Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

**By reading and using the thesis, the reader understands and agrees to the following terms:**

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.

2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.

3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

---

**IMPORTANT**

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.
DEEP HIERARCHICAL ARCHITECTURES FOR SALIENCY PREDICTION AND SALIENT OBJECT DETECTION

YU HU

M.Phil

The Hong Kong Polytechnic University

2016
The Hong Kong Polytechnic University

Department of Electronic and Information Engineering

Deep Hierarchical Architectures for Saliency Prediction and Salient Object Detection

Yu Hu

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Philosophy

February 2016
Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

______________________________ (Signed)

Yu Hu (Name of student)
Abstract

This thesis presents hierarchical deep architectures for saliency prediction and object detection in natural scenes. Three major contributions are reported in the thesis: (1) a deep architecture based on a Convolutional Neural Network (CNN) for saliency prediction, (2) a hybrid pixel-based and segmentation-based approach for salient object detection, and (3) learning the heat maps of human eye gaze data using a CNN-based model.

In the first investigation, an Adaptive Saliency Model (ASM) based on CNN for saliency prediction is proposed. The model consists of convolutional layers and subsampling layers and a Two-Dimensional Output (TDO) layer which performs the prediction of image saliency. The kernels in the CNN perform feature extraction and the TDO aims to generate a saliency map. Two levels of kernels are utilized in two convolutional layers in ASM. The first-level kernels are used to learn low-level features from an input image while the second-level kernels aim at capturing high-level features. In this thesis, an approach of training ASM to generate a saliency map with only the original image as the input is studied. In my study, I explore the Long-Term-Dependency (LTD) problem during training ASM using a Gradient Descent (GD) Back-Propagation (BP) algorithm. Resilient Propagation (Rprop) has shown a superior performance in training ASM. Various aspects including the saliency prediction performance, the training time and the number of kernels used in convolutional layers are systematically studied. Experimental results on the Object
and Semantic Images and Eye-tracking (OSIE) dataset demonstrate the effectiveness of my proposed ASM compared with state-of-the-art algorithms for saliency prediction based on the Histogram Intersection (HI) metric.

In the second investigation, I propose a hybrid Salient Object Detection (SOD) model that consists of the modified ASM and the potential Region-Of-Interest (p-ROI) approximation. Different from the ASM used in first investigation in which the ground truth of continuous saliency values is required to train the model, the ASM used in this investigation needs the binary ground truth only to detect salient objects. Specifically, the ASM aims to assign pixels in the input image with saliency values and p-ROI is used to validate the saliency region with a segmentation approach. Both ASM and PROI contribute to the improvement of object detection performance. ASM is used to refine performance of p-ROI by targeting at details, while p-ROI is to enhance the capability of ASM by exploring on the entire input image. The metrics including precision and recall curve and Area Under Curve (AUC) are adopted to evaluate the performance of my approach of SOD. Experimental results on a dataset with manually demarcated ground truth demonstrate a superior performance of the hybrid SOD model comparing with each individual method.

In the third investigation, ASM is utilized to learn the heat maps of human eye gaze data. I first employ ASM with the Rprop algorithm to generate heat maps and show that the deep learning method can only achieve a moderate performance. Then I modify the approach to have the deep neural network pre-trained on Itti saliency maps
and show that this pre-training process can slightly improve the performance. The metrics including precision and recall curve, Receiver Operating Characteristic (ROC) and AUC are adopted to evaluate the performance of my leaning model on both the OSIE dataset and the CAT2000 dataset.
List of Publications

Conference Papers:


Acknowledgements

I would like to extend my gratitude and acknowledgements to a number of individuals who, directly or indirectly, contributed to the successful completion of my study and this thesis.

First of all I would like to express my most sincere thanks and gratitude to Dr Zheru Chi, my supervisor, for his outstanding and constructive guidance, and unrelenting support in my MPhil study and research. Whenever I arrived at his office, he would put aside what he was working on and would eagerly discuss with me about my research. He also granted me the freedom of choosing a research topic that interested me, guided me throughout the whole study period. Moreover, he helped me enhancing the mathematical derivations and my English writing skills with great patience. He further helped me revising every academic paper I wrote with his invaluable advices and comments.

Next, I wish to sincerely thank Dr. Hong Fu, for her helpful suggestions and generous assistance. I deeply appreciate her valuable help and support in my research, and study. I had been furnished with generous advices and comments, whenever we discussed about my study and research.

I would also like to thank Dr. Zenghai Chen, for sharing his experience on research methods and views on research directions. In addition, thanks are expressed to Mr. Junkai Chen, Dr. Zhen Liang, Ms. Hui Zhang, Miss Ziqian Zhou, my friends, for their selfless assistance, encouraging and decorating of my MPhil study with joy.
and friendship.

Last but not least, I am deeply indebted to my parents. I can hardly believe that I could have overcome all the hard times during these years without the lasting encouragement and support from them.
# Table of Contents

Certificate of Originality .................................................................i

Abstract............................................................................................ii

List of Publications.............................................................................v

Acknowledgements.............................................................................vi

Table of Contents...............................................................................viii

List of Figures....................................................................................xii

List of Tables.......................................................................................xvi

List of Abbreviations...........................................................................xvii

Chapter 1  Introduction ......................................................................1

1.1 Motivation ..................................................................................1

1.2 Statements of Originality ..........................................................4

1.3 Outline of the Thesis .................................................................6

Chapter 2  Literature Review............................................................9

2.1 Basic Definitions ........................................................................9

2.1.1 Visual Attention ....................................................................9

2.1.2 Computational Saliency Models ..........................................10

2.2 Non-learning Based Saliency Model .........................................11

2.2.1 Bottom-Up Approaches ......................................................11

2.2.2 Top-Down Approaches ......................................................14

2.2.3 Mixture of Bottom-UP and Top-Down Approaches ............15
2.2.4 Itti’s Saliency Model ................................................................. 16

2.3 Learning Based Saliency Model ..................................................... 21

2.3.1 Feature Extractor Learning ....................................................... 21

2.3.2 Classifier Learning .................................................................. 22

2.3.3 Mixture of Feature Extractor and Classifier Learning ................. 23

2.4 Deep Learning and Feature Learning ............................................. 25

2.4.1 Convolutional Neural Network .................................................. 26

2.5 Eye Tracking Techniques and Applications .................................... 29

2.5.1 Eye Structure Analysis .............................................................. 29

2.5.2 Eye Tracking Techniques ......................................................... 32

2.5.3 Eye Tracking Applications ....................................................... 33

Chapter 3 Adaptive Saliency Model for Saliency Prediction .................... 38

3.1 Background.................................................................................. 38

3.2 Adaptive Saliency Model (ASM) .................................................. 42

3.2.1 Preprocessing ........................................................................ 42

3.2.2 Neural Layers ........................................................................ 45

3.2.3 Post processing....................................................................... 49

3.3 Algorithm for Training ASM Deep Structure .................................. 51

3.3.1 Resilient propagation (Rprop) ................................................. 51

3.4 Experimental Results and Discussion .......................................... 53

3.4.1 Datasets and Experiment Setup .............................................. 53
Chapter 4 Deep Learning for Salient Object Detection........................................... 73

4.1 Background.............................................................................................. 73

4.2 Model Structure....................................................................................... 76

4.2.1 ASM Component .................................................................................. 77

4.2.2 Potential Region of Interest (p-ROI).................................................... 82

4.3 Experimental Results and Discussion.................................................... 85

4.3.1 Datasets and Experiments Setup......................................................... 85

4.3.2 Metrics for Performance Evaluation..................................................... 85

4.3.3 Output of Adaptive Saliency Model...................................................... 86

4.3.4 Combination of Adaptive Saliency Model with p-ROI ....................... 87

4.4 Summary.................................................................................................... 96

Chapter 5 Deep Learning of Heat Maps of Human Eye Gaze Data................. 97

5.1 Background.............................................................................................. 97

5.2 Deep Learning for Learning Heat Maps of Human Eye-Gaze............... 99


List of Figures

Figure 2-1. Procedures of a non-learning based approach (left) and a learning-based approach (right) for generating saliency maps. .................. 11

Figure 2-2. Four chessmen in red, yellow, green and blue with a pure background. ........................................................................................................... 14

Figure 2-3. The procedure to obtain Itti’s saliency map of an image (Itti et al., 1998). ........................................................................................................ 16

Figure 2-4. Example of cross-scale summation. ........................................... 19

Figure 2-5. The cross section of the human eye (adopted from (Duchowski, 2007)). ........................................................................................................ 30

Figure 2-6. The first heat map (adopted from (Sneath, 1957)). ...................... 36

Figure 2-7. Color heat map of fixation captured by observing a short paragraph.

(Adopted from (Outing and Ruel, 2004)) ...................................................... 37

Figure 3-1. The framework of my proposed CNN-based adaptive saliency model. ........................................................................................................ 41

Figure 3-2. Example of down-sampling without changing data size ............... 43

Figure 3-3. Examples of down-sampled images at different scales ............... 44

Figure 3-4. The connections between input layer I1 and convolutional layer C2. ........................................................................................................ 46

Figure 3-5. An input image (a) and examples of the feature maps in C2 layer

(b)-(e). ........................................................................................................ 47
Figure 3-6. Examples of the feature maps in C4 layer.......................... 48
Figure 3-7. Plot of the sigmoid function........................................... 50
Figure 3-8. Dataflow in convolution during FW and BP...................... 55
Figure 3-9. Simplified diagrams of three possible ASM structures......... 57
Figure 3-10. The individually connected output layer in the second structure. 58
Figure 3-11. The error curves of six training processes.......................... 61
Figure 3-12. Visualized outputs from six training processes together with the original image and the target output................................................. 62
Figure 3-13. ROC curve on test samples............................................ 66
Figure 3-14. Histogram intersections on training samples (a) and test samples (b). ........................................................................................................ 66
Figure 3-15. A comparison of the saliency maps generated by my model with Itti saliency maps on the OSIE data set........................................ 69
Figure 4-1. The framework of the convolutional neural network based component of my proposed salient object detection model................................. 75
Figure 4-2. Block diagram of my proposed hybrid salient object detection model. ........................................................................................................ 77
Figure 4-3. Bilinear Interpolation used in up-sampling.......................... 78
Figure 4-4. Example of data down-sampling then up-sampling back to its original size................................................................................................ 79
Figure 4-5. Examples of downsampled images at different scales............ 80
Figure 4-6. A comparisons between down-sampling methods introduced in Chapter 3 and Chapter 4. (a) Use the method introduced in Chapter 3 and (b) use the method introduced in Chapter 4. ......................................................... 81

Figure 4-7. Comparison among (a) the original image, (b) detected edge map, (c) edge map after filtering, (d) the final mask generated from filtered edge map, (e) masked image, and (f) the salient object circled with green line........ 84

Figure 4-8. A comparison among (a) the original image, (b) the saliency map generated by ASM component of my model, (c) the ground truth, (d) the salient region generated with p-ROI, and (e) the final result of my model. ...................................................................................................................... 87

Figure 4-9. A comparison among (a) the original image, (b) the saliency map generated by the ASM, (c) the ground truth, (d) the salient region generated by p-ROI, and (e) the final salient object detected from my combined model. ............................................................................................................. 88

Figure 4-10. A performance comparison between my model and other methods tested. .......................................................................................................................... 90

Figure 4-11. Saliency maps generated from my model and several other models. (a) the original image, (b) the ground truth, (c) ASM, (d) my hybrid model, (e) P-ROI, (f) AC, (g) AIM, (h) CA, (i) IM, (j) IT, (k) SeR, (l) SR, (m) SUN and (n) SWD. ......................................................................................................................... 95

Figure 5-1. Block diagram of ASM based model for learning heat maps of
human eye gaze. ................................................................. 99

Figure 5-2. An example of constructed heat map of human eye gaze and its original image ................................................................. 101

Figure 5-3. Training error (a) and ROCs of the training set (blue) and the test set (red) (b). ................................................................. 107

Figure 5-4. Example of the original image, the ground truth and the output of ASM ................................................................. 107

Figure 5-5. Training error (a) and ROCs of the training set (blue) and the test set (red) (b) with random initial weights. ............................................. 108

Figure 5-6. Comparison on the original image, the ground truth and the output of ASM ................................................................. 108

Figure 5-7. ROCs of the training set (blue) and the test set (red). ................. 109

Figure 5-8. Comparison on the original image, the ground truth and the output after distribution processing for the model with random initial weights. 110

Figure 5-9. ROCs of the training set (blue) and the test set (red) with pre-trained weights. ................................................................. 111

Figure 5-10. Comparison on the original image, the ground truth and the output after distribution processing for the model with pre-trained weights. ..... 112

Figure 5-11. Examples of the original image, the ground truth and the output after distribution processing from the CAT2000 dataset. .................. 114
List of Tables

Table 3-1. The parameter setting of the first structure. ........................................ 58
Table 3-2. The parameter setting of the second structure. ................................. 59
Table 3-3. The parameter setting of the third structure. .................................... 59
Table 3-4. Six trainings of three models and two training algorithms. ............... 60
Table 3-5. The average output HI’s with different number of kernels in C2 as well
as C4 and the number of Channels in I1 (without using Rprop). .................... 64
Table 3-6. The average HI’s with different structure parameter settings .......... 68
Table 3-7. Parameter settings for channels and feature maps in different layers70
Table 3-8. HI’s for the training and test set with different N_2 an N_4. .............. 71
Table 4-1. Parameter setting of ASM for saliency object detection................... 79
Table 4-2. Results of different saliency models on the test set ......................... 90
Table 5-1. Parameter setting in different layers of the ASM for learning heat maps.
......................................................................................................................... 102
List of Abbreviations

AC: Achanta’s saliency model
AIM: Attention based on Information Maximization
ASM: Adaptive Saliency Model
AUC: Area Under Curve
BP: Back-Propagation
CA: Context-Aware saliency detection
CD: Contrastive Divergence
CNN: Convolutional Neural Network
DBN: Deep Belief Nets
DNN: Deep Neural Network
EOG: Electro-OculoGraphy
FN: False Negative
FP: False Positive
FPR: False Positive Rate
FW: Feedforward
GD: Gradient Descent
GT: Ground Truth
HI: Histogram Intersection
IM: non-parametric Low-Level Vision Model
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT</td>
<td>Itti’s salience model</td>
</tr>
<tr>
<td>JAUC</td>
<td>Judd’s Area Under Curve</td>
</tr>
<tr>
<td>LTDP</td>
<td>Long-Term-Dependency Problem</td>
</tr>
<tr>
<td>MLP</td>
<td>Multi-Layer Perceptron</td>
</tr>
<tr>
<td>POG</td>
<td>Photo-OculoGraphy</td>
</tr>
<tr>
<td>p-ROI</td>
<td>Potential Region-Of-Interest</td>
</tr>
<tr>
<td>PR</td>
<td>Precision-Recall</td>
</tr>
<tr>
<td>RBM</td>
<td>Restricted Boltzmann Machine</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic</td>
</tr>
<tr>
<td>Rprop</td>
<td>Resilient Propagation</td>
</tr>
<tr>
<td>SAE</td>
<td>Stacked AutoEncoder</td>
</tr>
<tr>
<td>SeR</td>
<td>saliency detection by Self-Resemblance</td>
</tr>
<tr>
<td>SOD</td>
<td>Salient Object Detection</td>
</tr>
<tr>
<td>SR</td>
<td>Spectral Residual</td>
</tr>
<tr>
<td>SUN</td>
<td>Saliency Using Natural statistics</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>SWD</td>
<td>Spatially Weighted Dissimilarity</td>
</tr>
<tr>
<td>TDO</td>
<td>Two-Dimensional Output</td>
</tr>
<tr>
<td>TN</td>
<td>True Negative</td>
</tr>
<tr>
<td>TP</td>
<td>True Positive</td>
</tr>
<tr>
<td>TPR</td>
<td>True Positive Rate</td>
</tr>
</tbody>
</table>
VOG: Video-OculoGraphy
Chapter 1 Introduction

1.1 Motivation

Digital cameras have been prevalently used to produce digital visual contents in numerous devices including surveillance monitors, personal computers and mobile phones. Consequently, an enormous amount of digital images are being recorded in a daily basis. An issue brought forth is how to reduce unnecessary efforts spent in unimportant visual contents and efficiently search appropriate visual contents from a huge number of images. To accomplish this, the contents of images should be marked with labels or values that indicate their importance. Saliency prediction and salient object detection are two fundamental and challenging topics in computer vision. Although many efforts have been made, there are still rooms for improvement.

