Image Saliency Prediction using Deep Learning Models

A Project as a Course requirement for

Master of Technology in Computer Science

Goutam Kelam

17555

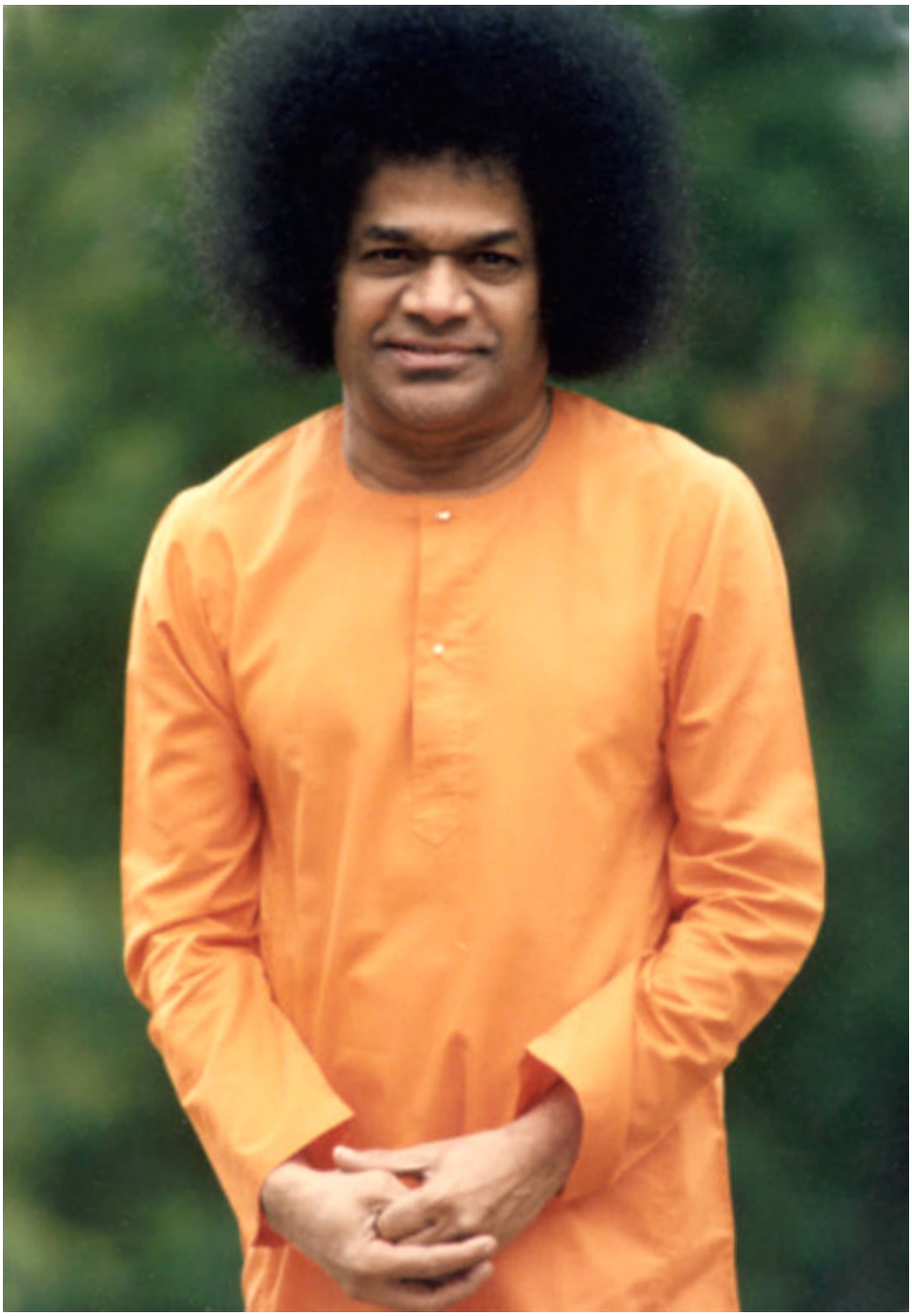
SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING

(Deemed to be University)

Department of Mathematics and Computer Science

Prasanthi Nilayam Campus

March 2019



A pen in my hand and my hand in His hands

"Dedicated to Sai Maa"

and

"My Loving Parents"

DECLARATION

The Project titled **Image Saliency Prediction using Deep Learning Models** was carried out by me under the supervision of Dr. S. Balasubramanian, Department of Mathematics and Computer Science, Prasanthi Nilayam as a Course requirement for the Degree of Master of Technology in Computer Science and has not formed the basis for the award of any degree, diploma or any other such title by this or any other University.

……..

Place : Goutam Kelam

17555

Date : M.Tech Computer Science

Prasanthi Nilayam

SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING
(Deemed to be University)

Dept. of **Mathematics & Computer Science**
Prasanthi Nilayam Campus

CERTIFICATE

This is to certify that this Project titled **Image Saliency Prediction using Deep Learning Models** submitted by Goutam Kelam, 17555, Department of Mathematics and Computer Science, Prasanthi Nilayam Campus is a bonafide record of the original work done under my supervision as a Course requirement for the Degree of Master of Technology in Computer Science.

……..
Dr. S. Balasubramanian
Project Supervisor

Countersigned by

Place: ……...
Dr. Pallav Kumar Baruah
Date: Head of the Department

Vidyagiri, Prasanthi Nilayam - 515 134, Andhra Pradesh, India
Tel: +91 8555 287235 | hoddmacs@sssihl.edu.in | www.sssihl.edu.in

Acknowledgement

"Gratitude is a miracle of its own recognition.
It brings out a sense of appreciation and
sincerity of a being." – Auliq-Ice

Dearest Swami, I express my deep sense of gratitude to you for being with me always. This project is just another reminder to me saying with you, I am capable of climbing the steepest mountain and without you unfit to even walk on the plains. Thank you, Swami, for all the love you have showered on me.

This project helped me to understand my mother's advise " See good even in the toughest situation", I did not know back then that it is called saliency. I understand it now after going through the ups and downs, twists and turns which the project threw at me. I thank my parents Mr. K. Appa Rao, Mrs. K. Rajkumari and my sister K. Renu for being my constant pillar of support in both happy and difficult times. My sincere thanks to Nalini Chinta for being my strongest ally, absorbing my rants and encouraging me.

I would like to thank my guide, Dr. S. Balasubramanian, for introducing me to the fascinating world of deep learning. His guidance has led me this far in my endeavor and this thesis is the byproduct of his trust which he has shown in me. Thank you for all the brainstorming sessions which we had, it gave me a different perspective of handling a problem.

I am grateful to my DMACS family who made me feel that I am always at home. I take a moment to thank Dr. Pallav Kumar Baruah, Head of the Department, DMACS, for always providing us with the facilities we desired from him. I thank our class teacher Dr. N. Uday Kiran for giving his timely feedback. I thank Sri. C.Uday Kiran and Sri Varun Vallampatla for taking their valuable time to set up the lab.

I would like to thank my batchmates Saichand, Varshaneya, Nikhil, Karam, Akhilesh, Saurav, Gunashekhar, Sharavan, Naren, Sathya, Pranav, Hemanth, Sairam, Veera, Ramkrishna, Shyam and Abhishek for helping me to overcome

all the tough situations, for being a part of my life, and shaping the way I look at life.

I thank all my "Maintenance Family" and "2+2" group for their constant support and encouragement.

I thank Sri Karthik Aiyer, our room teacher, and a caring brother, for always being available in time of need.

I would like to thank especially all the hostel workers and Sri Venkatachalam sir for preparing delicious food and striving hard to make us feel comfortable in the hostel.

Last but not the least thanks to the Google search engine which was my companion throughout. I thank all those who directly and indirectly helped in the completion of this work.

This thesis has been written using LaTeX.

Goutam Kelam
Regd. No. 17555

Abstract

Salient object detection is one of the important properties of the Human Visual System. In many cases, the focus of interest of the human eye gets attracted to a region or objects that appear distinctly in the foreground of a scene. The ability to automatically detect the objects of interest in an image is known as " saliency prediction".

Saliency prediction is used in many computer vision tasks such as automatic cropping and thumbnailing, object recognition, content-aware resizing, image-retargeting, object tracking in videos and so on. It acts as a pre-processing step and helps in reducing the search space for many other (intermediate or high-level) processes to follow, such as feature-extraction, matching, enhancement, compression, etc., hence reducing the computational time.

In this work, we used deep learning models to predict the saliency maps. We implemented two different approaches to predict saliency, namely, a convolutional approach and a generative-discriminative approach. We further proposed an improvisation of a generative-discriminative approach.

The performance of our proposed improvisation is shown using visual illustrations, while the F-measure and mean absolute error metrics are used to quantitatively compare the same with the state-of-the-art techniques using challenging benchmark real-world datasets such as MSRA10K, ECSSD, and DUT-OMRON. Our proposed improvisation performs better than 12 of the state-of-the-art models.

Contents

1 Introduction **1**

1.1 Motivation 2

1.2 Objective 4

1.3 Problem Definition 4

1.4 Work Done and Our Contributions 4

1.5 Chapterization 6

2 Literature Review **7**

2.1 Introduction 7

2.2 Evolution of the Saliency Models 8

2.2.1 The 1st wave: 8

2.2.2 The 2nd wave: 9

2.2.3 The 3rd wave: 10

2.3 Deep Learning based Saliency Prediction 10

2.3.1 The CCN based approaches 10

2.3.2 The FCN based approaches 12

2.4 Advantages of FCN based approaches 13

3 ConvNets for Saliency Prediction **15**

3.1 Objective of the chapter 15

3.2 What is a saliency map? 15

3.3 Why deep learning? 16

3.4 Shallow ConvNet 16

3.4.1 Architecture 17

3.4.2 Training 18
3.4.3 Results 18
3.5 Deep ConvNet 20
3.5.1 Architecture 20
3.5.2 Training 21
3.5.3 Results 22

4 GANs for Saliency Prediction 25
4.1 Introduction 25
4.2 The SalGAN architecture 26
4.2.1 Generator 27
4.2.2 Discriminator 28
4.3 Training 30
4.3.1 Content Loss 30
4.3.2 Adversarial Loss 30
4.4 Results 32

5 Improvisation of GANs for Saliency Prediction 37
5.1 Introduction 37
5.2 Overview of the proposed method 38
5.3 Architecture 39
5.3.1 Generator 39
5.3.2 Discriminator 41
5.4 Training 42
5.4.1 Content Loss 43
5.4.2 Adversarial Loss 44
5.5 Results 44

6 Conclusion & Future Work 51
6.1 Conclusion 51
6.2 Future Work 52

A Saliency Prediction Metrics **55**
A.1 Introduction 55
A.2 Saliency Prediction Metrics 56
A.2.1 Location-based saliency metrics 57
A.2.2 Distribution-based saliency metrics 59
A.3 Summary 61

List of Figures

1.1 **Images and their corresponding saliency maps** 3
1.2 **Applications of saliency detection** 5

3.1 **The ShallowNet Architecture** 17
3.2 **Results of our implementation of ShallowNet** 19
3.3 **The DeepNet Architecture** 21
3.4 **Results of our implementation of DeepNet** 22

4.1 **Overall architecture of the SalGAN** 27
4.2 **The encoder part of SalGAN** 28
4.3 **The decoder part of SalGAN** 29
4.4 **The discriminator architecture** 29
4.5 **Results of our implementation of the SalGAN model** . 32
4.6 **Qualitative assessment: Perceptual Loss Vs. Content Loss** 34

5.1 **The overall architecture of our model** 39
5.2 **Generator model** 40
5.3 **Loss Computation** 43
5.4 **Qualitative Result: MSRA10K dataset** 45
5.5 **Qualitative Result: ECSSD dataset** 46
5.6 **Qualitative Result: DUT-OMRON dataset** 47
5.7 **Qualitative comparison of our model with the SalGAN model on MSRA10K dataset.** 48

A.1 **Evaluation of saliency models using different metrics**[1] . 56

List of Tables

3.1 **Comparison of Our implementation of the models with the Author's implementation** 23

4.1 **Comparison of our implementation of SalGAN with the Author's implementation** 33

4.2 **Quantitative assessment: Perceptual Loss Vs. Content Loss** 35

5.1 **Architecture of shallow network (conv_int) which generate intermediate saliency maps** 41

5.2 **Architecture of shallow network (conv_final) which generate final saliency maps** 41

5.3 **The discriminator architecture** 42

5.4 **Comparison of our model with the state-of-the-art models for saliency prediction.** 49

5.5 **Quantitative comparison of our model with the SalGAN model.** 50

A.1 **Categorization of different saliency metrics** 62

Chapter 1

Introduction

"When you change the way you look at things, the things you look at change." – Max Planck

Every minute 49380 photos get uploaded on Instagram, about 4333560 videos get streamed on YouTube [1]. The amount of visual content is increasing at an overwhelming rate. The humongous surge of image data can be an outstanding source of information; however, it also raises the challenge of processing these data and extracting relevant information at an equally faster rate.

Computer vision is a field of computer science which deals with extracting intelligent information about the real world from the digital content such as images and videos. Computer vision tries to model the human visual system (HVS) and aims to develop autonomous systems which can perform a similar task as HVS can perform [2]. In order to imitate the HVS, a series of tasks are performed such as image acquisition, pre and post processing, analysis and understanding of images. Computer vision has many subfields such as image classification, object detection, saliency detection *(a.k.a visual attention)* to name a few.

