INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Light Field Saliency Detection using Deep Learning: A Comparative Study

Ronielson Baptista Lima

Master in Computer Engineering

Supervisor:
PhD, Caroline Conti, Assistant Professor,
ISCTE-IUL

Co-supervisor:
PhD, Paulo Jorge Lourenço Nunes, Associate Professor,
ISCTE-IUL

October, 2022

TECNOLOGIAS
E ARQUITETURA

Department of Information Science and Technology (DCTI)

Light Field Saliency Detection using Deep Learning: A Comparative Study

Ronielson Baptista Lima

Master in Computer Engineering

Supervisor:
PhD, Caroline Conti, Assistant Professor,
ISCTE-IUL

Co-supervisor:
PhD, Paulo Jorge Lourenço Nunes, Associate Professor,
ISCTE-IUL

October, 2022

I dedicate this work to my mother, Germana Maria Lima (in memorial), who is no longer with us but continues to be my greatest strength in life. Eternal longing.

Acknowledgments

The achievement of this stage would not have been possible without several people's collaboration, help, care and dedication throughout my training. For this very reason, I do not want to miss this opportunity to thank all those who, directly or indirectly, have contributed to my success and my arrival here.

First of all, I would like to thank God for my life and for allowing me to overcome all the obstacles encountered during the course of this work.

To my supervisor and my co-supervisor, who conducted the work with patience and dedication, always available to share all their vast knowledge.

To Instituto de Telecomunicações (IT) for supporting this work.

To all my professors of the Master's degree in Computer Engineering of the ISCTE-IUL University for the excellence of the technical quality of each one.

To my parents, Marcos Baptista Lima and Germana Maria Lima, for the support and incentive that served as the foundation for my accomplishments.

To my siblings Francisco Lima, Rosa Morais, Isabel Lima, Lúcio Lima, Constantino Lima, Adilson Lima, Anísio Lima, Estanislau Lima, Herculano Lima, Evanisia Lima, and Pedro Lima for their companionship, complicity, and support in all the delicate moments of my life.

To my girlfriend Libânia Silva for the understanding and patience shown during the project period.

Finally, to all those who contributed, in some way, to the realization of this work.

Resumo

Salient Object Detection (SOD) despertou um interesse crescente na investigação na última década, e actualmente se baseia em aprendizagem automática e imagens 2D. Recentemente, imagens *Light Field* (LF) apresentam uma alternativa atraente à imagens 2D, devido ao fato que LF contêm informações abrangente de cenas naturais que beneficiam SOD. A utilização das imagens LF para melhorar a detecção da saliência em relação às imagens 2D convencionais é uma tendência emergente. Nos últimos anos, *Deep Learning* (DL) tem sido dominados pelas chamadas *deep Convolutional Neural Networks* (CNNs), que tiram partido da capacidade computacional. Para imagens 2D, os métodos que utilizam CNNs têm alcançado excelentes resultados em várias tarefas, incluindo SOD. Contudo, a utilização de métodos baseados em CNNs para a detecção de saliências em imagens LF não é trivial porque não são explicitamente concebidos para o processamento de conteúdos LF. Além disso, os conjuntos de dados actuais de LF não são suficientemente grandes para treinar CNNs a partir do zero. Portanto, nesta dissertação, analisamos e comparamos a eficiência de cinco modelos existentes para LF SOD que são baseados em DL. Além disso, propomos um método alternativo baseado em LF que utiliza imagens LF como entrada para verificar as vantagens de imagem LF em relação à imagem 2D para a tarefa de SOD. Conduzimos os experimentos em dois conjuntos de dados com diferentes propriedades para validar os modelos escolhidos assim como o modelo proposto. A análise experimental mostrou que os modelos baseados em DL utilizando imagens LF superam o modelo convencional 2D.

Palavras-chave: Campo de Luz, Detecção de Objetos Salientes, Aprendizagem Profunda, Redes Neuronais Convolucionais

Abstract

Salient Object Detection (SOD) has drawn an increasing amount of research interest in the past decade and is currently heavily based on Machine Learning (ML) and 2D imaging. Recently, Light Field (LF) imaging became an attractive alternative to 2D imaging due to LF recording comprehensive information of natural scenes that benefit SOD. Using the LF imaging instead conventional 2D imaging to improve SOD is, therefore, an emerging trend. Over the past years, Deep Learning (DL) methods have been dominated by the so-called deep Convolutional Neural Networks (CNNs), which take advantage of the accompanying advances on computing power and data availability. For 2D imaging, methods using CNNs have achieved excellent results at various tasks, including saliency detection. However, using CNN-based methods for LF-based saliency detection is not trivial because these methods are not, typically, explicitly designed for processing LF inputs. In addition, current LF datasets are not sufficiently large to train CNNs from scratch. Therefore, for this dissertation, we analyse and compare the efficiency of five existing models for LF SOD that are based on DL, using common test conditions. Additionally, we propose an alternative LF-based method that uses LF images as input to verify the advantages of LF imaging over 2D imaging for the task of SOD. We conducted the experiments on two different datasets with different properties to validate all five existing approaches and our proposed model. Experimental analysis has shown that models based on DL using the LF imaging outshine the conventional 2D model.

Keywords: Light Field, Salient Object Detection, Deep Learning, Convolutional Neural Networks

Contents

Acknowledgments iii

Resumo v

Abstract vii

List of Figures xi

List of Tables xiii

List of Acronyms xiv

Chapter 1. Introduction 1
1.1. Motivation and Research Context 1
1.2. Research Questions 2
1.3. Objectives of this Dissertation 2
1.4. Outline 3

Chapter 2. Light Field Imaging 5
2.1. Light Field Definition and the Plenoptic Function 5
2.2. 4D Light Field Acquisition 6
2.2.1. Camera Arrays 7
2.2.2. Plenoptic Cameras 7
2.2.3. Other Devices 9
2.3. Light Field Image Representation and Processing 10
2.4. Final Remarks 11

Chapter 3. Background on Deep Learning 13
3.1. Introduction to Deep Learning 13
3.2. Convolutional Neural Networks 14
3.3. Methods for Training Deep Neural Networks 17
3.4. Deep Learning Frameworks 18
3.5. Final Remarks 18

Chapter 4. Literature Review on Salient Object Detection 21
4.1. Introduction to Salient Object Detection 21
4.2. 2D and 3D Salient Object Detection 22

4.2.1. Traditional 2D and 3D SOD Methods 23
4.2.2. Deep Learning 2D and 3D SOD Methods 23
4.3. 4D Light Field Salient Object Detection 25
4.3.1. Traditional 4D Light Field SOD Methods 25
4.3.2. Deep Learning 4D Light Field SOD Methods 26
4.4. Final Remarks 26

Chapter 5. 4D Light Field SOD using Deep Learning: Classifying and Reviewing 27
5.1. Classification Taxonomy 27
5.2. Synergistic Attention Network (SA-Net) 28
5.3. Patch-Aware Network (PANet) 29
5.4. Exploit and Replace Network (ERNet) 30
5.5. Model Angular Changes (MAC) 31
5.6. Deep LF driven Saliency Detection (DLSD) 32
5.7. Deep LF driven Saliency Detection with Multiview (DLSD-MV): A Proposal . . 33
5.8. Final Remarks 33

Chapter 6. Performance Evaluation of 4D LF SOD Deep Learning Models 35
6.1. Experimental Setup 35
6.2. Dataset for Training and Evaluation 36
6.3. Evaluation Metrics 37
6.4. Overall Quantitative Evaluation 39
6.5. Cross Evaluation on LFSD Dataset 40
6.6. Visual Analysis 42
6.7. Single View versus Multiview 4D LF SOD 44
6.8. Final Remarks 45

Chapter 7. Conclusions and Future Work 47
7.1. Conclusions 47
7.2. Future Work 49

Bibliography 51

List of Figures

1.1 Sample challenging images for SOD: (a) Multiple objects, (b) Large objects, (c) Reflection, (d) Small object, (e) Low contrast, and (f) Complex scene. 2

2.1 The 7D Plenoptic Function [**1**]. 5
2.2 4D Plenoptic Function two-plane parameterization [**2**]. 6
2.3 Example of an LF captured with a camera array [**3**]. 7
2.4 Basic imaging process of a Plenoptic Camera [**3**]. 8
2.5 LF captured with a Plenoptic Camera: (a) Microlens images (micro-images), (b) SAIs, and (c) Horizontal EPIs [**3**]. 9
2.6 LF capturing devices: (a) A Lytro Illum Plenoptic Camera. (b) Technicolor camera array prototype [**4**], composed of 16 video cameras. (c) Raytrix camara. 9

3.1 A cartoon drawing of a biological neuron (a) and its mathematical model (b) [**5**]. 14
3.2 Convolutional network LeNet-5 [**6**]. 15

4.1 Typical SOD flowchart. 21
4.2 Classification of SOD methods. 22
4.3 The whole pipeline of the C2FNet model [**7**]. 25

5.1 Architectures of existing deep 4D LF SOD models: (a) Late-fusion scheme, (b) Middle-fusion scheme, (c) Knowledge distillation-based scheme, (d) Single-stream scheme, (e) Reconstruction-based scheme, and (f) proposed architecture for 4D LF SOD (adapted from [**8**]). 27
5.2 The whole pipeline of the SA-Net model [**9**]. 29
5.3 The whole pipeline of the PANet model [**10**]. 30
5.4 The whole pipeline of the ERNet model [**11**]. 31
5.5 The whole pipeline of the MAC model [**12**]. 32
5.6 The whole pipeline of the DLSD model [**13**]. 33
5.7 The whole pipeline of the proposed DLSD-MV model. 34

6.1 Example of the LF data elements in the DUTLF-V2 dataset: (a) All-in-focus, (b) SOD ground-truth, (c) Depth map, (d) Focal stack, (e) Micro-images, and (f) Sub-aperture images (multiview). 36
6.2 Example of the LF data elements in the LFSD dataset: (a) All-in-focus, (b) SOD ground-truth, (c) Depth map and (d) Focal stack. 37

6.3 PR Curve for DUTLF-V2 Dataset. PR curves closer to the upper-right corner correspond to better performances. 40
6.4 PR Curve for LFSD Dataset. PR curves closer to the upper-right corner correspond to better performances. 41
6.5 Comparative results for the five 4D LF SOD models (SA-Net, PANet, ERNet, DLSD, and MAC) and the 2D SOD model (C2FNet) under analysis on the DUTLF-V2 and LFSD datasets: (a) F-measure, (b) S-measure, (c) E-measure and (d) MAE. 42
6.6 Visual comparison of the LF SOD on the DUTLF-V2 dataset. 43
6.7 Visual comparison of the Light Field SOD on the LFSD dataset. 44
6.8 Visual comparison of the DLSD and DLSD-MV models on the DUTLF-V2 dataset. 45

List of Tables

6.1 Hyper-parameters optimization. 35
6.2 Specification of the system used for training. 36
6.3 Overview of the used LF SOD datasets. About the abbreviations: FS=Focal Stacks, DE=Depth maps, MV=Multiview images, ML=Microlens images, GT=Ground-Truth, RAW=Raw LF data. 37
6.4 Quantitative comparisons of E-measure, S-measure, F-measure and MAE scores on the DUTLF-V2 dataset for different models. The best scores are in boldface. 40
6.5 Quantitative comparisons of E-measure, S-measure, F-measure and MAE scores on LFSD dataset for different models. The best scores are in boldface. 41
6.6 Quantitative comparisons of DLSD and DLSD-MV models on the DUTLF-V2 dataset. The best scores are in boldface. 44

List of Acronyms

2D	two-dimensional
3D	three-dimensional
4D	four-dimensional
5D	five-dimensional
7D	seven-dimensional
ACFM	Attention-induced Cross-level Fusion Module
AI	Artificial Intelligence
AiF	All-in-Focus
ANN	Artificial Neural Network
ASPP	Atrous Spatial Pyramid Pooling
C2FNet	Context-aware Cross-level Fusion Network
CNN	Convolutional Neural Network
cuDNN	CUDA Deep Neural Network
DGCM	Dual-branch Global Context Module
DL	Deep Learning
DLSD	Deep LF driven Saliency Detection
DLSD-MV	Deep LF driven Saliency Detection with Multiview
EPIs	Epipolar Plane Images
ERNet	Exploit and Replace Network
FS	Focal Stack
GD	Gradient Descent
GPU	Graphics Processing Unit

ILSVRC12	ImageNet Large Scale Visual Recognition Challenge 2012
LF	Light Field
LFR	Light Field Raw
LL	Lenslet
MAC	Model Angular Changes
MAE	Mean Absolute Error
MFRM	Multi-Focusness Recruiting Modules
MFSM	Multi-Focusness Screening Modules
ML	Machine Learning
MLA	Microlens Array
MLP	Multilayer Perceptron
MSCA	Multi-Scale Channel Attention
MSLM	Multi-source Learning Module
MV	Multiview
MVAM	Multiview Attention Module
MVSD-SNet	MultiView Saliency Detection Sub-network
NLPR	National Laboratory of Pattern Recognition
PANet	Patch-Aware Network
PF	Progressive Fusion
PR	Precision-Recall
RDFD	Region-based Depth Feature Descriptor
ReLU	Rectified Linear Unit
SA	Synergistic Attention
SA-Net	Synergistic Attention Network
SAIs	Sub-Aperture Images
SD	Saliency Detection
SF	Sigmoid Function

SOD Salient Object Detection

SRM Sharpness Recognition Module

CHAPTER 1

Introduction

1.1. Motivation and Research Context

Audio-visual content has an increasing role in modern societies in different activities, such as entertainment, telemedicine, autonomous driving, and so on. These activities are intended to obtain a more immersive representation of the surrounding reality. In this context, recent decades have observed significant research efforts in new visual modalities, such as Light Field (LF) imaging.

LF is a parametric representation of a four-dimensional (4D) light radiation field that contains the position and direction of the light rays in the three-dimensional (3D) space. In other words, the LF contains images of the same object taken at different viewing positions and angles. The concept of an LF was initially used mainly in computer graphics as a powerful tool to describe the scene's appearance. Recently, with the rapid development of LF acquisition and display technologies, the research on LF imaging has been getting more and more attention. Thus, LF is being widely studied in computer vision, pattern recognition, image processing, and in many other real consumer applications, such as virtual reality and augmented reality, cinematography, 3D television, biometric recognition, medical imaging, and so on.

Despite the recent advances in the area of LF imaging [**10**], [**13**], many open research questions remain to be solved. For instance, LF technology may still suffer from crucial problems such as low spatial and/or angular resolution, poor reconstructed depth quality, or large data storage requirements. Lately, huge improvements have been made to bring solutions to the above mentioned LF challenges. However, LF research still needs better theories and methods, targeting more efficient data capture, analysis, processing, and storage. More specifically, in this dissertation, we analyze and compare the efficiency of using LF approaches for the task of Salient Object Detection (SOD). SOD, or simply Saliency Detection (SD), is a computer vision task commonly interpreted as a process that includes two stages: (i) detecting the most salient object, and (ii) segmenting the accurate region of that object. It is a prerequisite for many other computer vision tasks, such as image retrieval, scene classification and semantic segmentation. SOD can greatly benefit from using the richer light information of LFs. For instance, when using LF imaging, the object(s) of interest can be refocused after being acquired. Also, an LF can provide images of a scene from an array of viewpoints, and a scene's depth information can be estimated using LF data. Additionally, LF imaging has shown to perform better for challenging scenes, such as those with similar foreground and background, complex occlusions, multiple objects, large objects or low contrast, as illustrated in Figure 1.1.

Moreover, this dissertation mainly focuses on LF SOD solutions that are based on Deep Learning (DL) methods. DL is a recent sub-field of Machine Learning (ML) whose significant advantage is the ability to induce effective feature representations from data automatically. These learned features have shown the ability to be generalized to various visual recognition tasks. They have displayed exceptional performance in many computer vision and ML problems, for instance, image classification, speech recognition, automated control, object detection, and so on. One particular type of DL approach that has achieved practical success is the Convolutional Neural Network (CNN). CNNs have been acknowledged as the most successful and widely used DL method due to the remarkable results on several computer vision tasks. CNNs have been used in various processing tasks for conventional two-dimensional (2D) and 3Dimaging and also LF imaging [**14**], [**15**]. However, applying a CNN for LF SOD is challenging since CNN models were initially proposed for 2D images as input, and adapting then to deal with the huge amount of LF data is usually not straightforward.

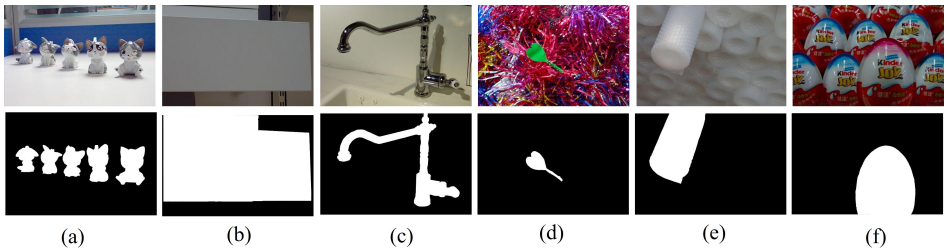


Figure 1.1. Sample challenging images for SOD: (a) Multiple objects, (b) Large objects, (c) Reflection, (d) Small object, (e) Low contrast, and (f) Complex scene.

1.2. Research Questions

As mention above, applying a CNN for LF SOD is a challenging task. First, data redundancy needs to be preserved during the processing stage. Second, although the depth is implicitly embedded in the LF image, it may be unavailable, inconvenient, and/or imperfectly estimated to segment the full image. Furthermore, CNNs are not originally designed to process LF images. Additionally, training CNNs specifically to LF SOD may be limited by the lack of a large-scale LF datasets. Therefore, in this dissertation, we aim to answer the following research questions about how to apply CNNs to LF SOD:

- What are the main advantages of using LF imaging for SOD?
- Does the type of LF data format influence the performance of LF SOD?
- Which existing DL architectures perform better for LF SOD?

1.3. Objectives of this Dissertation

In recent years, DL methods, particularly CNNs, have demonstrated dramatic improvement over traditional methods due to their powerful ability to learn image representations. The main goal of this dissertation is, therefore, to evaluate and compare the efficiency of existing DL methods for the task of LF SOD by using a common dataset with different challenges. An alternative DL

method will also be proposed to further extend this analysis. To achieve this goal, the following specific objectives were defined for this dissertation:

- Perform a bibliographic search to understand the state-of-the-art on LF image processing and DL techniques for SOD;
- Investigate how CNNs used for 2D images can be extended for 4D LF imaging;
- Evaluate the performance of existing DL architectures for the 4D LF SOD;
- Investigate the main advantages of using 4D LF imaging over 2D images for SOD.