Saliency prediction is to mark the regions that attract human attention and evaluate how much attention human pays to them. Visual sense is one of the most important senses for human beings. The human brain processes visual information extraordinarily fast and reliably on succession of priority according to attention. Inspired by the principle of human visual attention system, computational saliency prediction models are designed to determine prominent regions with saliency-based methods. Saliency prediction models have been comprehensively developed to mimic the visual search process and automatically determine the importance of
regions that might attract humans’ attention. Having acquired the importance of visual contents, further tasks can be carried out such as object detection (Liang et al., 2012) and image retrieval (Fu et al., 2006).

In general, a saliency model belongs to one of two categories: pixel-based and region-based. For the first type, a given scene is divided into pixels. Consequently, a pixel is regarded as the smallest region that shares the same saliency value. The saliency value of a single pixel is computed from its contrast to neighboring pixels or all other pixels in a given image. However, pixel-based saliency models have a problem in balancing local and global contrasts.

In a region-based model, a segmentation step is performed to partition an image into several regions. Each region is therefore treated as a unit that shares the same saliency value. The saliency value of each region is generated from the contrast and consistency measures between regions. A drawback of a region-based model is overlooking the details of segment contents and slighting the relationships among segments. Some of the region-based methods adopt frequency analysis to avoid the drawback. Yet, the performance of segmentation limits the overall performance.

Meanwhile, machine learning methodologies have been rapidly developing since a few years ago as computational power grows fast. In recent researches, machine learning shows a promising capability in processing digital visual information. Support Vector Machine (Cortes and Vapnik, 1995) is a popular methodology in recent saliency models. In most of these models, SVM is used as a
classifier to classify a pixel/region as salient or non-salient. Apart from SVMs, deep neural network has recently been a popular methodology in machine learning. It has been illustrated that a deep hierarchical architecture can manifest more abstract and higher level features than a shallow architecture. Beside a classifier, a Deep Neural Network (DNN) might be used as a feature extractor, which is different from SVMs. Therefore, my study concentrates on deep architectures.

Among DNN architectures, Convolutional Neural Network (CNN) (Krizhevsky et al., 2012; LeCun et al., 1998) has a highly hierarchical architecture which is similar to Itti’s model, a state-of-art saliency model. Thus, CNN has a structural advantage in saliency detection. Similar to other DNN models, a CNN contains multiple hidden layers that reduce the dimension of data gradually. As the data moves feed-forward, low-level and high-level features could be extracted. Some of those saliency models with DNNs involve the use of DNN as a classifier, some use DNN as a feature extractor; yet some use two DNNs where one is used to perform feature extraction and the other is to perform classification. To my best knowledge, DNN is seldom used as a function approximator.

In saliency models which are based on machine learning techniques, hand-designed feature extractors are often adopted. A hand-designed feature extractor is usually effective and meaningful. However, a hand-designed feature extractor also has drawbacks of neglecting detailed, important and useful information. The superiority of feature extraction with machine learning is to acquire important
information that is easily overlooked by human beings. Nonetheless, unwanted and useless feature or even those with negative effects could also be extracted concurrently.

This thesis addresses CNN based saliency map generation and saliency object detection, very challenging topics in computer vision. How well a CNN can handle a function approximation type of task is studied. In particular, I made an investigation into a suitable training algorithm for CNN-based saliency prediction and saliency object detection.

1.2 Statements of Originality

This thesis presents deep hierarchical architectures and learning algorithms for saliency prediction and salient object detection. The work described in this thesis was carried out at the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, between September 2013 and November 2015, under the supervision of Dr. Zheru Chi.

The thesis consists of six chapters. The work presented in this thesis was originated by the author except where acknowledged and referenced, or where the conclusions are widely known. The main original contributions in this thesis are summarized as:

(1) An adaptive saliency model based on the convolutional neural network for saliency prediction is the work of the author. In this investigation, I proposed a
two-dimensional-output deep neural network model for saliency prediction. The deep neural network is trained to generate saliency maps directly from input images. The 2-dimensional output of the deep neural network makes my proposed model a function approximator. To overcome the long-term-dependency problem during the DNN training progress using a gradient descent algorithm, a modified error back-propagation algorithm called Rprop was used to train my proposed model. The model automatically learns low-level and high-level features as well as the mappings between regional visual features and saliency using original images and the saliency ground truth of training images. The kernels in the first convolutional layer learn low-level features from input images. In the second convolutional layer, the kernels learn high-level features from the feature maps containing low-level features. The output layer establishes the relationship between the feature maps and saliency values.

(2) Combining pixel-based bottom-up and region-based top-down approaches for salient object detection. In this investigation, I first employed p-ROI approximation method to perform salient object/region detection, and showed that p-ROI approximation was a promising region-based approach for detecting regions containing salient objects. Then I proposed a hybrid pixel-based bottom-up and region-based top-down approach to effectively detect salient objects. The pixel-based approach aims to evaluate possibility of salient object located in each pixel. The region-based approach is to detect the salient region and refine the result from the
pixel-based approach.

(3) Learning the heat maps of human eye gaze data using a CNN based model. I first employ ASM with the Rprop algorithm to generate heat maps and show that the deep learning method can only achieve a moderate performance. I then modify the approach to have the deep neural network pre-trained on Itti saliency maps and show that this pre-training process can slightly improve the performance. I also propose a distribution processing step to produce more meaningful heat maps from the results of my model.

1.3 Outline of the Thesis

The thesis consists of six chapters. The thesis is organized as follows.

Chapter 2 introduces the basic principles and definitions of saliency prediction, salient object detection, feature learning and deep neural network, and reviews important developments achieved in these fields.

In Chapter 3, I investigate an Adaptive Saliency Model (ASM) based on the convolutional neural network to perform saliency prediction. The proposed ASM consists of convolutional layers as well as subsampling layers from the convolutional neural network and a two-dimensional output layer. In order to overcome the long-term-dependency problem that hinders the DNN training process with a gradient descent algorithm, a modified error back-propagation algorithm called Rprop is used to train my proposed model. Low-level and high-level features as well
as the mapping between regional visual features and saliency are automatically learnt
with original images and the saliency ground truth of training images. Various
aspects including the saliency prediction performance, the training time and the
number of kernels used in convolutional layers are systematically studied.
Experimental results on the Object and Semantic Images and Eye-tracking (OSIE)
dataset demonstrate the effectiveness of my proposed ASM compared with
state-of-the-art algorithms for saliency prediction based on the Histogram Intersection
(HI) metric.

In Chapter 4, a combination method of pixel-based bottom-up and region-based
top-down approaches for salient object detection is proposed. I first employ the
p-ROI approximation method to generate a saliency map indicating salient
object/regions. Experiment results show that p-ROI approximation is a promising
region-based approach for detecting salient object. Then I propose a hybrid
pixel-based bottom-up and region-based top-down approach to perform salient object
detection. The pixel-based method is targeting at evaluating the possibility of a
salient object located at pixel level. The region-based method aims to detect the
salient region containing salient objects and to improve the overall salient object
detection performance. The metrics including precision and recall curve and Area
Under Curve (AUC) are adopted to evaluate the performance of my approach of SOD.
Experimental results on a dataset with manually demarcated ground truth demonstrate
a superior performance of the hybrid SOD model comparing with each individual
Chapter 5 proposes a CNN based approach for learning the heat maps of human gaze data. I first employ ASM with the Rprop algorithm to generate heat maps and show that the deep learning method can only achieve a moderate performance. I then modify the approach to have the deep neural network pre-trained on Itti saliency maps and show that a pre-training process can slightly improve the performance. Inspired by the “spotlight” effect of human eye gaze, I apply a distribution processing step to the output of ASM in order to produce more meaningful heat maps. The metrics including precision and recall curve, Receiver Operating Characteristic (ROC) and AUC are adopted to evaluate the performance of my leaning model on both the OSIE dataset and the CAT2000 dataset.

Chapter 6 concludes this thesis with final remarks and discusses some possible directions of future research.
Chapter 2 Literature Review

In this chapter, the basic principles and definitions of saliency prediction, salient object detection, feature learning and deep neural network are given. The important developments in these fields are reviewed and discussed.

2.1 Basic Definitions

2.1.1 Visual Attention

Visual attention provides human beings with the capability of searching and selecting most important information and differentiating the targets from crowded scenes that consists of a huge amount of irrelevant information. It has become an influential and important research topic with many papers published in the past few decades. In an early research, James stated a clear description on visual attention (James, 1890):

“Everyone knows what attention is. It is the taking possession of the mind, in clear and vivid form, of one out of what seem several simultaneously possible objects or trains of thought. Focalization, concentration of consciousness are of its essence. It implies withdrawal from some things in order to deal effectively with others, and is a condition which has a real opposite in the confused dazed, scatterbrain state …..”

Psychologists (Treisman, 1986; Treisman and Gormican, 1988) claimed that visual attention manipulates like a “spotlight” to focus on certain objects or events.
In the further investigations targeting at learning how human visual attention works, Helmholtz (von Helmholtz and Southall, 2005) discovered that human beings select only part of the interested events while attention points moving dynamically. On the other hand, the demonstration of the independence of attentional and ocular focal points proved that attention is free to direct by consciousness and autonomous to a particular location without making eye movements.

2.1.2 Computational Saliency Models

Altogether, saliency prediction models and salient object detection models are called saliency models. By mechanism, they can be categorized into two groups: non-learning based and learning based. A non-learning based saliency model generates saliency maps using predefined formulas and feature extractors only, while a learning based saliency model generates saliency maps by adopting a machine learning technique. The procedures of a non-learning based and a learning-based strategies for generating saliency maps are shown in Figure 2-1. For a given image, a non-learning based approach first extracts visual features and then map visual features to a saliency map with pre-specified relationships. By contrast, a learning based approach performs feature extraction and classification sequentially while at least one of the two tasks is carried out by the model generated from a machine learning technique.
Figure 2-1. Procedures of a non-learning based approach (left) and a learning-based approach (right) for generating saliency maps.

2.2 Non-learning Based Saliency Model

Non-learning based approaches can be further grouped into three categories: bottom-up, top-down and hybrid approaches. In a bottom-up approach, attention is driven by pre-specified low-level visual features of the stimuli such as an object with an outstanding color. While in a top-down approach, attention is led by high-level understanding from human consciousness such as searching a certain object in a cluttered scene.

2.2.1 Bottom-Up Approaches

In a bottom-up approach, saliency is determined by fundamental features, such
as orientation and color. Koch and Ullman proposed a saliency model to characterize the conspicuousness of pixels using a selective visual attention technique (Koch and Ullman, 1987). Extending from the same concept, a number of computational visual attention models have been proposed. Among them, Itti’s model (Itti et al., 1998) is the most widely adopted one. Feature and contrast maps are computed based on center-surround contrasts. The saliency map that represents the importance of pixels in an image is generated feature and contrast maps.

Following the center-surround concept, a saliency model based on the discriminant power of feature sets and center-surround biases was proposed by Gao et al. (Gao et al., 2008). Arbitrarily scaled features and local center-surround pairs were combined in Klein’s model (Klein and Frintel, 2011) to compute saliency maps that characterize divergence of objects.

A region-based saliency model proposed by Aziz and Mertsching (Aziz and Mertsching, 2008) first constructs feature maps in terms of color, size, symmetry, orientation and eccentricity. The overall saliency map produced from their model is weighted summation of those feature maps.

Beside pixel-based models, region-based models adopt the principle of contrast as well. Spatial information enhanced region-based contrast and histogram-based contrast were combined as regional features to generate a saliency map in a region-based model proposed by Cheng et al (Cheng et al., 2015). Hou and Zhang (Hou and Zhang, 2007) proposed a model which analyzes information beyond spatial
domain. A spectral residual approach was adopted to appraise saliency of images in the frequency domain in which the log-spectrum of a given image was used to extract spectral residual. Spectrum residual were then divided into innovation part and redundant part where redundant part was suppressed while innovation part was transformed into a saliency map.

Another non-learning based saliency model utilized color and luminance features in both the input image and its corresponding filtered image where dissimilarity of the two images was exploited to perform saliency detection (Achanta et al., 2008). Instead of contrast, local symmetry was adopted as a measure of saliency in Kootstra and Schomaker’ model (Kootstra and Schomaker, 2009). Saliency map, known as symmetry map in their model, is the combination of feature maps which are obtained though symmetry operators. It achieved a good matching to human eye tracking data.

A simple bottom-up visual saliency model proposed by Harel et al (Harel et al., 2006) consists of two steps. The first step is to form activation maps on defined feature channels, and the second is to normalize activation maps to highlight conspicuity and to obtain a combination of all the maps.

In a saliency model proposed by Yan et al (Yan et al., 2014), the images of different scales from the input image are generated to compute saliency cues for each layer. All the cues are then combined to deduce the final saliency values in a local consistent hierarchical inference model.
2.2.2 Top-Down Approaches

![Figure 2-2. Four chessmen in red, yellow, green and blue with a pure background.](image)

Since a top-down approach needs a prior knowledge, it is usually a task-dependent approach. For example, in an image with a pure background, there are four chessmen with, respectively, red, yellow green and blue colors as shown in Figure 2-2. In a bottom-up saliency model, the saliency levels of four chessmen should have comparable values because in the visual cortex, red, green and blue are in different colors channels, and four chessmen share the similar proportion in the image. However, in a top-down saliency model, the green chessmen might have a high saliency level than other three chessmen if a certain task is given (e.g. locate the green object in the image). Therefore, the top-down saliency model is greatly determined by the task given.

Rao et al proposed a top-down saliency prediction model by exploiting in a coarse-to-fine fashion between the input and task-relevant target’s filter responses
information (Rao et al., 2002). It illustrates that with the aid of prior knowledge it indeed improves prediction accuracy. A top-down region partition based approach was proposed by Wu et al (Wu et al., 2008) to demonstrate that a top-down based saliency model could become an efficient algorithm.

### 2.2.3 Mixture of Bottom-UP and Top-Down Approaches

In a mixture of bottom-up and top-down approach, bottom-up part acts as an important role to detect all possible objects/regions. Simultaneously top-down part assists to further pick out regions that meet the requisite of a certain task. Thus, the saliency is computed by both fundamental features and prior knowledge on the task.

Accumulated statistical information of the features was extracted from the task-relevant target and the background to refine a bottom-up map in Navalpakkam and Itti’s model (Navalpakkam and Itti, 2006). Multi-scale frequency/phase-based information and statistical cue were extracted as bottom-up and top-down attention maps in Fang’s mixture model (Fang et al., 2011).