Research in the computer vision field started adopting the usage of deep learning (DL) in order to solve the complex tasks. The credit for this goes

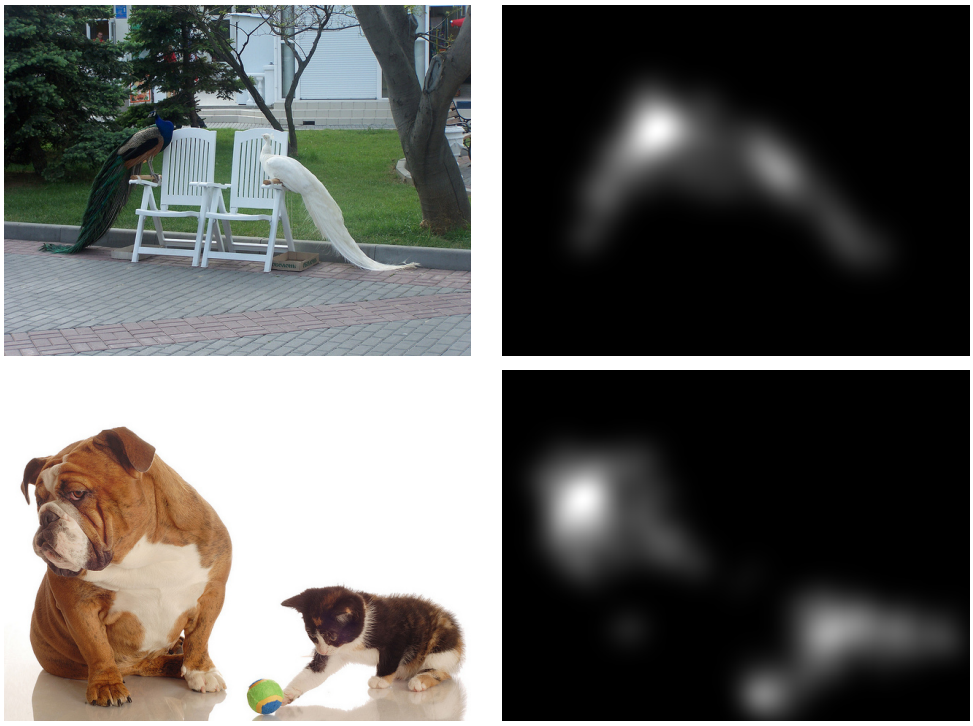
to the humongous amount of data available and the drastic rise of computational power. DL aims to empower computational models to learn the best representation of the data through multiple layers of abstraction. DL model has several hidden layers and each layer has several nodes known as *neurons*. These neurons are responsible for performing basic operations. The parameters involved in the model are constantly updated using a *backpropagation* algorithm. The updation of parameters is known as the *learning process*. DL has provided numerous state-of-the-art models in tasks such as object detection, image classification, etc. This project uses DL tools for saliency prediction.

The HVS receives about 10^8 to 10^9 bits of information every second. In order to process such huge data in real time, HVS uses its ability to selectively focus on different parts of the scene. Given an image, the nervous system selects part of the scene for further detailed processing, while discarding the rest. It also prioritizes the selected part such that the most relevant parts are processed first. This selection and ordering process is known as *selective attention* or *visual saliency* [3].

The visual attention models such as Shallow and Deep Convolutional Networks for Saliency Prediction [4], SalGAN: visual saliency prediction with adversarial networks [5] etc, aims to predict the salient regions of the image. The salient region detection can save computational resources as only the relevant information is processed. It can also be used as a preprocessing step for many other computer vision tasks such as object detection, object recognition, etc.

1.1 Motivation

The capability of the region to be visibly prominent amongst its surroundings is called the ***saliency*** of the region. There are two ways in which a region can be salient - stimulus driven and task specific. Stimulus driven is also known as ***bottom-up*** saliency, whereas the task driven is known as ***top-***



(a) Images (b) Saliency Maps

Figure 1.1: **Images and their corresponding saliency maps**

down saliency. In the bottom-up approach, the saliency is content dependent and is independent of the human observer. In the top-down approach, the observer has an idea about which region could be salient.

Saliency detection using a bottom-up approach is a challenging task, as there is no prior knowledge about which part of the image can be significant. In this approach the features extracted are purely driven by data present in the image. Thus the saliency prediction is entirely context dependent. In Figure 1.1, the first column shows the images taken from the SALICON dataset [6] and the second column shows their corresponding saliency maps.

Application of Saliency Prediction

The importance of saliency prediction lies on their applications in the various fields which they serve, such as computer vision, robotics, and graphics. One of the advantages of saliency map prediction is that it narrows down the region of interest. Thus, reducing the amount of computation and time needed. The saliency models are used in applications such as content-aware resizing [7], automatic cropping and automatic thumbnailing [8], content-based image retrieval [9], Human-Robot interactions [10] etc. Figure 1.2 shows some of the applications of saliency prediction.

1.2 Objective

The objective of this thesis is to develop an accurate and computationally efficient saliency detection model that can aid as a pre-processing step for many computer vision tasks such as object detection, semantic segmentation, etc. The model developed must be a supervised model, such that the learning process is guided in order to extract meaningful features.

1.3 Problem Definition

Given a natural image, we need to develop a novel DL model that detect the region of interest (ROI) in the image which contains the salient objects.

1.4 Work Done and Our Contributions

We have attempted saliency prediction using two DL models viz. Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GAN). Further we have proposed an improvised GAN model that emphasizes on both high level and low level features to produce sharp and smooth saliency maps.

(a) Automatic Cropping

(b) Automatic Thumbnailing

(c) Content-based Image retrieval

(d) Content-aware resizing

Figure 1.2: **Applications of saliency detection**

In summary,

1. Implemented the Shallow Convnet and the Deep Convnet architectures for saliency prediction. [4]
2. Implemented the SALGAN architecture for saliency prediction. [5]
3. Proposed and implemented an improvised GAN model for saliency prediction.

1.5 Chapterization

The thesis is organized as:

Chapter 2: This chapter presents a limited survey of literature in the field of saliency prediction.

Chapter 3: This chapter elucidates on the first end-to-end convolutional architecture designed for saliency prediction. This architecture designed by Junting Pan et al. [4] won the LSUN challenge 2015 for saliency prediction.

Chapter 4: This chapter describes in detail the first adversarial training based architecture designed for saliency prediction. This architecture is named SalGAN and is designed by Junting Pan et al. [5]

Chapter 5: The proposed improvised GAN model to predict the saliency map is elicited in this chapter.

Chapter 6: This chapter presents the concluding remarks and enumerates the directions for future work.

Appendix A: The appendix gives an overview of the different kinds of saliency metrics used to evaluate the saliency prediction models.

Chapter 2

Literature Review

"The more extensive a man's knowledge of what has been done,
the greater will be his power of knowing what to do."
– Benjamin Disraeli

2.1 Introduction

Information overload is one of the greatest challenges of visual perception. The HVS receives a massive amount of information (approximately 10^8 to 10^9 bits) every second. The HVS process such a massive amount of data in real-time by selectively focusing its attention on various parts of the scene. The HVS detect the visually distinct objects in the scene quickly. This ability of HVS is known as the "***pre-attentive stage***". The selected parts are further processed to extract meaningful information. This stage is known as the "***attentive stage***".

The term "*saliency*" is coined for the first time for visual attention by Tsotsos et al. [11]. Since then the research in the field of saliency prediction is evergreen. The history of research in saliency prediction is split into three phases:

- The first phase started when Itti et al. [12] developed one of the first saliency prediction models. It created an interest in many areas of re-

search such as neuroscience, cognitive psychology, and computer vision. Itti et al. demonstrated that his model was able to spatially detect the discontinuities present in a scene.

- The second phase started when Achanta et al. [13] and Liu et al. [14] defined saliency prediction as a binary classification problem. This new definition inspired the discovery of many saliency models.

- The third phase started very recently, with the return of CNNs. The classical approaches depend largely on the contrast difference between the background and the salient object. In contrast to the classical approaches, the CNN based approaches [15], [4] discarded the use of hand-crafted features for saliency detection. Deep Learning (DL) based models for saliency prediction outperform their traditional counterparts by a huge margin. In MIT300 benchmark for saliency prediction, out of the top 10 models 9 are DL models.

2.2 Evolution of the Saliency Models

The saliency models have evolved significantly over the last 20 years. In this section, we give a brief chronological overview of the evolution of the saliency models.

2.2.1 The 1st wave:

- **1998**: Itti et al. [12] developed one of the first computational models which combined features of the image at multiple scales and formed a topographical saliency map. The salient regions are further processed using a dynamic neural network in decreasing order of saliency.

2.2.2 The 2nd wave:

- **2007**: Liu et al. [13] defined the saliency detection problem as binary classification problem. The focus of their work was to localize the salient objects using bounding boxes.

- **2009**: Achanta et al. [14] used frequency based approach to compute the saliency maps. Every pixel x of an Image I is given a saliency score as:

$$S_{score}(x) = ||I_{mean} - \tilde{I}(x)||^2$$

  where S_{score} denotes the saliency score, I_{mean} denotes the mean pixel value of the image and $\tilde{I}$ denotes the input image after Gaussian blurring.

- **2010**: Goferman et al. [16] developed one of the first content-aware saliency prediction models. They defined their global contrast metric as the uniqueness of the patch in an image (i.e if a patch is characteristically different from its neighboring patches then it is salient). Goferman et al. argue that the salient object's context is as important as the salient object itself.

- **2011**: Chang et al. [17] detects saliency using global contrast distinction based models. They used two models viz. Histogram-based Contrast (HC) and Region based Contrast(RC). The HC model uses the statistics about the color in the image to assign a saliency score to each pixel in an image. The RC model defines the saliency of each region based on its contrast with its neighboring regions. The HC model generates finer saliency maps and is computationally efficient whereas the RC model generates saliency maps of high quality at the cost of computation.

- **2013**: Yang et al. [18] detect saliency using the different types of priors namely, contrast, center, and smoothness. In this method, an initial saliency map is computed by taking the contrast and center

priors into consideration. In the traditional approach, the image center acts as a center prior, but Yang et al. decided the salient object's center by using the convex hull approach. This approach asserts the robustness of the saliency map with respect to the location of an object. The initial saliency map is further refined using graph regularization strategy which assigns same saliency score to adjacent salient pixels.

2.2.3 The 3rd wave:

The above-mentioned methods depend hugely on heuristics i.e. hand-crafted methods. Even though these methods detect saliency in real-time, they suffer from the inability to generalize well. CNNs tend to perform better than the classical approaches due to their ability to extract low-level and high-level features. The high-level feature helps in the robustness of the saliency map with respect to localization of the object and the low-level feature helps in refining the saliency map.

2.3 Deep Learning based Saliency Prediction

The saliency prediction using the DL approach can be broadly categorized as Classical Convolutional Networks (CCN) based approach and Fully Convolutional Network (FCN) based approach.

2.3.1 The CCN based approaches

In the CCN based approaches, the input image is initially segmented into smaller regions. The smaller regions are then passed as input to the CNN models which extract the high-level semantic features of these regions. The high-level semantic features are further passed to multi-layer perceptron (MLP) (i.e. fully connected networks) models which assign the saliency score for the segmented regions.

In this section, we will discuss some of the CCN based approaches.

1D Convolutional approach

He et al. [19] developed one of the earliest CCN models for predicting the saliency maps. They proposed a computationally efficient approach by defining regions in an image and learning to represent these regions using a superpixel. He et al. grouped regions based on the colors and used the mean of the color as a representation for their superpixel. The model predicted cluttered saliency maps, the reason being the inability to preserve the spatial structure of the image due to the usage of 1D convolution and pooling.

Local and Global Context approach

Aiming to predict better saliency maps Wang et al. [20] designed two deep networks namely DNN-L and DNN-G for estimating the local and the global context respectively. The DNN-L model learns the pixelwise saliency score and the DNN-G model is used to predict the saliency of the regions in an image using the local saliency map. The final saliency map is a weighted sum of the salient regions.

Lee et al. [21] use encoded low-level features to enhance the saliency maps predicted using the high-level features. In this model, VGG network [22] is used as the base model to extract the low-level and high-level features. The low-level features become the input to a series of 1×1 convolutional layers and ReLU [23] layers. The encoded low-level features and the high-level features are further concatenated and passed through two MLP layers resulting in a saliency map for the query image.

Bounding Box approach

Kim et al. [24] proposed to detect saliency using a combination of deep learning and hierarchical segmentation technique. In their approach Kim et al. extract the patches from the input image and pass the patches as input to their CNN model. The CNN model outputs the fine and coarse representations of the input patches. The fine and the coarse representation

are then concatenated and passed through a fully connected network which predicts the saliency map. The saliency map predicted is further refined using a hierarchical segmentation technique.

Li et al. [25] predict the saliency map by using the low-level as well as high-level features. The low-level features are extracted using the hand-crafted methods; whereas, the high-level features are extracted using a CNN. Li et al. slightly modified the R-CNN [26] model by inserting local response normalization (LRN) layers in the first two layers. The input to the modified CNN is a set of image patches. The features extracted are further passed to an SVM to compute the saliency map.

All the above mentioned CCN based approaches segment the input image into regions and then pass these patches to the CNNs to determine the saliency maps of the region. Even though the CNNs extract the high-level features, in reality, they do not capture the entire high-level semantics because they are trained using patches of the image and not the whole image itself. The MLPs (fully connected networks) fail to preserve the spatial information of the image leading to information loss at a global scale.

2.3.2 The FCN based approaches

Long et al. [27] introduced the FCN architecture for saliency prediction for the first time. The FCN model predicts the saliency map by performing pixelwise operations on the input image. This allows it to overcome the shortcomings of the CCN based approaches namely, miss-predictions at the boundaries and blurriness of the saliency maps.
In this section, we will discuss some of the FCN based approaches.

Liu et al. [28] designed a neural network consisting of two parts to predict the saliency map. The first subnet, a deep hierarchical saliency network (DHSNet), acts as an encoder network and predicts coarser global features. The coarser global features are then refined using the second subnet, hierarchical recurrent convolutional neural network (HRCNN), to obtain finer local

features.

Kümmerer et al. [29] proposed the first transfer learning model for saliency prediction. Their model DeepGaze is a modification of AlexNet architecture [30]. DeepGaze omitted all of the fully connected layers and passed the features of the convolutional layers to a linear model as input to learn the weights.

Huang et al. [31] introduced a deep neural network (DNN) model which tries to reduce the semantic gap present between the predicted saliency map and the humans' behavior. They redesigned the DNN for object recognition and used it for saliency prediction. In this approach, they used the saliency evaluation metric as the loss function to train their model and integrated the finer and coarser representation of the input image.

Pan et al. [4] designed a shallow and a deep convolutional model, trained end-to-end to detect the saliency in an image. The shallow network is trained from scratch and the deep network is trained using transfer learning.

Pan et al. [5] designed a generative adversarial network (GAN) to predict the saliency map. They used perceptual loss function to train their model. The generator model used is an encoder-decoder architecture, where the encoder is a VGG network [22] and the decoder is VGG network stacked in reverse order. The discriminator used is a shallow convolutional network.

