CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

# Salient Object Detection with Deep Learning

by

Wajeeha Sultan

A thesis submitted in partial fulfillment for the degree of Master of Science

in the
Faculty of Computing
Department of Computer Science

2020

*I dedicate my dissertation work to my family and my teachers. A special of gratitude to my loving father, mother and husband for their love, endless support and encouragement.*

# CERTIFICATE OF APPROVAL

## Salient Object Detection with Deep Learning

by

Wajeeha Sultan

(MCS181053)

### THESIS EXAMINING COMMITTEE

| S. No. | Examiner | Name | Organization |
|---|---|---|---|
| (a) | External Examiner | Dr. Muhammad Zubair | Riphah International |
| (b) | Internal Examiner | Dr. Basit Siddique | CUST, Islamabad |
| (c) | Supervisor | Dr. Nadeem Anjum | CUST, Islamabad |

Dr. Nadeem Anjum
Thesis Supervisor
October, 2020

Dr. Nayyer Masood
Head
Dept. of Computer Science
October, 2020

Dr. Muhammad Abdul Qadir
Dean
Faculty of Computing
October, 2020

# *Author's Declaration*

I, **Wajeeha Sultan** hereby state that my MS thesis titled "**Salient Object Detection with Deep Learning**" is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my MS Degree.

**(Wajeeha Sultan)**
Registration No: MCS181053

# *Plagiarism Undertaking*

I solemnly declare that research work presented in this thesis titled “**Salient Object Detection with Deep Learning**” is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been dully acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS Degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

**(Wajeeha Sultan)**
Registration No: MCS181053

# *Acknowledgements*

And your god is one God. There is no deity [worthy of worship] except Him, the Entirely Merciful, the Especially Merciful [2:163]. First and foremost, I wish to say thanks to Allah (S.W.T) for giving me blessings, power and knowledge to finish this research. Secondly, I wish to express my gratitude to my supervisor **Dr. Nadeem Anjum** for his help, precious time and supervision. I pay my thanks to him sincerely for his assistance, motivation and advice in this field of research. He helped me from the understanding of this subject till the write up of final thesis. I am deeply indebted to my family and my parents for their support and encouragement till the end of my MS thesis. Their prayers and guidance have lead me here. A special thanks to my husband for his support and encouragement in the completion of my research work. I pray to Allah that may he bestows me with true success in all fields in both worlds and shower his blessed knowledge upon me for betterment of all muslims and whole mankind.

**(Wajeeha Sultan)**
Registration No: MCS181053

# *Abstract*

Everyday life has enormous amount of visual data and information which is accessible and generated every minute. The rise in image data has created new problems of extracting the correct information and fast processing to ease different task from searching in images to image compression and spreading of images over network. In recent years it has been one particular problem of computer vision algorithms to find object of interest in images as its importance lies in many areas of medicine, robotics, graphics and computer vision. Taking advantage of rich features extracted by deep learning technology, recently salient object detection has seen tremendous progress but it is very difficult for deep model to segment accurate objects in low contrast images due to excessive noise and most of the models have not focused on boundary the quality of boundary and the output objects are blurred near boundaries. To address these problems, Salient object detection with deep learning (SODL) is proposed. Our model extracts features at local and global level and then integrates them which gets the information from pixel level to exactly locate the object. GCB (Global Convolutional Block) and BRB (Boundary Refinement Block) modules are also embedded in model to preserve the spatial information. For the further refinement of map refinement module is added which refines the overall saliency. Experimental analysis have shown that proposed model outshines state of the art methods.

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **AUC** | Area Under Curve |
| **BRB** | Boundary Refinement Block |
| **CNN** | Convolutional Neural Network |
| **FCN** | Fully Convolutional Network |
| **GCB** | Global Convolutional Block |
| **GT** | Ground Truth |
| **MAE** | Mean Absolute Error |
| **MLP** | Multilayer Perceptron |
| **SOD** | Salient Object Detection or Segmentation |
| **WFM** | Weighted F Measure |

# Chapter 1

# Introduction

Human beings can easily and rapidly discern unique, prominent regions. By processing these regions higher level or rich information is extracted. This capability has been intensively analysed in different fields along with computer vision because it aids in finding an object or area of interest.

The method of highlighting and segmentation of objects from image or video is referred to as Salient Object Detection. It mainly comprises of two stages:

- Detection of salient object
- Accurate segmentation of that object

Generally, it is accepted that a model should fulfill following subsequent criteria for successful saliency detection [1].

- Less number of false positives and false negatives.
- High resolution saliency maps.
- Model should be efficient in terms of computational cost

Extensive work has been done on detection of sailent objects and many arcietectures have been proposed, In next section bakground of salient object detection is presented.

## 1.1 Background

Existing models for SOD can be generally classified into two groups [1]:

- Traditional methods that use handcrafted features (low level features i.e. color, contrast).
- Deep learning based techniques.