Another combined top-down and bottom-up non-learning-based model was proposed by Zhang et al (Zhang et al., 2008). The features used in the model are hand-designed self-information and log likelihood. Self-information is computed from statistics information to represent bottom-up saliency and log likelihood reflects top-down knowledge.
2.2.4 Itti’s Saliency Model

Itti et al proposed a state-of-the-art model to topographically determine visual saliency using visual features and center-surround contrasts (Itti et al., 1998). The procedure of generating Itti’s saliency map is shown in Figure 2-3.

![Diagram of Itti's Saliency Model](#)

Figure 2-3. The procedure to obtain Itti’s saliency map of an image (Itti et al., 1998).

To progressively analyzing fundamental visual features, feature maps are generated from the multi-scale input. The input image is subsampled at different resolutions as a hierarchical structure, in which Gaussian pyramids introduced by Greenspan et al (Greenspan et al., 1994) are adopted to obtain nine spatial scale images with $\sigma \in \{0,1,2,\ldots\}$. Fundamental/low-level features particularly, color, intensity and orientation, are extracted from each image at each scale, respectively. The fundamental/low-level feature maps are discussed as follows. The input color map is obtained with Equation (2.1) from the original RGB image where R, G, B and
Y are four color channels representing red, green, blue and yellow, and \( r, g, b \) are color amplitudes in the RGB space.

\[
\begin{align*}
R &= r - \frac{g + b}{2} \\
G &= g - \frac{r + b}{2} \\
B &= b - \frac{r + g}{2} \\
Y &= \frac{r + g}{2} - \frac{|r - g| - b}{2} 
\end{align*}
\]

(2.1)

A further subsampling is carried out by Gaussian pyramid to obtain multi-scale color maps. The intensity map is obtained by

\[
I = \frac{(r + b + g)}{3}
\]

(2.2)

In the same way, nine intensity maps are generated by Gaussian pyramid. Orientation maps are further extracted from the original intensity map \( I \) using Gabor pyramid \( O(\sigma, \theta) \), with the scales \( \sigma \in [0, 1, 2, \ldots] \) and the orientation \( \theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \).

To implement the concept of center-surround contrast, a center-surround algorithm was proposed to produce conspicuous maps based on feature maps. The “center” and “surround” are defined in two different scales: \( c \) and \( s \). Both \( c \) and \( s \) are defined as

\[
\begin{align*}
c &\in \{2, 3, 4\} \\
\delta &\in \{3, 4\} \\
s &= c + \delta
\end{align*}
\]

(2.3)

The center-surround contrast operator between a map of “center” and a map of “surround” is defined as \( \Theta \). Thus, the feature maps based on a low-level feature can
be derived individually. The intensity feature map can be defined as

\[ I(c,s) = \|I(c) \Theta I(s)\| \] (2.4)

In the model, only two contrasts based on color are defined to represent all color feature maps:

\[
\begin{align*}
RG(c,s) &= \|(R(c) - G(c)) \Theta (G(s) - R(s))\| \\
BY(c,s) &= \|(B(c) - Y(c)) \Theta (Y(s) - B(s))\|
\end{align*}
\] (2.5)

In Equation (2.5), \( RG \) and \( BY \) are the red-green color opponent and blue-yellow color opponent, respectively. \( R, G, B, Y \) are defined in equation (2.1). \( c \), and \( s \) are the “center” and the “scales”, respectively.

Similarly, the orientation feature maps are derived as the orientation contrast between the “center” and the “scales”.

\[ O(c,s,\theta) = \|O(c,\theta) \Theta O(s,\theta)\| \] (2.6)

The next step of Itti’s model is to sum all feature maps for each feature into a normalized conspicuous map. The normalization operation in this step is aiming to avoid strong peaks (if any) of activities from suppressing the conspicuous map and to enhance the conspicuous map. The normalization operator is defined as

\[
\begin{align*}
\bar{I} &= \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{4} N(I(c,s)) \\
\bar{C} &= \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{4} \left( N(RG(c,s)) + N(BY(c,s)) \right) \\
\bar{O} &= \sum_{\theta \in \{0^{\circ}, 45^{\circ}, 90^{\circ}, 135^{\circ}\}} N(\bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{4} N(O(c,s,\theta))
\end{align*}
\] (2.7)

where \( \bar{I}, \bar{C}, \) and \( \bar{O} \) are the conspicuous maps of intensity, color and orientation, \( \bigoplus \) stands for a cross-scale summation, and \( N(\bullet) \) is a map normalization operator defined as follows:
\[ N(x) = M \times (M - \bar{m})^2 \times \frac{x}{\max(x)} \]  
(2.8)

where \( x \) is the input map, \( M \) stands for a threshold used to limit the global maximum value of a normalized map, and \( \bar{m} \) denotes the average of all other local maxima.

Figure 2-4. Example of cross-scale summation.

As shown in Figure 2-4, a cross-scale summation is an operation used to sum up two maps of different sizes. The map with a smaller size is firstly up-sampled to be the same size as the map of a larger size. Then, two maps are simply summed up.

Finally, a saliency map is produced by taking the average of three conspicuous maps:

\[ S = \frac{1}{3} \left( N(\bar{I}) + N(\bar{C}) + N(\bar{O}) \right) \]  
(2.9)

In the saliency map, the zero-value regions are non-salient regions. On contrary, the
non-zero regions represent the salient regions in the input image and their amplitudes reflect the conspicuous levels of pixels within the salient regions.
2.3 Learning Based Saliency Model

Learning based approaches can also be grouped into three categories: feature extractor learning, classifier learning and mixture learning. In a feature extractor learning approach, saliency is obtained by trained feature extractors and hand-designed classifiers. While in a classifier learning approach, visual features are extracted by hand-designed extractors. Then the relationship between extracted features and saliency maps are learnt with a learning method. In a mixture approach, both feature extraction and classification is performed using the models generated from machine learning techniques.

2.3.1 Feature Extractor Learning

The first category of learning based saliency model is the feature extractor learning saliency model. The common characteristic of models in this category is that a model consists of two steps of processing. The first step is adopting a machine learning methodology to obtain a feature extractor while the second step is exploiting the mappings between feature maps and the saliency maps.

Bruce and Tsotsos (Bruce and Tsotsos, 2005) proposed a bottom-up saliency model with feature extractor learning. A neural circuit was employed as an unsupervised self-information extractor to perform feature extraction. With pre-specified mappings between the saliency map and feature maps, the final saliency map was produced.
2.3.2 Classifier Learning

The second category of learning based saliency model is classifier learning saliency model. The common characteristic of models in this category is that a model consists of two steps processing where the first step is exploiting a feature extractor while the second step is utilizing a machine learning method to learn the mappings between feature maps and the saliency map.

Itti and Koch (Itti and Koch, 2001) extended their saliency model of a non-learning based model (Itti et al., 1998) to a learning base saliency model. The only modification is to replace the linear combination of conspicuous maps to trainable weighted summation. This change makes the pre-specified mapping between feature maps and the saliency map became the mapping learnt by a machine learning technique.

Peter and Itti proposed a model following the feature-extractor-classifier pattern (Peters and Itti, 2007). The hand-designed features are sets of dyadic pyramid and Fourier components, random feature and eye tracking data feature while the classifiers are linear network, non-linear multilayer network, Gaussian mixture model and SVM for different feature sets. The classification result is exploited as top-down part of the whole model which is combined with bottom-up part (non-learning) to obtain the final saliency map.

Judd et al (Judd et al., 2009) proposed a learning-based model considering 33 hand-designed features of low-, mid- and high-level to generate the saliency map.
with the assistance of SVMs. SVMs are trained with multiple kernel learning to
determine the weights of feature maps. This learning-based model shows a promising
result.

The saliency model proposed by Xu et al (Xu et al., 2014) specified 20
hand-designed features at different levels such as pixel-, object- and semantic-level.
Their model utilized linear SVMs to produce final saliency maps by training the
weights. The trained linear SVMs generates a map similar to the human fixation
map.

2.3.3 Mixture of Feature Extractor and Classifier Learning

The third category of learning based saliency model is based on the mixture of
feature extractor and classifier learning. A model in this category consists of one or
two machine learning models. In majority, a saliency model consists of a learnt
feature extractor and a learnt classifier. In case of one learning model, it should have
the capability of acting as both feature extractor and classifier simultaneously.

Milanese et al (Milanese et al., 1994) proposed an integrated bottom-up and
top-down learning based saliency model. In the top-down subsystem, distributed
associative memories were adopted to learn object categories from training images.
The distributed associative memories could be regards as both feature extractor and
classifier since the technique could recognize the category of an object directly from
the input image.
Yang and Yang (Yang and Yang, 2012) proposed a top-down visual saliency model with a customized Conditional Random Field (CRF) with sparse coding and SVMs, acting as a feature extractor that learn to generate a discriminative dictionary. SVM is trained on sparse coding and corresponding patch labels.

A multi-layer sparse network plays a role of feature extractor in Shen and Zhao’s saliency model (Shen and Zhao, 2014). Hierarchical features are learned from human eye fixations with greedy layer-wise training using sparse coding. After feature learning, a linear SVM is trained to obtain the weights of all the features. The weighted summation of all the feature maps forms the final saliency map.

Vig et al. (Vig et al., 2014) proposed a learning based saliency model that consists of two machine learning components. The first one is a feature learning pipeline which is learned to extract high-level features from hand-designed low-level features and fixation maps, while the second one is an SVM as a classifier. Both hand-designed feature maps and learnt feature maps are fed into the SVM to generate the final saliency map. Each pixel is regarded as individual unit in the classification.
2.4 Deep Learning and Feature Learning

Traditional feature extraction focuses on feature optimization, which is to manually design and enhance feature representation mechanisms in accordance with experiments and experiences. Usually, the manually designed feature representations perform well for some tasks. However, they may not perform well on the other tasks. Moreover, manually designed features require manually tuning of some parameters. Choosing proper parameters becomes a difficult job even for experienced researchers. Consequently, feature learning has attracted significant attention in the recent years.

Feature learning is expected to learn good features automatically. Feature learning did not accomplish much progress until 2006 when deep learning came back to the sight of researchers (Bengio, 2009; G. Hinton et al., 2006; G. E. Hinton et al., 2012). Since deep architectures have difficulties in training with error back-propagation, little research focus on deep learning before 2006 (Glorot and Bengio, 2010). A deep architecture performs well in extracting high-level features (Bengio, 2009). Training multilayer architectures using the traditional Gradient Descent (GD) algorithm, however, usually leads to failure. Bengio and his colleagues also found when weights have proper initialization, it is possible to successfully train a deep architecture using a GD algorithm. Hinton and Salakhutdinov (G. Hinton et al., 2006) proposed a method known as greedy layer-wise to pre-train deep neural networks. By applying this method, weights in DNN gain so good initialization that training the DNN with a GD algorithm known as fine-tuning becomes possible. In
their research, the whole training progress of Deep Belief Nets (DBN) consists of a pre-training step in which every two connected layers are regarded as Restricted Boltzmann Machine (RBM) and the connection weights are trained with Contrastive Divergence (CD) algorithm, and applying a GD algorithm in fine-tuning the whole network. They showed that DBN achieved an excellent performance on digit recognition. However, due to the reduction of dimensionality of data during feed-forward leads to its failure in extracting spatial cues from images.

2.4.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep neural network which shares a few similarities with DBN. Since 2006, both CNN and DBN have become ones of the hottest research topics.

2.4.1.1 Early Development on Convolutional Neural Network

CNN was a discovery during Hubel and Wiesel researching on cat cortex back in 1962 (Hubel and Wiesel, 1962). In their research, when they were researching neurons that used to control local sensitivity and direction selection in cat cortex, they found a unique structure that could effectively reduce the complexity of feedback neural network. Inspired by the structure of CNN in cat cortex, Fukushima proposed a self-organizing neural network model for pattern recognition (Fukushima, 1980). This self-organizing neural network became the first implementation on digital CNN.
In 1995, LeCun (LeCun et al., 1995) proposed a model called LeNet-5 to recognize handwritten digits. It became the prototype of most CNN people are researching nowadays. Due to the limitation of computational power, even though LeNet-5 showed an extraordinary performance, it could hardly be extended to other complicated fields in computer vision. At the same year, LeCun and Bengio (LeCun and Bengio, 1995) successfully extended their research to image and speech recognition.

Lawrence et al proposed a face recognition model using convolutional neural network approach (Lawrence et al., 1997). Their model consists of two machine learning components. The first one is a self-organizing map while the second is a CNN. In their model, a self-organizing map is utilized as a dimensionality reduction method while a CNN is used as both feature extractor and classifier.

2.4.1.2 New Development on Convolutional Neural Networks

Since DNN started to boom in 2006, researchers have not only found new algorithms for training CNN but also secured the drastically improved computational power. Consequently, CNN models for various applications were proposed in the past decade.

Ciresan et al (Ciresan et al., 2011) proposed an image classification model using CNN. The CNN model they used shared the same setting as LeNet-5 but with a larger number of kernels. GPUs were employed to perform parallel computing of CNN training. Beside handwriting recognition, a hybrid NN-HMM model for speech
recognition was proposed by Abdel-Hamid et al (Abdel-Hamid et al., 2012). In their research, CNN was acting as both feature extractor and classifier.

Not until Krizhevsky et al (Krizhevsky et al., 2012) proposed an image classification model with CNN, did people realize CNN had a natural advantage in spatial information extraction comparing to other DNNs. In their research, they constructed an enormous CNN to classify ImageNet (Deng et al., 2009). A promising result was demonstrated comparing with other methods. At the same year, researchers explored CNN application in video classification (Karpathy et al., 2014). Similar to the one on ImageNet, they adopted a CNN and showed it was a promising tool in processing a large amount of visual data.
2.5 Eye Tracking Techniques and Applications

Eyes are indispensable organs of vision for human being to observe the world. In the human visual system, visible light is detected with eyes and then it transforms into electro-chemical impulses in neurons. Having this fascinating mechanism, eyes have been researched by scientists since ancient times. Duchowski’s (Duchowski, 2007) research results show that, in order to observe the fine details of visual contents, human beings tend to have their eyes moved to bring a specific part of the visible field of view into high resolution. This research reveals a possibility that if human’s eye movement could be detected and tracked, the observation path and portions that attract human attention could be read. The technique that records human beings’ eye movements is named eye tracking. The devices invented for measuring eye gaze locations and durations is called eye tracker.

2.5.1 Eye Structure Analysis

Human eyes are composed of series of optical structures listed as follows:

- **Cornea**: The cornea is a transparent film in the outer front part of the eye. It covers and protects other parts of the eye such as iris and pupil.

- **Iris**: The iris is a thin circular structure covered by cornea. It controls the size and diameter of the pupil.

- **Pupil**: The pupil is the opening in the center of the eye. Light have to go through the pupil to enter eyes.
- **Lens**: The lens is a transparent biconvex multilayer structure behind iris and pupil. Its main function is to refract light to be focused on the retina.

- **Retina**: The retina is a thin layer of photosensitive cells that lines the back of the eyeball. Light is converted to electro-chemical pulse in these cells.

- **Fovea**: The fovea is a small pit in the center of the retina. It is responsible for sharp central vision which is a necessity in observing details.

The optical structures are shown in Figure 2-5. There are two axes shown in the figure. One is named optical axis defined between the center of the pupil and the center of the eye ball. Another is known as visual axis which connects the center of cornea and the fovea. According to Oyekoya’s research (Oyekoya and Stentiford, 2007), instead of optical axis, visual axis is the one determines human visual attention. The research also reveals the angle between two axes of the eye is $5\pm 2$ degree.

