Sciences 618 (2022) 53–71.

[21] D. Zhang, J. Han, Y. Zhang, D. Xu, Synthesizing supervision for learning deep saliency network without human annotation, IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (7) (2019) 1755–1769.

[22] P. Isola, J. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1125–1134.

[23] H. Pan, X. Niu, R. Li, S. Shen, Y. Dou, Supervised adversarial networks for image saliency detection, International Society for Optics and Photonics 11373 (2020) 113730H.

[24] X. Cai, and H. Yu, "Saliency detection by conditional generative adversarial network" International Society for Optics and Photonics, 10615(2018) 1061541.

[25] J. Wu, X. Liu, Q. Lu, Z. Lin, N. Qin, Q. Shi, FW-GAN: Underwater image enhancement using generative adversarial network with multi-scale fusion, Signal Processing: Image Communication 109 (2022), 116855.

[26] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, F. Durand, What do different evaluation metrics tell us about saliency models, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (3) (2019) 740–757.
[27] S. Jetley, N. Murray, E. Vig, End-to-end saliency mapping via probability distribution prediction, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 5753–5761.
[28] C. Wang, C. Xu, C. Wang, D. Tao, Perceptual adversarial networks for image-to-image transformation, IEEE Transactions on Image Processing. 27 (8) (2018) 4066–4079.
[29] F. Zhan, Y.Yu, K. Cui, G. Zhang, S. Lu, J. Pan, C. Zhang, F. Ma, X.Xie, C. Miao, Unbalanced feature transport for exemplar-based image translation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 15028–15038.
[30] Q. Yan, L. Xu, J. Shi, J. Jia, Hierarchical saliency detection, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 1155–1162.
[31] Y. Li, X. Hou, C. Koch, J.M. Rehg, A.L. Yuille, The secrets of salient object segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 280–287.
[32] G. Li, Y. Yu, Visual saliency based on multiscale deep features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5455–5463.
[33] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, X. Ruan, Learning to detect salient objects with image-level supervision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 136–145.
[34] C. Yang, L. Zhang, H. Lu, X. Ruan, M.H. Yang, Saliency detection via graph-based manifold ranking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 3166–3173.
[35] M. Cheng, N.J. Mitra, X. Huang, S.M. Hu, Salientshape: group saliency in image collections, The Visual Computer. 30 (4) (2014) 443–453.
[36] R. Zhao, W. Ouyang, H. Li, X. Wang, Saliency detection by multi-context deep learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1265–1274.
[37] P. Zhang, D. Wang, H. Lu, H. Wang, B. Yin, Learning uncertain convolutional features for accurate saliency detection, in: Proceedings of the IEEE International Conference on Computer Vision (CVPR), 2017, pp. 212–221.
[38] G. Li, Y. Yu, Deep contrast learning for salient object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 478–487.
[39] N. Liu, J. Han, Dhsnet: Deep hierarchical saliency network for salient object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 678–686.
[40] D. Sun, H. Wu, Z. Ding, S. Li, B. Luo, Salient object detection based on deep multi-level cascade network, in: International Conference on Brain Inspired Cognitive Systems, 2019, pp. 86–95.
[41] X.i. Li, L. Zhao, L. Wei, M.-H. Yang, F. Wu, Y. Zhuang, H. Ling, J. Wang, Deepsaliency: Multi-task deep neural network model for salient object detection, IEEE Transactions on Image Processing 25 (8) (2016) 3919–3930.
[42] T. Wang, L. Zhang, H. Lu, C. Sun, J. Qi, Kernelized subspace ranking for saliency detection, European Conference on Computer Vision (ECCV) (2016) 450–466.
[43] M. Qian, J. Qi, L. Zhang, M. Feng, H. Lu, Language-aware weak supervision for salient object detection, Pattern Recognition 96 (2019) 106955.
[44] L. Wang, H. Lu, X. Ruan, M.-H. Yang, Deep networks for saliency detection via local estimation and global search, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3183–3192.
[45] Y. Liang, H. Liu, N. Ma, A novel deep network and aggregation model for saliency detection, The Visual Computer 36 (9) (2019) 1–13.
[46] F. Cao, Y. Liu, D. Wang, Efficient saliency detection using convolutional neural networks with feature selection, Information Sciences 456 (2018) 34–49.
[47] N. Tong, H. Lu, X. Ruan, M.-H. Yang, Salient object detection via bootstrap learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1884–1892.
[48] M.-U. Hassan, D. Niu, X. Zhao, M.-S. Ahamed Shohag, Y. Ma, M. Zhang, Salient object detection based on cnn fusion of two types of saliency models, in: IEEE 2019 International Conference on Image and Vision Computing New Zealand (IVCNZ), 2019, pp. 1–6.
[49] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, S. Li, in: SAlient Object Detection: A DiscriminAtive RegionAl FeAture IntegrAtion ApproAch, 2013, pp. 2083–2090.