### 1.1.1 Traditional Methods

Traditional methods use handcrafted features or heuristics for salient object detection. These methods must first describe a set of features. After that these features are used for object detection or classification [1].
For SOD two different kinds of visual subsets have been used: regions and blocks. Blocks were used in early approaches whereas regions became famous when super pixel algorithm were introduced [1]. In Some block bases techniques, pixel wise contrast was used to detect saliency , after that salient regions were detected by calculating contrast features, edge detection and other geometric properties [21][22]. Some techniques emphasize on patch uniqueness by calculating distance from average patch but these methods had some drawbacks:

- high contrast regions were captured instead of salient region.
- loss of saliency information near boundaries.

To solve these problems region base saliency detection techniques were proposed, these techniques uses features extracted from regions to calculate saliency map.
In region based models saliency score of region was calculated by average score of pixels in a region, after that many others parameters such as texture, color uniqueness, background score of region and structure were also considered in saliency detection [22]. Further graph base segmentation for regions and clustering techniques were also adopted for better saliency computation. These techniques provide some

benefits first, efficient and false algorithms can be developed because number of regions is much less than blocks. Second more revealing features can be extracted leading to better results.

With the accessibility of vast variety of visual content on the internet, models adopted to detect salient objects with similar images by using set of training images which contains similar salient objects [23][24]. These methods work well if large variety of images are available. After this co-saliency detection grab the attention which focuses on detecting similar salient objects present in various input images.

Many existing works uses supervised models for saliency detection, these methods learn by the help of training images with ground truth annotations to differentiate salient object from background. By taking advantage of large set of training images, classifier learns to pick most distinct features therefore these methods provide better performance as compare to heuristic methods [1]. All traditional methods mainly use heuristics for salient object detection which limits their potential to detect salient objects in complex scenarios.

### 1.1.2 Deep Learning Based Methods

Recently Deep Learning has attained remarkable success in salient object detection because it provides rich and discriminative representation of images. CNNs is one of very effective tools in machine learning and it has been shown that they work efficiently in salient object detection because of their ability to extract both high and low level features [1].

Early deep saliency models utilize multi-layer perceptron (MLPs) for detection. In these methods segmentation is performed on input image and is segmented into small regions, then a Convolutional neural network (CNN) is used to extract features which are then passed to MLP to compute saliency of region [25]. But these models can not completely extract high level semantic information and this information cannot be passed to fully connected layers which results in information loss. Due to these shortcomings FCN (Fully Convolutional networks) have been

adopted, these models examine pixel level operations to reduce the problems.
In recent FCN base techniques encoder-decoder architecture, recurrent networks have been used which enables end-to-end training strategies [26]. In spite of the fact that FCN has produce great results, these models still lack for images which have low contrast, images with complex background and fine prediction near boundaries because:

- saliency is heavily affected by the intense noise of low contrast images which results is poor detection of low contrast images.
- and because of frequent pooling operations in deep learning methods loss of object structure and semantic information is unavoidable which causes poor detection of objects particularly in low contrast images.
- Since the saliency is determined by the global contrast of image instead of local features so it becomes hard for model to examine detailed boundary knowledge of object.

To overcome these problems, we propose a boundary aware fully convolutional network for detection of salient objects which captures both local and global context with a boundary refinement module to achieve segmentation with fine boundaries. Global features help in object detection and classification on the other hand local features are used for object identification and recognition. Global features contain texture features, shape descriptors and color representations, where as local feature points to well define, sharp feature or pattern present in an image such as corner, edges etc. Local features are typically related to an image patch that varies from its surrounding by intensity, color or texture. Integration of both local and global features enhance the accuracy of object recognition.

## 1.2 Contributions of Research

Contribution of this research can be summed up as follows:

1. A fully convolutional network to capture global as well as local contexts to learn detailed structure of objects.
2. A residual refinement module is embedded which process the predicted saliency map to refine boundaries by learning residual among ground truth and coarse saliency map.

## 1.3 Problem Statement

Extensive research has been done on detection of salient object and produced impressive results as compare to traditional approaches but it is still an issue in case of low contrast images and fine segmentation near boundaries so, there is need to develop a method that should refine segmentation of salient objects near boundaries and to capture global and local contexts to learn detailed structure of objects which will help in locating object in low contrast images.

## 1.4 Research Questions

The problems mentioned in problem statement lead us to following research questions:

1. How objects can be segmented in low contrast images?
2. Which features can play important role in identification of salient object in low contrast images?
3. Can deep learning produce better results than traditional methods?
4. Can we combine different deep learning models for successful saliency detection?

## 1.5 Purpose

This research aims to detect and segment salient object with high boundary precision and to enhance the localization and detection of salient areas in low contrast images. This work helps in developing an effective and efficient deep learning method for salient object detection.
We propose a method that uses global and local features and a boundary refinement module for accurate segmentation of objects.