1.4. Outline

The rest of this dissertation is organized as follows:

Chapter 2 covers the general aspects in the context of LF imaging to build the necessary foundations to understand this dissertation. First, we present the definition of the LF, followed by the basic concepts of 4D LF acquisition. Then, we briefly describe the fundamental concepts of LF representation and processing.

Chapter 3 briefly describes the fundamental concepts about DL related to this dissertation. First, we briefly present the historical context of DL. In the subsequent sections, we present the concept of CNN. After that, methods for training deep neural networks are presented. The last section presents the DL frameworks used for training the CNN models.

Chapter 4 presents the literature review on SOD. First, we briefly introduce the concept of SOD. After that, we present the traditional and DL methods for 2D and 3D SOD. The subsequent sections show the 4D LF methods using conventional and DL methods.

Chapter 5 presents a classification and review of the network architectures used in this dissertation. First, we introduce the five architectures analysed in this dissertation, which are the most influential for deep neural networks for the LF SOD models in the literature for the last few years. An alternative method for 4D LF SOD is also proposed in the subsequent sections.

Chapter 6 introduces the training configurations and the datasets used for training and testing. Also, we present the most popular evaluation metrics for SOD. After that, the quantitative and qualitative experimental results are presented, as well as its analysis and the most relevant conclusions of the proposed comparative study.

Chapter 7 concludes the dissertation with a summary of its main achievements and and presents a set of suggestions for future work.

CHAPTER 2

Light Field Imaging

This chapter presents the fundamental concepts on LF imaging. Specifically, first, we introduce the formal definition of LF and Plenoptic Function along with the important taxonomy and notations used in this work. Then we discuss different ways for capturing LF data. After this, we describe some of the LF imaging representation and processing limitations.

2.1. Light Field Definition and the Plenoptic Function

The concept of LF was first adopted in 1936 by Gershun for analytically describing the *"light beams that propagate in a straight line through a homogeneous medium and which is the carrier of radiant energy in a space"* [**16**]. From a purely geometric point of view, light is constituted of rays: directed lines in a 3D space with an intensity value or *radiance*. Although an LF may be represented in different formats, the important theory is always the same: it is a illustration of a 3D scene as a collection of light rays coming throughout every point in space and flowing in all possible directions.

With the popularity of computer vision and computer graphics in the early 1990s, the problem of geometrically describing the visible space gained significant importance. For instance, Adelson and Bergen proposed, in [**17**], to define the total distribution of light as a seven-dimensional (7D) function, which models the light rays at every possible location in space (x, y, z), toward every possible direction (θ, ϕ), over any range of wavelengths (λ), and at any time (t) – referred to as the Plenoptic Function (see Figure 2.1):

$$P\left(x, y, z, \theta, \phi, \lambda, t\right) \tag{2.1}$$

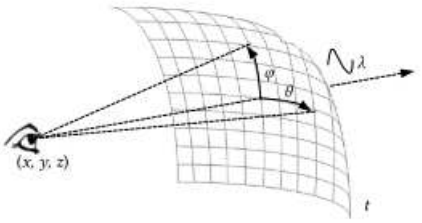


Figure 2.1. The 7D Plenoptic Function [**1**].

Though, such high-dimensional data is challenging to record and handle in practice. Thus, the LF model has been simplified in two ways for practical usage. In the first simplification, the calculated function is presumed to be monochromatic and time-invariant, the wavelength, λ, of

every light ray is recorded separately in different colour channels, and the time sequence can be recorded in distinct frames for a dynamic LF. This way, we can eliminate the wavelength, λ, and time, t, dimensions from the Plenoptic Function, decreasing the model from 7D to five-dimensional (5D).

Levoy *et al.* [**2**] and Gortler *et al.* [**18**], made a second simplification as they realised the 5D representation still included some redundancy and could be reduced to 4D by assuming the radiance of the light rays remains constant along a straight line (in free space). Thus, making a dimension in the 5D Plenoptic Function redundant.

There are three key issues in parameterizing a 4D LF [**2**]: (i) computational efficiency, (ii) control over the ray set, (iii) and uniform sampling of the LF space. The most common 4D LF representation solution is to parameterize the light rays through their intersection coordinates with two planes placed at arbitrary positions, as shown in Figure 2.2.

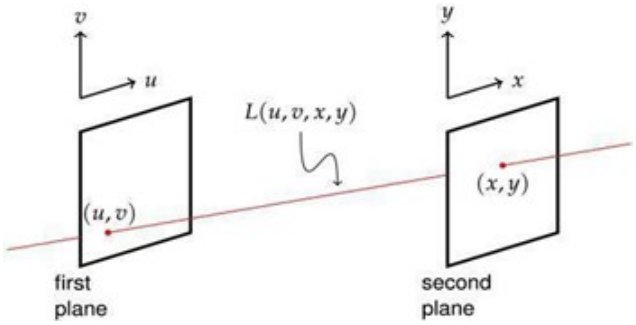


Figure 2.2. 4D Plenoptic Function two-plane parameterization [**2**].

The coordinate system is here denoted by (u, v) for the first plane and (x, y) for the second plane. An oriented light ray identified in the system first traverses the uv plane at coordinate (u, v) and then crosses the xy plane at coordinate (x, y) and is thus denoted by $L(u, v, x, y)$. Thus, the Plenoptic Function describing an LF can be reduced from 7 to 4 dimensions and parameterised by four coordinates (u, v, x, y) [**2**].

2.2. 4D Light Field Acquisition

As mentioned above, due to the high dimensionality of the data, it is difficult to obtain the full Plenoptic Function of a scene. Alternatively, LF mainly focus on the distribution of light rays as a function of position and angle [**2**]. With this lower dimensionality and the progress in hardware and software, today's LF acquisition devices may be portable and commercially available [**19**]. As an example, they have been integrated with microscopy [**20**], allowing rapid scan-less volumetric photographs of biological specimens. The widespread use of LF cameras has allowed several new applications, ranging from their initial purpose of photo-realistic image-based rendering [**2**] to current computer vision applications that use their rich encoded information. These include 3D reconstruction, segmentation and matting, saliency detection [**21**], object detection and recognition, and so on. Therefore, we classify these devices into three broad categories.

2.2.1. Camera Arrays

As their name recommends, camera arrays are matrices of synchronized cameras coordinated on a plane, often at a regular interval. Due to a camera array multiplexes the spatial dimension into the angular dimension, the outcome is a matrix of views with different viewpoints. Considering the *thin lens model* and the case where the cameras are the same and absolutely placed, the group of images is directly a 4D LF where cameras match to angular samples, and each image captures the spatial dimensions. The LF captured by a camera array can be exemplified as a matrix of views, as shown in Figure 2.3.



Figure 2.3. Example of an LF captured with a camera array [**3**].

Most camera arrays have rather large *baselines* and camera resolutions, yielding a spatially dense but angularly sparse LF. Designing a camera array is not trivial. The camera synchronization and their geometrical and color calibration are significant technical challenges. Furthermore, they are bulky and often power-hungry. For these motives, camera arrays are still quite rare, but there are a little notable designs. The most impressive ones on the academic side are the Stanford Camera Arrays [**22**] with their specific synchronization hardware, but other efforts have been proposed as well [**23**], [**24**]. In the industry, we find both experimental video camera arrays [**4**], [**25**] with modern cameras, capturing high resolution and high frame rate synchronized sequences. However, not all camera arrays are expensive and cumbersome. Smaller all-in-one devices can also be assimilated to camera arrays, though they don't have quite the similar baseline like the Pelican Imaging Sensor [**26**].

2.2.2. Plenoptic Cameras

Although the LF formalism we explained is rather recent, the foundation for its capture was laid more than a century ago by Gabriel Lippmann [**27**]. He suggested an imaging device constituted of an array of 12 small lenses on top of a photosensitive plate to capture a scene from different viewpoints. The group of images was then viewed across the same lens, providing a different viewpoint for each eye, depending on the pair of lenses the observer looks into. The new version of such apparatus is the Plenoptic Camera. A Plenoptic Camera consists of a Microlens Array (MLA) put between the main lens and the image sensor, as shown in Figure 2.4. In this way, the Plenoptic Camera captures the complete 4D LF on the sensor plane. Specifically, this is

realized by putting the sensor plane in the focal plane of the microlens and having the microlens focused at infinity. However, it is important to mention that not all Plenoptic Cameras work as specified above. Unlike a regular camera, when two rays come from the main lens arrive at the same sensor spot with different angles, as an alternative to being integrated by one sensor pixel, they are re-oriented to different pixels by the microlens. In other words, the microlens multiplex the rays directly into the sensor. Each image under a microlens, the so-called micro-images, corresponds to a specific beam of light rays coming from a point in scene and passing through the main lens. The Plenoptic Camera has been an innovation in photography, allowing for acquiring 2D images focused at different depths, from a single exposure.

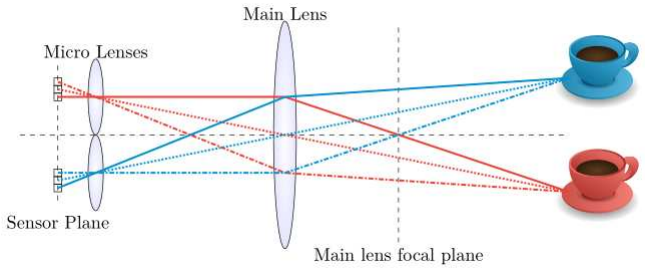


Figure 2.4. Basic imaging process of a Plenoptic Camera [**3**].

In contrast to a camera array, a Plenoptic Camera multiplexes the angular dimension onto the spatial dimensions of the 2D sensor. Hence, it captures what we can denote by $L(u, v, x, y)$. As a consequence, an LF from a Plenoptic Camera is a collection of small microlens images (micro-images), as illustrated in Figure 2.5a, which is difficult to visualize but comprises the different angles. That is why a popular method is to multiplex the captured LF in order to obtain the so-called Sub-Aperture Images (SAIs). It consists simply in gathering pixels with the same relative position in each microlens image and placing them in a new image depending on the microlens image it came from. Doing so, we get an array of images like to a camera array view matrix but with a much smaller baseline, as shown in Figure2.5b. The number of SAIs depends on the number of pixels underneath each microlens, while the view resolution differs on the number of the microlens.

Another way of slicing such LF can be noted $L(u, x)$ and $L(v, y)$. Firstly proposed to analyze a spatio-temporal volume [**28**], Epipolar Plane Images (EPIs) intuitively offer a way to visualize pixel intensities as a function of their horizontal (or vertical) spatial and angular coordinates. The obtained image, as the one in Figure 2.5c for $L(u, x)$, for instance, can be hard to make sense of, but it shows the variations in the spatial position x of a point in the scene while changing its angular position u.

Apart from custom-made Plenoptic Cameras [**29**], there are a few Plenoptic Cameras available in the market. The Lytro Illum[1] camera and Technicolor camera array, as in Figures 2.6a

[1]http://www.lytro.com

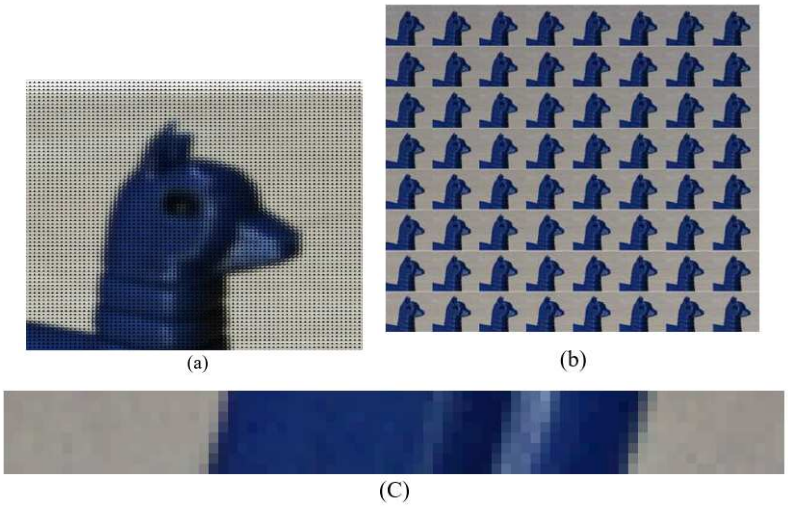


Figure 2.5. LF captured with a Plenoptic Camera: (a) Microlens images (micro-images), (b) SAIs, and (c) Horizontal EPIs [**3**].

and 2.6b, are two affordable cameras focused on the consumer level. Nevertheless, the Raytrix camera, as shown in Figure 2.6c, is the most used camera from the Raytrix company[2].

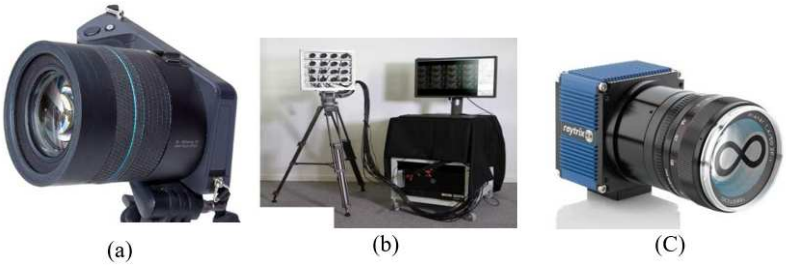


Figure 2.6. LF capturing devices: (a) A Lytro Illum Plenoptic Camera. (b) Technicolor camera array prototype [**4**], composed of 16 video cameras. (c) Raytrix camara.

2.2.3. Other Devices

There are some devices that capture an LF but do not fall into the two above-mentioned categories. Trendy in this class are camera gantries, where a camera is pushed, often along a plane, and shots are taken at standard intervals. Intuitively, they sample the LF in the same way as camera arrays but are comparatively cheaper and somehow easier to execute. For this reason, camera gantries were used before camera arrays [**2**]. The camera resolution and the displacement in each shot dictate the angular and spatial sampling. However, this type of device is limited to capturing static LFs. A sample of camera gantries is the simple but exceptionally cost-effective Stanford Lego gantry[3] . An interesting mix between a camera gantry and a camera array is also proposed by Google[4] for Virtual Reality content generation. Lastly, it is worth mentioning that some more exotic devices can use mirrors instead of lenses to capture the LF [**30**], [**31**], [**32**].

[2]http://www.raytrix.de
[3]http://lightfield.stanford.edu/lfs.html
[4]https://www.blog.google/products/google-vr/experimenting-light-fields

2.3. Light Field Image Representation and Processing

LF imaging is a promising solution over 2D images since it is a richer imaging modality with spatial and angular information that enables new functionalities and provides more immersive and closer to reality multimedia experiences. LF data is stored in files that hold the raw image sensor data, specifically Light Field Raw (LFR) files. The LF data comprises a set of micro-images.

The LF data can be represented in different formats, for instance, including the Focal Stack (FS), All-in-Focus (AiF), Multiview (MV), center-view, MLA (also known as Lenslet (LL) format) and depth maps. The focal stack contains a series of images focusing on different depths. Such images are generated by processing the LF data using digital refocusing techniques [**33**]. After the focal stack is obtained, an AiF can be generated by putting all the clear pixels together, where the associated gradients can estimate the clarity of pixels. Besides, it can also be acquired by computing a weighted average of all the focus slices. In addition to focal stacks, multiview images can also be derived from LF data. By varying the angular direction, multiview images can be synthesized. Especially when the angular direction is equal to that of the central view, the center-view image is achieved. Alternatively, microlens images can be generated by sampling the coordination dimensions. Furthermore, depth maps containing scene depth information can also be estimated from an LF.

Due to the importance of LF, researchers have devoted significant attention to LF processing in recent decades. Due to the enhanced features that LF imaging offers, a vast amount of data is created during the acquisition step. Additionally, the rays are multiplexed by direction into a single sensor. Plenoptic Cameras trade spatial resolution for angular resolution because the distance between the microlens is very small, and their angular sampling is *dense*. On the other hand, because there is a restrict to the number of microlens we can use, the produced images are fairly low resolution: the spatial sampling is *sparse*. Despite these limitations, the amount of data acquired is always large.

In this dissertation, we will focus on LF content captured by Plenoptic Cameras. However, processing and analysis techniques for LF are affected by their high dimensionality, which raises the computational complexity and enforces more challenging restrictions on the design of algorithms. Despite this large advance in LF imaging technologies in the last 30 years, many techniques for LF processing have been proposed. For instance, Wu *et al.* [**34**] provide a timely and comprehensive overview of the field, including recent advances. The authors review and discuss key technologies, challenges, and future development trends, including the basic representation and theory of LF, acquisition, super-resolution, depth estimation, compression, editing, display, interaction and applications.

Currently available state-of-the-art techniques for LF processing are based on DL techniques, in particular, CNNs. Those methods have achieved better performance than the traditional approach, especially when taking advantage of the properties of LF structure, such as the

EPI structure [**35**]. Among several super-resolution methods, the learning-based approaches, e.g., Wu *et al.* [**34**], and Kalantari *et al.* [**36**] are the most relevant works in this field.

2.4. Final Remarks

As a summary of this chapter, we provided the theoretical background necessary for understanding this dissertation, which should be kept in mind throughout this dissertation. First, the basic definition and the different ways to capture LF images are presented to understand the concept of LF better. Then, we described the limitation of LF image processing as well. In the following chapter, the historical context around DL is briefly described for a better understanding of its development.

CHAPTER 3

Background on Deep Learning

This chapter presents the fundamental concepts of Deep Learning methods used in this dissertation. First, the definition and historical context of DL are presented briefly. After that, the following section focus on different ways to train a deep CNN. Finally, the last section mainly describes the frameworks used to train a deep CNN.

3.1. Introduction to Deep Learning

Deep Learning (also known as deep machine learning, hierarchical learning or deep structured learning) is a new subdivision of ML that has been developed with the objective of moving ML closer to one of its original goals: Artificial Intelligence (AI). Historically, DL can be drawn back to 1943, when Walter Pitts and Warren McCulloch modelled a simple neural network using electrical circuits to explain how the neurons might work in the brain. However, only recently, DL has attracted the attention of many researchers from both academia and industry.

DL is about learning multiple levels of representation and concept that help to create a sense of data, such as images. It is inspired by the deep structure of the brain, and researchers want to build a model to mimic how the brain works, as shown in Figure 3.1. Recently, DL has become a hot area of ML research, and it has shown significant improvements in many fields like computer vision [**37**], [**38**], [**39**], [**6**], natural language processing [**37**], [**38**], [**40**], speech recognition [**41**], [**42**], and machine translation [**43**], [**44**].

