Yingyue Xu

COMPUTATIONAL MODELING FOR VISUAL ATTENTION ANALYSIS
YINGYUE XU

COMPUTATIONAL MODELING FOR VISUAL ATTENTION ANALYSIS

Academic dissertation to be presented, with the assent of the Doctoral Training Committee of Information Technology and Electrical Engineering of the University of Oulu, for public defence in the Oulun Puhelin auditorium (L5), Linnanmaa, on 5 June 2020, at 12 noon
Visual scenes typically contain massive amounts of content that cannot be processed in a short time due to the limited processing capacity of the human visual system. The term, visual attention, is a biologically inspired and psychologically driven mechanism that works by selecting visually relevant information and filtering out the redundant contents.

This thesis is a thorough summary of the main subjects around computational modeling for visual attention analysis, consisting of several published papers corresponding to my research progress. First, the data preparation for computational modeling will be introduced, including eye movement data, eye tracking data collection and eye tracking datasets facilitating the evaluation of computational modeling of visual attention. Second, computational models for visual attention analysis, or saliency models, are presented from traditional unsupervised methods to deep saliency models. Third, the subject about saliency integration will be illustrated that unifies multiple saliency maps from the multiple candidate saliency models for better accuracy.

The contributions of this study are three folds. Firstly, we collect a task-driven eye tracking dataset for visual attention analysis. Secondly, we propose three saliency models for in-depth investigation in modeling visual attention, including an unsupervised model using the bi-directional propagation method, a Convolutional Neural Networks based model by connecting the Dense Conditional Random Fields for multi-scale saliency refinement, and a Convolutional Neural Networks based model with cascade Conditional Random Fields for joint model training. Thirdly, we propose a saliency integration method and conduct comprehensive experiments and analysis on the topic.

Finally, we summarize the contributions of the work and propose the potential applications of saliency models and the extended saliency related topics to boost applications of saliency approaches on other computer vision topics.

Keywords: eye tracking data, saliency model, visual attention
Xu, Yingyue, Laskennallinen mallintaminen visuaalisen huomion analyysiin.  
Oulun yliopiston tutkijakoulu; Oulun yliopisto, Tieto- ja sähköteknikan tiedekunta  
*Acta Univ. Oul. C 753, 2020*  
Oulun yliopisto, PL 8000, 90014 Oulun yliopisto

**Tiivistelmä**

Kuvat sisältävät tyyppisesti valtavan määrän informaatiota, jota ei pystytä prosessoinnissa lyhyessä ajassa ihmisen näköjärjestelmän rajoitetun prosessointi kapasiteetin takia. Termi, visuaalinen huomio, on biologian ja psykologian motivaatiomekanismo, joka toimii valiten oleellisen informaation ja suodattamalla ylimääräisen informaation. Mallintaaaksemme huomion mekanismia koneenäön käyttöön on olennaisa, että laskennallinen malli visuaaliseen informaatioon ehdottaa tärkeitä alueita kuvasta, jotka ihmisen näköjärjestelmä on nähnyt.  

Tämä väitöskirja on perusteellinen yhteenveto kokeneiden osa-alueista liittyen visuaalisen huomion analyysiin laskennalliseen mallintamiseen, koostuen useasta julkaistusta vastaten minun tutkimuksen etenemiseen. Ensimmäiseksi, esittelemme datan esikäsittelyn laskennallista mallia varten, mukaan ottaen silmänliikkeen datan, silmänhyljitys datan kerääminen ja silmänjaljitys tietokantojen hyödyntäminen visuaalisen huomion laskennallisen mallien evaluoinnissa. Toiseksi, laskennalliset mallit visuaalisen huomioon, tai tärkeys mallit, esittelään perinteisiä ohjaamattomista menetelmiä syviin tärkeyds malleihin. Kolmanneksi, havainnollistamme tärkeyttä menettämistä, joka yhdistää useita tärkeitä ehdotuksia useista eri tärkeistä mallista saavutamme paremman tarkkuuden.  


Lopuksi, tiivistämme kontribuutioon tiivistämme ja ehdotamme mahdollisia sovelluksia tärkeysmallista ja laajennamme tärkeys-aiheeseen liittyviä sovelluksia tehostamaan tärkeys menetelmää eri konenöön aiheisiin.

**Asiasanat:** silmän jaljitys datan, tärkeys malli, visuaalinen huomio
To my family, and friends.
Acknowledgements

This dissertation is a compilation of several academic papers that summarize my research work in the Center of Machine Vision and Signal Analysis (CMVS), University of Oulu.

My research work on visual attention analysis is conducted under the supervision of Professor Guoying Zhao, who continuously supports my research works in CMVS from the master’s thesis to Ph.D. studies. Meanwhile, Doctor Xiaopeng Hong provides technical supervision during my research works. I would like to express deep gratitude to Guoying and Xiaopeng since they not only support my research works but also give me valuable suggestions on life and career.

I am grateful to all my co-authors. I would like to thank Doctor Wanli Ouyang and his SenseTime Computer Vision Group for sharing their in-depth understanding of deep learning during my research visit to the University of Sydney, Australia. I would also like to thank Doctor Dan Xu from the University of Oxford for discussions on conditional random fields and for his vast suggestions on revising our collaborated paper. I want to thank Associate Professor Min Xu from the University of Technology Sydney for hosting me during my research visit to Australia. I would also like to thank Doctor Jie Chen for giving me technical suggestions on deep learning. I shall thank Professor Fatih Porikli for his suggestions on paper revision.

In particular, I shall thank Professor Irene Gu from Chalmers University of Technology and Professor Simone Frintrop from the University of Hamburg for their comments and suggestions in reviewing my dissertation. They make this manuscript with improved quality.

I would also like to thank Professor Moncef Gabbouj from Tampere University of Technology for serving as the opponent in my defense.

Also, I am grateful to Nokia Foundation for the granted scholarship.

Last but not least, I would like to express deep gratitude to my parents for their everlasting support during all the stages of my life. And I would also like to thank my friends in Oulu who have been accompanied and supported me ever since my master’s studies and made my life in Oulu interesting.

In Oulu, Finland, 31th of March, 2020
Yingyue Xu
# List of abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC</td>
<td>Area Under the ROC Curve</td>
</tr>
<tr>
<td>BU</td>
<td>Bottom-up</td>
</tr>
<tr>
<td>CA</td>
<td>Cellular Automaton</td>
</tr>
<tr>
<td>CC</td>
<td>Correlation Coefficients</td>
</tr>
<tr>
<td>CNNs</td>
<td>Convolutional Neural Networks</td>
</tr>
<tr>
<td>CRF</td>
<td>Conditional Random Field</td>
</tr>
<tr>
<td>DBN</td>
<td>Dynamic Bayesian Networks</td>
</tr>
<tr>
<td>EOG</td>
<td>Electrooculography</td>
</tr>
<tr>
<td>FCNs</td>
<td>Fully Convolutional Neural Networks</td>
</tr>
<tr>
<td>FP</td>
<td>Fixation Prediction</td>
</tr>
<tr>
<td>GMM</td>
<td>Gaussian Mixture Model</td>
</tr>
<tr>
<td>HCI</td>
<td>Human Computer Interaction</td>
</tr>
<tr>
<td>HVS</td>
<td>Human Visual System</td>
</tr>
<tr>
<td>ICL</td>
<td>Incremental Coding Length</td>
</tr>
<tr>
<td>IR</td>
<td>Infrared</td>
</tr>
<tr>
<td>MAE</td>
<td>Mean Absolute Error</td>
</tr>
<tr>
<td>PCR</td>
<td>Pupil-corneal Reflection</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic</td>
</tr>
<tr>
<td>ROI</td>
<td>Regions of Interest</td>
</tr>
<tr>
<td>SOD</td>
<td>Salient Object Detection</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>TD</td>
<td>Top-down</td>
</tr>
<tr>
<td>VOG</td>
<td>Videooculography</td>
</tr>
</tbody>
</table>
## List of symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I$</td>
<td>Image</td>
</tr>
<tr>
<td>$F$</td>
<td>Predicted Fixation</td>
</tr>
<tr>
<td>$\bar{F}$</td>
<td>Mean Fixation Value</td>
</tr>
<tr>
<td>$G$</td>
<td>Ground Truth</td>
</tr>
<tr>
<td>$\bar{G}$</td>
<td>Mean Ground Truth Value</td>
</tr>
<tr>
<td>$S_F$</td>
<td>Foreground Map</td>
</tr>
<tr>
<td>$S_B$</td>
<td>Background Map</td>
</tr>
<tr>
<td>$P$</td>
<td>Probability</td>
</tr>
<tr>
<td>$\text{min}$</td>
<td>Minimum</td>
</tr>
<tr>
<td>$\text{max}$</td>
<td>Maximum</td>
</tr>
<tr>
<td>$\exp$</td>
<td>Expectation</td>
</tr>
<tr>
<td>$\text{logit}$</td>
<td>Logit</td>
</tr>
<tr>
<td>$\ln$</td>
<td>In Function</td>
</tr>
<tr>
<td>$\text{sign}(\cdot)$</td>
<td>The Sign of The Value</td>
</tr>
<tr>
<td>$i, j, k, m, n, t$</td>
<td>Scalar Index Variable</td>
</tr>
<tr>
<td>$K, M, N, T, L$</td>
<td>Non-negative Integers</td>
</tr>
<tr>
<td>$|\cdot|_2$</td>
<td>$l_2$ Norm</td>
</tr>
<tr>
<td>$|\cdot|_1$</td>
<td>$l_1$ Norm</td>
</tr>
<tr>
<td>$\varphi$</td>
<td>CIELab Features</td>
</tr>
<tr>
<td>$\mu$</td>
<td>CIELab-XY Features</td>
</tr>
<tr>
<td>$x_n, y_n$</td>
<td>Positions of The n-th Unit in X-Y Space</td>
</tr>
<tr>
<td>$W$</td>
<td>Generic Weights</td>
</tr>
<tr>
<td>$w$</td>
<td>Index Specific Weights</td>
</tr>
<tr>
<td>$D$</td>
<td>Diagonal Degree Matrix</td>
</tr>
<tr>
<td>$\alpha, \beta, \alpha$</td>
<td>Attenuation Parameter</td>
</tr>
<tr>
<td>$\mathcal{L}$</td>
<td>Labeled Set</td>
</tr>
<tr>
<td>$\mathcal{U}$</td>
<td>Unlabeled Set</td>
</tr>
<tr>
<td>$\mathcal{P}$</td>
<td>Selected Propagation Set</td>
</tr>
<tr>
<td>$\mathcal{C}$</td>
<td>Candidate Set for Propagation Selection</td>
</tr>
<tr>
<td>$d$</td>
<td>Difficulty</td>
</tr>
<tr>
<td>$f_n$</td>
<td>Saliency Value of The n-th Superpixel</td>
</tr>
<tr>
<td>$b_n$</td>
<td>Unsaliency Value of The n-th Superpixel</td>
</tr>
<tr>
<td>$s$</td>
<td>Scale Specific Prediction</td>
</tr>
<tr>
<td>$f$</td>
<td>Scale Specific Features</td>
</tr>
</tbody>
</table>
Scale Specific Hidden Prediction
Scale Specific Hidden Features
Thresholded Saliency Prediction
Encoder-decoder Networks
Scale Specific Parameters
Unary Term
Pairwise Term
Gaussian Kernel
Contribution of The Gaussian Kernel
Position
Bandwidth of The Gaussian Kernel
Energy Function
Logarithm of The Posterior Ratio
List of original publications

This dissertation is based on the following articles, which are referred to in the text by their Roman numerals (I–V):


This dissertation only collects the papers that closely related to my Ph.D. research topics. As the first author, I carried out all the experiments and wrote the first drafts. My co-authors gave me valuable suggestions on experiments and paper revision.
Contents

Abstract
Tiivistelmä
Acknowledgements 9
List of abbreviations 11
List of symbols 13
List of original publications 15
Contents 17
1 Introduction 19
  1.1 Visual attention analysis ........................................... 19
  1.2 Objectives ................................................................... 21
  1.3 Summary of original articles ........................................ 22
  1.4 Organization of the thesis .......................................... 23
2 Saliency data collection 25
  2.1 Introduction .................................................................. 25
    2.1.1 Eye movements ...................................................... 25
    2.1.2 Eye trackers ............................................................ 27
  2.2 Related work .................................................................. 28
    2.2.1 Free viewing datasets ............................................ 28
    2.2.2 Task driven datasets .............................................. 29
  2.3 Task-driven eye tracking dataset with multiple semantics .... 30
    2.3.1 Task specific watching materials ............................... 30
    2.3.2 Experimental environment .................................... 30
    2.3.3 Ground truth .......................................................... 31
    2.3.4 Statistics analysis .................................................. 33
    2.3.5 Baseline results of saliency models ......................... 34
  2.4 Summary ..................................................................... 34
3 Propagation method for salient object detection 37
  3.1 Introduction .................................................................. 37
  3.2 Related work .................................................................. 38
    3.2.1 Traditional approaches in saliency detection ............ 38
    3.2.2 Propagation for saliency detection ......................... 40
  3.3 Salient object detection via bi-directional propagation .... 41
    3.3.1 Method ................................................................. 41
    3.3.2 Experiments .......................................................... 46
4 Multi-scale CNNs with CRFs for salient object detection
4.1 Introduction
4.2 Related works
4.2.1 CNN-based saliency models
4.2.2 Disconnected CRF with CNNs
4.2.3 Joint CRF with CNNs
4.3 Multi-scale CNNs with disconnected Dense-CRFs
4.3.1 Formulation
4.3.2 Model training
4.3.3 Experiments
4.4 Multi-scale CNNs with joint CRFs
4.4.1 Formulation
4.4.2 Implementing mean-field iteration with neural networks
4.4.3 Model training
4.4.4 Experiments
4.5 Summary
5 Saliency integration
5.1 Introduction
5.2 Challenges
5.3 Arbitrator model
5.3.1 Bayesian integration framework
5.3.2 Cellular automaton
5.3.3 Reference generator
5.3.4 Model expertise estimator
5.4 Experiments
5.5 Summary
6 Summary
6.1 Contributions
6.2 Applications and extensions
6.3 Future work
References
Original publications
1 Introduction

1.1 Visual attention analysis

The concept of “attention” was firstly proposed in the cognitive psychology field by William James [1], who described attention as focalization and concentration of consciousness from the visual scene. Later, visual attention studies became vast over the past three decades in computer vision and image processing community.

Given the visual scenes, although human eyes can perceive a rich representation, our human visual system (HVS) may only focus on the regions of interest (ROIs) that are distinguished from the surroundings. Visual attention, in the computer vision field, is an integrated research topic by taking into account factors such as psychophysical cues and neurophysiological constraints, and thus to implement biologically-plausible mechanisms and architectures to address the ROIs on perception.

Eye movement is believed to be the most essential element to accurately reflect and interpret human visual attention [2]. Theoretically, when directing the gaze rapidly towards the objects on the scenes, the observers may fixate on some distinctive regions for a longer duration. This visual selection process shows how the ROIs catch the observer’s attention. Today, as the advance of eye tracking technology [3], it is able to detect accurate eye movements with economic eye tracking devices, which provides highly reliable biological ground truth to facilitate computational modeling for visual attention analysis.

Saliency refers to the state or quality of certain pixels or regions that stand out from their neighbors on the visual scenes. Saliency modeling, in computer vision, is the computational modeling process for visualizing human visual attention. Generally, saliency models are categorized into two types based on the factors that drive attention, known as bottom-up models (BU) and top-down models (TD). Bottom-up models [4] are scene-driven and feature-based, accounting for the low-level features derived solely from the visual scenes which are sufficient to differentiate the salient regions from the surroundings. On the other hand, top-down models [5] are expectation-driven and task-based, which are computed by cognitive factors such as prior knowledge, expectations, and goals. Usually, BU models are fast, reflexive and automatic [6], while TD models are voluntary or certainly cued ([7]). However, the features extracted from BU models can also be manipulated as cues in TD models.

Different from other topics such as object detection or recognition, saliency itself is not a well-defined term [8]. Most of the previous works build saliency models by
following one specific task towards saliency detection: fixation prediction (FP) or salient object detection (SOD).

Fixation prediction is accurately based on the eye movement data, as the participants are invited to view the given scenes while their eye fixations are recorded. Then, the saliency models are built to compute a probabilistic map of which the continuous saliency density values on the map predict the real human eye fixation patterns. Early bottom-up saliency models mostly focused on fixation prediction.

The concept of salient object detection, or salient object segmentation, is developed to depict saliency in object level and in a discrete domain. The salient object segmentation datasets are collected by asking the labelers to annotate the scenes by drawing pixel-accurate silhouettes of the objects that are believed to be salient. Consequently, a binary salient object mask is created for each image to represent a pixel as salient or not. And the objective of saliency models is to compute a saliency map that matches the annotated mask. In computer vision, salient object detection is usually interpreted as a process with two stages, including detecting the salient objects and segmenting the accurate regions of the objects.

Saliency models, either fixation prediction models or salient object detection models, contribute to a broad range of computer vision based researches in various fields. Fixation prediction models can provide a good understanding of human visual attention, which can largely assist works on advertisement design [9], scene understanding [10], human visual behavior analysis, and psychological implication understanding [11]. Salient object detection models can be a pre-processing technique for various computer vision tasks, such as image/video segmentation [12], image/video compression [13], image cropping [14], video summarization [15], image fusion [16], etc.

Saliency integration, or saliency aggregation, explores optimal approaches to unify saliency maps from multiple existing saliency models. Although many existing saliency models claim high performances, none of them can outperform the others on every single image under evaluation [17]. Thus, saliency integration is proposed to take the advantages of multiple saliency models and make up for the defects of any specific ones, for enhanced accuracy and robustness of saliency detection.

Generally, visual attention analysis is a broad research topic, of which the subsets include but not limited to eye movement, saliency modeling and integration, and attention-based applications. My thesis will briefly summarize these closely related topics and detail the outcomes in accordance with my Ph.D. research path.
1.2 Objectives

My research work in the Center for Machine Vision and Signal Analysis (CMVS) started when I did my internship during my master’s studies. At that time, I participated in the eye tracking data collection project by creating the demo and conducting the data collection experiments using eye tracking devices. The internship took me into the visual attention related topics, which later became the research topic during my Ph.D. studies.