![Figure 2-5. The cross section of the human eye (adopted from (Duchowski, 2007)).](image)
Eye movements consists of the voluntary or involuntary eye movements to assist in acquiring visual stimuli. There are six types of eye movements to ensure that the fovea is aiming at observing objects.

- **Fixation:** Fixation represents a constant focusing on one spot. Fixations with focusing duration equals or above 80 milliseconds are regarded as intentional fixations. Fixations with duration less than 80 milliseconds are negligible. Because fixation locations are interrelated to interesting point, it is possible to find people’s interesting content by using fixations. Researches also point out complex scenes or images takes more time and more fixations to observe comparing to simple ones.

- **Saccade:** Saccade is rapid simultaneous movement of both eyes to change the focus point. It is a voluntary movement that responsible for repositioning the fovea to new spot in the scene. The saccade movement is actually ballistic. In other words, once the movement initiates the trajectory cannot be changed. According to researches, the movement speed is very slow before a selection, then it becomes one of the fastest movements that produced by human body.

- **Pursuit:** Pursuit is only involved when eye sight tracking a moving object. Eyes are able of adapting the velocity the moving object based on the range of object motion.

- **Vergence:** Vergence is a simultaneous movement for both eyes to converge
to point at the same spot. In general, vergence is an involuntary movement.

- **Vestibular Nystagmus**: Vestibular Nystagmus is a linear slow form of involuntary eye movement. Eye travels at constant speed in certain direction, and then reset saccade in the opposite direction.

- **Physiological Nystagmus**: Physiological Nystagmus is a form of involuntary eye movement that continuously shifts the image on the retina in a high frequency.

Among all six types of eye movements, fixation, saccade and pursuit are the most important in eye-tracking data analysis and applications.

### 2.5.2 Eye Tracking Techniques

There are four eye tracking techniques that have been developed in the past. They are sclera contact lens techniques, Electro-OculoGraphy (EOG), Photo-OculoGraphy (POG)/Video-OculoGraphy (VOG), and video-based hybrid of pupil and corneal reflection technique.

In the early development of eye tracking techniques, most popular eye trackers were intrusive. Robinson (Robinson, 1963) proposed to integrate contact lens with a small coil as eye tracker. The eye tracker was directly worn on human eyes. The eye tracker operated in a uniform magnetic field. Once the eye moves, the coil on the contact lens would response to the magnetic field and output electronic signals. By recording the output voltage, this technique was the most accurate and sensitive
methods. The measurement accuracy would reach 0.08 degree.

Electro-OculoGraphy (EOG) was the most popular non-intrusive eye tracking technique in the early days. It detected small difference of electric potential of skin around the eye by placing electrodes on the skin. According Kaufman’s work (Kaufman et al., 1993), the sensitivity of this eye tracking technique is set as $20\mu V/deg$ while recorded potentials are in the range of $15–200\mu V$. The measurement accuracy is about 2 degrees.

For the type of Photo-OculoGraphy (POG) or Video-OculoGraphy (VOG), eye movements are tracked based on the changes of pupil shape and the position of boundary between the iris and the sclera. Colors on both sides of the boundary are in a significant contrast. Therefore, the tracker is sensitive to horizontal boundary change. However, since part of iris is covered by eyelids, the vertical boundary change is relatively low.

Video-based hybrid of pupil and corneal reflection technique adopt infrared ray to produce reflection on the corneal surface in order to measure the location of pupil center. Due to the corneal reflection effect, the pupil could be easily detected. Therefore, nowadays most of eye gaze trackers are based on this technique because of its simplicity and good accuracy.

2.5.3 Eye Tracking Applications

Eye tracking techniques provide scientists new ways to analyze human thoughts.
With the aid of eye trackers, it is possible to study active tasks such as walking, driving, playing tennis and ordinary everyday activities like cooking (Land, 2006). In addition, eye tracking techniques can be applied in active modes and passive modes. Eye tracking techniques are used as controlling methods through the motions of the eyes in active modes. A decades ago, Hansen et al (Hansen et al., 1995) proposed a multimedia system to use eye gaze as a pointer to be a mouse alternate. Merchant and Schnell (Merchant and Schnell, 2000) utilized eye tracking technique as a replacement control tool in aircraft. Eye movement was further used in a display system presented by Numajiri et al (Numajiri et al., 2002) that enables people to find interested information quickly. In the same year, eye tracking technology was applied to assists disabled users in Corno et al’s research (Corno et al., 2002). Text entry device was also combined with eye tracking. Characters could be selected and input according to users’ eye gaze. Ward and Mackay’s research (Ward and MacKay, 2002) showed the desired words could be selected easily and quickly. Eye tracking technologies could be used to perform diagnostic analysis in passive modes. Eye tracking studies were conducted to analyze how users interact with search engines on website in order to improve the interface design of website (Cutrell and Guan, 2007; Granka et al., 2004; Pan et al., 2004). Chu et al (Chu et al., 2009) studied the impact of different digital design combinations on website with the aid of eye tracking. Besides marketing researches and designing researches, passive modes can be used in medical diagnosis. For example, Chen et al (Z. Chen et al., 2015) proposed a
Eye tracking enables researchers to collect evidence of where visual attention locates. Parkhurst and Niebur (Parkhurst and Niebur, 2003) claimed that eye movements were determined by visual attention in the conditions of free viewing. Back in 1967, Mackworth and Morandi (Mackworth and Morandi, 1967) found that the density of information determines the fixation density in corresponding regions of an image. Researches revealed that human beings tend to staring at interested regions with longer duration and more fixations (De Graef et al., 1990; Henderson and Hollingworth, 1998). Therefore, eye gaze data reflect both a bottom-up visual searching procedure driven by low-level features, and a top-down observing process driven by knowledge and understanding of scenes. In other words, eye tracking techniques can provide an imperceptible and objective way to capture human beings’ focus, analyze their interests and understand their behavior.

The majority of visual attention applications are in two categories. One category is to use eye movement data as feedbacks from users. Faro et al (Faro et al., 2010) presented an implicit relevance feedback method to determine the importance of image feature with collated eye gaze data. Recommender systems were introduced to propose the most interesting web documents to users based on their eye gaze data while web browsing (Castagnos et al., 2010; Giordano et al., 2012). Based on the regions of interest that users tend to focus on, an image retrieval interface was designed to retrieve plausible candidate images by Oyekoya and Sentiford (Oyekoya
and Stentiford, 2006, 2007). The same group of scientists (Oyekoya and Stentiford, 2004) discovered the relationship between eye gaze and a visual attention model which recognizes segments of interest in images. Another category of visual attention application is using eye movement data as the ground truth. Eye tracking technique could help to find clues in human being’s attention as the ground truth so that it can be extend to help designing image quality metrics (H. Liu and Heynderickx, 2011; Ninassi et al., 2007). The ground truth is also useful in training and evaluating computational visual attention models (Judd et al., 2009; Le Meur et al., 2006).

A heat map is a graphical representation of data matrix where individual values are shown in color pixels or gray pixels. The name of heat map was originally trademarked in 1993 to display 2D real time financial market information. It originated in the visualization of the magnitudes or values in a two-dimensional data matrix of a cluster analysis by rearranging the rows and columns for better clustering (Sneath, 1957) (Figure 2-6).

Figure 2-6. The first heat map (adopted from (Sneath, 1957)).
Robert Ling (Ling, 1973) improved the heat map by utilizing overlapped printing characters to represent different value of data in pixels. Nowadays, in many gray level heat maps, larger magnitudes/values are shown by light pixels and smaller magnitudes/values are represented by dark pixels. To obtain an aggregate view of the eye gaze data (fixations) from a number of observers, scientists adopted color heat maps (Outing and Ruel, 2004) (Figure 2-7). In their work, the heat map represented the area where people looked at more often (hot spots) with a warm color. The area that people looked at less often was shown with a cold color. In my study reported in Chapter 5, a CNN based model was trained to learn the heat maps of human eye gaze data.

Figure 2-7. The color heat map of fixations captured by observing a short paragraph (adopted from (Outing and Ruel, 2004)).
Chapter 3 Adaptive Saliency Model for Saliency Prediction

3.1 Background

Visual saliency prediction, usually termed saliency prediction, is a process that employs various computational models and algorithms to obtain the importance of visual contents in an image, and then plots the level of importance of visual contents on a map. The level of importance of visual contents is named as saliency or salience, and the plotted map is termed as saliency map. In recent years, apart from the hand-designed visual feature extraction and fixed direct mapping between visual features and saliency maps, trainable visual feature extractor and machine learning of the mapping between visual features and saliency have also been adopted to enhance the performance of saliency prediction (Bruce and Tsotsos, 2009; Itti and Koch, 2001; Judd et al., 2009; Peters and Itti, 2007; Xu et al., 2014). However, existing learning-based saliency prediction models adopt machine learning techniques either for feature extraction or classification with one component hand-designed. Note that the hand-designed component is possible to miss important information. Hand-designed features were extracted with pre-designed functions such as the orientation and illumination functions defined in Itti’s research (Itti et al., 1998). Hand-designed features normally requires expensive human labor and usually rely on expert knowledge and experience. Also, some hand-designed features do not
generalize well as they are originally designed on one biased data set. Hand-designed feature extractors may not be able to discover some meticulous features which are easily ignored. Hand-designed classifiers may have the similar drawback in neglecting some meticulous features and mappings. Both circumstances are possible to result in a performance drop. Hence I propose a new approach that has both a trainable feature extractor and a trainable classifier. In view of this, I seek machine learning methodologies that can be employed for both feature extraction and classification in saliency prediction. An advantage of utilizing a single machine learning technique for saliency prediction model is that the transition between the feature extractor and classifier can be made smooth with less tuning effort. The disadvantage, however, is a single learning machine which is capable of both feature extraction and classification needs a complicated structure that results in being difficult in training. To address these problems, in this chapter, a new learning structure is proposed and its algorithm is proposed.

Neural networks have been widely adopted in many domains, such as digital signal processing (Burse et al., 2010; Q. Zhu and Cao, 2011; X.-L. Zhu and Wang, 2011), system control (C.-H. Chen et al., 2009) and pattern recognition (Chang et al., 2009; Tsurugai et al., 2008; Yu et al., 2010; Zhao et al., 2008), because of their excellent abilities on information processing. Especially in the computer vision domain, neural networks have shown their advantages for various applications including image annotation (Tsurugai et al., 2008; Zhao et al., 2008), which requires
powerful computational capability on image processing, and saliency prediction (Bruce and Tsotsos, 2005; Xu et al., 2014). Bruce et al. (Bruce and Tsotsos, 2005) made use of a neural network for feature extraction. However, in their mode, hand-designed feature extractors and classifiers are still dominant. Xu et al. (Xu et al., 2014) designed different levels of features such as complexity, eccentricity and watchability. Those features were arranged as maps. Together with human fixation maps, they were feed into a linear SVM to train a classifier. Shen et al. (Shen and Zhao, 2014) adopted two machine learning methods in their saliency model. A neural network is used as feature extractor and SVM is used to perform classification. To my best knowledge, there is no saliency prediction model that uses one learning model only.

In view of this, I propose an Adaptive Saliency Model (ASM) for image saliency prediction in this chapter. Different from most learning-based saliency models using two or hybrid learning techniques, ASM uses a deep architecture to automatically learn low-level and high-level features from training images, and then learn the mapping between feature maps and the final saliency maps.
Figure 3-1. The framework of my proposed CNN-based adaptive saliency model.
3.2 Adaptive Saliency Model (ASM)

Figure 3-1 shows the hierarchical framework of my deep structure adaptive saliency model. The architecture consists of six layers in which four layers are hidden layers. Two of the hidden layers are convolutional layers and the other two are subsampling layers. The deep structure model takes the preprocessed image as the input and generates a two-dimension gray-level saliency map as the output. This two-dimensional output is regarded as the raw saliency prediction result. The low-level and high-level visual feature detectors representing saliency of regions learned during the training process are stored in the kernels of convolutional layers. After feature extraction, the last layer is to learn the mapping between the extracted visual features and the raw saliency prediction result. In my deep architecture, the boundary between feature extractors and mapping (classification) is relatively fuzzy compared to other neural networks such as convolutional neural networks (CNN) and stacked autoencoders (SAE).

3.2.1 Preprocessing

The input data of ASM is derived from the original image in RGB format. Before being fed into the input layer, the size of input image is resized to 396 x 300 which has the same size as that of each input channel. The raw data $x_{raw}$ is normalized to have a maximum value of one and a minimum value of zero.
The normalized data $x_{nor}$ is down-sampled into a lower resolution with several scales $s$ while the size remains unchanged, as shown in Figure 3-2.

$$x_i = \text{down}(x_{nor}, s), s \in \{2, 4, 8, 16, \ldots\}$$  \hspace{1cm} (3.2)

The down-sampled data in different scales are stocked channel by channel to form the final input data $x_i^j$ ($i=1,2,\ldots,N_1$ where $N_1$ is the number of channels).

Examples of an input image with different scales are shown in Figure 3-3.
Figure 3-3. Examples of down-sampled images at different scales.
3.2.2 Neural Layers

3.2.2.1 Input Layer

The input to the deep neural network (Input layer I1) is divided into multiple channels. The number of channel $N_1$ is determined by the number of scales used in the preprocessing step. The RGB components of the original and down-sampled images are fed into different channels. Each channel shares the same size of 396 by 300 in pixel. For instance, having a multi-scale factor of 4, normalized images and down-sampled images with $s \in \{2,4,8\}$ are fed into the input layer. Since there are RGB components for each image, altogether, the input layer has 12 (4x3) channels. If the multi-scale factor is 5, then we have $s \in \{2,4,8,16\}$, and 15 (5x3) channels are used.

3.2.2.2 Convolutional Layers

Two convolutional layers C2 and C4 are utilized in my model. A number of feature maps are generated in a convolutional layer where each feature map represents the combination of the filtered responses from the previous layer. The connections between I1 and C2 are shown in Figure 3-4.
As shown in Figure 3-4, C2 is the first convolutional layer in which all channels in I1 have their own contribution to every feature maps in C2. The data in a channel is convolved with several 5 x 5-sized kernels. Therefore each kernel provides 25 trainable parameters. The number of kernels for each channel is determined by the number of feature maps N_2. Altogether, N_1 x N_2 kernels are to be trained to generate N_2 low-level feature maps. Due to the convolutional operation, the size of each feature map in this layer is reduced to 392 x 296. Figure 3-4 shows the computational steps and the connections between layers I1 and C2. The feature maps are generated with a sigmoid function defined as

$$ x_j^L = \text{sigm} \left( \sum_i x_i^{L-1} \otimes k_{ij}^L + b_j^L \right) $$

(3.3)

where $x_j^L$ represents output feature map $j$ in layer $L$, $k_{ij}^L$ stands for kernel that connecting input channel $i$ and output feature map $j$ in layer $L$. Symbol $\otimes$
stands for convolution operation and bias is shown as \( b \).

\[
\text{sign}(x) = \frac{1}{1 + e^{-x}}
\] (3.4)

Examples of the feature maps in C2 are shown in Figure 3-5:

![Feature Maps in C2](image1)

Figure 3-5. An input image (a) and examples of the feature maps in C2 layer (b)-(e).

