

RESEARCH

Open Access



Are RGB-based salient object detection methods unsuitable for light field data?

Yu Liu¹, Huaxin Xiao^{1*} , Hanlin Tan¹ and Ping Li²

*Correspondence:

huaxinxiao@hotmail.com

¹Department of Systems Engineering, National University of Defense Technology, Deya Road, Changsha 410073, China
Full list of author information is available at the end of the article

Abstract

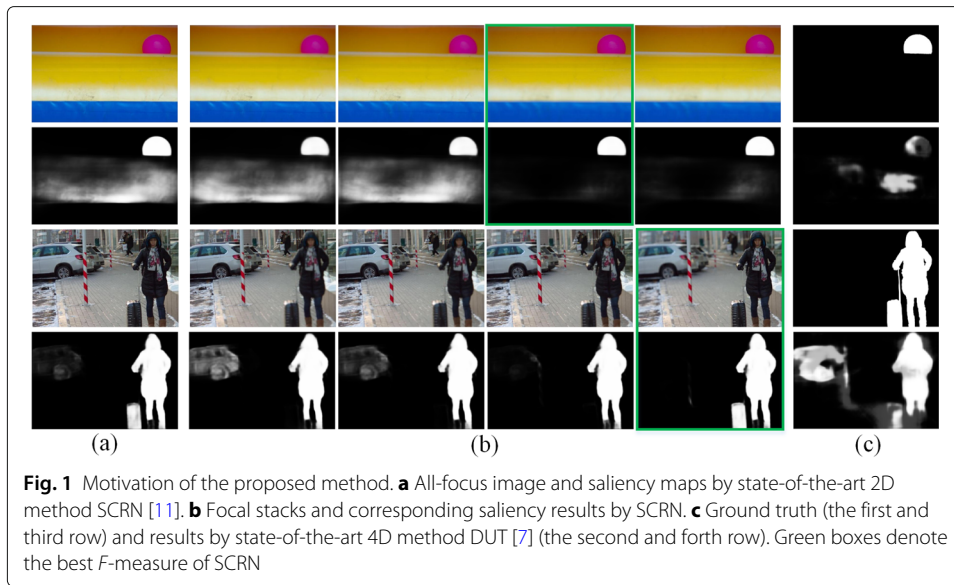
Considering the significant progress made on RGB-based deep salient object detection (SOD) methods, this paper seeks to bridge the gap between those 2D methods and 4D light field data, instead of implementing specific 4D methods. We observe that the performance of 2D methods changes dramatically with the input refocusing on different depths. This paper attempts to make the 2D methods available for light field SOD by learning to select the best single image from the 4D tensor. Given a 2D method, a deep model is proposed to explicitly compare pairs of SOD results on one light field sample. Moreover, a comparator module is designed to integrate the features from a pair, which provides more discriminative representations to classify. Experiments over 13 latest 2D methods and 2 datasets demonstrate the proposed method can bring about 24.0% and 5.3% average improvement of mean absolute error and *F*-measure, and outperform state-of-the-art 4D methods by a large margin.

Keywords: Salient object detection, Light field, Learning to compare, Attention network

1 Introduction

Salient object detection (SOD), also known as saliency detection, refers to simulate visual attention processes of human vision systems, which helps humans to quickly understand visual scenes and filter irrelevant information. Given a natural image, SOD aims to localize and segment the most visually attractive objects and ignore other region. Such technology has been regarded as a fundamental step for various computer vision problems, such as tracking [1], image fusion [2], detection [3], and segmentation [4, 5].

Usually, existing SOD methods can be categorized into 2D, 3D, and 4D, which depends on the types of input, i.e., RGB, RGBD, and light field data, respectively. 2D methods have attracted a lot of attention from the community and dominated the field. In the past few years, the evolution of handheld light field cameras makes the acquisition of 4D light field data much easier, which promotes the development of light field SOD [6–9]. The commercial light field camera Lytro [10] adopts a microlens array, consisting of thousands of tiny lenses to measure light from multiple directions. Through rendering and post-shot refocusing techniques, the Lytro camera can synthesize different types of 2D images, including focal stacks, depth maps, and all-focus images [7]. In the task of light field SOD,



the 4D light field representation is converted into one all-focus image (as shown in the first row of Fig. 1a) and multiple focal stacks at different depths (as shown in the first row of Fig. 1b).

State-of-the-art 4D methods [7–9] can be regarded as fusion-based methods, which attempt to combine all light field slices by different strategies. However, the slices of one light field sample¹ are not always beneficial for saliency detection. As shown in Fig. 1b, the variation of focus depth may make the original salient object blurred and the undesired background noticeable. Fusing such feature could lead to inferior results. As shown in the second and fourth row of Fig. 1c, fusion-based method DUT [7] detects much false positive region, which could be attributed to the combination of some undesired focal stacks.

The target of this paper is to provide a new perspective to solve 4D SOD. We find that using different input data for 2D methods changes performance dramatically, as shown in Fig. 1b. Some focal stacks happen to correctly focus on the target object and blur the background, which makes the segmentation far easier and better than the other stacks and all-focus image. If we can automatically select proper slice(s) from the light field sample, can 2D methods outperform state-of-the-art 4D ones? To the best of our knowledge, no prior work has attempted to select the input of 2D methods on the task of light field SOD.

To this end, we reformulate the task of light field SOD by selecting the best input of 2D methods, which makes this task continuously benefit from the development of 2D methods. Concretely, this paper proposes to explicitly compare pairs of light field slices. The input of our model is two images randomly selected from the same light field sample. For evaluating a certain 2D method, we also concatenate the two images with their corresponding saliency maps. However, such paired inputs could bring a new challenge that same two images with different order should correspond to different outputs. A comparator module is proposed to ensure the model has the ability of distinguishing which of

¹In this paper, a light field sample is defined as the all-focus image plus the corresponding focal stacks. For simplicity, a slice of light field sample can be either an all-focus image or an image for focal stacks.

the two inputs is better. The features of two inputs are treated separately as the exemplar and query of the comparator module. The comparator adopts the attention mechanism to reweigh the query feature based on the correlation between the query and exemplar, which allows the whole model to pay more attention to the correlated areas and provide more discriminative representation. To train such a model, the label is formulated as the relative performance of the two SOD results, which can be measured by common SOD metrics, e.g., F -measure [12] and E -measure [13]. Instead of regressing the relative performance, we adopt the binary classification loss, which intends to make the learning process easier and more effective.

Once trained, the model can operate as a bubble sorting algorithm. The model can iteratively compare two slices from the same light field sample until the best one is predicted. However, such testing strategy is highly unstable and easy to fail in the process of prediction if an error is injected. This paper considers the prediction from a global perspective. The score of one slice should be simultaneously determined by the others in the same sample. Thus, at each time, we compare one slice with the others and adopt the average score as the prediction of the target slice. In this way, the strategy brings more information from a global view and offsets any inaccurate predictions by the model.

To sum up, the main contributions of this work are threefold:

- This paper demonstrates that there exists an alternative way to perform light field SOD without designing specialized 4D methods, i.e., optimizing and selecting the best input to existing 2D methods.
- A novel convolutional neural network (CNN)-based model is proposed to compare any two slices from the same light field sample, the relative performance of which is regularized by an effective attention-based module and a simple binary classification loss.
- We verify the proposed model with 13 latest 2D deep SOD methods over 2 light field datasets. The experimental results demonstrate that the proposed method effectively boosts the performances and makes most of the involved 2D methods outperform state-of-the-art 4D methods.

2 Related work

Here, we briefly introduce the related work about salient object detection according to different types of data.