2.4 Advantages of FCN based approaches

The advantages of FCN based approaches for predicting the saliency map are as follows:

1. In CCN based approaches the local and the global features are incorporated either explicitly or implicitly. The FCN based approaches learn both these features internally allowing it to have a better understanding of the semantic information for predicting the saliency maps.

2. The FCN networks allows the usage of the pre-trained network to fine-tune the models for predicting better saliency maps.

3. The FCN networks are diverse and this allows the researchers to tailor make an architecture using different differentiable functions for detecting the salient objects.

The FCN based models achieve better saliency predictions compared to the CCN models. In spite of all the success, FCN models are still far from being the go-to model in the area of saliency prediction. FCN models face difficulty when the image contains a low contrast between the background and the foreground, transparent objects, cluttered objects, and complex backgrounds, as depicted in [32]. This demands for better saliency prediction models in the future. Hence, saliency prediction continues to remain an evergreen field for research.

Chapter 3

ConvNets for Saliency Prediction

"A humble knowledge of thyself is a surer way to God than a deep search after learning." – Thomas a Kempis

In this chapter, we will delve into the first end-to-end, data-driven CNN architecture for saliency prediction. The chapter will explore two network architectures proposed by Junting Pan, et al. in [4].

3.1 Objective of the chapter

The objective of this chapter is to show the implementation of the two neural networks described in [4], namely, ShallowNet and DeepNet, which are used to predict the saliency map for any given image.

3.2 What is a saliency map?

The saliency map is an output of a saliency model and represents the likelihood of visual attention present in the input image. The visual attention, in general, means the points in the image where human eye gaze is usually focused.

3.3 Why deep learning?

In the era prior to deep learning (DL), the problem of saliency prediction was solved using traditional hand-crafted feature extractors. The feature extractors were developed with the help of neurological studies. These hand-crafted features had poor generalization abilities. They fail to perform well when the images have complex texture, cluttered background or high contrast components.

On the other hand, CNNs became popular DL model as they started providing state-of-the-art results for computer vision tasks such as object segmentation [33], image classification [34] etc. CNN based models outperformed hand-crafted methods as they learn multiple features due to their hierarchical design. This hierarchy of layers in CNNs makes it easier for the CNN model to locate the salient objects in an image. The low-level features determine the edge information, and the high-level feature extract the semantic knowledge about the salient object.

Junting Pan, et al. chose to address the saliency problem as a regression problem rather than a classification problem. The classification problem needs the class labels to be orthogonal, i.e. if a pixel is predicted as salient, then it cannot be for sure non-salient. In the case of regression, each pixel is assigned a saliency score to determine its saliency.

3.4 Shallow ConvNet

Shallow convnet is a lightweight CNN model with less number of parameters trained for predicting saliency from scratch. The model is trained using SALICON dataset [6] which contains 10,000 training images along with their corresponding saliency maps.

3.4.1 Architecture

The ShallowNet consists of five weight layers, where the first three are convolutional layers and the rest two are fully connected layers. Rectified non-linear units (ReLU) and maxpool units follow the convolutional layers. The ReLU layer applies non-linearity to each element and the maxpool layer reduces the size of the input feature progressively while extracting the important features such as edges. The ShallowNet architecture is shown in Figure 3.1

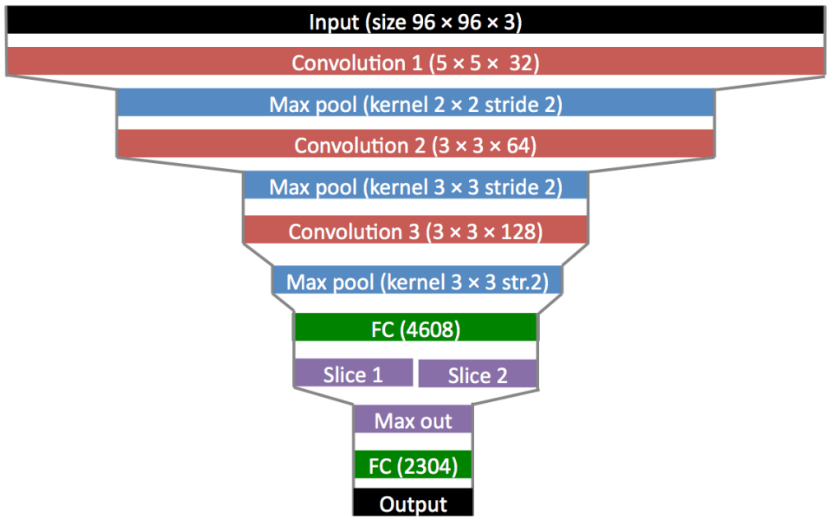


Figure 3.1: **The ShallowNet Architecture**

The ShallowNet architecture consists of about 64 million learnable parameters. The input to the ShallowNet is an image of size 96×96. After passing through three convolutional layers and three maxpool layers, the feature size becomes 10×10. The feature set then passes through a fully connected layer which outputs 4,096 activations. These $4,096$ activations are divided into two slices each consisting of $2,304$ activations and Maxout [35] is applied on each of the slices. In the Maxout layer, the maximum activation from the corresponding pair of slices is considered, this helps in reducing the overfitting. After the Maxout layer, the remaining $2,304$ activations passes through another fully connected layer which outputs the same number of

activations. The final $2,304$ activations are reshaped to generate a saliency map of size 48×48.

3.4.2 Training

The ShallowNet model is trained for 1,200 epochs. The initial learning rate is set to 3×10^{-2} and is gradually decreased to 1×10^{-4}. In every layer, the weights are initialized using Gaussian distribution whose mean and standard deviation are set to zero and 0.01 respectively and the biases are initialized with a constant value of 0.1. The weight decay and the momentum term is set to 5×10^{-4} and 0.9 respectively. Stochastic gradient descent (SGD) [36] with Nesterov momentum is used for training. The Nesterov momentum aided in faster convergence of the model. The data augmentation methods used are random horizontal flipping and normalizing pixel values to $[0, 1]$ range. Data augmentation is applied to the images and their saliency maps. The loss function used is Euclidean distance also known as L2 norm.

3.4.3 Results

In this subsection we compare the results of our implementation of ShallowNet with the ground truth saliency map qualitatively. In Figure 3.2, the first column consists of the query images, the second column consists of the ground truth saliency maps for the corresponding images and the third column shows the predicted saliency maps. The predicted saliency maps show that our implementation of ShallowNet is reasonably good at predicting the actual salient objects; however, it is also giving some saliency score to the background objects.

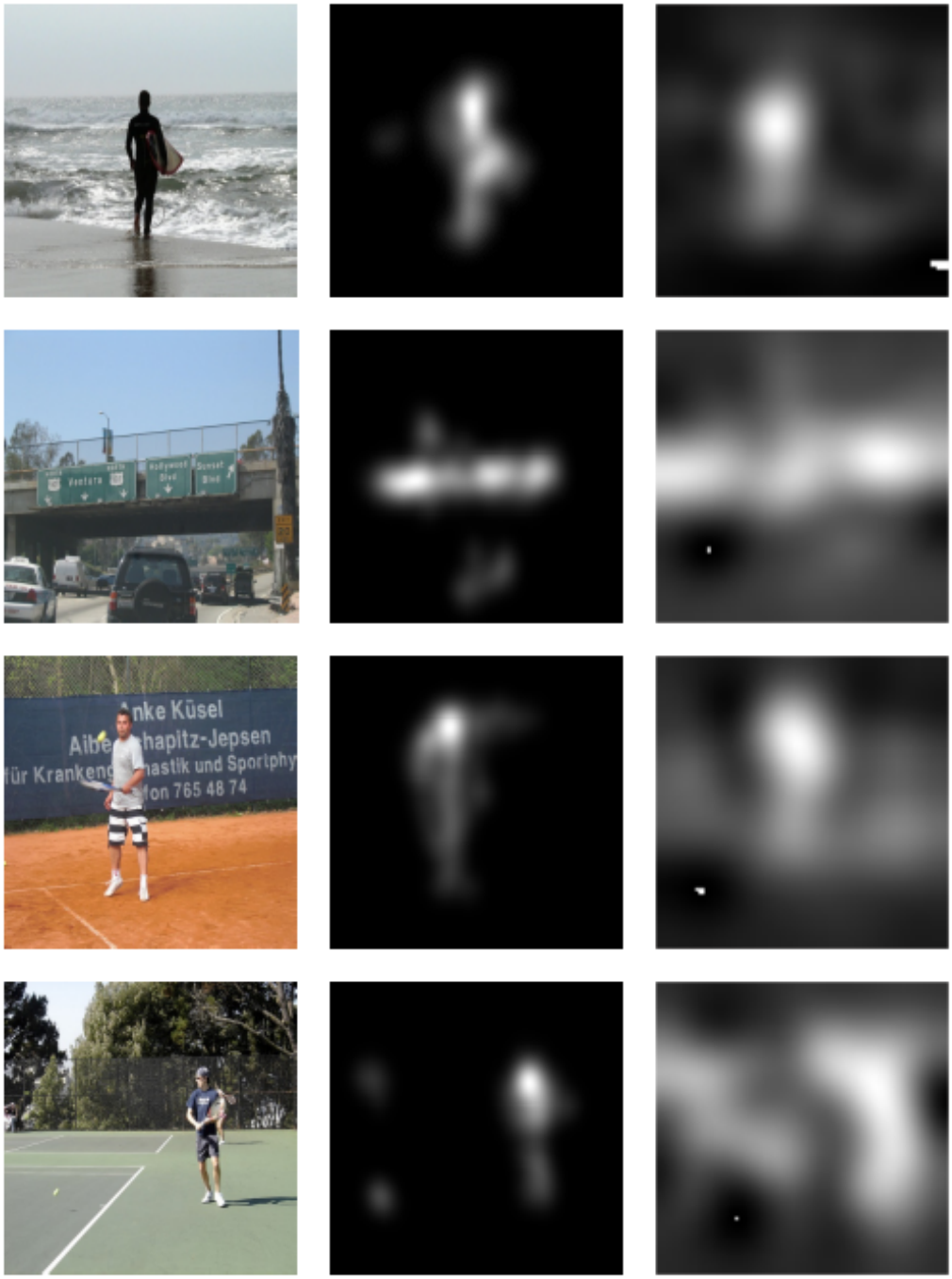


Figure 3.2: **Results of our implementation of ShallowNet Query image (left), Ground truth (centre) and Saliency map predicted using our implementation (right).**

3.5 Deep ConvNet

The following observations inspire DeepNet architecture:

1. In image classification model, the earlier layers converge after a few epochs [37].

2. On visualizing the features learned by earlier layers [38], it indicates that the lower layers tend to learn low-level computer vision tasks such as texture detection, color detection etc.

These observations led to the hypothesis that, the lower layers trained for image classification is reuseable for saliency prediction through transfer learning [39]. The design of the DeepNet architecture is such that it can use the pre-trained parameters from the image classification model with the layers designed for saliency prediction.

3.5.1 Architecture

As the name suggests, DeepNet is a deeper architecture compared to the ShallowNet. It consists of 10 weight layers and has a total of 25.8 million parameters. DeepNet has 60% fewer parameters compared to the ShallowNet, even though it has more layers. The reason for the reduction in parameters is the exclusion of the fully connected layers in DeepNet. Figure 3.3 illustrates the DeepNet architecture.

The input to the DeepNet model is an image of size 320×240. The model is trained using transfer learning. Weights of the first three convolution layers of the DeepNet are initialized with *VGG_CNN_M* model used in [40]. The weights of the remaining layers are randomly initialized using He initialization [41]. The ReLU layer follows every convolutional layer and applies elementwise non-linear activation. The feature maps size reduces to approximately one-fourth of input image size by the end of the second max-pool layer. The feature shape becomes $37 \times 17 \times 1$ after the input image

passes through all the convolutional layers. Finally, the deconvolution layer generates the saliency map whose size is the same as that of the input image.

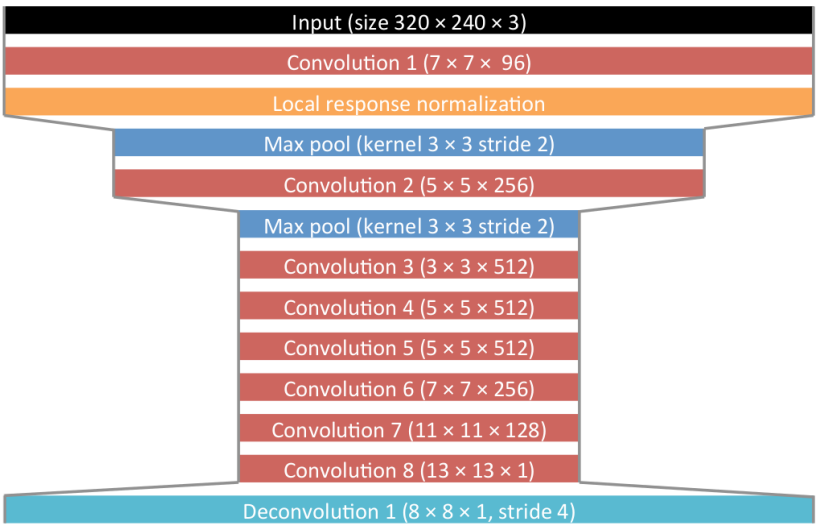


Figure 3.3: **The DeepNet Architecture**

3.5.2 Training

The images and their corresponding saliency maps in the SALICON dataset are of size 640×480. These are reduced by a factor of 2 and their mean value across all the channels are computed. The images and their saliency maps are zero-centered by subtracting their mean values respectively. After zero-centering the pixel values are scaled to $[-1, 1]$ interval. The base learning rate is 0.01 as in the case of ShallowNet. The learning rate is set to 1.3×10^{-7} by normalizing the base learning rate with the total number of predictions made per image. The authors claim that setting the learning rate as $0.01/76,800$ helps in faster convergence of the model. The learning tends to diverge when larger learning rate is used for training. The model is trained for $24,000$ iterations and the learning rate is reduced by a factor of 2 after every 100 iterations. The weight decay parameter is set to 5×10^{-4} and Euclidean Norm is used as the loss function. SGD with Nesterov momentum is used for updating the weights of the parameters.

3.5.3 Results

In this subsection, we compare the results of our implementation of DeepNet with the ground truth saliency maps visually.