## 1.6 Significance of the Solution

Everyday life has enormous amount of visual data and information which is accessible and generated every minute. The rise in image data has created new problems of extracting the correct information and fast processing to ease different task from searching in images to image compression and spreading of images over network.
In recent years it has been one particular problem of computer vision algorithms to find object of interest in images as its importance lies in many areas of medicine, robotics, graphics and computer vision. Therefore, there is a need to develop a technique that is capable of fine detection of salient objects. The propose model emphasis on local and global features with boundary refinement. The combination of these features helps in accurate detection and segmentation of salient objects.

## 1.7 Tool and Techniques

Tools and technologies used in this research are mentioned below:

1. Operating System for Testing: Windows 10, 64 bit
2. Operating System for Training: Linux: Ubuntu 16.04, .6LTS

3. GPU for Training: Nvidia GeForce GTX 1080 (8119 MiB)

4. Feature Extraction Network: Pre-trained VGG16

5. PyCharm Tool: For testing model

6. Interpreter: Python 3.7

## 1.8 Applications of SOD

SOD can be utilized in several dimensions of Computer vision, Graphics and robotics. Some applications include captioning of images in which objects are detected and model learns to describe content of images [1] [2], detection of targets from images [3], Classification of the scene, which consists of assigning a label based on the overall content of the picture [4], detection of salient object in vedios [6], semantic segmentation through object region mining [9], semantic segmentation by utilization of image level annotations [10], re identification of human from mutipe photographs of same person taken from same or different camera [11], recognition of objects by using robots [13], video abstraction [17] and image quality evaluation [19].

# Chapter 2

# Literature Review

In general, methods for detection of salient object can broadly be classified into two groups: traditional approaches which uses low level features (e.g. Color, Contrast) to differentiate salient objects and deep learning base approaches which eradicates the use of hand crafted features. These approaches are able to learn high level features by training set. This chapter provides detailed overview of the research work done for detection of salient objects.

## 2.1 Traditional Approaches

In traditional approaches there are two types of visual subsets: regions and blocks, blocks were used in early approaches and regions become more favorable with the initiation of super pixel techniques.

Achanta et al., utilizes features of color and luminance for detection and to preserve boundaries they have used more frequency content from image. For calculation of saliency map difference between image pixel vector and its corresponding mean image feature vector is computed, fixed and adaptable thresholding is used for segmentation. They focused only on detecting large salient objects [27].

Zhang et al., proposed a technique which uses color and texture features to extract image information. Saliency of boundary areas is calculated in first step and they eliminate the ones that have high saliency value as they can be a part of salient object. In second step they use clustering and graph base manifold ranking schemes for object detection. Background base detection targets to acquire maximum salient regions. In foreground detection similar method is used with slight difference. Saliency is computed by integrating patches in each clusters. To compute saliency map, they have combined texture and color cues saliency maps [28].

Rahtu et al., presents a cost efficient method for detection of salient areas. This method is set up on intensity based search of image segments, whose intensity rates are more precisely drawn by intensity distribution as compare to intensity of surrounding area. This approach requires no extra segmentation algorithms and no training. Saliency map detects salient object automatically and adaptive thresholding is used for segmentation. The segmentation is utilized to streamline and divide the image into sections with compatible information to analyze [29].

Sokalski et al., proposed a technique that is built on present salient object detection models. Their technique utilizes an image contrast map resulting from combination of pioneering work in this field, multi-channel edge knowledge and mean scale partition with histogram reinforcement. This is utilized to construct a saliency map from a given image in the existence of noise influencing image quality [30].

Lee et ., has developed a method that is based on regional contrast to detect salient objects which analyze weighted spatial coherence score and global contrast differences which results in production of fine quality saliency maps. The produced maps are also used to initiate repeated version of GrabCut for high resolution segmentation. Under certain scenarios this method achieves good results but they need a thorough search before locating object which consumes time [31].

Hou et al., introduced an approach called image signature for figure ground partition problem. They have demonstrated that a simple descriptor contains details about image foreground. They have proved that a saliency map produced from image signature can be better than many saliency algorithms. They have also presented a reaction time data gathered from subjects' in change blindness experiment. They have shown that distance in images created by image signature matches with perceptual distance [32] .

Li et al., presented a bottom up approach for detection of salient object. They have shown that image amplitude spectrum convolution with an appropriate scale low pass Gaussian kernel is similar to an image saliency detector. The saliency map is produced by rebuilding of 2g signal using initial phase and amplitude spectrum, refined at a selected scale by minimizing entropy of saliency map. The pixels are graded according to color coexistence o inter frame pixel changes. Maintained background image is subtracted from original image for detection of object. Their approach is precise even in dynamic backgrounds [53].

Yang et al., proposed a method by using both background and foreground cues. To rank the likeness between image pixel and regions they have used graph baaed manifold ranking, image saliency is defined on the basis of their significance with given queries or seeds. Image is depicted as a graph along with super pixels as nodes. Ranking of nodes is built on background and foreground queries similarity, set up on affinity matrices. In a two stage process, saliency detection is performed to draw out salient objects effectively [33].