2.1 RGB-based salient object detection

SOD methods are initially devoted to single color or grayscale image, which can be also regarded as 2D methods. Early works mainly depend on intrinsic cues, e.g., hand-crafted local and global features [14, 15] and heuristic priors [12, 16], to extract underlying salient regions. In recent years, the introduction of CNN leads the SOD into a new era of rapid development [17]. At the beginning, the CNN is adopted as a feature extractor to provide multi-context information [18, 19]. With the introduction of fully convolutional network (FCN), the CNN-based SOD is formulated as a task of pixel-wise estimation. To provide necessary low- and high-level context, most methods attempt to integrate the features from multi-stages of FCN. For example, Li and Yu [20] proposed

a multi-scale FCN branch to capture visual contrast among multi-scale feature maps. Hou et al. [21] exploited stage-wise supervision to explicitly learn on multi-scale feature maps. In [22], the features from deep and shallow layers of FCN are iteratively optimized using residual refinement blocks. Besides individually learning a segmentation model, some works [23–27] attempt to utilize boundary information for auxiliary training. Li et al. [23] proposed a novel method to alternately train a contour detection model and a SOD model. In [26], logical interrelations were adopted to constrain the simultaneous training of SOD and edge detection. A edge guidance network [27] is employed to couple edge features with saliency features at multi-scales. Usually, attention modules in CNN are exploited to mimic the visual attention mechanism, which is consistent with the purpose of SOD. A pixel-wise contextual attention network [28] was introduced to selectively attend to informative context locations for each pixel. Chen et al. [29] attempted to reverse the attention results for expanding object regions progressively. The work in [30] proposed an attentive feedback module to refine the features from encoder and pass them to the corresponding decoder.

2.2 RGBD-based salient object detection

Depth maps containing various depth cues such as spatial structure and 3D layout provide necessary complementary information for 2D SOD [31, 32]. CNN also shows powerful ability in the field of 3D SOD. Qu et al. [33] fused the depth maps with different low-level saliency cues as the input of CNN. Chen and Li [34] designed a novel complementarity-aware fusion module to explicitly integrate cross-modal and cross-level features. In [35], the depth cue was processed by an independent encoder network to provide extra prior. Recently, Zhao et al. [32] argued that fusing the CNN features of depth maps in RGB branch is sub-optimal. Before combining with RGB features, they adopted contrast prior to enhance the depth cues.

2.3 Light field salient object detection

This task was firstly defined in [6] where objectness was adopted to integrate the saliency candidates from all-focus images and corresponding focal stacks. Zhang et al. [36] extracted background priors by weighting focusness contrast and presented effectiveness of light field data properties. Li et al. [37] built a saliency dictionary by selecting a group of salient candidates from the focal stacks, where saliency was measured by the reconstruction error. In [8], multiple cues, e.g., color, depth, and multiple viewpoints, were generated from light field features and integrated by a random-search-based weighting strategy. Compared with 2D/3D methods, CNN-based light field SOD is still on its primary stage. One of the main reasons is in insufficient labeled data. Recently, Wang et al. [7] introduced a large dataset and adopted CNN models to solve this task. A recurrent attention network was proposed in [7] to integrate every slice in the focal stacks and lately combined with another stream over all-focus images. Similar to [7], Zhang et al. [9] proposed a complicated framework to fuse the focal stacks, which aimed at emphasizing the ones related to the salient object. Piao et al. [38] introduced an asymmetrical two-stream network to distill focusness knowledge to a student network, which is computation-friendly. Instead of learning the implicit relationship among the focal stacks, we attempt to explicitly select the slices of light field sample, which are compatible with the well-developed CNN-based 2D methods.

3 Methodology

3.1 Problem definition

We consider solving the task of light field SOD by employing existing CNN-based 2D methods. Formally, a complete light field sample consists of one all-focus image I_0 and multiple focal stacks $\{I_n\}_{n=1}^N$ focusing at different depths. The problem of this paper is how to select the best performance from the saliency maps $\{M_n\}_{n=0}^N$ by a given 2D method w.r.t. $\{I_n\}_{n=0}^N$. The performance $\{y_n\}_{n=0}^N$ of $\{M_n\}_{n=0}^N$ can be quantified by standard evaluation metrics, such as mean absolute error, F -measure [12], E -measure [13], and S -measure [39].

One straightforward solution is to treat individual images in $\{I_n\}_{n=0}^N$ as independent input and learn to regress the quantitative performances $\{y_n\}_{n=0}^N$. When testing, the slice with the maximum predicted score can be selected from each test sample. However, learning such a model may suffer from ambiguous data, where similar quantitative values may correspond to totally different SOD results. Besides, existing light field SOD datasets are quite small. The largest dataset DUTLF [7] only provides 1000 training samples with 7354 focal stacks, which is far from enough to train a regression model.

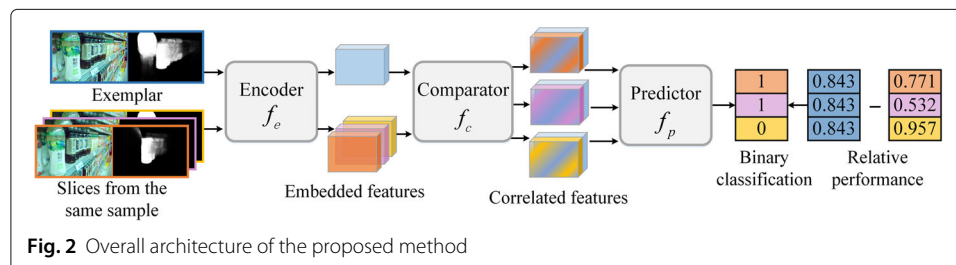
To achieve satisfied performance with limited data, this paper proposes to explicitly compare pairs of light field slices, which naturally augments the training data. Thus, the problem is reformulated as predicting the relative performance $y_i - y_j$ between two different slices I_i and I_j . Such strategy expands the training data by $(N + 1) \times (N)$ times. For example, each sample of DUTLF [7] dataset on average has 8.2 slices, which can provide more than 55,000 training samples in the above definition. Such amount of data is enough to support the training of a powerful model.

3.2 The proposed method

3.2.1 Overview

The proposed method consists of three key modules: an encoder f_e , a comparator f_c , and a predictor f_p , as shown in Fig. 2. At each iteration, we randomly select two slices I_i and I_j from a light field sample $\{I_n\}_{n=0}^N$ as an exemplar and a query, respectively. The goal of the proposed method is to predict whether the exemplar I_i achieves better performance than the query I_j for a given 2D method. As described in above subsection, the relative performance $y_i - y_j$ is calculated on the saliency maps $\{M_i, M_j\}$ by the given 2D method. Thus, we concatenate the input slice with its corresponding saliency map, which guides the model to learn better.

The paired inputs $\{[I_i, M_i], [I_j, M_j]\}$ are first passed to the Siamese network-based encoder f_e and embedded into feature representations $\{\mathbf{F}_i, \mathbf{F}_j\}$. In our definition, similar inputs $\{[I_i, M_i], [I_j, M_j]\}$, and $\{[I_j, M_j], [I_i, M_i]\}$ correspond to totally different



labels $y_i - y_j$ and $y_j - y_i$. To distinguish such similar pairs, we propose a comparator f_c to reweigh the query feature \mathbf{F}_j based on the correlation between the exemplar and query. The comparator adopts co-attention mechanism [40, 41] to couple $\{\mathbf{F}_i, \mathbf{F}_j\}$ and generates the correlated features \mathbf{F}_{ij} . The comparator enables the whole model to attend more to the informative regions and provide more discriminative features to the predictor. The predictor f_p is formulated as a binary classifier and outputs the probability that exemplar I_i is outperforming query I_j . The whole model is trained by a binary cross entropy loss:

$$\mathcal{L} = - \sum_i \sum_{j \neq i} [\mathbf{1}(y_i \geq y_j) \cdot \log \sigma(f_p(\mathbf{F}_{ij})) + \mathbf{1}(y_i < y_j) \cdot \log(1 - \sigma(f_p(\mathbf{F}_{ij})))] \quad (1)$$

where σ is the Sigmoid function and $\mathbf{F}_{ij} = f_c(f_e(I_i, M_i), f_e(I_j, M_j))$. $\mathbf{1}(\cdot) = 1$ if the condition (\cdot) is true. Otherwise, $\mathbf{1}(\cdot) = 0$.