At the start of my Ph.D. researches, I analyzed and summarized the works of eye movement data collection and published the eye tracking dataset with a benchmark on saliency fixation prediction models. The dataset was task-driven and provides images with a variety of semantic categories. After analyzing the eye movement dataset, I obtained a brief understanding of some subsets about visual attention analysis, such as eye movement analysis and saliency models. My special interest was in building computational models for visual attention analysis, which became the main area that I focused on for the following a few years of my research works.

Thus, my main objective for visual attention analysis is to build efficient computational models for saliency detection, especially for salient object detection. Firstly, I proposed an unsupervised model for salient object detection, based on a bi-directional propagation method. Then, as the prevalence of convolutional neural networks (CNNs), deep saliency models continuously received state-of-the-art performances. Thus, I started to work on salient object detection tasks using different structures of deep neural networks. By taking the advantage of multi-scale contextual information from CNNs, I proposed a saliency model with CNNs and multi-scale conditional random fields (CRFs), of which the multi-scale CRFs refine prediction maps from CNNs at multiple scales. The model is efficient and compact. However, the CRF layer is disconnected from the training of CNNs. Thus, I proceeded to explore a seamless integration of CRFs with CNNs, and thus proposed a cascade CRFs block that can be jointly trained with CNNs. The newly proposed CRFs are also able to jointly refine the continuous features and discrete prediction maps at each scale of the CNNs, and thus result in the state of the art performance.

Together with the researches on saliency models, I also aimed at integrating existing saliency models with a flexible and efficient approach to increase the general accuracy and robustness of saliency detection. Thus, I worked on saliency integration tasks and proposed a saliency integration model to appropriately combine saliency maps from multiple saliency models for a more accurate result. Saliency integration is a relatively
new topic and my work on it was published with the most comprehensive experimental analysis and results.

1.3 Summary of original articles

Five articles are included in the thesis on the main topic of visual attention analysis, of which Paper I concentrates on eye movement data analysis, Papers II-IV focus on computing saliency models, and Paper V is about saliency integration. Figure 1 summarizes the outline of the thesis.

Paper I focuses on eye movement data and establishes a new task-driven eye tracking dataset. Inspired by psychological findings that human visual attention is tightly dependent on the viewing tasks, we designed specific tasks per the contents of 111 images with various semantic categories. This work results in a dataset of 111 fixation density maps and over 5000 scanpaths, and provides baseline results of thirteen state-of-the-art saliency models.

Paper II-IV present three computational models for salient object detection. Paper II introduces a bottom-up saliency model by using a bi-directional propagation method, which results in the state-of-the-art performance among unsupervised saliency models. Paper III proposes a flexible CRF refinement framework by embedding the Dense-CRF inference to multiple levels of side outputs from CNNs for multi-scale saliency refinement. The model is efficient in training and receives comparable performances to the state of the arts with much simpler network architectures. Paper IV is a collaborated work that proposes a novel cascade CRFs architecture with CNN to jointly refine deep

![Fig. 1. Summary of original articles.](image-url)
features and predictions at each scale for a refined saliency map, which results in state-of-the-art performance among deep saliency models.

Paper V explores saliency integration tasks and proposes an arbitrator model for saliency integration that substantially outperforms the state-of-the-art integration methods. Most importantly, the work presents the most comprehensive experimental analysis with various combinations based on a pool of twenty-seven candidate saliency models, covering both traditional and deep models over four datasets.

1.4 Organization of the thesis

The thesis is organized in six chapters, per my research path on visual attention as well as a gradual process in understanding the topic.

Chapter 1 briefly introduces the background of visual attention analysis, and also indicates the aims and objectives of my Ph.D. studies, the contents of the five original articles, and the outline of the thesis.

Chapter 2 introduces one of the subset topics on eye movement data and presents my research outcomes on a task-driven eye tracking dataset in Paper I.

Chapter 3 focuses on the subset topic about computational modeling for salient object detection. In this chapter, I will introduce the topic by illustrating three waves of developing saliency models, from early bottom-up unsupervised models to recent deep saliency models. Specifically, unsupervised methods will be discussed and I will present the bi-directional propagation method for saliency detection proposed by Paper II.

Chapter 4 focuses on deep saliency models. Paper III and Paper IV illustrating two CNN based saliency models are introduced, which incorporate CNN with disconnected Dense-CRF and joint CRF respectively for saliency detection.

Chapter 5 gives an introduction to the subset topic of saliency integration, and the contents will mainly follow the research outcomes from Paper V.

Chapter 6 summarizes the contributions of all the works and discuss the potential applications of saliency models and possible future works.
2 Saliency data collection

2.1 Introduction

Eye movement is fundamental in revealing visual attention. The topic of eye movement has been explored in the medical field since 1868 by a French ophthalmologist [18]. He observed that human eyes do not move continuously. Instead, they produce short rapid movements (saccades) and short stops (fixations) on visual scenes.

Early eye movement studies efficiently facilitate diagnosis and treatment in the medical and biological fields. Later, as the progress of eye tracking devices largely improves the accuracy and efficiency in eye movement data collection, it brought further applications in computer science related studies.

Firstly, eye movement data can be a valuable modality for human computer interaction (HCI), to improve the usability and user experience of a system. For instance, the wireless eye movement controlling systems can assist handicapped users to interact with electrical devices [19] or to facilitate gaming control [20]. In the pattern recognition field, [21] took advantage of eye movement data in recognizing human activities (i.e. copy, read, write, browse).

In the 1960s, [22] conducted the well-known experiments with viewing tasks to explore the relationship between eye movements and human visual attention. In his studies, it is shown that the perception of objects by human eyes can carry essential and useful contextual information on the scenes. Based on this finding, image processing, and computer vision based studies mainly focus on interpreting eye movement data in terms of detecting the regions of interest of the human visual system, and thus to facilitate the development of computational models for visual attention analysis.

2.1.1 Eye movements

Benefiting from recent eye tracking techniques, eye movement data can be collected for further visual attention analysis. Typically, two modalities of eye movement data can be captured, that are saccades or fixations [23]. Specifically, fixations indicate the positions where the human eyes fixate for a long enough duration, while saccades refer to fast eye movement shifts.

The eye movement data for further analysis are translated from the raw data from eye trackers [23], especially fixations. The fixations are collected by removing raw saccade points and collapsing raw fixation points into one tuple for representation, and
thus to reduce the complexity of the eye movement protocol. The reasons to reduce the complexity are two folds. Firstly, saccades [24] are too fast and thus may deliver little or even no visual processing. Secondly, eye movements such as tremors, drifts, and flicks are irrelevant factors for visual attention analysis [25]. Therefore, fixations are regarded as a convenient data type of minimizing the complexity of raw eye movement data while retaining the most necessary characteristics for visual attention analysis.

In visual attention related studies, eye fixations show the regions where the observers draw attention. A fixation map can be computed by aggregating fixation data from multiple observers when viewing the same scene (Figure 2-(b)). Further, a fixation density map is convolved by a Gaussian over the fixation map, and thus to create a continuous heat map showing the regions of interest of the human visual system (Figure 2-(d)). A sequence of fixations and the saccades between fixations formulate a scanpath, which visualizes the sequence of ROIs and provides an insight into individual visual behavior with temporal information (Figure 2-(c)).
2.1.2 Eye trackers

Various techniques have been applied to detect human eye movements. [26] categorize existing eye tracking techniques into three types by different detecting techniques, including (1) videooculography (VOG), video based eye trackers using head-mounted or remote visible light video cameras, (2) video based infrared (IR) pupil-corneal reflection (PCR), and (3) Electrooculography (EOG). The first two video-based eye tracking techniques remain many similar properties.

Based on different research topics towards computer vision, current eye trackers differ in two types, the optical trackers and the electrooculography (EOG) trackers. The optical trackers are video-based non-contact devices with specific optical sensors, and the watching materials are usually images or videos on screens. Thus, optical trackers are widely used for studies such as visual attention analysis on images or videos [27]. Electrooculography trackers are wearable eye trackers with electrodes around human eyes to detect EOG signals and thus are adopted to visual behavior analysis based on daily activities [28].

For visual attention analysis on visual scenes, optical eye trackers are essential devices in data collection, which forms a basis for evaluating the performance of computational models and algorithms. In the data collection process, researchers obtain eye movement data by conducting eye tracking experiments with multiple participants as viewers. The prototypical procedure is to ask the participants to view images (or watching materials), either in a free manner or with viewing tasks, and at the same time to collect their eye fixations with optical eye trackers.

Typically, each image is viewed by several participants and the fixations from all the viewers are aggregated as the fixation map. Then, the fixation map is convolved by Gaussian to obtain the fixation density map. This fixation density map reflects the intensity of visual attention from viewers, and thus is regarded as the ground truth for computational models for saliency detection.

Figure 3 presents a typical experimental environment for eye movement data collection. In the experiment, a display (Figure 3.A) shows the watching materials to viewers (Figure 3.B). MyGaze eye tracker ([29]) (Figure 3.C) is located at the lower center place to the display, which is utilized for collecting eye tracking data, as it is an optical real time eye tracking system with low latency as well as high performance. A web camera (Figure 3.D) is optional to record the videos of eye movements from the faces of the observers.
2.2 Related work

Optical eye tracker based datasets significantly facilitate evaluating computational models for visual attention analysis. More specifically, a broad range of eye tracking datasets is collected for measuring saliency models.

2.2.1 Free viewing datasets

Most of the existing eye tracking datasets are collected with participants observing the images in a free viewing manner. Early eye tracking datasets are built with a collection of images without specific image semantics [30]. Then, it has been addressed that images with certain semantics can facilitate visual attention analysis in specific fields. Thus, a few semantic-driven datasets are established.

Le Meur dataset [31] is the earliest publicly available eye tracking dataset for building computational approach to model visual attention. The eye movement data are collected by involving 40 observers viewing 46 images in a free viewing manner. The watching materials are color images of natural scenes. Together with the selected images, fixation density maps and scanpaths are computed for modeling visual attention.

MIT 2009 dataset [30] contains 1003 images from 15 viewers. The eye tracking experiment is still in a free viewing mode. However, more image semantics are included such as landscape scenes and portraits. Fixation data along with fixation density maps are provided as the ground truth for building saliency models.

NUSEF dataset [32] is a dataset of 758 images from 75 participants. Each image is viewed by 25 participants on average in a free-viewing mode. The NUSEF dataset is the first eye tracking dataset that addresses the importance of image semantics. The
watching materials contain various semantic categories such as portraits, nudes, faces, emotions, and actions. Thus the fixation data are also influenced by image semantics.

**Eye crowd dataset** [33] is a free viewing dataset in order to provide eye movement data to detect saliency in the crowd. The 500 watching materials contain varying crowd densities and are viewed by 16 observers. In addition to fixations, crowd faces are also labeled in each image with two annotations, that are pose and partial occlusion.

**FiWI dataset** [34] is a specific eye tracking dataset for webpage saliency modeling. The watching materials are 149 webpages including pictorial webpages, text webpages, and mixed webpages. Pictorial webpages are webpages with one dominant image or several recognizable images. Text webpages contain intensive and informative texts. And mixed webpages are a combination of pictures and texts. The webpages are viewed by 16 participants without performing viewing tasks. Fixation density maps are publicly provided.

**CAT 2000 dataset** [35] contains 4000 images from 20 different categories with 18 participants. The dataset is divided into two sets, including the training set and the testing set. Fixation density maps are provided for modeling image saliency models.

### 2.2.2 Task driven datasets

As suggested by previous psychological studies, human viewing behavior is tightly dependent on viewing tasks [36]. First, visual searching tasks lead to faster fixations and reduce distractions [37]. Second, task-driven datasets can provide more targeted data for computational models for visual attention analysis. Thus, a few task-driven eye tracking datasets are established.

**Ehinger dataset** [38] is a task-driven dataset from 14 observers performing person detection tasks on 912 images, of which half images contain pedestrians while the rest do not include any targets. The fixation data, fixation density maps are provided to facilitate salient human detection.

**POET dataset** [37] is a task-driven eye tracking dataset containing 6270 images selected from PASCAL VOC 2012 [39]. The images include ten object categories, *i.e.* cat, dog, bicycle, motorbike, boat, plane, horse, cow, sofa, and dining table. The participants are required to perform visual classification tasks.

Previous eye tracking datasets for saliency analysis are mainly in two types: free viewing datasets with natural or specific semantic categories and task-driven datasets with simple semantic categories. Firstly, as task-driven eye tracking datasets lead to faster fixations and may reduce center bias, designing visual searching tasks is essential and needs to be further explored in data collection. Secondly, various semantic
categories may assist further understanding on other computer vision related topics. Thus, we establish a task-driven eye tracking dataset with watch materials containing various semantic categories for visual attention analysis.

2.3 Task-driven eye tracking dataset with multiple semantics

We establish the Oulu task-driven eye tracking dataset in order to further facilitate visual attention analysis, as in Paper I. Firstly, most of the watching materials in the new dataset are designed with specific visual searching tasks, making in-depth evaluation on human visual behaviors possible. Secondly, the image semantics cover several computer vision related topics instead of simple object categories, including text, facial expression, texture, pose, and gaze. Thirdly, the newly established dataset involves more participants (up to 47) and higher image resolution (1920×1200 px) as watching materials, which leads to more accurate fixation predictions.

2.3.1 Task specific watching materials

The watching materials are collected from Google and Flickr, and are processed and combined into 111 images, covering various semantic categories. Among the total 111 images, 102 images are designed with a viewing task based on the semantic implications and the rest are free viewing images. The tasks, inspired by currently active computer vision topics (e.g., micro/normal facial expression recognition and pedestrian detection), are designed according to the image semantics. For instance, in Figure 4, the first image of the first row belongs to face images and the corresponding task is to find out a shocking face; the fourth image belongs to vehicle/pedestrian category, and the task is to detect pedestrians. Table 1 indicates the detailed semantic categories of the watching materials and the corresponding viewing tasks.

2.3.2 Experimental environment

47 participants aging from 18 to 40 are involved in data collection, who are students, staffs and researchers in University of Oulu. Environmental conditions such as glasses, eye colors, and contact lenses are not limited. The myGaze, an optical eye tracker, is utilized for eye movement data collection. The collection process is as shown in Figure 3. First, the eye tracker (Figure 3-C) is calibrated by a five-point algorithm. Second, the screen displays the assigned task. Then, the watching material is shown for a certain duration according to the complexity of the task. Each participant (Figure 3-B)
Table 1. Image semantics categories and the corresponding viewing tasks in the Oulu task-driven eye tracking dataset. Reprinted by permission, Paper I © Springer.

<table>
<thead>
<tr>
<th>Number</th>
<th>Semantics</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Text/map</td>
<td>Read texts; watch maps.</td>
</tr>
<tr>
<td></td>
<td>Normal/micro expressions</td>
<td>Distinguish facial expressions.</td>
</tr>
<tr>
<td></td>
<td>Face images</td>
<td>Compare similar faces; distinguish gender/ethnicities/celebrities/aging/real faces.</td>
</tr>
<tr>
<td></td>
<td>Pose/gaze</td>
<td>Distinguish pose/gaze/directions.</td>
</tr>
<tr>
<td></td>
<td>Vehicle/pedestrian</td>
<td>Find out vehicles/pedestrians.</td>
</tr>
<tr>
<td></td>
<td>Motion/blurring/special lighting</td>
<td>Observe blur/lighting/motion effects.</td>
</tr>
<tr>
<td></td>
<td>Periodity/symmetry/material/mosaic images</td>
<td>Observe periodity/symmetry/materials; find out main objects and distinguish main categories in small patches in mosaic images.</td>
</tr>
<tr>
<td></td>
<td>Landscape/scenes</td>
<td>Watch landscape images; spot differences between similar scenes.</td>
</tr>
<tr>
<td></td>
<td>Others</td>
<td>Free view.</td>
</tr>
</tbody>
</table>

is asked to sit in front of the display (Figure 3-A) to watch the images and to finish the tasks by only watching the images. The eye tracker records the eye movements and the Logitech C930 web camera (Figure 3-D) captures the videos of observers’ eye movements for future studies.

2.3.3 Ground truth

The collected eye tracking dataset is focused on visual saliency analysis and thus provides visual saliency data for evaluating computational salient fixation models.

The ground truth fixation density map are computed by aggregating the fixations from all the participants when viewing the same image and then being convolved with the Gaussian Mixture Model [40] with 10 components. During the computation of aggregation maps, only fixations within an accuracy of 1° visual angle is considered. Figure 4 presents some examples of the fixation density maps provided in the dataset.

In addition, the scanpaths are generated to indicate how the fixations shift when each participant views each image. Figure 5 shows an example of the last frame of the scanpath animation when a participant is required to distinguish the real face from the synthetic one.
Fig. 4. Visualizations of fixations. The first row shows examples of watching materials; the second row illustrates the corresponding fixation density maps; the third row presents color images with fixations density maps as masks. Preprinted by permission, Paper I © Springer.

Fig. 5. Example of a scanpath. The circles are fixation positions and the length of radius correlates to the duration between fixations. The line shows the direction of fixation shifts. The first five fixations are marked in this image. Preprinted by permission, Paper I © Springer.
2.3.4 Statistics analysis

Compared to existing datasets, the newly established eye tracking dataset shows a much weaker center bias because of the visual tasks. As shown in Figure 6-a, by aggregating all the fixations of all the images from all the observers, it is shown that 12% of the fixations fall in the top 20% center region, while 60% of the fixations in the top 50% center region, which results in much lower center bias than the statistics given by previous findings [30, 33]. Thus, the newly established dataset is only slightly center biased.