C4 is the second convolutional layer where all the feature maps in S3 are connected to each feature map in C4. Same as the kernels in C2, each of the kernels in this layer is a 5 x 5 matrix. The number of kernels for each subsampled feature map in S3 is determined by the number of feature maps \( N_2 \) in C2. Totally \( N_2 \times N_4 \) kernels can be trained to learn high-level features. The connections between S3 and C4 are the same as the connections between I1 and C2. Each feature map in layer C4 is of size of 192 x 144. Examples of the feature maps in C4 are shown in Figure 3-6.
3.2.2.3 Subsampling Layers

The subsampling layer in my model is an average pooling layer which reduce the dimensionality of data from the previous layer. No trainable parameters are contained in a subsampling layer since it only downsamples the feature maps by a scale. Two convolutional layers are followed by two subsampling layers.

S3 is the first subsampling layer which takes the output of C2 as the input. This layer reduces the data dimensionality with a scale $f = 2$. As a consequence, size of a feature map in S3 becomes $196 \times 148$. Each feature map in C2 contributes to only one down-sampled feature map in S3 with $N_3 = N_2$. The computation is given below:

$$ x^i_{\ell}(m,n) = \frac{1}{f^2} \sum_{a=0}^{f-1} \sum_{b=0}^{f-1} x^{i-1}_{\ell}(fm-a, fn-b) $$

(3.5)
where $x_i^L$ is the output of channel $i$ in layer $L$, $f$ is the scale and $m,n$ represents the coordinates of pixels belonging to $x_i^L$.

S5 is the second subsampling layer which shares the same function as S3 with a scale factor $f=3$. It is located between C4 and the output layer so that $N_5$ is equal to $N_4$. The size of feature maps in S5 is reduced to $64 \times 48$.

### 3.2.2.4 Output Layer

The weighted summation of all the feature maps in S5 is the input to the output layer. The output of this layer is the raw saliency map from my learning-based model. The size of output is the same as a feature map in S5 which is $64 \times 48$. The sigmoid function is used as the active function and the computation is given below

$$x^L = \text{sigm}\left( \sum_i x_{i}^{L-1} \times k_i^L \right)$$

(3.6)

where $k_i^L$ is the trainable weights.

Being different from most of other deep neural networks, my model produces a 2-dimensional output that carries locational information among output nodes. With this special format of output, the whole neural network could be categorized as a function approximator with a large number of input and output nodes instead of a classifier.

### 3.2.3 Post processing

The size of an input image in my model is $396 \times 300$ pixels and the size of
output in my model is 64 x 48 pixels. Since the ratio of output is 4:3 and the ratio of input is close to 4:3, we can use the output as saliency map without changing its size.

In the neural network, the sigmoid function is adopted in the output layer as active function. As a consequence, 0 or 1 is hardly to be reached in the output as shown in Figure 3-7. Even though the data feed into node in the output layer could reach a value of -4, it is still impossible to output a zero since the sigmoid function is used. Therefore, these small but non-zero values are regarded as noise. To refine the raw saliency map, a threshold equal to 0.12 (\( \text{sigm}(-2) \approx 0.12 \)) is firstly applied to eliminate noises generated by the neural network. Then the whole saliency map is normalized to guarantee the maximum value of 1.

![Figure 3-7. Plot of the sigmoid function.](image)
3.3 Algorithm for Training ASM Deep Structure

3.3.1 Resilient propagation (Rprop)

Since a deep neural network has multiple hidden layers, it suffers from the long-term dependency problem, i.e., the gradient reduces gradually when the error is back-propagated from the output nodes to the hidden layers. To alleviate the problem, I use the Rprop algorithm introduced by Riedmiller (Riedmiller and Braun, 1993) to train my deep structure saliency model.

In the Rprop algorithm, the weight updating is no longer computed from gradients. They are generated from their previous values and the signs of gradients instead. After an update value is initialized, it grows when its gradient keeps the same sign within two continues iterations and it decreases when the gradient changes its sign within two continues iterations. Even though the values of gradients still attenuate during the back-propagation process, the signs of the gradients will not suffer from any negative effect. Therefore, adopting this algorithm will substantially solve the long-term dependency problem.

The update values of weights and biases are obtained in Equation (3.7) where the update values are forced within a range between $10^{-6}$ and 50.

$$dW^{(i)}_{ij} = \begin{cases} 1.2 \times dW^{(i-1)}_{ij}, & \text{if } \gamma > 0 \\ 0.5 \times dW^{(i-1)}_{ij}, & \text{if } \gamma < 0 \\ dW^{(i-1)}_{ij}, & \text{if } \gamma = 0 \end{cases}$$

$$\gamma = \frac{\delta E^{(i-1)}}{\delta W_{ij}} \times \frac{\delta E^{(i)}}{\delta W_{ij}}$$
where \( \frac{\delta E^{(t)}}{\delta w_{ij}} \) represents the gradient of the weight connecting nodes \( i \) and \( j \) in iteration \( t \), \( dw_{ij}^{(t)} \) is the update value of the weight, and \( \text{sign}() \) is the sign function. The initial value \( \frac{\delta E^{(0)}}{\delta w_{ij}} \) is set to 0.1 as it is recommended in Riedmiller’s research (Riedmiller and Braun, 1993).

\[
\begin{align*}
\frac{\delta E^{(t)}}{\delta w_{ij}} &= \text{sign} \left( \frac{\delta E^{(t)}}{\delta w_{ij}} \right) \times dw_{ij}^{(t)} , \text{if } \gamma \geq 0 \\
\frac{\delta E^{(t)}}{\delta w_{ij}} &= \text{sign} \left( \frac{\delta E^{(t)}}{\delta w_{ij}} \right) \times dw_{ij}^{(t-1)} , \text{if } \gamma < 0 \\
\text{then, set } &\frac{\delta E^{(t)}}{\delta w_{ij}} = 0 , \text{if } \gamma < 0 \\
\gamma &= \frac{\delta E^{(t-1)}}{\delta w_{ij}} \times \frac{\delta E^{(t)}}{\delta w_{ij}}
\end{align*}
\]

The update rule is the same as the delta rule when \( \gamma \) is non-negative. However, if \( \gamma \) is found negative, it will reverse the last change made to the corresponding weight. The gradient is reset to 0 once \( \gamma \) becomes negative to prevent a double reversion.
3.4 Experimental Results and Discussion

3.4.1 Datasets and Experiment Setup

Experiments were conducted on the Object and Semantic Images and Eye-tracking (OSIE) data set (Xu et al., 2014). The OSIE dataset contains 700 natural indoor and outdoor color images from Flickr and Google. All the images share the same size of 800x600 in pixels. A large portion of them have multiple dominant objects in each image. The experiments were targeted at acquiring a proper trained ASM which has an image as the input and a saliency map of the image as the output. The input of my model is an image from OSIE data set. The target output is the intermediate result of Itti model on the image, i.e., the result from the linear combinations in the hierarchical process of Itti’s model (Itti et al., 1998) as shown in Figure 2-3. Two reasons to use the output of Itti model: (1) Itti model is a classical and most popular pixel-based bottom-up saliency prediction model without involving learning process; (2) my model shares structural similarity with Itti model.

In Itti model, several images are generated using Gaussian pyramids with several scales, which shares the similarity with my model. My model also has the capability to perform the most important step of Itti model in extracting three channels of low-level features using linear filters.

The data set were partitioned into two subsets: one for training and the other for testing. Specifically, 100 images were used as training samples and another 200
images were used as test samples. In general, 100 training images are not enough to train such a large neural network. However, my model has a very large number of input nodes (118,800 nodes) and therefore the training process is very time consuming. A single image would generate 116,032 sets of data for each kernel in layer C2. All sets of data will make contributions to weight updating during BP as shown in Figure 3-8. As a reference, LeNet-5 (LeCun et al., 1998) generates only 784 sets of data. By applying multiple scales in the preprocessing step, a single image generates much more sets of data compared to the traditional CNN. With the computer power that I had (without using GPU), it can handle the training set of 100 images only. In the training process, a large amount of physical memories is required. Specifically, for 100 images, on average 28 GB memories is required. Therefore, together with the system usage, the peak value could reach 31.4 GB. For a 64-bit Windows based computer, at most 32 GB memories can be supported. Consequently, there is difficulty in training more than 100 images at a time without any Graphics Processing Unit (GPU). Actually, training my ASM with 100 training images took around 365 seconds for each iteration. In other words, training my ASM for 100 iterations would take more than 10 hours. It had been a long time to train my model for different parameter settings even for 100 images only.
During the experiments, the numbers of channels and features maps in the neural network were not fixed. Instead, experiments were conducted for different settings in order to find the most appropriate parameters.

### 3.4.2 Metrics for Performance Evaluation

Two metrics are used to evaluate the performance of my model: Histogram Intersection (HI) and Receiver Operating Characteristic (ROC).

Since both the target outputs and the saliency maps generated by ASM are in continuous values, ROC has its disadvantage in precisely evaluating the performance. Hence, the histogram intersection is chosen to be the primary evaluation metric to measure the similarity between the target output and the real output. HI is also used as one of important metrics to evaluate the performances of saliency models in MIT saliency benchmark (Bylinskii et al.). The Histogram Intersection is defined as follows:
\[ HI = \sum_{i,j} \min \left[ \text{Norm}(H(r_{ij})), \text{Norm}(H(t_{ij})) \right] \]  

(3.9)

where \( r_{ij} \) stands for the real output of the pixel at coordinate \((i,j)\), \( t_{ij} \) denotes the target output of the pixel and function \( H(\bullet) \) is defined as follows:

\[
\Delta = \max_{\forall m,n} (x_{mn}) - \min_{\forall m,n} (x_{mn})
\]

\[
H(x_{ij}) = \begin{cases} 
1, & \text{if } \Delta = 0 \\
\frac{x_{ij} - \min_{\forall m,n} (x_{mn})}{\Delta}, & \text{otherwise}
\end{cases}
\]  

(3.10)

where \( x_{ij} \) stands for the intensity of pixel \((i,j)\) in the given image. In Equation(3.9), \( \text{Norm}(\bullet) \) is defined as follows:

\[
\text{Norm}(x_{ij}) = \begin{cases} 
\frac{x_{ij}}{\sum_{m,n} (x_{mn})}, & \text{if } \max_{\forall m,n} (x_{mn}) > 0 \\
1, & \text{if } \max_{\forall m,n} (x_{mn}) = 0
\end{cases}
\]  

(3.11)

where \( x_{ij} \) stands for the pixel intensity at coordinate \((i,j)\). This function is used to ensure the summation of all x’s is 1 for any image. Note that both the real output and the target output values are non-negative.

From the definition of HI in the above equations, it is easy to know that HI can measure the similarity between two images with pixel values from 0 to 1. Zero HI means that two images are completely converse while one means that the contents of two images are exactly the same after normalization.
3.4.3 Experiments on Network Structures and Resilient Propagation Algorithm

In the early stage of my research, the chosen structures of ASM were different from that presented in Section 3.2. Experiments have been conducted to test three possible structures as shown in Figure 3-9. The most suitable structure was employed in my ASM to obtain the best performance.

The most important difference among these three structures is on the last layer. For the first structure shown in Figure 3-9(a), the last layer is a fully connected layer with size of 64 x 48. Thus, there are totally more than 9 million connections with trainable weights. This fully connected layer is a common setting for a CNN based classification model with the input being the S5 output. The sizes of channels and feature maps, the number of kernels, and scale factors for three model structures are shown in Table 3-1.

![Figure 3-9. Simplified diagrams of three possible ASM structures.](image-url)
Table 3-1. The parameter setting of the first structure.

<table>
<thead>
<tr>
<th></th>
<th>Input Layer</th>
<th>C2</th>
<th>S3</th>
<th>C4</th>
<th>S5</th>
<th>Output Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>396x300</td>
<td>392x296</td>
<td>196x148</td>
<td>192x144</td>
<td>64x48</td>
<td>64x48</td>
</tr>
<tr>
<td># Maps</td>
<td>24</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td># Kernels</td>
<td>240</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># Weights</td>
<td>6,000</td>
<td>2,500</td>
<td></td>
<td></td>
<td></td>
<td>9,437,184</td>
</tr>
<tr>
<td>Scale factor</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td></td>
<td>3</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 3-10. The individually connected output layer in the second structure.

For the second structure (Figure 3-9(b)), the last layer is an individually connected layer. There are 30,720 connections between S5 and the last layer. As shown in Figure 3-10, each connection connects two nodes that shares same coordinates in the maps they belong to. The input of the output layer is the outputs of the ten nodes in S5 with the same coordinates. Table 3-2 shows the network parameters of the second structure.
Table 3-2. The parameter setting of the second structure.

<table>
<thead>
<tr>
<th></th>
<th>Input Layer</th>
<th>C2</th>
<th>S3</th>
<th>C4</th>
<th>S5</th>
<th>Output Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>396x300</td>
<td>392x296</td>
<td>196x148</td>
<td>192x144</td>
<td>64x48</td>
<td>64x48</td>
</tr>
<tr>
<td># Maps</td>
<td>24</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td># Kernels</td>
<td>240</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># Weights</td>
<td>6,000</td>
<td>2,500</td>
<td></td>
<td></td>
<td></td>
<td>30,720</td>
</tr>
<tr>
<td>Scale factor</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

For the third structure (Figure 3-9(c)), the last layer performs a weighted summation of ten feature maps in S5. This structure is the same as the one introduced in Section 3.2. The sizes of channels and feature maps, number of kernels and scale factors are shown in Table 3-3.

Table 3-3. The parameter setting of the third structure.

<table>
<thead>
<tr>
<th></th>
<th>Input Layer</th>
<th>C2</th>
<th>S3</th>
<th>C4</th>
<th>S5</th>
<th>Output Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>396x300</td>
<td>392x296</td>
<td>196x148</td>
<td>192x144</td>
<td>64x48</td>
<td>64x48</td>
</tr>
<tr>
<td># Maps</td>
<td>24</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td># Kernels</td>
<td>240</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># Weights</td>
<td>6,000</td>
<td>2,500</td>
<td></td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Scale factor</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
Table 3-4. Six trainings of three models and two training algorithms.

<table>
<thead>
<tr>
<th>Training</th>
<th>Type of the Last Layer</th>
<th>Training Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Full Connection</td>
<td>GD</td>
</tr>
<tr>
<td>B</td>
<td>Full Connection</td>
<td>Rprop</td>
</tr>
<tr>
<td>C</td>
<td>Individual Connection</td>
<td>GD</td>
</tr>
<tr>
<td>D</td>
<td>Individual Connection</td>
<td>Rprop</td>
</tr>
<tr>
<td>E</td>
<td>Summation Layer</td>
<td>GD</td>
</tr>
<tr>
<td>F</td>
<td>Summation Layer</td>
<td>Rprop</td>
</tr>
</tbody>
</table>

In order to test the effectiveness of the Rprop algorithm in training the three different structures of ASM, I used both the gradient decent algorithm and Rprop to train these models. Six trainings shown in Table 3-4 were carried out.

Each training process was initialized with random weights. My model was trained on the OSIE data set. The error (loss) function is defined as follows:

\[
E = \sum (x_r - x_t)^2
\]

where \(x_r\) denotes the real output and \(x_t\) stands for the target output. The error charts are shown in Figure 3-11.

Obviously, the error curves show that both training B and training D fail to converge, suggesting that the GD algorithm cannot train the first two saliency models. Even all the trainings of A, C, E and F seem converge, I need to further evaluate the performance more closely. Hence, the actual outputs from these six trainings are extracted and visualized for the image shown in Figure 3-12(g).
I noted that the output saliency maps from trainings A and C had only minor changes regardless of whatever input image feed into the model. Precisely, the output saliency maps are actually the average of all training target outputs as shown in Figure 3-12(a) and (c). The reason why the training model can produce such an output...
is that the trained weights are trapped into a local optimum. The smaller difference between the real output and the average target outputs, the smaller error of each output node. In this case, even though the training error has a trend to converge as shown in Figure 3-11(a) and (c), the trained models are still meaningless for the task.