During the testing phase, the trained model can operate as a bubble sorting algorithm. If $\sigma(f_p(\mathbf{F}_{ij})) \geq 0.5$, the exemplar I_i is considered to achieve better performance than query I_j and would be compared with other queries. Otherwise, the query I_j is regarded as a better one. This process of comparison passes forward until all slices in the sample are compared. However, such testing strategy is sub-optimal when the trained model cannot guarantee 100% accuracy. The process of comparison would fail if an error is injected. We consider the prediction from a global perspective. We simultaneously compare one slice with the others and adopt the average score as the final prediction of the target slice. The best input slice is simply determined by the maximum score:

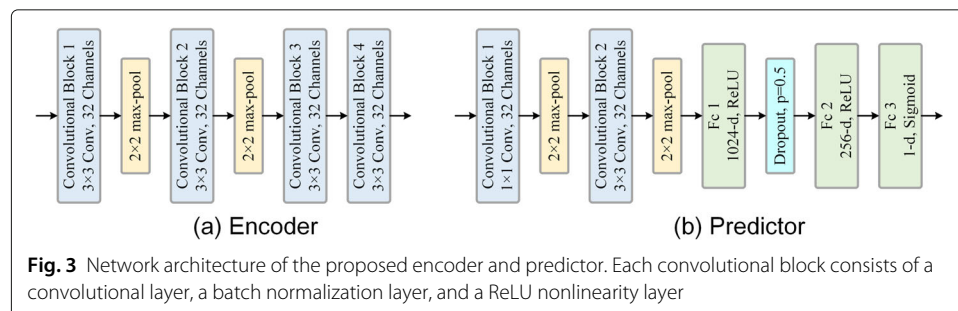
$$i \leftarrow \arg \max_{i=0, \dots, N} \frac{1}{N} \sum_{j=0, j \neq i}^N \sigma(f_p(\mathbf{F}_{ij})) \quad (2)$$

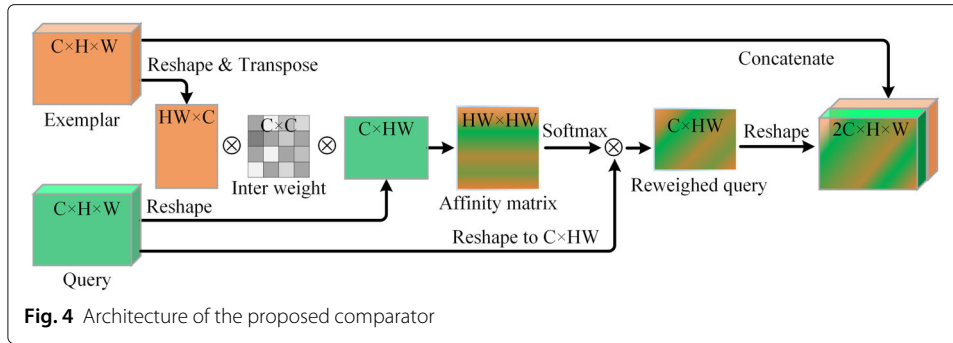
3.2.2 The encoder

The encoder module is a Siamese CNN. It maps a pair of images into the same feature space and provides comparable features. Inspired by the works of few-shot learning [42, 43], we adopt a simple but effective network as shown in Fig. 3a. The encoder consists of four convolutional blocks and two 2×2 max pooling layers. Each convolutional block is a 3×3 convolutional layer with 32 channels, followed by a batch normalization layer and a ReLU nonlinearity activation. The two max pooling layers are deployed after the first two blocks to reduce the spatial size of the features.

3.2.3 The comparator

Although paired input can expand the training data, they would incur a problem of same inputs with different order, e.g., $\{[I_i, M_i], [I_j, M_j]\}$ and $\{[I_j, M_j], [I_i, M_i]\}$. Those inputs should correspond to the completely opposite results. To enable the model to distinguish





such difference, we propose a comparator as shown in Fig. 4 to asymmetrically reweigh the features $\{\mathbf{F}_i \in \mathbb{R}^{H \times W \times C}, \mathbf{F}_j \in \mathbb{R}^{H \times W \times C}\}$ embedded by the encoder. The comparator treats $\{\mathbf{F}_i, \mathbf{F}_j\}$ separately, \mathbf{F}_i as exemplar and \mathbf{F}_j as query. Co-attention mechanism [40, 41] is employed to calculate the affinity matrix \mathbf{A} between \mathbf{F}_i and \mathbf{F}_j :

$$\mathbf{A} = \mathbf{F}_i^T \mathbf{W} \mathbf{F}_j, \quad (3)$$

where $\mathbf{F}_i \in \mathbb{R}^{C \times HW}$ and $\mathbf{F}_j \in \mathbb{R}^{C \times HW}$ are reshaped into matrix form. $\mathbf{W} \in \mathbb{R}^{C \times C}$ denotes the interweight, which can be formulated as a fully connected layer.

Each row in \mathbf{F}_i^T and each column in \mathbf{F}_j both define a C -dimension feature at each spatial position of $H \times W$. Thus, each element in $\mathbf{A} \in \mathbb{R}^{HW \times HW}$ represents an affinity score of features corresponding to pairs of spatial location in \mathbf{F}_i and \mathbf{F}_j . In this paper, we only reweigh the query feature to emphasize the order difference of same input images. Concretely, \mathbf{A} is normalized column-wise to generate attention across the exemplar \mathbf{F}_i for each position in the query \mathbf{F}_j :

$$\bar{\mathbf{A}} = \left[\eta(\mathbf{A}^{(1)}), \dots, \eta(\mathbf{A}^{(c)}), \dots, \eta(\mathbf{A}^{(HW)}) \right] \in \mathbb{R}^{HW \times HW}, \quad (4)$$

where η denotes the softmax normalization and $\mathbf{A}^{(c)}$ denotes the c th column of \mathbf{A} , which reflects the relevance of each feature in \mathbf{F}_i to the c th in \mathbf{F}_j .

Next, we compute the attention contexts of the query in light of the exemplar:

$$\mathbf{F}_j^i = \mathbf{F}_j \bar{\mathbf{A}} \in \mathbb{R}^{C \times HW}, \quad (5)$$

where $\mathbf{F}_j \in \mathbb{R}^{C \times HW}$.

Finally, the output feature $\mathbf{F}_{ij} \in \mathbb{R}^{2C \times H \times W}$ of comparator is formulated as the concatenation of \mathbf{F}_j^i and \mathbf{F}_i . Through above transformation, the order difference is formulated as reweighing the query based on the correlations between exemplar and query. When exchanging the order of i and j , the comparator would output totally different features.

3.2.4 The predictor

Given a calculated feature, the predictor can be regraded as a simple CNN-based classifier to identify the corresponding label. As shown in Fig. 3b, the predictor module consists of two convolutional blocks, two 2×2 max-pooling layers, and three fully connected layers. The convolutional block has the same configuration as the one in the encoder. The first block adopts a 1×1 convolution to reduce the feature dimension. The first fully connected layer is followed by a dropout layer with a probability of 0.5 to avoid overfitting. The dimension of fully connected layers is 1024, 256, and 1, respectively.

