Fast and Accurate Light Field Saliency Detection through Deep Encoding

Sahan Hemachandra[*], Ranga Rodrigo[*], Chamira U. S. Edussooriya[*,#]

[*] *Department of Electronic and Telecommunication Engineering, University of Moratuwa, Sri Lanka*
[#] *Department of Electrical and Computer Engineering, Florida International University, Miami, FL, USA*

Abstract

Light field saliency detection—important due to utility in many vision tasks—still lacks speed and can improve in accuracy. Due to the formulation of the saliency detection problem in light fields as a segmentation task or a memorizing task, existing approaches consume unnecessarily large amounts of computational resources for training, and have longer execution times for testing. We solve this by aggressively reducing the large light field images to a much smaller three-channel feature map appropriate for saliency detection using an RGB image saliency detector with attention mechanisms. We achieve this by introducing a novel convolutional neural network based features extraction and encoding module. Our saliency detector takes 0.4 s to process a light field of size 9 × 9 × 512 × 375 in a CPU and is significantly faster than state-of-the-art light field saliency detectors, with better or comparable accuracy. Furthermore, model size of our architecture is significantly lower compared to state-of-the-art light field saliency detectors. Our work shows that extracting features from light fields through aggressive size reduction and the attention mechanism results in a faster and accurate light field saliency detector leading to near real-time light field processing.

Keywords: Light fields, saliency detection, feature extractor, fast algorithms, convolutional neural networks.

1. Introduction

Light fields capture both spatial and angular information of light emanating from a scene compared to spatial-only information captured by images. The additional angular information available with light fields paves the way for novel applications such as post-capture refocusing [1, 2, 3] and depth-based filtering [4, 5, 6], which are not possible with images. Furthermore, light fields support numerous computer vision tasks which are traditionally based on images and videos [7, 8, 9, 10, 11, 12, 13, 14, 15].

Saliency detection is a prerequisite for many computer vision tasks such as semantic segmentation, image retrieval, and scene classification. Saliency detection using light fields provides better accuracy compared to what is provided by RGB images, in particular, for challenging scenes having similar foreground and background, and complex occlusions [16, 17]. However, data available with light fields (i.e., pixels per light field) are significantly higher than data available with a single RGB image, e.g., a light field having 9 × 9 sub-aperture images contains 81 times more data (with the same resolution). Therefore, computational time of light field saliency detection algorithms is substantially higher compared to that of RGB image saliency detection algorithms [17].

We can categorize existing light field saliency detectors in to three classes depending on the input: focal stack and all-focus image, RGB-D images, and light fields. In the first class, a set of two-dimensional (2-D) images focused at different depths, called a focal stack, and a sub-aperture image, called all-focus image, are used as the input. Here, a focal stack is generated from a light field using a refocusing algorithm [1, 2], and this step acts as a preprocessing step with additional computations. Furthermore, focal stack generation requires human intervention because the number of 2-D images in a focal stack depends on a light field. The second class employs RGB-D images consisting of an all-focus image and a depth map. In this case, the depth map is generated using a depth estimation algorithm [18, 19, 20] and incur additional computations. Similar to the generation of a focal stack, generation of a depth map is also a preprocessing step. Compared to these two classes, the third class employs a light field as the input without any preprocessing steps. Recent algorithms of these three categories predominately use convolutional neural networks (CNNs) to learn the relationship between the image features and saliency of light fields. Even though the available light field datasets are limited in size, we can freely augment focal stack and RGB-D data in the first two classes. On the other hand, inability to freely augment light field images prevents training deep CNNs from scratch in the third class. These constraints demand the use of pre-trained networks, of course, followed by fine tuning.

In this paper, we propose a novel feature *extraction and encoding* (FEE) module for *fast* light field saliency detection by employing a 2-D RGB image saliency detection algorithm. Our FEE module takes light field as the input (so, belongs to the third class), and provides an RGB encoded feature map. The proposed FEE module comprises of a CNN with five convolutional layers. Here, we employ sub-aperture images as our input in contrast to most of the previously proposed light field saliency detectors, where focal stack is the input. This prevents the use of computationally-high recurrent neural network layers such as long short-term memory (LSTM) and ConvLSTM [21] and enables to employ computationally- and memory-efficient convolutional layers. We employ the 2-D saliency detector proposed in [22] with our FEE module. Furthermore, we employ the LYTRO ILLUM saliency detection dataset [23] and DUTLF-v2 dataset [24] for the training and testing the performance of our light field saliency detector. Here we augment both light field data sets without affecting angular features. Experimental results obtained with five-fold cross validation confirms that our saliency detector provides a *significant improvement* in computational time with accuracy comparable or better than state-of-the-art light field saliency detectors [23, 25]. Furthermore, model size of our architecture is significantly lower compared to state-of-the-art light field saliency detectors leading to a lower memory requirement. Such low complexity saliency detectors are required for applications such as field robotics [9, 12, 15], where real-time operation on embedded systems are mostly required. In summary, our contributions are:

- introducing a novel and computationally and memory-wise efficient method to detect salient objects in light fields using the FEE module in combination with an RGB salient object detector.
- introducing a combination of techniques, such as random rotation by 90^o, random changes of brightness, saturation and contrast, and shuffling channels randomly, to augment light fields in lens array format without affecting angular features.

The paper is organized as follows. In Section 2, we review different approaches employed and algorithms proposed for saliency detection of RGB images and light fields. We present our light field saliency detector in detail in Section 3. In Section 4, we present experimental results to verify the accuracy and the speed of the proposed light field saliency detector. Finally, in Section 5, we present conclusion and future works.

2. Related Works

2.1. Saliency Detection on RGB images

Saliency detection is one of the oldest problems in computer vision research and there have been many research done on various approaches for this task in the recent time. Earliest research [26, 27, 28, 29] were mainly based on handcrafted features like boundaries or contrast of the images to detect the most salient objects in the RGB images. [26] proposed a graph based manifold ranking algorithm for salience detection based on background and foreground cues. [27] proposed an regional contrast based algorithm, where global contrast and spatial weighted coherence scores are used simultaneously to accurately detect the salient objects. [28] introduced a contrast based approach using high dimensional Gaussian filters to unifying detect salience and complete contrast while [29] used background priors to detect the salient regions on images. Even though these methods are less computationally expensive, they tends to fail in complex backgrounds. Recently, [30] introduced an adaptive, weighted k-means-based superpixel segmentation with self-adjustable distance measures for accurate superpixel segmentation for salient object detection in RGB images.

With the popularity of deep learning in the last decade, many approaches based on deep learning has been introduced for the RGB saliency detection using neural networks. There is rich body of work in saliency detection in RGB images: pyramidal, feature based, recurrent network based, and attention based. Most non-recurrent methods use VGG-16- or VGG-19-like feature extractors [31] pre-trained on ImageNet dataset for feature extraction. Pyramidal saliency detectors [32, 33, 22] have the advantage of the ability to use information from multiple layers. Some that build up on CNN feature computers defer the actual saliency detection to latter layers or combine features from many layers [34, 35]. [36] introduced a gate-based contextual feature extraction module for salient object detection in RGB images where learnable gates act as a filter to extract relevant contextual information. Methods that employ recurrent networks generally work well [37, 38] with the possible disadvantage of slowness. RGB saliency detectors greatly benefit from attention models, by focusing on features that truly capture saliency without the interference of unnecessary features. [39] introduced multi-scale feature extraction based CNN combining an adjacent layer attention block and a partial encoder–decoder block. This approach mitigates the issues in existing CNN-based approaches like embedding of abstract information and loss of spatial information due to late fusion of detailed features. Although these methods show success in RGB images, they are unsuitable for direct use with light field images because their architecture and input are not specifically designed to extract the geometry information of light fields embedded in angular dimensions. This information is vital to improve the quality of predicted saliency maps. For a comprehensive review of saliency detection on RGB images, the reader is referred to [40, 41, 42].