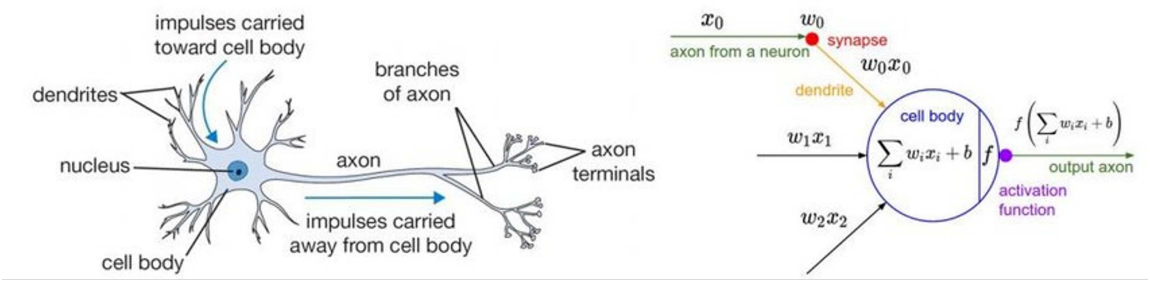
DL uses deep neural networks to solve ML tasks. Neural networks comprise of a collection of neurons and connections. A neuron accepts many inputs from predecessor neurons and produces one output. The output is a weighted sum of the inputs followed by the neuron's activation function, which is usually nonlinear. Neurons are organized into layers where neurons in the same layer are not connected. Neurons with no predecessor are called input neurons, and neurons with no successor are called output neurons. If the number of layers between the input neuron and the output neuron is large, then this neural network is called a deep neural network.

There is no strict definition of a "large number of layers", but generally, a neural network with more than eight layers, it is considered "deep" [**37**]. Modern deep neural networks can have more than hundreds layers [**6**]. Neurons are wired through connections and each connection transfers the output of a neuron i to the input of another neuron j. Each connection has a weight w_{ij} that will be multiplied with the activation, which will increase or decrease the signal. This weight will be adjusted during the learning process, and this process is called training.

Gradient Descent (GD) is the most used technique for training deep neural networks. It is a first-order optimization method for calculating the gradient of the loss function over the variable and affecting the variable in the negative direction of the gradient. The step size is proportional to the absolute value of the gradient. The ratio between the step size and the absolute value of the gradient is called the learning rate.

Calculating the gradient is the main step when performing GD, which is based on the back-propagation algorithm. To calculate the gradient with back-propagation, we need to first calculate each layer's activation by performing a feed-forward pass from the input neuron to the output neuron. This feed-forward pass is also called inference; the inference output can either be a value in regression problems or a discrete value in classification problems. The inference result can be correct or wrong, which is quantitatively measured by the loss function.

Next, we compute the gradient of the loss function, L, for each neuron and each weight. The gradients are calculated iteratively from the output layer to the input layer allowing the chain rule. Then, we update the weights with GD as $w_{ij}^{t+1} = w_{ij}^{t} - \alpha \frac{\partial L}{\partial w_{ij}}$ where α denotes the learning rate, a parameter that defines how much an updating step influences the current value of the weights, i.e., how much the model learns in each step. Such feed-forward, back-propagation, and weight update constitute a training iteration. It usually takes hundreds of thousands of iterations to train a deep neural network.



(a) (b)

Figure 3.1. A cartoon drawing of a biological neuron (a) and its mathematical model (b) [**5**].

3.2. Convolutional Neural Networks

The visual cortex, responsible for processing visual information in the brain, has a sequence of areas that process the information at a low to a high abstraction level. Visual cortex studies show that the existing neurons in it get activated by stimuli generated by localized fields. Linear filtering in image processing is executed through convolution in the spatial domain (or element-wise multiplication in the frequency domain). However, the idea behind CNN is to learn these filters in a data-driven manner. In ML, DL networks have multiple non-linear hidden layers and can represent the data hierarchically with lower to higher abstraction. CNN are variants of the Multilayer Perceptron (MLP), which are inspired by the visual cortex and have a DL architecture. CNNs were first introduced a long time ago by [**45**]. Fukushima designed neural networks

with multiple pooling and convolutional layers. In 1980, he developed an Artificial Neural Network (ANN) called Neocognitron, which uses a hierarchical, multilayered design. This design allows the computer to "learn" to recognize visual patterns. The networks resembled modern versions but were trained with a reinforcement strategy of recurring activation in multiple layers, which gained strength over time. Furthermore, Fukushima's design allowed important features to be adjusted manually by increasing the "weight" of certain connections. Not many successful attempts were reported before 2006, as the reported positive experimental results are typically based on two or three levels (i.e., one or two hidden layers).

Training deep networks consistently yielded poor results. However, there was an exclusion; LeNet-5, proposed by [**6**] was a particular type of DL model that has achieved an excellent generalization on visual inputs. This kind of model stacks many layers (eight layers) of trainable convolutional filters and pooling operations on top of each other to recognize the hand-written digits, (Figure 3.2), composed of multiple alternating layers of convolutions and subsampling followed by a few fully connected layers and finally output units for discriminating between 10 classes of handwritten digits which is very well known for its success in the digit recognition task.

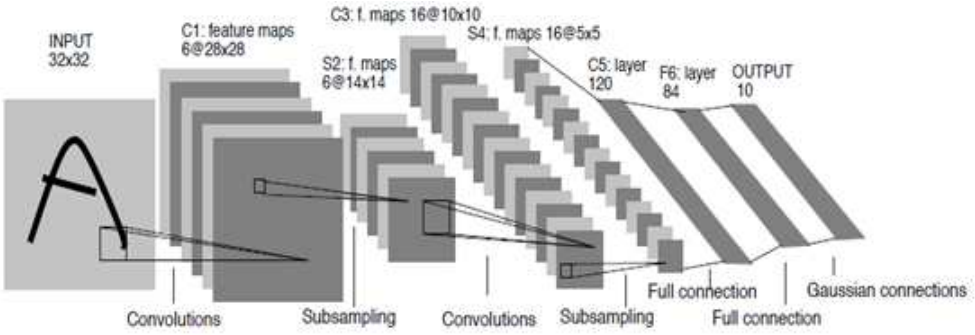


Figure 3.2. Convolutional network LeNet-5 [**6**].

The basic elements inside a CNN (e.g., LeNet-5) can be categorized into the following 5 classes:

(1) **Input layer:** It will hold the raw pixel values of the input image. In LeNet-5, the input is an image of 32 pixels width by 32 pixels height.

(2) **Convolution layer:** It will compute the output of nodes that are connected to local regions in the input, each computing a dot product between their weights and the region they are connected to in the previous layer. It is usually followed by a non-linear function to introduce a non-linearity. This may result in a next layer volume such as $28 \times 28 \times 6$ with 6 filters.

(3) **Normalization layer:** It will normalize the output of a hidden layer to a range of (0; 1) or (-1; 1) to prevent the large magnitude of data flow and to ensure convergence in the long run. This leaves the size of the volume ($28 \times 28 \times 6$) unchanged.

(4) **Pooling layer:** It will perform a downsampling operation along the spatial dimensions (width, height), resulting in a next layer, for example, $14 \times 14 \times 6$ with 2×2 sample window. There are multiple pooling methods for reducing the feature dimension. Max pooling is commonly used in CNNs. By eliminating non-maximal values, max pooling reduces computation for the following layers.

(5) **Fully-connected layer:** It will compute the class scores, resulting in a volume of size $1 \times 1 \times 1000$, where each of the 1000 numbers corresponds to a class score, such as the 1000 categories of ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC12). As the name implies, each node in this layer will be connected to all the nodes in the previous layer.

The most appealing success of deep CNNs came in 2012 when Krizhevsky proposed a deep CNN called AlexNet [**37**]. This model uses the same fundamental elements as LeNet-5 [**6**], i.e., convolution layer, pooling layer, normalization layer and fully-connected layer. However, there are some differences between AlexNet and LeNet-5, which lies in three folds:

(1) AlexNet uses the Rectified Linear Unit (ReLU) function to replace the Sigmoid Function (SF) in LeNet-5 to maintain the non-linearity, which makes the deep model faster to train.

(2) AlexNet adopts Dropout to further penalize the model and prevent overfitting. Dropout is a technique that reduces complex co-adaptations of nodes and forced to learn more robust features that are useful in conjunction with many different random subsets of the other nodes. Meanwhile, Dropout roughly doubles the number of iterations required to converge.

(3) AlexNet uses a Graphics Processing Unit (GPU) based implementation with two GPUs, while LeNet-5 is implemented in one CPU, which improves the efficiency of the training process. Probably the most important piece was the use of GPUs to train the model. GPUs are essentially parallel floating-point calculators with 100s-1000s of cores. The speedup offered by GPUs meant they can train larger models, which leads to lower error rates.

The success of AlexNet showed the power of deep models dealing with images when they can be trained efficiently using GPUs and large datasets. AlexNet went on to be known as the baseline architecture of modern CNNs. Thus, since 2012, with the success of AlexNet, recent research has focused primarily on improving the accuracy, and many other deep models have been proposed, VGGNet [**46**], GoogleNet [**47**], InceptionV4 [**48**], ResNet [**6**], DenseNet [**49**], and so on. The performance of deep CNNs is improving rapidly: the winner of ImageNet challenge has increased the classification accuracy from 84.7% in 2012 (AlexNet, [**37**]) to 96.5% in 2016 (ResNet-152, [**6**]).

Deep CNNs have a wide range of applications, and the performance of computer vision tasks definitively benefited from DL, especially CNNs by swapping hand-crafted features with features automatically extracted from deep neural networks, [**37**]. With recent techniques such as batch normalization [**39**], that consists in making training of deep CNN faster and more stable and residual blocks [**6**], that solves the neural network degradation problem. We can train even deeper neural networks and the image classification accuracy can surpass human beings, [**6**], [**50**]. These advancements have spawned many vision-related applications, such as self-driving cars [**51**], medical diagnostics [**52**], and video surveillance [**53**]. DL techniques have made significant advancements in generative models, which have improved the efficiency and quality of compressed sensing [**54**], super-resolution [**55**]. Generative models have originated many new applications such as image style transfer [**56**], visual manipulation [**57**], and image synthesis [**57**].

3.3. Methods for Training Deep Neural Networks

Training a deep network from scratch requires a considerable amount of data as well as a lot of computational power. For many applications, few labelled data are available; therefore, training a new network is a challenging task. Training a network from scratch (with random initialization of the filter weights) is useful when the dataset is large enough to make a network converge and has several advantages, such as: (i) extractors tuned specifically for the dataset, which tend to generate more accurate features, and (ii) full control of the network. However, fully training a network from scratch requires a lot of computational and data resources [**58**], since the convergence of the network is pruned to overfitting. Also, the convergence of a CNN can not be totally confirmed (given its highly non-convex property), making tuning not so trivial.

In practice, few people train an entire CNN from scratch (with random initialization) because it's relatively rare to have a dataset of sufficient size. Instead, it is common to use the transfer learning method, which is a strategy that aims to improve the performance of DL tasks by leveraging knowledge obtained from a different task. The most popular application of transfer learning is image recognition using CNNs. A bunch of high-performing, state-of-the-art CNN-based image recognition models, trained on ImageNet data (1.2 million images with 100 categories), are available publically. Examples of such models include AlexNet [**37**], VGGNet [**46**], InceptionV4 [**48**], GoogleNet [**47**] and ResNet [**6**]. These deep CNNs often contain a large number of parameters. For instance, AlexNet [**37**] has 60 million parameters. In this case, the weights are learned from a dataset containing 1.2 million high-resolution images for training. However, directly learning so many parameters from only a few thousand examples is problematic. Transfer learning allows us to deal with these situations by leveraging the already existing labelled data of some related task or domain. CNNs can act as a generic extractor of mid-level representations, which can be pre-trained in a source task and then re-used in other target tasks. This dissertation uses the transfer learning method instead of training from scratch.

3.4. Deep Learning Frameworks

Directly programming a multi-core CPU or a GPU can be difficult, but luckily the neural network computation can be abstracted into a few basic operations such as convolution, matrix multiplication, and so on. Using these operations that were already highly optimized for high-performance hardware, users only need to focus on high-level neural network architectures rather than low-level implementations. DL frameworks are essential today as they provide a quick and elegant way of representing and training a neural network. A DL framework gives a representation of the skeleton of the application. The developer needs to define his/her model by ordering a set of pre-implemented DL operations (Convolution, pooling, linear function, activation functions), which give a hand to manipulate and modify the detailed parameters of the model. Frameworks also help with the training process by optimizing the gradient calculations. Many popular ML frameworks and libraries already offer the possibility to use GPU accelerators to speed up the learning process. Some of them also allow the use of optimised libraries such as CUDA Deep Neural Network (cuDNN) library[1], and OpenCL to improve the performance even further. Many frameworks are available such as TensorFlow [**59**], PyTorch [**60**], Caffe [**61**], Theano [**62**], and so on. Although DL frameworks have a lot of differences between them, they share the same structure and components, such as Tensor Object, Computation Graph and Optimizations, Auto-Differentiation tools, and so on.

In this dissertation, we used three different frameworks. The first framework used is Caffe2 [**63**], which has been deployed at Facebook to help developers and researchers train large ML models and deliver AI-powered experiences in our mobile apps. Now, developers will have access to many of the same tools, allowing them to run large-scale distributed training scenarios and build ML applications for mobile. The second framework we used is PyTorch [**60**]. As the most popular dynamic framework, PyTorch embeds primitives for constructing dynamic dataflow graphs in Python, where the control flow is executed in the Python interpreter. The third framework we used is Tensorflow [**59**], which is an open-source framework developed by Google Brain Team. It is currently released under the Apache 2.0 open-source license. TensorFlow is designed for large-scale distributed training and inference. Its computation graph went from static in the pre-2.0 versions to dynamic with a static option that can be enabled at anytime. Nodes in the graph represent mathematical operations, while the graph edges represent tensors communicated between them. These operations involve two hundred normal operations, including C++ mathematical, array manipulation, managing flow, and state management.

3.5. Final Remarks

In the first part of this chapter, we briefly introduced the basic concepts of DL and presented the essential historical context of DL methods. After that, we introduced the most commonly used DL method, the CNN. Also, we presented the different methods for training a deep CNN. The most widely used DL frameworks are briefly presented in the last part of this chapter. In summary, DL has seen tremendous growth in its popularity and usefulness, primarily due to

[1] https://developer.nvidia.com/gpu-accelerated-libraries

more powerful computers, larger datasets, and techniques to train deeper networks. The years ahead are full of challenges and opportunities to improve DL further and bring it to new frontiers. The following chapter will present the literature review on SOD.

CHAPTER 4

Literature Review on Salient Object Detection

In this chapter, we first introduce the concept of SOD. After that, the following sections review the traditional and DL methods for 2D and 3D SOD. Finally, in the last section, we review the 4D LF SOD using traditional and DL methods.

4.1. Introduction to Salient Object Detection

Salient object detection is a difficult open research problem aimed at retrieving the most conspicuous visually distinct foreground information from an image in a manner reminiscent of the human visual system. It is challenging because human vision is difficult to mimic by automated systems. SOD methods attempt to extract points and regions of a visual scene that are more significant to human visual attention by forming a map that defines how a region stands out from its background and by analysing image surroundings. Saliency detection has recently attracted much attention as a crucial pre-processing technique in computer vision to reduce computational complexity.

SOD has boosted a wide variety of applications in computer vision tasks, such as image recognition [**64**], video SOD [**65**], semantic segmentation [**66**], visual tracking [**67**] and image retrieval [**68**]. There has been an increasing interest in SOD in recent years. Countless applications of SOD have led to the proposal of numerous methods for saliency computation. Most of the conventional SOD methods follow the flowchart as shown in Figure 4.1.

The input to a SOD system can be a simple 2D image (or set of 2D images) or more complex visual information, such as an LF. The SOD flowchart is followed by three major steps. Firstly, several pre-processing techniques are used, such as resize, crop, rotation, flipping, and so on, in order to have the image ready to be used. Secondly, necessary features are extracted from the image by using several methods, such as contrast saliency, colour contrast saliency, Region-based Depth Feature Descriptor (RDFD) and more. Afterwards, the classifier step is trained with those extracted features by using a binary mask. Finally, the last step is the saliency map that corresponds to the output of the SOD system.

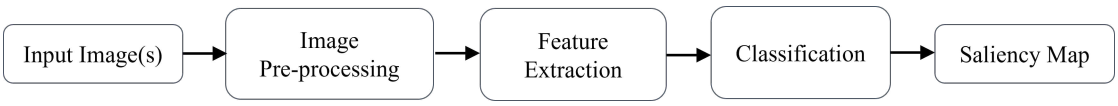


Figure 4.1. Typical SOD flowchart.

Existing models for SOD can be generally classified into two groups: (i) traditional methods that use handcrafted features, and (ii) DL methods that use automatic feature extraction,

as illustrated in Figure 4.2. The traditional SOD methods considered in this section belong to the various advancements made in SOD before the resurgence of DL techniques. These traditional methods mainly exploit low-level visual features such as intensity, color, and orientation to design handcrafted features especially useful for SOD. In recent years DL has been the most used approach for SOD. Those DL models can be categorised based on the level of supervision during the model training. The arrangement of the most recent developments in categories of supervised, weakly-supervised, and unsupervised learning is useful in understanding the key design issues of SOD models.

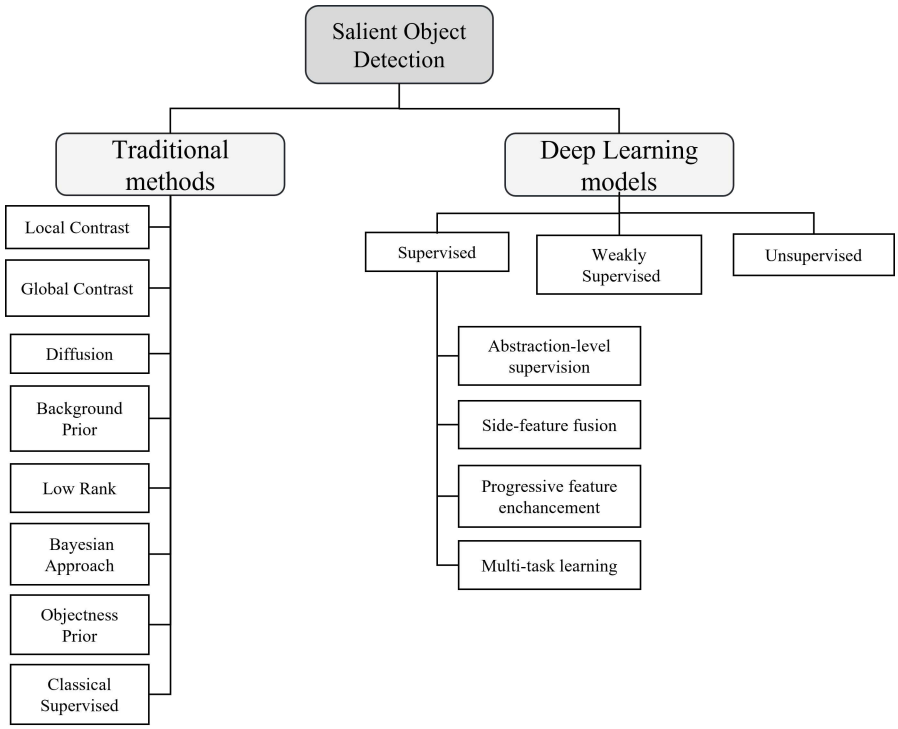


Figure 4.2. Classification of SOD methods.