Figure 3-12. Visualized outputs from six training processes together with the original image and the target output.

As for training E, the output did change when different input images were seen.
As shown in Figure 3-12(e), the result is far from the target output. More experiments on case E with different combinations of $N_2$ and $N_4$ were conducted to test if this model was workable without Rprop.

Table 3-5 shows the average output HI’s of every combination of $N_2$ and $N_4$ used in the experiments. From the table, I can see that some combinations have HI’s equal to 0.001. It is because the model generates an adverse saliency map, that is, a salient region is falsely predicted as a non-salient region and a non-salient region is falsely regarded as a salient one. For those combinations with higher HI’s, the generated saliency map is similar to the one shown in Figure 3-12(e). From these results, it is obvious that without using the Rprop algorithm, my model cannot be trained properly to generate a meaningful saliency map, especially in some combinations of $N_2$ and $N_4$.

The major problem of GD is the long-term dependency problem. The GD causes more problems in my ASM because (1) two-dimensional map output in my model has a large number of output nodes and (2) the amplitudes of gradients reduce drastically during back-propagation. The majority of deep neural networks with a gradient decent training algorithm suffer from problem (2) more or less. However, in my ASM, the problem is more serious because of problem (1). Therefore, the Rprop algorithm is used in training my ASM.
Table 3-5. The average output HI’s with different number of kernels in C2 as well as C4 and the number of Channels in I1 (without using Rprop).

<table>
<thead>
<tr>
<th>Number of kernels in C4 (N₄)</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.609</td>
<td>0.001</td>
<td>0.597</td>
<td>0.593</td>
<td>0.001</td>
<td>0.609</td>
</tr>
<tr>
<td>11</td>
<td>0.555</td>
<td>0.577</td>
<td>0.001</td>
<td>0.606</td>
<td>0.602</td>
<td>0.611</td>
</tr>
<tr>
<td>12</td>
<td>0.001</td>
<td>0.610</td>
<td>0.585</td>
<td>0.001</td>
<td>0.522</td>
<td>0.605</td>
</tr>
<tr>
<td>13</td>
<td>0.001</td>
<td>0.001</td>
<td>0.605</td>
<td>0.614</td>
<td>0.611</td>
<td><strong>0.619</strong></td>
</tr>
<tr>
<td>14</td>
<td>0.549</td>
<td>0.492</td>
<td>0.001</td>
<td>0.607</td>
<td>0.611</td>
<td>0.583</td>
</tr>
<tr>
<td>15</td>
<td>0.609</td>
<td>0.615</td>
<td>0.531</td>
<td>0.597</td>
<td>0.611</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Number of Channels in I1 = 24, average HI = 0.4427

### 3.4.4 Determination of the Number of Channels and the Number of Feature Maps

Parameter setting affects much on the final result of my model. Determining structure parameters for training on a given task is an indispensable step.

The sizes of channels and feature maps used during this experiment were the same as that discussion in Section 3.2.2. Different combinations of N₂, N₄ and multi-scale factors were experimentally determined.

Rprop algorithm was applied to train my model. For each set of parameters, the
model was trained for 50 iterations with a random initialization of weights. The average HI’s between the real output and the target output on the test set are shown in Table 3-6. Table 3-6 is divided into 4 quarters with each sharing the same multi-scale factor. As discussed before, all R, G and B channels were down-sampled for each scale, that is, the number of channels being 3 multiplied by the multi-scale factor. For instance, when multi-scale is 5, totally 15 channels were utilized in I1. The underlined figures in bold are the best results in each quarter.

As shown in Table 3-6, the average HIs for four different multi-scale factors are 0.6125, 0.6264, 0.6275 and 0.6352, respectively. The result shows a tendency of a better result for a larger multi-scale factor. Meanwhile it also shows that the more kernels in the convolutional layer, the better performance it appears. However, this trend becomes weak when \( N_4 \) is larger than 12. The most possible reason for this phenomenon is that a deep neural network with more free parameters needs more iterations to train. Hence, it is possible to improve the performance of the model with more kernels if a longer training period is affordable.

Compared with the results shown in Table 3-5, the results in Table 3-6 (d) are much better both averagely and individually. To verify if the generated output saliency maps are meaningful, the output saliency maps are visualized. As shown in Table 3-6, the combination of \( N_2-N_4 \) works the best at 15-13 with a multiple-factor of 8. Therefore, I will use this configuration to generate output saliency maps.

Figure 3-15 shows (a) the original images, (b) the intermediate result of Itti
model which is used as the target output for my model, and (c) the real output produced by my model. With the Rprop algorithm, my model was successfully trained to generate valid 2-dimensional saliency maps.

![ROC Curve](image1)

**Figure 3-13.** ROC curve on test samples.

![Histogram Intersections](image2)

**Figure 3-14.** Histogram intersections on training samples (a) and test samples (b).

For the model with $N_2=15$ and $N_4=13$, the ROC curve and bar chart of individual HI’s for both training and test samples are shown in Figure 3-13. Since my task is not a classification type, ROC cannot reflect the performance precisely. As shown in Figure 3-14 (a) and (b), both bar charts are similar to Gaussian distribution.
The average HI’s of training and testing samples are 0.71 and 0.71, respectively.
Table 3-6. The average HI’s with different structure parameter settings

<table>
<thead>
<tr>
<th>Number of kernels in C2 (N2)</th>
<th>Number of kernels in C4 (N4)</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.604</td>
<td>0.627</td>
<td>0.596</td>
<td>0.623</td>
<td>0.562</td>
<td>0.611</td>
<td>0.644</td>
<td>0.596</td>
<td>0.631</td>
<td>0.644</td>
<td>0.596</td>
<td>0.644</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.588</td>
<td>0.627</td>
<td>0.610</td>
<td>0.608</td>
<td>0.610</td>
<td>0.625</td>
<td>0.628</td>
<td>0.675</td>
<td>0.592</td>
<td>0.619</td>
<td>0.618</td>
<td>0.634</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.618</td>
<td>0.619</td>
<td>0.616</td>
<td>0.635</td>
<td>0.536</td>
<td>0.642</td>
<td>0.601</td>
<td>0.644</td>
<td>0.642</td>
<td>0.603</td>
<td>0.672</td>
<td>0.635</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.587</td>
<td>0.632</td>
<td>0.620</td>
<td>0.621</td>
<td>0.626</td>
<td>0.603</td>
<td>0.624</td>
<td>0.602</td>
<td>0.616</td>
<td>0.616</td>
<td>0.633</td>
<td>0.625</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.607</td>
<td>0.590</td>
<td>0.627</td>
<td>0.581</td>
<td>0.630</td>
<td>0.626</td>
<td>0.611</td>
<td>0.580</td>
<td>0.697</td>
<td>0.620</td>
<td>0.642</td>
<td>0.638</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.622</td>
<td>0.609</td>
<td>0.633</td>
<td>0.631</td>
<td>0.620</td>
<td>0.630</td>
<td>0.594</td>
<td>0.634</td>
<td>0.636</td>
<td>0.630</td>
<td>0.639</td>
<td>0.593</td>
<td></td>
</tr>
</tbody>
</table>

Number of scales = 5, average HI = 0.6125

Number of scales = 6, average HI = 0.6264

<table>
<thead>
<tr>
<th>Number of kernels in C2 (N2)</th>
<th>Number of kernels in C4 (N4)</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.564</td>
<td>0.667</td>
<td>0.574</td>
<td>0.600</td>
<td>0.663</td>
<td>0.627</td>
<td>0.652</td>
<td>0.629</td>
<td>0.691</td>
<td>0.638</td>
<td>0.639</td>
<td>0.648</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.609</td>
<td>0.609</td>
<td>0.641</td>
<td>0.630</td>
<td>0.638</td>
<td>0.638</td>
<td>0.636</td>
<td>0.547</td>
<td>0.637</td>
<td>0.612</td>
<td>0.608</td>
<td>0.689</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.631</td>
<td>0.653</td>
<td>0.561</td>
<td>0.628</td>
<td>0.636</td>
<td>0.618</td>
<td>0.636</td>
<td>0.661</td>
<td>0.639</td>
<td>0.682</td>
<td>0.565</td>
<td>0.631</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.656</td>
<td>0.573</td>
<td>0.641</td>
<td>0.635</td>
<td>0.611</td>
<td>0.643</td>
<td>0.604</td>
<td>0.613</td>
<td>0.665</td>
<td>0.648</td>
<td>0.618</td>
<td>0.635</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.593</td>
<td>0.643</td>
<td>0.633</td>
<td>0.677</td>
<td>0.667</td>
<td>0.678</td>
<td>0.634</td>
<td>0.612</td>
<td>0.645</td>
<td>0.648</td>
<td>0.639</td>
<td>0.626</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.639</td>
<td>0.660</td>
<td>0.596</td>
<td>0.622</td>
<td>0.631</td>
<td>0.624</td>
<td>0.608</td>
<td>0.658</td>
<td>0.626</td>
<td>0.710</td>
<td>0.602</td>
<td>0.639</td>
<td></td>
</tr>
</tbody>
</table>

Number of scales = 7, average HI = 0.6275

Number of scales = 8, average HI = 0.6352
Figure 3-15. A comparison of the saliency maps generated by my model with Itti saliency maps on the OSIE data set.
3.4.5 Experiments on Size of Channels and Feature Maps

The experiments on the size of channels and feature maps were carried out. For a 4:3 input image, the model will produce a 4:3 saliency map. However, due to the characteristics of convolutional neural networks, the input and output cannot be 4:3 at the same time. A sacrifice was made to the input and I kept the output to be of an aspect ratio of 4:3. Of course, keeping the input image with a 4:3 aspect ratio and sacrificing the output is also an option. Two possible settings are shown in Table 3-7.

Table 3-7. Parameter settings for channels and feature maps in different layers

<table>
<thead>
<tr>
<th></th>
<th>Input Layer</th>
<th>C2</th>
<th>S3</th>
<th>C4</th>
<th>S5</th>
<th>Output Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>396 x 300</td>
<td>392 x 296</td>
<td>196 x 148</td>
<td>192 x 144</td>
<td>64 x 48</td>
<td>64 x 48</td>
</tr>
<tr>
<td>(b)</td>
<td>400 x 300</td>
<td>396 x 296</td>
<td>198 x 148</td>
<td>194 x 144</td>
<td>72 x 97</td>
<td>72 x 97</td>
</tr>
</tbody>
</table>

Adopting the settings shown in Table 3-7(b), my model was trained for 150 iterations with the Rprop algorithm. The model with a structure setting of N₂=15 and N₄=13 as well as N₂=15 and N₄=15 was trained. The average HI’s are shown in Table 3-8. These two combinations of N₂ and N₄ are chosen because one of them achieves the best performance as reported in Section 3.4.4 and the other has a potential to have better performance than that reported in Section 3.4.4. I note that the model with N₂=15 and N₄=13 produces the similar results in both the training and test sets with no over-training phenomenon observed. The model with a setting shown in Table 3-7(a) was trained for 150 iterations as well. Increasing the length of training process to 150 iterations leads to a significant improvement in the performance. As shown in
Table 3-8, the average HI’s of the training and test sets can reach 0.7502 and 0.7449, respectively. The experiment on the model with N$_2$=15 and N$_4$=15 shows slightly better average HI’s of 0.7585 and 0.7435 on the training and test set, respectively.

Table 3-8. HI’s for the training and test set with different N$_2$ an N$_4$.

<table>
<thead>
<tr>
<th>N$_2$</th>
<th>N$_4$</th>
<th>Output Size</th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>13</td>
<td>64 x 48</td>
<td>0.7502</td>
<td>0.7449</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>64 x 48</td>
<td>0.7585</td>
<td>0.7435</td>
</tr>
<tr>
<td>15</td>
<td>13</td>
<td>97 x 72</td>
<td>0.7381</td>
<td>0.7297</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>97 x 72</td>
<td>0.7791</td>
<td>0.7518</td>
</tr>
</tbody>
</table>
3.5 Summary

In this chapter, I propose a deep architecture ASM for saliency prediction. ASM consists of two convolutional layers, two subsampling layers and a full-connection layer. Both convolutional layers contain several kernels used to learn and detect visual features. The kernels in the first convolutional layer are used to extract low-level visual features while the kernels in the second convolutional layer are used to extract high-level visual features. To overcome the long-term dependency problem encountered in the traditional gradient descent algorithm, an enhanced back-propagation algorithm called Rprop was adopted to train ASM. The experiments were conducted on three possible structures of ASM with different parameter settings. A workable ASM structure is finally identified with plausible parameter settings suggested. Experimental results show that my model can produce meaningful two-dimensional saliency maps which are close to the target saliency maps generated from Itti’s model.
Chapter 4 Deep Learning for Salient Object Detection

4.1 Background

Salient object detection aims to locate objects that differ from their surroundings in images. In general, saliency prediction could be further used to perform salient object detection (or the other way around). Pixel-based and region-based are two main strategies for saliency prediction. Both of them have their advantages and disadvantages. When an image has a large number of objects, a pixel-based approach can achieve a satisfactory performance. However, for an image that contains single or a few objects, a pixel-based approach may not work well because it does not take into account the global information (pixels in the background could contribute as noise). When an image consists of a known number of objects, the region-based approach is more suitable. Segmentation is an important method to locate objects. However, under-segmentation may lead to the inclusion of portions of the background in the object regions. On the other hand, over-segmentation results in incorrectly marking non-salient region as salient one. Moreover, different from a pixel-based approach, the performance of a region-based approach could be influenced significantly by accuracy of segmentation. Yet, a pixel-based approach and a region-based approach might be complementary to each other. In light of this, in this chapter, I propose a hybrid model that combines pixel-based and region-based
approaches for salient object detection. In addition, a learning-based methodology is adopted for the pixel-based approach for object detection.

In this chapter, I first introduce in Section 4.2.1 the modified ASM, a deep learning model for salient object detection and report its performance in Section 4.3.3. In Section 4.3.4, I show the superiority of a hybrid approach compared with individual methods.
Figure 4-1. The framework of the convolutional neural network based component of my proposed salient object detection model.
4.2 Model Structure

Salient object detection is different from saliency prediction. Saliency prediction aims to acquire saliency values of every region in a given image (scene). However, Salient Object Detection (SOD) intents to locate the object with high saliency. The common part of these two tasks is that computing saliency values is a necessity. A saliency prediction model could be modified to perform salient object detection. Therefore, ASM proposed in Chapter 3 is modified to perform salient object detection (Figure 4-1). Since ASM is originally designed for saliency prediction, it is difficult to eliminate non-object but salient regions. To overcome this drawback, a region-based method is used together with ASM to form a hybrid SOD model. ASM is responsible for computing saliency values of pixels while potential Region-of-Interest (p-ROI) approximation component is responsible for finding a potential region where the salient object probably locates at.

Figure 4-2 shows the flowchart of my procedure which detects salient objects in given scenes with a combination of ASM and p-ROI approximation. An input image is fed to both components of the models. ASM component will produce a saliency map. The p-ROI approximation component will generate the most possible salient region for the salient object. With mutual validation between the saliency map and p-ROI, the final region with the salient object will be produced. Note that my model will detect the most dominant object on each image.
4.2.1 ASM Component

The ASM structure applied in this chapter is the same as that discussed in Chapter 3 except for parameter setting. The details of setting and parameters of the ASM are shown in Table 4-1. If necessary, an input image is rotated 90 degree and stretched to fit the size of the input layer (400 x 300). The rotation is to ensure that an image has its width larger than its height before stretching in order to alleviate pixel loss in the preprocessing step. Then the raw data is normalized to have the maximum value of one and the minimum value of zero as defined in Equation(3.1).