2.2. Saliency Detection on Light Fields

Light field saliency detection [16] improves the accuracy of saliency detection in challenging scenes having similar foreground/background and complex occlusions. This improvement achieves in [16] by exploiting the refocusing capability available with light fields which provides focusness, depths, and objectness cues. [17] employs depth map, all focus image and focal stack available with a light field for saliency detection. [43] further exploits light field flow fields over focal slices, and multi-view sub-aperture images improve the accuracy in saliency detection by enhancing depth contrast. [44] employs a dictionary learning based method to combine various light field features for a universal saliency detection framework using sparse coding. This method handles various types of input data by building saliency and non-saliency dictionaries using focusness cues of focal stack as features for light fields. All these methods works on super-pixel level features of light fields, and do not exploit high-level semantic information properly in order to have robust performance in complex scenarios.

2.3. Light Field Saliency Detection with Deep Learning

Recent advances in light field saliency detection successfully use deep neural networks. [21] introduced a two-stream neural network architecture with two VGG-19 feature extractors and ConvLSTM-based attention module to process the all-focus image and focal stack to generate saliency maps. The saliency detection model in [45] use a deep neural network pipeline containing light field synthesising network using center view and a light field driven saliency detection network to detect salient objects in single view images. Similarly, [25] employed a multi-task collaborative network (MTCN) for light field saliency detection with two streams for central view image and multi-view images by exploring the spatial, depth and edge information in different parts of their neural network with the help of a complicated loss function with different components for different parts of the network. [23] introduced a "model angular changes block" to process light field images with a modified version of Deeplabs v-2 segmentation network (LFNet), which is a computationally heavy backbone, considering the similarity between the segmentation and saliency detection. On the other hand, the suitability of a semantic segmentation network, not specifically trained on light fields, may affect accuracy. [24] introduced, two-stream network containing teacher and student network to detect salient objects, exploiting focal stack and all focus image in their respective streams of the network. Most of these methods have the inherent disadvantage of slowness due to use of heavy segmentation networks, several feature extractors, recurrent blocks and several streams. Furthermore, full light field data are employed for most of the parts and layers of the neural networks hindering the speed.

3. Proposed Light-Field Saliency Detection Architecture

Speeding-up light-field saliency detection requires avoiding computationally heavy one or more backbones and predominantly working in bulky light-field features maps. On the other hand, inability to freely augment light field images prevent training deep light field saliency detectors from scratch. These constraints demand using a pre-trained network (of course, followed by fine tuning). There are well-known pre-trained networks that detect saliency in 2-D RGB images [22, 46, 47]. In this paper we propose a FEE module that can be integrated into 2-D saliency detectors without any architectural changes to the *base model*, to extract and encode the features in light fields. Figure 1 shows an overview of the architecture of our system. The input to this neural network is a light field of size $S \times T \times U \times V$ in the form of a micro-lens image array of of size $W \times H$, where $W = S \times U$ and $H = T \times V$. Here, (S, T) denotes the spatial resolution and (U, V) denotes the angular resolution of a light field. Figure 2 shows a light field with sub-aperture images and micro lens array image. Then the extracted feature maps can be fed into the 2-D saliency detector to get the saliency maps. This whole network can be trained end-to-end manner after the integration.

3.1. 2-D Saliency Detector

Task of saliency detection in regular images is similar to binary semantic segmentation, and for this task requires both high level contextual information and low level spatial structural information. However, all of the high-level and low-level features are not suitable for saliency detection, and some features might even cause interference [22]. An attention mechanism can avoid such situations. The 2-D saliency detector proposed in [22] is such a system which we select as the saliency detector. This work especially uses channel wise attention module (CA) for high-level feature maps and spatial attention (SA) module for low-level feature maps with edge preserving loss function to preserve the edges of a saliency map. Along with the CA and SA modules, the pyramid feature network of the architecture leads to the state-of-the-art accuracy for RGB image saliency detection. However, using a single sub-aperture image or the all-focus image of a light field to feed the input of a 2-D saliency detector is ineffective as angular information of the light field gets lost. We solve this problem by using a carefully designed novel light field FEE module integrated in to

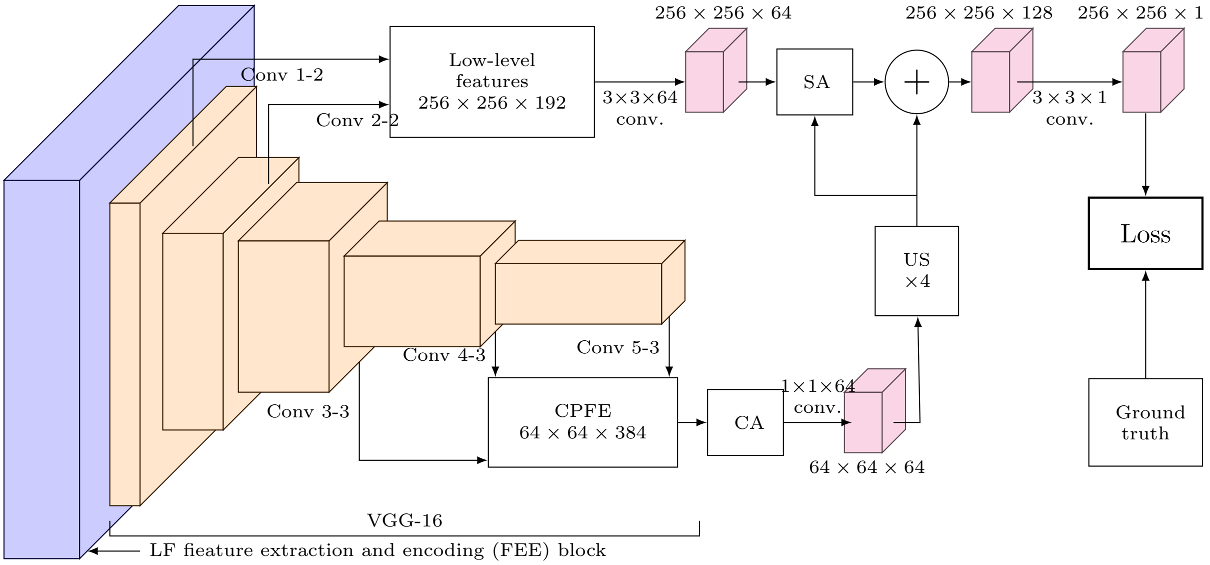


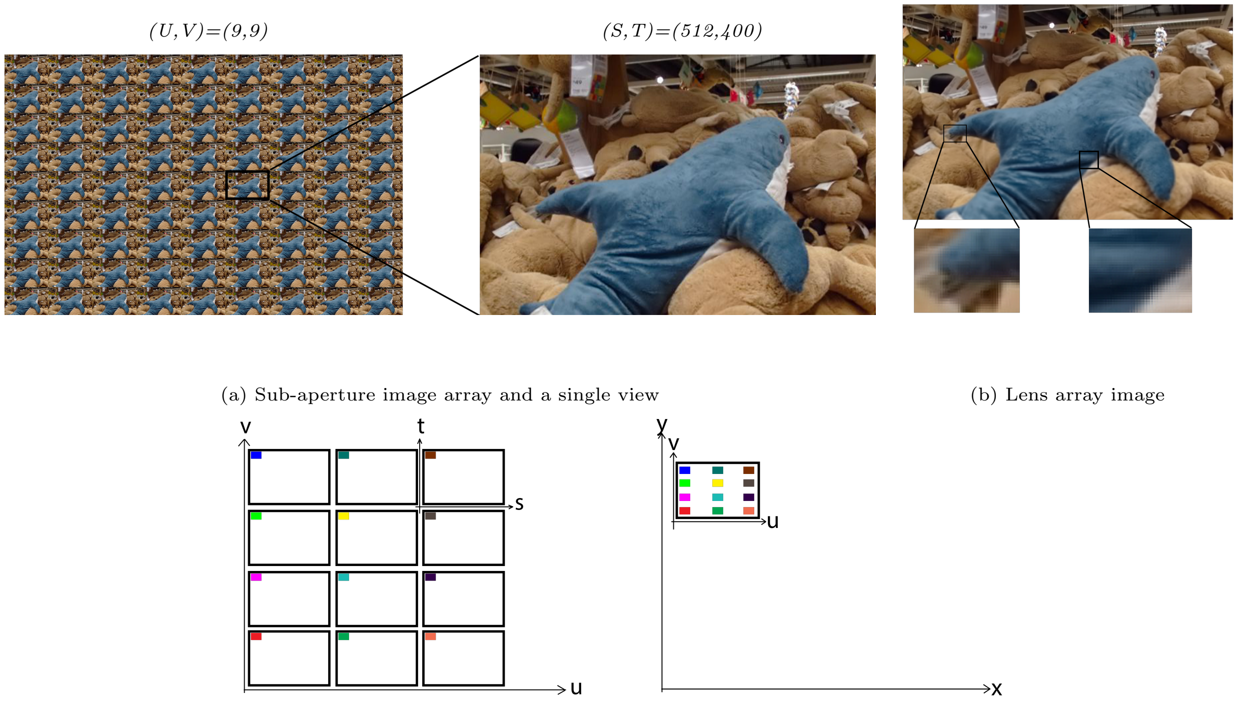
Figure 1: System architecture: light field FEE block receives the light fields and computes features. Spatial attention block (SA) and channel-wise attention block (CA) receives low level (Conv 1-2 and Conv 2-2) and high level (Conv3-3, Conv 4-3 and Conv 5-3) features, respectively. VGG-16, or a similar block, produces these feature maps. Note that light field processing happens only in the light field FEE block. CA block gives attention to more informative kernel outputs. CPFE: context aware pyramid feature extraction.