As well, correlation analysis is investigated to evaluate the variation of fixations among different participants for each image with the leave-one-subject-out strategy. For each participant, a fixation density map is computed without using the fixation data collected from that participant (leave-one-subject-out fixation density map). The ground truth fixation density map is based on the fixations from that participant. Then, the Correlation Coefficients (CC) between the leave-one-subject-out fixation density maps \( F \) and the ground truth fixation density maps \( G \) are calculated for all subjects. The CC is as follow:

\[
CC = \frac{\sum_{i,j}(F(i,j) - \bar{F})(G(i,j) - \bar{G})}{\sqrt{\sum_{i,j}(F(i,j) - \bar{F})^2} \sqrt{\sum_{i,j}(G(i,j) - \bar{G})^2}}
\]  

(1)

where \( F(i,j) \) and \( G(i,j) \) are the pixels on location \( (i, j) \) of \( F \) and \( G \) respectively, \( \bar{F} \) and \( \bar{G} \) are mean values of \( F \) and \( G \).
Figure 6-b illustrates the histogram of the average CC for each image. It can be observed an average CC for all the images as 0.97. Apparently, the fixations are highly consistent across participants. Thus, the computed fixation density maps appropriately represent the ROI of the majority of the participants. Additionally, as in Figure 6-c, 47% of the fixations from the left-out participant are within the top 5% salient regions on the leave-one-subject-out fixation density map, while over 90% in the top 30% salient regions. Therefore, the statistics analysis indicates that the fixation density maps from the newly established task-driven eye tracking dataset are highly consistent with the visual attention of the majority of the participants.

### 2.3.5 Baseline results of saliency models


Two evaluation metrics are adopted to measure the accuracy of saliency models: (1) Correlation Coefficient, (2) Area Under the ROC Curve (AUC) [52]: the saliency map is thresholded with positive and negative samples. By plotting the true positive vs. false positive line, a ROC curve is computed and the underneath area is AUC. Table 2 presents the performances of the evaluated saliency models on the newly task-driven eye tracking dataset and the MIT 2009 dataset [30].

### 2.4 Summary

This chapter presents a thorough introduction of the data preparation process for computational models for visual attention analysis. Firstly, the basic concepts of eye movements are introduced, especially the eye fixations which are the fundamental elements in computational models for visual attention analysis.
### Table 2. Mean and standard deviations of CC and AUC of thirteen saliency models. Preprinted by permission, Paper I © Springer.

<table>
<thead>
<tr>
<th>Model</th>
<th>Our Dataset CC</th>
<th>MIT Dataset CC</th>
<th>Our Dataset AUC</th>
<th>MIT Dataset AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>0.34 ± 0.22</td>
<td>_</td>
<td>0.68 ± 0.11</td>
<td>_</td>
</tr>
<tr>
<td>ProtoObj</td>
<td>0.05 ± 0.15</td>
<td>0.14</td>
<td>0.50 ± 0.04</td>
<td>0.54</td>
</tr>
<tr>
<td>GBVS</td>
<td>0.47 ± 0.17</td>
<td>0.48</td>
<td>0.74 ± 0.09</td>
<td>0.80</td>
</tr>
<tr>
<td>IT</td>
<td>0.36 ± 0.21</td>
<td>0.37</td>
<td>0.69 ± 0.10</td>
<td>0.74</td>
</tr>
<tr>
<td>FT</td>
<td>0.05 ± 0.30</td>
<td>0.04</td>
<td>0.49 ± 0.13</td>
<td>0.52</td>
</tr>
<tr>
<td>JUD</td>
<td>0.44 ± 0.16</td>
<td>0.47</td>
<td>0.75 ± 0.08</td>
<td>0.80</td>
</tr>
<tr>
<td>CovSal</td>
<td>0.39 ± 0.21</td>
<td>0.45</td>
<td>0.72 ± 0.09</td>
<td>0.67</td>
</tr>
<tr>
<td>SWD</td>
<td>0.50 ± 0.19</td>
<td>0.49</td>
<td>0.76 ± 0.09</td>
<td>0.80</td>
</tr>
<tr>
<td>SIM</td>
<td>0.41 ± 0.16</td>
<td>0.43</td>
<td>0.71 ± 0.08</td>
<td>0.76</td>
</tr>
<tr>
<td>VIK</td>
<td>0.30 ± 0.19</td>
<td>0.38</td>
<td>0.67 ± 0.07</td>
<td>0.74</td>
</tr>
<tr>
<td>FittedECSF</td>
<td>0.28 ± 0.19</td>
<td>0.27</td>
<td>0.65 ± 0.09</td>
<td>0.66</td>
</tr>
<tr>
<td>SUN</td>
<td>0.30 ± 0.17</td>
<td>0.25</td>
<td>0.68 ± 0.09</td>
<td>0.66</td>
</tr>
<tr>
<td>RC</td>
<td>0.33 ± 0.16</td>
<td>0.47</td>
<td>0.68 ± 0.09</td>
<td>0.78</td>
</tr>
</tbody>
</table>

Eye trackers are the primary tools for human fixation detection. Although both optical eye trackers and EOG based eye trackers can be used for eye movement detection, optical eye trackers are the main tools for data collection to facilitate fixation prediction. Accordingly, several eye tracking datasets are built for evaluating fixation prediction models. Since most of the existing datasets are recorded in a free viewing manner that results in high center bias and slow concentration, a new Oulu task-driven eye tracking dataset is established by designing specific visual tasks for images with various semantic meanings. The newly established dataset results in a dataset of 111 fixation density maps together with the original images and over 5,000 scanpaths for human visual behaviour analysis. Moreover, the baseline results of thirteen saliency models are provided.

The Oulu task-driven eye tracking dataset is established almost four years ago. After this eye tracking data collection work, I proceeded to investigate computational models for salient object detection. With sufficient experiences in saliency detection, I would like to look back my primary work and re-address some merits and limitations of the Oulu task-driven eye tracking dataset.

The limitations are two folds. Firstly, the established dataset is very difficult for fixation prediction approaches since the designed tasks increase the difficulty of the images. Second, although there are rich semantics, the number of images for each semantics category is limited and thus making supervised learning for semantic specific approaches unrealized.

However, the merits of the datasets also come from multiple semantic categories and specific tasks. Firstly, it offers in-depth inspections on how tasks guide visual attention,
and thus can inspire effective designs of semantic specific eye tracking datasets. For instance, observers tend to focus on whole facial regions when recognizing facial expressions, but mainly focus on eyes when distinguishing gaze directions. We also designed a pair of images captured from the same scene but with different camera focuses. The scanpaths indicate that observers tend to fixate prior to clear objects in an image. The above phenomenon indicates that both tasks and image contents are significant factors influencing human visual behavior. By the same token, we suppose that with appropriate image semantics and tasks, it is possible to guide observers fixate on desired regions to facilitate salient detection in specific fields. For instance, with facial expression recognition tasks, an eye tracking dataset can be collected that indicates significant action units for different facial expressions and thus to further facilitate facial expression analysis.

Meanwhile, we recorded rich eye movement data during experiments. Unlike most existing datasets that only provide fixation data, the established dataset also provides an opportunity of exploring the intrinsic relation between tasks and human visual behavior. Moreover, in addition to the obtained fixation density maps and scanpaths, we also recorded the videos of the reactions of the participants when they are doing the viewing tasks for further studies. Thus, the dataset can be further explored for research purposes.

Moreover, data collection with eye trackers always accompanies with strict experimental environments such as light and calibrations, and the collecting process can be rather time consuming. Thus, in future eye tracking data collection works, new data collection techniques can be further considered. For instance, Salicon [53] is a crowd sourcing dataset containing 10,000 images. The fixations are approximated by mouse-contingent and multi-resolutional paradigm based on neurophysiological and psychophysical findings and thus are collected with general-purpose mouses instead of an eye tracker. Similar methods can be further investigated to make large-scale data collection possible.
3 Propagation method for salient object detection

3.1 Introduction

The history of computational models for visual attention analysis is as long as thirty years. Based on the theories and methodologies applied to model visual attention, existing saliency approaches can be divided into three stages.

At its earliest stage, saliency approaches focus on fixation prediction that results in fixation density maps. Primary investigations on saliency modeling are bottom-up models that involve low-level image features based on knowledge from cognitive psychology and computer vision. For instance, [10] are believed to be the pioneers in modeling visual saliency, and they extract image features such as color, intensity and local orientations to distinguish saliency. Afterward, the topic of saliency modeling attracts more attention and explorations on top-down methods produce more accurate results by involving high level features such as faces and people descriptors [30].

In the second stage, saliency models are extended to object level, known as salient object detection, which catches more attention. The salient object detection models aim at detecting and segmenting out the salient objects from the scenes, which is an integration of fixation prediction and semantic segmentation. The charm of salient object detection is that it stimulates a variety of hypotheses and assumptions about saliency based on psychological theories and image composition properties. For example, [54] suggest that the center-bias prior can assist salient object detection as most of the salient objects usually locate in the center areas on the scenes. [55] propose the background prior (or boundary prior) based on the basic rule of photographic composition, indicating that the features of the salient objects are apparently very different from the features of the background regions. [56] introduce the dark-channel prior based on the observation that there exist pixels with low intensities in one of the RGB channels for the salient regions on the scenes. Further, there are also salient object assumptions based on contour prior [57], depth prior [58], etc. In practice, salient object detection may have a broad range of potential computer vision applications, such as segmentation [59], image cropping [60], image fusion [61], image classification [62], video compression [63], etc.

During the last five years, as the prevalence of the convolutional neural networks (CNNs), CNN-based approaches are boosted and significantly improve the performance
of saliency models. Different from traditional saliency models, CNN-based methods get rid of extracting conventional hand-crafted features and rely on high dimensional deep features for state-of-the-art performances [64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74].

My research works on saliency models start with salient object detection with traditional approaches. Thus, this chapter mainly focuses on unsupervised methods for object level saliency detection. Section 3.2 reviews previous saliency detection approaches. In Section 3.3, the work in Paper II is presented which proposes a bi-directional propagation method for salient object detection. Section 3.4 summarizes this chapter.

3.2 Related work

3.2.1 Traditional approaches in saliency detection

Traditional methods for modeling visual saliency can be categorized in different ways. In this section, we group the saliency approaches based on the mathematical and recognition methods based on [75], including cognition-based models, information-based models, frequency-based models, Bayes-based models, decision-based models, learning-based models, and graph-based models.

Most of the traditional models are more or less based on cognitive concepts. For instance, to predict saliency, many early saliency models perform center-surround feature comparisons to discriminate the regions or objects that stand out from the neighborhood. [10] extract three features, including color, intensity, and orientation, and compute center-surround feature comparison and then conspicuous map fusion with the Gaussian pyramid. [51] further extend the approach to attention-based object detection. [31] then extend feature extraction into Human Visual System (HVS) and compute contrast sensitivity functions and center-surrounding interactions for more efficient visual feature comparison. [76] propose symmetric operators and perform multi-scale symmetry saliency detection. [44] take advantage of color appearance and adopt the wavelet decomposition method for center-surround feature comparisons.

Some saliency models are computed based on information theory, which predicts the salient regions or objects as the most informative parts of the scenes. [77] propose an information maximization approach to compute the salience likelihood for saliency detection. [11] introduce the Incremental Coding Length (ICL) to estimate the entropy gain of the features for saliency detection. [78] further introduce conditional entropy and predict the salient regions as the uncertain locations with minimum conditional entropy.
[79] also extend to simulate human saccadic scanpaths based on three multi-band filtering response maps.

The frequency domain can also be adopted for saliency detection. [45] propose a spectral residual model that analyses the log-spectrum of the scene for saliency detection. [80] introduce the phase spectrum and make saliency detection based on the amplitude transform. [81] propose a spectral whitening model that transforms the input image to the Fourier domain and computes the saliency as the inverse Fourier transform of the whitened signals. [46] also introduce a frequency-tuned method for saliency detection with low-level features of color and luminance.

Bayesian models for saliency detection is a combination of sensory evidence with the prior constraints. Thus, prior knowledge based on scene contexts and sensory information such as image features are combined according to probability theory to compute the saliency maps. [82] propose a Bayesian framework to adopt statistical regularities to estimate the likelihood of a salient object in a given scene. [83] define the surprising stimuli and formulate a Bayesian framework by computing the KL divergence between the posterior and the prior beliefs. [48] propose the SUN model that forms a Bayesian framework combining both bottom-up and top-down features for saliency detection on natural scenes.

Decision theoretic models are based on perception mechanisms that evolve to produce the optimal decisions about the states of the units on the scenes, of which the decision is usually based on optimal strategies such as the minimum probability of error. [84] indicate that salient features yield to a class of interest from other visual classes, and thus define top-down visual attention as a classification issue with the minimal expected error. [85] generate an activation map by extracting visual features and detecting salient objects on the scenes. ROIs are predicted based on an adaptable retinal filter using an iterative adjustment algorithm.

Learning-based models usually integrate both bottom-up and top-down image features and learn a classifier to predict saliency on the scenes. [83] design a classification framework based on Bayesian theory and train the model to predict saliency. [86] adopt the support vector machine (SVM) to train a saliency classifier based on 169-dimensional feature vectors. [87] investigate a nonlinear learning framework using AdaBoost for optimally fusing conspicuity maps for saliency detection. [33] adopt multi-kernel learning for saliency detection.

Graph-based models regard the units on the scene as the nodes of a graph and model structural interaction among them for saliency detection. The connections between nodes can be represented by graphical models or neural networks. [43] propose to regard the input image as a fully connected graph and use the Markov approach for
saliency detection. [88] and [89] both adopt random walks on graph for salient object detection. [90] utilize a graphical model and adopt Dynamic Bayesian Networks (DBN) for saliency representation. [91] propose to model Conditional Random Field (CRF) formulation and a visual dictionary to predict saliency.

Although recent salient object detection models adopt different strategies in computing saliency maps, they tend to further refine the computed coarse saliency maps with graph-based approaches [92, 93, 94, 95, 96, 57]. The graph-based refinement takes advantage of feature relationship between one unit on the scene and the other units in its neighborhood to reconcile its saliency value, and thus produces fine-grained saliency maps. Therefore, we explore graph-based methods for salient object detection.

### 3.2.2 Propagation for saliency detection

Propagation is a prevalent bottom-up methodology for graph-based models, which has been widely employed in recent years. The input image is over-segmented into superpixels and is constructed as an undirected graph, which comprises of vertices of the superpixels together with edges representing the relationship between adjacent vertices. Then, the propagation seeds, selected from a coarse saliency map [92, 93, 94, 95], are spatially diffused iteratively to the whole graph to refine the coarse saliency map.

Traditional evolution schemes involve all the superpixels on the image into each iteration. However, [96] suggest that it is not necessary that all the superpixels participate in the propagation in each iteration. Thus, they propose a teach-to-learn and learn-to-teach scheme that measures the difficulty of each unlabeled superpixel with the knowledge of the labeled set and in each iteration propagates those “simple” ones that are easy to judge. The “propagation from simple to difficult” strategy largely optimizes the propagation results through manipulating the propagation sequence.

There are two types of propagation: foreground propagation and background propagation. As in Figure 7, foreground propagation (Figure 7.1) chooses the most salient values from the coarse saliency map as foreground seeds, which is a direct approach to discriminate the salient regions on the given image. The propagation performance is highly dependent on the quality of seeds selection. Background propagation (Figure 7.2) chooses background seeds by background assumptions to propagate an unsalient map. Generally, background seeds are easier in the selection and can better distinguish the unsalient regions on the image. However, background propagation cannot judge the distinctness within salient regions.

Existing propagation methods are only based on foreground propagation [92, 93, 94], or by firstly background propagation and then foreground propagation [96, 95]. However,
the propagation is always single directional. Different from previous methods, we explore to perform foreground propagation and background propagation simultaneously in one evolution. The motivations are two folds: 1. to improve the propagation quality; 2. to enhance the propagation efficiency. After inferences and validations, we propose a bi-directional propagation (BIP) model for salient object detection in Paper II.

3.3 Salient object detection via bi-directional propagation

In this section, we will first introduce the method of the proposed bi-directional propagation (BIP) model. Then experimental results together with prevalent salient object detection datasets and evaluation metrics will be detailed. This work is originally published in Paper II.

3.3.1 Method

The bi-directional propagation (BIP) model efficiently performs both foreground propagation and background propagation in one iteration (Figure 7.3) and manipulates the propagation sequence with a difficulty-based rule. More specifically, in each iteration, the propagation scheme only chooses the relatively simple superpixels instead of all for either foreground propagation or background propagation, by measuring the difficulty of the unlabeled superpixels to the labeled foreground set and the labeled background set respectively. The framework of the proposed BIP model is illustrated in Figure 8.
Fig. 8. Framework of the proposed BIP model. Given an input image, both foreground seeds and background seeds are chosen as two initial labeled sets. In each iteration, the unlabeled superpixels are evaluated according to their difficulties to the labeled foreground set and the labeled background set respectively, only those with the lowest difficulties to each labeled set are selected and are accordingly spread to refine the foreground set or the background set. After all the unlabeled superpixels are labeled, the results from foreground propagation and background propagation are combined as the final saliency map. Reproduced by permission, Paper II © Elsevier.

**Foreground seeds and background seeds**

Foreground seeds for propagation are chosen from the computed coarse foreground map $S_F$, while background seeds are selected from the computed coarse background map $S_B$.

The coarse foreground map $S_F$ is based on the boundary prior, by assuming that the more discrepant a superpixel is from the boundary ones, the higher saliency values the superpixel possesses. Thus, the superpixels along the image boundaries are selected as background seeds, and are grouped into $K$ clusters by K-means algorithm. The number of superpixels belonging to the $k$-th cluster is denoted as $N_k$, $k = 1, \ldots, K$. If the $n$-th superpixel is still quite different from its most similar cluster, it is more likely to be salient. Thus, the coarse foreground map $S_F$ is as:

$$S_F(n) = \min_{k \in \{1, \ldots, K\}} \left( \frac{1}{N_k} \sum_{m=1}^{N_k} \| \varphi_n - \varphi_m \|_2^2 \right),$$

where $\| \varphi_n - \varphi_m \|_2^2$ computes the Euclidean distance between the $n$-th superpixel and the $m$-th superpixel on CIELab features.