Imamoglu et al., introduced a model by making use of low level features gathered from wavelet transform domain. Wavelet transform is used to construct multi scale feature maps that can reflect features from texture to edge, then a model is proposed to construct saliency ma from these features. This model is directed to regulate local contrast with its global saliency which is calculated from probability of features, and this model examine global contrast and local center surround differences in produced saliency map [34].

Zou et al., presented a methodology to separate salient objects from background. unlike former unidirectional saliency detection methods, in which detected saliency map is utilized to direct the segmentation this approach manually makes use of detection or segmentation cues. Segmentation guided low rank matrix model is used to produce saliency map. This saliency map is utilized to initiate segmentation model. Collectively segmentation is used to improve quality of saliency map to produce better results of segmentation [35].

Tong et al., proposed a SOD algorithm which is based in bootstrap learning which exploits both weak and strong methods. A weak saliency map is constructed from image priors to produce training sample for strong classifier, after that a strong classifier is used to detect salient pixels directly from input image. Results from both methods are combined to improve detection [36].

## 2.2 Deep Learning Based Techniques

In deep learning, a model learns directly from image to perform classification. Deep learning model can attain state-of-the-art results. Training of model is performed by using wide collection of labeled data and architecture of several layers that include several layers.

CNN and FCNs are most popular types of deep models. It removes the necessity for manual feature extraction. CNNs collect features from images, the related features aren't pre trained they are learned throughout the training of model. Review of deep learning based models is presented in this section.

Lee et al., proposed a technique that uses both high and low level features for saliency detection in a single deep learning model. VGG-Net is used to collect high level features, then different parts of image is compared with low level features to produce low level distance map. This map is encoded with convolutional and RELU layers. To compute saliency of region encoded low level distance maps and high level features are concatenate and passed to fully connected classifier. They have used super pixel methodology for saliency methodology. For segmentation of image in super pixels SLIC technique is used, after segmentation low level features are calculated. Their hand crafted feature includes texture, color etc. for encoding of distance map 1x1 convolution is used. MSRA10K, PASCAL-S, ECSSD, DUT-OMRON and THUR15K datasets were used for evaluation. This method does not perform well when the color difference is less in background and foreground and for low contras images [37].

Wang et al., developed a technique that make use of prior saliency detection techniques. Repetitive architecture allows this method to refine the saliency map by rectifying previous errors which results in fine predictions. They have introduced pre training strategy using segmentation data for training of network, which helps in segmentation or successful training and allows the network to grasp objects for saliency detection. They train a FCN network to estimate nonlinear mapping of saliency values from raw pixels and neglects the saliency priors, then these saliency priors are combined with deeply learned features which is passed for the iterative refinement, in this module the whole network is propagated forward which causes an increase in computational cost and memory usage. SED1, ECSSD, PASCAL-S and HKU-IS datasets were used for experiments. This approach does not work well in difficult situations such as when the color contrast between the object and the background is not great and when the object is very big and has multiple colors. Another limitation is the computational cost caused by iterative refinement [38].

Vivek et al., in which they have used many saliency detection methods to increase quality of saliency map. First they generate initial saliency maps by selecting some saliency models then they integrate these maps to form a binary map after that final saliency map is generated using an integration logic. The combined binary map defines pixel in finer way than single binary map. By using these labels final saliency map is produced by using logic in which maximum and minimum saliency values are assigned to pixels. The efficiency of this method rely on saliency detection model being selected. Therefore, the method of selection of existing technique plays a key role. MSRA10K dataset, ECSSD, DUT-OMRON data set, PASCAL-S, THUR15K and SED2 datasets were used for experimentation. This method performs well when selected methods can detect images and fails in case those methods cannot detect and in some cases it marks saliency region as a background region [39].

Liu et al., proposed a strategy in which side outputs of deeper level are used to guide feature maps of shallower level, by this way deeper level outputs can be transmitted to high resolution version. Another convolutional block is proposed which divide the feature maps into groups which are passed to convolutional blocks to produce discriminative features. VGG16 is used as a backbone from where 5 feature maps are produced after that convolutional module is applied to produce fine feature maps, for integration of multilevel context information guidance technique is used. For computation of initial saliency map feature maps produced from convolutional block and VGG are used. Convolutional module uses 3x3 kernel across 4 channels. ECSSD, DUT-OMRON and HKU-IS datasets were used for experiments. Although this method has made progress but it fails in the case of images that contains complex semantic knowledge [40].