Existing works for SOD can also be characterized based on the type of input image data, e.g., 2D, 3D and 4D LF. In the following section, more information is given about those three categories.

4.2. 2D and 3D Salient Object Detection

This section reviews the most prominent 2D and 3D SOD methods. 2D methods [**69**], [**70**] have achieved appealing performance and have long been dominant in saliency detection. However, 2D methods may produce unsatisfactory results in complex scenes, e.g., the foreground objects share a similar appearance, or the background is cluttered. 3D saliency detection methods [**71**], [**72**], in which the depth map is introduced with rich 3D geometric information, have also received a lot of interest. Such methods have significantly improved detection accuracy, especially in some complex scenarios. However, 3D saliency detection methods rely on the quality of the depth map, which is usually limited by technical conditions. And when the quality of the depth map is poor, the performance of 3D SOD methods may decrease correspondingly.

4.2.1. Traditional 2D and 3D SOD Methods

As the saliency of a region can be defined in various ways, different works made varying assumptions for SOD model formulation. Priors such as backgrounds, objectness, focuses, and spatial distribution allow the use of more sophisticated frameworks to attack visual saliency. However, early 2D and 3D SOD methods [**69**], [**70**] focus on exploiting low-level hand-crafted features, such as colour, region contrast, and so on. Therefore, different from these low-level feature-based fast approaches for SOD have been proposed, such as [**73**], [**74**], [**75**]. However, these models also depend on the manually designed regional saliency descriptors and therefore differ from DL-based SOD in the feature extraction process.

For complex scenes, some methods may produce unsatisfactory results. Therefore, the depth map is introduced with rich 3D geometric information. Additionally, over the past several years, many traditional 3D models based on handcrafted features have been developed [**76**], [**77**], [**78**], [**79**], [**80**], [**81**], [**82**]. For example, the early work [**83**] focused on modelling the interaction between layout and shape features generated from the 2D image and depth map. Besides, the representative work [**71**] developed a novel multi-stage 3D model, and constructed the first large-scale 3D benchmark dataset, termed National Laboratory of Pattern Recognition (NLPR).

4.2.2. Deep Learning 2D and 3D SOD Methods

Recently, benefiting from the powerful ability of CNNs in extracting hierarchical feature representations, most of the traditional saliency methods based on hand-crafted features and prior knowledge [**84**] have been gradually surpassed by DL methods. Therefore, several methods for SOD based on DL have been proposed in recent years. For instance, Li and Yu [**85**] adopt a CNN to extract multi-scale features to predict saliency for each super-pixel which carry more information than pixels and align better with image edges than rectangular image patches.

Wang *et al.* [**86**] propose two CNNs to integrate local super-pixel estimation and the global search for accurate saliency prediction. Zhao *et al.* [**70**] extract both global and local contexts by using two independent CNNs. Lee *et al.* [**87**] propose to combine the low-level distant maps with high-level semantic features of deep CNNs for SOD. Although, these methods have achieved a very good performance, they might suffer from time-consuming computation and limited spatial information of the input images due to the image patches as inputs and fully-connected layers.

To remedy those problems, Liu and Han [**88**] develop a deep hierarchical saliency network to learn a coarse global estimation and then hierarchically and progressively refine saliency maps to recover image details by integrating local context information. Hou *et al.* [**89**] design an intensely supervised saliency network with dense short connections, which focus on skip-layer structure with deep supervision and can extract abundant multi-scale features for SOD. Zhang *et al.* [**90**] propose an aggregating saliency network, combining multi-level features with different resolutions to recursively predict saliency maps.

More recently, Luo *et al.* [**91**] proposed a simplified CNN to combine both local and global information and design a loss to penalise boundary errors. Deng et al. [**92**] present a refinement network to learn the residual between ground-truth and the saliency map using the residual refinement block for SOD. Liu *et al.* [**93**] introduce an attention mechanism to guide feature integration. Zhang *et al.* [**94**] propose an attention-guided network that selectively progressively integrates multi-level contextual information. Wu *et al.* [**95**] discard the low-level features and design a novel cascaded partial decoder to obtain the salient objects effectively and accurately.

Zhao *et al.* [**96**] propose an edge-guided FCN which embeds the prior edge knowledge into multi-level features and utilises them to preserve good boundaries of salient objects. In [**72**], two-stream models are used to process the 2D image and depth map separately, and cross-modal features are combined to make predictions jointly. Cong *et al.* [**97**] propose a novel depth-guided transformation model going from 2D saliency to 3D saliency. Zhu *et al.* [**98**] present a robust prior-model guided depth-enhanced saliency detection network based on CNNs. Chen and Li [**14**] design a progressive fusion network to fuse cross-modal multi-level features to predict saliency maps. Chen an Li [**72**] propose a three-stream network to extract 3D features and use an attention mechanism to select complements adaptively. Zhao *et al.* [**96**] enhance the depth information as an attention map and utilize it with a novel fluid pyramid structure to obtain more accurate salient objects. These methods demonstrate that depth cues contribute to detecting saliency objects.

However, due to the effort of many researchers to address the problem of SOD, it is still a challenging task due to the appearance of objects varying significantly, e.g., object shape and size, aggravating the difficulties of accurate SOD. To address some of those challenges of SOD, Sun *et al.* [**7**] propose a new model Context-aware Cross-level Fusion Network (C2FNet), as shown in Figure 4.3. Specifically, the model is divided into two major modules: (i) the Attention-induced Cross-level Fusion Module (ACFM) module aims to extract the cross-level features from the backbone to integrate the features with the guidance from a Multi-Scale Channel Attention (MSCA) element, computes attention coefficients from multi-level features, uses the obtained attention coefficients to refine the features, and then integrates the obtained features into fused features; and (ii) the Dual-branch Global Context Module (DGCM), which is used to manipulate the rich context information from the fused features, converts the input feature into multi-scaled ones with two comparable branches, employs the MSCA elements to compute the attention coefficients, and integrates the features by considering the attention coefficients. The ACFM and the DGCM are cascaded into two stages, from high-level to low-level. The final DGCM predicts an image's saliency map of the salient objects.

In this work, we will use the C2FNet model as benchmark for the existing 4D LF models, since it has reported good results for 2D SOD in the literature. This comparison aims to understand the advantages of 4D LF SOD approach over the 2D approach. As far as we know, the C2FNet model represents the state-of-the-art on 2D models for SOD.

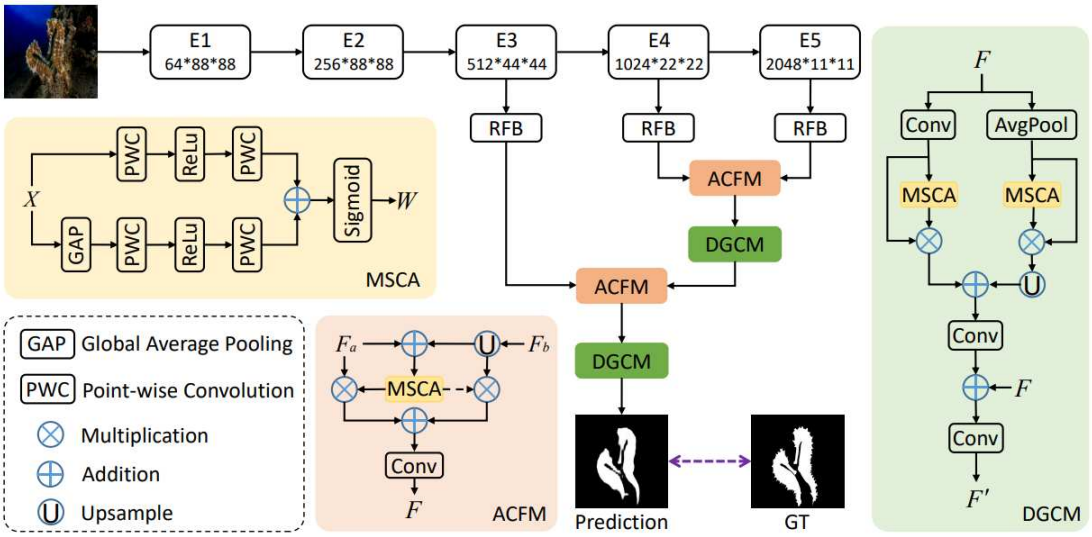


Figure 4.3. The whole pipeline of the C2FNet model [**7**].

4.3. 4D Light Field Salient Object Detection

4D LF images are more likely to match the physiological sensing capabilities of the human visual system compared to 2D and 3D image [**99**]. Generally, people can distinguish regions by adjusting the focus of the eyes across different depth planes, called refocusing ability. In other words, when human eyes focus on specific objects, the other regions are automatically blurred [**99**]. The LF data show a high resemblance to the human visual perception mechanism, in which a set of focal slices focusing at different depth layers are synthesized through LF rendering and refocusing techniques [**33**]. In the following subsection, we review and discuss existing models proposed for 4D LF SOD, including traditional and DL models.

4.3.1. Traditional 4D Light Field SOD Methods

Traditional LF SOD models often extend various hand-crafted features which are widely adopted in conventional saliency detection [**100**], such as global or local colour contrast, background priors, and object location cues, to the case of LF data. Besides, these traditional models tend to employ some post-refinement steps, e.g., an optimization framework [**100**], [**11**], to achieve saliency maps with better spatial consistency and more accurate object boundaries. Regarding the data format leveraged, almost all traditional models work with focal stacks, while depth is incorporated by some of them. Only two traditional models consider using the multiview [**90**] and microlens data [**101**]. Further, due to early dataset scarcity, almost all traditional models are evaluated only on the small LF dataset constructed by [**99**]. Despite early progress made by these traditional models, due to general limitations of hand-crafted features, they can hardly generalize well to challenging and complex scenarios compared to modern DL models.

There have been few traditional 4D LF SOD solutions proposed in the literature compared to 2D and 3D saliency detection. For instance, Li *et al.* [**99**] built the first LF saliency dataset and proved that LF data with valuable focuses, depths and objectness cues could better locate

and detect salient objects. Then, they also present a weighted sparse coding structure to handle different types of LF data for SOD [**102**]. Wang *et al.* [**103**] design a two-stage Bayesian integration framework to fuse useful features extracted from LF data to detect saliency regions. Zhang *et al.* [**104**] integrate multiple visual cues from LF images to detect salient regions. These methods consistently demonstrate the effectiveness of LF data in saliency detection. However, these LF saliency methods only rely on hand-crafted features to generate saliency maps.

4.3.2. Deep Learning 4D Light Field SOD Methods

Recently, a few methods for 4D LF SOD using DL have been presented in the literature. Among the different data formats adopted for organizing the LF content, the focal stack has attracted the attention of various researchers due to its multi-variable property [**105**]. Thanks to the effort of many researchers, several methods for constructing focal stacks have been proposed, such as [**11**], [**13**], [**21**], [**106**]. However, other LF formats, such as multiview, all-in-Focus, lenslet and center-view only, have also been considered in DL-based SOD methods proposed in the literature [**21**], [**105**], [**11**], [**106**] [**13**], [**107**], [**12**].

It is worth mentioning that DL methods are currently state-of-art in the field of LF SOD. In the next chapter, we will present a complete review of 4D LF SOD based on DL solutions and classify them according to the used architectures. It is also essential to state that the main idea behind this dissertation is to evaluate the effectiveness of different architectures of existing deep 4D LF SOD models and propose a new model for 4D LF SOD.

4.4. Final Remarks

As a summary of this chapter, we covered the most relevant work in the literature about 2D, 3D and 4D LF SOD, including traditional and DL methods. We presented the definition and the different ways to capture LF. The following chapter will show the implementation and configuration of the DL architectures for 4D LF SOD used for this dissertation.

CHAPTER 5

4D Light Field SOD using Deep Learning: Classifying and Reviewing

This chapter presents a meaningful classification taxonomy for deep neural networks adopted for LF SOD. Guided by this classification, some of the most relevant 4D LF SOD solutions available in the literature are reviewed and compared in terms of their differences, commonalities and relationships. In addition, an alternative and novel LF SOD architecture is proposed in this chapter to further study the influence of different parameter configurations. In the next chapter, we will introduce the performance evaluation of 4D LF SOD using DL models.

5.1. Classification Taxonomy

As mentioned above, LF imaging has been getting attention from many researchers in recent years. Therefore, a few deep neural network architectures have been proposed in the literature for LF SOD. In this context, we can classify existing deep neural networks for SOD into five categories according to their architectures: (i) the late-fusion scheme, (ii) middle-fusion scheme, (iii) knowledge distillation-based scheme, (iv) single-stream scheme, (v) reconstruction-based scheme and (vi) the proposed architectures for 4D LF SOD, as shown in Figure 5.1.

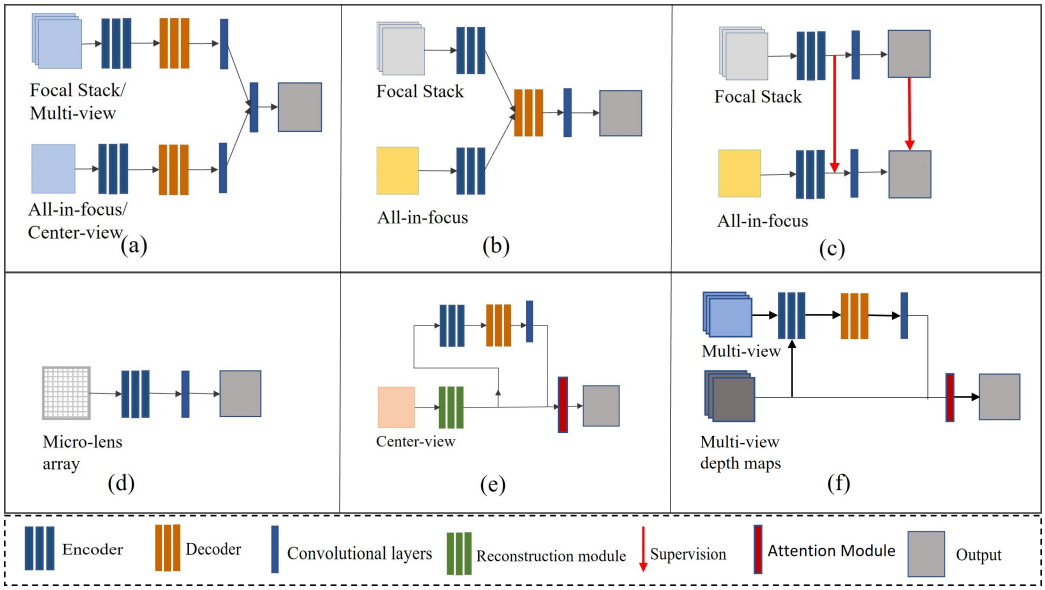


Figure 5.1. Architectures of existing deep 4D LF SOD models: (a) Late-fusion scheme, (b) Middle-fusion scheme, (c) Knowledge distillation-based scheme, (d) Single-stream scheme, (e) Reconstruction-based scheme, and (f) proposed architecture for 4D LF SOD (adapted from [**8**]).

The late-fusion scheme (Figure 5.1a) aim to obtain individual predictions from the input focal stack/multiview images and all-in-focus/center-view image and then simply fuse the results. The fusion process is restrained to the last step with relatively simple integrative computation.

The middle-fusion strategy (Figure 5.1b) extracts features from the focal stack and all-in-focus image in a two-stream manner. An elaborated and complex decoder then does fusion across intermediate features. Compared to the late-fusion strategy, the main differences are that the features fused are usually hierarchical and intermediate.

In contrast to the late-fusion and middle-fusion schemes that investigate how to fuse focal stack and all-in-focus features, the knowledge distillation-based scheme (Figure 5.1c), attempts to transfer focusness knowledge of a teacher network that handles focal stacks, to a student network that processes all-in-focus images. It uses both the features and prediction from the focal stack stream to supervise those features and prediction obtained from the all-in-focus stream, effectively boosting the performance of the latter.

The single-stream scheme (Figure 5.1d) is inspired by the fact that LF can be formulated in a single image representation, namely the microlens image array. This scheme processes the microlens image array directly using a single bottom-up stream without explicit feature fusion.

In comparison with the categories above, the reconstruction-based scheme (Figure 5.1e) focuses on a different aspect, namely reconstructing LF data from a single input image. With the assistance of the reconstructed LF, an encoder-decoder architecture with a middle-/late-fusion strategy can then be employed to complete LF SOD.

To show the advantage of using LF data as input, we propose an new architecture (Figure 5.1f), which is inspired by the reconstruction-based architecture. However, different the reconstruction-based architecture, our proposed architecture uses multiview images and the depth maps as input. The main advantage of this proposed architecture is that it can benefit from the growth of LF content in recent years by using LF images as input which contain more information than 2D images.

In the following sections, we will describe all the deep models, including our proposed model, used for all the experiments, using the above architecture.

5.2. Synergistic Attention Network (SA-Net)

The Synergistic Attention Network (SA-Net), proposed by [**9**] uses the late-fusion scheme (Figure 5.1a). This model was proposed to address the LF SOD by establishing a synergistic effect between multi-modal features with advanced attention mechanisms. This model exploits the rich information of focal stacks via 3D CNN, decodes the high-level features of multi-modal LF data with two cascaded synergistic attention modules, and predicts the saliency map using an effective feature fusion module in a progressive manner.

Figure 5.2 shows an overview of the SA-Net model, which consists of three major components, including: (i) a multi-modal encoder consisting of 2D and 3D CNNs, (ii) two cascaded Synergistic Attention (SA) modules, and (iii) a Progressive Fusion (PF) module. The first component, employs 3D CNNs to progressively extract the sequential features from FSs. Specifically, the encoder is a dual-branch architecture for synchronous feeding of the two modalities, i.e., AiF images and FSs. The second component is the decoding stage, with a SA module, where the features from AiF images and FSs are selectively fused and optimized to achieve a synergistic effect for SOD. Finally, the last component is the PF module to obtain the final prediction, which fuses multi-modal features and predicts the saliency map progressively.