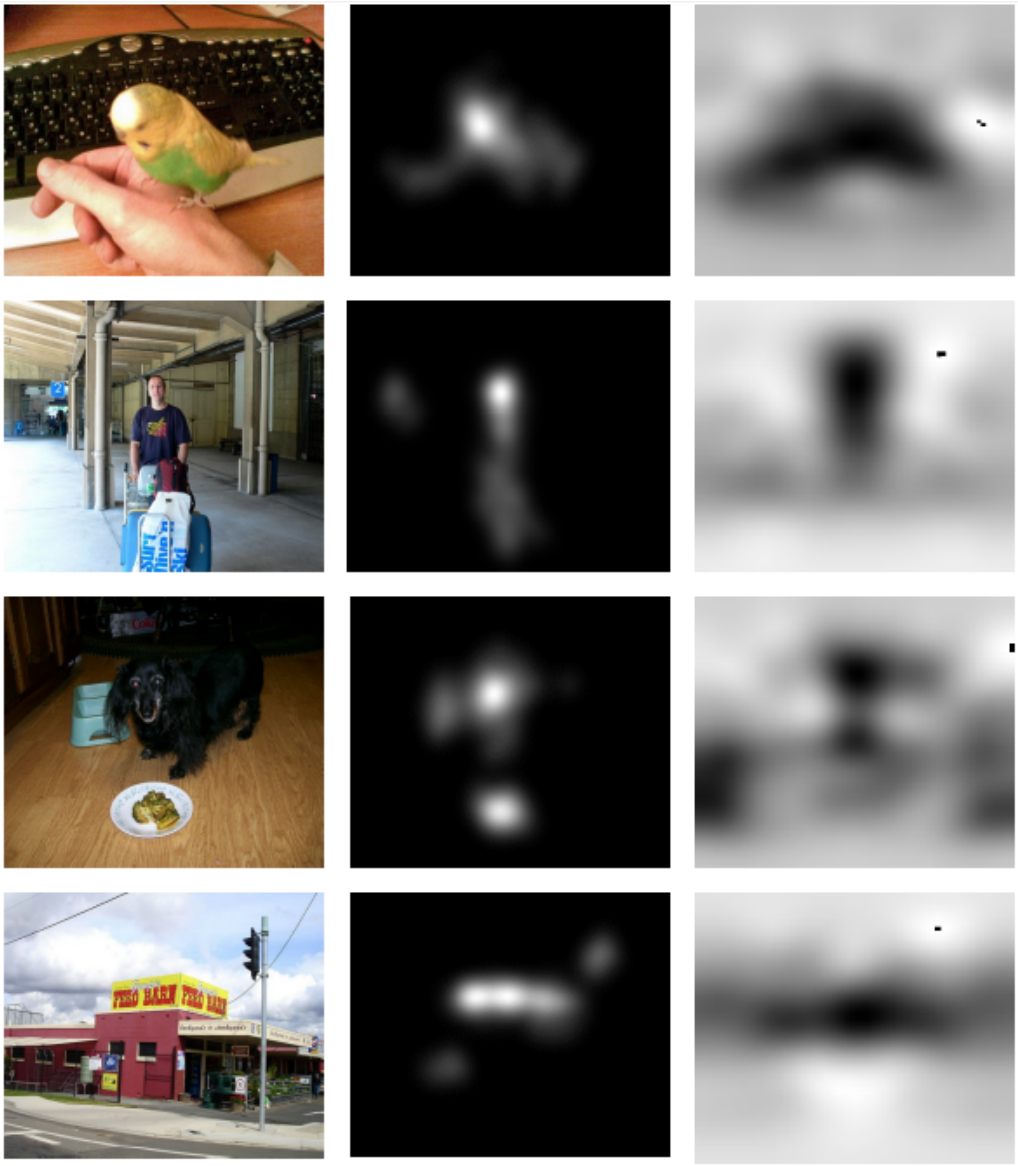


Figure 3.4: **Results of our implementation of DeepNet Query image (left), Ground truth (centre) and Saliency map predicted using our implementation (right).**

	Saliency Prediction Models			
	ShallowNet Implementation		DeepNet Implementation	
Metrics	**Author's**	**Our 's**	**Author's**	**Our's**
CC	0.58	0.49	0.61	0.54
AUC Shuffled	0.67	0.59	0.73	0.68
AUC Borji	0.83	0.71	0.86	0.79

Table 3.1: **Comparison of Our implementation of the models with the Author's implementation**

In Figure 3.4, the first column consists of the query images, the second column consists of the ground truth saliency maps for the corresponding images and the third column shows the predicted saliency maps. In the predicted saliency maps, the black region denotes the salient areas. We inverted the saliency maps for better visualization.

In Table 3.1 we quantitatively compare the results of our implementation with that of the authors against the metrics Shuffled Area under Curve (sAUC), Area under Curve Borji (AUC-B) and Pearson's Correlation Coefficients (CC). These metrics are explained in Appendix A. Though our results are closer to the actual ground truth qualitatively, they differ from that of the authors. The reason for this can be attributed to the hyper-parameter tuning and the difference in framework used for implementation of the networks. The authors themselves suggested these reasons for the low performance of our implementation.

The author's implementation of ShallowNet and DeepNet are publically available in GitHub (Author's Implementation)[1]. The authors chose to implement ShallowNet using Lasagne and DeepNet using Caffe. We used PyTorch framework [42] to implement both the models. Our implementation is also available in GitHub (Our Implementation) [2]

[1]https://github.com/imatge-upc/saliency-2016-cvpr
[2]https://github.com/Goutam-Kelam/Visual-Saliency/tree/master/Deep_Net

Chapter 4

GANs for Saliency Prediction

"The function of wisdom is to discriminate between good and evil." – Marcus Tullius Cicero

4.1 Introduction

In the previous chapter, ConvNets for Saliency Prediction, two convolutional neural networks are used to predict the saliency map. These CNNs use L2 norm loss function and are evaluated against the metrics Shuffled AUC, AUC Borji and CC. Apart from these metrics, there are other metrics available for evaluating saliency prediction such as Similarity (SIM), Kullback-Leibler divergence (KL), Normalized Scanpath Saliency (NSS) etc. Variety in metrics leads to a variety of loss functions. Different authors [43], [44], [45] use different loss functions to predict saliency that perform well on a subset of metrics. The above mentioned metrics are explained in Appendix A.

In this chapter we implement a model that uses agnostic loss [5] and perform well when evaluated on a variety of metrics. We replicate the results corresponding to the emphasis made in [5] about how simple binary classification between real and synthetic samples can facilitate saliency prediction to perform well on a wide range of metrics. The model is a generative adversarial network (GAN) and is known as SalGAN.

Generative adversarial networks (GANs) [46] are generally used to synthesize images which have realistic data distribution. A conventional GAN model consists of two competing networks namely, a generator and a discriminator. The jobs of these networks are exactly as their name suggests. The generator generates samples whose data distribution is the same as that of the training dataset. The discriminator discriminates between the sample synthesized by the generator and the real sample drawn from the training dataset. The training of the GAN models proceeds by training the discriminator and the generator alternatively.

The idea of using GANs for saliency prediction has few challenges of its own. These challenges are listed below:

1. In traditional GANs, the input to the generator is some random noise and it tries to generate realistic images. In case of SalGAN, the input to the generator is an image and it must learn to generate a realistic saliency map.

2. SalGAN desires the generated saliency map must correspond to the input image. Hence, SALGAN provides both the image as well as the generated saliency map as input to the discriminator. Traditional GANs does not have such requirements and only provide the generated images as input to the discriminator.

3. Traditional GANs does not have any ground truth to compare its generated images; however, in the case of SalGAN, the ground truth saliency maps are accessible for comparison.

4.2 The SalGAN architecture

The SalGAN architecture consists of two CNNs: a generator, SalGAN, which synthesizes saliency maps and a discriminator that differentiates between the synthesized and the real saliency maps. Figure 4.1 illustrates the overall SalGAN architecture.

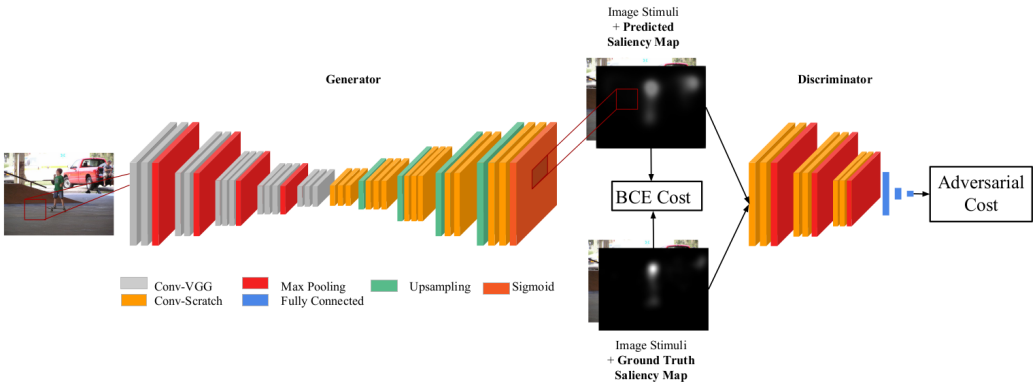


Figure 4.1: **Overall architecture of the SalGAN**

4.2.1 Generator

SalGAN comprises of an encoder-decoder model. The encoder, as well as the decoder, are convolutional architectures. The encoder consists of convolutional layers followed by maxpool layers which aid in the reduction in the size of the feature maps. The decoder consists of upsampling layers followed by convolutional layers which aid in generating the saliency map which has the same size as that of the input image.

The encoder is a VGG-16 [47] architecture whose fully connected layers and the final pooling layer is removed. The weights of the encoder are initialized with the weights of VGG-16 network which is trained for image classification on the ImageNet dataset [48]. During the training of encoder, except the last two set of convolution layers, the earlier layers are not updated. The fixing of weights introduces a trade-off between performance and computational cost. Figure 4.2 shows the encoder network in detail.

The decoder is similar to the encoder part whose convolutional layers are in reverse order and the upsampling layers replace the maxpooling layers. The weights of the decoder are initialized stochastically. Figure 4.3 shows the decoder network in detail. The output saliency map is predicted when the feature maps pass through a final 1×1 layer followed by a sigmoid non-linearity function. The saliency map generated is of the same size as

layer	depth	kernel	stride	pad	activation
conv1_1	64	1×1	1	1	ReLU
conv1_2	64	3×3	1	1	ReLU
pool1		2×2	2	0	-
conv2_1	128	3×3	1	1	ReLU
conv2_2	128	3×3	1	1	ReLU
pool2		2×2	2	0	-
conv3_1	256	3×3	1	1	ReLU
conv3_2	256	3×3	1	1	ReLU
conv3_3	256	3×3	1	1	ReLU
pool3		2×2	2	0	-
conv4_1	512	3×3	1	1	ReLU
conv4_2	512	3×3	1	1	ReLU
conv4_3	512	3×3	1	1	ReLU
pool4		2×2	2	0	-
conv5_1	512	3×3	1	1	ReLU
conv5_2	512	3×3	1	1	ReLU
conv5_3	512	3×3	1	1	ReLU

Figure 4.2: **The encoder part of SalGAN**

that of the input image. In the SalGAN architecture, ReLU is used as the non-linearity function after every convolutional layer.

4.2.2 Discriminator

Figure 4.4 illustrates the discriminator network in detail. The discriminator consists of six convolutional layers with a kernel size of 3×3. A ReLU layer follows each of the convolutional layer, and after every set of two convolutional layers, a maxpool layer follows which reduces the feature size by half. Finally, three fully connected layers follow the convolutional layers. Tanh is used as an activation function for the first two fully connected layers whereas the final fully connected layer uses sigmoid.

conv6_1	512	3×3	1	1	ReLU
conv6_2	512	3×3	1	1	ReLU
conv6_3	512	3×3	1	1	ReLU
upsample6		2×2	2	0	-
conv7_1	512	3×3	1	1	ReLU
conv7_2	512	3×3	1	1	ReLU
conv7_3	512	3×3	1	1	ReLU
upsample7		2×2	2	0	-
conv8_1	256	3×3	1	1	ReLU
conv8_2	256	3×3	1	1	ReLU
conv8_3	256	3×3	1	1	ReLU
upsample8		2×2	2	0	-
conv9_1	128	3×3	1	1	ReLU
conv9_2	128	3×3	1	1	ReLU
upsample9		2×2	2	0	-
conv10_1	64	3×3	1	1	ReLU
conv10_2	64	3×3	1	1	ReLU
output	1	1x1	1	0	Sigmoid

Figure 4.3: **The decoder part of SalGAN**

layer	depth	kernel	stride	pad	activation
conv1_1	3	1×1	1	1	ReLU
conv1_2	32	3×3	1	1	ReLU
pool1		2×2	2	0	-
conv2_1	64	3×3	1	1	ReLU
conv2_2	64	3×3	1	1	ReLU
pool2		2×2	2	0	-
conv3_1	64	3×3	1	1	ReLU
conv3_2	64	3×3	1	1	ReLU
pool3		2×2	2	0	-
fc4	100	-	-	-	tanh
fc5	2	-	-	-	tanh
fc6	1	-	-	-	sigmoid

Figure 4.4: **The discriminator architecture**

4.3 Training

The SalGAN architecture uses perceptual loss [49], which is a combination of content loss and adversarial loss. The content loss computes per pixel similarity between the predicted saliency map and the ground truth saliency map. The adversarial loss determines how good the discriminator is able to distinguish the generated saliency map as real or fake.

4.3.1 Content Loss

Let us consider an image I whose height and width be H and W respectively. Let the dimension of the image be defined as $N = H \times W$. Let S and $\tilde{S}$ denote the ground truth saliency map and the predicted saliency map respectively. Let saliency maps be treated as probabilities of each pixel being salient. In order to achieve this, the ground truth pixel values are scaled to $[0, 1]$ interval. The loss function used is binary cross entropy (BCE). The content loss is the sum of BCE losses across every pixel and is defined as:

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{k=1}^{N}(S_k log(\tilde{S}_k) + (1 - S_k)log(1 - \tilde{S}_k))$$

where S_k and $\tilde{S}_k$ represent the probability of the kth pixel being salient in the ground truth and predicted saliency map respectively. In summary, the content loss is computed by comparing the similarity between the predicted saliency map with that of the ground truth saliency map for every pixel.