The coarse background map $S_B$ is also computed based on the boundary prior with a basic propagation method. The over-segmented image can be regarded as an undirected graph $G = (V, E)$, which comprises a set $V$ of the superpixels together with a set $E$ of edges representing the similarity between adjacent superpixels. The constructed graph
$G$ can be described as an adjacent matrix $W = [w_{nm}]_{N \times N}$. The similarity between two superpixels is as follow:

$$w_{nm} = \exp\left(-\frac{\|\mu_n, \mu_m\|^2}{2\theta^2}\right),$$  \hspace{1cm} (3)

where $\|\mu_n, \mu_m\|^2$ computes the Euclidean distance between superpixel $\mu_n$ and $\mu_m$ on CIELab-XY features, where $\mu_n = [\phi_n^T, x_n, y_n]^T$, $x_n$ and $y_n$ are the coordinates of the $n$-th superpixel in X-Y space.

The superpixels along the image boundaries are used as background seeds and the propagation function is as follow:

$$S^{t+1} = I \cdot D^{-1} \cdot W \cdot S^t,$$  \hspace{1cm} (4)

where $I$ is the identity matrix and $D$ is the diagonal degree matrix with $D_{nm} = \sum_m w_{nm}$, and the initial $S^0$ is computed based on the boundary prior as follow:

$$S^0(n) = \begin{cases} 1, & \text{the } n\text{-th superpixel is a boundary one} \\ 0, & \text{otherwise}. \end{cases}$$  \hspace{1cm} (5)

After $T_1$ times of iterations, the final propagated $S^{T_1}$ is computed as the coarse background map $S_B$. Then a coarse saliency map $S_{\text{ coarse}}$ is computed as follow:

$$S_{\text{ coarse}} = \frac{S_F}{S_B + \alpha},$$  \hspace{1cm} (6)

where $\alpha$ is set as 0.001 to avoid the division-by-zero problem.

Finally, $S_{\text{ coarse}}$ is thresholded by $\gamma$ to obtain the foreground seeds and the superpixels on the four boundaries of the image are background seeds.

**Bi-directional propagation**

The Bi-directional propagation scheme is applied to spread the labeled foreground superpixels and the labeled background superpixels to the unlabeled ones with the difficulty-based rule.

At the $t$-th iteration, the unlabeled set are denoted as $\mathcal{U}^t$ and the labeled set as $\mathcal{L}^t$. $\mathcal{L}^t = \mathcal{L}_F^t \cup \mathcal{L}_B^t$, where $\mathcal{L}_F^t$ refers to the set labeled by foreground propagation, while $\mathcal{L}_B^t$ is propagated by background seeds. Every superpixel on the image has two measures, that is its saliency value $f_n^t$ for foreground propagation and its unsaliency value $b_n^t$ for background propagation. At time $t = 0$, $\mathcal{L}_F^0$ (or $\mathcal{L}_B^0$) consists of foreground seeds (or background seeds) with saliency (or unsaliency) values $f_n^0 = 1$ (or $b_n^0 = 1$).

43
At time $t$, the superpixels are divided into three subsets $\mathcal{P}_F^t$, $\mathcal{P}_B^t$ and $\mathcal{P}_D^t$, according to how difficult to assign a superpixel to $\mathcal{L}_F^t$ and $\mathcal{L}_B^t$ respectively. $\mathcal{P}_F^t$ (or $\mathcal{P}_B^t$) consists of the superpixels with the lowest difficulties to $\mathcal{L}_F^t$ (or $\mathcal{L}_B^t$) and will be propagated by $\mathcal{L}_F^t$ (or $\mathcal{L}_B^t$) at time $t$. For the superpixels in $\mathcal{P}_D^t$, they are regarded as ambiguous ones that will not be involved in the $t$-th iteration of propagation. The unlabeled set $\mathcal{U}$ are iteratively labeled by this rule until $\mathcal{U}$ is completely labeled.

A set of $\mathcal{P}_F^t$ (or $\mathcal{P}_B^t$) from those superpixels $\mathcal{C}_F^t$ (or $\mathcal{C}_B^t$) is chosen that are directly connected to the labeled set $\mathcal{L}_F^t$ (or $\mathcal{L}_B^t$) on the undirected graph $G$. If the $n$-th superpixel is under consideration, its difficulty to the labeled set $\mathcal{L}_F^t$ (or $\mathcal{L}_B^t$) is $d_n^F$ (or $d_n^B$). $\mathcal{P}_F^t$ and $\mathcal{P}_B^t$ are two sets of superpixels with the lowest difficulties selected from $\mathcal{C}_F^t$ and $\mathcal{C}_B^t$ respectively.

To measure the difficulty that an unlabeled superpixel compared to a labeled set, we consider two aspects: distinctness to its neighborhood and connectivity to the labeled set. Distinctness computes the appearance difference between the unlabeled superpixel and its neighbors, while connectivity measures the strength that the unlabeled superpixel is connected to the labeled set. Thus, $d_n^F$ and $d_n^B$ are computed as follow:

$$
\begin{align*}
  d_n^F &= \frac{1}{|\mathcal{N}(\varphi_n)|} \sum_{m \in \mathcal{N}(\varphi_n)} \| \varphi_n, \varphi_m \|^2 + \frac{1}{|\mathcal{L}_F^t|} \sum_{m \in \mathcal{L}_F^t} \mathcal{D}(\mu_n, \mu_m), \\
  d_n^B &= \frac{1}{|\mathcal{N}(\varphi_n)|} \sum_{m \in \mathcal{N}(\varphi_n)} \| \varphi_n, \varphi_m \|^2 + \frac{1}{|\mathcal{L}_B^t|} \sum_{m \in \mathcal{L}_B^t} \mathcal{D}(\mu_n, \mu_m),
\end{align*}
$$

(7)

where $\mathcal{N}(\varphi_n)$ contains all the neighboring superpixels of $\varphi_n$, $|\mathcal{N}(\varphi_n)|$ is the number of $\varphi_n$’s neighbors, $\| \varphi_n, \varphi_m \|^2$ computes the Euclidean distance between superpixel $\varphi_n$ and $\varphi_m$, $|\mathcal{L}_F^t|$ and $|\mathcal{L}_B^t|$ are the numbers of superpixels in $\mathcal{L}_F^t$ and $\mathcal{L}_B^t$ respectively. $\mathcal{D}(\mu_n, \mu_m)$ computes the geodesic distance between $\mu_n$ and $\mu_m$ as follow:

$$
\mathcal{D}(\mu_n, \mu_m) = \min_{v_1 = n, v_2, ..., v_r = m} \max_{k=1}^{r-1} \left( \| v_k, v_{k+1} \|^2 - a, 0 \right),
$$

(8)

s.t. $v_k, v_{k+1} \in V$, $v_k$ and $v_{k+1}$ are connected in the undirected graph $G$, $\| v_k, v_{k+1} \|^2$ computes the Euclidean distance between $v_k$ and $v_{k+1}$, and $a$ is an adaptive threshold preventing the “small-weight-accumulation” problem ([55, 96]). Thus, the $\mathcal{D}(\mu_n, \mu_m)$ measures the shortest path (geodesic) between $\mu_n$ and $\mu_m$ in $G$. 

44
Saliency map

After computing the difficulty of all the candidate superpixels in $C^F_t$ and $C^B_t$ to their corresponding labeled sets $L^F_t$ and $L^B_t$, two sets of superpixels are selected, $P^F_t$ for foreground propagation and $P^B_t$ for background propagation. The difficulty scores $d_n^F$ of the superpixels in $C^F_t$ are sorted in ascending order, and the first $q_F$ superpixels are selected to $P^F_t$. $q_F$ at time $t$ is computed by

$$q_F = \left\lfloor \frac{|C^F_t| \times \delta^F_t}{} \right\rfloor,$$

where $\delta^F_t$ is computed as follow:

$$\delta^F_t = 1 - \frac{2}{q_F} \sum_{n=1}^{q_F-1} \min(f^{(t-1)}_n, 1 - f^{(t-1)}_n).$$

$\delta^F_t$ is learned from the labeled set $L^F_{t-1}$ at time $t-1$, which determines the percentage of superpixels we will select from $C^F_t$. $\delta^F_t$ is high if the saliency values of the superpixels in $L^F_{t-1}$ are close to 1 (foreground) or 0 (background). When the saliency values are close to 0.5, it becomes ambiguous to judge whether the values are salient or not. Thus, $\delta^F_t$ is set small to avoid choosing ambiguous superpixels from $C^F_t$.

In a similar way, a set of superpixels $P^B_t$ for background propagation can be selected by ranking the difficulty scores $d_n^B$ of $C^B_t$ in ascending order and choose the first $q_B$ superpixels.

$$q_B = \left\lfloor \frac{|C^B_t| \times \delta^B_t}{} \right\rfloor,$$

$$\delta^B_t = 1 - \frac{2}{q_B} \sum_{n=1}^{q_B-1} \min(b^{(t-1)}_n, 1 - b^{(t-1)}_n).$$

There is a special case when the $n$-th superpixel belongs to a set $P^t$, where $p^t_n \in P^t = P^F_t \cap P^B_t \neq \emptyset$. In such case, we classify the $n$-th superpixel by comparing its difficulty scores $d_n^F$ and $d_n^B$. If $d_n^F \geq d_n^B$, $P^t = P^F_t \setminus p^t_n$; otherwise, $P^t = P^B_t \setminus p^t_n$.

As the superpixels for foreground propagation $P^F_t$ and background propagation $P^B_t$ are both determined, we need to spread the saliency values in $L^F_t$ to $P^F_t$ and the unsaliency values in $L^B_t$ to $P^B_t$ respectively by

$$f^{t+1} = A^F_t \cdot D^{-1} \cdot W \cdot f^t,$$

$$b^{t+1} = A^B_t \cdot D^{-1} \cdot W \cdot b^t.$$
where $A^t_F$ is a diagonal matrix with $A^t_{nn} = 1$ if the $n$-th superpixel belongs to $\mathcal{L}_F^t \cup \mathcal{P}_F^t$, otherwise $A^t_{nn} = 0$. Similarly, $A^t_B$ is diagonal with $A^t_{nn} = 1$ if the $n$-th superpixel belongs to $\mathcal{L}_B^t \cup \mathcal{P}_B^t$, otherwise $A^t_{nn} = 0$. After the $t$-th iteration, the labeled set is $\mathcal{L}^{t+1} = \mathcal{L}^t \cup \mathcal{P}_F^t \cup \mathcal{P}_B^t$, and the unlabeled set is $\mathcal{U}^{t+1} = \mathcal{U}^t \setminus (\mathcal{P}_F^t \cup \mathcal{P}_B^t)$.

When all the unlabeled superpixels are labeled after $T_2$ times iteration, it results in a set of superpixels involved in foreground propagation $\mathcal{F}$ and a set of superpixels involved in background propagation $\mathcal{B}$. If the $n$-th superpixel is involved in foreground propagation, its saliency value is now $f_n^{T_2}$; otherwise, its unsaliency value is $b_n^{T_2}$. Then, the unsaliency values $b_n^{T_2}$ are transferred into saliency values based on $\mathcal{F}$ by $\overline{b}_n^{T_2} = \min(\mathcal{F}) \times (1 - b_n^{T_2})$. Finally, the saliency values in $\mathcal{F}$ and the transferred saliency values in $\mathcal{B}$ are mapped to the original image for the final saliency map.

### 3.3.2 Experiments

#### Datasets

Salient object detection (or salient object segmentation) is investigated in object level and in the discrete domain. The data collection process is based on the annotation from the labelers who draw pixel-wise accurate silhouettes of the objects that are believed to be salient. A binary salient object mask is created for each image to represent the pixel-wise saliency information. This part briefly introduces some widely used salient object detection datasets, which will appear for evaluations in the remaining chapters.

**MSRA-B** [97] is one of the most prevalent datasets for salient object detection, with 5000 images of various categories provided by 3-9 labelers. The images may consist of one or more salient objects on the scenes.

**ECSSD** [98] contains a pool of 1000 images with even more complex salient objects on the scenes. Meanwhile, the objects on the images are semantically meaningful.

**DUT-OMRON** [95] is a large-scale salient object detection dataset with a number of 5168 images. Each image contains one or more salient objects and relatively a complex background.

**PASCAL-S** [8] is a dataset for salient object detection consisting of a set of 850 images from PASCAL VOC 2010 ([99]) with multiple salient objects on the scenes. The salient objects on each image are segmented with different saliency levels. Usually the ground truth maps are thresholded by 128 for further evaluation.

**HKU-IS** [65] is a dataset contains a pool of 4447 images with pixel-wise annotations of salient objects, of which most images are either of low contrast or with multiple salient objects on the scenes.
**ImgSal** [100] is a relative complex dataset of 235 images in six levels of complexity, including 50 images with large salient regions, 80 with intermediate salient regions, 60 with small salient regions, 15 with cluttered backgrounds, 15 with repeating detractors, and 15 with both large and small salient regions.

**ICoSeg** [101] dataset is primarily designed for co-segmentation tasks, which contains 643 images with pixel-wise annotation on the scenes. Each image may consist multiple salient objects.

**ASD** [46] dataset is one of the most widely used datasets with 1000 images from the MSRA-5000 Saliency Object Database [97], with distinct salient objects on the scenes.

We investigate the bi-directional propagation model (BIP) by evaluating it over four challenging datasets: DUT-OMRON, ECSSD, PASCAL-S, and ASD. We compare the proposed BIP model with fourteen state-of-the-art saliency models including BSCA [102], COV [50], DRFI [103], GBVS [104], GC [105], GP [106], HS [98], LR [107], MB [108], MR [95], PCAS [109], RB [55], TLLT [96], and UFO [103].

**Evaluation Metrics**

There are usually two types of evaluation metrics to evaluate the performance of the salient object detection models, that is F-measure and mean absolute error (MAE).

**F-measure** is a region based evaluation metric by setting a segmentation threshold for binary segmentation to evaluate the ground truth map and the saliency map. When a given saliency map is slidingly thresholded from 0 to 255, a precision-recall (PR) curve can be computed based on the ground truth. F-measure is computed to count for the saliency maps with both high precision and recall:

$$F\text{-}measure = \frac{(1 + a^2) \cdot \text{precision} \cdot \text{recall}}{a^2 \cdot \text{precision} + \text{recall}},$$

where $a^2$ is set as 0.3 [46] to emphasize the precision.

In practice, there are different ways to count the final F-measure. The maximum F-measure account for the maximum value of F-measure, the mean F-measure computes the mean values of the F-measure, and the weighted F-measure is proposed by [110] for handling the existing flaws of F-measure, the mean F-measure. Optionally, the threshold setting can also be defined as adaptive that is twice the average value of the whole saliency map.

**Mean Absolute Error** (MAE) measures the overall pixel-wise difference between the saliency map and the ground truth [111]. MAE is defined as follow:

$$\text{MAE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} |s_{ij} - g_{ij}|,$$
Fig. 9. Performance enhancement by comparing the proposed BIP model to the fourteen state-of-the-art saliency models (increase in max F-measure and decrease in MAE) on ASD, ECSSD, DUT-OMRON and PASCAL-S datasets. Preprinted by permission, Paper II © Elsevier.

\[ \text{MAE} = \frac{1}{N} \sum_{i=1}^{N} \|F_i - G_i\|. \]  

(14)

Experimental Analysis

The proposed BIP model is compared with fourteen state-of-the-art unsupervised saliency models, by comparing F-measure and MAE scores over four datasets. Figure 9 further plots four bar charts about the performance enhancement in F-measure and MAE by comparing the BIP model to every selected saliency model. Obviously, the BIP model outperforms every selected saliency model in both F-measure and MAE. Figure 10 shows some selected examples of the proposed BIP model and the state of the arts.

For single directional propagation, the average iteration number of foreground propagation is 6 while that of background propagation is 7. However, the bi-directional propagation only needs averagely \( T_2 = 4 \) iterations, which is much less than single directional method. Moreover, the average computational time of bi-directional propagation is
Fig. 10. Examples of the results of the state-of-the-art saliency models and the proposed BIP model. The original images, results of state-of-the-art saliency models and BIP model, and ground truth (GT) are sequentially presented. Permitted by permission, Paper II © Elsevier.

0.023s per image, while that of the single directional propagation is 0.030s (foreground propagation) and 0.039s (background propagation) for each image. Thus, bi-directional propagation outperforms single directional propagation in both iteration numbers and computational time.

3.4 summary

This chapter reviews the development of computational models for salient object detection and specifically focuses on object level unsupervised saliency methods. A bi-directional propagation model (BIP) for salient object detection in Paper II is introduced. The BIP model performs foreground propagation and background propagation at the same time in every individual iteration with a difficulty-based rule. The difficulty-based rule evaluates the difficulties of each unlabeled superpixel to the labeled foreground set.
and the labeled background set respectively by its distinctness to the neighborhood and its connectivity to the two unlabeled sets accordingly. Moreover, experimental results show that the BIP model largely reduces both the iteration numbers and the computational time compared to the previous single directional propagation methods. The BIP model outperforms fourteen saliency models on four challenging datasets and is still among the state of the arts in terms of unsupervised approaches.

Paper II is my first work on salient object detection. I started from the unsupervised method and later moved on to deep learning based methods for salient object detection. Since the proposed bi-directional propagation model in Paper II is graph-based methods, I continued to explore graph-based methods for saliency refinement in deep saliency models and thus propose two works as published in Paper III and Paper IV.
4 Multi-scale CNNs with CRFs for salient object detection

4.1 Introduction

Due to the emergence of deep learning, the performance of saliency detection has been largely improved [57, 66, 112, 113, 70, 72, 65, 114, 115, 116]. These recently proposed deep saliency models break the limits of extracting handcrafted features as done in traditional methods, such as color, intensity, orientation, textures and so forth. Instead, convolutional neural networks (CNNs) are employed to extract high-level features of rich semantics. Further, the integration of fully convolutional neural networks (FCNs) facilitates salient object detection tasks to an end-to-end phase [70, 71, 117, 118, 119]. It is notable that the deep neural networks are also inspired by biological processes, in which the connectivity between neurons perceives the organization of the visual cortex. Thus, many deep saliency models, although adopt deep neural network architectures, still follow some similar saliency detection hypotheses and strategies.

Firstly, the integration of multi-scale contextual information is still one of the core strategies in saliency detection. Early deep saliency models construct multi-scale inputs to train the CNNs for saliency detection, of which the inputs with multiple contexts are either the multi-scale re-sized images [68] or global and local contextual image segments [65, 66, 67, 69]. Later, as the end-to-end modeling gains much popularity, a number of deep saliency models take the advantages of the multi-scale side outputs [120, 119, 70, 117] from CNNs, and adopt various fusion strategies to produce the saliency maps.