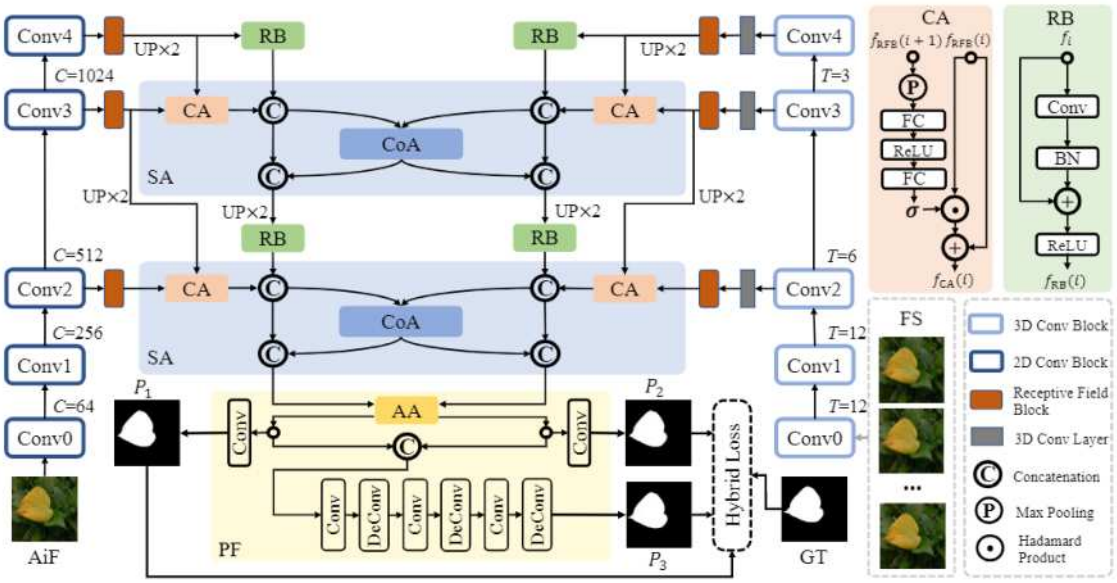


Figure 5.2. The whole pipeline of the SA-Net model [**9**].

5.3. Patch-Aware Network (PANet)

Patch-Aware Network (PANet) [**10**] is based on a middle-fusion architecture, (Figure5.1b), which was proposed to explore LF data in a regionwise way. The entire architecture of the network is illustrated in Figure 5.3. This model was adopted from the widely used VGG16 model, where the authors used the VGG16 model as the encoder and dropped the last pooling and fully connected layers for adapting their task. More specifically, the model is divided in two main modules.

The first module is the Multi-source Learning Module (MSLM), which is designed to locate the focused salient regions to generate a filtering strategy for integration followed by three guidances based on saliency, boundary, and position. Additionally, they design a classifier to generate guidance on the filtering strategy for selecting focused salient regions in focal slices. It leverages three guidances produced from the comparisons between predictions of the 2D image and focal slices to generate pseudo labels for supervision.

The second module is the Sharpness Recognition Module (SRM), which aims to integrate features of focal slices for highlighting salient regions and inhibiting non-salient regions. Their

approach is to leverage the spatial information of the LF and integrate focused salient regions by recognizing the sharpness of each region. However, their approach is unable to recognize the sharpness of textureless regions. Therefore, in order to recognize sharpness accurately, they adopt another classifier to distinguish each feature block into two categories according to the sharpness.

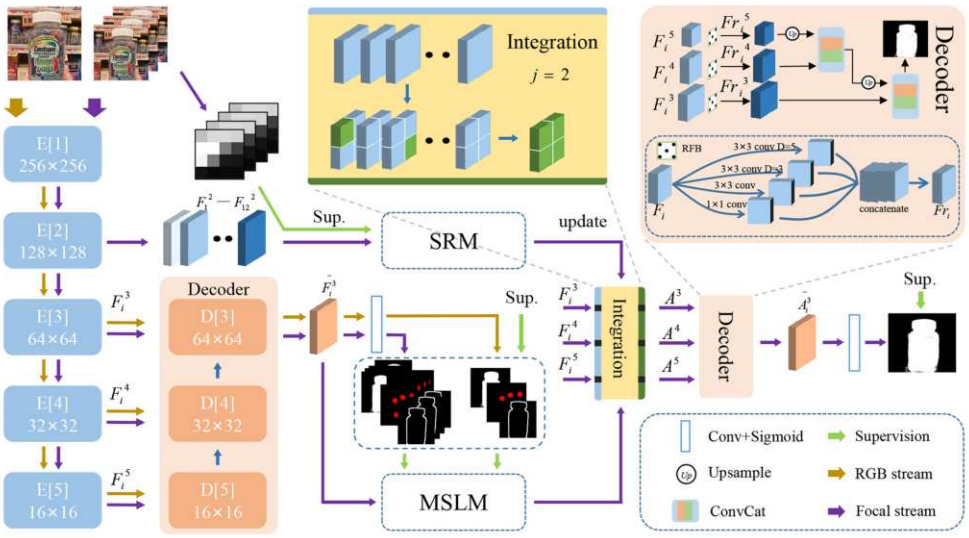


Figure 5.3. The whole pipeline of the PANet model [**10**].

5.4. Exploit and Replace Network (ERNet)

The Exploit and Replace Network (ERNet), proposed by [**11**], presents an asymmetrical two-stream architecture inspired by knowledge distillation (Figure 5.1 c), in which the main idea is to improve saliency detection in 2D images using detection learning in LF images. Therefore, as shown in Figure 5.4, the entire ERNet network structure consists of two main parts: (i) the focal stream, and (ii) the single image stream. The focal stream serves as the teacher network that aims to learn to exploit focal slices for higher requirements on desktop computers. On the other hand, the single image stream correspond to the student network which takes a single 2D image as input and aims to learn to replace focal slices for computational efficiency on mobile devices.

More precisely, a teacher network learns to exploit LF data, which are Multi-Focusness Recruiting Modules (MFRM) and Multi-Focusness Screening Modules (MFSM). The MFRM and MFSM are used to enable our teacher network with a more accurate prediction for higher requirements on desktop computers and duo transfer of rich focusness knowledge to the student network. The MFRM focuses on explicitly recruiting saliency information from each focal slice, and the MFSM aims to select useful features and suppress unnecessary ones.

The student network, which highlights the challenges of the teacher network, namely MFRM and MFSM, that directly transferring the output of the teacher to the student overlooks the inherent differences between an 2D image and focal slices. They propose multifocusness distillation by encouraging multi-focusness consistencies between the Focal stream and 2D stream

and screened focusness distillation schemes by learning complementarity between the screened focusness knowledge from the Focal stream and appearance information from the 2D stream. The distillation schemes ensure better absorption of focusness knowledge and enable the student to replace the focal slices with a single 2D image.

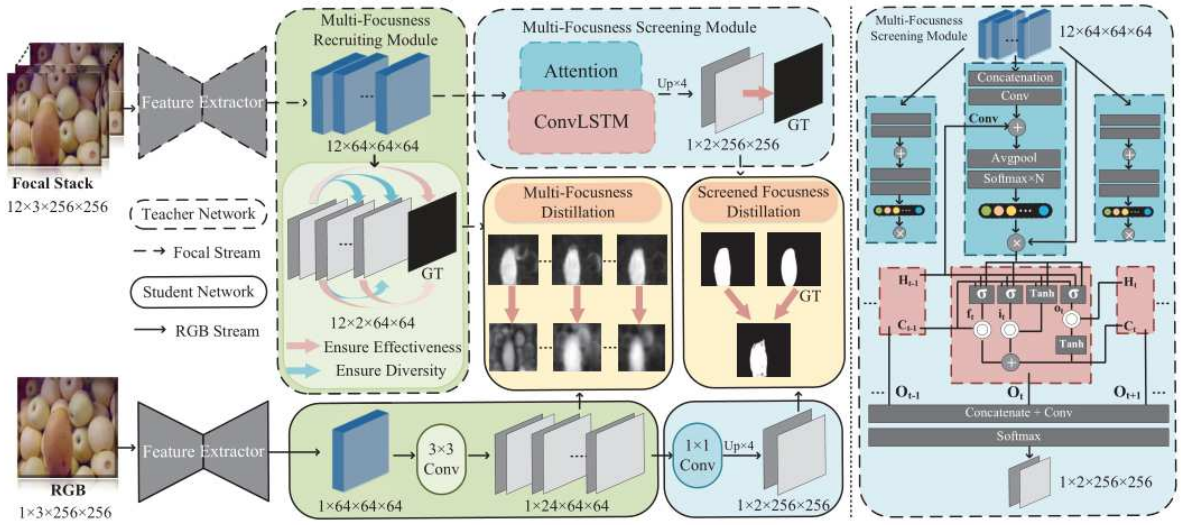


Figure 5.4. The whole pipeline of the ERNet model [**11**].

5.5. Model Angular Changes (MAC)

The Model Angular Changes (MAC) [**12**], which is an end-to-end deep convolutional network framework originally proposed for LF saliency detection based on the MLA LF format. This model is based on Single-stream scheme (Figure 5.1d). The goal is to extract the salient area by directly exploiting the angular information of the LF content. Features to represent the angular information are extracted in the MAC block and used later by DeepLab-v2 model [**108**] as shown in Figure 5.5.

More specifically, the MAC blocks are designed to extract and convert the LF MLA image in feature maps, which are then fed into the 2D to be transferred in different ways to be processed by a modified DeepLab-v2 backbone model to predict saliency maps. The goal of the MAC block is to model angular changes the one-pixel location in an explicit manner. An essential part of learning angular features is the design of convolutional kernels applied to the microlens images. To do that, first, the spatial dimension of the MAC block output is ensured to be the same as that of the 2D SAI. Second, the number of convolutional kernels in the MAC block is the same as that of the convolution kernels in the first block of DeepLab-v2.

Additionally, the kernel size shares the same angular resolution of one microlens image, and the number of kernels and the stride size are set to extract angular features for each microlens image. Specifically, angular kernels are 64, each of which is a 9×9 filter. The stride of convolution operations is 9, which leads to $540 \times 375 \times 64$ feature maps. Then, the feature maps with size $540 \times 375 \times 64$ are fed to a modified DeepLab-v2 to predict saliency maps. It is worth mentioning that there are several variants of DeepLab. However, DeepLab-v2 [**108**],

which introduces Atrous Spatial Pyramid Pooling (ASPP) to capture multi-scale information and long-range spatial dependencies among image units was selected.

The DeepLab-v2 architecture is composed of five convolutional (Conv) blocks, each of which is divided into convolutions followed by a ReLU. Each Conv block is followed by a max pooling layer, which was created to reduce the variance of features by computing some operation of a particular feature over a region of the image. The ASPP is applied on top of Conv block5, which consists of four branches with atrous rates ($r = \{6, 12, 18, 24\}$). Each branch contains one 3×3 convolution and one 1×1 convolution. The resulting features from all branches are then passed through another 1×1 convolution and summed to generate the final score. After the ASPP, the network further employs bi-linear interpolation to up-sample the fused score map to the original resolution of the central viewing image, which produces the saliency prediction at the pixel level.

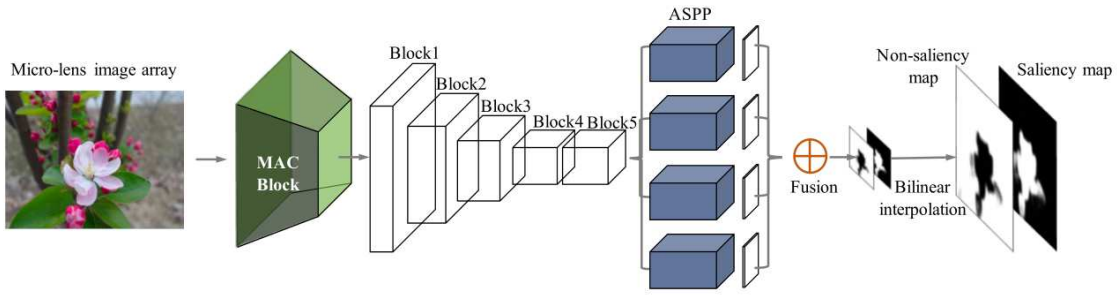


Figure 5.5. The whole pipeline of the MAC model [**12**].

5.6. Deep LF driven Saliency Detection (DLSD)

The Deep LF driven Saliency Detection (DLSD) model proposed by [**13**] is based on the reconstruction based architecture (Figure 5.1e). The idea of this model is to enrich the saliency detection in 2D images using LF rendering and an LF-driven SOD architecture. Therefore, the system learns and emulates the spatial and angular features of an LF image through a network specialized in rendering LF images from a 2D image.

As shown in Figure 5.6, they first expand a single view to an array of views. However, the huge amount of high-dimensional data makes existing LF processing ineffective and inefficient. Considering the redundancy of LF information, they only synthesize the views along the horizontal and vertical directions through the central view, which are good enough to represent the original LF. Therefore, after generating the LF images, they design an LF-driven saliency detection network to estimate saliency maps. The LF-driven saliency detection network consists of two parts: (i) the MultiView Saliency Detection Sub-network (MVSD-SNet), and (ii) the Multiview Attention Module (MVAM). The MVSD-SNet is based on the VGG19 model, where the inputs are the synthesized LF images with a size of $14 \times 256 \times 256 \times 3$, and the network outputs multi-level saliency predictions, which have 1, 1/2, 1/4, 1/8, 1/16 spatial resolution of the input images, respectively. The MVAM aims to integrate MV saliency maps in a view-wise

attention fashion, where the attention mechanism can learn the importance of saliency maps from different views by considering the geometric information (MV depth maps).

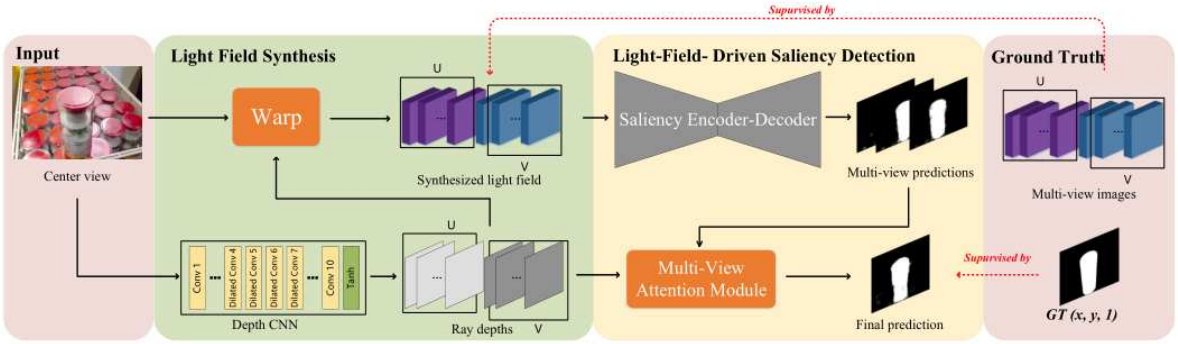


Figure 5.6. The whole pipeline of the DLSD model [**13**].

5.7. Deep LF driven Saliency Detection with Multiview (DLSD-MV): A Proposal

Due to technological innovation and new instrument development over the past decade, in particular, the use of camera arrays, LF images have gained popularity and are readily available and inexpensive. Therefore, to further analyze the advantages of using a neural network with LF images as inputs for SOD instead of 2D images, we propose an alternative LF SOD solution based on the DLSD model in [**13**]. However, the proposed model does not use Reconstruction-based architecture. Instead, we propose a new architecture as shown in (Figure 5.1f), which uses LF images as input.

Additionally, in the proposed model, we removed the LF synthesis module from the DLSD model, which estimated depth maps along the horizontal and vertical directions with two independent convolutional networks. According to the depth maps, the single view image was warped into horizontal and vertical viewpoints of the LF. Thus, our proposed model directly takes the original LF images as input. Therefore, the modified network, named the Deep LF driven Saliency Detection with Multiview (DLSD-MV) model receives as input the MV images and the depth information. The DLSD-MV model follows the patterns as depicted in Figure 5.7. More concretely, the DLSD-MV model consists of a MVSD-SNet and a MVAM designed for saliency prediction. However, it is important to mention that the MVSD-SNet and the MVAM were unchanged.

5.8. Final Remarks

In this chapter we have first presented the classification taxonomy for deep neural networks adopted for 4D LF SOD. After that, we introduced the five models we chose to use for all the experiments in this dissertation. In addition, a proposed novel LF SOD architecture are presented as an alternative to CNN models that use 2D images as input to reconstruct LF images for SOD. In the last part of this chapter, we presented the datasets used for training, testing and evaluation, the training configurations, and the evaluation metrics for SOD. The next chapter will be devoted to a detailed analysis of the results achieved.

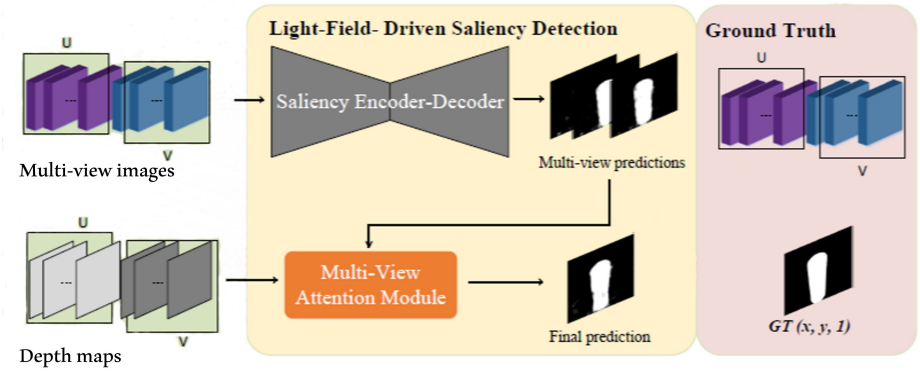


Figure 5.7. The whole pipeline of the proposed DLSD-MV model.

CHAPTER 6

Performance Evaluation of 4D LF SOD Deep Learning Models

This chapter presents first we present the experimental setup, including the training configurations and the datasets used for all experiments in this dissertation. After that, the performance evaluation of the various deep 4D LF SOD models reviewed in the previous chapter is presented. We will first present the overall quantitative evaluation of the 4D LF SOD solutions and compare them with a 2D SOD solution. Next, we will evaluate the cross-dataset to see how the models generalize to different datasets. Then, we present a visual comparison of the results from each model. Finally, we present the quantitative and qualitative results of the model proposed in this dissertation.

6.1. Experimental Setup

To evaluate the efficiency of each DL-based LF SOD model, which requires good computational power for both training and testing of the model, we preserved the parameters of the original models and we have only adjusted the parameters listed in Table 6.1 aiming to train the models efficiently without overfitting. All our experiment sessions were carried out using all five pre-trained models. We ran experiments on the three most popular frameworks, Tensorflow, PyTorch and Caffe2, and using an NVIDIA Tesla P100 GPU[1]. Specifications of the machine used in training and testing are shown in Table 6.2.

Table 6.1. Hyper-parameters optimization.

Model	**Hyper-parameters**				**Training Time [Hours]**
	Learning Rate	Momentum/ Weight Decay	Batch Size	Iterations	
SA-Net [**9**]	0.00001	0.1 and 0.0005	1	300000	13
PANet [**10**]	0.00001	0.9 and 0.0005	2	300000	16
ERNet [**11**]	0.00001	0.9 and 0.0005	64	300000	18
MAC [**12**]	0.01	–	–	160000	–
DLSD [**13**]	0.001	0.9 and 0.999	64	200000	21
DLSD-MV	0.001	0.9 and 0.999	64	200000	14

[1]https://www.nvidia.com/en-us/data-center/tesla-p100/

Table 6.2. Specification of the system used for training.