4.3.2 Adversarial Loss

In a conventional GAN model, the discriminator decides whether its input is real or fake. Similarly in SalGAN too, the job of the discriminator is to decide whether the saliency map which is given as its input is predicted or it belongs to the ground truth. The loss function for the discriminator architecture is defined as:

$$\mathcal{L}_{Dis} = L(\mathcal{D}(I, S), 1) + L(\mathcal{D}(I, \tilde{S}), 0)$$

where L denotes BCE loss, the number 1 represents the target belongs to ground truth and 0 represents it is predicted. $\mathcal{D}(I, \tilde{S})$ represent the probability of fooling the discriminator (i.e. given a predicted saliency map as input, the discriminator classifies it as real). $\mathcal{D}(I, S)$ represent the probability that given a ground truth saliency map, the discriminator predicts as real. The loss function used in adversarial training is defined as:

$$\mathcal{L} = \alpha \times \mathcal{L}_{BCE} + L(\mathcal{D}(I, \tilde{S}), 1)$$

The loss function $\mathcal{L}$ aids in improving the convergence rate and stability of the adversarial training.
The training of the saliency prediction network is done in two phases:

1. Bootstrap the SalGAN architecture. This is done by training it for 15 epochs with BCE as the loss function. The bootstrapping aids the generator to synthesize meaningful images.

2. After bootstrapping we add the discriminator architecture to the SalGAN model and start the adversarial training.

During the adversarial training, the input to the SalGAN is an RGB image of shape $256 \times 192 \times 3$. The SalGAN generates a saliency map of the same size as that of the input image. The saliency map is concatenated with the input image and is passed as input to the discriminator architecture. Thus, the input to the discriminator is an RGBS image of shape $256 \times 192 \times 4$. While training the saliency prediction network, the SalGAN network and the discriminator are trained in alternative iterations. The network is trained with the SALICON dataset [5]. The weight decay parameter is set to 1×10^{-4} for both generator and discriminator networks. The learning rate is set to 3×10^{-4} and AdaGrad is used for training the networks. The authors used a batch size of 32, but we used a batch size of 8 in our implementation of SalGAN due to limited availability of resources. The entire network is trained for 120 epochs, i.e. 150000 iterations and the hyperparameter α is set to 5×10^{-3}.

4.4 Results

In this section we compare the results of our implementation of the saliency prediction model defined in [5] with the ground truth saliency maps qualitatively.

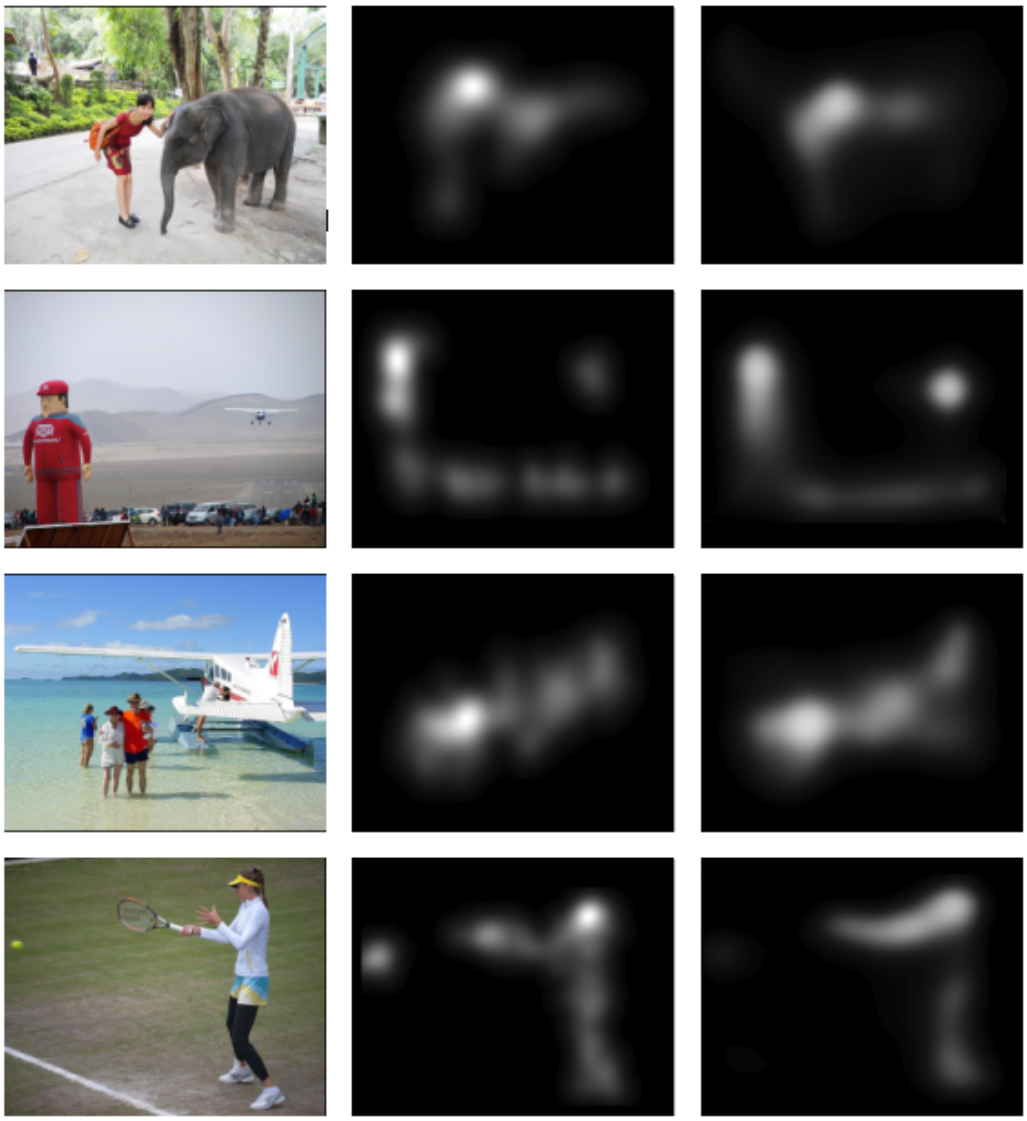


Figure 4.5: **Results of our implementation of the SalGAN model Query image (left), Ground truth (centre) and Saliency map predicted using our implementation (right).**

	SalGAN Implementation	
Metrics	**Author's**	**Our**
AUC Borji	0.859	0.741
Shuffled AUC	0.773	0.651
NSS	2.560	2.405
CC	0.786	0.814
Similarity	-	0.720
KL	-	0.424

Table 4.1: **Comparison of our implementation of SalGAN with the Author's implementation**

In Figure 4.5, the first column consists of the query images, the second column consists of the ground truth saliency maps for the corresponding images and the third column shows the predicted saliency maps. The query images belong to the SALICON validation set.

In Table 4.1 we quantitatively compare our implementation of the SalGAN model with the author's implementation against the metrics CC, AUC Shuffled, AUC Borji and NSS. These metrics are explained in Appendix A.

We also performed the experiment conducted by the authors which compare the improvement in saliency map generation by adding adversarial training to perceptual loss. In this experiment, the SalGAN model is trained without using the discriminator to predict the saliency map and uses only the content loss i.e. BCE. Table 4.2 quantitatively assesses the results of the experiment performed against various saliency metrics. Figure 4.6 qualitatively illustrates the results of our comparison of perceptual loss versus content loss.

In Figure 4.6, the first column consists of the query images, the second column consists of the ground truth saliency maps for the corresponding image, the third column shows the predicted saliency maps using perceptual loss and the fourth column shows the predicted saliency maps using content loss.

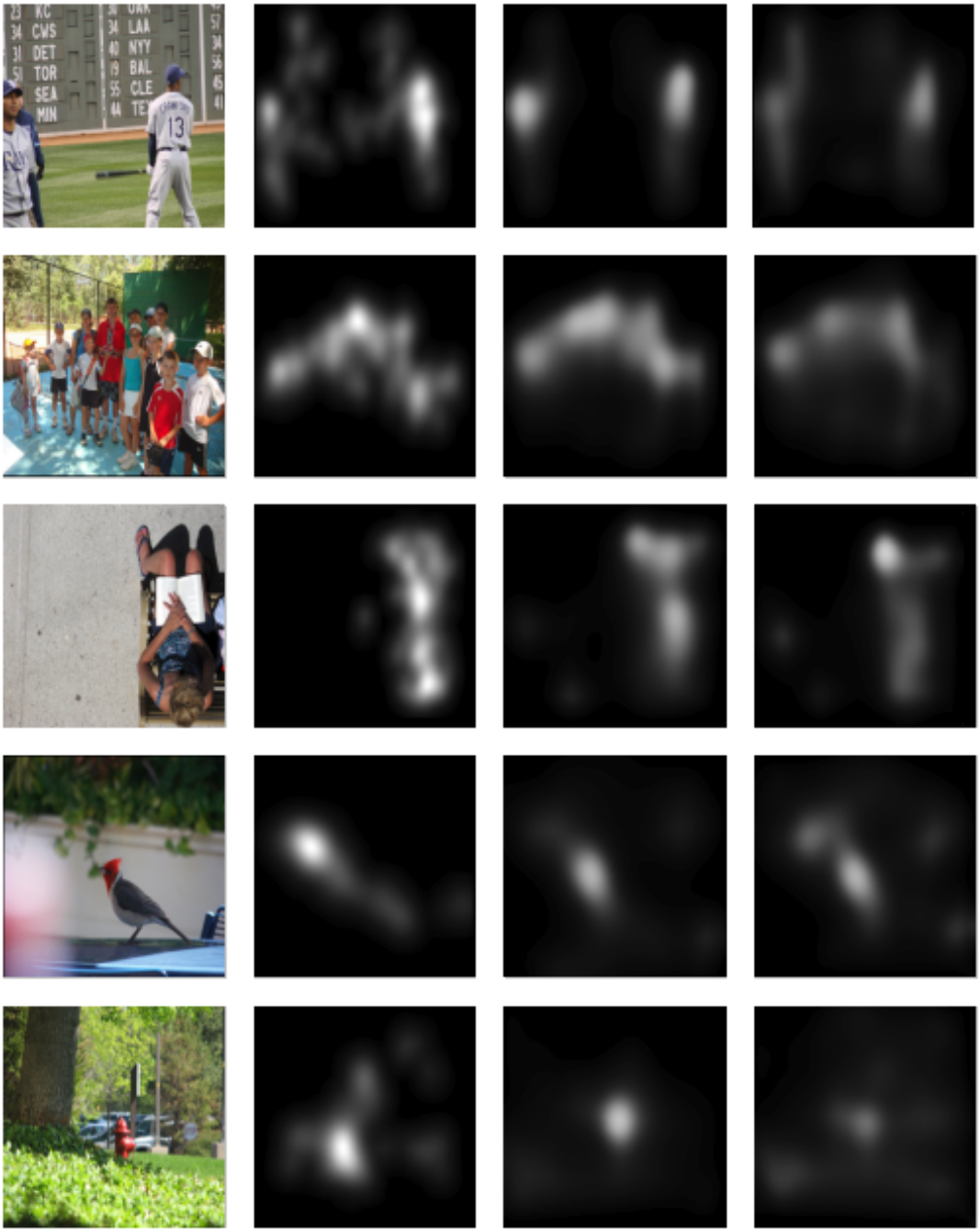


Figure 4.6: **Qualitative assessment: Perceptual Loss Vs. Content Loss**

Query image (left), Ground truth (centre left), Saliency map predicted using perceptual loss (centre right) and Saliency map predicted using conceptual loss (right).

	SalGAN Implementation	
Metrics	**with Content Loss**	**with Perceptual Loss**
AUC Borji	0.697	0.741
Shuffled AUC	0.634	0.651
NSS	2.483	2.405
CC	0.801	0.814
Similarity	0.707	0.720
KL	0.417	0.424

Table 4.2: **Quantitative assessment: Perceptual Loss Vs. Content Loss**

The query images belong to the SALICON validation set. It is clear from the quantitative, Table 4.2, as well as qualitative, Figure 4.6, results that perceptual training aids in better prediction of salient regions. One observation that can be made when using only content loss is that all pixels deemed as salient are almost equally likely with low confidence (low intensity).

The author's implementation of SalGAN is publically available in GitHub (Author's Implementation) [1]. The authors chose to implement SalGAN using Lasagne with Theano backend. We used PyTorch framework [42] to implement the model. Our implementation is also available in GitHub (Our Implementation) [2].

[1]https://github.com/imatge-upc/saliency-salgan-2017
[2]https://github.com/Goutam-Kelam/Visual-Saliency/tree/master/SalGAN

Chapter 5

Improvisation of GANs for Saliency Prediction

"Self study leads to improvisation and improvisation leads to evolution." – Oscar Auliq-Ice

5.1 Introduction

In the previous chapter, GANs for saliency prediction, adversarial training is used to predict the saliency maps. The SalGAN model consists of two competing models, namely generator and a discriminator, which uses perceptual loss [49] for training and allow the predicted saliency maps to be robust when compared against different saliency metrics. The saliency maps predicted by SalGAN model though being robust cannot preserve the shape of the salient object.

Conventionally, only the final layers predict the saliency maps independent of other layers. When an image passes through a neural network, the feature maps are constantly refined by the layers. The final layers use these enriched feature maps to make predictions about the salient objects in the image. Though this method presents good results, making predictions independent of other layers does not take multi-scale semantics into consideration. The major disadvantages of using only the high-level feature maps for

saliency prediction are:

- The boundaries of the predicted salient regions are a blur.
- The smaller and thinner salient regions get overlooked by the high-level feature maps (due to pooling layers).

The disadvantage of using only lower feature maps may lead to loss of the larger context in the image. The aim is to generate a saliency map which preserves the context of the image along with taking small salient regions into consideration. In order to generate such a saliency map, the easiest way is to predict a saliency map using the low-level feature maps and another saliency map using the high-level feature maps, then combine both these saliency maps to generate a single saliency map. The saliency map generated in such a way will have the desired qualities. The disadvantages of both high-level features and low-level features are mitigated to a large extent by combining these feature maps as one can complement the other.

5.2 Overview of the proposed method

We propose an improvised GAN architecture for accurate prediction of saliency maps. The proposed method leverages on the salient features learned across multiple layers of the deep neural network. Our architecture is inspired by $SalGAN$ [5] and the R^3Net [43] architectures used for saliency prediction. The adversarial training used in SalGAN aids in the generation of robust saliency maps. The recurrent residual refinement network (R^3Net) consists of residual refinement blocks (RRBs) which compute the residual by generating intermediate saliency maps and comparing it with the ground truth. These RRBs uses the residue to refine the subsequent intermediate saliency maps. The RRBs predict the saliency maps by alternative usage of low and high level integrated features.