the input of the network. We do not describe the architecture of the 2-D saliency detector in detail, and we refer the reader to [22] for more details.

3.2. Novel Feature Extraction and Encoding Module

The 2-D saliency detector accepts inputs with resolution of $256 \times 256 \times 3$ and produces saliency maps with resolution of $256 \times 256 \times 1$. Starting from this, our FEE module must extract and encode the pixel-wise angular information stored in a light field and produce an RGB image. In order to do that, by arranging a light field as a 2-D image of size $W \times H$, we run an $U \times V$ kernel with the stride of (U, V) to exploit the angular information related to each pixel as mentioned in [23]. Here, we consider the modified light fields in the LYTRO ILLUM [23], where $(U, V) = (9, 9)$ and $(S, T) = (512, 375)$ leading to $W = 4608$ and $H = 3375$, and DUTLF-v2 [24] ,where $(U, V) = (9, 9)$ and $(S, T) = (512, 400)$ leading to $W = 4608$ and $H = 3600$. *Because our light filed saliency detector shown in Figure 1 processes light fields only in the FEE module and prevents subsequent processing in the 2-D saliency detector, we can achieve significant saving of computational time.*

The FEE module as depicted in Figure 3 is the key component that leads to significant speed improvements. The FEE module aggressively downsamples a light field and encodes features suitable to be fed to a regular CNN. The FEE module consists of five convolutional layers. The first convolutional layer consist of 128 filters of size 9×9 and a $(9, 9)$ stride. This layer precedes two convolutional layers having 64 and



(a) Sub-aperture image array and a single view

(b) Lens array image

(c) pixels in sub-aperture image array

(d) pixels in lens array image

Figure 2: A light field consists of $U \times V$ sub-aperture images, where each sub-aperture image has $S \times T$ pixels as shown in (a) & (c). The pixels in sub-aperture image array can be restructured to get the lens-array image shown in (b), where extracted patches depict the pixel arrangement of the lens-array image. (c) & (D) depict how the pixels in sub-aperture images in (c) are rearranged in order to obtain a lens-array image shown in (d), where the first block of size $U \times V$ contains all the pixels of spatial position $(1, 1)$ in each sub-aperture image. As an example, when the lens-array image in (b) is considered, top left 9×9 block of pixels contains all 81 pixels located at $(1, 1)$ position of each 81 sub-aperture images. Using this arrangement, a convolution block stride (U, V) can be used to extract angular information from sub-aperture images.

32 filters of size 3×3 and a stride $(1, 1)$, respectively. A similar layer having 32 filters of size 3×3 and a stride $(2, 2)$ is used to downsample the feature maps into $(256, 256, 32)$ and the last convolution layer has 3 filters of size 1×1 and a stride $(1, 1)$ and compute the encoded output that is fed to the 2-D RGB saliency detector.

As the input light field is a micro lens array image, adjacent pixels in the first 9×9 block comprises the first pixel of each of the 81 sub-aperture images, see Figure 2d. Therefore, by using a stride of $(9, 9)$ in the first convolutional layer, we capture the same pixel for all the sub-aperture images at each convolution step. Following this, we select the layer-size parameters of hidden convolutional layers to be compatible with the architecture of VGG-16 network with decreasing number of filters at each layer to encode the light field in to a feature map of $256 \times 256 \times 3$ resolution. We note that VGG-16 is just one choice of the back bone, and other backbones, e.g., ResNets are also suitable.

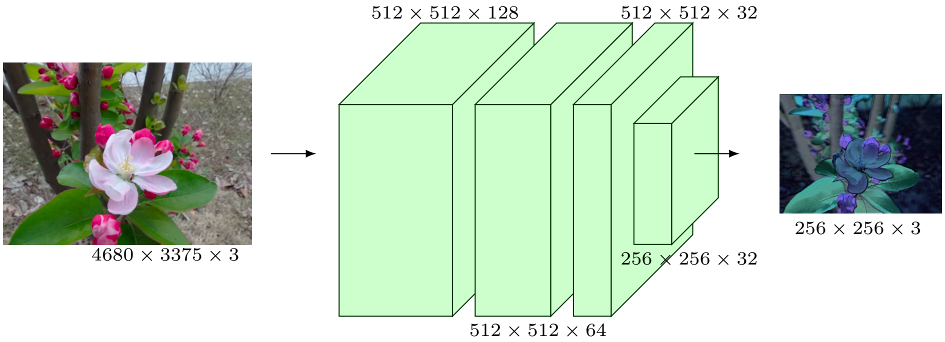


Figure 3: The light field FEE block receives micro lens array images of resolution 4608 × 4608 × 3 and encodes the light field image into an RGB image of resolution 256 × 256 × 3 through five convolutional layers.The FEE module consists of five convolutional layers. The first convolutional layer consist of 128 filters of size 9 × 9 and a $(9, 9)$ stride. This layer precedes two convolutional layers having 64 and 32 filters of size 3 × 3 and a stride $(1, 1)$, respectively. A similar layer having 32 filters of size 3 × 3 and a stride $(2, 2)$ is used to downsample the feature maps into $(256, 256, 32)$ and the last convolution layer has 3 filters of size 1 × 1 and a stride $(1, 1)$.

4. Experimental Results

We present experimental results in this section. We employ the LYTRO ILLUM [23] and DUTLF-v2 [24] datasets in the experiments with a computing platform comprising of an Intel Core i9-9900K (3.60 GHz) CPU, 32 GB RAM and Nvidia RTX-2080Ti GPU. Note that even though two other light field saliency datasets, HFUT-Lytro [43] and LFSD [48], are publicly available, they are not suitable for evaluation of our light field saliency detector due to the low angular resolution and unavailability of sub-aperture images. There are 640 light fields in the LYTRO ILLUM dataset, and we compare the performance the proposed light field saliency detector with the state-of-the-art light field saliency detectors LFNet [23] and MTCN [25] and state-of-the-art 2-D saliency detectors NLDF [49], PAGE-Net [50], GCPA Net [47], SCRNet [46] and SODGAN [51] in terms of the accuracy achieved with five-fold cross validation and computational time. The DUTLF-v2 [24] dataset contains 4208 light fields divided into test and train sets having 2961 and 1247 light fields, respectively. We compare the performance of the proposed method, fine-tuned in the train set, with state-of-the-art 2-D saliency detectors, DLFS [45] and student networks proposed in [24] and light field saliency detectors, LFNet [23], DisenFusion [52], ATAFNet [53], CPFP [54], MOLF [55] and teacher network proposed in [24].