OS	Linux: Ubuntu 16.04.6LTS
GPU	NVIDIA Tesla P100-PCIE (16GB)
RAM	32 GB
Cuda Version	11.2
Framework	Tensorflow/ PyTorch/ Caffe2
Interpreter	Python 3.7

6.2. Dataset for Training and Evaluation

Applying DL methods for SOD is a challenging task. However, the most challenging is the very scarce training dataset. There are not many publicly available datasets for saliency detection using LF images. In this dissertation we used two available datasets for training and evaluating all the models. Therefore, to evaluate the effectiveness of all models, experiments were conducted on the DUTLF-V2 dataset [**109**]. The DUTLF-V2, can be considered as the updated version of DUTLF-V1 dataset. The DUTLF-V2 dataset consists of 4204 LF samples, divided into ten main classes, which can be further divided into 102 sub-classes. It covers diverse realistic scenes, and increased by 18% from the previous version. More explicitly, each LF of DUTLF-V2 contains six elements, including (i) the raw LF data, (ii) a AiF 2D image, (iii) a saliency mask, (iv) a depth map, (v) a stack of 12 focal slices, and (vi) an array of 81 SAIs with a spatial resolution of 600×400 pixels corresponding to a 9×9 angular resolution. Figure 6.1 illustrates a sample LF of the DUTLF-V2 dataset. More specifications of the DUTLF-V2 dataset are given in Table 6.3. The complete dataset is available at `https://github.com/OIPLab-DUT/DUTLF-V2`. It is important to mention that the DUTLF-V2 dataset was used for training and testing because it is one of those available datasets large enough to train DL models.

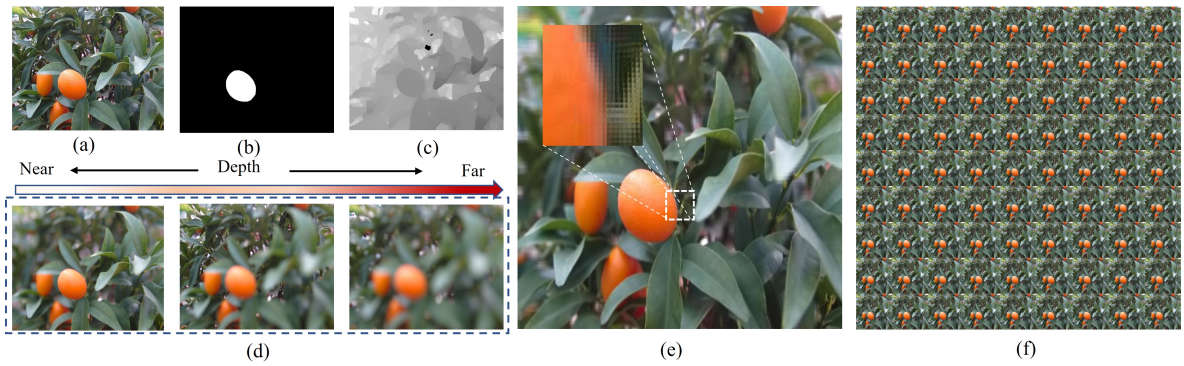


Figure 6.1. Example of the LF data elements in the DUTLF-V2 dataset: (a) All-in-focus, (b) SOD ground-truth, (c) Depth map, (d) Focal stack, (e) Micro-images, and (f) Sub-aperture images (multiview).

The complexity of the five deep networks described in the previous subsections makes them more difficult to optimize and easy to get overfitting, due to the limited amount of training data available. Therefore, to better evaluate how each model generalizes for different datasets for the task of LF SOD, we used a small dataset named LFSD [**99**] to evaluate each model.

Table 6.3. Overview of the used LF SOD datasets. About the abbreviations: FS=Focal Stacks, DE=Depth maps, MV=Multiview images, ML=Microlens images, GT=Ground-Truth, RAW=Raw LF data.

Dataset	Year	Size	FS	MV	DE	ML	GT	RAW
DUTLF-V2 [**109**]	2021	4204	x	x	x	x	x	x
LFSD [**99**]	2014	100	x		x		x	x

The LFSD dataset was the first LF dataset collected for SOD, which contains 60 indoor and 40 outdoor scenes, with each image having a spatial resolution of 360×360 pixels. This dataset was captured by a Lytro camera and provides: (i) all-in-focus image, (ii) SOD ground-truth, (iii) depth map, and (iv) focal stack, as illustrated in Figure 6.2. Unlike DUTLF-V2, the LFSD dataset is very small and does not have multiview and micro-images. More specifications of the LFSD dataset are given in Table 6.3.

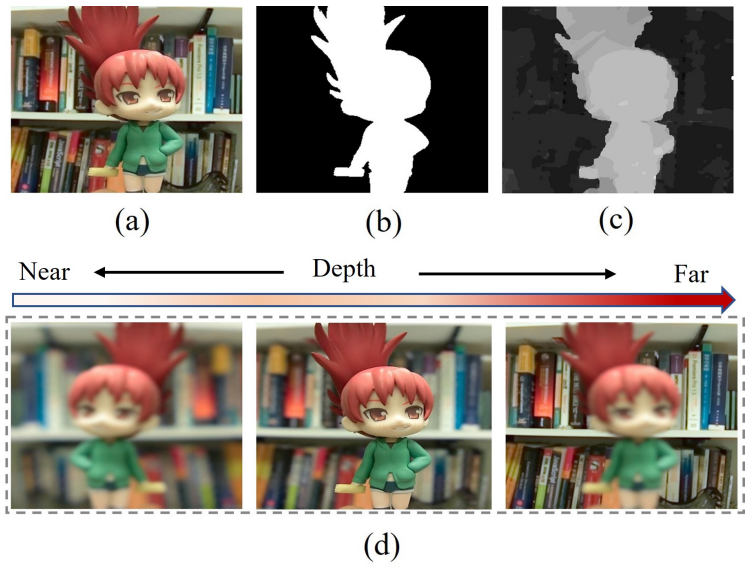


Figure 6.2. Example of the LF data elements in the LFSD dataset: (a) All-in-focus, (b) SOD ground-truth, (c) Depth map and (d) Focal stack.

6.3. Evaluation Metrics

In this section, we review five popular SOD evaluation metrics used usually to comprehensively examine the overall performance of numerous saliency detection methods. We undertake five assessment metrics: (i) Precision-Recall (PR) curve [**110**], (ii) Weighted F_β measure (F-measure) [**111**], (iii) Mean Absolute Error (MAE) [**112**], and recently proposed (iv) Structural measure (S-measure) [**113**] and (v) Enhance-alignment (E-measure) [**114**].

Precision-Recall Curve - Is calculated by conversion of predicted saliency map into binarized map and ground-truth. Thresholding of 0–255 is applied to produce a binary map. For all saliency maps present in the dataset, every binarizing threshold outcomes in a set of average precision and recall. An order of PR pair is generated when we vary the threshold from 0 to 1, which is used to plot the PR curve. Given a saliency map S, we can convert it to a binary mask M, and then compute the Precision and Recall by comparing M with ground-truth G:

$$\text{Precision} = \frac{|M \cap G|}{|M|} \tag{6.1}$$

$$\text{Recall} = \frac{|M \cap G|}{|G|} \tag{6.2}$$

A popular strategy is partitioning the saliency map S using a set of thresholds (from 0 to 255). For each threshold, we calculate a pair of precision and recall scores and then combine them to obtain a PR curve that describes the performance of the model as the threshold varies.

F-measure (F_β) - Is computed as a harmonic weighting means of Precision and Recall:

$$F_\beta = \frac{(1+\beta^2)\,\text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}} \tag{6.3}$$

where β^2 is set to 0.3 as suggested in [**111**], and is a parameter to control the weight of precision and recall metrics. Following the arguments presented by Borji *et al.* [**115**], where is stated that precision is more important than recall.

MAE - Computes the mean pixel-wise absolute difference between the predicted saliency map S and the binary ground-truth G, considering all pixels in the image, which can be defined by:

$$MAE = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} |G(i,j) - S(i,j)| \tag{6.4}$$

where W and H denote the width and height of the map, respectively, and the values of MAE are normalized to $[0, 1]$. A smaller MAE score relates to better performance as it reflects the high similarity between the estimated saliency map S and the ground-truth G.

S-measure (S_α) - Was proposed to measure the spatial structure similarities between the saliency map and ground-truth. It is defined as:

$$S_\alpha = \alpha^* S_0 + (1-\alpha) * s_\gamma \tag{6.5}$$

where S_α [**113**] is used to assess the structural similarity between the regional perception (S_γ) and object perception (S_0). $\alpha \in [0, 1]$ is a trade-off parameter, and $\alpha = 0,5$ is the default setting [**113**].

E-measure (E_ϕ) - Is a recently proposed metric which considers both the local and global similarity between the prediction and ground-truth. Thus, E_ϕ can be defined by:

$$E_{\phi} = \frac{1}{W * H} \sum_{i=1}^{W} \sum_{j=1}^{H} \phi_{FM}\left(i, j\right) \tag{6.6}$$

where ϕ_{FM} denotes the enhanced-alignment matrix [**114**]. W and H are the width and height of the ground-truth map, while (i, j) are pixel indexes.

6.4. Overall Quantitative Evaluation

In this section, we present the quantitative results from the experiments carried out in this dissertation. It is important to mention that in our experiments, we use the DUTLF-V2 dataset [**109**], and we split our data for all experiments following the common approach of putting 80% of of the data in the training set and 20% in the test set. It should also be noted that, for a fair comparison, all five deep models were compared by using the same training configuration. The experimental results obtained for the SA-Net [**9**], PANet [**10**], ERNet [**11**], DLSD [**13**], MAC [**12**] and C2FNet [**7**] models on the DUTLF-V2 dataset, are reported in Table 6.4.

A first conclusion that can be extracted from these experiments is that the performance of the five existing 4D deep architectures are different during the training process. Another important conclusion is that better performance was achieved with the PANet model in all metrics (i.e., F-measure, S-measure, E-measure and MAE). Moreover, it is worth mentioning that The PANet model exploits and integrates focal saliency regions that can contribute positively to accurate saliency detection. While the other saliency detection methods using LF exploit the data information of LF images in focal slices by processing the LF data in a slicewise manner.

One more straightforward conclusion that can be extracted from Table 6.4 is that the DLSD model performed significantly worse than the other 4D deep models. It is worth noting that the DLSD model uses as input a single 2D image (in the experiments performed in this dissertation the central view of each test LF has been used as input to the DLSD model). Therefore it does not use as input the spatial and angular information of the test LF content. Notice that, as described in Section 5.6, this model uses a high-quality LF synthesis network to produce reliable 4D LF information from a 2D image, but this process can lose information during the extraction of important features from multiview images.

Therefore, to understand the advantages of using 4D LF images for the SOD task, we compared the 4D LF SOD models with a 2D SOD model. We used the C2FNet model [**7**] (described in Subsection 4.2.2) because it is one of the state-of-the-art solutions for 2D SOD. Comparing the results of the five models under analysis, originally designed for 4D LF SOD, with the 2D model, we can see that all five 4D LF SOD models reach better results than the 2D model.

It is possible to conclude that using LF content, which contains angular information that can be exploited by taking advantage of refocusing on different objects in the scene, in the cases where a focal stack is used, or by exploiting the depth information (implicit or explicit) of the

Table 6.4. Quantitative comparisons of E-measure, S-measure, F-measure and MAE scores on the DUTLF-V2 dataset for different models. The best scores are in boldface.

Type	Methods	DUTLF-V2 [**109**]			
		$F_\beta\uparrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	MAE $\downarrow$
4D	SA-Net [**9**]	0.940	0.939	0.980	0.021
	PANet [**10**]	**0.964**	**0.973**	**0.991**	**0.010**
	ERNet [**11**]	0.936	0.892	0.941	0.033
	DLSD [**13**]	0.802	0.804	0.811	0.079
	MAC [**12**]	0.849	0.895	0.863	0.081
2D	C2FNet [**7**]	0.813	0.856	0.812	0.089

objects in the scene, leads to better performance results for SOD. Therefore, an important conclusion derived from the work of this dissertation is that it is advantageous to exploit angular and focal information instead of using only spatial features, as in the case of the 2D SOD solution. Figure 6.3 shows the PR Curves on DUTLF-V2 dataset. From this Figure it is possible to confirm that the PANet model outperforms other models independently of the used threshold. However, it is possible to see also that in terms of the PR curve, the MAC model performs worse that the other models.

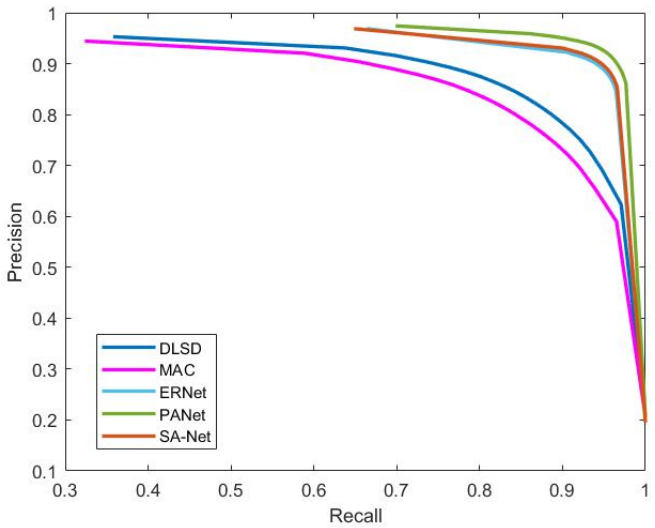


Figure 6.3. PR Curve for DUTLF-V2 Dataset. PR curves closer to the upper-right corner correspond to better performances.

6.5. Cross Evaluation on LFSD Dataset

Several researchers have pointed out that deeper and larger CNNs perform better than shallow networks. However, deep networks require a huge amount of data to be effectively trained and to prevent model overfitting. Therefore, to better analyze the efficiency of each model for 4D LF SOD, we firstly trained the five architectures in a large dataset as presented in Section 6.4. However, to further appraise the effectiveness of those models in SOD, we conducted an

Table 6.5. Quantitative comparisons of E-measure, S-measure, F-measure and MAE scores on LFSD dataset for different models. The best scores are in boldface.

Type	Methods	LFSD [**99**]			
		$F_\beta\uparrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	MAE $\downarrow$
4D	SA-Net [**9**]	0.866	0.856	0.930	0.056
	PANet [**10**]	**0.899**	**0.904**	**0.945**	**0.036**
	ERNet [**11**]	0.895	0.891	0.901	0.043
	DLSD [**13**]	0.788	0.818	0.893	0.091
	MAC [**12**]	0.805	0.802	0.882	0.150
2D	C2FNet [**7**]	0.718	0.771	0.799	0.124

experiment on a small dataset named LFSD [**99**] described in Section 6.2 in order to test the efficiency of each model on a small dataset.

As shown in Table 6.5 the PANet model achieved better results than all other models. Comparing the results of five 4D models with the C2FNet model, which is a 2D model, we can see that all the five 4D LF SOD provides better results than the 2D model. This evaluation indicates all five 4D LF SOD have better performance than to the 2D model. The Figure 6.4 show the PR Curves on LFSD dataset.

Therefore, Figure 6.5 show the comparative results for five representative 4D LF based SOD models on the DUTLF-V2 and LFSD datasets in terms of all evaluation metrics. All models perform better on the DUTLF-V2 dataset for all most all metrics. However, for the E-measure, Figure 6.5c, the DLSD and MAC models achieve better results on the LFSD dataset.

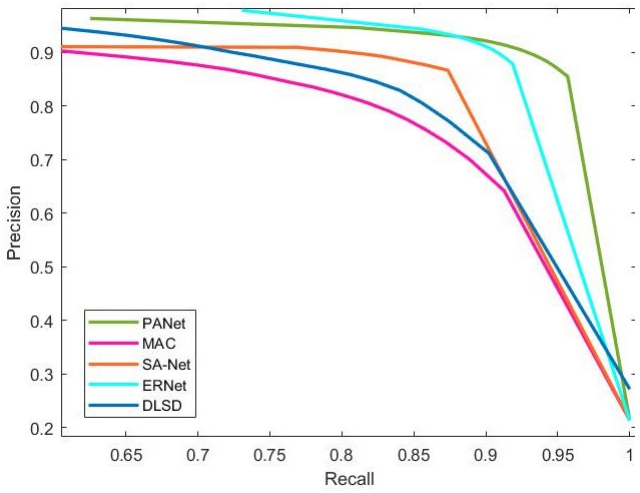


Figure 6.4. PR Curve for LFSD Dataset. PR curves closer to the upper-right corner correspond to better performances.

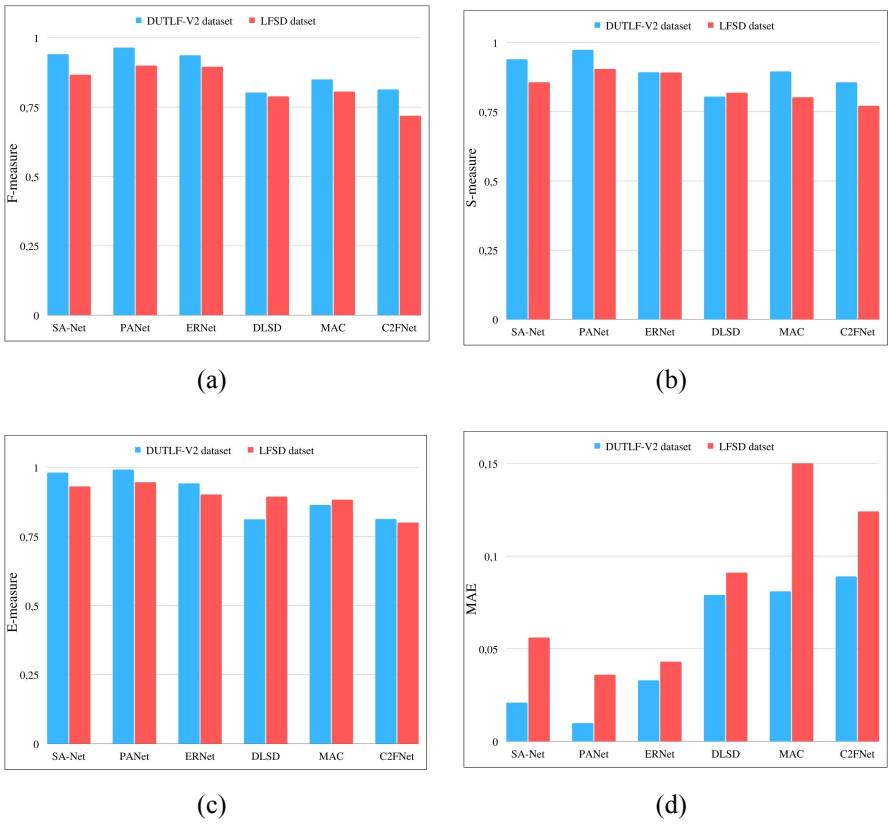


Figure 6.5. Comparative results for the five 4D LF SOD models (SA-Net, PANet, ERNet, DLSD, and MAC) and the 2D SOD model (C2FNet) under analysis on the DUTLF-V2 and LFSD datasets: (a) F-measure, (b) S-measure, (c) E-measure and (d) MAE.