The normalized data is down-sampled into lower resolutions with three different
scales $s$ and then up-sample back to its original size, as shown in Figure 4-4. This up-sample process is different from the one discussed in Chapter 3. I applied the bilinear interpolation to extenuate the mosaic effect caused by down-sampling. Without the mosaic effect, the edge of salient objects could be more accurate. The bilinear interpolation is defined as

$$P = f(x,y) = \frac{y-y'}{y_2-y_1} f(x,y_1) + \frac{y'-y_1}{y_2-y_1} f(x,y_2)$$
$$f(x,y_i) = \frac{x-x_i}{x_2-x_1} Q_{i1} + \frac{x-x_i}{x_2-x_1} Q_{i2}$$

where $P$ and $Q_{ij}$ are the coordinators shown in Figure 4-3.

![Figure 4-3. Bilinear Interpolation used in up-sampling.](image)

Examples of input images with different scales are shown in Figure 4-5. In Figure 4-6, a comparison is given on the results of a same input image using two different down-sample pre-processes.

$$x_i = \text{down}(x_{nor}, s)$$

Normalized images and down-sampled images with $s \in \{2, 4, 8\}$ are fed into the input layer. Consequently, altogether, 12 (3x4) channels are used in the input layer.
Table 4-1. Parameter setting of ASM for saliency object detection.

<table>
<thead>
<tr>
<th></th>
<th>Input Layer</th>
<th>C2</th>
<th>S3</th>
<th>C4</th>
<th>S5</th>
<th>Output Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>400x300</td>
<td>396x296</td>
<td>198x148</td>
<td>194x144</td>
<td>97x72</td>
<td>97x72</td>
</tr>
<tr>
<td># Maps</td>
<td>12</td>
<td>15</td>
<td>15</td>
<td>13</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td># Kernels</td>
<td>180</td>
<td>195</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td># Weights</td>
<td>4,500</td>
<td>4,875</td>
<td>13</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Kernel size</td>
<td>-</td>
<td>5 x 5</td>
<td>-</td>
<td>5 x 5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Scale factor</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
</tbody>
</table>

Two convolutional layers (C2 and C4) have the same kernel size of 5 by 5. Thus, totally 4,500 trainable parameters are in C2 while 4,875 trainable parameters are in C4. Meanwhile, the size of feature maps is reduced to 396 x 296 and 194 x 144 correspondingly.

Figure 4-4. Example of data down-sampling then up-sampling back to its original size.
Figure 4-5. Examples of down-sampled images at different scales.
Figure 4-6. A comparisons between down-sampling methods introduced in Chapter 3 and Chapter 4. (a) Use the method introduced in Chapter 3 and (b) use the method introduced in Chapter 4.

The feature maps are derived with the sigmoid function as defined in Equations (3.3) and (3.4). Subsampling layers (S3 and S5) are also the average pooling layers which reduce the dimensionality of data with a scale factor $f = 2$. No trainable parameters are contained in a subsampling layer. The size of feature maps becomes $198 \times 148$ in S3 and it becomes $97 \times 72$ in S5. The derivation function is shown
\[
x_i^L(m,n)=\frac{1}{4}\sum_{a=0}^{1}\sum_{b=0}^{1}x_{i}^{L-1}(2m-a,2n-b)
\]

where \(x_i^L\) is the output of channel \(i\) in layer \(L\) and \(m,n\) represents the coordinates of pixels belonging to \(x_i^L\).

The weighted summation of all feature maps in S5 is inputted to the sigmoid function to obtain the output. The output of ASM is a raw saliency map. Since the sigmoid function is applied in the ASM, a small threshold (e.g. 0.1) is applied to remove noise.

### 4.2.2 Potential Region of Interest (p-ROI)

Salient regions tend to be continuous and concentrated in natural scenes with dominant objects. To locate a salient region, a polygon could be used to approximate the region.

The p-ROI is a methodology introduced by Liang (Liang et al., 2012) to approximately mark a region in which salient objects potentially locate. In the method, the first step is to perform edge detection. The Canny edge detector is employed to generate an edge map. Then two filters are applied one after another to obtain p-ROI from the edge map. The first filter is used to select the edges of a proper length and the second filter is used to pick the edges of a suitable distribution.

Let \(EM_0\) represents the edge map and \(x_i\) stands for the edges that belonging to the edge map, as shown in Figure 4-7(b).
\[
\mathbf{EM}_0 = \{x_1, x_2, \ldots, x_n\}
\]

Let \(\mathbf{EM}_1\) represents the edge image that has been obtained by the first filter. \(\mathbf{EM}_1\) contains all the edges detected with their lengths larger than the quotient of the perimeter of the edge map and a threshold \(l_x\).

\[
\mathbf{EM}_1 = \left\{ x_i, \forall \|x_i\| > \frac{P}{l_x} \right\}, x_i \in \mathbf{EM}_0
\]

where \(\|x_i\|\) denotes the length of edge \(x_i\) and \(P\) stands for the perimeter of \(\mathbf{EM}_0\).

\(\mathbf{EM}_2\) is the edge image after being filtered by the second filter. The Euclidean distances \(d_i\) between the centroid of \(\mathbf{EM}_1\) and each edge pixel \(p_i\) in \(\mathbf{EM}_1\) is computed. All Euclidean distances \(d_i\) obtained are retained in Set \(D\) which is separated into two clusters by using histogram analysis. One of the clusters, \(S\) contains all smaller distances. \(S\) is the set that contains all the edge pixels whose \(d_i\) belongs to the cluster with small distances. \(\mathbf{EM}_2\) contains all edge pixels \(p_i\) if and only if its \(d_i\) belongs to \(S\) as shown in Figure 4-7(c).

\[
d_i = \|p_i - \left(\sum \forall p_i \in \mathbf{EM}_1\right)\|, D = \{\forall d_i\} \quad \mathbf{EM}_2 = \{\forall p_i \mid d_i \in S\}
\]

Finally, a raw convex polygon is acquired to represent the raw p-ROI. The smallest polygon could be obtained by applying a Delaunay triangulation based convex hull algorithm on the raw p-ROI. This polygon is the final p-ROI approximation. An example is shown in Figure 4-7(d).
Figure 4-7. Comparison among (a) the original image, (b) detected edge map, (c) edge map after filtering, (d) the final mask generated from filtered edge map, (e) masked image, and (f) the salient object circled with green line.
4.3 Experimental Results and Discussion

4.3.1 Datasets and Experiments Setup

Our model was trained and tested on the MSRA dataset (Cheng et al., 2015; T. Liu et al., 2011). The MSRA dataset is widely used in segmentation and salient object detection community. The database provides color images that have only single dominant object. Salient object annotations in terms of consistent bounding boxes provided by 3 to 9 users are contained in the database. The experiments were aimed at training the model to mark the salient object in a given image. Totally 300 images were used in the experiments. Among them, 100 images were used as the training samples and 200 images were used as the test samples. The binary ground truth provided in the MSRA dataset was resized to have the same size as the output layer of my model. The batch training mode was adopted to train the model for 200 iterations.

4.3.2 Metrics for Performance Evaluation

To objectively evaluate the performance of my model, the precision-recall curve is used. The curve is obtained by using a threshold that increases from 0 to 1 with 100 equal intervals. Each threshold leads to a single precision-recall point. By connecting 101 points, a precision-recall curve is plotted. In this measure, the higher the curve raises upwards, the better performance it represents. The Area Under Curve (AUC) value is also adopted as a metric because a quantized result is intuitive and
much more persuasive. Zero AUC means that the generated salient region is completely converse with the ground truth while an AUC of one means that the obtained salient region is perfectly aligned with the ground truth.

A compression was made between my model and other methods. The models compared include Achanta’s method (AC) (Achanta et al., 2008), Attention based on Information Maximization (AIM) (Bruce and Tsotsos, 2009), Context-Aware saliency detection (CA) (Goferman et al., 2012), non-parametric Low-Level Vision Model (IM) (Murray et al., 2011), Itti’s salience model (IT) (Itti et al., 1998), saliency detection by Self-Resemblance (SeR) (Seo and Milanfar, 2009), a Spectral Residual (SR) approach by Hou and Zhang (Hou and Zhang, 2007), Saliency using Natural Statistics (SUN) (Zhang et al., 2008), and Spatially Weighted Dissimilarity (SWD) (Duan et al., 2011). These models are well-performed saliency models in the MSRA dataset. Some of these methods targeted at saliency prediction and others were designed to perform salient object detection. They are different in terms of approaches, but they are all bottom-up models. The same set of images was used in the compression study.

### 4.3.3 Output of Adaptive Saliency Model

Figure 4-8 (a), (b) and (c) show the original image, the real output and the ground truth of a test sample. In this example, the coin should be the salient object and the salient region should cover and only cover the coin. However, the real output
contains some salient regions that do not overlay with salient object. In this task, it is necessary to remove those falsely labeled regions.

![Figure 4-8](image)

(a) (b) (c) (d) (e)

Figure 4-8. A comparison among (a) the original image, (b) the saliency map generated by ASM component of my model, (c) the ground truth, (d) the salient region generated with p-ROI, and (e) the final result of my model.

The ASM responses to all features it detects wherever they locate in the input image. The ASM indeed detects the salient region but not the salient object. By combining my ASM with p-ROI, I would be able to detect salient objects more accurately.

### 4.3.4 Combination of Adaptive Saliency Model with p-ROI

To reduce noise, p-ROI approximation is used to find a region which most
likely overlays with the salient object. Then the map generated from ASM and the one from p-ROI is combined to form the final saliency region that contains the salient object.

As shown in Table 4-2, my model is one of the best models among those tested. It is obvious that neither the ASM component nor the p-ROI component of my model can perform better than the combined model. The main reason is that the two components are complement with each other and therefore the combined model performs better than each individual component.

![Image of comparison](image)

Figure 4-9. A comparison among (a) the original image, (b) the saliency map generated by the ASM, (c) the ground truth, (d) the salient region generated by p-ROI, and (e) the final salient object detected from my combined model.

The ASM is a pixel-based model. It pays too much attention to the local contrast and overlooks the global characteristics of the image, leading to an existence of a
large amount of noise points. As a region-based model, this problem does not affect p-ROI. Therefore p-ROI can help the ASM by eliminating noise and consequently improves the performance. Figure 4-8 shows an example in which p-ROI approximation greatly improves the overall performance. On the contrary, p-ROI approximation is based on a fundamental concept that all salient regions should be concentrated and continuous. Yet, in natural scenes, it is not always true. If two or more separated salient objects dominate the image with a relatively large distance, the performance of p-ROI would drop drastically (assuming there is one object only). In this case, ASM provides the details on the local regions to improve the overall result. Figure 4-9 shows an example in which ASM can drastically improve the performance.

In a few circumstances, there is possibility that neither components of my model work fine. For example, the background is so noisy so that ASM works unsatisfactory and there are two or more objects in the image so that p-ROI approximation also works disappointingly. In these cases, the combined model also fails to produce satisfactory salient object detection results.

A comparison of precision-recall curves among my model and nine other models is shown in Figure 4-10. My model is comparable to some good models. From the figure, I can see that the combined model is better than each individual component. Figure 4-11 shows the saliency maps generated from my model and several other models on several images from the MSRA dataset.
Table 4-2. Results of different saliency models on the test set

<table>
<thead>
<tr>
<th>Model</th>
<th>AUC</th>
<th>Model</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASM</td>
<td>0.5935</td>
<td>p-ROI</td>
<td>0.5629</td>
</tr>
<tr>
<td><em>Our model</em></td>
<td>0.7984</td>
<td>AC</td>
<td>0.6149</td>
</tr>
<tr>
<td>AIM</td>
<td>0.6133</td>
<td>CA</td>
<td>0.6580</td>
</tr>
<tr>
<td>IM</td>
<td>0.5303</td>
<td>IT</td>
<td>0.6173</td>
</tr>
<tr>
<td>SeR</td>
<td>0.5980</td>
<td>SR</td>
<td>0.5058</td>
</tr>
<tr>
<td>SUN</td>
<td>0.5007</td>
<td>SWD</td>
<td>0.7503</td>
</tr>
</tbody>
</table>

Figure 4-10. A performance comparison between my model and other methods tested.
Figure 4-11. Saliency maps generated from my model and several other models. (a) the original image, (b) the ground truth, (c) ASM, (d) my hybrid model, (e) P-ROI, (f) AC, (g) AIM, (h) CA, (i) IM, (j) IT, (k) SeR, (l) SR, (m) SUN and (n) SWD.
4.4 Summary

Pixel-based and region-based approaches are two main strategies to detect saliency objects. In this chapter, I first demonstrate that ASM, the originally designed for saliency prediction, cannot achieve good results in some circumstances. I propose a combined ASM and region-based approach for salient object detection. In particular, a scene image is processed by ASM and the region-based approach in parallel. The matched output of that generated by ASM and that generated by the region-based method is marked as the salient object region. I show that the combined ASM and the region-based approach improves the salient object detection performance compared to two methods used individually. My model also achieves competitive results compared with some good-performed methods tested.
Chapter 5 Deep Learning of Heat Maps of Human Eye Gaze Data

5.1 Background

Because human visual attention system works like a “spotlight” to the regions which draw their attention, precisely acquiring valid ground truth has become crucial in computational models for saliency prediction. Eye-tracking technology has been rapidly developing in recent years. Having obtained sufficient human eye gaze date, simulating human eye gaze becomes possible. The most common means to represent human eye gaze is to visualize them in the format of a heat map. Therefore, simulating (learning) heat maps of human eye gaze data becomes an important and challenging task in the field of computer vision.

In Chapter 3, I proposed a deep neural network, ASM, for saliency prediction. The deep neural network shows effectiveness in generating 2-D saliency maps. Similarly, heat maps are 2-D maps to represent regions that attract human beings’ attention. Therefore, ASM is modified and trained to simulate the heat maps of human eye gaze data, which is reported in this chapter. Experimental results show that ASM is not effective enough to accomplish this task because human eye gaze is less related to what is contained in the images. Besides, the “spotlight” effect of the human visual attention system makes a heat map different from a computational saliency map. To overcome this problem, additional process called distribution
processing is introduced in this chapter. By adopting ASM and distribution processing, I proposed a deep learning model to learn the 2-D heat maps of human eye gaze data.
5.2 Deep Learning for Learning Heat Maps of Human Eye-Gaze

Learning heat maps of human eye gaze is a much more difficult task comparing to saliency prediction. Therefore, utilizing only ASM can hardly accomplish the task. The human attention system is basically a combined bottom-up and top-down process. However, ASM is designed as a bottom-up model. Hence, more investigation needs to be made for the generation of heat maps of human eye gaze.

Figure 5-1. Block diagram of ASM based model for learning heat maps of human eye gaze.

As shown in Figure 5-1, three main steps are necessary in my proposed approach. The first step is a conversion of the fixation data to eye gaze heat maps. This step not only obtains the target output for training ASM, but also generates the
ground truth of test samples for performance evaluation. The second step is to train ASM on the heat map data in order to generate a useful intermediate output for further processing. The structure of ASM used in this step is the same as the one proposed in Chapter 3 except for different post-processing applied. In Chapter 3, a normalization step with a threshold is applied to reduce noise and enhance saliency values. While in this chapter, the normalization step is replaced by a distribution processing step. The output generated from ASM is regarded as the intermediate result. In the distribution processing step, the intermediate result is converted to the final heat map.