Feng et al., introduced a model to enhance segmentation results near boundaries. Attentive feedback module is proposed to produce fine boundaries. Features from encoder blocks are passed to decoders where feedback module is applied for segmentation, this module learns the structure and then perform segmentation. Further they have proposed a boundary enhance loss to assist feedback module. VGG 16 is used as a base model by modifying it into an encoder network which is followed by perception module which uses fully connected layer which leads to decoder module. Every decoder layer has 3x3 convolutional layer. Multi scale features are collected via the encoder block then global saliency prediction is calculated which is passed to decoder for finer saliency predictions, for boundary correction attentive feedback is applied. ECSSD, PASCAL-S, HKU-IS, DUT-OMRON and DUTS datasets were used for evaluation [41].

Wang et al., proposed a deep model which uses boundary knowledge to accurately locate salient object. In boundary guided network two sub networks are defined one is for mask and other is for boundaries, features are shared between these two networks. They have also proposed a focal loss to learn the loss of hard pixel near boundaries. Both sub networks follow encoder decoder network, features are extracted in encoder and decoder gives the output. Decoders of both subnetworks are connected. The features extracted by encoder have different resolutions, bidirectional flow of information is enabled in encoders. Features maps from two subnetworks are combined and passed to one convolutional layer. Decoder of mask and boundary sub-networks are connected. Mask decoder refines the mask prediction and uses focal loss which pays attention toward boundary pixels. MSRA-B, ECSSD, DUT OMRON data set, SOD and HKU-IS datasets were used [42].

Eitel et al., proposed a RGB-D strategy for object detection. The network consists of two different CNNs for each modality paired with late fusion network. They set up a multi stage training approach for effective training and a data augmentation framework for learning with images by manipulating them with noise patterns. The two convolutional models utilize color and depth information to examine RGB and depth data which is integrated in fusion technique. ImageNet is used for pre training of data then multimodal CNN is trained. The parameters are tuned for classification of data followed cy combined training of parameters and fusion network. Images are resized before passing to CNNs. They have proposed an effective method for encoding of depth images which normalizes depth values fall between 0 to 255. Then the color amp is applied on images which transforms it into three channel image. RGB-D dataset is used for evaluation [43].

Zhu et al., presented a hierarchical structural learning technique for salient object detection. An object is defined by a combination of hierarchical tree structures in which nodes reflect object parts. The spatial movement of nodes allow the deformation of local and global shape. SVM learning can be used to train model in which nodes position are latent variables. They have introduced a concave convex technique (iCCCP) which effectively learns two or three layer models. The focus is to utilize deep structure to learn hierarchical model. One root node in first layer stand for object, in second layer child nodes are shown. For particular objects and views number of nodes and layers are same. Every tree model is related to latent variables. In detection the task is to find class label and locations. Bounding box is use for detection, incremental CCCP is utilized for training to minimize cost. PASCAL dataset is used for evaluation of model [44].

Girshick et al., developed a robust and versatile detection technique which enhances the mean average precision over 50 percent compared to prior techniques. This approach incorporates two concepts: one is applying high CNNs to localize objects from bottom up regions and second one is supervised pertaining when abundant data is available followed by fine tuning to increase performance. The whole technique comprises of three modules. First originates regions, these regions possible detections accessible to detector. Second module consist of convolutional layers to collect features from regions. Last module is linear SVMs. They have used selective search to select technique for region proposals. To extract features regions are first converted into form that is compatible to CNN, after features are extracted optimization is applied. Evaluation of method is performed on PASCAL VOC dataset [45].

Ding et al., introduced a former knowledge based approach designed to help the robot to identify indoor objects in vision. They have joined a public indoor dataset to train CNN. After that mean images are generated which are used for color knowledge. At the end when detection initiated, the two vectors together are multiplied with a classification vector to generate a classification decision vector. The distance from input and mean image generates the class weight vector. Number of occurrences of an object in a scene is used as a prior knowledge in scene understanding. After combination of datasets, CNN is trained followed by scene understanding. When detection starts image is redirected to generate vector classification probability, then class weight vector is calculated, after calculating scene weight vector the product of these three vectors generates decision vector. Index of maximum value in decision vector is the output. They have proved from experiments that detection can be facilitated by using color and scene knowledge. The results indicated usefulness of method in robot vision [46].

Zhang et al., developed a block wise scene analysis using deep learning complemented with a spatiotemporal model. By making use of de noise auto encoder this method seeks to generate image features encoding scene information which results in effective feature description. In addition, the model encapsulates scene information from Hamming space, ensuring effectiveness of moving object. The introduced deep learning based feature learning is able to gather structural properties of scene and create filter patterns that are helpful in noise. They have also introduced a hash method to reduce memory usage by binarization of features. Focusing on hash method a binary scene modeling technique is also proposed which gathers spatiotemporal information which helps in construction of binary model. For evaluation of method nine video datasets were used [47].

Zhao et al., represented a review on object detection techniques based on deep learning. Analysis starts with a short introduction of deep learning and CNN. Then they concentrated on typical object detection techniques along with the suggestions to improve performance. They have also thrown light on face detection and pedestrian detection methods. For the comparison of methods and produce conclusion experimental results are also provided. At the end some guidelines are also given for future work [48].