4 Experiments

4.1 Experimental setup

4.1.1 Datasets

We evaluate the proposed model on two light field SOD datasets: LFS [6] and DUTLF [7]. LFS provides 100 light field samples by the Lytro light field camera, including 60 indoor and 40 outdoor scenes. This is the first dataset for solving the light field SOD problem. This dataset is captured at a resolution of 360×360 . Then, an all-focus image is composed by using online open-source tools. DUTLF is the latest and largest dataset for improving the development of CNN-based light field SOD. It is a more challenging dataset with a wide range of scenes and multiple salient objects. This dataset is captured by a Lytro Illum camera at a resolution of 600×400 . The DUTLF dataset consists of 1000 training and 465 testing images.

4.1.2 Evaluation metrics

We adopt five metrics for extensive evaluation, including mean absolute error (\mathcal{M}), F -measure [12] (\mathcal{F}), E -measure [13] (\mathcal{E}), S -measure [39] (\mathcal{S}), and precision-recall (PR) curve. These metrics have been widely used to evaluate a SOD method. More detailed explanations are found in [13, 39].

4.1.3 Implementation details

The proposed model is trained end-to-end from scratch with random initialization. All training and testing images are resized to 182×182 . Thus, the input data dimension of the first fully connected layer in predictor is equal to $32 \times 10 \times 10$. The proposed model is learned on the training set of DUTLF, where data is augmented by horizontal flipping. The training label y_i in Eq. (1) is calculated by the E -measure [13]. The network is trained by standard SGD and converges after 30 epochs with batch size of 32. Each entry of the mini-batch consists of two images randomly selected from the same light field sample. The learning rate, momentum, and weight decay of the SGD optimizer are set to $5e-3$, $5e-4$, and 0.9, respectively. The learning rate is set to $5e-4$ after 20 epochs. Our proposed model is implemented by the publicly available Pytorch library. All the experiments and analyses are conducted on a Nvidia 1080Ti GPU.

4.2 Comparisons with state-of-the-arts

To verify the effectiveness of our method, we collect 13 latest 2D SOD methods, including GCPA [44], F3Net [45], SCRNet [11], EGN [27], DUCRF [46], BASNet [24], CPD [26], AFNet [30], PoolNet [47], DGRL [25], BMP [48], C2SNet [23], and RAS [29]. All these CNN-based methods are designed for RGB data and provide public pre-trained model. We also compare with 4 RGBD methods, i.e., DMRA [31], CFP [32], PCA [34], and DFRGBD [33]; 3 CNN-based 4D state-of-the-arts, i.e., DUT [7], MoLF [9], and Piao et al. [38]; and 4 traditional 4D methods, i.e., LFS [49], MCA [8], WSC [37], and DILF [36].

4.2.1 Quantitative comparisons

As shown in Table 1, we present the quantitative scores of \mathcal{M} , \mathcal{F} , \mathcal{E} , and \mathcal{S} . *Baseline* in Table 1 denotes the results of 2D methods over all-focus images, while *+Ours* denotes the results after the selection of our proposed method. Compared with the baseline, our method brings a large improvement, especially on the dataset DUTLF. Concretely, the average improvement of \mathcal{M} , \mathcal{F} , \mathcal{E} , and \mathcal{S} on DUTLF is 29.5%, 5.5%, 5.0%, and 5.3%,

Table 1 Quantitative comparisons of mean absolute error (\mathcal{M}), F-measure (\mathcal{F}), E-measure (\mathcal{E}), and S-measure (\mathcal{S}) on two light field datasets, i.e., DUTLF [7] and LFSD [6]. *Baseline* denotes the results of 2D methods over all-focus images, while *+Ours* denotes the results after the selection of our proposed method. Red and blue denote the best and second scores, respectively. The up arrow \uparrow means larger is better while the down arrow \downarrow means smaller is better

Methods	Venue	Variants	DUTLF				LFSD			
			$\mathcal{M} \downarrow$	$\mathcal{F} \uparrow$	$\mathcal{E} \uparrow$	$\mathcal{S} \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{F} \uparrow$	$\mathcal{E} \uparrow$	$\mathcal{S} \uparrow$
2D methods										
GCPA [44]	AAAI'20	Baseline	.076	.859	.874	.861	.098	.821	.835	.810
		+ Ours	.045	.909	.928	.908	.070	.876	.882	.853
F3Net [45]	AAAI'20	Baseline	.070	.858	.878	.860	.105	.814	.830	.797
		+ Ours	.045	.905	.926	.902	.070	.880	.892	.854
SCRN [11]	ICCV'19	Baseline	.074	.858	.868	.859	.087	.850	.861	.835
		+ Ours	.048	.903	.917	.902	.066	.897	.900	.871
EGNet [27]	ICCV'19	Baseline	.084	.841	.864	.837	.088	.842	.862	.827
		+ Ours	.056	.890	.907	.885	.092	.861	.870	.824
DUCRF [46]	ICCV'19	Baseline	.109	.799	.825	.799	.114	.822	.818	.791
		+ Ours	.079	.849	.870	.844	.089	.874	.873	.838
BASNet [24]	CVPR'19	Baseline	.064	.868	.894	.868	.084	.861	.870	.824
		+ Ours	.048	.897	.924	.895	.078	.874	.888	.839
CPD [26]	CVPR'19	Baseline	.067	.872	.885	.864	.089	.855	.860	.826
		+ Ours	.046	.905	.922	.900	.071	.886	.892	.855
AFNet [30]	CVPR'19	Baseline	.080	.845	.873	.844	.094	.832	.840	.807
		+ Ours	.058	.886	.908	.879	.072	.886	.887	.852
PoolNet [47]	CVPR'19	Baseline	.082	.846	.864	.837	.096	.833	.850	.818
		+ Ours	.058	.893	.905	.884	.091	.849	.876	.829
DGRL [25]	CVPR'18	Baseline	.082	.821	.868	.824	.092	.842	.863	.812
		+ Ours	.060	.867	.906	.864	.074	.871	.891	.845
BMP [48]	CVPR'18	Baseline	.082	.843	.856	.828	.103	.834	.834	.792
		+ Ours	.063	.880	.893	.870	.084	.867	.879	.833
C2SNet [23]	ECCV'18	Baseline	.116	.797	.815	.794	.113	.834	.822	.797
		+ Ours	.087	.836	.864	.838	.097	.861	.866	.828
RAS [29]	ECCV'18	Baseline	.127	.732	.795	.723	.151	.749	.760	.696
		+ Ours	.097	.809	.847	.793	.107	.836	.841	.781
3D methods										
DMRA [31]	ICCV'19	–	.048	.883	.927	.888	.075	.849	.899	.847
CPFP [32]	CVPR'19	–	.101	.730	.808	.741	.186	.524	.669	.599
PCA [34]	CVPR'18	–	.100	.760	.858	.801	.112	.794	.846	.800
DFRGBD [33]	TIP'17	–	.163	.722	.684	.687	.180	.841	.737	.732
4D methods										
Piao et al. [38]	AAAI'20	–	.049	.891	.922	.887	.080	.842	.889	.838
MoLF [9]	NeurIPS'19	–	.059	.887	.909	.873	.089	.853	.874	.820
DUT [7]	ICCV'19	–	.070	.868	.905	.852	.093	.863	.877	.826
LFS [49]	TPAMI'17	–	.259	.439	.545	.517	.168	.779	.625	.655
MCA [8]	TOOM'17	–	–	–	–	–	.150	.815	.841	.749
WSC [37]	CVPR'15	–	–	–	–	–	.156	.706	.794	.706
DILF [36]	IJCAI'15	–	–	–	–	–	.168	.728	.810	.755