5.2.1 Preparation of Heat Maps

The Object and Semantic Images and Eye-tracking (OSIE) data set contains eye-tracking data from 15 observers. The data provided in the OSIE dataset contains fixations instead of eye gaze heat maps. Therefore it is necessary to convert the fixation data to eye gaze heat maps. According to the description of fixations in the OSIE dataset (Xu et al., 2014), the fixation should be processed with the Gaussian distribution as follows:

$$H = \text{Norm} \left( \sum_i T_i \times N \left( \mu_i + \mu_{\text{offset}}, \Sigma_0 \right) \right)$$

(5.1)

where \(H\) stands for the heat map value, \(T_i\) is the time duration of each gaze point, \(N(\cdot)\) denotes the Gaussian distribution (Normal distribution), \(\mu_i + \mu_{\text{offset}}\) represents the center location of Gaussian distribution, and \(\Sigma_0\) is a covariance.
matrix with diagonal elements only:

\[ \Sigma_0 = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} \]  

(5.2)

In the process, each fixation point of each observer is regarded as the center \( \mu_i \) of the 2-D Gaussian distribution. There is also an offset which is obtained from statistics of the dataset with \( \mu_{\text{offset}} = (-0.02, 0.05) \). Similarly, squared scales are acquired from statistics with \( \sigma_1 = 1.86 \) and \( \sigma_2 = 1.9 \).

After obtaining the 2-D Gaussian distributions of all fixation points, they are converted into one map by performing a weighted summation, where the weight for each point is the time duration of the corresponding eye gaze point. With a normalization process, the final heat map is obtained:

\[ \text{Norm}(x) = \frac{x}{\max(x)} \]  

(5.3)

Figure 5-2. An example of constructed heat map of human eye gaze and its original image.

Heat maps are visualized in gray-level format. The regions in black having a zero value catch no human attention during the human observation. On the contrary,
those regions in white with a non-zero value are the regions at which human eyes
stare. An example of grey-level heat map is shown in Figure 5-2 together with the
original image.

5.2.2 Outline of Deep Structure

The deep structure adopted for this task is based on the one proposed in Section
3.2. The parameter setting of the deep structure is shown in Table 5-1.

Table 5-1. Parameter setting in different layers of the ASM for learning heat maps.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Input Layer</th>
<th>C2</th>
<th>S3</th>
<th>C4</th>
<th>S5</th>
<th>Output Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>396x300</td>
<td>392x296</td>
<td>196x148</td>
<td>192x144</td>
<td>64x48</td>
<td>64x48</td>
</tr>
<tr>
<td># Maps</td>
<td>24</td>
<td>15</td>
<td>15</td>
<td>13</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td># Kernels</td>
<td>360</td>
<td>195</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td># Weights</td>
<td>9,000</td>
<td>4,875</td>
<td>13</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Kernel size</td>
<td>-</td>
<td>5 x 5</td>
<td>-</td>
<td>5 x 5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Scale factor</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>3</td>
<td>-</td>
</tr>
</tbody>
</table>

The original image is normalized and down-sampled into a lower resolution
with three different scales $s$ while the data size remains unchanged as discussed in
Section 3.2.1.1.

$$x_s = down(x_{nor}, s), s \in \{2, 4, 8, 16, 32, 64, 128\}$$  \hspace{2cm} (5.4)

Down-sampled data in different scales are stocked channel by channel to form the
input data. Input layer I1 has 24 (3x8) channels from R, G and B components of 8
scaled input images. All the channels have the same size of 396 x 300 in pixels.

Both convolutional layers (C2 and C4) have the data convolved by several 5 x
5-sized kernels. Each kernel has 25 trainable parameters. As a result, 9,000 trainable
parameters are for the C2 layer and 4,875 trainable parameters for C4 layer. The size
of each feature map in layer C2 is reduced to 392 x 296 and that in layer C4 becomes
192x144. The subsampling layer used is an average pooling layer. S3 reduces the
data dimension with a scale factor of \( f = 2 \). As a consequence, size of each feature
map becomes 196 × 148 in S3. S5 subsamples data maps with a scale factor of \( f = 3 \).
Size of each feature map in S5 is reduced to 64 × 48 after subsampling. The weighted
summation of all feature maps in S5 is taken as the input of the output layer with a
size of 64 x 48.

5.2.3 Distribution Processing

Due to the nature of ASM, the map generated from ASM is still more like a
caliency map instead of a heat map even though the ASM is trained on heat map data.
To address this issue, distribution processing is employed. Inspired by the procedure
to convert the fixation data to a heat map of human eye gaze, distribution processing
is designed to use suitable pixels in the image as the center of 2-D Gaussian
distributions. All available 2-D Gaussian distributions will be summed up into a
single heat map. This process is defined as follows:
\[ M_i = \begin{cases} \text{Norm}(O_i), & \text{if } \text{Norm}(O_i) > 0.5 \\ 0, & \text{otherwise} \end{cases} \]  \hspace{1cm} (5.5)

where \( O_i \) is the intermediate output from the ASM, \( M_i \) stands for the magnitude of the normalized intermediate output, and \( \text{Norm}(\bullet) \) is the same as defined in Equation (5.3).

\[ H = \text{Norm}\left( \sum_i M_i \times N(\mu_i, \Sigma) \right) \]  \hspace{1cm} (5.6)

where \( N(\bullet) \) denotes a Gaussian distribution, \( \mu_i \) represents the selected coordinates (the center location of the Gaussian distribution), and \( \Sigma \) is the covariance matrix as defined as

\[ \Sigma = \begin{pmatrix} 3.5340 & 0 \\ 0 & 3.5340 \end{pmatrix} \]  \hspace{1cm} (5.7)

The covariance matrix \( \Sigma_0 \) introduced in Section 5.2.1 is directional biased since the diagonal elements are of different values. The directional bias is caused by many reasons such as experimental setup, equipment setup and object orientations. Assuming that the human eye-gaze heat map is normally not directional biased, \( \Sigma \) is obtained with identical diagonal elements without changing its determinant.
5.3 Experimental Results and Discussion

5.3.1 Datasets and Experiment Setup

The experiments on my model were conducted on the Object and Semantic Images and Eye-tracking (OSIE) data set (Xu et al., 2014). The input of my model is the original image from the OSIE data set. The target output (ground truth) is the generated heat map of the human eye gaze based on the fixation data provided in the OSIE data set. The fixation data was collected using a 2,000 Hz eye tracker with 15 observers aged between 18 and 30. Each observer was given 3 seconds to perform free viewing on each image. All fixations from 15 observers in each image were merged into one fixation map to make the data more generalized. The objective of the experiments is to train a CNN model to produce the eye fixation heatmap of an input image. The data set were divided into two subsets: one was used as the training set and the other as the test set. Specifically, 80 images were used as training samples and another 120 images were used as test samples. The experiments were conducted across datasets as well. The CAT2000 (Borji and Itti, 2015) dataset was also adopted for my experiments. In particular, 150 images from this dataset were used as test samples as well. The CAT2000 dataset is a dataset which consists of 20 different categories of images sized at 1920x1080 in pixels. With a 1,000 Hz eye tracker, 18 observers aged between 18 and 27 performed free viewing for 5 seconds on each image. All fixations from all observers in each image were merged into one fixation
map to make the data more generalized. This dataset was adopted to perform the cross-dataset evaluation to test the generality of our model.

5.3.2 Metrics for Performance Evaluation

Two metrics are used to evaluate the performance of my model, namely Receiver Operating Characteristic (ROC) and Area Under Curve (AUC). ROC is usually presented with figures whose X axis denotes False Positive Rate (FPR) and Y axis stands for True Positive Rate (TPR):

\[
TPR = \frac{TP}{TP + FN} \\
FPR = \frac{FP}{FP + TN}
\]  \hspace{1cm} (5.8)

where true positive (TP) means a hit, true negative (TN) represents correct rejection, false positive (FP) is a false alarm, and false negative (FN) stands for a miss.

5.3.3 Experimental Results without Distribution Processing

As mentioned in Section 5.2, there should be three steps to obtain the final heat map. In this experiment, however, only the first two steps are involved in order to compare the outputs with and without distribution processing. The ASM was trained on 80 training images and heat maps for 100 iterations with the Rprop algorithm, and then tested on 120 test samples. The model was initialized with a set of random weights. The training error is shown in Figure 5-3(a).
Our model achieves AUCs of 0.6783 and 0.6260 for the training set and the test set, respectively. The ROCs for training samples and test samples are shown in Figure 5-3(b). As shown in Figure 5-4, visually, the intermediate output from ASM is quite different from the ground truth. The intermediate output looks more like a saliency map instead of a heat map because of the nature of ASM which was designed to generate saliency maps at the first place.

Figure 5-4. Example of the original image, the ground truth and the output of ASM.

Considering that the ASM is still generating saliency-map-like output, another experiment was conducted to examine whether an ASM with pre-trained weights in
Chapter 3 could improve the performance. The ASM was trained on 80 training images and heat maps for 100 iterations with the Rprop algorithm, and then tested on 120 test samples. The model was initialized with the weights from pre-training on Itti model with structure parameters shown in Table 5-1. The training error is shown in Figure 5-5(a).

Figure 5-5. Training error (a) and ROCs of the training set (blue) and the test set (red) (b) with random initial weights.

The overall performance of the model in AUCs is 0.6732 for training and 0.6299 for testing. The ROCs for training and test samples are shown in Figure

Original Image  Ground Truth  Intermediate Output
A comparison among original image, ground truth and output is shown in Figure 5-6.

From the above result, it is easy to see that the performances of two initializations are competitive and the pre-trained weights help little in improving the performance. It is difficult to generate something visually similar to the heat map with ASM only.

### 5.3.4 Experimental Results with Distribution Processing

In this experiment, all three steps were carried out. The ASMs used in this experiment are those trained in Section 5.3.3. For the model with the random initialization, the ROC is shown in Figure 5-7. The overall performance of the model in AUCs is 0.7257 for training samples and 0.6764 for test samples. As shown in Figure 5-8, visually, the final output is quite similar to the ground truth. It is obvious that the final output is more like a heat map compared to the intermediate output from ASM.

Figure 5-7. ROCs of the training set (blue) and the test set (red).
Figure 5-8. Comparison on the original image, the ground truth and the output after distribution processing for the model with random initial weights.
As for the model with pre-trained weights, the performance improves slightly.

The ROCs are shown in Figure 5-9.

![ROC Graph](image)

Figure 5-9. ROCs of the training set (blue) and the test set (red) with pre-trained weights.

The overall performance of the model in AUCs is 0.7180 for the training set and 0.6764 for the test set. As shown in Figure 5-10, visually, the final output is quite similar to the ground truth.

As a conclusion on these experiments, utilizing only ASM can barely learn the heat maps of human eye-gaze. When the ASM was used only, a proper weight initialization did slightly improve the performance. The better initialization it has, the better performance it obtains. However, with the distribution processing, the weight initialization becomes less important. The distribution processing does help to improve the performance of my model both statistically or visually.
Figure 5-10. Comparison on the original image, the ground truth and the output after distribution processing for the model with pre-trained weights.
5.3.5 Experimental Results on CAT2000 Dataset

This experiment was conducted on the CAT2000 dataset to test the generalization ability of my model. The model was firstly trained on the OSIE dataset with initially random weights as mentioned in Section 5.3.4. The CAT2000 dataset was only used as test samples to perform a cross dataset testing.

The overall performance of the model in AUC is 0.7257 for the OSIE training samples while the overall performance is 0.6684 for the CAT2000 test samples. Comparing to the results in Section 5.3.4, the results (0.6684 vs. 0.6764) are rather competitive under a cross dataset evaluation. Examples of the results produced by my model for the CAT2000 data set are shown in Figure 5-11.
Figure 5-11. Examples of the original image, the ground truth and the output after distribution processing from the CAT2000 dataset.
5.4 Summary

In this chapter, I propose a model for learning the heat maps of human eye gaze data. The model consists of a modified ASM and a distribution processing step. The ASM exploits regional visual features and predicts the saliency of each region. The distribution processing is capable of simulating the diffusion of human gaze. Experiments are conducted on the Object and Semantic Images and Eye-tracking (OSIE) data set. The fixation data provided is converted to heat maps which are used as the ground truth to train my model. I first show that deep learning (ASM) is possible to simulate the heat maps of human eye gaze in a bottom-up approach. The defects of this deep learning without distribution processing are also revealed during the experiments. Experimental results on CAT2000 and OSIE datasets show some positive results although there is much room for improvement.
Chapter 6 Conclusions and Future Work

6.1 Conclusions of the Thesis

In the thesis, I investigated deep neural networks for saliency prediction and salient object detection. By using a saliency model, I further investigated the learning of human eye-gaze heat maps. The thesis can be concluded as follows.

In Chapter 3, an adaptive saliency model (ASM) based on convolutional neural network is proposed to generate saliency maps. The proposed model consists of two convolutional layers and two subsampling layers. ASM is capable for both feature extraction and classification or function approximation. The special characteristic of my proposed ASM is that it generates the 2-dimentional saliency map output. Such an output makes the deep neural network difficult to be trained. The Rprop training algorithm was adopted to overcome the long-term dependency problem encountered in training my deep neural network with a large number of output nodes. The parameters of ASM were determined by trial and error. Experimental results on the Object and Semantic Images and Eye-tracking (OSIE) dataset demonstrate the effectiveness of my proposed ASM compared with state-of-the-art algorithms for saliency prediction based on the Histogram Intersection (HI) metric.

I then extended my proposed ASM to perform salient object detection by combining it with a region-based saliency method in Chapter 4. In the model, both ASM and p-ROI approximation components have contributed to the performance
improvement. The ASM component improves the performance when the details of local regions are needed. On the other hand, p-ROI component ameliorates the result when the details become a burden. Experimental results show that my hybrid model works better than individual components and is competitive comparing with other good-performed methods.

I also extended my ASM to learn human eye-gaze heat maps. The fixation data provided was converted to heat maps before being used as the ground truth to train the model. The ASM was trained directly on heat maps. Experimental results show that utilizing ASM only is difficult to learn heat maps. The performance can be improved by using a distribution processing step.

6.2 Future Research Directions

More work can be pursued along the line of my research, which is discussed below.

(1) Due to a limitation in the computer power, in the current investigations, I can afford to train on the relatively small data sets. In the future, I will explore the use of GPUs in the training process and significantly increase the data size.

(2) The ASM proposed in Chapter 3 achieves a promising performance for producing the two-dimensional output (function approximation) by overcoming several problems encountered during the training process. In
my experiments, the ASM is trained for producing the continuous-valued outputs of Itti’s saliency model. Continuous values can be treated as confidence levels of a certain class. In other words, training the ASM on the target outputs representing different keywords (classes) could make the ASM useful in pixel-wised image annotation.

(3) The third direction is to use the ASM to perform face detection, i.e., to label image pixels that are part of human faces. I would like to modify the deep structure in Chapter 4 to carry out this task. In particular, the target map could be binarized to indicate where it contains faces or not. As my model was originally designed to detect object salience, it can be extended to detect human faces after proper training.
References


for Strabismus Diagnosis,” *2015 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, October 9-12, Hong Kong, Hong Kong, pp. 2305-2309.


implicit relevance feedback in a content based image retrieval,” *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, March 22-24, Austin, Texas, USA, pp. 73-76.


symmetry,” the 31st Annual Conference of the Cognitive Science Society, July 29 - August 1, Amsterdam, Netherlands, pp. 56-61.


*ICME’02*, August 26-29, Lausanne, Switzerland, pp. 741-744.