Borji el al., provided a survey on work done in field of salient object detection. At first they have provided introduction of saliency detection and its applications. They have discussed from traditional models to deep learning models along with datasets and evaluation measures used. They have also mentioned problems in recent techniques and future work [49].

Wang et al., provided a review on SOD techniques from different angles such as architecture, learning paradigm and level of supervision. They have also discussed datasets and metrics for evaluation. A thorough analysis of techniques by calculating their results is also given. They have also discussed the strengthens of SOD algorithms in case of attacks. They have also pointed out problems in recent SOD techniques and future guidance [50].

Mu et al., developed a deep learning network with global convolutional module followed by a boundary refinement part for efficient detection in low contrast images. They have used VGG16 for feature extraction. And those features are refined by GCM module which consist of left and right convolutional layers, results from these two layers are combined to produce a dense feature map. For boundary refinement they have also included a module which has two branches. One has two convolutional layers and other branch combines the input and out without any operation to refine boundary pixels. The output dimensions of both blocks are same as input. To enhance the contrast features they have also included a stage which computes the dissimilarity between feature map and its average. They have also proposed a low contrast dataset to evaluate model on low contrast images.

Other data sets include DUT-OMRON, MSRA-B, PASCAL data set, DUTS and HKU-IS [51].

Qin et al., developed a detection algorithm which focuses on boundaries. This architecture consists of encoder decoder network and refinement module. They have also introduced a hybrid loss for learning of model. This loss is the integration of Intersection Over Union, Binary Cross Entropy and structural similarity losses. BASNet comprises of two modules one is for the prediction of saliency map and other one is for the refinement of predicted saliency map. Prediction module is made up of encoder decoder network. Encoder block consist of convolutional layers, a bridge stage is integrated between encoder and decoder to further collect global information, it includes three convolutional layers followed by batch normalization. Decoder follow same structure as encoder and comprises of three convolutional layers. The refinement module also follows same structure but in this each network have four stages. Every stage consists of one convolutional layer. For down sampling max pooling is used and bilinear interpolation for up sampling. For evaluation DUTS, PASCAL-S, ECSSD, DUT-OMRON data set, HKU-IS and SOD dataset were used [52].

## 2.3 Critical Survey of SOD Models

After a thorough analysis of SOD techniques, we present the strength and limitation of related techniques in Table 2.1.

TABLE 2.1: Critical Analysis Table of Literature

| Ref | Methods | Strengths | Limitation |
|---|---|---|---|
| [27] | Frequency tuned model for large salient objects | Efficiency in computation<br>Fine detection in large objects | Difficult to align without proper instruments<br>Not good for images which have low frequency content |

Continued on next page

**Table 2.1 – continued from previous page**

| Ref | Methods | Strengths | Limitation |
|---|---|---|---|
| [30] | Combination of mean shift segmentation, contrast features with histogram enhancement | Avoid the effect of noise on neighbor pixels | In some cases, detect false negative because of noise |
| [31] | Intensity and Contrast Based Segmentation of images | Production of fine saliency map at low computational cost | The production of saliency map is difficult to generate which causes difficulty in processing large videos<br>Locating object is time consuming for large images |
| [32] | Detection based on image signature | This method is scalable and does not alter image quality<br>Flexibility | If file size is large it is hard to locate small elements in it |
| [53] | Background construction based saliency detection | Accuracy of of model is high even with images of dynamic backgrounds | Computational cost is high |
| [33] | Graph based | It combines | High dimensionality |

Continued on next page

**Table 2.1 – continued from previous page**

| Ref | Methods | Strengths | Limitation |
|---|---|---|---|
| | manifold ranking to detect saliency Ranking on the basis of image similarity is performed. It involves both background and foreground objects | local and some other methods to produce effective results | makes it less fir for nearest neighbors |
| [34] | Integration of local and global saliency maps Wavelet transform is used to extract details | Relation between both feature maps provide effectiveness for saliency detection | Global scene disorders can be more prominent then local Loss of spatial information can be occur from high up sampling |
| [35] | Unsupervised technique. For segmentation saliency map is used | Joint and iterative optimizations | Computational cost is high |
| [36] | Bootstrap algorithm for detection of | Reduction of time used in training | It is unsupervised and limited |

Continued on next page

**Table 2.1 – continued from previous page**

| Ref | Methods | Strengths | Limitation |
|---|---|---|---|
| | salient object Initial saliency map is generated using image information and then training is used or strong features | process | within multiple scale of image |
| [37] | Combination of low level and and high level features. Low level features are based on color and texture. Deep learning model is used for high level features | Integration of low and high level features enables fine segmentation by taking advantages from both features | Bad segmentation in case of low contrast images |
| [38] | FCN is trained to produce saliency map. Uses iterative enhancement of saliency map to produce high quality | Iterative module gives great benefit in refinement of saliency resulting in fine segmentation | Costly in terms of computation because of iterative module |