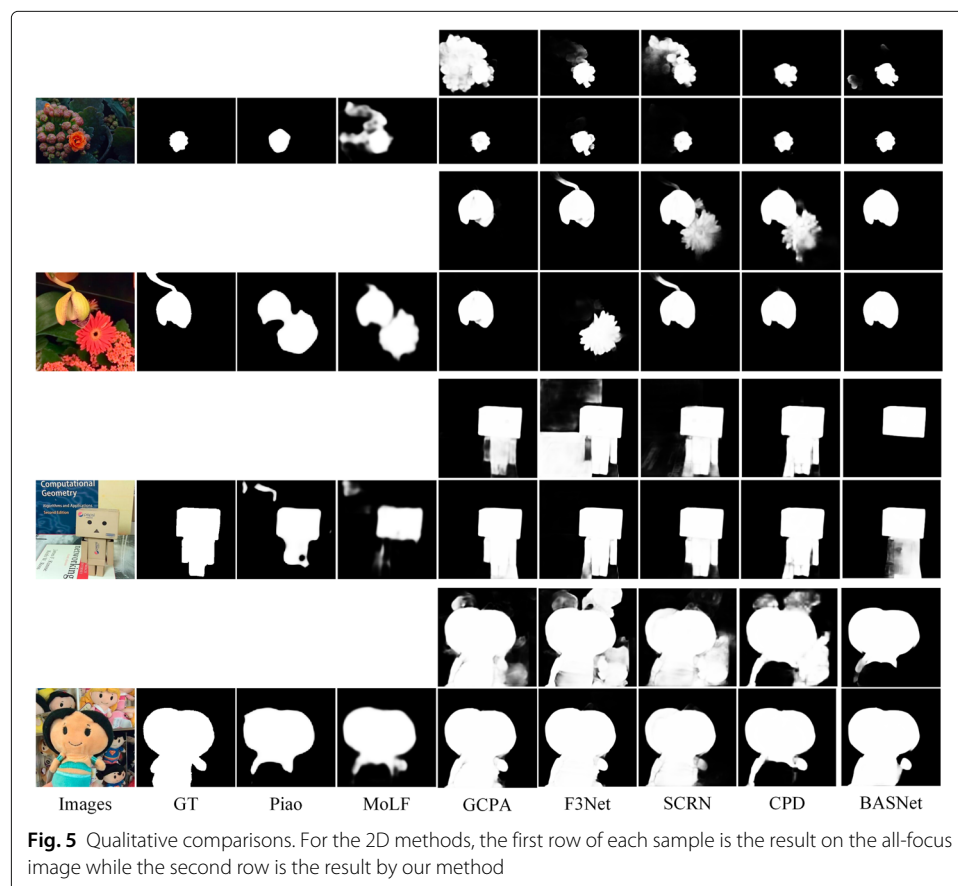
respectively, while the number on LFSD is 18.5%, 5.1%, 5.0%, and 4.6%, respectively. The consistent improvement on different datasets demonstrates that the proposed method is a general strategy for various 2D methods and data. Compared with 4D methods, we observe that CNN-based methods outperform the baselines of all 2D methods. After the refinement of our method, GCPA, F3Net, SCRN, and CPD achieve superior results than custom 4D methods on the dataset DUTLF and LFSD. In summary, combining our method with latest 2D methods provides a new state-of-the-art for the task of light field SOD.

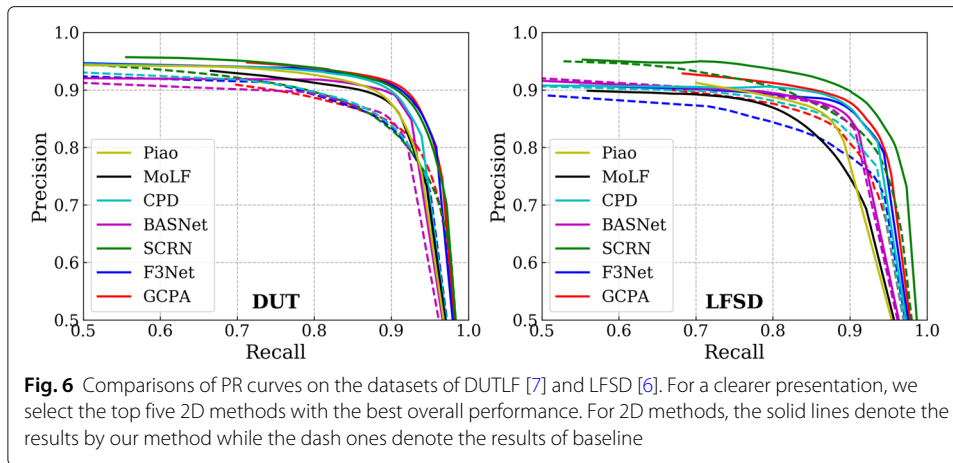
4.2.2 Qualitative comparisons

In Fig. 5, we visually compare five best performing 2D methods, including GCPA, F3Net, SCRN, CPD, and BASNet, and 4D methods Piao et al. and MoLF. For the 2D methods, the first row of each sample is the result on the all-focus image while the second row is the result by our method. At most cases, the proposed method provides a better option than the all-focus image. Compared with state-of-the-art 4D methods, the proposed method effectively suppresses the false positive detection and improves the true positive performance.

4.2.3 Precision-recall curve

In Fig. 6, we compare the PR curves of different methods. For a clearer presentation, we select the top five 2D methods with the best overall performance. For these 2D methods,





the solid lines denote the results by our method while the dash ones denote the results of baseline. With our method, the 2D methods always exceed the baselines.

4.3 Ablation study

In this subsection, we first analyze the effectiveness of the key components and how it benefits the 2D methods. Next, we investigate the generalization of our method.

4.3.1 Key components

In Table 2, we summarize the contributions of key components, including training data, the loss function, comparator, testing strategy, and the architecture of encoder. Compared with the *Baseline*, using every possible slice in the column of *Avg.* always leads to inferior performance, which proves that some slices in one light field sample could bring negative effect. Thus, carefully selecting the proper slice input is necessary. Comparing the rows of *Img. Input* and *Sal. Input*, we find that saliency maps are critical, which provide necessary information about the performance of 2D method and guide the model learning.

To demonstrate that classification is more suitable than regression, we replace the binary cross entropy loss in Eq. (1) with L1 loss:

$$\mathcal{L} = \sum_i \sum_{j \neq i} |(y_i - y_j) - f_p(\mathbf{F}_{ij})|. \quad (6)$$

As shown in the *Reg. Loss* rows of Table 2, learning with the classification loss is always beneficial to the performance.

Next, we use plain concatenated features $[\mathbf{F}_i, \mathbf{F}_j]$ to take the place of the comparator. The results in the *w/o Com.* rows of Table 2 indicate that direct concatenation is inferior, especially in the dataset DUTLF. Finally, we evaluate the effect of testing strategy. We test the trained model as a bubble sorting algorithm. As shown in the rows of *Bub. Test*, proposed Eq. 2 considers one slice from a global perspective and provides more stable and superior prediction. Finally, we analyze the effect of encoder architecture. The column of *R-18* and *SER-18* denote replacing the proposed encoder with well-designed architecture ResNet-18 [50] and SE-ResNet-18 [51], respectively. However, using different architectures cannot bring further improvement. We attribute the reason to the representation embedded by the encoder to be powerful enough. It seems that the limitation of performance is mainly determined by the comparator and predictor.