Our architecture consists of a generator and a discriminator. Given an image to the generator, it extracts low-level and high-level features by passing

the image through a feature extractor network. It then integrates all the lower level and higher level features respectively. The integrated lower and higher features are used to predict intermediate saliency maps respectively. The integrated feature maps are further concatenated and used to predict a combined saliency map as output. The intermediate saliency maps are used to compute the content loss and the combined saliency map becomes the input to the discriminator for adversarial training.

5.3 Architecture

The proposed architecture consists of two deep CNNs: a generator and a discriminator. The generator tries to extract low-level and high-level salient features and fuse them to form a high-resolution saliency map. The discriminator tries to differentiate between the synthesized high-resolution saliency map and the real saliency map which is the ground truth. Perceptual loss [49] is used to train our model. Figure 5.1 illustrates the overall architecture of our proposed model.

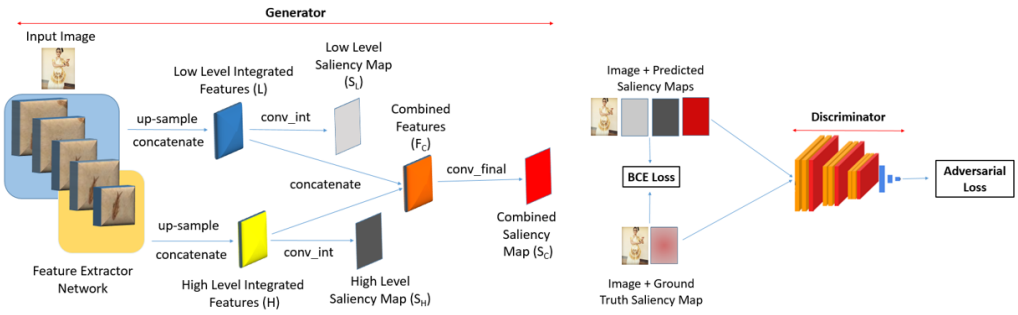


Figure 5.1: **The overall architecture of our model**

5.3.1 Generator

The generator in our improvised GAN network uses ResNeXt [50] as the feature extractor. Given an image to the generator, the ResNeXt model

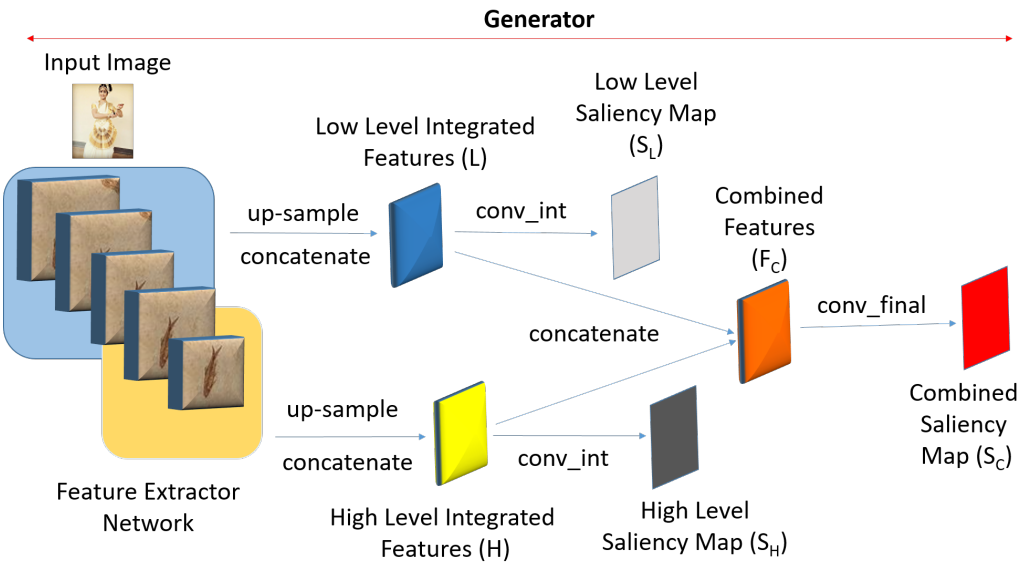


Figure 5.2: **Generator model**

yields a set of feature maps. These feature maps contain low-level as well as high-level semantic information of varying scales. The low-level features and the high-level features are extracted by the shallow layers (grouped in light blue) and the deep layers (grouped in light yellow) respectively. These features are up-sampled and concatenated to produce low-level integrated feature map, L (denoted in dark blue) and high-level integrated feature map, H (denoted in dark yellow) respectively. The low-level integrated feature, L, and the high-level integrated feature, H, are passed through a shallow convolutional network (denoted as conv_int) to produce low-level saliency map, S_L (denoted in light grey) and high-level saliency map, S_H (denoted in dark grey) respectively. The low-level integrated feature, L, and the high-level integrated feature, H, are further combined to produce a richer feature map, F_C (denoted in orange). The combined feature map, F_C, is then passed through another shallow convolutional network (denoted as conv_final) to produce a combined saliency map, S_C (denoted in red). The generator therefore produces three saliency maps for each input image. Figure 5.2 reproduces the generator model for easy reference.

Table 5.1 and Table 5.2 describes the detailed architecture of the shallow convolutional networks used to generate intermediate and final saliency maps.

Layer	In-channels	Out-channels	Kernel	Activation
Conv1	256	128	3 x 3	PReLU
BatchNorm1	128	-	-	-
Conv2	128	128	3 x 3	PReLU
BatchNorm2	128	-	-	-
Conv3	128	1	1 x 1	sigmoid

Table 5.1: **Architecture of shallow network (conv_int) which generate intermediate saliency maps**

Layer	In-channels	Out-channels	Kernel	Activation
Conv1	2	128	3 x 3	PReLU
BatchNorm1	128	-	-	-
Conv2	128	128	3 x 3	PReLU
BatchNorm2	128	-	-	-
Conv3	128	1	1 x 1	sigmoid

Table 5.2: **Architecture of shallow network (conv_final) which generate final saliency maps**

5.3.2 Discriminator

Table 5.3 illustrates the discriminator network in detail. The discriminator network used is that of the SalGAN network [5]. It consists of six convolutional layers with a kernel size of 3×3. A ReLU layer follows each of the convolutional layer, and after every set of two convolutional layers, a

maxpool layer follows which reduces the feature size by half. Finally, three fully connected layers follow the convolutional layers. Tanh is used as an activation function for the first two fully connected layers whereas the final fully connected layer uses sigmoid. The discriminator takes the combined saliency map, S_C, synthesized by the generator, and the image I as input and use it for adversarial training.

Layer	Out-channels	Kernel	Stride	Pad	Activation
Conv1_1	3	1 x 1	1	1	ReLU
Conv1_2	32	3 x 3	1	1	ReLU
Pool1	-	2 x 2	2	0	-
Conv2_1	64	3 x 3	1	1	ReLU
Conv2_2	64	3 x 3	1	1	ReLU
Pool2	-	2 x 2	2	0	-
Conv3_1	64	3 x 3	1	1	ReLU
Conv3_2	64	3 x 3	1	1	ReLU
Pool3	-	2 x 2	2	0	-
fc4	100	-	-	-	tanh
fc5	2	-	-	-	tanh
fc6	1	-	-	-	sigmoid

Table 5.3: **The discriminator architecture**

5.4 Training

Our improvised saliency prediction network uses perceptual loss [49] for training. The perceptual loss is a combination of content loss and adversarial loss. The content loss in our model is computed by combining the losses of

the intermediate saliency maps and the final saliency map with respect to the ground truth respectively. The adversarial loss determines the discriminator's ability to distinguish the combined saliency map, S_C, as real or fake. Figure 5.3 reproduces the loss computation diagram.

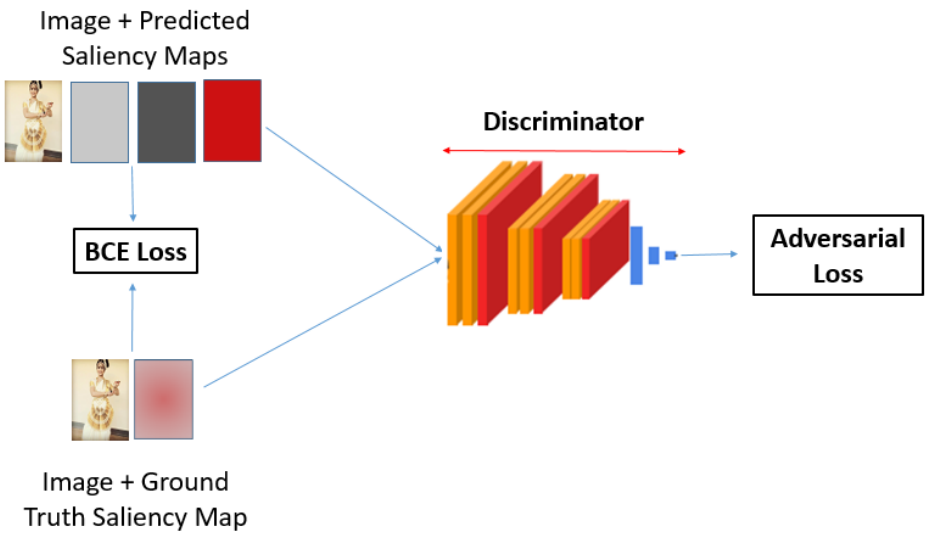


Figure 5.3: **Loss Computation**

5.4.1 Content Loss

Let us consider an image I whose height and width be H and W respectively. Let the dimension of the image be defined as $N = H \times W$. Let S denote the ground truth saliency map. Let S_L, S_H and S_C denote the low-level saliency map, high-level saliency map and the combined saliency map respectively. Let saliency maps be treated as probabilities of each pixel being salient. In order to achieve this, the ground truth pixel values are scaled to $[0, 1]$ interval. BCE is used as the loss function. The content loss is the sum of per pixel BCE losses of S_L, S_H and S_C with respect to S. The content loss is defined as:

$$BCE_{S_L} = -\frac{1}{N}\sum_{k=1}^{N}(S^k log({S_L}^k) + (1 - S^k) log(1 - {S_L}^k))$$

$$BCE_{S_H} = -\frac{1}{N}\sum_{k=1}^{N}(S^k log({S_H}^k) + (1 - S^k) log(1 - {S_H}^k))$$

$$BCE_{S_C} = -\frac{1}{N}\sum_{k=1}^{N}(S^k log({S_C}^k) + (1 - S^k) log(1 - {S_C}^k))$$

$$\mathcal{L}_{BCE} = BCE_{S_L} + BCE_{S_H} + BCE_{S_C}$$

where S^k and ${S_i}^k$, $i = \{L, H, C\}$; represent the probability of the kth pixel being salient in the ground truth and predicted saliency maps respectively. In summary, the content loss is computed by comparing the similarity between the predicted saliency maps with respect to the ground truth saliency map for every pixel.

5.4.2 Adversarial Loss

The adversarial loss is as defined for SalGAN [5] in chapter 4. The training procedure and the hyperparameter settings are same as used for adversarial training of SalGAN (refer to subsection 4.3.2 for details).

5.5 Results

In this section, we qualitatively and quantitatively report the results of our improvised GAN model for saliency prediction. The model is trained on MSRA10K dataset [51] and is tested on ECSSD [52] and DUT-OMRON [53] datasets. Figures 5.4, 5.5, 5.6 illustrates our results on MSRA10K, ECSSD and DUT-OMRON datasets. In the above mentioned figures, the first column consists of the query images, the second column consists of the ground truth saliency maps for the corresponding images and the third column shows the predicted saliency maps.

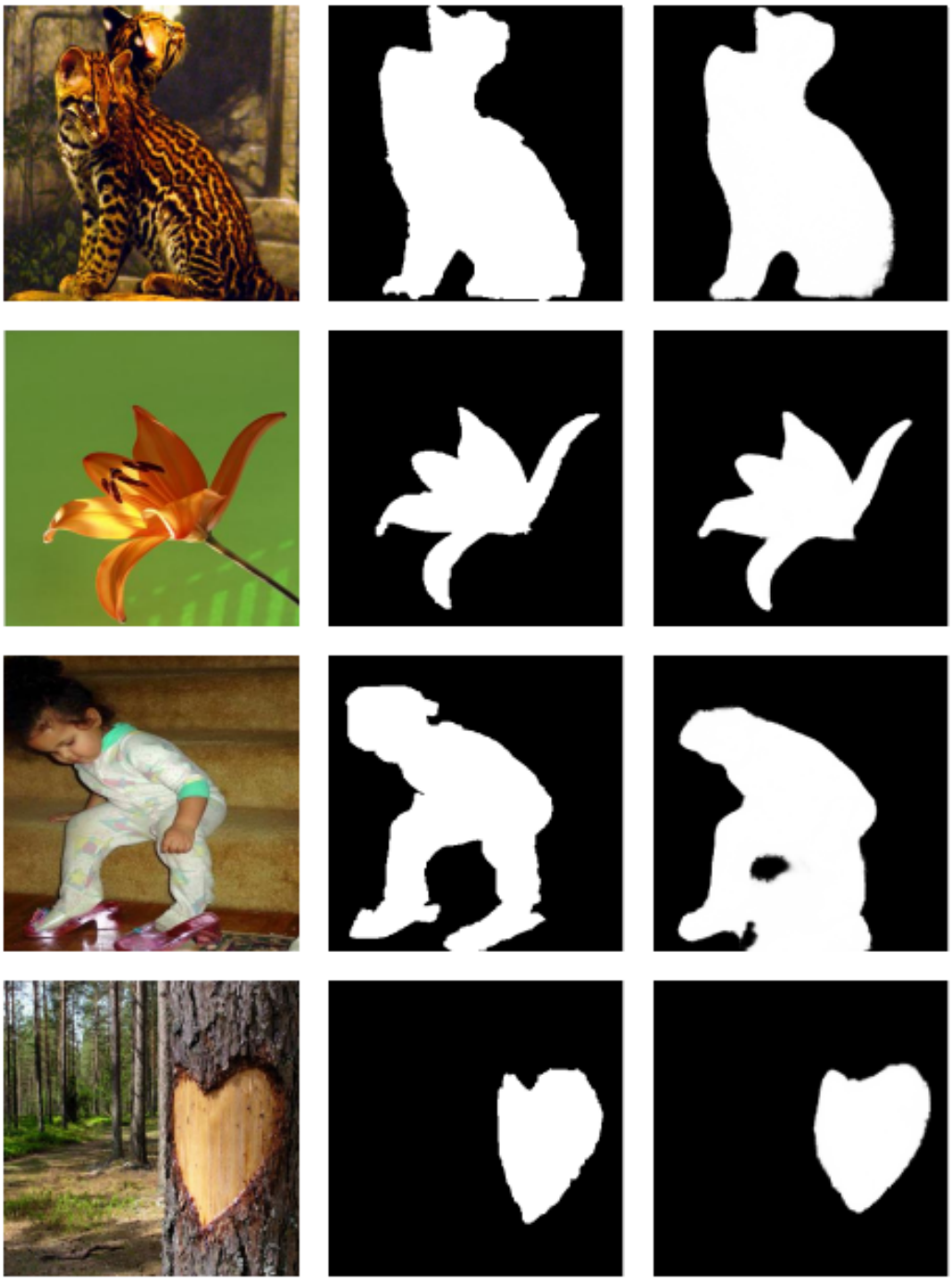


Figure 5.4: **Qualitative Result: MSRA10K dataset**
Query image (left), Ground truth (centre) and Saliency map predicted by our model (right).