Continued on next page

**Table 2.1 – continued from previous page**

| Ref | Methods | Strengths | Limitation |
|---|---|---|---|
| | map. | | |
| [39] | Combination of different existing saliency detection approaches. Generation of binary maps from different models Integration of these maps to produce final map | Performs well when efficient SOD model is selected for that case | Poor detection in case Selected method does not produce fine binary map |
| [40] | Contextual information based technique to detect objects. Features produced from deep model are complimented with contextual information | Good quality detection in some cases | Not able to detect properly in complex backgrounds |
| [43] | Training is applied in multiple stages | This method is capable of learning of rich features. Training works well in case | This process takes much time. |

Continued on next page

**Table 2.1 – continued from previous page**

| Ref | Methods | Strengths | Limitation |
|---|---|---|---|
| | | of noise | |
| [44] | Hierarchical learning SVM is also used for learning | Lowers cost for computation Deep features outshine other features and are enough for good result. | Does not work well in complex cases |
| [45] | Regions are used for extraction of features | Remarkably efficient For many vision problems | Training of network consumes much time |
| [46] | Uses scene knowledge, color and texture for | Upgrade results in term of precision | It also consumes much time while training |
| [47] | Block vise scene scanning with spatiotemporal scene modeling. Scene understanding with the help of binary scene model | Highly successful for moving objects | Cost for computation is high |

By the analysis of previous SOD techniques, it is shown that mostly techniques fall in either traditional, methods category while other in deep learning base methods. Some of them are hybrid methods which make use of both.
Although Deep learning based models have made significant progress and set the bar high for salient detection techniques but these methods still fail in some cases, which includes images that have low contrast between background and foreground, scenes with transparent objects and scenes with complex background. This enables the development of more effective and efficient methods in future.

# Chapter 3

# Research Methodology

This section describes the proposed methodology. We discuss different modules of our technique, how global and local features are extracted and architecture of refinement module.

## 3.1 Overview of Architecture

The proposed method consists of two main modules one is for the prediction of saliency map and other one refines the predicted saliency module. The prediction module consists of fully convolutional network which captures both local and global features. It also contains some modules that focuses on the refinement of features and better understanding of boundary features. Overview of model is presented in Figure 3.1. To further emphasize on low contrast images a contrast module is also embedded in model and to precisely highlight the object and to enhance the contours 5 global convolutional blocks and 5 refinement blocks are also added in the model.
Section 3.2 explains prediction module (which predicts saliency map) and section 3.5 explains refinemnet module which refines predicted saliency map.

## 3.2 Prediction Module

We have designed our prediction module on fully convolutional layers the entire network of our model is based on convolution operation and deconvolution layer having variable output dimensions which enables our model to capture global and local features from different resolutions as shown in Figure 3.1.

Input image is passed to five convolutional blocks to produce five feature maps $F_1$, $F_2$, $F_3$, $F_4$, $F_5$ respectively. Every convolutional block has kernel size 3x3 followed by a max pooling operation with stride 2 in order to reduce spatial resolution to 13x13 from 208x208. Pooling decreases, the number of parameters and computation. Max pooling is obtained by using max filter which captures the maximum of selected region and makes a new one. After that 5 Global Convolutional Blocks are added in the network which allows connection in features and convolutional layers, which enables the features to acquire diverse neural information and immune to local disruptions. We have also added 5 Boundary refinement blocks for 5 feature maps to keep boundary information. These boundary refinement blocks are based on residual structure in which the input to this block is directly added into the output of deeper layer which helps to maintain the information present in initial layers and this information is needed by the up sampling layers. These kind of structure also helps to avoid the loss of information due to multiple layers of network.

To generate local feature maps $F_L$, five more convolutional blocks are added in the network which processes the output of first five convolutional blocks. These convolutional layers also have kernel of 3x3 with 128 channels. The resulting diverse scale local feature maps are $F_6$, $F_7$, $F_8$, $F_9$, $F_10$. To further capture contrast features of each feature map $F_i^c$ (where i = 6, ... , 10) difference between feature map $F_i$ and its local average $F_i^*$ is calculated, which helps in better feature detection in low contrast images. Calculation of Local average is performed by average pooling with kernel size 3x3, $F_i^c$ is the resulting contrast feature.