Table 2 Ablation study of the proposed method. *Baseline* denotes the results of 2D methods over all-focus images while *Avg.* denotes the average results of 2D methods over all slices. *Img. Input* and *Sal. Input* denote the model trained only with RGB images and saliency maps, respectively. *Reg. Loss* and *w/o Com.* denote learning the model with the regression loss and without the proposed comparator, respectively. *Bub. Test* denotes testing the trained model as a bubble sorting algorithm. *R-18* and *SER-18* denote replacing the proposed encoder with well-designed architecture ResNet-18 [50] and SE-ResNet-18 [51], respectively

Dataset	Variant	GCPA		F3Net		SCRN		BASNet		CPD	
		$\mathcal{M} \downarrow$	$\mathcal{E} \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{E} \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{E} \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{E} \uparrow$	$\mathcal{M} \downarrow$	$\mathcal{E} \uparrow$
DUTLF	Ours	.045	.928	.045	.926	.048	.917	.048	.924	.046	.922
	Baseline	.076	.874	.070	.878	.074	.868	.064	.894	.067	.885
	Avg.	.085	.851	.084	.847	.088	.838	.082	.873	.084	.869
	Img. Input	.054	.903	.056	.892	.061	.881	.058	.900	.055	.897
	Sal. Input	.046	.926	.049	.920	.500	.913	.053	.918	.050	.914
	Reg. Loss	.050	.919	.050	.917	.053	.910	.052	.918	.053	.914
	w/o Com.	.049	.921	.053	.914	.053	.908	.057	.910	.054	.911
	Bub. Test	.048	.923	.051	.915	.050	.914	.049	.923	.049	.914
	R-18	.049	.919	.047	.923	.051	.916	.054	.911	.048	.916
	SER-18	.048	.922	.047	.922	.049	.920	.051	.917	.048	.916
LFSD	Ours	.070	.882	.070	.892	.066	.900	.078	.888	.071	.892
	Baseline	.098	.835	.105	.830	.087	.861	.084	.870	.089	.860
	Avg.	.105	.829	.108	.820	.096	.843	.089	.857	.101	.839
	Img. Input	.072	.882	.072	.878	.073	.885	.080	.878	.079	.882
	Sal. Input	.070	.881	.073	.884	.067	.895	.087	.872	.074	.888
	Reg. Loss	.076	.874	.079	.876	.070	.893	.080	.884	.074	.890
	w/o Com.	.078	.871	.075	.884	.065	.897	.086	.873	.071	.891
	Bub. Test	.076	.875	.081	.874	.066	.896	.079	.887	.073	.892
	R-18	.077	.875	.072	.884	.067	.892	.079	.884	.075	.881
	SER-18	.076	.875	.073	.886	.066	.896	.079	.885	.073	.887

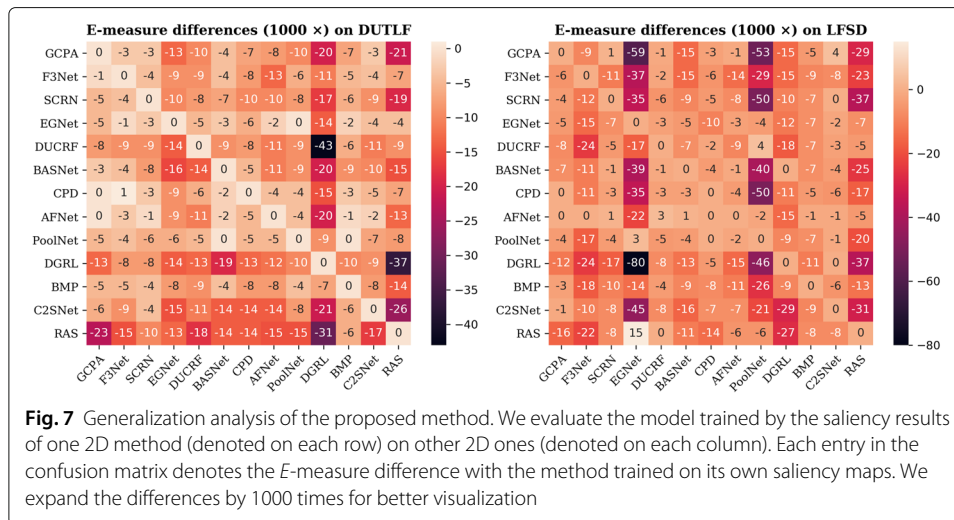
4.3.2 The ability of generalization

All above results of a certain 2D method are based on the model trained on its own saliency maps. Actually, the individual model can be deployed on any 2D methods without retraining. To verify the generalization ability of our method, we conduct some analyses in Fig. 7. At each confusion matrix, the method on each row denoted the training data resources. Then, the trained model is evaluated on different methods, which are denoted on each column. Each entry in the confusion matrix denotes the E -measure difference with the method trained on its own saliency maps. We expand the differences by 1000 times for better visualization. We notice that the model trained by most methods generalizes well with minor performance descending, except DGRL, C2SNet, and RAS. From the columns of the matrix, most generalized 2D methods obtain unsatisfied performance on EGNet, PoolNet, DGRL, and RAS. We attribute the main reason to the distribution difference between the saliency maps of these methods that cannot be generalized well and other methods.

4.4 Discussions

4.4.1 Relationship with the 4D methods

Existing CNN-based 4D methods can be summarized as an implicit selection of focal stacks. Attention mechanism [7, 9] is adopted to emphasize useful features in the focal slices, where the salient objects happen to be in focus. Such features will be fused with the



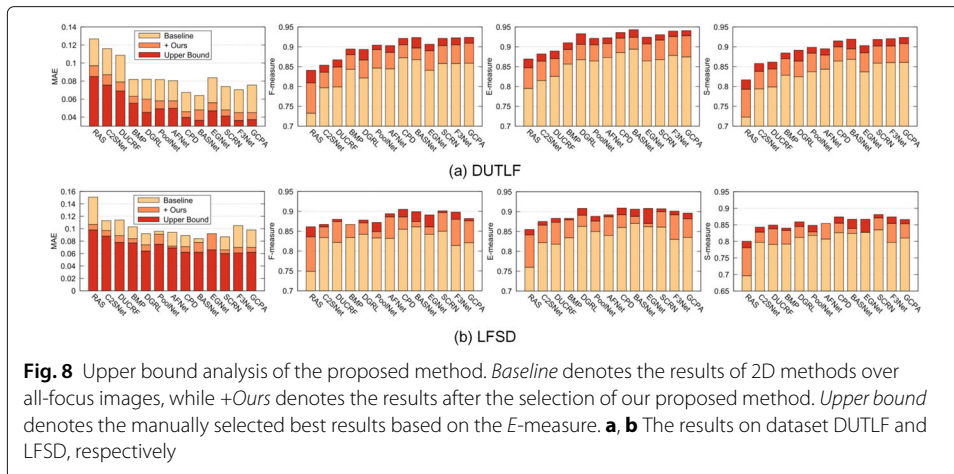
segmentation branch on the all-focus images. Unlike other fusion-based tasks, e.g., video and action recognition, where sequential information is important, light field SOD only needs concentrating on the images where the salient object is in focus. Fusing features from focal stacks could confuse the segmentation. Observing the qualitative results of MoLF in Fig. 5, we find that the boundary of salient object is not as sharp as the one of 2D methods. The main reason may attribute to the features where the salient object is blurred. On the contrary, our method explicitly selects the proper inputs and degenerates light field SOD to a common 2D task. It maintains the strength of existing segmentation networks and thereby provides superior results.

4.4.2 The upper bound analysis

The capacity of the proposed method is limited by the power of 2D methods. In Fig. 8, we analyze the upper bound of our method. *Baseline* denotes the results of 2D methods over all-focus images, while *+Ours* denotes the results after the selection of our proposed method. *Upper bound* denotes the manually selected best results based on the E -measure. On average, the proposed method achieves 83.9% (\mathcal{M}), 97.9% (\mathcal{F}), 98.1% (\mathcal{E}), and 97.9% (\mathcal{S}) performance of the upper bound on the dataset DUTLF. The corresponding number on LFS is 87.0% (\mathcal{M}), 98.5% (\mathcal{F}), 98.5% (\mathcal{E}), and 98.1% (\mathcal{S}), respectively. The results in Fig. 8 demonstrate that most results are quite close to the upper bound, especially on the dataset LFS.