Secondly, the integrity of smoothing or refinement is still crucial to enhance the quality of the outputs from deep neural networks. One common approach is to integrate smoothing functions [121, 122] directly to the output layer of the CNNs. Many previous works adopt the Conditional Random Fields (CRFs) module fully connected to the CNNs, known as Dense-CRF [123], to reconcile the spatial and appearance coherence on the output saliency map [65, 124, 72]. Although Dense-CRF is disconnected from the training of the CNNs, it still efficiently improves the quality of the saliency maps in the post-processing stage.

My research works on salient object detection with deep neural networks focus on two perspectives. Firstly, incorporating multi-scale CNN side outputs for a better saliency map is investigated. Secondly, the integration of CRF with CNNs for coarse to
fine saliency map refinement is further explored. With these two objectives, two papers are published. Paper III takes advantage of multi-scale contexts from CNNs and embed multi-scale Dense-CRFs for salient object detection. The proposed model is compact and efficient in training, but the Dense-CRFs are disconnected from the training of the CNNs. Thus, further exploration is later published in Paper IV that proposes a cascade CRF structure for salient object detection, which can be jointly trained with CNNs via back propagation.

In this chapter, a brief literature review will first be given in Section 4.2 to introduce recent works applying CNNs for saliency detection and recent CNN based saliency models utilizing Dense-CRF as post-processing method for saliency refinement. And Section 4.3 reports Paper III, which proposes the salient object detection with CNNs and multi-scale CRFs. In Section 4.4, Paper IV will be introduced which proposes a cascade CRF structure for salient object detection that can be jointly trained with CNNs via back propagation. Section 4.5 summarizes the contributions of the two included papers and discusses the future works.

4.2 Related works

4.2.1 CNN-based saliency models

In the past few years, a broad range of saliency models based on deep neural networks has been proposed.

The first attempt of adopting CNNs for visual saliency modeling is for fixation prediction task. [125] propose the eDN model leveraging CNNs for salient fixations prediction. [126] adopt deep neural networks for fixation prediction. [127] propose a shallow convnet and a deeper convnet for end-to-end saliency detection. [128] propose to use convnet and generate saliency maps with Bernoulli distribution. [129] propose to predict saliency based on deep neural network features. [130] further adopt long-term recurrent convolutional network structure for saliency detection.

Early deep salient object detection models benefit from adjusting the inputs to VGG [64], of which the inputs are either multi-scale resized images [68] or global and local image segments [65, 66, 67, 69]. Later, deep saliency models extensively take the advantages of the multi-scale contexts from CNNs and adopt various fusion strategies to produce the saliency map. A hierarchical architecture can effectively refine the CNN side outputs from coarse to fine scales [70, 131, 74]. PiCANet [74] hierarchically embeds global and local contexts. Moreover, some saliency models adopt recurrent or cascade structures to progressively learn saliency maps from coarse to
fine scales [71, 117, 73, 132]. [73] introduce a multi-path recurrent feedback scheme to progressively enhance the saliency prediction map. RA [132] introduces reverse attention with side-output residual learning to refine the saliency map in a top-down manner. Also, skip connections are widely applied to integrate prediction maps from CNNs [119, 72]. DSS [72] adopts short connections to the side output layers of CNNs to fuse multiple prediction maps.

### 4.2.2 Disconnected CRF with CNNs

The conditional random field (CRF) is based on a flexible graphical model to incorporate label agreement assumptions, which is broadly adopted to labeling refinement tasks. Deeplab [133] firstly connects a Dense-CRF to deep neural networks for semantic segmentation refinement, based on unary and pairwise potentials proposed by [123]. The proposed Dense-CRF is fully connected on top of CNNs as a post-processing method for end-to-end refinement.

Dense-CRF works on the discrete domain, which yields an effective iterative message-passing algorithm using mean-field theory. The mean-field approximation can be performed using highly efficient Gaussian filtering in feature space, reducing the complexity from quadratic to linear. Several deep salient object detection models [112, 72, 124] take the advantages of Dense-CRF and fully connect a CRF layer to the end of CNNs as a post-processing method, which effectively improves the quality of saliency maps. We propose the multi-scale CRFs saliency model (MCRF) that connects multi-scale Dense-CRFs to CNNs in Paper III, which effectively incorporates and refines multi-scale CNN side outputs for a refined saliency map.

### 4.2.3 Joint CRF with CNNs

Dense-CRF, although efficient in saliency refinement, is disconnected from the training of CNNs, and thus its parameters are pre-selected by cross validations from a large number of trials.

[134] firstly implement the CRF layer on top of CNN predictions that enables joint model training through back propagation for semantic segmentation tasks, in the discrete domain. To solve depth estimation in the continuous domain, [135] introduce the continuous CRF that incorporates multi-scale CNN prediction maps. Later, [136] also propose the attention gated CRF that allows message-passing among the continuous features for contour prediction.
All the previous models formulate CRF with message-passing only among features or among predictions. We propose the deep unified CRF saliency model in Paper IV, which firstly formulates the continuous feature variables and the discrete prediction variables into a deep unified CRF model.

4.3 Multi-scale CNNs with disconnected Dense-CRFs

Paper III proposes the multi-scale CRFs saliency model (MCRF) based on multi-scale side outputs from FCNs. Specifically, a fully convolutional neural network based on the encoder-decoder architecture with three scales of side output maps is trained with pixel-wise labels. Then, a CRF layer is connected to each side output layer to refine the delineation and smoothness of the side output maps. Finally, the refined side output maps are fused and then refined by another CRF layer for the final saliency map.

4.3.1 Formulation

The standard encoder-decoder network is adopted as the front-end networks for pre-training, in order to obtain multi-scales of side output maps with multiple contextual information.

Given the input image \( I = \{I_i, i = 1, \cdots, |I|\} \) with three-dimensional size of \( \text{Height} \times \text{Width} \times 3 \), and the ground truth \( G = \{G_i, i = 1, \cdots, |G|\} \), \( G_i \in \{0, 1\} \) with the size of \( \text{Height} \times \text{Width} \times 1 \), the encoder-decoder networks \( \mathcal{F} \) is adopted to produce \( M = \{m = 1, \cdots, M\} \) scales of side output feature maps, denoted as \( s_m \) as follow:

\[
\tag{15}
 s_m = \mathcal{F}(W, w_m),
\]

where \( W \) denotes the generic weights of the encoder-decoder networks and \( w_m \) denotes the scale specific weights. In the training phase, the cross-entropy loss is utilized as the side objective function.

Through the encoder-decoder networks, \( M \) scales of side output maps are computed to primarily locate the salient objects. In order to further improve the prediction accuracy, a fully connected CRF [123] layer is integrated to each side output layer for refinement as follow:

\[
\tag{16}
 \hat{s}_m = \text{CRF}_m(s_m, I, \Theta_m),
\]

where \( \Theta_m \) refers to all the parameters for the \( m \)-th CRF layer, and \( \hat{s}_m \) represents the refined side output map at the \( m \)-th scale.

To each side output map \( \hat{s}_m \), the energy function of the CRF is
Fig. 11. Framework of the proposed multi-scale CRFs model. Three scales of side output maps are selected from the encoder-decoder networks. The encoder network is based on the VGG-16 net [64]. Then, the decoder network is connected to the “pool5” layer, which gradually unpools the features from the corresponding pooling layers. The decoder convolutional layers are all followed by a BN layer and a ReLU layer. To upsample the three scales of side outputs from “Deconv1”, “Deconv2” and “Deconv3”, a convolutional layer with $1 \times 1$ kernel size is used to compute the one channel feature map and a deconvolutional layer followed with a crop layer is connected to upsample the feature maps to the image size respectively. To finetune the front-end encoder-decoder networks, each side output map is connected with a side loss for optimization. Then, one CRF layer is connected to each side output map for multi-scale refinement and the refined side output maps are fused by element-wise product. Finally, another CRF layer is connected to refine the fused map for the final saliency map. The CRF layers are tuned by cross validations and all the CRF layers share the same parameter settings. Prepinted by permission, Paper III © Springer

$$E(G) = \sum_i \phi(s^i_m) + \sum_{i < k} \psi(s^i_m, s^k_m).$$

(17)

$\phi(s^i_m)$ refers to the unary term, where the side output maps are directly regarded as the input. $\psi(s^i_m, s^k_m)$ is the pairwise term, which accounts for the coherence of the saliency information and image features between the current pixel and its neighbors. Thus, the pairwise term is defined as:

$$\phi_p(s^i_m, s^k_m) = \mu(s^i_m, s^k_m)(K^1 + K^2),$$

(18)
where \( \mu(s_m^i, s_m^k) = 1 \) if \( s_m^i = s_m^k \) and otherwise 0. \( I_i \) represents the RGB image features of the \( i \)-th pixel, while \( p_i \) is the pixel position. The Gaussian kernel \( K_1 = \nu_1 \exp(-\frac{\|p_i - p_k\|^2}{2\sigma^2_a}) \) measures the appearance coherence which refines the nearby pixels with similar features with similar saliency scores, while the Gaussian kernel \( K_2 = \nu_2 \exp(-\frac{\|p_i - p_k\|^2}{2\sigma^2_s}) \) measures the spatial coherence which reconciles close pixels with similar saliency scores. Parameters \( \nu_1 \) and \( \nu_2 \) control the contributions of each Gaussian kernel respectively.

The energy minimization is based on the mean field approximation to the CRF distribution proposed by [123], and high-dimensional filtering can be utilized to speed up the computation.

Then, the refined saliency maps from each scale of the CRF layer are fused by element-wise production:

\[
\bar{s} = \prod_{m=1}^{M} \hat{s}_m. \tag{19}
\]

Finally, another CRF layer is connected to further refine the fused map as the final saliency map:

\[
\bar{s}_{\text{final}} = \text{CRF}_{\text{final}}(\bar{s}, I, \Theta_{\text{fuse}}). \tag{20}
\]

### 4.3.2 Model training

The training protocol is the same as in [72, 112], using the MSRA-B dataset [97] as the training data for fair comparisons, with 2,500 training images, 500 validation images and 2000 testing images. The input images are resized to 240 \times 320. Horizontal flipping is used for data augmentation such that the number of training samples is twice as large as the original number.

The network architecture is demonstrated in Figure 11 with detailed layer descriptions. The front-end encoder-decoder network is adopted to obtain the multi-scale side output maps. The encoder network is based on the VGG-16 net [64].

The hyper-parameters for the finetuning the encoder-decoder network are set as: a fixed learning rate (1e-8), weight decay (0.0005), momentum (0.9), loss weight for each side output (1). The batch size is set as 12, and 100 epochs are performed for tuning the encoder decoder network. The sigmoid cross entropy loss layers are used for model optimization. The fully connected dense CRF layers share the same parameter settings and are tuned via cross validations on the validation set, and \( \nu_1, \nu_2, \sigma_a, \sigma_s, \) and \( \sigma_y \) are set to 3, 3, 60, 5, and 3, respectively. 3 iterations of the meanfield approximation are set.
Table 3. Evaluation results over four datasets, with models including MDF [137], RFCN [71], DHS [70], Amulet [119], UCF [131], DCL [112], MSR [124], DSS [72], RA [132] and the proposed MCRF model. "+" marks the models utilizing dense CRF [123] for post-processing. "-" means that the corresponding dataset is used as the training data. The evaluation on MSRA-B is performed on the testing set. The best performances are in bold while the second best results are underlined. Preprinted by permission, Paper III © Springer.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>MDF</th>
<th>RFCN</th>
<th>DHS</th>
<th>Amulet</th>
<th>UCF</th>
<th>DCL</th>
<th>MSR</th>
<th>DSS</th>
<th>RA</th>
<th>MCRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSRA-B</td>
<td>$F_\beta$</td>
<td>0.862</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.898</td>
<td>0.880</td>
<td><strong>0.917</strong></td>
<td>0.901</td>
<td><strong>0.917</strong></td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.066</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.047</td>
<td>0.053</td>
<td><strong>0.034</strong></td>
<td>0.050</td>
<td><strong>0.038</strong></td>
</tr>
<tr>
<td>DUT-OMRON</td>
<td>$F_\beta$</td>
<td>0.677</td>
<td>0.693</td>
<td>-</td>
<td>0.668</td>
<td>0.630</td>
<td>0.716</td>
<td>0.695</td>
<td><strong>0.757</strong></td>
<td>0.719</td>
<td><strong>0.726</strong></td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.092</td>
<td>0.095</td>
<td>-</td>
<td>0.098</td>
<td>0.120</td>
<td>0.080</td>
<td>0.078</td>
<td><strong>0.063</strong></td>
<td>0.076</td>
<td><strong>0.074</strong></td>
</tr>
<tr>
<td>HKU-IS</td>
<td>$F_\beta$</td>
<td>0.809</td>
<td>0.868</td>
<td>0.863</td>
<td>0.849</td>
<td>0.816</td>
<td>0.851</td>
<td>0.864</td>
<td><strong>0.904</strong></td>
<td>0.879</td>
<td><strong>0.894</strong></td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.076</td>
<td>0.073</td>
<td>0.052</td>
<td>0.050</td>
<td>0.061</td>
<td>0.049</td>
<td><strong>0.040</strong></td>
<td><strong>0.040</strong></td>
<td>0.057</td>
<td><strong>0.040</strong></td>
</tr>
<tr>
<td>ECSSD</td>
<td>$F_\beta$</td>
<td>0.818</td>
<td>0.871</td>
<td>0.875</td>
<td>0.865</td>
<td>0.835</td>
<td>0.874</td>
<td>0.838</td>
<td><strong>0.899</strong></td>
<td>0.876</td>
<td><strong>0.893</strong></td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.106</td>
<td>0.091</td>
<td>0.059</td>
<td>0.061</td>
<td>0.071</td>
<td>0.068</td>
<td>0.062</td>
<td><strong>0.055</strong></td>
<td>0.080</td>
<td><strong>0.058</strong></td>
</tr>
</tbody>
</table>
to each CRF layer. All the implementation is based on the public Caffe library [138]. The CRF is based on the PyDenseCRF implementation by [123].

4.3.3 Experiments

Table 3 lists the mean F-measure and MAE of nine saliency models and the proposed MCRF model over four datasets. It is clearly observed that the MCRF model surpasses most of the existing saliency models with much better performances. Compared to MDF [137], DCL [112], MSR [124] that apply single CRF layer, the multi-scale CRF model results in superior performances. Moreover, the proposed MCRF model receives comparable performances with the DSS [72] model. Compared to the DSS [72] that uses the enhanced HED architecture with five scales of side outputs (totally 53 convolutional and deconvolutional layers), the proposed MCRF model is based on the simple encoder-decoder architecture and only three scales of side outputs (totally 31 convolutional and deconvolutional layers) are fused for multi-scale integration. Thus, the multi-scale CRF structure is proved to be efficient.

4.4 Multi-scale CNNs with joint CRFs

Paper IV proposes a deep cascade CRFs architecture that is seamlessly incorporated with CNN to integrate and refine multi-scale deep features and predictions for a refined saliency map. At each scale, a CRF block is embedded that takes the features and predictions from the lower scale as observed variables to estimate the hidden features and predictions at the current scale. Within each CRF inference, feature-feature, feature-prediction and prediction-prediction messages passing are built. Then, the output refined features and the prediction map are incorporated into the CRF block at the next scale. Thus, a series of CRFs are constructed in a cascade flow and progressively learn a unified saliency map from the coarse scale to the finer scale. The new CRF formulation provides explainable solutions for the features, the predictions and the interactions among them, leading to distinct model formulation, inference, and neural network implementation.

4.4.1 Formulation

Given the input image $I$ of $N$ pixels, suppose that a backbone CNN network computes $L$ scales of deep feature maps $F = \{f^l\}_{l=1}^L$, where $f^l = \{f^l_{i,m}\}_{i=1,m=1}^{N,M}$ consists a set of $M$ feature vectors. Accordingly, $L$ scales of prediction maps $S = \{s^l\}_{l=1}^L$ can be computed,
Fig. 12. Framework of the proposed deep unified CRF saliency model for jointly modeling structural deep features and predictions. Multi-scale features \( f^1 \cdots f^5 \) and the corresponding prediction maps \( s^1 \cdots s^5 \) are extracted from the backbone CNN. At each scale, a CRF block is embedded to jointly refine features and prediction maps with message-passing between features and features \( (f-f) \), features and predictions \( (f-s) \), and predictions and predictions \( (s-s) \). "h^1 \cdots h^5" and "o^1 \cdots o^5" are the estimated features and predictions at each scale respectively. "f^1 \cdots f^5" correspond to "pool5a", "conv5_3", "conv4_3", "conv3_3" and "conv2_2" in the enhanced HED [139] structure, while "s^1 \cdots s^5" are "upscore-dsn6", "upscore-dsn5", "upscore-dsn4", "upscore-dsn3", and "upscore-dsn2". The dashed arrows omit the details within the backbone CNN. Prepinted by permission, Paper IV © IEEE 2019.

where \( s^l = \{s^l_i\}_{i=1}^N \). The ground truth saliency map corresponding to the input image is denoted as \( g = \{g_i\}_{i=1}^N \), and each element \( g_i \) takes binary values of 1 or 0.

The CRF inference is formulated to jointly refine multi-scale features and predictions. The objective is to approximate the hidden multi-scale deep feature maps \( H = \{h^l\}_{l=1}^L \) and the hidden multi-scale prediction maps \( O = \{o^l\}_{l=1}^L \). In particular, at the \( l \)-th scale, the observed variables are the features \( f^{l-1} \), \( f^l \) and the prediction \( s^{l-1} \), and the objective is to estimate the corresponding \( h^l \) and \( o^l \). With a cascade flow of a series of CRFs, the side outputs are progressively refined from coarse \((l = 1)\) to fine \((l = L)\). The refined prediction map \( o^L \) is the final saliency map.