4.1. Implementation and Training of the Proposed Light Field Saliency Detector

To facilitate the proposed FEE module to encode a light field into 256 × 256 × 3 feature map, we crop the initial micro lens array images of the LYTRO-ILLUM dataset[23] of size 4860 × 3375 × 3 into four images of size 4608 × 3375 × 3, removing pixels at the borders. This leads to a dataset of 2560 light fields, and we incorporate data augmentation, such as random rotations of 90° and 180°, random brightness, saturation and contrast changing, and random shuffling of the colour channels without affecting the angular information available with a light field. For the DUTLF-v2 [24], we use the same configuration except cropping micro lens array images into an one $4608 \times 3600 \times 3$ micro lens array image. We train our saliency detector in three steps. First, we train 2-D saliency detector [22] on a combined dataset of DUTS-TR [56] and ECSSD [57] datasets with DUTS-TE [56] as the test set because the trained model of 2-D RGB saliency detector [22] is not available. We use the best performing model with a mean absolute error (MAE) of 0.0698 as the base model even though this best model does not achieve the performance measures reported in [22], i.e., (MAE= 0.0405). Then, we train the FEE module with the overall architecture shown in Figure 1 using the light field dataset with the 2-D saliency detector frozen for 10 epochs. Finally, we train both FEE module and the 2-D saliency detector for another 40 epochs. For all the training, we employ the SGD optimizer [58] with a momentum of 0.9, decay of 0, and initial learning rate of 10^{-2} with a batch size 8. We use the loss function used in [22], i.e.,

$$L = -\sum_{i=1}^{B} (\alpha_s Y_i \log(P_i)) + (1 - \alpha_s)(1 - Y_i) \log(1 - P_i), \tag{1}$$

where P_i is the predicted saliency map, Y_i is the ground truth saliency map, B is the batch size, and $\alpha_s = 0.528$ [22].

4.2. Comparison with State-of-the-Art Light Field Saliency Detectors

We employ the evaluation metrics F_β measure (with $\beta^2 = 0.3$ as suggested in [59], MAE, and F_β^w measure to compare the performance of the saliency detectors. The metrics F_β, F_β^w and MAE are, respectively, defined as

$$F_\beta = \frac{(1 + \beta^2) \times Precision \times Recall}{(\beta^2 \times Precision) + Recall}, \tag{2}$$

$$F_\beta^w = \frac{(1 + \beta^2) \times Precision^w \times Recall^w}{\beta^2 \times Precision^w + Recall^w}, \tag{3}$$

$$MAE = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} |P(i, j) - G(i, j)|, \tag{4}$$

where $P(i, j)$ and $G(i, j)$ are the output saliency map of a saliency detector and the ground truth saliency map, respectively, and w is an Euclidean distance based weighting function [60]. We present the performance achieved with the proposed, LFNet [23] and MTCN [25] light field saliency detectors and five other 2-D saliency detectors in LYTRO ILLUM [23] dataset in Table 1. Accordingly, performance of our saliency detector is superior compared to LFNet while is slightly behind compared to MTCN in terms of all the three metrics. We show the saliency maps of twelve light fields obtained with the proposed, LFNet [23] and MTCN [25] light field saliency detectors in Figure 5 for qualitative comparison. Our saliency maps are closer to the ground truth compared to those of LFNet and comparable with the salience maps achieved with

Table 1: Comparison with state-of-the-art light field saliency detectors, LFNet [23] , MTCN [25] and RGB saliency detectors, NLDF [49], PAGE-Net[50], GCPANet [47] and SODGAN [51]. Our results surpass LFNet [23], DLFS[45], and are slightly behind MTCN [25] in LYTRO-ILLUM[23] dataset

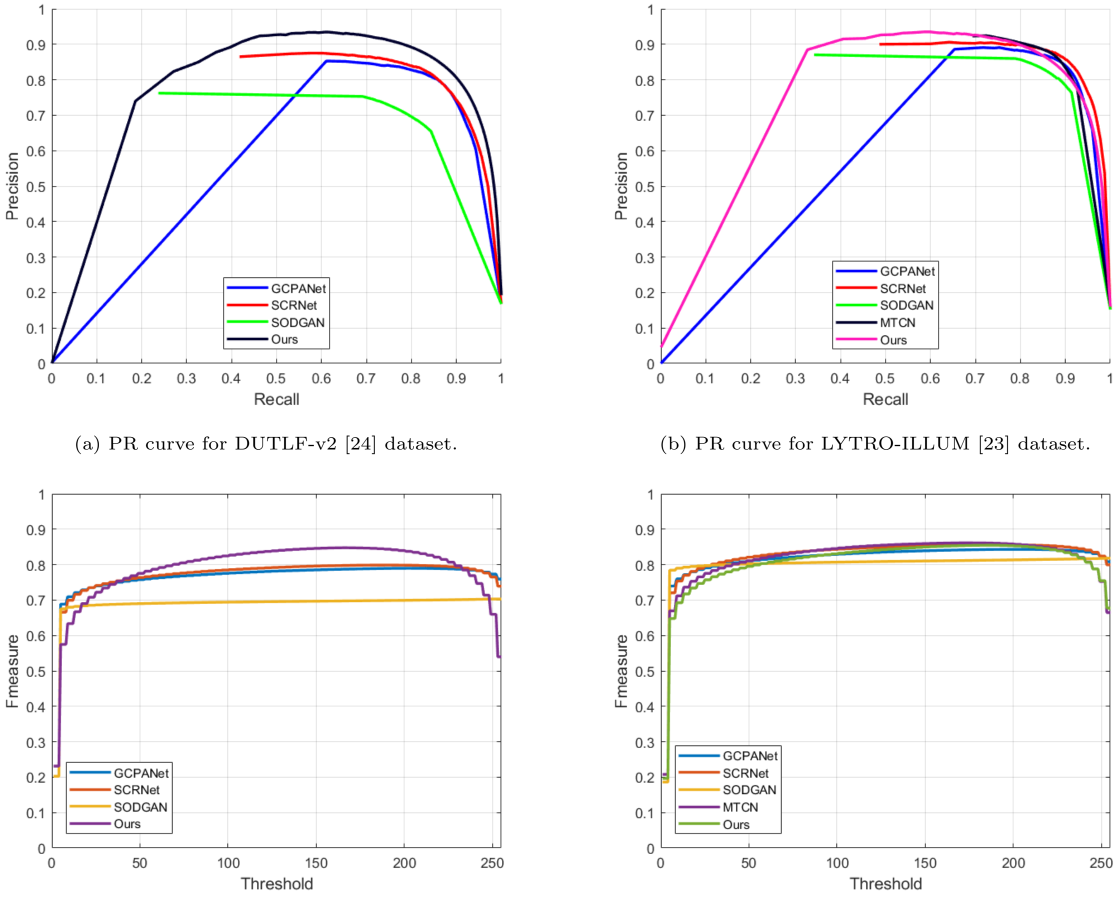
	2-D						Light field		
Metric	NDLF[49]	PAGE-Net [50]	GCPA Net [47]	SCRNet [46]	SODGAN [51]	DLFS [45]	LFNet [23]	MTCN [25]	Ours
F_β	0.7866	0.8047	0.8306	0.8473	0.8313	0.7391	0.8116	0.8729	0.8558
F_β^w	0.7299	0.7826	0.8100	0.8097	0.7969	0.6655	0.7540	0.8534	0.7671
MAE	0.0764	0.0723	0.0580	0.0551	0.0624	0.0843	0.0551	0.0483	0.0541

Table 2: Comparison with state-of-the-art lightfield saliency detectors in DUTLF-v2[24] dataset and our model has comparable performance and lags slightly behind Teacher model[24] surpassing all the other algorithms in terms of F_β measure.