Figure 5.5: **Qualitative Result: ECSSD dataset**
Query image (left), Ground truth (centre) and Saliency map predicted by our model (right).

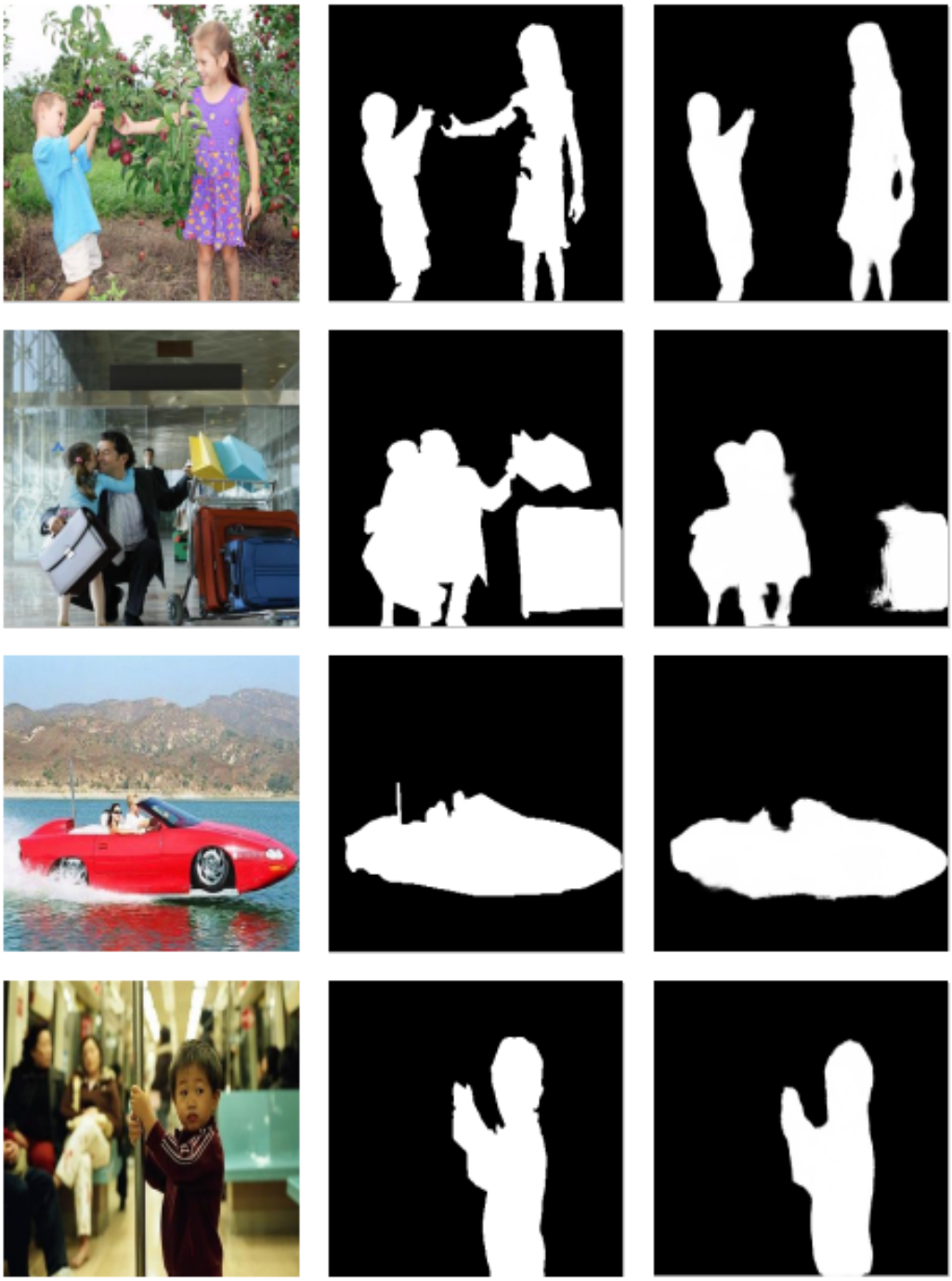


Figure 5.6: **Qualitative Result: DUT-OMRON dataset Query image (left), Ground truth (centre) and Saliency map predicted by our model (right).**

Figure 5.7: **Qualitative comparison of our model with the SalGAN model on MSRA10K dataset.**
Query image (left), Saliency map predicted by the SalGAN model (centre) and Saliency map predicted by our model (right).

Method	ECSSD		DUT-OMRON	
	F-measure	**MAE**	**F-measure**	**MAE**
MR [54]	0.736	0.189	0.610	0.187
wCtr* [55]	0.716	0.171	0.630	0.144
BSCA [56]	0.758	0.183	0.616	0.191
MC [57]	0.822	0.106	0.703	0.088
LEGS [58]	0.827	0.118	0.669	0.133
MDF [59]	0.831	0.108	0.694	0.092
ELD [60]	0.867	0.080	0.716	0.091
DS [61]	0.882	0.123	0.745	0.120
RFCN [62]	0.898	0.097	0.747	0.095
DCL [63]	0.898	0.071	0.757	0.080
DHSNet [64]	0.907	0.059	-	-
NLDF [65]	0.905	0.063	0.753	0.080
Improvised GAN (ours)	**0.912**	**0.053**	**0.759**	**0.076**

Table 5.4: **Comparison of our model with the state-of-the-art models for saliency prediction.**

We compared our proposed improvisation of GANs for saliency prediction with 12 of the state-of-the-art models in literature, using the F-measure and MAE metrics. Table 5.4 shows our proposed model outperforms its competitors on both the ECSSD [53] and DUT-OMRON [53] datasets. We conducted an experiment to highlight our contribution to the GANs for saliency prediction. We trained the SalGAN architecture [5] with MSRA10K dataset [51] for 120 epochs. The trained model is further used to predict the saliency maps for the same dataset. Figure 5.7 and Table 5.5 illustrate that our proposed method outperforms the SalGAN model. In Figure 5.7 the first column consists of the query images, the second column consists of the saliency maps predicted by SalGAN model and the third column shows the saliency maps predicted using our proposed model.

Method	MSRA10K	
	F-measure	**MAE**
SalGAN [5]	0.869	0.103
Our proposed method	**0.945**	**0.027**

Table 5.5: **Quantitative comparison of our model with the SalGAN model.**

We used PyTorch framework [42] to implement our model. The implementation of our proposed model is publicly available in GitHub. The proposed model trained on MSRA10K dataset is available at (Improvised GAN) [1] and the SalGAN model trained on MSRA10K dataset is available at (SalGAN_MSRA10K) [2]

[1]https://github.com/Goutam-Kelam/Visual-Saliency/tree/master/Improvised_GAN
[2]https://github.com/Goutam-Kelam/Visual-Saliency/tree/master/SalGAN_on_MSRA10K

Chapter 6

Conclusion & Future Work

"There is no real ending. It's just the place where you stop the story." – Frank Herbert

6.1 Conclusion

In this thesis, we have studied and implemented two deep learning methods to predict the saliency map of a given image. As our contribution, we designed an improvised version of the SalGAN model to predict the saliency maps.

The first method is an end-to-end convolutional network. The end-to-end convolutional networks are capable of learning the global semantics of the salient objects, thereby predict better saliency maps. We have implemented two end-to-end convolutional network models, namely, ShallowNet and DeepNet as described in [4]. Our implementation of ShallowNet and DeepNet performed as close to the author's implementation, both qualitatively and quantitatively; however, the models tend to assign a small amount of saliency score to the neighboring pixels also. The reason for this can be attributed to the usage of different hyperparameters and framework for the implementation of these models.

The second method uses GANs to predict the saliency maps. It uses a

perceptual loss to learn the semantic information about the salient objects. Our implementation of the SalGAN model [5] is comparable to that of the author's implementation both qualitatively and quantitatively. We also observed that using perceptual loss aids in better prediction of saliency maps compared to the saliency maps predicted by the SalGAN model trained using content loss alone.

The saliency maps generated using the SalGAN architecture have blurred boundaries and using them to segment the salient objects may either add a non-salient part to the segmented object or may ignore some part of the salient object. Such segmentation may affect the accuracy in case of medical imaging scenario. In order to eliminate the blurriness of the boundary and retain the advantages provided by the SalGAN architecture, we designed an improvised SalGAN to predict saliency map with clear boundaries. Our improvised SalGAN model considers both low-level features and high-level feature as equally important. Our improvised SalGAN architecture performed better than 12 of the state-of-the-art models when compared using MAE and F-measure metrics.

6.2 Future Work

The directions for the future work are enlisted as follows:

- We would like to fine tune our implementation of the models to perform at par with the author's implementations.
- In the field of saliency prediction, extensive literature works are available about models that deal with single input image. Saliency detection for multiple input images, such as salient object prediction using video input are less explored. One possible reason for this is minimum availability of benchmark datasets for the above mentioned problem.
- The HVS system is capable of predicting salient objects at instance level; whereas, most of the current saliency prediction models available

in literature tend to group the salient objects together if they are relatively close. Predicting instance level saliency map will allow us to perform better image segmentation, video compression, etc. Instance level saliency prediction is a rich area to be explored.

- The CNN based models are ineffective in handling all kind of affine transformations such as rotation. The saliency map produced using the CNN models may not be as expected when the input image is rotated by some angle. Hinton et al. [66] came up with CapsNet which is capable of handling all type of affine transformations. No literature work is available in this area. CapsNet for saliency prediction is another option to be explored in the future.

Appendix A

Saliency Prediction Metrics

"Idleness is an appendix to nobility" – Robert Burton

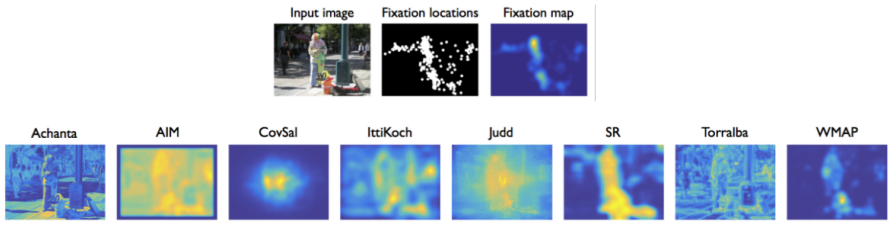
A.1 Introduction

One of the open research problems in the area of saliency prediction is to evaluate the ability of a saliency model to accurately predict the human's eye fixation points in an image. The performance of saliency metrics depends upon many factors. Some of these factors are:

- The context-dependent definition of saliency, e.g. as a classification problem or a regression problem.
- The representation of the ground truth, i.e. ground truth is either represented as a saliency map or a feature map.
- The pre-processing of saliency maps.
- The type of metric in consideration, i.e. similarity or dissimilarity metric.

The extensive research in the area of saliency prediction has designed a plethora of saliency models for predicting the salient regions in an image. The choice of selecting the "best" saliency model objectively is a challenging

task. The performance of a saliency model depends largely on the interpretability due to the irregularity in which different saliency metrics rank the saliency detection models. Figure A.1 illustrates the irregularity in ranking the saliency models by the metrics.



	Achanta	AIM	CovSal	IttiKoch	Judd	SR	Torralba	WMAP
SIM ↑	0.29	0.4	0.49	0.2	0.42	0.41	0.39	0.42
CC ↑	0.04	0.31	0.46	0.14	0.47	0.3	0.25	0.34
NSS ↑	0.13	0.79	1.17	0.43	1.18	0.8	0.68	0.97
AUC ↑	0.52	0.77	0.79	0.59	0.81	0.71	0.68	0.74
sAUC ↑	0.52	0.66	0.53	0.53	0.6	0.64	0.63	0.63
IG ↑	-2.02	-0.03	0.06	-16.71	0.19	-0.39	-0.02	-1.32
KL ↓	2.66	1.30	1.25	13.09	1.12	1.56	1.30	2.33
EMD↓	5.77	4.73	3.35	5.17	4.45	4.63	4.99	4.49

Figure A.1: **Evaluation of saliency models using different metrics**[1] ***First row :*** *Input image and representation of ground truth - discrete fixation location and continuous fixation map.* ***Second row:*** *For a given image, saliency maps predicted by 8 different models.* ***Third row:*** *A table representing the saliency score awarded by 8 different saliency metrics. The yellow, green and blue color represents the top 3 models respectively.*

A.2 Saliency Prediction Metrics

In this section, we discuss different saliency metrics. The saliency metrics are broadly categorized as "location-based" or "distribution-based" metrics. This categorization depends upon whether the ground truth used for evaluation is discrete fixation locations or continuous fixation maps.

[1]Image Courtesy: [67]

A.2.1 Location-based saliency metrics

The location-based saliency metrics are used when the ground truth is represented using discrete fixation points. The following metrics rate the saliency prediction model based on its ability to predict saliency map which approximate the discrete fixation maps.