The conditional distribution of the CRF at the \( l \)-th scale is defined as follow:

\[
P(h^l, o^l | I, \Theta) = \frac{1}{Z(I, \Theta)} \exp \left\{ -E \left( h^l, o^l, I, \Theta \right) \right\},
\]

where \( \Theta \) refers to the relative parameters. The energy function \( E = E(h^l, o^l, I, \Theta) \) is formulated as follow:
\[ E = \sum_i \phi_h(h_i', f_i') + \sum_i \phi_o(s_i', o_i') + \sum_{i \neq j} \psi_h(h_i', h_j') + \sum_i \psi_o(h_i', o_i') + \sum_{i \neq j} \psi_o(o_i', o_j'). \]  

(22)

The first term of Eq. 22 is a feature level unary term corresponding to an isotropic Gaussian:

\[ \phi_h(h_i', f_i') = -\frac{\alpha_i'}{2} \|h_i' - f_i'\|^2, \]  

(23)

where \( \alpha_i' > 0 \) is a weighting factor.

The second term is a prediction level unary term as follow:

\[ \phi_o(s_i', o_i') = \|s_i' - o_i'\|^2. \]  

(24)

The third term is a feature level pairwise term describing the potential between features as follow:

\[ \psi_h(h_i', h_j') = h_i' W_{i,j} h_j'^{-1}, \]  

(25)

where \( W_{i,j} \in \mathbb{R}^{M \times M} \) is a bilinear kernel.

The fourth term is a feature level pairwise term defining the potential between features and predictions, where

\[ \psi_o(h_i', o_j') = h_i' V_{i,j} o_j'^{-1}, \]  

(26)

where \( V_{i,j} \in \mathbb{R}^{M \times M} \) is also a bilinear kernel to couple the features and the predictions. \( o_j'^{-1} \) denotes a concatenation of \( M \) prediction maps \( o_j^{-1} \). The fifth term is a prediction level pairwise term defining the potential between the predictions as follow:

\[ \psi_o(o_i', o_j') = \beta_i K_{i,j}^1 |o_i' - o_j'|^2 + \beta_2 K_{i,j}^2 |o_i' - o_j'|^2, \]  

(27)

where \( K_{i,j}^1 \) and \( K_{i,j}^2 \) are Gaussian kernels that measure the relationship between two pixels.

The mean-field approximation is adopted to estimate a distribution \( q(h', o'| \mathbf{I}, \Theta) = \prod_{i=1}^N q_i(h_i', o_i') \) that is an approximation to \( P(h', o'| \mathbf{I}, \Theta) \) by minimizing the Kullback-Leibler divergence ([140]).

By considering \( J_{i,j} = \log q_{i,j}(h', o'| \mathbf{I}, \Theta) \) and rearranging its expression into an exponential form, the mean-field updates can be derived.

The latent features of the mean-field inference can be estimated as:

\[ \bar{h}_i' = \frac{1}{\alpha_i'} \left( \alpha_i' f_i' + \sum_{l \neq i} \sum_{l \neq j} W_{i,j} h_{l}'^{-1} + \sum_{l \neq i} \sum_{l \neq j} V_{i,j} o_{l}'^{-1} \right). \]  

(28)
The prediction variables of the mean-field inference can be estimated as:

\[
\mu^l_i = \frac{o^l_i}{\rho^l_i} + \frac{2}{\rho^l_i} \left( \beta_1 \sum_{j \neq i} K^1_{ij} \mu^l_j + \beta_2 \sum_{j \neq i} K^2_{ij} \mu^l_j \right),
\]

(29)

where the variance of the mean-field approximated distribution used as the normalization factor is computed as

\[
\rho^l_i = 1 + 2 \left( \beta_1 \sum_{j \neq i} K^1_{ij} + \beta_2 \sum_{j \neq i} K^2_{ij} \right),
\]

(30)

At the \(l\)-th scale, the optimal \(o^l\) can be approximated by mean-field updates of \(T\) iterations on the prediction level. At each time \(t\) of the mean-field iteration, an estimated saliency map \(\mu^l_t\) can be approximated. After \(T\) mean-field iterations, the estimated prediction map \(\mu^l_T\) is regraded as the estimation of \(o^l\) from the CRF at the \(l\)-th scale. The details of the mean-field updates are presented in Figure 13.

In the cascade flow, the observation \(s^l\) is obtained via integrating the prediction map \(s^l\) and the estimated map \(o^{l-1}\) from the CRF at the previous scale, i.e., \(s^l_i = s^l_i + o^{l-1}_i\).

4.4.2 Implementing mean-field iteration with neural networks

The inference of the CRF block is based on the mean-field approximation, which can be implemented as a stack of CNN layers to facilitate joint training as in Figure 13. The mean-field updates for Eq. 28 is implemented with convolutions. The similarity kernel \(K^1\) and the proximity kernel \(K^2\) in Eq. 29 and 30 are computed based on permutohedral lattice [141] to reduce the computational cost from quadratic to linear [140]. The weighting of \(\beta_1\) and \(\beta_2\) is convolved with an \(1 \times 1\) kernel. By combining the outputs, the normalization matrix \(\rho^l\) and the corresponding \(\mu^l\) can be computed. The weights \(\beta_1\) and \(\beta_2\) are obtained by back propagation.

4.4.3 Model training

The model uses the MSRA-B dataset [97] which consists of 2,500 training images, 500 validation images, and 2000 testing images. The images are resized to \(240 \times 320\) and horizontal flipping is used for data augmentation.

The front-end CNN is based on the implementation of DSS [72] with the enhanced HED [139] structure. The only difference is that the side output prediction maps computed from the layer “conv1_2” is discarded. Thus, totally five scales of side outputs are extracted.
Fig. 13. Details of the mean-field updates within CRF. The circled symbols indicate message-passing operations within the CRF block. (i) Message-passing to estimate $h'$ by convolutions (Eq. 28): “C” indicates a convolutional layer followed by the corresponding deconvolutional layer, crop layer and a scale layer. (ii) Message-passing to estimate $o'$ with Gaussian pairwise kernels in $\tau$ iterations (Eq. 29): “G” means the Gaussian filtering. “F” is the process of computing a prediction map by a convolutional layer with kernel size $1$ followed with the corresponding deconvolutional layer and a crop layer. “+” refers to element-wise sum. Preprinted by permission, Paper IV © IEEE 2019.

To reduce training time, the proposed deep unified CRF model is optimized with two stages, including a pre-training and an overall optimization. In the pre-training stage, the model is optimized by adding feature-feature and feature-prediction messages passing to the front-end CNN, and the side output prediction map at each scale is added with a sigmoid cross entropy loss function for optimization. In the second stage, the prediction-prediction message-passing at each scale is added and sigmoid cross-entropy loss function is computed for the final scale.

The VGG-16 [64] is adopted to initialize the parameters for the pre-training stage. The parameters for the pre-training stage are set as: batch_size (1), learning rate (1e-9), max_iter (14000), weight decay (0.0005), momentum (0.9), iter_size (10). The learning rate is decreased by 10% when the training loss reaches a flat. In the second training stage, the parameters learned from the pre-training stage are optimized with a learning rate of 1e-12, while the parameters for prediction-prediction message-passing are learned with the learning rate as 1e-8. Another 10 epochs are trained for the overall optimization. The pre-training takes about 6 hours and the overall training takes about 14 hours.
Table 4. F-measure of the estimated prediction map $o^5$ by implementing pairwise terms in Eq. 22 to the deep unified CRF model for message-passing comparisons. “P” refers to message-passing between predictions, “F” means message-passing between features, and “/w P & F” means CRF with feature-feature, feature-prediction and prediction-prediction messages passing. Preprinted by permission, Paper IV © IEEE 2019.

<table>
<thead>
<tr>
<th>Method</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.884</td>
</tr>
<tr>
<td>Baseline + Dense-CRF (post-processing)</td>
<td>0.902</td>
</tr>
<tr>
<td>Baseline + CRF (/w P) (backbone output)</td>
<td>0.899</td>
</tr>
<tr>
<td>Baseline + CRF (/w P) (CRF output)</td>
<td>0.921</td>
</tr>
<tr>
<td>Baseline + CRF (/w F) (CRF output)</td>
<td>0.904</td>
</tr>
<tr>
<td>Baseline + CRF (/w P &amp; F) (CRF output)</td>
<td>0.928</td>
</tr>
</tbody>
</table>

4.4.4 Experiments

Ablation studies are carried out around the deep unified CRF model by involving variable combinations of message-passing within the inference. As in Table 4, the baseline is the backbone CNN and the F-measure is evaluated on the output prediction map $o^5$. By adding a Dense-CRF for post processing, the F-measure is 0.902. Then, the message-passing comparisons are conducted by implementing pairwise terms in Eq. 22 to the cascade CRFs architecture for joint model training. 1) By adding message-passing between predictions, the F-measure rises to 0.921. Also, by joint training CRF through back propagation, the prediction map from the baseline framework improves from 0.884 to 0.899. 2) By adding message-passing between features, the F-measure is 0.904, with 2% increase to the baseline output. Finally, by adding feature-feature, feature-prediction and prediction-prediction messages passing, the F-measure further improves to 0.928.

Table 5 lists the max F-measure and MAE of the ten saliency models and the proposed deep unified CRF model over six datasets. It is observed that the deep unified CRF model results in better F-measure and significantly reduced MAE.

4.5 Summary

In this chapter, two papers, Paper III Paper IV, are introduced to present two CNN based salient object detection models. Specifically, Paper III is based on multi-scale CNNs and embeds multi-scale CRFs at each scale of CNNs as a post-processing method for saliency refinement, while Paper IV first jointly refines multi-scale deep features and the corresponding multi-scale predictions with CRFs.
Table 5. Evaluation results on six dataset and with models DRFI [103], MDF [137], RFCN [71], DHS [70], Amulet [119], UCF [131], DCL [112], MSR [124], DSS [72], RA [132] and the deep unified CRF model. "+" marks the models utilizing Dense-CRF for post-processing. "-" means the corresponding dataset is used as the training data. The evaluation on MSRA-B is performed on the testing set. Prepinted by permission, Paper IVc © IEEE 2019.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>DRFI</th>
<th>MDF</th>
<th>RFCN</th>
<th>DHS</th>
<th>Amulet</th>
<th>UCF</th>
<th>DCL</th>
<th>MSR</th>
<th>DSS</th>
<th>DSS</th>
<th>RA</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSRA-B</td>
<td>maxF</td>
<td>0.851</td>
<td>0.885</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.916</td>
<td>0.930</td>
<td>0.920</td>
<td>0.928</td>
<td>0.931</td>
<td>0.935</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.123</td>
<td>0.066</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.047</td>
<td>0.042</td>
<td>0.043</td>
<td>0.035</td>
<td>0.036</td>
<td>0.029</td>
<td></td>
</tr>
<tr>
<td>PASCAL-S</td>
<td>maxF</td>
<td>0.690</td>
<td>0.759</td>
<td>0.829</td>
<td>0.824</td>
<td>0.832</td>
<td>0.818</td>
<td>0.822</td>
<td>0.852</td>
<td>0.826</td>
<td>0.831</td>
<td>0.829</td>
<td>0.858</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.210</td>
<td>0.142</td>
<td>0.118</td>
<td>0.094</td>
<td>0.100</td>
<td>0.116</td>
<td>0.108</td>
<td>0.081</td>
<td>0.102</td>
<td>0.093</td>
<td>0.101</td>
<td>0.089</td>
</tr>
<tr>
<td>ECSSD</td>
<td>maxF</td>
<td>0.664</td>
<td>0.694</td>
<td>0.747</td>
<td>-</td>
<td>0.743</td>
<td>0.730</td>
<td>0.757</td>
<td>0.785</td>
<td>0.764</td>
<td>0.781</td>
<td>0.786</td>
<td>0.802</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.150</td>
<td>0.092</td>
<td>0.095</td>
<td>-</td>
<td>0.098</td>
<td>0.120</td>
<td>0.080</td>
<td>0.069</td>
<td>0.072</td>
<td>0.063</td>
<td>0.062</td>
<td>0.057</td>
</tr>
<tr>
<td>HKU-IS</td>
<td>maxF</td>
<td>0.775</td>
<td>0.860</td>
<td>0.894</td>
<td>0.892</td>
<td>0.897</td>
<td>0.888</td>
<td>0.904</td>
<td>0.916</td>
<td>0.900</td>
<td>0.916</td>
<td>0.913</td>
<td>0.920</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.146</td>
<td>0.129</td>
<td>0.088</td>
<td>0.052</td>
<td>0.051</td>
<td>0.061</td>
<td>0.049</td>
<td>0.039</td>
<td>0.050</td>
<td>0.040</td>
<td>0.045</td>
<td>0.039</td>
</tr>
<tr>
<td>ECSSD</td>
<td>maxF</td>
<td>0.664</td>
<td>0.694</td>
<td>0.747</td>
<td>-</td>
<td>0.743</td>
<td>0.730</td>
<td>0.757</td>
<td>0.785</td>
<td>0.764</td>
<td>0.781</td>
<td>0.786</td>
<td>0.802</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.150</td>
<td>0.092</td>
<td>0.095</td>
<td>-</td>
<td>0.098</td>
<td>0.120</td>
<td>0.080</td>
<td>0.069</td>
<td>0.072</td>
<td>0.063</td>
<td>0.062</td>
<td>0.057</td>
</tr>
<tr>
<td>DUT-OMRON</td>
<td>maxF</td>
<td>0.775</td>
<td>0.860</td>
<td>0.894</td>
<td>0.892</td>
<td>0.897</td>
<td>0.888</td>
<td>0.904</td>
<td>0.916</td>
<td>0.900</td>
<td>0.916</td>
<td>0.913</td>
<td>0.920</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.146</td>
<td>0.129</td>
<td>0.088</td>
<td>0.052</td>
<td>0.051</td>
<td>0.061</td>
<td>0.049</td>
<td>0.039</td>
<td>0.050</td>
<td>0.040</td>
<td>0.045</td>
<td>0.039</td>
</tr>
<tr>
<td>ICoseg</td>
<td>maxF</td>
<td>0.664</td>
<td>0.694</td>
<td>0.747</td>
<td>-</td>
<td>0.743</td>
<td>0.730</td>
<td>0.757</td>
<td>0.785</td>
<td>0.764</td>
<td>0.781</td>
<td>0.786</td>
<td>0.802</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.150</td>
<td>0.092</td>
<td>0.095</td>
<td>-</td>
<td>0.098</td>
<td>0.120</td>
<td>0.080</td>
<td>0.069</td>
<td>0.072</td>
<td>0.063</td>
<td>0.062</td>
<td>0.057</td>
</tr>
<tr>
<td>ECSSD</td>
<td>maxF</td>
<td>0.664</td>
<td>0.694</td>
<td>0.747</td>
<td>-</td>
<td>0.743</td>
<td>0.730</td>
<td>0.757</td>
<td>0.785</td>
<td>0.764</td>
<td>0.781</td>
<td>0.786</td>
<td>0.802</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.150</td>
<td>0.092</td>
<td>0.095</td>
<td>-</td>
<td>0.098</td>
<td>0.120</td>
<td>0.080</td>
<td>0.069</td>
<td>0.072</td>
<td>0.063</td>
<td>0.062</td>
<td>0.057</td>
</tr>
<tr>
<td>DUT-OMRON</td>
<td>maxF</td>
<td>0.775</td>
<td>0.860</td>
<td>0.894</td>
<td>0.892</td>
<td>0.897</td>
<td>0.888</td>
<td>0.904</td>
<td>0.916</td>
<td>0.900</td>
<td>0.916</td>
<td>0.913</td>
<td>0.920</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.146</td>
<td>0.129</td>
<td>0.088</td>
<td>0.052</td>
<td>0.051</td>
<td>0.061</td>
<td>0.049</td>
<td>0.039</td>
<td>0.050</td>
<td>0.040</td>
<td>0.045</td>
<td>0.039</td>
</tr>
<tr>
<td>ICoseg</td>
<td>maxF</td>
<td>0.664</td>
<td>0.694</td>
<td>0.747</td>
<td>-</td>
<td>0.743</td>
<td>0.730</td>
<td>0.757</td>
<td>0.785</td>
<td>0.764</td>
<td>0.781</td>
<td>0.786</td>
<td>0.802</td>
</tr>
<tr>
<td></td>
<td>MAE</td>
<td>0.150</td>
<td>0.092</td>
<td>0.095</td>
<td>-</td>
<td>0.098</td>
<td>0.120</td>
<td>0.080</td>
<td>0.069</td>
<td>0.072</td>
<td>0.063</td>
<td>0.062</td>
<td>0.057</td>
</tr>
</tbody>
</table>
Paper III integrates CNN with the Dense-CRF. The proposed network structure, named as MCRF, is simple and flexible. But the training of CRF parameters is disconnected from the model training of CNN. Thus, the limitation is that we need to manually search for optimal parameter settings for the Dense-CRF layers. On the contrary, the unified CRF proposed by Paper IV supports joint CRF parameter training together with CNN. Thus, the CRF parameters can be optimized through back propagation.

Although the MCRF model in Paper III and the unified CRF model in Paper IV adopt different baseline networks, experimental results based on max F-measure over the ECSSD dataset are evaluated to compare the two models. Firstly, the max F-measure of the baseline encode-decoder network in Paper III is 0.893, while that of the enhanced HED network in Paper IV is 0.884. Despite the settings of the hyper-parameters for the two baseline structures are different, it can be perceived that the former baseline network is with simpler structure and thus is efficient in model finetuning. Finally, with the multi-scale Dense-CRF, the F-measure of the MCRF model climbs to 0.915; while with the multi-scale joint CRF, the F-measure of the unified CRF increases to 0.928. It shows that the joint CRF structure can contribute more in performance improvement.

However, the joint training time for CRF with CNN is rather long in the current implementation. Thus, it will be more economic to integrate the joint CRF if the dataset is not very large, as we use a training set of 2500 images as in Paper IV. In case a large scale training set containing over 10000 images is given, the training of joint CRF will be time consuming and thus perhaps the Dense-CRF will be a trade-off option. For further improvement on joint CRF, an implementation with GPU acceleration in the CRF layer will increase the training efficiency. On the other hand, the current cascade CRF structure yields to a continuous implementation. If we can modify it to discrete implementation, the performance will be further improved.
## Saliency integration

### Introduction

Saliency integration (or saliency aggregation) refers to unifying saliency maps from multiple existing saliency models for more accurate and robust prediction. The introduction of saliency integration is based on the fact that although many of modern saliency models claim high performance in the statistical sense on different public benchmarks, none of them can outperform the others for every image under evaluation [17, 54].