	2-D		RGB-D			Focal stack		Light field	
Metric	DLFS [45]	Student [24]	DisenFusion[52]	ATAFNet[53]	CPFP[54]	Teacher [24]	MoLF [55]	LFNet [23]	Ours
F_β	0.684	0.813	0.686	0.808	0.707	0.852	0.723	0.803	0.8491
F_β^w	0.641	0.771	0.636	0.775	0.629	0.792	0.709	0.786	0.7279
MAE	0.087	0.055	0.093	0.051	0.075	0.050	0.065	0.049	0.052
Size(M)	119	47	166	291.5	278	92.5	186.6	175.8	63

MTCN. Furthermore, our method greatly outperforms the accuracy obtained with the base model, i.e., the 2-D saliency detector in [22]. We also present the performance of the proposed light field saliency detector achieved for the DUTLF-v2 [24] dataset in comparison to state-of-the-art 2-D saliency detectors, DLFS [45] and student networks proposed in [24] and light field saliency detectors, LFNet [23], DisenFusion [52], ATAFNet [53], CPFP [54], MOLF [55] and teacher network proposed in [24] in Table 2. Our method achieves second place in F_β measure and achieves comparable performance in other measures. More importantly, our model is much smaller in size compared to other models except student [24] network. Accordingly, our methods provides *a significant reduction in memory requirement*, especially compared to state-of-the-art light field saliency detectors of all the three categories: focal stack, RGB-D and light field. Therefore, our model is appropriate to be implemented in for resource constrained devices. Furthermore, we present saliency maps of our model for four light fields in Figure 6. Note that outputs of our model are more closer to the ground truth compared to those of the base model. In addition, we show the PR and F-measure curves for the models GCPANet [47], SCRNet [46], SODGAN [51], and MTCN [25] obtained with the LYTRO-ILLUM [23] and DUTLF-v2 [24] datasets in Figure 4. The PR curve of our method is better or on par with the state-of-the-art light field and 2-D saliency detectors for both datasets.

We present the computational time required by each light field and RGB saliency detector to process a light field in the LYTRO ILLUM dataset. We present the computational time required by each saliency detector in Table 3 for both CPU and GPU implementations. Here, the computational time required by each saliency detector implemented on Tesla P100 GPU are obtained from [25]. We mainly consider saliency detectors LFNet [23] and MTCN [25], which employ the full light field, for a fair comparison with the proposed saliency detection method. Our saliency detector is 25 *times faster* than the LFNet in the



(a) PR curve for DUTLF-v2 [24] dataset.

(b) PR curve for LYTRO-ILLUM [23] dataset.

(c) F-measure curve for DUTLF-v2 [24] dataset.

(d) F-measure curve for LYTRO-ILLUM [23] dataset.

Figure 4: PR and F-measure curves of the proposed method for the DUTLF-v2 [24] and LYTRO-ILLUM [23] datasets.

CPU implementation, and *require* 55% *and* 40% *less time* compared to LFNet and MTCN, respectively, for GPU implementation. Here, we present an approximated value for MTCN obtained based on the ratios of computational times reported in [25] for an implementation using a Nvidia Tesla P100 GPU and inference times for the same models in an Nvidia RTX 2080Ti GPU. In particular, we employ the closest available ratio to the median of inference time ratios (2.25 [51], 2.51 [47], 3.16 [23], and 3.80 [46]) between two GPUs which is 3.16. Accordingly, the proposed light field saliency detector *significantly outperforms* state-of-the-art light field saliency detectors, and achieves *near real-time* processing even in the CPU implementation.

4.3. Ablation Study

In this section, we present the comparison between the base model [22] and our architecture with the FEE module. In Table 4, we compare the performance increment obtained through the FEE block in LYTRO ILLUM [23] and DUTLF [24] datasets. As we can see from the results, FEE model provides a *significant boost* in performance compared to the base model [22] in both cases. It is worthwhile to note

Table 3: Computational time required to process a light field: our saliency detector is significantly faster than the state-of-the-art light field saliency detectors.

Method	i9-9900K	Tesla P100	RTX-2080Ti
PAGE-Net[50]	-	0.0822 s	-
NDLF[49]	-	0.0818 s	-
SODGAN[51]	0.3916 s	0.4078 s	0.1812 s
DLFS[45]	-	0.4336 s	-
GCPANet[47]	0.1957 s	0.0400 s	0.0159 s
SCRNET [46]	0.1071 s	0.0578 s	0.0152 s
LFNet [23]	10.4813 s	1.6820 s	0.5321 s
MTCN [25]	-	1.2610 s	0.3989* s
Ours	0.4175 s	0.7526* s	0.2381 s

*These values are approximated using the results in [25] and ours using a linear mapping.

Table 4: Comparison of F_β scores between base-model and the neural network.As depicted in the table, addition of FEE module gives a huge boost to the performance of the model for both datasets

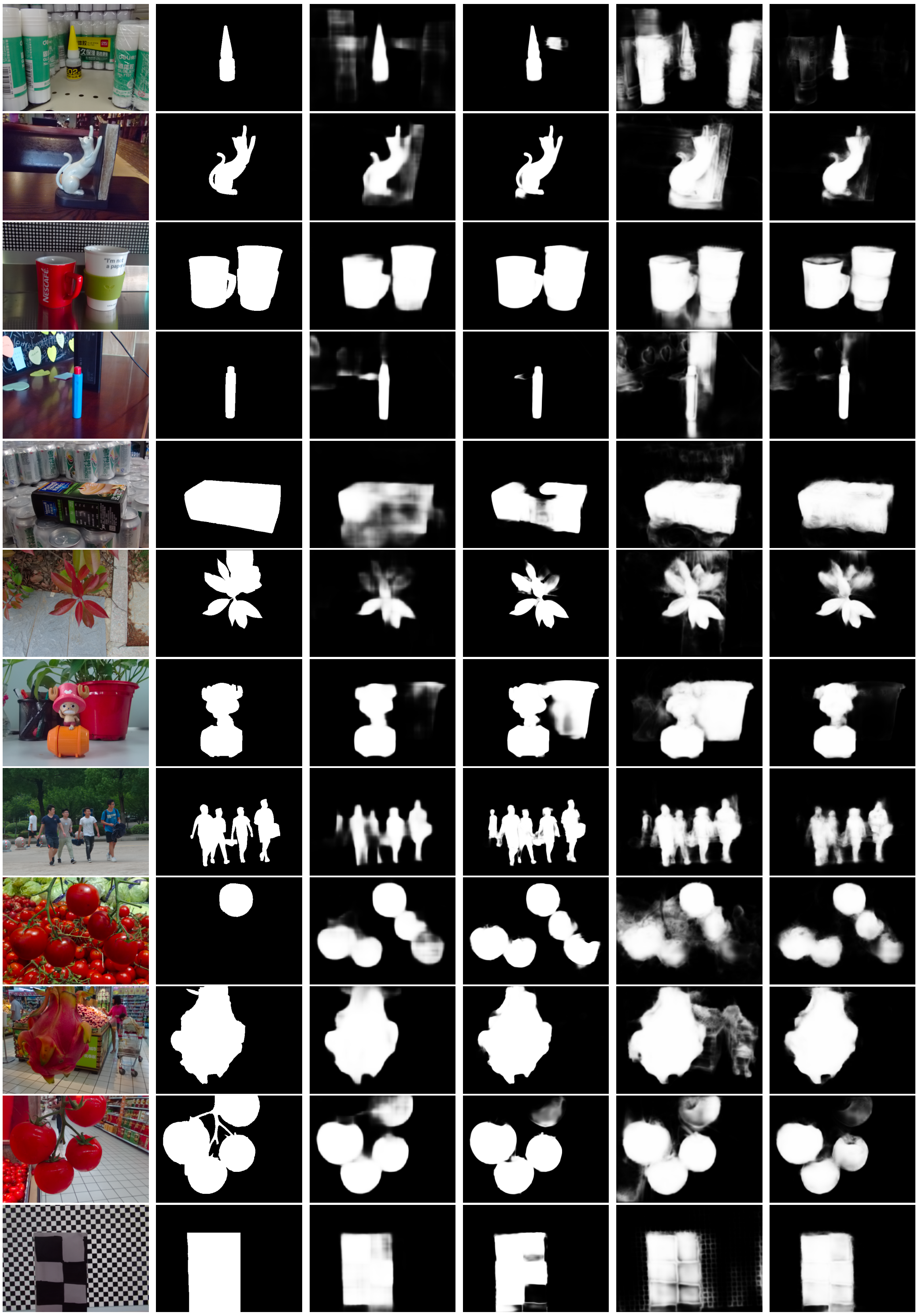
Method	LYTRO ILLUM	DUTLF-V2
Base model	0.7322	0.7275
FEE + Base model	0.8558	0.8491

that the F_β measure of our method could be further improved by employing the original base model with MAE= 0.0405 [22] than our trained model with MAE= 0.0698. In Figure 7, we present the scaled outputs from the FEE block for light fields of the LYTRO ILLUM dataset [23]. We can observe in the outputs of the FEE module that the regions with the higher disparity, i.e., salient objects in the foreground, are emphasized from the background by a border. This verifies that FEE module successfully encodes features of a light field required for salient object detection. This feature embedding helps to accurately segment the salient regions while avoiding computationally heavy backbones.