6.6. Visual Analysis

In this section we qualitatively analyse the performance of the various 4D LF based SOD models showing saliency maps generated for various challenging scenes. Figure 6.6 shows some representative examples for five used models for 4D LF SOD and also the 2D based SOD model on the DUTLF-V2 dataset: SA-Net [**9**], PANet [**10**], ERNet [**11**], DLSD [**13**], MAC [**12**] and C2FNet [**7**].

Comparing the five 4D models, both achieve similar results. However, the PANet model generates better in different scenes. In these complex scenes, the PANet model can predict accurate and complete results when several salient objects or a big salient object is over a wide range of depth, as shown in rows (3 and 6), when the background is complex, as shown in row (5) and when foreground and background are similar, as shown in rows (3 and 4). Row (1) has low light. However, the SA-Net and the ERNet models also achieve good results, as shown in rows (2 and 4) of Figure 6.6.

Therefore, 4D LF SOD methods such as DLSD and MAC models can locate salient objects precisely due to the utilization of LF information but cannot predict complete salient objects due to small objects and complex background, as show in rows (3 and 7).

For more, compared to the 2D model, we can see that C2FNet [**7**] fail to detect accurate salient objects due to the lack of spatial information, as shown in rows (1, 5 and 7) of Figure 6.6. From the visual analyses it is clear that all five 4D models have better results in those challenging scenes, confirming the power of 4D LF models over the 2D model.

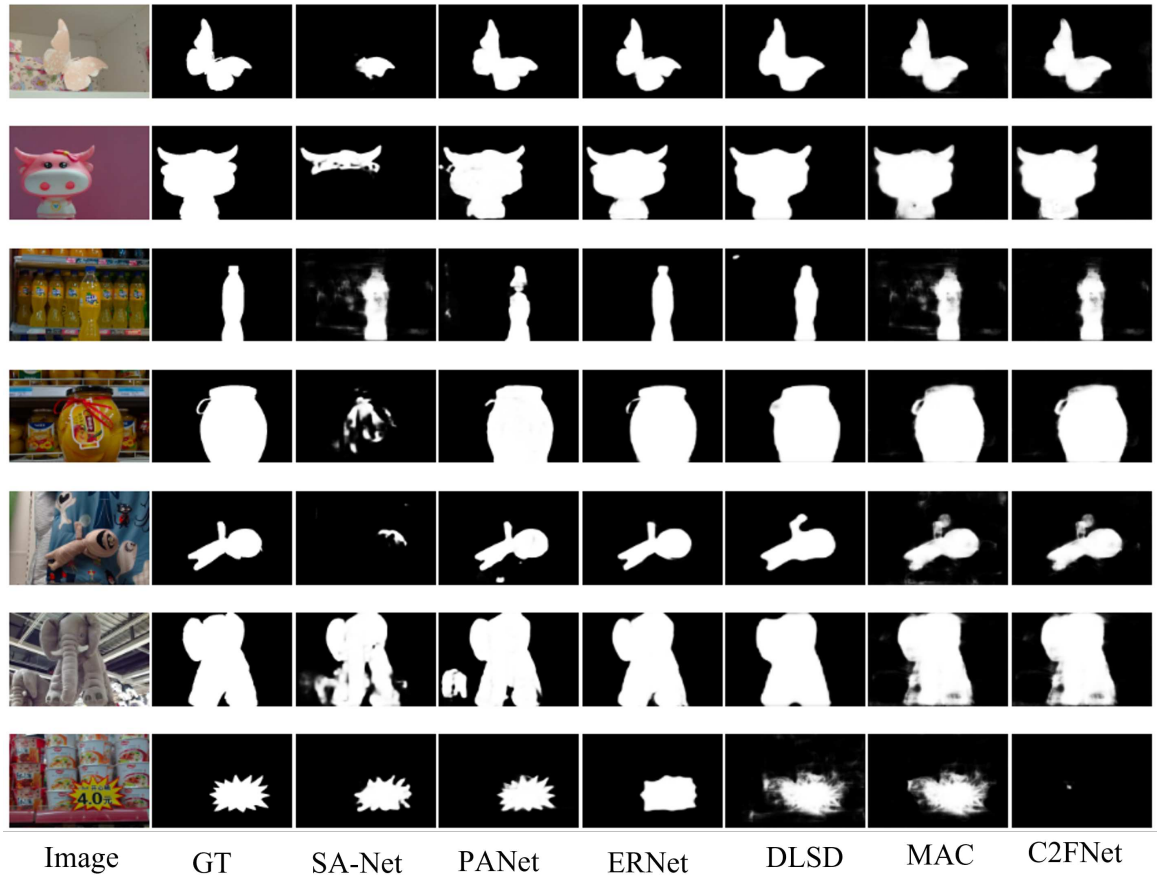


Figure 6.6. Visual comparison of the LF SOD on the DUTLF-V2 dataset.

As done for the quantitative results, to evaluate the performance of all five 4D models, including the 2D model, in different challenging scenes, we conducted the experiment on LFSD dataset [**7**], which is a challenging dataset for saliency detection with different challenges such as multiple objects, large objects, reflection, small object, low contrast and complex scene, and very small compared to DUTLF-V2 dataset. Figure 6.7 shows some visual results comparing the five models for 4D LF SOD and the 2D based SOD model on LFSD dataset.

As shown in Figure 6.7, the PANet model achieves better results than other models, as shown in rows (1 - 5). However, the SA-Net and the ERNet models achieve also have better results than the DLSD and the MAC models. Therefore, all the five 4D models auto-performance the C2FNet model, which is a 2D model. The C2FNet model [**7**] fails to detect accurate salient objects in the complex scene with multiple-object as shown in row (5).

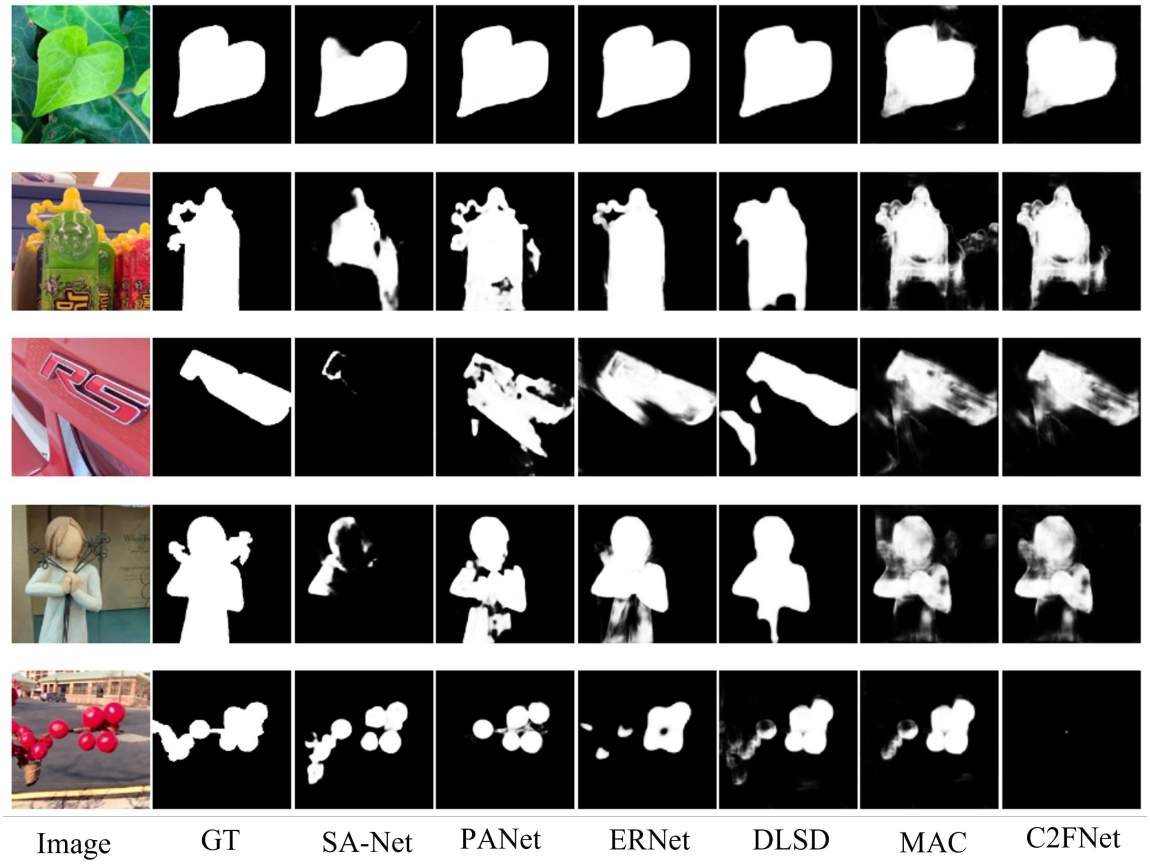


Figure 6.7. Visual comparison of the Light Field SOD on the LFSD dataset.

Table 6.6. Quantitative comparisons of DLSD and DLSD-MV models on the DUTLF-V2 dataset. The best scores are in boldface.

Type	Methods	DUTLF-V2 [**109**]			
		$F_\beta\uparrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	MAE $\downarrow$
4D	DLSD [**13**]	0.802	0.804	0.811	0.079
	DLSD-MV	**0**.860	**0**.867	**0**.911	**0**.071

6.7. Single View versus Multiview 4D LF SOD

In order to evaluate the proposed DLSD-MV model, we compare the result with the original model by using the DUTLF-V2 dataset. It is also important to mention that we used the same evaluation metrics described above to compare the models. As shown in Table 6.6 by using the proposed DLSD-MV model without the LF synthesis network module, we achieve better results than the DLSD model used from a single view image to produce reliable 4D LF information. Therefore, with the availability of a 4D LF image, it is not necessary to have the LF synthesis. Thus, the proposed model can be trained directly using a multiview image as input. However, comparing the results of the proposed DLSD-MV model with the existing models, as shown in Table 6.5, we can see that results are similar. PANet model outperformed all the existing models, including our proposed model. On the other hand, the DLSD and MAC models performed significantly worse than all deep models evaluated in this dissertation, including our proposed model.

Figure 6.8 shows visual comparisons of our proposed model with the DLSD model in some challenging scenes. It can be seen that the proposed model can achieve a more complete and accurate prediction when the foreground and background are similar, as shown in the rows (1 and 6), when the salient object is small as shown in the row (7) and when the background is complex as shown in the row (3).

From the visual results, it is worth noted that the proposed model without the LF synthesis can positively achieve better results. The result proved that it is more advantageous to use an LF image as input than 2D image for the task of SOD.

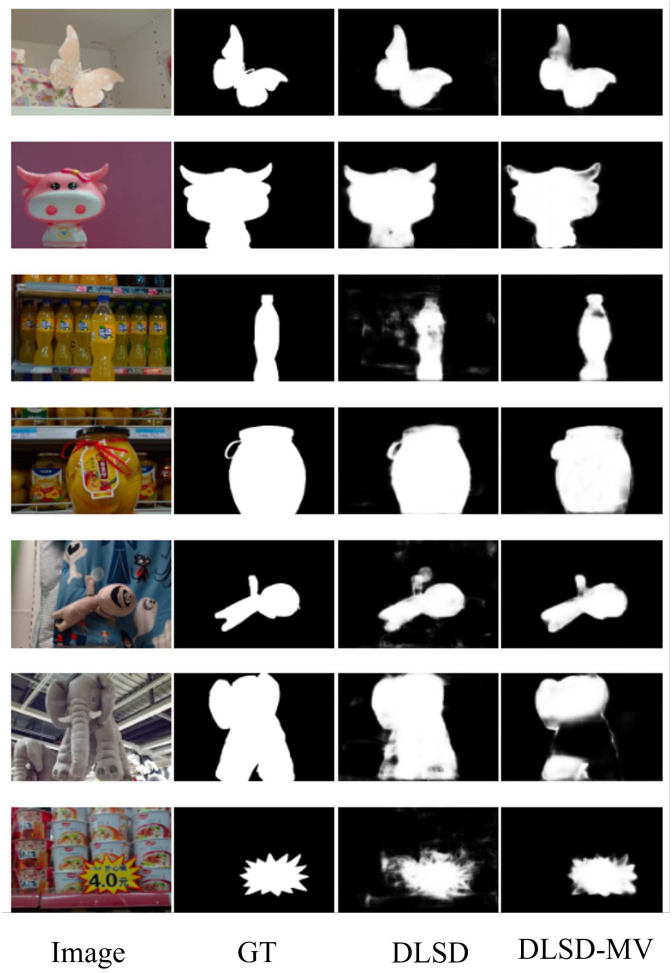


Figure 6.8. Visual comparison of the DLSD and DLSD-MV models on the DUTLF-V2 dataset.

6.8. Final Remarks

In this chapter, we presented the experimental setup used in this dissertation. Also, we presented the results from the experiments carried out in this dissertation. First, We presented the overall quantitative evaluation of the five existing 4D LF SOD models. After that, we compared the 4D LF models with the 2D SOD model. Then, we presented the cross dataset evaluation to see how the models generalize to different datasets, and we presented a visual comparison of the results. Finally, we presented the quantitative and qualitative results of the proposed model using LF

images as input that demonstrate the effectiveness of our proposed model in comparison to the existing DLSD model for 4D LF SOD.

CHAPTER 7

Conclusions and Future Work

In this chapter we summarize the major achievements of this dissertation and highlight the more relevant conclusions of this work. In addition, we will also discuss some possible future work directions along DL and 4D LF imaging.

7.1. Conclusions

In this dissertation, we have investigated the effectiveness of five existing deep models (i.e., SA-Net [**9**], PANet [**10**], ERNet [**11**], MAC [**12**] and DLSD [**13**]) for 4D LF SOD. These models have been chosen as they represent, to the best of our knowledge, the state-of-the-art on 4D LF SOD for the various types of DL architectures proposed for 4D LF SOD, i.e., late-fusion, middle-fusion, knowledge distillation-based, single-stream, and reconstruction-based. We have evaluated and analyzed the five existing deep networks by pre-training the existing models following a common strategy for exploiting the deep model.

Additionally, we have proposed an alternative solution to the existing DLSD model for 4D LF SOD – the DLSD-MV model. When comparing the existing DLSD model to our proposed model (DLSD-MV), a straightforward advantage is the use of 4D LF images as input, which can benefit the task of SOD. To demonstrate the effectiveness of the 4D LF SOD approach based on deep models, we have also compared the above models with the sate-of-the-art on 2D SOD approach based on DL (i.e., C2FNet [**7**]).

Therefore, to achieve the objectives defined for this dissertation (see Section 1.3), we have conducted experiments on two datasets (DUTLF-V2 [**109**] and LFSD [**99**]) with different visual properties to better evaluate the effectiveness of the five existing deep architectures and our proposed one under different demanding conditions. In our first set of experiments, we used the DUTLF-V2 dataset [**109**], which is a challenging dataset that includes LFs with, for example, low contrast, complex occlusions, multiple objects, large objects and similar foreground and background. Those challenges are very difficult to handle while adhering to the high prediction requirements of SOD. The second set of experiments was conducted on the LFSD dataset [**99**], which is very small and does not have multiview and microlens images.

The extensive experimental results obtained in this dissertation show that the five existing 4D LF deep models investigated achieve different, though similar, results when compared under the same test conditions on both datasets. However, the PANet model performs better than all other existing 4D LF SOD models investigated in this dissertation. Moreover, it is important to notice that the PANet model achieved better results on the DUTLF-V2 dataset than on the

LFSD one. It is also important to notice that the DLSD model performed significantly worse than most of the other 4D LF models, followed closely by the MAC model. Notwithstanding the excellent performance of all models on the two datasets, the results indicate that the PANet model, which is based on a middle-fusion architecture (see Figure 5.1) is a better option for 4D LF SOD. With this conclusion, we answer the third research question – *Which existing DL architectures perform better for LF SOD*?

Therefore, the model based on microlens images, such as the MAC model, is not effective as those based on focal stacks, such as the PANet and ERNet models, which contain a series of images focusing on different depths. However, we should carefully choose the different focal planes. This shows that the LF input format has an important influence the performance results of 4D LF SOD solutions. This answers our second research question – *Does the type of LF data format influence the performance of LF SOD*?

Furthermore, apart from investigating the effectiveness of the five existing 4D LF SOD deep models, we proposed a new model based on the existing DLSD model. This model was chosen because it performed significantly worse than other deep models for 4D LF SOD. We changed the DLSD model by removing the LF synthesis module from the model. Therefore, we proposed a model named DLSD-MV, which uses LFs as input directly instead of 2D images (as in the DLSD model). The experimental results show that the proposed model archives better results when compared to the original DLSD model. Thus allowing us to conclude that LF imaging brings added performance to the SOD task.

Additionally, comparing the results of the 4D LF SOD models to the state-of-the-art C2FNet 2D model, we could see that all existing 4D LF models outperform the 2D model. From the results obtained, we may conclude that the reason for this is the fact that LF data acquired by a Plenoptic Camera records more comprehensive and complete information of natural scenes, covering, for example, depth information, focusness cues, as well as different angular perspectives. These properties can benefit SOD in three different ways: Firstly, LFs can be refocused after being acquired, by enabling the generation of a stack of images focusing on different depths, providing, this way, focusness cues that have been demonstrated useful for SOD. Secondly, an LF can provide images of a scene from an array of viewpoints and such images have abundant spatial parallax and geometric information. Lastly, the depth information of a scene is embedded in the LF data and can be estimated from a focal stack or from multiview images by different means. These results answer our main research question – *What are the main advantages of using LF imaging for SOD*?

To conclude this work, we suggest that DL methods should be a good option for the problem of 4D LF SOD. This is supported by the results obtained for 4D LF SOD by using DL methods, where we can see that 4D LF models were able to match and even outperform 2D models. However, despite we have achieved excellent results in this work, future investigation should

be conducted to further improve the SOD task. In the following section, we describe some promising directions to explore in the future to pursue this goal.

7.2. Future Work

The conclusions drawn in this work are a step forward to gather a systematic and high-level view of the SOD task using LF content to further launch solid and consistent advancements in the field of LF SOD. However, there are still some open issues that deserve to be considered. In addition, it can still be useful to compare existing and future DL techniques with other existing traditional methods for LF SOD to infer the most effective and efficient ways of solving this problem.