For example, as one of the state-of-the-art approaches, the deep model DHSNet [70] is usually considered to surpass the traditional methods e.g., GP [106], and MB+ [108]. However, there are still images where DHSNet shows inferior predictions to GP and MB+, as shown in Figure 14. Thus, saliency (heat map) integration is proposed to take the advantages of multiple saliency models and make up for the defects of any specific ones, for enhancing accuracy and robustness of saliency detection.

Saliency integration is essentially a weighted combination of saliency maps [142] from multiple saliency models. The weights, assuredly, play a central role in saliency integration. According to different ways of obtaining the weights, existing saliency integration approaches can be categorized into the two types, offline models and online models.

![Fig. 14. Examples where saliency maps from the state-of-the-art deep model show inferior predictions to the traditional models. From left to right there are original images, ground truth (GT), traditional saliency models e.g., MB+ [108] and GP [106], deep saliency model e.g., DHSNet [70], naive integration approach e.g., average map (AVE), and our proposed arbitrator model (AMS-Bound and AML-Bound). Examples are selected from the ECSSD dataset. Prepinted by permission, Paper V © IEEE 2019.](image)
Offline saliency integration models weigh candidate models by optimizing a specific energy function using a collection of data prepared in advance [142, 143, 144]. [142] propose to estimate the weights by minimizing the residual. [143] select a subset of images similar to the input one from a training set and train a CRF aggregation model for saliency aggregation. [144] apply saliency maps from different models as confidence scores and feed them into paraboosting learner for the final saliency map.

Offline model is fixed once the learning phase has been completed. However, they usually require extra efforts in providing the training samples with ground truth labels. Moreover, the scalability is limited [145], as the parameter settings of the integration model are only valid for a particular combination of candidate models. If a new candidate model is added, the integration model has to be retrained. Furthermore, there is an underline assumption that the known samples for learning and the unknown samples for prediction possess similar distributions. If the distribution of the unknown samples is significantly different from those of the known ones, the learned parameters may fail in prediction.

Online integration models [142, 146, 102] are brought forth as a means of addressing the aforementioned problems of the offline models. Online models determine the weights of saliency maps by adapting to the image under evaluation directly, without the demand of any (pre-)collection of known samples. The resulting weights are, therefore, image-specific. [142] propose four functions for weighted summation of multiple saliency maps using uniform, median, M-estimator, and global minimization of an energy function. [146] propose a Naive Bayesian evidence accumulation for saliency integration. [102] further adopt cellular automaton for saliency integration. Compared to the offline models, the online ones are free from fixing a model in advance and thus much more flexible and efficient.

5.2 Challenges

Existing saliency integration models face with two main challenges. First, appropriately estimating the expertise of the candidate saliency models remain the core topic for the integration. Second, the integration approaches should ensure solid performance enhancement.

*How to efficiently estimate the expertise of candidate models?*

Most of the previous works assume that the expertise (a.k.a., weights or contribution) of each candidate saliency model is equal (e.g., BN [146] and MCA [102]).
assumption greatly eases the computational burden. However, it loses sight of the fact that each candidate saliency model shows discrepant ability in predicting an image. In fact, the performance of an integration without consideration of the expertise of candidate models may decrease, as the voices of the superior models are easily drowned out by the mistakes made by those inferior ones. However, it is extremely difficult for online models to weigh each saliency model accurately, since there is no supervised information of the test images.

[17] propose to rank the performances of the saliency models on an image without the ground truth. However, since the ranking is a sequence of ordinal numbers, it cannot numerically measure the performance of each saliency model on the image in details. [142] estimate the expertise of the candidate saliency models by a weight function called M-estimator. The M-estimator decreases the expertise of the outliers that are detected according to their distances to a linear summation of the candidate saliency maps.

However, as shown by the experimental results [142], the M-estimators perform similarly to average weighting, indicating that the computed weights are far from accurately specifying the expertise of the candidate models. Recently, some integration approaches [147, 148] explore expertise estimation by bringing the concept of superpixel difficulty, as each superpixel of an image may possess different difficulty for saliency assessment. This concept of using superpixel difficulty together with model expertise as hidden variables facilitates the expertise estimation process from a more refined superpixel level.

How to ensure solid performance enhancement?

[142] also indicate that saliency integration models may decrease the performance in many cases. For instance, when most candidate saliency models misjudge a region on an image, the integration result will be highly susceptible to error. In Figure 15, we present the integration maps given by four typical online integration models using three popular saliency candidate methods on two images. The red rectangles on the ground truth indicate the regions that the candidate saliency models misjudge. From the integrated saliency maps, it can be perceived that when candidate saliency models misjudge a region on an image, the region will also be misjudged on the integrated map. Thus, overcoming the misleading by most of the candidate saliency models for solid performance enhancement becomes another big challenge in saliency integration.
5.3 Arbitrator model

In this section, Paper V is introduced which proposes an arbitrator model that answers the two challenges faced by existing saliency integration models.

Firstly, the proposed arbitrator model efficiently determines the expertise of each candidate saliency model, in an online manner. To estimate the expertise of the saliency models without ground truth labels, two distinct online model-expertise estimation methods are proposed: one is a statistics-based and the other is latent-variable-based. The two methods measure the expertise of the candidate saliency models without supervised information of the given test image, and meet the requirements of computing rational expertise of the candidate models.

Secondly, the arbitrator model incorporates a mechanism to rectify the misleading by candidate models, even if most of the models misjudge a region on an image. The consensus of multiple saliency models and the external knowledge are incorporated into a reference map to effectively rectify the misleading by candidate models.

Finally, a Bayesian integration framework is utilized to reconcile the saliency models of varying expertise and the reference map for the integrated saliency map.

5.3.1 Bayesian integration framework

Given an image of $N$ superpixels, each superpixel has a unique saliency label $l_n \in \{0, 1\}$. We define the events that the $n$-th superpixel is salient (foreground) and inconspicuous (background) by $F_n$ and $\bar{F}_n$ respectively. Apparently, we have $P(F_n) = P(l_n = 1)$, while $P(\bar{F}_n) = 1 - P(F_n) = P(l_n = 0)$. 

Fig. 15. Examples of misleading caused by misjudgement from candidate saliency models. From left to right columns are original images, ground truth (GT), candidate saliency models including CA [149], IT [10], IS [150], average map (AVE), integrated maps of BN [146], M-estimator [142], MCA [102], and our proposed arbitrator model (AML-Bound). The red rectangles on GT indicate the misjudged regions by the candidate saliency models. Reproduced by permission, Paper V © IEEE 2019.
Suppose there are \( P \) saliency models, each model is able to assign a saliency intensity value \( s_{p,n} \in [0,1] \) to the \( n \)-th superpixel on the \( p \)-th saliency map. The binary saliency label of the \( n \)-th superpixel by the \( p \)-th model, is denoted as \( t_{p,n} \in \{0,1\} \). \( t_{p,n} = 1 \) indicates the \( n \)-th superpixel is considered as a foreground one by the \( p \)-th model and vice versa. It can be easily obtained via a binarization process on the saliency intensity \( s_{p,n} \) with a threshold \( \gamma_p \), e.g., OTSU thresholding [151]. More specifically, we have \( t_{p,n} = 1 (t_{p,n}) \), if \( s_{p,n} \geq \gamma_p \), otherwise, \( t_{p,n} = 0 \). Similarly, \( t_{q\neq p,n} = 1 (t_{q\neq p,n}) \), if \( s_{q\neq p,n} \geq \gamma_p \), otherwise, \( t_{q\neq p,n} = 0 \). Given the intensity of the \( n \)-th superpixel from the \( p \)-th model \( s_{p,n} \) and the \( n \)-th superpixel being labeled as foreground on the binary maps by the rest models \( t_{q\neq p,n} \), the probability that the \( n \)-th superpixel is measured as foreground by the \( p \)-th model is \( P(F_n|s_{p,n}, t_{q\neq p,n}) \).

The probability \( P(F|s_p, t_{q\neq p}) \) is derived under the Bayesian probability framework:

\[
P(F|s_p, t_{q\neq p}) \propto P(F) P(s_p|F) P(t_{q\neq p}|s_p, F) = P(F) P(s_p|F) \prod_{q\neq p} P(t_q|F),
\]

(31)

with the assumption that all \( P \) saliency models make decisions independently, either with respect to the saliency intensity \( s \) or the binary saliency label \( t \). \( s_p \) represents the \( p \)-th saliency intensity map, while \( t_p \) is the \( p \)-th binary saliency map.

The ratio of \( P(F|s_p, t_{q\neq p}) \) is computed as follow:

\[
\Lambda(F|s_p, t_{q\neq p}) = \frac{P(F|s_p, t_{q\neq p})}{P(F|s_p, t_{q\neq p})} = \frac{P(F) P(s_p|F) \prod_{q\neq p} P(t_q|F)}{P(F) P(s_p|F) \prod_{q\neq p} P(t_q|F)}.
\]

(32)

Then the logarithm function of \( \Lambda(F|s_p, t_{q\neq p}) \) is used to form the integration framework, namely the arbitrator model (AM), as follow:

\[
\ln \Lambda(F|s_p, t_{q\neq p}) = \ln \frac{P(F)}{P(F)} + \ln \frac{P(s_p|F)}{P(s_p|F)} + \sum_{q\neq p} \ln \frac{P(t_q|F)}{P(t_q|F)}.
\]

(33)

5.3.2 Cellular automaton

Cellular Automaton (CA) is a discrete model in computability theory and mathematics [152]. A CA consists of a regular grid of cells. Each cell is with states, which are
Fig. 16. Framework of the proposed arbitrator model (AM). The arbitrator incorporates the consensus of multiple saliency maps and the external knowledge into a reference map via the reference generator. A Bayesian integration framework reconciles the reference map and the P saliency maps of varying expertise with cellular automaton (CA), to compute the final result. $\alpha_p$ and $\beta_q$ are the expertise of the $p$-th method and the $q$-th saliency map, respectively. After each generation of the CA, the P saliency maps are updated. Accordingly, the expertise and the reference map are updated based on the new P saliency maps. Preprinted by permission, Paper V © IEEE 2019.

either discrete (e.g., ‘On’ and ‘Off’) or continuous (e.g., between 0 and 1). The neighborhood of one specific cell can influence the states of the specific cell in next generations (advancing $t$ by 1) in line with certain updating rules. Generally, the rule of updating the states of cells is a mathematical function, which is usually synchronous to all cells and time invariants.

The left side of Eq. 33 is the logarithm of the posterior ratio $\Lambda (F | s_p, 1_{q \neq p})$; and thus a logit function of $P (F | s_p, 1_{q \neq p})$. $P (F | s_p, 1_{q \neq p})$ is defined as $s_p^{t+1}$, which stands for the saliency value (of the $n$-th superpixel) on the $p$-th saliency intensity map at time $t + 1$.

There are three terms on the right side of Eq. 33. 1) The first term is defined as $\logit (S_{Ref}^t)$, where $S_{Ref}^t$ represents the saliency reference map at time $t$. 2) The second term is defined as $\ln (\alpha_p^t \cdot s_p^t)$, where $\alpha_p$ is the expertise of the $p$-th method and $s_p^t$ is the $p$-th saliency intensity map at time $t$. 3) Similarly, the third term refers to thresholded binary map $\text{sign}(s_q^t - \gamma_q^t)$ of the $q$-th saliency map with the expertise denoted by $\beta_q$.

At time $t = 0$, the reference map $S_{Ref}^0$ is initialized from a reference generator. At each generation $t > 0$, based on the reference map $S_{Ref}^t$ and the saliency maps ($s_p^t$) of varying expertise $\alpha_p^t$ and $\beta_p$, CA is executed to compute the corresponding $S_{Ref}^{t+1}, s_p^{t+1}, \alpha_p^{t+1}$ and $\beta_p^{t+1}$ at time $t + 1$. The synchronous updating rule of the cellular automaton
derived from Eq. 33 is as follow:

\[
\text{logit}(s_{\text{p}}^{t+1}) = \text{logit}(S_{\text{Ref}}^t) + \ln(\alpha_t^p) \cdot s_{\text{p}}^t + \sum_{q \neq p} \ln(\beta_{t}^q) \cdot \text{sign}(s_{\text{q}}^t - \gamma_{t}^q),
\] (34)

\[
S_{\text{Ref}}^t = \frac{1}{p} \sum_{p=1}^{P} s_{\text{p}}^t,
\] (35)

\[
S_{\text{Final}}^t = \frac{1}{p} \sum_{p=1}^{P} s_{\text{T}}^p.
\] (36)

5.3.3 Reference generator

To acquire the reference map, an external saliency map with external knowledge is firstly computed. Then, a consensus map is obtained by the consistency of the candidate models and the external knowledge map. Finally, the consensus map is propagated for the reference map.

External knowledge map

The external knowledge map is expected to rectify the errors by the candidate models. Basically, the external knowledge can be any reasonable assumptions about salient object detection or currently existing saliency models.

In Paper V, three methods are used to compute the external knowledge map. The first one is a handy and fast method based on the widely accepted assumptions, such as boundary prior [55]. The second is the saliency map from one of the state-of-the-art traditional saliency models such as CCM [57]. The third one is the saliency map from one of the state-of-the-art deep models such as DHSNet model [70].

Consensus map

Even though an external knowledge map is introduced, its accuracy in saliency detection can not be guaranteed just as the uncertainty in the candidate saliency models. Thus, a strict consistency scheme is introduced to reach a prudent consensus. The arbitrator model judges the superpixel as salient only if the majority of the candidate saliency models vote it as salient as well as the external knowledge confirms its saliency.

Given \( P \) saliency maps, \( S_p(n) \) is defined as the mean intensity value of the \( n \)-th superpixel on the \( p \)-th saliency map. The majority voting map is computed as
\[ S_{\text{Maj}}(n) = \begin{cases} 1, & \sum_{p=1}^{P} \lambda_{p,n} > \frac{P}{2} \\ 0, & \text{otherwise.} \end{cases} \] (37)

A consensus map \( S_{\text{Con}} \) is computed by hearing voices from both the majority voting map and the external knowledge map:

\[ S_{\text{Con}} = S_{\text{Ext}} \times S_{\text{Maj}}. \] (38)

Reference map via propagation

The consensus map \( S_{\text{Con}} \) is a saliency map of high precision but only holds saliency information for certain parts of the image, and thus a single directional propagation is performed to expand the saliency information to the whole image, to obtain \( S_0^{\text{Ref}} \).

5.3.4 Model expertise estimator

Two expertise estimation methods are proposed by the arbitrator model, a statistical approach and a latent-variable-based approach.

Statistics-based Expertise

The statistics-based method analyzes the probability distributions of foreground and background samples on saliency maps and statistically computes \( \alpha_p \) and \( \beta_p \). 

\( \beta_p \) is the expertise of the \( p \)-th binary saliency map, which is originally derived from 
\[ \frac{P(t_p|F)}{P(t_p|\bar{F})}. \] More specifically, \( P(t_p|F) \) is \( P(t_p, n = 1|F_n) \), indicating the probability that the \( n \)-th superpixel on the \( p \)-th saliency map is labeled as foreground given the superpixel is a foreground one. Similarly, \( P(t_p|\bar{F}) \) is \( P(t_p, n = 1|\bar{F}_n) \), indicating the probability that the \( n \)-th superpixel is miss-labeled as foreground given the superpixel is a background one. Although it is impossible to get the ground-truth \( F \) in online methods, the reference map can be regarded as the ‘best current’ knowledge to approximate \( F \). Thus, \( \beta_p \) is computed as follows:

\[ \beta_p = \frac{P(t_p|F)}{P(t_p|\bar{F})} = \frac{P(t_p, n = 1|F_n)}{P(t_p, n = 1|\bar{F}_n)} \propto \frac{P(t_p = 1|F)}{P(t_p = 1|\bar{F})}. \] (39)

While \( \alpha_p \) represents the expertise of the \( p \)-th saliency intensity map, which can be computed in a similar way as computing \( \beta_p \).
Latent-variable-based expertise

Besides the expertise, each superpixel on the image is assumed to possess difficulty for saliency assessment, namely $\pi_n$. The expertise $\beta_p$ as well as the difficulty of the superpixel $\pi_n$ are assumed as latent variables and are solved by optimizations.

$\beta_p$ is assumed to range $\beta_p \in (-\infty, +\infty)$. If $\beta_p < 0$, the $p$-th candidate model makes wrong measurements and shows inferior ability in saliency detection. If $\beta_p > 0$, the $p$-th model makes correct measurements and shows superior ability in saliency detection. When $\beta_p = 0$, the $p$-th model is not able to distinguish saliency objects. $\beta_p = +\infty$ implicates that the $p$-th model always makes correct decisions about saliency objects, while $\beta_p = -\infty$ means that the $p$-th binary saliency map always misjudge saliency information. Besides, $\pi_n \in [0, +\infty)$ represents the difficulty of a superpixel. $\pi_n = 0$ means that the superpixel possesses extremely low difficulty such that even an inexperienced saliency model can distinguish its saliency. On the contrary, $\pi_n = +\infty$ means the superpixel is so ambiguous that even the best saliency model has a chance to misjudge it.

$l_n$ is the true binary saliency label of the $n$-th superpixel on the given image, while $\iota_{p,n}$ refers to the actual binary saliency label of the $n$-th superpixel by the $p$-th model. Thus, the probability that the $p$-th model correctly labels a superpixel on an image is

$$p(\iota_{p,n} = l_n | \beta_p, \pi_n) = \begin{cases} 1, & \pi_n = 0 \\ \frac{1}{1 + e^{-\beta_p \pi_n}}, & \text{otherwise.} \end{cases}$$

(41)

More skilled saliency models (higher $\beta_p$) have a higher probability of correctly labeling a superpixel. As the difficulty $\pi_n$ of a superpixel increases, the probability of correctly labeling the superpixel decreases, and vice versa.