5. Conclusion and Future Work

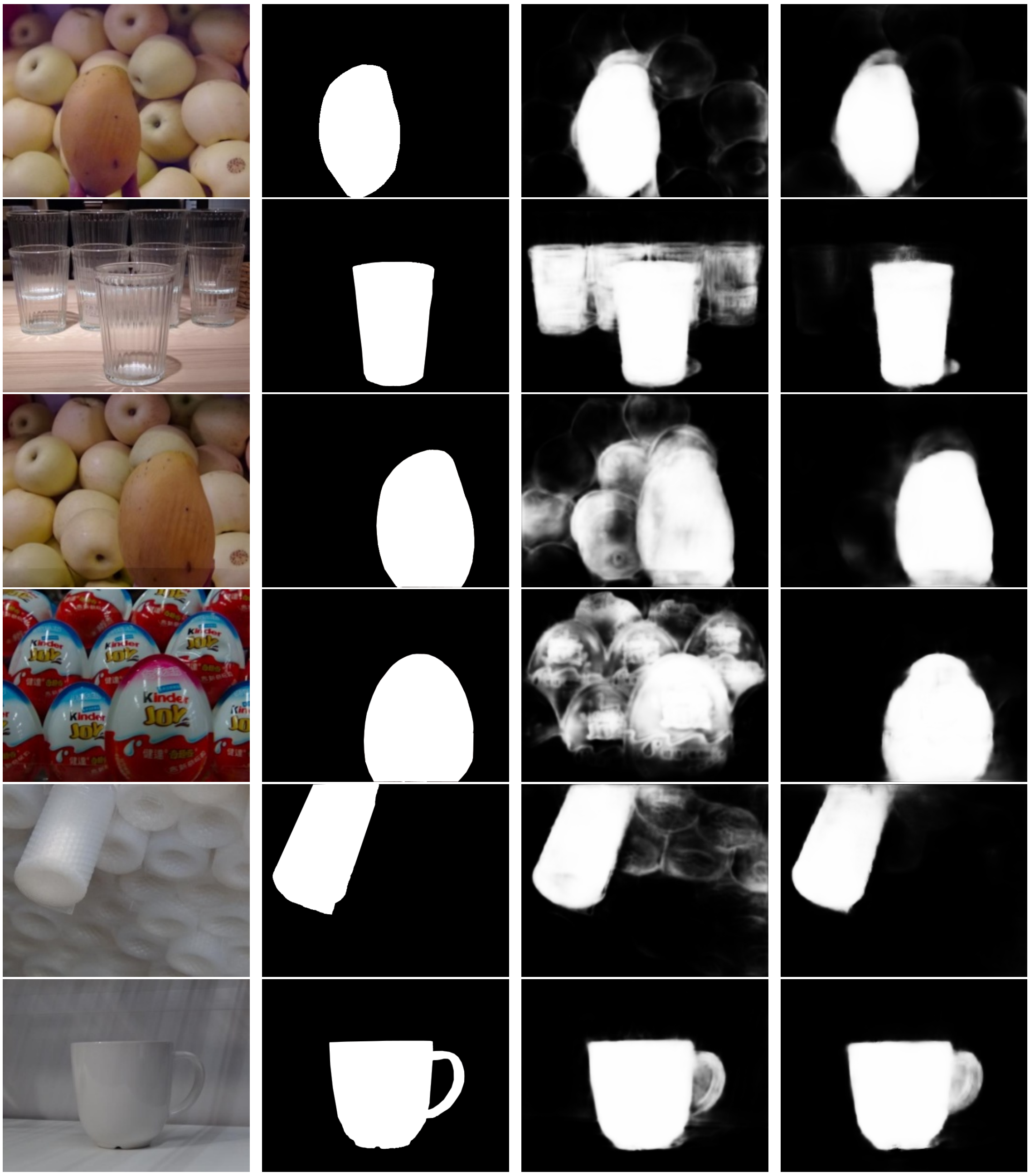
We proposed a fast and accurate light field saliency detector that feeds carefully computed light field features to a saliency detector with an attention mechanism. It is fast and runs on an i9 CPU at approximately 2 light fields/s and on a 2080Ti GPU at 4 light fields/s leading to near real-time processing. Furthermore, the memory requirement of our model is significantly lower compared to state-of-the-art light field saliency detectors making our model appropriate for resource constrained devices. The accuracy of our model surpasses most of the existing methods, and is only slightly inferior to a very recent work. The speed is due to faster feature extraction that constrains light field processing only to the FEE module and using a single stream without resorting to recurrent networks. The high accuracy is due to the light field saliency specific feature extractor and the use of an attention mechanism. Our work brings light field saliency detection closer to real-time implementations which would enable, e.g., cameras to refocus on objects of interest.

Future directions include making the network faster and more accurate by changing or improving the 2-D detector backbone and FEE module. Adapting this method to other computer vision tasks which benefit from the angular information embedded in light fields and lack reasonably-sized datasets—such as, material recognition, segmentation, and object detection—which use 2-D-input neural networks would also be interesting.



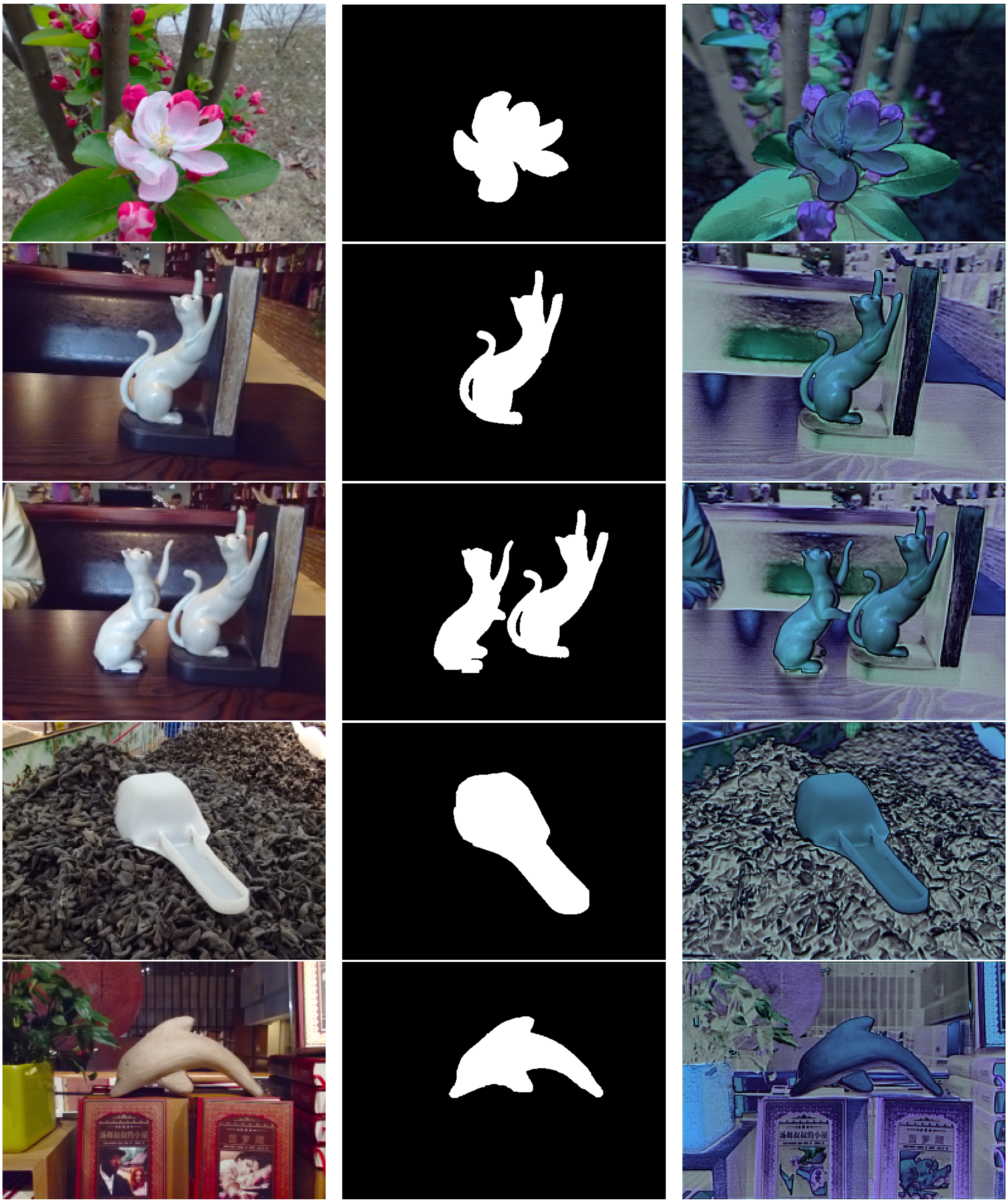
(a) Center SAI (b) GT (c) LFNet [23] (d) MTCN [25] (e) Base [22] (f) Ours