We envision much future research on DL for LF SOD following this dissertation. Regarding future work, it would be interesting to develop and test a new model for LF SOD, which would be able to receive any LF format as input, for instance, multiview, focal stack, all-in-focus, and so on. Additionally, testing different hyper-parameters to achieve even higher performance results for LF SOD, while maintaining a manageable level of computational complexity, is another research direction that deserves some attention in the near future.

Having a large dataset for any LF format is also a challenging task, since DL methods require large datasets to be trained efficiently. Thus, another important research direction is the development of large annotated LF datasets, namely for the multiview format, enough to efficiently train any DL model used for the LF SOD task.

Finally, another important research direction on LF SOD is the development of an efficient CNN architecture capable of generating optimal focal planes for LF content, since focal stack images have proven to work well for tasks such as saliency detection.

Bibliography

[1] Caroline Conti, Luis Ducla Soares, and Paulo Nunes. Dense Light Field Coding: A Survey. *IEEE Access*, 8:49244–49284, 2020.

[2] Marc Levoy and Pat Hanrahan. Light field rendering. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1996*, pages 31–42, 1996.

[3] Matthieu Hog. *Light field editing and rendering*. PhD thesis, Université Rennes 1, November 2018.

[4] Neus Sabater, Guillaume Boisson, Benoit Vandame, Paul Kerbiriou, Frederic Babon, Matthieu Hog, Remy Gendrot, Tristan Langlois, Olivier Bureller, Arno Schubert, and Valerie Allie. Dataset and Pipeline for Multi-view Light-Field Video. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017-July:1743–1753, August 2017.

[5] Weitian Tong, Lixin Li, Xiaolu Zhou, Andrew Hamilton, and Kai Zhang. Deep learning PM2.5 concentrations with bidirectional LSTM RNN. *Air Quality, Atmosphere and Health*, 12(4):411–423, April 2019.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:770–778, December 2016.

[7] Yujia Sun, Geng Chen, Tao Zhou, Yi Zhang, and Nian Liu. Context-aware Cross-level Fusion Network for Camouflaged Object Detection. *IJCAI International Joint Conference on Artificial Intelligence*, pages 1025–1031, May 2021.

[8] Keren Fu, Yao Jiang, Ge Peng Ji, Tao Zhou, Qijun Zhao, and Deng Ping Fan. Light field salient object detection: A review and benchmark. *Computational Visual Media*, 8(4):509–534, dec 2022.

[9] Yi Zhang, Geng Chen, Qian Chen, Yujia Sun, Yong Xia, Olivier Deforges, Wassim Hamidouche, and Lu Zhang. Learning Synergistic Attention for Light Field Salient Object Detection. *The 32nd British Machine Vision Conference (BMVC)*, November 2021.

[10] Yongri Piao, Yongyao Jiang, Miao Zhang, Jian Wang, and Huchuan Lu. PANet: Patch-Aware Network for Light Field Salient Object Detection. *IEEE Transactions on Cybernetics*, 2021.

[11] Yongri Piao, Zhengkun Rong, Miao Zhang, and Huchuan Lu. Exploit and replace: An asymmetrical two-stream architecture for versatile light field saliency detection. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, 34(07):11865–11873, April 2020.

[12] Jun Zhang, Yamei Liu, Shengping Zhang, Ronald Poppe, and Meng Wang. Light Field Saliency Detection with Deep Convolutional Networks. *IEEE Transactions on Image Processing*, 29:4421–4434, June 2020.

[13] Yongri Piao, Zhengkun Rong, Miao Zhang, Xiao Li, and Huchuan Lu. Deep light-field-driven saliency detection from a single view. *IJCAI International Joint Conference on Artificial Intelligence*, 2019-August:904–911, 2019.

[14] Hao Chen, Qi Dou, Lequan Yu, Jing Qin, and Pheng Ann Heng. VoxResNet: Deep voxelwise residual networks for brain segmentation from 3D MR images. *NeuroImage*, 170:446–455, April 2018.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):386–397, March 2020.

[16] A. Gershun. The Light Field. *Journal of Mathematics and Physics*, 18(1-4):51–151, 1939.

[17] Edward H Adelson and James R Bergen. The Plenoptic Function and the Elements of Early Vision. *Computational Models of Visual Processing*, pages 3–20, 2020.

[18] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1996*, pages 43–54, August 1996.

[19] Raytrix GmbH. Raytrix 3D Light Field Camera Technology, 2017.

[20] Marc Levoy, Ren Ng, Andrew Adams, Matthew Footer, and Mark Horowitz. Light field microscopy. *ACM SIGGRAPH 2006 Papers, SIGGRAPH '06*, pages 924–934, 2006.

[21] Tiantian Wang, Yongri Piao, Huchuan Lu, Lihe Zhang, and Xiao Li. Deep learning for light field saliency detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October:8837–8847, 2019.

[22] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Marc Levoy, and Mark Horowitz. High-speed videography using a dense camera array. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2, 2004.

[23] Jason C. Yang. A Light Field Camera For Image Based Rendering. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, page 55, 2000.

[24] Cha Zhang and Tsuhan Chen. A self-reconfigurable camera array. *ACM SIGGRAPH 2004 Sketches, SIGGRAPH'04*, 2004.

[25] Da□bała, M. Ziegler, P. Didyk, F. Zilly, J. Keinert, K. Myszkowski, H. P. Seidel, P. Rokita, and T. Ritschel. Efficient Multi-image Correspondences for On-line Light Field Video Processing. *Computer Graphics Forum*, 35(7):401–410, October 2016.

[26] Kartik Venkataraman, Dan Lelescu, Jacques Duparrë, Andrew McMahon, Gabriel Molina, Priyam Chatterjee, Robert Mullism, and Shree Nayar. PiCam: An ultra-thin high performance monolithic camera array. *ACM Transactions on Graphics*, 32(6), 2013.

[27] L. Gabriel. La photographie intégrale. *Comptes-Rendus, Académie des Sciences*, 146:446–551, 1908.

[28] Robert C. Bolles, H. Harlyn Baker, and David H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, March 1987.

[29] Andrew Lumsdaine and Todor Georgiev. The focused plenoptic camera. *2009 IEEE International Conference on Computational Photography, ICCP 09*, 2009.

[30] Sreenath Krishnan. Capturing wide field of view light fields using spherical mirrors. *2009 IEEE International Conference on Computational Photography, ICCP 09*, pages 1–4, 2010.

[31] Dorian Tsai, Donald G. Dansereau, Thierry Peynot, and Peter Corke. Image-Based Visual Servoing with Light Field Cameras. *IEEE Robotics and Automation Letters*, 2(2):912–919, 2017.

[32] Yuichi Taguchi, Amit Agrawal, Srikumar Ramalingam, and Ashok Veeraraghavan. Axial light field for curved mirrors: Reflect your perspective, widen your view. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 499–506, 2010.

[33] Ren Ng, Marc Levoy, Mathieu Bredif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light Field Photography with a Hand-held Plenoptic Camera. *Main*, pages 1–11, 2005.

[34] Gaochang Wu, Mandan Zhao, Liangyong Wang, Qionghai Dai, Tianyou Chai, and Yebin Liu. Light field reconstruction using deep convolutional network on EPI. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:1638–1646, November 2017.

[35] Gaochang Wu, Yebin Liu, Qionghai Dai, and Tianyou Chai. Learning Sheared EPI Structure for Light Field Reconstruction. *IEEE Transactions on Image Processing*, 28(7):3261–3273, July 2019.

[36] Nima Khademi Kalantari, Ting Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics*, 35(6), November 2016.

[37] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[38] Ossama Abdel-Hamid, Abdel Rahman Mohamed, Hui Jiang, and Gerald Penn. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 4277–4280, 2012.

[39] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *32nd International Conference on Machine Learning, ICML 2015*, 1:448–456, February 2015.

[40] Ruwan Nawarathna, Jung Hwan Oh, Jayantha Muthukudage, Wallapak Tavanapong, Johnny Wong, Piet C. de Groen, and Shou Jiang Tang. Abnormal image detection in endoscopy videos using a filter bank and local binary patterns. *Neurocomputing*, 144:70–91, November 2014.

[41] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel Rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[42] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, and Case et al. Deep speech 2: End-to-end speech recognition in English and Mandarin. *33rd International Conference on Machine Learning, ICML 2016*, 1:312–321, December 2016.

[43] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, September 2015.

[44] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, January 1989.

[45] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, September 2015.

[47] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June-2015:1–9, 2015.

[48] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-ResNet and the impact of residual connections on learning. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pages 4278–4284, February 2017.

[49] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January(4):2261–2269, 2017.

[50] Minh Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, August 2015.

[51] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. DeepDriving: Learning affordance for direct perception in autonomous driving. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 International Conference on Computer Vision, ICCV 2015:2722–2730, 2015.

[52] Jian Zhou and Olga G. Troyanskaya. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods*, 12(10):931–934, September 2015.

[53] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. No-scope: Optimizing neural network queries over video at scale. *Proceedings of the VLDB Endowment*, 10(11):1586–1597, 2017.

[54] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G. Dimakis. Compressed sensing using generative models. *34th International Conference on Machine Learning, ICML 2017*, 2:822–841, March 2017.

[55] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:105–114, September 2017.

[56] Leon Gatys, Alexander Ecker, and Matthias Bethge. A Neural Algorithm of Artistic Style. *Journal of Vision*, 16(12):326, August 2016.

[57] Jun Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9909 LNCS:597–613, September 2016.

[58] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[59] Martín Abadi Ashish Agarwal et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[60] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[61] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM.

[62] The Theano Development Team et al. Theano: A Python framework for fast computation of mathematical expressions, May 2016.

[63] Caffe2. Caffe2 | A New Lightweight, Modular, and Scalable Deep Learning Framework, 2020.

[64] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, June 2017.

[65] Wenguan Wang, Jianbing Shen, Jianwen Xie, Ming Ming Cheng, Haibin Ling, and Ali Borji. Revisiting Video Saliency Prediction in the Deep Learning Era. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):220–237, January 2021.

[66] Yin Li, Xiaodi Hou, Christof Koch, James M. Rehg, and Alan L. Yuille. The secrets of salient object segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 280–287, June 2014.

[67] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. Online tracking by learning discriminative saliency map with convolutional neural network. *32nd International Conference on Machine Learning, ICML 2015*, 1:597–606, February 2015.

[68] Ling Shao and Michael Brady. Specific object retrieval based on salient regions. *Pattern Recognition*, 39(10):1932–1948, October 2006.

[69] Dashan Gao, Vijay Mahadevan, and Nuno Vasconcelos. The discriminant center-surround hypothesis for bottom-up saliency. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 497–504, Red Hook, NY, USA, 2007. Curran Associates Inc.

[70] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June-2015:1265–1274, October 2015.

[71] Houwen Peng, Bing Li, Weihua Xiong, Weiming Hu, and Rongrong Ji. RGBD salient object detection: A benchmark and algorithms. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8691 LNCS(PART 3):92–109, 2014.

[72] Hao Chen and Youfu Li. Three-Stream Attention-Aware Network for RGB-D Salient Object Detection. *IEEE Transactions on Image Processing*, 28(6):2825–2835, June 2019.

[73] Tie Liu, Jian Sun, Nan Ning Zheng, Xiaoou Tang, and Heung Yeung Shum. Learning to detect a salient object. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[74] Paria Mehrani and Olga Veksler. Saliency segmentation based on learning and graph cut refinement. *British Machine Vision Conference, BMVC 2010 - Proceedings*, 2010.

[75] Jimei Yang and Ming Hsuan Yang. Top-down visual saliency via joint CRF and dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(3):576–588, March 2017.

[76] Runmin Cong, Jianjun Lei, Changqing Zhang, Qingming Huang, Xiaochun Cao, and Chunping Hou. Saliency Detection for Stereoscopic Images Based on Depth Confidence Analysis and Multiple Cues Fusion. *IEEE Signal Processing Letters*, 23(6):819–823, June 2016.

[77] Jingfan Quo, Tongwei Ren, and Jia Bei. Salient object detection for RGB-D image via saliency evolution. *Proceedings - IEEE International Conference on Multimedia and Expo*, 2016-August, August 2016.

[78] David Feng, Nick Barnes, Shaodi You, and Chris McCarthy. Local Background Enclosure for RGB-D Salient Object Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:2343–2350, December 2016.

[79] Yanlong Tang, Ruofeng Tong, Min Tang, and Yun Zhang. Depth incorporating with color improves salient object detection. *Visual Computer*, 32(1):111–121, January 2016.

[80] Huan Du, Zhi Liu, Hangke Song, Lin Mei, and Zheng Xu. Improving RGBD saliency detection using progressive region classification and saliency fusion. *IEEE Access*, 4:8987–8994, 2016.

[81] Hao Chen, Youfu Li, and Dan Su. RGB-D saliency detection by multi-stream late fusion network. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10528 LNCS:459–468, 2017.

[82] Fangfang Liang, Lijuan Duan, Wei Ma, Yuanhua Qiao, Zhi Cai, and Laiyun Qing. Stereoscopic saliency model using contrast and depth-guided-background prior. *Neurocomputing*, 275:2227–2238, January 2018.

[83] Arridhana Ciptadi, Tucker Hermans, and James Rehg. An In Depth View of Saliency. *Georgia Institute of Technology*, pages 112.1–112.11, January 2014.

[84] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[85] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. *31st International Conference on Machine Learning, ICML 2014*, 2:988–996, 2014.

[86] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming Hsuan Yang. Deep networks for saliency detection via local estimation and global search. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June-2015:3183–3192, October 2015.

[87] Gayoung Lee, Yu Wing Tai, and Junmo Kim. Deep Saliency with Encoded Low level Distance Map and High Level Features. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:660–668, April 2016.

[88] Nian Liu and Junwei Han. DHSNet: Deep hierarchical saliency network for salient object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:678–686, December 2016.

[89] Qibin Hou, Ming-Ming Cheng, Xiao-Wei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):815–828, November 2016.

[90] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating Multi-level Convolutional Features for Salient Object Detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-October:202–211, August 2017.

[91] Zhiming Luo, Akshaya Mishra, Andrew Achkar, Justin Eichel, Shaozi Li, and Pierre Marc Jodoin. Non-local deep features for salient object detection. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:6593–6601, 2017.

[92] Zijun Deng, Xiaowei Hu, Lei Zhu, Xuemiao Xu, Jing Qin, Guoqiang Han, and Pheng Ann Heng. R3Net: Recurrent residual refinement network for saliency detection. *IJCAI International Joint Conference on Artificial Intelligence*, 2018-July:684–690, 2018.

[93] Nian Liu, Junwei Han, and Ming Hsuan Yang. PiCANet: Learning Pixel-Wise Contextual Attention for Saliency Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3089–3098, 2018.

[94] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive Attention Guided Recurrent Network for Salient Object Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 714–722, 2018.

[95] Zhe Wu, Li Su, and Qingming Huang. Cascaded partial decoder for fast and accurate salient object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:3902–3911, April 2019.

[96] Jiaxing Zhao, Jiang Jiang Liu, Deng Ping Fan, Yang Cao, Jufeng Yang, and Ming Ming Cheng. EGNet: Edge guidance network for salient object detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October:8778–8787, 2019.

[97] Runmin Cong, Jianjun Lei, Huazhu Fu, Junhui Hou, Qingming Huang, and Sam Kwong. Going from RGB to RGBD Saliency: A Depth-Guided Transformation Model. *IEEE Transactions on Cybernetics*, 50(8):3627–3639, 2020.

[98] Chunbiao Zhu, Xing Cai, Kan Huang, Thomas H. Li, and Ge Li. PDNet: Prior-model guided depth-enhanced network for salient object detection. *Proceedings - IEEE International Conference on Multimedia and Expo*, 2019-July:199–204, March 2019.

[99] Nianyi Li, Jinwei Ye, Yu Ji, Haibin Ling, and Jingyi Yu. Saliency Detection on Light Field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8):1605–1616, 2017.

[100] Ali Borji, Ming Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *Computational Visual Media*, 5(2):117–150, June 2019.

[101] Hao Sheng, Shuo Zhang, Xiaoyu Liu, and Zhang Xiong. Relative location for light field saliency detection. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2016-May:1631–1635, 2016.

[102] Nianyi Li, Bilin Sun, and Jingyi Yu. A weighted sparse coding framework for saliency detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June-2015:5216–5223, 2015.

[103] Anzhi Wang, Minghui Wang, Xiaoyan Li, Zetian Mi, and Huan Zhou. A Two-Stage Bayesian Integration Framework for Salient Object Detection on Light Field. *Neural Processing Letters*, 46(3):1083–1094, 2017.

[104] Jun Zhang, Meng Wang, Liang Lin, Xun Yang, Jun Gao, and Yong Rui. Saliency detection on light field: A multi-cue approach. *ACM Transactions on Multimedia Computing, Communications and Applications*, 13(3), 2017.

[105] Miao Zhang, Jingjing Li, Wei Ji, Yongri Piao, and Huchuan Lu. Memory-oriented decoder for light field salient object detection. *Advances in Neural Information Processing Systems*, 32, 2019.

[106] Miao Zhang, Wei Ji, Yongri Piao, Jingjing Li, Yu Zhang, Shuang Xu, and Huchuan Lu. LFNet: Light Field Fusion Network for Salient Object Detection. *IEEE Transactions on Image Processing*, 29:6276–6287, 2020.

[107] Qiudan Zhang, Shiqi Wang, Xu Wang, Zhenhao Sun, Sam Kwong, and Jianmin Jiang. A Multi-Task Collaborative Network for Light Field Salient Object Detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(5):1849–1861, May 2021.

[108] Liang Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, April 2018.

[109] Yongri Piao, Zhengkun Rong, Shuang Xu, Miao Zhang, and Huchuan Lu. DUT-LFSaliency: Versatile Dataset and Light Field-to-RGB Saliency Detection, December 2020.

[110] Ming Ming Cheng, Guo Xin Zhang, Niloy J. Mitra, Xiaolei Huang, and Shi Min Hu. Global contrast based salient region detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 409–416, 2011.

[111] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. *IEEE Transactions on Image Processing*, pages 1597–1604, March 2010.

[112] Federico Perazzi, Philipp Krahenbuhl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 733–740, 2012.

[113] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A New Way to Evaluate Foreground Maps. *International Journal of Computer Vision*, 129(9):2622–2638, August 2017.

[114] Deng Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming Ming Cheng, and Ali Borji. Enhanced-alignment Measure for Binary Foreground Map Evaluation. *IJCAI International Joint Conference on Artificial Intelligence*, 2018-July:698–704, May 2018.

[115] Ali Borji, Ming Ming Cheng, Huaizu Jiang, and Jia Li. Salient Object Detection: A Benchmark. *IEEE Transactions on Image Processing*, 24(12):5706–5722, December 2015.