Area under ROC curve (AUC)

One of the most widely used saliency metrics is AUC. Given a threshold, the Receiver Operating Characteristic (ROC) curve determines the difference between true positives (TP) and false positives (FP). The AUC measure treats the prediction of a saliency map as a binary classification problem, where a pixel is either salient or non-salient. AUC metric determines the performance of a model on a two alternative forced choice (2AFC) task. Given 2 locations on an image, the model must choose the location which is salient. However, the calculation of TP and TN differ with respect to variants of AUC metrics.

- The AUC-Judd metric [68] calculates TP and FP as: for a given threshold, TP-rate is the ratio of TP to the total number of fixation points and FP-rate is the ratio of FP to the total number of saliency map pixels above the threshold. The AUC-Judd metric defines TP as saliency map values above the threshold at the fixation location. The FP is defined as saliency map values above the threshold at the non-fixation location.

- The AUC-Borji metric [69] uniformly samples image pixels at random and assign them as negatives. The saliency map values at these locations are considered as false positives if the saliency map values are above the threshold value.

- The Shuffled AUC (sAUC)metric penalizes the saliency prediction models that consider the center prior. In contrast to AUC-Borji's method

of sampling negatives uniformly at random, sAUC tries to mitigate the center prior by sampling negatives from fixation points of other images in the dataset. The model capable of predicting only the center of the image as salient gets 0.5 as its sAUC score.

Information Gain (IG)

Kümmerer et al. [45] designed the IG metric which ranks the saliency prediction model irrespective of their assumption of center bias. IG evaluates the theoretic information of the proposed saliency model over a baseline model such as center prior baseline model, etc.

Let us consider, the predicted saliency map 'S', the baseline map 'B' and the binary fixation map 'F^B'. The IG is computed as:

$$IG(S, F^B) = \frac{1}{N} \sum_{k=1}^{N} F_k^B [log_2(\epsilon + S_k) - log_2(\epsilon + B_k)]$$

where, k denotes the kth pixel, N denotes the total number of fixation points in the ground truth and $F^B = 1$.

The IG metric assumes that the saliency maps are probabilistic and regularized and optimized to include a center prior. If the IG value of the model is greater than zero, then the saliency prediction model is said to perform better than the baseline model.

Normalized Scanpath Saliency (NSS)

The NSS measure [70] of a saliency prediction model is calculated as the average saliency value of the fixated points in the normalized saliency maps. It is defined as:

$$NSS(S, F^B) = \frac{1}{N} \sum_{k=1}^{N} \tilde{S}_k \times F_k^B$$

where, k denotes the kth pixel, N represents the total number of fixation points, S represents the predicted saliency map, F^B represents the binary

fixation map, $\mu(S)$ represents the mean value of S, $\sigma(S)$ represents the standard deviation of S and

$$\tilde{S} = \frac{S - \mu(S)}{\sigma(S)} \quad and \quad F_k^B = 1$$

A zero or positive score of NSS indicates the similarity between the ground truth and the predicted saliency map; whereas, a negative NSS score denotes dissimilarity.

A.2.2 Distribution-based saliency metrics

The distribution-based saliency metrics are used when the ground truth is represented using continuous fixation points. The following metrics rate the saliency prediction model based on its ability to predict saliency map which approximate the continuous fixation maps.

Similarity (SIM)

The SIM metric measures the similarity between the distributions of the predicted saliency map and the ground truth. The SIM score is calculated as pixelwise summation of minimum values of the normalised input maps. The SIM metric is defined as :

$$SIM(S, F^C) = \sum_{k=1}^{N} min(S_k, F_k^C)$$

where, k denotes the kth pixel, N represents the total number of pixel in the image, S represents the predicted saliency map, F^C represents the continuous fixation map and

$$\sum_{k=1}^{N} S_k = 1 \quad and \quad \sum_{k=1}^{N} F_k^C = 1$$

SIM score of one indicates the predicted saliency map is exactly same to ground truth and a score of zero represent otherwise. The side effect of using SIM is that it penalizes false negatives (FN) more than false positives (FP).

Pearsons Correlation Coefficient (CC)

The CC metric is also known as the "linear correlation coefficient" metric is a statistical based measure which computes the inter-dependence of one variable with respect to the other. The CC considers the predicted saliency map and the ground truth as random variables and evaluates the correlation among the two. The CC metric is defined as:

$$CC(S, F^C) = \frac{\sigma(S, F^C)}{\sigma(S) \times \sigma(F^C)}$$

where, S represents the predicted saliency map, F^C represents the continuous fixation map, $\sigma(S, F^C)$ denote the covariance between S and F^C. The CC metric penalizes both FP and FN equally as it is symmetric. A larger score of CC indicates a similar magnitude of the predicted saliency map and the ground truth.

Kullback-Leibler divergence (KL)

The KL metric measures the difference in the probability distribution of the predicted saliency map and the ground truth. The KL metric is defined as:

$$KL(S, F^C) = \sum_{k=1}^{N} F_k^C log\left(\epsilon + \frac{F_k^C}{\epsilon + S_k}\right)$$

where, k denotes the kth pixel, N represents the total number of pixel in the image, S represents the predicted saliency map, F^C represents the continuous fixation map. and ϵ represent the regularization constant [2]. The KL metric is an asymmetric dissimilarity metric. The predicted saliency map is a better approximation of the ground truth if it has a small KL score.

Earth Movers Distance (EMD)

The EMD metric incorporates the spatial distance of pixels in the predicted saliency map and the ground truth into consideration by measuring

[2]The MIT saliency Benchmark assumes $\epsilon = 2.2204 \times 10^{-16}$

the spatial distance between the two distribution. A large EMD score indicates a large difference between the predicted saliency map and the ground truth and a score of zero indicates the similarity of the two distribution.

F-Measure

The F-Measure is defined as the harmonic mean of average precision and average recall. It is calculated as:

$$F_\beta = \frac{(1+\beta^2)Precision \times Recall}{\beta^2 Precision + Recall} \qquad where, \quad \beta^2 = 0.3$$

The Precision and Recall are defined as:

$$Precision = \frac{|M \cap G|}{|M|} \quad and \quad Recall = \frac{|M \cap G|}{|G|}$$

where, M is the binary map of the predicted saliency map S and G represent the ground truth. One of the simplest way of converting S to M is do divide each pixel of S by 255.

Mean Absolute Error (MAE)

The MAE metric is used to compare the predicted saliency map S and the binary map of the ground truth M. It normalizes both the inputs in the range $[0, 1]$. The MAE value is calculated as:

$$MAE = \frac{1}{N}\sum_{k=1}^{N} |S_k - M_k|$$

A.3 Summary

We summarize the saliency metrics in Table A.1.
Different metrics use different formats of ground truth for evaluating saliency models. The location-based metrics consider saliency map values at discrete fixation locations, while the distribution-based metrics treat both ground truth fixation maps and saliency maps as continuous distributions. Good saliency models should have high values for similarity metrics and low values for dissimilarity metrics.

Metrics	***Location-based***	***Distribution-based***
Similarity	AUC, NSS, IG	SIM, CC, F-Measure
Dissimilarity		EMD, KL, MAE

Table A.1: **Categorization of different saliency metrics**

Bibliography

[1] Bernard Marr, "How much data do we create every day? the mind-blowing stats everyone should read," May,2018.

[2] T Huang, "Computer vision: Evolution and promise," 1996.

[3] Laurent Itti, Christof Koch, and Ernst Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[4] Junting Pan, Elisa Sayrol, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E O'Connor, "Shallow and deep convolutional networks for saliency prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 598–606.

[5] Junting Pan, Cristian Canton Ferrer, Kevin McGuinness, Noel E O'Connor, Jordi Torres, Elisa Sayrol, and Xavier Giro-i Nieto, "Salgan: Visual saliency prediction with generative adversarial networks," *arXiv preprint arXiv:1701.01081*, 2017.

[6] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao, "Salicon: Saliency in context," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[7] Guo-Xin Zhang, Ming-Ming Cheng, Shi-Min Hu, and Ralph R Martin, "A shape-preserving approach to image resizing," in *Computer Graphics Forum*. Wiley Online Library, 2009, vol. 28, pp. 1897–1906.

[8] Hua Huang, Lei Zhang, and Hong-Chao Zhang, "Arcimboldo-like collage using internet images," in *ACM transactions on graphics (TOG)*. ACM, 2011, vol. 30, p. 155.

[9] Liang Li, Zhipeng Wu, Qingming Huang, Shuqiang Jiang, and Zhengjun Zha, "Partial-duplicate image retrieval via saliency-guided visually matching," *IEEE Multimedia*, vol. 99, no. 1, pp. 1, 2013.

[10] David Meger, Per-Erik Forssén, Kevin Lai, Scott Helmer, Sancho McCann, Tristram Southey, Matthew Baumann, James J Little, and David G Lowe, "Curious george: An attentive semantic robot," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 503–511, 2008.

[11] John K Tsotsos, Scan M Culhane, Winky Yan Kei Wai, Yuzhong Lai, Neal Davis, and Fernando Nuflo, "Modeling visual attention via selective tuning," *Artificial intelligence*, vol. 78, no. 1-2, pp. 507–545, 1995.

[12] Laurent Itti, Christof Koch, and Ernst Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, , no. 11, pp. 1254–1259, 1998.

[13] Radhakrishna Achanta, Francisco Estrada, Patricia Wils, and Sabine Süsstrunk, "Salient region detection and segmentation," in *International conference on computer vision systems*. Springer, 2008, pp. 66–75.

[14] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum, "Learning to detect a salient object," *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 33, no. 2, pp. 353–367, 2011.

[15] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[16] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal, "Context-aware saliency detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 10, pp. 1915–1926, 2012.

[17] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2015.

[18] Chuan Yang, Lihe Zhang, and Huchuan Lu, "Graph-regularized saliency detection with convex-hull-based center prior," *IEEE Signal Processing Letters*, vol. 20, no. 7, pp. 637–640, 2013.

[19] Shengfeng He, Rynson WH Lau, Wenxi Liu, Zhe Huang, and Qingxiong Yang, "Supercnn: A superpixelwise convolutional neural network for salient object detection," *International journal of computer vision*, vol. 115, no. 3, pp. 330–344, 2015.

[20] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang, "Deep networks for saliency detection via local estimation and global search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3183–3192.

[21] Gayoung Lee, Yu-Wing Tai, and Junmo Kim, "Deep saliency with encoded low level distance map and high level features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 660–668.

[22] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[23] Abien Fred Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[24] Jongpil Kim and Vladimir Pavlovic, "A shape-based approach for salient object detection using deep learning," in *European Conference on Computer Vision*. Springer, 2016, pp. 455–470.

[25] Hongyang Li, Jiang Chen, Huchuan Lu, and Zhizhen Chi, "Cnn for saliency detection with low-level feature integration," *Neurocomputing*, vol. 226, pp. 212–220, 2017.

[26] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[28] Nian Liu and Junwei Han, "Dhsnet: Deep hierarchical saliency network for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 678–686.

[29] Matthias Kümmerer, Lucas Theis, and Matthias Bethge, "Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet," *arXiv preprint arXiv:1411.1045*, 2014.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[31] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao, "Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 262–270.

[32] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip HS Torr, "Deeply supervised salient object detection with short connections," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3203–3212.

[33] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[35] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio, "Maxout networks," *arXiv preprint arXiv:1302.4389*, 2013.

[36] Sebastian Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[37] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[38] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[39] Matthew E Taylor and Peter Stone, "Transfer learning for reinforcement learning domains: A survey," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1633–1685, 2009.

[40] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[42] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan, "Pytorch," 2017.

[43] Zijun Deng, Xiaowei Hu, Lei Zhu, Xuemiao Xu, Jing Qin, Guoqiang Han, and Pheng-Ann Heng, "R3net: Recurrent residual refinement network for saliency detection," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 2018, pp. 684–690.

[44] Shizhong Dong, S Shanhui, W Xin, L Ming, Z Heye, Y Guang, L Huafeng, and L Shuo, "Holistic and deep feature pyramids for saliency detection," BMVC, 2018.

[45] Matthias Kümmerer, Thomas Wallis, and Matthias Bethge, "How close are we to understanding image-based saliency?," *arXiv preprint arXiv:1409.7686*, 2014.

[46] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[47] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[48] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 2009, pp. 248–255.

[49] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*. Springer, 2016, pp. 694–711.

[50] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, "Aggregated residual transformations for deep neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 5987–5995.

[51] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2015.

[52] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia, "Hierarchical saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1155–1162.

[53] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang, "Saliency detection via graph-based manifold ranking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3166–3173.

[54] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang, "Saliency detection via graph-based manifold ranking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3166–3173.

[55] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun, "Saliency optimization from robust background detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2814–2821.

[56] Wei Shen and Rujie Liu, "Learning residual images for face attribute manipulation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4030–4038.

[57] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang, "Saliency detection by multi-context deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1265–1274.

[58] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang, "Deep networks for saliency detection via local estimation and global search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3183–3192.

[59] Guanbin Li and Yizhou Yu, "Visual saliency based on multiscale deep features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5455–5463.

[60] Gayoung Lee, Yu-Wing Tai, and Junmo Kim, "Deep saliency with encoded low level distance map and high level features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 660–668.

[61] Xi Li, Liming Zhao, Lina Wei, Ming-Hsuan Yang, Fei Wu, Yueting Zhuang, Haibin Ling, and Jingdong Wang, "Deepsaliency: Multi-task deep neural network model for salient object detection," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3919–3930, 2016.

[62] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan, "Saliency detection with recurrent fully convolutional networks," in *European conference on computer vision*. Springer, 2016, pp. 825–841.

[63] Guanbin Li and Yizhou Yu, "Deep contrast learning for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 478–487.

[64] Nian Liu and Junwei Han, "Dhsnet: Deep hierarchical saliency network for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 678–686.

[65] Zhiming Luo, Akshaya Mishra, Andrew Achkar, Justin Eichel, Shaozi Li, and Pierre-Marc Jodoin, "Non-local deep features for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6609–6617.

[66] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton, "Dynamic routing between capsules," in *Advances in neural information processing systems*, 2017, pp. 3856–3866.

[67] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand, "What do different evaluation metrics tell us about saliency models?," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[68] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba, "Learning to predict where humans look," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 2106–2113.

[69] Ali Borji, Dicky N Sihite, and Laurent Itti, "Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 55–69, 2013.

[70] Jordi Pont-Tuset and Ferran Marques, "Supervised evaluation of image segmentation and object proposal techniques," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 7, pp. 1465–1478, 2016.