Figure 5: Comparison of saliency maps: (a) centre sub-aperture image (SAI) of the light field, (b) ground truth (GT), (c) LFNet results [23], (d) our results. Our saliency maps are closer to the ground truth compared to those of LFNet[23] and base model[22]



(a) Center SAI (b) GT (c) Base [22] (d) Ours

Figure 6: Comparison of saliency maps: (a) centre sub-aperture image (SAI) of the light field, (b) ground truth (GT), (c) base network, (d) our results in DUTLF-v2 dataset.As it can be seen in the results, our outputs are more closer to the ground truth compared to the base model[22] outputs.



(a) Center SAI

(b) GT

(c) FEE

Figure 7: This figure contains (a) center sub aperture image, (b) ground truth and (c) scaled outputs from the FEE module. We observe that the regions with the higher disparity are emphasized from the background by borders. This verifies that the FEE module successfully encodes the features required for for salient object detection.

References

[1] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, P. Hanrahan, Light field photography with a hand-held plenoptic camera, Computer science technical report, Stanford Univ. (2005).

[2] D. G. Dansereau, O. Pizarro, S. B. Williams, Linear volumetric focus for light field cameras, ACM Trans. Graph. 34 (2) (2015) 15:1–15:20.

[3] S. S. Jayaweera, C. U. S. Edussooriya, C. Wijenayake, P. Agathoklis, L. Bruton, Multi-volumetric refocusing of light fields 28 (2021) 31–35.

[4] D. Dansereau, L. T. Bruton, A 4-D dual-fan filter bank for depth filtering in light fields, IEEE Trans. Signal Process. 55 (2) (2007) 542–549.

[5] C. U. S. Edussooriya, D. G. Dansereau, L. T. Bruton, P. Agathoklis, Five-dimensional depth-velocity filtering for enhancing moving objects in light field videos 63 (8) (2015) 2151–2163.

[6] N. Liyanage, C. Wijenayake, C. Edussooriya, A. Madanayake, P. Agathoklis, L. Bruton, E. Ambikairajah, Multi-depth filtering and occlusion suppression in 4-D light fields: Algorithms and architectures, Signal Process. 167 (2020) 1–13.

[7] M. Levoy, P. Hanrahan, Light field rendering, in: Proc. Annu. Conf. Comput. Graph., 1996, pp. 31–42.

[8] D. G. Dansereau, I. Mahon, O. Pizarro, S. B. Williams, Plenoptic flow: Closed-form visual odometry for light field cameras, in: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., 2011, pp. 4455–4462.

[9] F. Dong, S.-H. Ieng, X. Savatier, R. Etienne-Cummings, R. Benosman, Plenoptic cameras in real-time robotics, Int J. Rob. Res. 32 (2) (2013) 206–217.

[10] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, Y. Liu, Light field image processing: An overview, IEEE J. Sel. Topics Signal Process. 11 (7) (2017) 926–954.

[11] J. Yu, A light-field journey to virtual reality, IEEE Multimedia Mag. 24 (2) (2017) 104–112.

[12] N. Zeller, F. Quint, U. Stilla, From the calibration of a light-field camera to direct plenoptic odometry, IEEE J. Sel. Topics Signal Process. 11 (7) (2017) 1004–1019.

[13] H. Lu, Y. Li, T. Uemura, H. Kim, S. Serikawa, Low illumination underwater light field images reconstruction using deep convolutional neural networks, Future Gener. Comput. Syst. 82 (2018) 142–148.

[14] D. G. Dansereau, B. Girod, G. Wetzstein, LiFF: Light field features in scale and depth, in: Proc. IEEE/CVF Conf. Comput. Vision and Pattern Recog. (CVPR), 2019, pp. 8042–8051.

[15] D. G. Dansereau, S. B. Williams, Seabed modeling and distractor extraction for mobile auvs using light field filtering, in: 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 1634–1639. `doi:10.1109/ICRA.2011.5979852`.

[16] N. Li, J. Ye, Y. Ji, H. Ling, J. Yu, Saliency detection on light field, IEEE Trans. Pattern Anal. Mach. Intell. 39 (8) (2017) 1605–1616.

[17] J. Zhang, M. Wang, J. Gao, Y. Wang, X. Zhang, X. Wu, Saliency detection with a deeper investigation of light field., in: Int. Jt. Conf. Artif. Intell. (IJCAI), 2015, pp. 2212–2218.

[18] M. W. Tao, S. Hadap, J. Malik, R. Ramamoorthi, Depth from combining defocus and correspondence using light-field cameras, in: Proc. of IEEE Int. Conf. on Comput. Vision(ICCV), 2013, pp. 673–680.

[19] T.-C. Wang, A. A. Efros, R. Ramamoorthi, Occlusion-aware depth estimation using light-field cameras, in: Proc. of IEEE Int. Conf. on Comput. Vision(ICCV), 2015, pp. 3487–3495.

[20] J. Chen, J. Hou, Y. Ni, L.-P. Chau, Accurate light field depth estimation with superpixel regularization over partially occluded regions, IEEE Trans. Image Process. 27 (10) (2018) 4889–4900.

[21] T. Wang, Y. Piao, X. Li, L. Zhang, H. Lu, Deep learning for light field saliency detection, in: Proc. of the IEEE/CVF Int. Conf. Comput. Vision (ICCV), 2019, pp. 8838–8848.

[22] T. Zhao, X. Wu, Pyramid feature attention network for saliency detection, in: Proc. of IEEE Conf. Comput. Vision and Pattern Recogn. (CVPR), 2019, pp. 3085–3094.

[23] J. Zhang, Y. Liu, S. Zhang, R. Poppe, M. Wang, Light field saliency detection with deep convolutional networks, IEEE Trans. Image Process. 29 (2020) 4421–4434.

[24] Y. Piao, Z. Rong, S. Xu, M. Zhang, H. Lu, Dut-lfsaliency: Versatile dataset and light field-to-rgb saliency detection, ArXiv abs/2012.15124.

[25] Q. Zhang, S. Wang, X. Wang, Z. Sun, S. Kwong, J. Jiang, A multi-task collaborative network for light field salient object detection, IEEE Trans. Circuits Syst. Video Technol.

[26] C. Yang, L. Zhang, H. Lu, X. Ruan, M.-H. Yang, Saliency detection via graph-based manifold ranking, in: 2013 IEEE Conf. Comput. Vision and Pattern Recogn. (CVPR), 2013, pp. 3166–3173. `doi:10.1109/CVPR.2013.407`.

[27] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, S.-M. Hu, Global contrast based salient region detection, IEEE Trans. Pattern Anal. Mach. Intell. 37 (3) (2015) 569–582. `doi:10.1109/TPAMI.2014.2345401`.