4.4.3 Typical failure cases

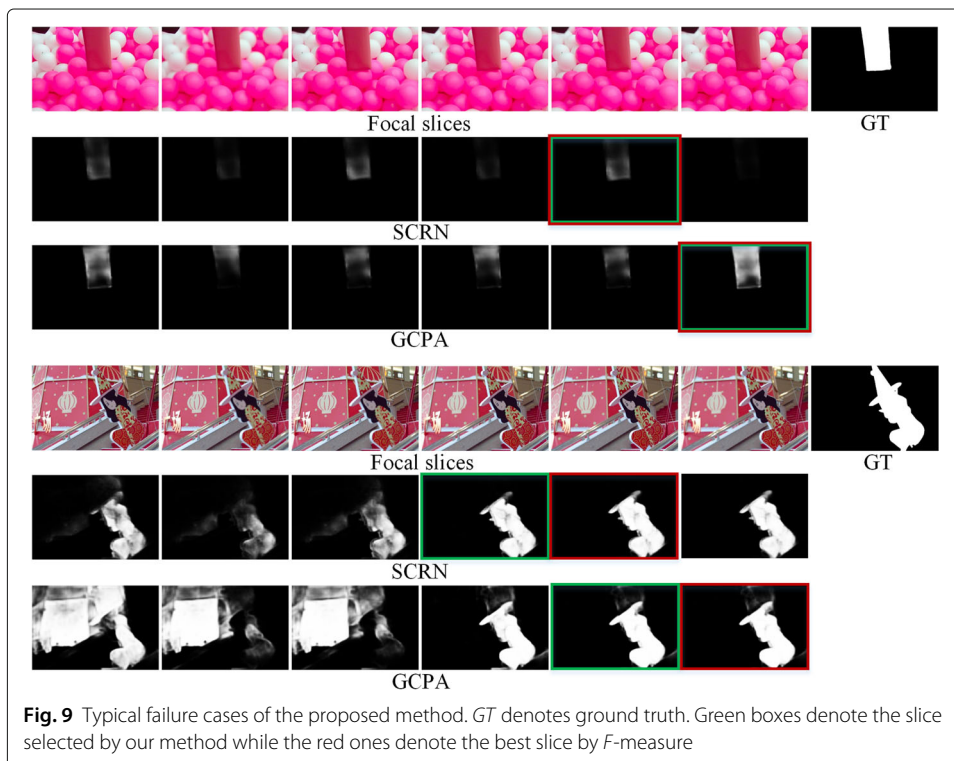
In Fig. 9, we present some typical failure cases of the proposed method. Green boxes denote the slice selected by our method while the red ones denote the best slice by F -measure. The first example demonstrates one case when the 2D methods fail to detect the salient object. Although the proposed method can properly select the best slice, the saliency results are still far away from the ground truth. The second example shows another challenging situation when the saliency regions have smoothly varying depths, as shown in the last three columns of the second example. These saliency results are too similar to be correctly classified by the proposed method. However, the quantitative scores



of these results are very close and affect the overall performance little. For example, the *F*-measure of last three slices of SCRN is 0.92447, 0.9288, and 0.9274, respectively.

4.4.4 Processing speed

When testing, we predict the score of each input slice with Eq. (2), which avoids the calculation of bubble sorting. At each time, we build a mini-batch by copying the target slice as the number of the rest slices in the same light field sample. Feeding forward such a mini-batch takes about 0.0137 s. The total running time depends on the number of slices and the processing speed of 2D method. Take 2D method CPD and dataset DUTLF for instance. Each sample of DUTLF dataset on average has 8.2 slices. The processing speed of CPD is about 62 fps. Therefore, the total time to deal with one light field sample is



equal to $8.2 \times (0.0137 + 1/62) = 0.245$ s. Correspondingly, the processing speed of Piao et al. [38] and MoLF [9] is 0.07 and 0.11 s.

5 Conclusions

In this paper, we provide an alternative solution for the task of light field SOD. Without designing specialized segmentation network for light field data, a model is proposed to optimize the input of existing 2D methods, which have made significant progress. The proposed model learns to predict the relative performance of any two slices from one light field sample. An attention-based comparator is proposed to emphasize the distinctiveness of same two slices but in different order of comparison. Experiments on 13 latest 2D methods demonstrate that the proposed strategy dramatically improves the performance of 2D methods on 2 light field datasets. Moreover, extra analyses demonstrate that the model trained on one method results has an impressive generalization ability, which means the proposed method can continuously benefit from the improvement of 2D methods.

Abbreviations

CNN: Convolutional neural network; SOD: Saliency object detection; FCN: Fully convolutional network; ReLU: Rectified linear unit; SGD: Stochastic gradient descent; GPU: Graphic processing unit

Acknowledgements

Not applicable.

Authors' contributions

YL and HX carried out the main part of this manuscript. HT participated in the design of the comparator. PL participated in the discussion. All authors read and approved the final manuscript.

Funding

This work was supported in part by the National Natural Science Foundation (NSFC) of China under project no. 61906206.

Availability of data and materials

Data are publicly available.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Department of Systems Engineering, National University of Defense Technology, Deya Road, Changsha 410073, China.

²School of Computer Science and Technology, Hangzhou Dianzi University, Baiyang Road, Hangzhou 310018, China.

Received: 23 June 2020 Accepted: 7 October 2020

Published online: 02 November 2020

References

1. A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, Visual tracking: an experimental survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(7), 1442–1468 (2013)
2. G. Liu, Z. Liu, S. Liu, J. Ma, F. Wang, Registration of infrared and visible light image based on visual saliency and scale invariant feature transform. *EURASIP J. Image Video Process.* **2018**(1), 1–12 (2018)
3. M. Sharif, M. A. Khan, T. Akram, M. Y. Javed, T. Saba, A. Rehman, A framework of human detection and action recognition based on uniform segmentation and combination of euclidean distance and joint entropy-based features selection. *EURASIP J. Image Video Process.* **2017**(1), 89 (2017)
4. Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, T. S. Huang, in *CVPR*, Revisiting dilated convolution: a simple approach for weakly-and semi-supervised semantic segmentation (IEEE, Salt Lake City, 2018), pp. 7268–7277
5. H. Xiao, Y. Wei, Y. Liu, M. Zhang, J. Feng, in *AAAI*, Transferable semi-supervised semantic segmentation (AAAI Press, New Orleans, 2018)
6. N. Li, J. Ye, Y. Ji, H. Ling, J. Yu, in *CVPR*, Saliency detection on light field (IEEE, Columbus, 2014), pp. 2806–2813
7. T. Wang, Y. Piao, X. Li, L. Zhang, H. Lu, in *ICCV*, Deep learning for light field saliency detection (IEEE, Seoul, 2019), pp. 8838–8848
8. J. Zhang, M. Wang, L. Lin, X. Yang, J. Gao, Y. Rui, Saliency detection on light field: a multi-cue approach. *ACM TOMM.* **13**(3), 1–22 (2017)
9. M. Zhang, J. Li, W. Ji, Y. Piao, H. Lu, in *NeurIPS*, Memory-oriented decoder for light field salient object detection (Curran Associates, Inc., Vancouver, 2019), pp. 896–906
10. M. Sharif, M. A. Khan, T. Akram, M. Y. Javed, T. Saba, A. Rehman, Light field photography with a hand-held plenoptic camera. *Comput. Sci. Tech. Rep.* **2**(11), 1–11 (2005)