The Expectation-Maximization (EM) algorithm is used to achieve the optimal values of the latent parameters. In the E-step, the posterior probabilities of $l_n$ with the parameters $\beta, \pi$ obtained from the last M-step and the actual labels are computed as:
\[ p(l_n|t, \beta, \pi) = p(l_n|t_n, \beta, \pi_n) \]
\[ \propto p(l_n|\beta, \pi_n)p(t_n|l_n, \beta, \pi_n) \]
\[ \propto p(l_n) \prod_p (t_{p,n}|l_n, \beta_p, \pi_n), \]

where \( t_n \) denotes the actual labels of a superpixel by all the \( P \) candidate models and the parameters \( \beta, \pi \) are conditionally independent of \( l_n \). In practice, we use Gaussian distribution \((\mu = \theta = 1)\) for \( \beta \), re-sample \( 1/\pi \) as \( e^{(1/\pi')} \), and use the same Gaussian distribution on \( 1/\pi' \) to avoid \( \pi \) being negative.

The M-step computes the expected value of the log likelihood function with respect to the conditional distribution of \( l \) given \( t \) under the current estimate of \( \beta \) and \( \pi \) as follows:

\[ Q(\beta, \pi) = E[\ln p(t,l|\beta, \pi)] \]
\[ = E \left[ \ln \prod_n \left( p(l_n) \prod_p p(t_{p,n}|l_n, \beta_p, \pi_n) \right) \right] \]
\[ = \sum_n E[\ln p(l_n)] + \sum_{p,n} E[\ln p(t_{p,n}|l_n, \beta_p, \pi_n)], \]

where \( t_{p,n} \) are conditionally independent given \( l, \beta, \pi \).

With gradient ascent method, the parameters \( \beta \) and \( \pi \) are set to maximize the quantity function \( Q \) in Eq. 43.

### 5.4 Experiments

Experiments are conducted on a pool of twenty-seven candidate saliency models, including BSCA [102], CA [149], CEOS [153], COV [50], DRFI [103], FT [46], GBVS [104], GC [105], GP [106], HS [98], IS [150], IT [10], LR [107], MB [108], MB+ [108], MR [95], PCAS [109], RB [55], RC [105], SR [45], TLLT [96], UFO [103], DSS [72], DCL [112], RFCN [113], MDF [137], and DHSNet [70].

For clarity, the arbitrator model with statistics-based expertise is termed as AMS and with latent-variable-based expertise as AML. Moreover, the experiments refer “-B”, “-C”, “-D” as the boundary-based external knowledge, contour-closure-based external knowledge and deep-based external knowledge respectively. In general, the arbitrator model outperforms the existing integration models with all the following four combination strategies as in Figure 17, on ECSSD dataset [98].
Fig. 17. Mean F-measure of the average saliency maps (AVE), the resulted BN, M-estimator (M-est), and MCA saliency maps and the resulted AMS and AML saliency maps. The subscripts “B”, “C” and “D” represent the boundary-based reference map, contour-based reference map and deep-network-based reference map respectively. The first column shows the combination strategy, and for every combination the highest F-measure of the candidate saliency models are displayed in the “Top” column. The best result for each combination is in bold with dark background color, the second best is in bold, and the third best is underlined. Deep candidate models are underlined. Preprinted by permission, Paper V © IEEE 2019.
**Superior models combination.** Only saliency models with the best performances are chosen for integration. Thus, two best saliency models are selected for 2-model-combination, three best saliency models for 3-model-combination and so forth. Both AMS and AML outperform the top candidate saliency model as well as existing integration model in every combination.

**Inferior models combination.** Only saliency models with the worst performances are chosen for integration. For example, the worst two saliency models are chosen for 2-model-combination, the worst three saliency models for 3-model-combination and so forth. The AM model largely improves the F-measure of the top candidate saliency model with an average increase of 6.6%, 11.0%, 17.8%, 7.4%, 12.7%, and 20.9% for AMS-B, AMS-C, AMS-D, AML-B, AML-C, and AML-D correspondingly, while other online integration models such as AVE, BN, MCA and M-est decrease the F-measure by averagely 3.6%, 3.6%, 1.8% and 3.6% respectively.

**Random combination.** From 2-model combination to 8-model combination, candidate saliency models are randomly selected from the model pool and five different combinations are randomly evaluated for each fixed number combination. The proposed model solidly improves the performance regardless of the number of models being chosen for combination.

**Deep models combination.** Deep models are chosen including DSS, DCL, RFCN, MDF, and DHSNet for evaluation. The AMS-D and AML-D with deep external knowledge maps surpass the top saliency models averagely by 0.9% and 1.4% respectively. In the last four rows in Figure 17, the F-measure of the AVE, BN, MCA and M-est integration models drop sharply compared to the top models by averagely 5.3%, 5.2%, 9.7%, and 4.6% respectively, while AMS-D and AML-D averagely increase by 0.3% and 0.2% respectively.

Meanwhile, the AM model is evaluated on four challenging datasets, ECSSD [98], ASD [46], ImgSal [154] and DUT-OMRON [95], with the combination of MB, BSCA, RC, GBVS, COV and FT as an example. Figure 18 presents the average MAE and F-measure of the candidate models being combined, the average saliency maps (AVE), results from BN, M-est and MCA model, and results from the AM model on the four datasets.

### 5.5 Summary

In this chapter, the concept of saliency integration is introduced, which is a relatively new topic with fewer related works than the number of saliency models. Existing saliency integration models face with two main challenges. One challenge is to appropriately
estimating the expertise of the candidate saliency models, especially when the ground truth labels are not available. The other challenge is to ensure solid performance enhancement even most of the saliency models misjudge a region on the image. Thus, Paper V proposes an arbitrator model that answers the two challenges.

Firstly, the arbitrator model efficiently determines the expertise of each candidate saliency model with two distinct online methods: statistics-based approach and latent-variable-based approach. Secondly, the arbitrator model incorporates the consensus of multiple saliency models and the external knowledge to rectify the misleading by candidate models. Finally, a Bayesian integration framework is utilized to reconcile the saliency models of varying expertise and the reference map for the integrated saliency map.
Reviewing this work, we believe that the proposed Bayesian based framework not only contributes to essential performance enhancement, but also shows flexibility. But the selection of the external knowledge in generating the reference map is important.

For the integration of saliency maps from traditional saliency models, we suggest to choose the external knowledge based on common accepted and effective assumptions or resulted saliency maps from the state-of-the-art saliency methods.

For the combination of saliency maps from deep saliency models, the external knowledge should be carefully selected. For instance, the DHSNet model [70] adopted in Paper V shows superior performance and thus is chosen to be the external knowledge. The proposed deep unified CRF saliency model in Paper IV results in better performance than DHSNet, and can be an even better option as the external knowledge. But if the candidate saliency models contain the DSS [72] model, we believe that the enhancement from the deep unified CRF saliency model as external knowledge may be limited. The reasons are two. First, the deep unified CRF saliency model is only slightly better than DSS model as reported in Table 5. Second, the two models are trained with the same training protocol using the same dataset, which means that they are likely to mislabel the same regions on one image. Thus, the rectification may lose efficacy.

In general, the arbitrator model contributes to more efficient performance enhancement on combination of unsupervised saliency models than that of deep saliency models.

Meanwhile, the proposed integration framework is implemented by the cellular automaton, of which the state of a cell is only affected by the superpixels at the same location of all the saliency maps. Further studies may also explore the influences of the adjacent superpixels of a cell. Also, the reference map and the expertise estimation of the AM framework can also be applied to co-saliency detection tasks.
6 Summary

6.1 Contributions

Visual attention analysis concentrates on understanding visually salient cues and knowledge guided by the human perception mechanism. My Ph.D. research works on visual attention analysis follow the road map from the early stage eye movement data collection to building computational models, known as saliency models, for saliency detection.

Eye movement data is one of the most essential elements to interpret human visual attention, and thus is collected to facilitate computational models for visual attention analysis. Chapter II outlines the basic concepts about eye movements, eye tracking data collection, and current eye tracking datasets. As most of the existing eye tracking datasets are task-free and the images are with few semantic categories, Paper I presents a new task-driven eye tracking dataset. Firstly, most of the watching materials are designed with specific tasks, which reduces the center bias and distractions from the observers. Secondly, the image semantics covers several computer vision related topics. Further, a baseline by evaluating thirteen saliency models is provided. The new task-driven eye tracking dataset can facilitate fixation prediction models.

Then, my research works mostly focus on building computational models for visual attention analysis. In particular, I explore approaches for salient object detection and publish three papers on this topic. Paper II is my first work on salient object detection, which focuses on unsupervised methods. It presents a bi-directional propagation method based on graphical model, which performs foreground propagation and background propagation simultaneously with difficulty-based rule for computation efficiency. The bi-directional propagation model receives state-of-the-art performance among unsupervised saliency models.

Then, as the prevalence of CNNs, I move to explore deep saliency models. Paper III proposes a compact model that extracts multi-scale side output maps from CNNs and embeds the Dense-CRF layer at each scale for post refinement. The proposed model is based on a simple encoder-decoder network and receives comparable performances to existing models with complex network architectures. Although the Dense-CRF is rather effective in saliency refinement, it is disconnected from the training of the CNNs. Motivated by this, I explore to integrate CRF formulation into CNNs for joint end-to-end training. Thus, Paper IV proposes a deep cascade CRFs architecture that is seamlessly incorporated with CNNs to integrate and refine multi-scale deep features.
and predictions for a refined saliency map. At each scale, a CRF block is embedded to facilitate feature-feature, feature-prediction and prediction-prediction messages passing. The new CRF formulation provides explainable solutions for the features, the predictions and the interactions among them, leading to distinct model formulation, inference, and neural network implementation.

Further, as there exist numerous saliency models but none of them can outperform the others for every image under evaluation, I also investigate saliency integration methods to unify saliency maps from multiple existing saliency models for better prediction. Paper V proposes the arbitrator model which incorporates a mechanism to rectify the misleading by candidate models via the consensus of multiple saliency models and the external knowledge. Also, two online methods of estimating the expertise of multiple saliency models are proposed. To our best knowledge, Paper V conducts the largest scale of experimental analysis on saliency integration with a model pool of twenty-seven saliency models, covering both traditional and deep learning ones, on various combinations over four datasets.

6.2 Applications and extensions

Computational models for visual attention analysis, or saliency models, are developed to reflect human visual behaviors. Besides its psychological and psychological implications, saliency models are applicable in many related areas towards computer vision.

Compression, either for images or videos, aims at reducing the cost of storage or transmission by preventing the corruption of visually important elements and curtailing redundant information. Saliency models play an active role in preserve the visually important regions and thus are adopted for compression tasks. [13] propose a multi-resolution spatio-temporal saliency detection model as the pre-processing step for image and video compression. [155] firstly compute a saliency map representing salient objects and then incorporate saliency values and motion vectors into entropy encoder for video compression. [156] propose a saliency-aware video compression model to reduce the salient coding artifacts in non-ROI regions and keep user attention on ROIs. Further, [157] propose deep visual attention networks and embed saliency detection module to encoding module to improve video compression.

Video summarization refers to selecting informative video frames in order to produce a summary in large video archive management. The summary of a video depends on the frames of interest to the users. As one of the solutions, saliency detection can be applied to key frames selection. [15] propose to extract key frames based on an attention curve from saliency information.
Medical image processing is one of the important research areas in computer vision field, and the subjects are diverse, such as classification, detection and segmentation tasks for healthcare treatment or diagnosis purposes. [158] suggest that not all the regions on the medical image are equally significant to the image retrieval application. For example, in lung x-ray images, the lung region may contain a tumor, and thus is salient compared to the surrounding areas without significant information for medical diagnosis. Thus, they propose to classify medical images from saliency-based folded data. [159] apply salient point detector for medical image retrieval. [160] are inspired by the fact that saliency is biologically motivated and thus propose a visual saliency based bright lesion detection method on retinal images.

Image processing for aesthetic purposes such as image style transfer or photographic image processing may also adopt saliency information. For instance, image style transfer approaches take advantage of salient regions to keep informative regions for the style transfer step [161, 162]. [163] firstly compute a saliency segmentation map and then compute the depth of field effect based on the saliency map.

Affective computing takes advantage of salient regions on faces as significant features for facial expression analysis. [164] propose a saliency-based framework to adopt saliency maps to evaluate appropriate weights for the handcrafted features on images for facial expression recognition. [165] learn visual saliency maps from deep multi-layer networks and feed the saliency detection into CNN stream for facial expression recognition.

Meanwhile, visual saliency starts from detecting salient regions or objects on 2D visual scenes. Later, additional information is added to complement the RGB color features. For instance, RGBD saliency refers to saliency detection based on RGB color as well as depth features on visual scenes. RGBD saliency introduces the extra depth information for saliency detection. In addition, temporal information can also be introduced for 3D visual saliency, i.e., video saliency. Videos contain temporal information which provides sequential and motion features. The salient objects in videos are repetitive, motion-related, and distinct targets. The videos can be daily scenes from a third view or even from the first person view. In particular, as the emergence of wearable eye trackers, videos recorded from the first person view, namely egocentric videos boost visual saliency analysis from new perspectives. Meanwhile, saliency is also extended to apply inter-image correspondence constraints to discover common salient objects in an image group, namely co-saliency. Co-saliency is an extended topic from saliency detection, which refers to the discovery of common and salient objects from the image group including multiple related images. Co-saliency detection approaches find out the intra-image similarity to ensure the detected regions are salient in single image, and the
inter-image similarity as the common constraint to search common objects from all the salient ones.

In fact, the applications of saliency, including its related extended topics, are not limited to these several areas. The concept of saliency is also important in robotics, human computer interaction, gaming and so forth. Therefore, visual saliency is essential in computer science and engineering.

6.3 Future work

This thesis thoroughly summarizes the main research works in my Ph.D. studies on visual attention analysis. First, data preparation for computational modeling is explored. Second, computational models for visual attention analysis are investigated from traditional unsupervised methods to deep saliency models. Third, a saliency integration model is proposed.

From the perspective of personal research improvement, there are several issues can be further explored. Firstly, although Paper I established a task-driven eye tracking dataset with various semantic categories, the image numbers for each category are limited such that learning-based approaches are difficult to be trained on the dataset. Thus, a semantic-specific and task-driven eye tracking dataset containing adequate images might be a direction for improving the usability of the data. Secondly, the deep unified CRF model in Paper IV can be improved in implementation for further performance improvement. On one hand, an implementation with GPU acceleration in the CRF layer will increase the training efficiency. On the other hand, the current cascade CRF structure yields a continuous implementation. If we can modify it to discrete implementation, the performance will be further improved.

Visual attention analysis, in the long run, arouses our concerns in the following research aspects.

Visual attention analysis needs eye tracking data of better quality. Firstly, the improvement of data quality relies on not only accurate eye tracking devices but also the amount of data. Current eye trackers depend on strict experimental environments and the data collection process can be rather time consuming. Thus, it is substantial to explore more fixation approximation methods, such as Salicon [53] using a mouse-contingent and multi-resolutional paradigm for crowd sourcing data collection. Secondly, visual searching tasks are important in reducing center bias and filtering out irrelevant collapsing raw fixations, and thus should be specifically designed. Thirdly, existing salient object detection datasets, besides the limited image number, are also restricted by the resolution of images. Although low-resolution images do not affect the detection of
the locations of the salient objects, it may prohibit accurate boundary divisions of the objects. Thus, high-resolutional and large-scale salient object detection datasets are in demand.

Computational modeling for visual attention analysis can be further explored. Firstly, a number of video saliency models [166, 167, 168] are proposed in recent years benefiting from several published video saliency datasets, such as UVSD [169], VOS [170] and DAVIS [171]. We may investigate on efficient embedding of temporal information into deep neural networks for more accurate video saliency models. Secondly, salient fixation prediction can also be explored in time sequences to predict the scanpath on an image. Yet, saliency modeling has been widely studied, but scanpath related topics including generation and comparison remain new in computer vision fields. Thus, the public scanpath datasets are needed to facilitate scanpath prediction, evaluation, and analysis.

Further, we should also concern the extension of saliency models to other related areas to facilitate computer vision tasks, such as facial expression recognition, depth of field analysis, medical image processing and so forth.

My future career plan is to apply my previous research experiences on salient object detection and deep neural networks to some related topics such as semantic segmentation, depth estimation, and object detection, for solving practical issues. Moreover, I would like to apply visual attention analysis to facilitate some practical topics. For instance, in medical image segmentation tasks such as brain tumor detection on magnetic resonance images (MRI), saliency detection can be a pre-processing step or assumption in detecting the brain tumor regions. I would keep tracking with research outcomes in the computer vision field and apply them to industry works.
References


Original publications


Reprinted with permission from Springer (I, III), Elsevier (II), and IEEE (IV, V).

Original publications are not included in the electronic version of the dissertation.
738. Wijewardhana, Uditha (2020) Sparse recovery algorithms for streaming and multidimensional signals
739. Li, Panpan (2020) Deep eutectic solvents in the production of wood-based nanomaterials
740. Nuortimo, Kalle (2020) Public acceptance in energy technology development and deployment: opinion mining case study
742. Mäkynäki, Pekka (2020) Chemical precipitation in the simultaneous removal of NH$_4^+$-N and PO$_4^{3-}$-P from wastewaters using industrial waste materials
744. Frondelius, Tero (2020) Development of methods in engine design process
745. Särstöniemi, Mariella (2020) UWB in-body propagation and radio channel characteristics for wireless body area network applications
746. Ashraf, Faisal Bin (2020) River regimes and energy demand interactions in Nordic rivers
747. Kuula, Seppo (2020) Continuous co-creation of knowledge-intensive business services
748. Mustonen, Erno (2020) Vertical productisation over product lifecycle: co-marketing through a joint commercial product portfolio
749. Halttula, Heikki (2020) Enhancing data utilisation in the construction project lifecycle through early involvement and integration
750. Nguyen, Hoang (2020) Ettringite-based binder from ladle slag: From hydration to its fiber-reinforced composites
752. Xu, Yingyue (2020) Computational modeling for visual attention analysis

Book orders:
Virtual book store
http://verkkokauppa.juvenesprint.fi
Yingyue Xu

Computational Modeling for Visual Attention Analysis

University of Oulu Graduate School, University of Oulu, Faculty of Information Technology and Electrical Engineering