[28] F. Perazzi, P. Krähenbühl, Y. Pritch, A. Hornung, Saliency filters: Contrast based filtering for salient region detection, in: 2012 IEEE Conf. Comput. Vision and Pattern Recogn. (CVPR), 2012, pp. 733–740. `doi:10.1109/CVPR.2012.6247743`.

[29] W. Zhu, S. Liang, Y. Wei, J. Sun, Saliency optimization from robust background detection, in: 2014 IEEE Conf. Comput. Vision and Pattern Recogn. (CVPR), 2014, pp. 2814–2821. `doi:10.1109/CVPR.2014.360`.

[30] A. Gupta, A. Seal, P. Khanna, O. Krejcar, A. Yazidi, AWkS: Adaptive, weighted k-means-based superpixels for improved saliency detection, Pattern Analysis and Applications 24 (2021) 1–15. `doi:10.1007/s10044-020-00925-1`.

[31] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proc. Int. Conf. Learning Represent., 2015, pp. 1–14.

[32] X. Zhang, T. Wang, J. Qi, H. Lu, G. Wang, Progressive attention guided recurrent network for salient object detection, in: Proc. of the IEEE Conf. Comput. Vision and Pattern Recogn. (CVPR), 2018, pp. 714–722.

[33] G. Li, Y. Yu, Deep contrast learning for salient object detection, in: Proc. of the IEEE Conf. Comput. Vision and Pattern Recogn. (CVPR), 2016, pp. 478–487.

[34] R. Zhao, W. Ouyang, H. Li, X. Wang, Saliency detection by multi-context deep learning, in: Proc. of the IEEE Conf. Comput. Vision and Pattern Recogn. (CVPR), 2015, pp. 1265–1274.

[35] N. Liu, J. Han, M. Yang, Picanet: Learning pixel-wise contextual attention for saliency detection, in: Proc. of the IEEE Conf. Comput. Vision and Pattern Recogn. (CVPR), 2018, pp. 3089–3098.

[36] A. K. Gupta, A. Seal, P. Khanna, A. Yazidi, O. Krejcar, Gated contextual features for salient object detection, IEEE Transactions on Instrumentation and Measurement 70 (2021) 1–13. `doi:10.1109/TIM.2021.3064423`.

[37] L. Wang, L. Wang, H. Lu, P. Zhang, X. Ruan, Saliency detection with recurrent fully convolutional networks, in: Proc. European Conf. Comput Vision, 2016, pp. 825–841.

[38] J. Kuen, Z. Wang, G. Wang, Recurrent attentional networks for saliency detection, in: Proc. of the IEEE Conf. comput. Vision and Pattern Recogn. (CVPR), 2016, pp. 3668–3677.

[39] A. K. Gupta, A. Seal, P. Khanna, E. Herrera-Viedma, O. Krejcar, ALMNet: Adjacent layer driven multiscale features for salient object detection, IEEE Transactions on Instrumentation and Measurement 70 (2021) 1–14. `doi:10.1109/TIM.2021.3108503`.

[40] A. K. Gupta, A. Seal, M. Prasad, P. Khanna, Salient object detection techniques in computer vision—a survey, Entropy 22 (10). `doi:10.3390/e22101174`.
URL `https://www.mdpi.com/1099-4300/22/10/1174`

[41] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, R. Yang, Salient object detection in the deep learning era: An in-depth survey, IEEE Transactions on Pattern Analysis and Machine Intelligence PP (2021) 1–1. `doi:10.1109/TPAMI.2021.3051099`.

[42] A. Borji, Saliency prediction in the deep learning era: Successes and limitations, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (2) (2021) 679–700.

[43] J. Zhang, M. Wang, L. Lin, X. Yang, J. Gao, Y. Rui, Saliency detection on light field: A multi-cue approach, ACM Trans.

Multimedia Comput., Commun., and Appl. 13 (3) (2017) 1–22.

[44] N. Li, B. Sun, J. Yu, A weighted sparse coding framework for saliency detection, in: Proc. of IEEE Conf. Comput. Vision and Pattern Recogn. (CVPR), 2015, pp. 5216–5223.

[45] Y. Piao, Z. Rong, M. Zhang, X. Li, H. Lu, Deep light-field-driven saliency detection from a single view., in: Int. Jt. Conf. Artif. Intell. (IJCAI), 2019, pp. 904–911.

[46] Z. Wu, L. Su, Q. Huang, Stacked cross refinement network for edge-aware salient object detection, in: 2019 IEEE/CVF Int. Conf. Comput. Vision, (ICCV), 2019, pp. 7263–7272. doi:10.1109/ICCV.2019.00736.

[47] Z. Chen, Q. Xu, R. Cong, Q. Huang, Global context-aware progressive aggregation network for salient object detection, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020, pp. 10599–10606.

[48] N. Li, J. Ye, Y. Ji, H. Ling, J. Yu, Saliency detection on light field, in: Proc. of IEEE Conf. Comput. Vision and Pattern Recogn.(CVPR), 2014, pp. 2806–2813.

[49] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, P.-M. Jodoin, Non-local deep features for salient object detection, in: 2017 IEEE Conf. Comput. Vision and Pattern Recogn. (CVPR), 2017, pp. 6593–6601. doi:10.1109/CVPR.2017.698.

[50] W. Wang, S. Zhao, J. Shen, S. C. H. Hoi, A. Borji, Salient object detection with pyramid attention and salient edges, in: 2019 IEEE/CVF Conf. Comput. Vision and Pattern Recogn. (CVPR), 2019, pp. 1448–1457. doi:10.1109/CVPR.2019.00154.

[51] Y. Wu, Z. Liu, X. Zhou, Saliency detection using adversarial learning networks, J. Vis. Commun. Image Represent. 67 (2020) 102761.

[52] H. Chen, Y. Deng, Y. Li, T.-Y. Hung, G. Lin, Rgbd salient object detection via disentangled cross-modal fusion, IEEE Trans. Image Process. 29 (2020) 8407–8416. doi:10.1109/TIP.2020.3014734.

[53] M. Zhang, Y. Zhang, Y. Piao, B. Hu, H. Lu, Feature reintegration over differential treatment: A top-down and adaptive fusion network for rgb-d salient object detection, in: Proceedings of the 28th ACM ACM Int. Conf. Multimed., MM '20, 2020, p. 4107–4115. doi:10.1145/3394171.3413969.

[54] J.-X. Zhao, Y. Cao, D.-P. Fan, M.-M. Cheng, X.-Y. Li, L. Zhang, Contrast prior and fluid pyramid integration for rgbd salient object detection, in: 2019 IEEE/CVF Conf. Comput. Vision and Pattern Recogn. (CVPR), 2019, pp. 3922–3931. doi:10.1109/CVPR.2019.00405.

[55] M. Zhang, J. Li, W. Ji, Y. Piao, H. Lu, Memory-oriented decoder for light field salient object detection, in: Adv. Neural Inf. Process. Syst., 2019, pp. 896–906.

[56] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, X. Ruan, Learning to detect salient objects with image-level supervision, in: Proc. of the IEEE Conf. Comput. Vision and Pattern Recogn. (CVPR), 2017, pp. 136–145.

[57] Q. Yan, L. Xu, J. Shi, J. Jia, Hierarchical saliency detection, in: 2013 IEEE IEEE Conf. Comput. Vision and Pattern Recogn. (CVPR), 2013, pp. 1155–1162. doi:10.1109/CVPR.2013.153.

[58] I. Sutskever, J. Martens, G. Dahl, G. Hinton, On the importance of initialization and momentum in deep learning, in: Proceedings of the 30th Int. Conf. Mach. Learn., 2013, pp. 1139–1147.

[59] R. Achanta, S. Hemami, F. Estrada, S. Susstrunk, Frequency-tuned salient region detection, in: IEEE Conf. Comput. Vision and Pattern Recogn. (CVPR), 2009, pp. 1597–1604.

[60] R. Margolin, L. Zelnik-Manor, A. Tal, How to evaluate foreground maps, in: 2014 IEEE Conf. Comput. Vision and Pattern Recogn.(CVPR), 2014, pp. 248–255. doi:10.1109/CVPR.2014.39.