11. Z. Wu, L. Su, Q. Huang, in *ICCV*, Stacked cross refinement network for edge-aware salient object detection (IEEE, Seoul, 2019), pp. 7264–7273
12. R. Achanta, S. Hemami, F. Estrada, S. Susstrunk, in *CVPR*, Frequency-tuned salient region detection (IEEE, Miami, 2009), pp. 1597–1604
13. D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, A. Borji, in *IJCAI*, Enhanced-alignment measure for binary foreground map evaluation (AAAI Press, Stockholm, 2018), pp. 698–704
14. M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, S.-M. Hu, Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(3), 569–582 (2014)
15. Y. Niu, Y. Geng, X. Li, F. Liu, in *CVPR*, Leveraging stereopsis for saliency analysis (IEEE, Providence, 2012), pp. 454–461
16. X. Shen, Y. Wu, in *CVPR*, A unified approach to salient object detection via low rank matrix recovery (IEEE, Providence, 2012), pp. 853–860
17. H. Xiao, J. Feng, Y. Wei, M. Zhang, S. Yan, Deep salient object detection with dense connections and distraction diagnosis. *IEEE TMM.* **20**(12), 3239–3251 (2018)
18. G. Li, Y. Yu, in *CVPR*, Visual saliency based on multiscale deep features (IEEE, Boston, 2015), pp. 5455–5463
19. R. Zhao, W. Ouyang, H. Li, X. Wang, in *CVPR*, Saliency detection by multi-context deep learning (IEEE, Boston, 2015), pp. 1265–1274
20. G. Li, Y. Yu, in *CVPR*, Deep contrast learning for salient object detection (IEEE, Las Vegas, 2016), pp. 478–487
21. Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, P. H. Torr, in *CVPR*, Deeply supervised salient object detection with short connections (IEEE, Hawaii, 2017), pp. 3203–3212
22. Z. Deng, X. Hu, L. Zhu, X. Xu, J. Qin, G. Han, P.-A. Heng, in *IJCAI*, R3net: recurrent residual refinement network for saliency detection (AAAI Press, New Orleans, 2018), pp. 684–690
23. X. Li, F. Yang, H. Cheng, W. Liu, D. Shen, in *ECCV*, Contour knowledge transfer for salient object detection (Springer, Munich, 2018), pp. 355–370
24. X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, M. Jagersand, in *CVPR*, Basnet: boundary-aware salient object detection (IEEE, Long Beach, 2019), pp. 7479–7489
25. T. Wang, L. Zhang, S. Wang, H. Lu, G. Yang, X. Ruan, A. Borji, in *CVPR*, Detect globally, refine locally: a novel approach to saliency detection (IEEE, Salt Lake City, 2018), pp. 3127–3135
26. Z. Wu, L. Su, Q. Huang, in *CVPR*, Cascaded partial decoder for fast and accurate salient object detection (IEEE, Long Beach, 2019), pp. 3907–3916
27. J.-X. Zhao, J.-J. Liu, D.-P. Fan, Y. Cao, J. Yang, M.-M. Cheng, in *ICCV*, Egnnet: edge guidance network for salient object detection (IEEE, Seoul, 2019), pp. 8779–8788
28. N. Liu, J. Han, M.-H. Yang, in *CVPR*, Picanet: learning pixel-wise contextual attention for saliency detection (Springer, Munich, 2018), pp. 3089–3098
29. S. Chen, X. Tan, B. Wang, X. Hu, in *ECCV*, Reverse attention for salient object detection (Springer, Munich, 2018), pp. 234–250
30. M. Feng, H. Lu, E. Ding, in *CVPR*, Attentive feedback network for boundary-aware salient object detection (IEEE, Long Beach, 2019), pp. 1623–1632
31. Y. Piao, W. Ji, J. Li, M. Zhang, H. Lu, in *ICCV*, Depth-induced multi-scale recurrent attention network for saliency detection (IEEE, Seoul, 2019), pp. 7254–7263
32. J.-X. Zhao, Y. Cao, D.-P. Fan, M.-M. Cheng, X.-Y. Li, L. Zhang, in *CVPR*, Contrast prior and fluid pyramid integration for RGBD salient object detection (IEEE, Long Beach, 2019), pp. 3927–3936
33. L. Qu, S. He, J. Zhang, J. Tian, Y. Tang, Q. Yang, RGBD salient object detection via deep fusion. *IEEE Trans. Image Process.* **26**(5), 2274–2285 (2017)
34. H. Chen, Y. Li, in *CVPR*, Progressively complementarity-aware fusion network for RGB-D salient object detection (IEEE, Salt Lake City, 2018), pp. 3051–3060
35. C. Zhu, X. Cai, K. Huang, T. H. Li, G. Li, in *ICME*, Pdnet: prior-model guided depth-enhanced network for salient object detection (IEEE, Shanghai, 2019), pp. 199–204
36. J. Zhang, M. Wang, J. Gao, Y. Wang, X. Zhang, X. Wu, in *IJCAI*, Saliency detection with a deeper investigation of light field (AAAI Press, Buenos Aires, 2015), pp. 2212–2218
37. N. Li, B. Sun, J. Yu, in *CVPR*, A weighted sparse coding framework for saliency detection (IEEE, Boston, 2015), pp. 5216–5223
38. Y. Piao, Z. Rong, M. Zhang, H. Lu, in *AAAI*, Exploit and replace: an asymmetrical two-stream architecture for versatile light field saliency detection (AAAI Press, New York, 2020)
39. D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, A. Borji, in *ICCV*, Structure-measure: a new way to evaluate foreground maps (IEEE, Venice, 2017), pp. 4548–4557
40. X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, F. Porikli, in *CVPR*, See more, know more: unsupervised video object segmentation with co-attention siamese networks (IEEE, Long Beach, 2019), pp. 3623–3632
41. C. Xiong, V. Zhong, R. Socher, in *ICLR*, Dynamic coattention networks for question answering (ICLR, Puerto Rico, 2016)
42. J. Snell, K. Swersky, R. Zemel, in *NIPS*, Prototypical networks for few-shot learning (Curran Associates, Inc., Long Beach, 2017), pp. 4077–4087
43. F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, T. M. Hospedales, in *CVPR*, Learning to compare: relation network for few-shot learning (IEEE, Salt Lake City, 2018), pp. 1199–1208
44. Z. Cao, Q. Xu, R. Cong, Q. Huang, in *AAAI*, Global context-aware progressive aggregation network for salient object detection (AAAI Press, New York, 2020)
45. J. Wei, S. Wang, Q. Huang, in *AAAI*, F3net: fusion, feedback and focus for salient object detection (AAAI Press, New York, 2020)
46. Y. Xu, D. Xu, X. Hong, W. Ouyang, R. Ji, M. Xu, G. Zhao, in *ICCV*, Structured modeling of joint deep feature and prediction refinement for salient object detection (IEEE, Seoul, 2019), pp. 3789–3798
47. J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, J. Jiang, in *CVPR*, A simple pooling-based design for real-time salient object detection (IEEE, Long Beach, 2019), pp. 3917–3926
48. L. Zhang, J. Dai, H. Lu, Y. He, G. Wang, in *CVPR*, A bi-directional message passing model for salient object detection (IEEE, Salt Lake City, 2018), pp. 1741–1750

49. N. Li, J. Ye, Y. Ji, H. Ling, J. Yu, Saliency detection on light field. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(8), 1605–1616 (2017)
50. K. He, X. Zhang, S. Ren, J. Sun, in *CVPR*, Deep residual learning for image recognition (IEEE, Las Vegas, 2016), pp. 770–778
51. J. Hu, L. Shen, G. Sun, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Squeeze-and-Excitation Networks, (Salt Lake City, 2018), pp. 7132–7141. <https://doi.org/10.1109/CVPR.2018.00745>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