$$Fi^c = Fi - Fi^{'} \tag{3.1}$$

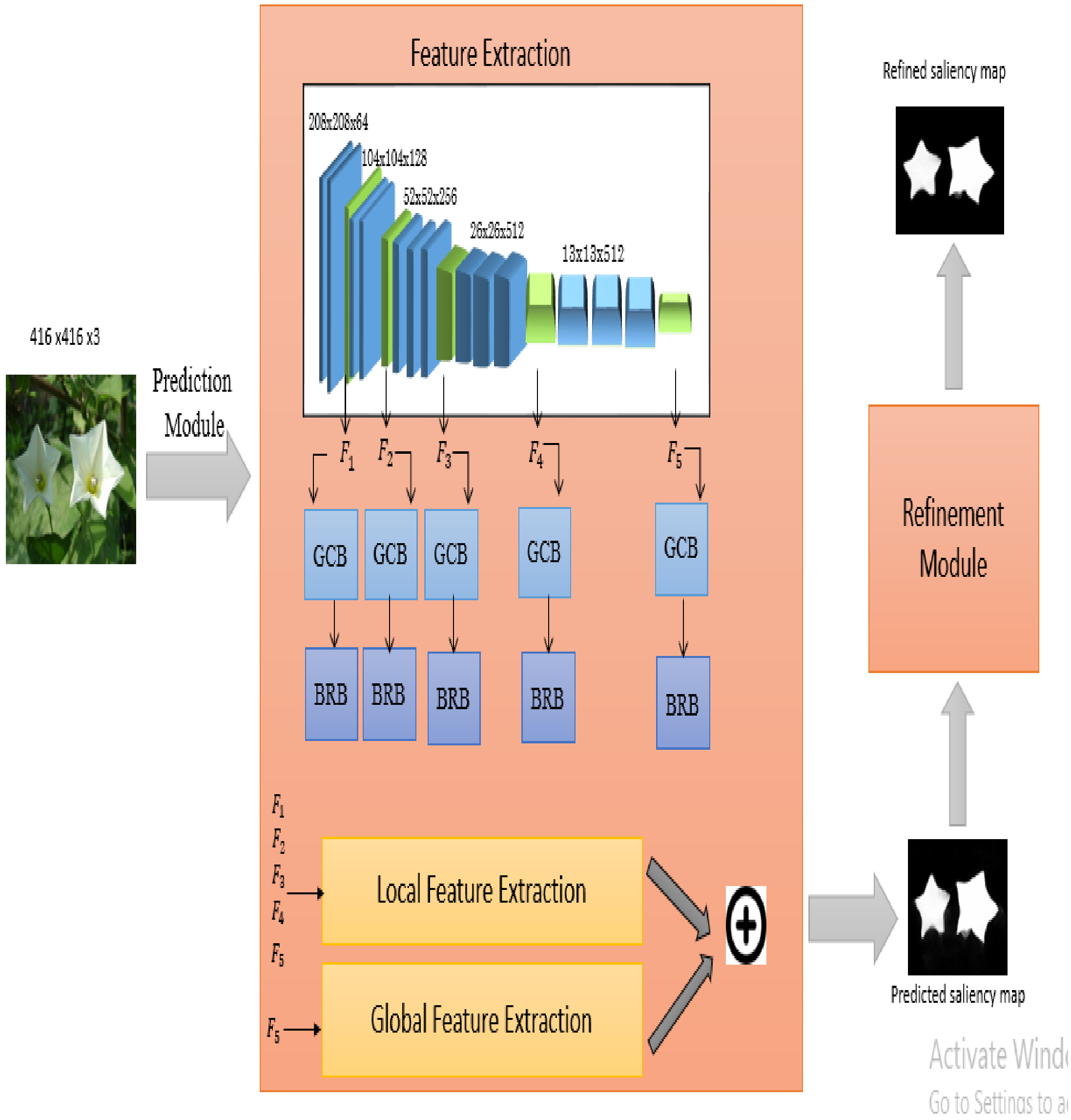


Figure 3.1: Overview of Proposed Model.

As we have performed pooling operations after convolutional layers to decrease the spatial size of image which greatly reduce computational cost so, to increase the spatial size of the feature map up sampling is applied, to achieve this de convolutional layer is attached to every feature map which increases its resolution by up sampling using kernel size 5x5 and stride 2. The final up sampled feature map is generated concatenation of contrast feature $F_i^c$, local feature $F_i$ and the up sampled feature map $F_(i+1)^u$.

$$Fi^u = Upsamp[CAT(Fi), Fi^c, Fi + 1^u] \tag{3.2}$$

To calculate Final local map $F_L$, local contrast feature $F_6^c$, local feature $F_6$ and up sampled feature map $F_7^u$ is concatenated and passed to convolution layer of kernel 1x1.

$$F^L = Conv(CAT(F_6, F_6^c, F_7^u)) \tag{3.3}$$

Figure 3.2 shows actual image and its local feature map is presented in Figure 3.3. For the computation of Global feature map $F_G$, global features are extracted before allocating saliency information of small region. For this purpose, three convolutional layers with kernel size 3x3 and 512 dilations (where dilation = 2) is applied. Dilated convolutions reduces the loss of resolution. Each convolutional layer is accompanied by RELU activation function and Boundary refinement block.Global Features extracted from Figure 3.2 are shown in Figure 3.4.

To produce predicted saliency map global $F_G$ and local feature maps $F_L$ are combined and the resulting saliency map $S_P$ is passed as input to refinement module. Figure 3.5 shows predicted saliency map produced by the combination of Local and Global feature maps.

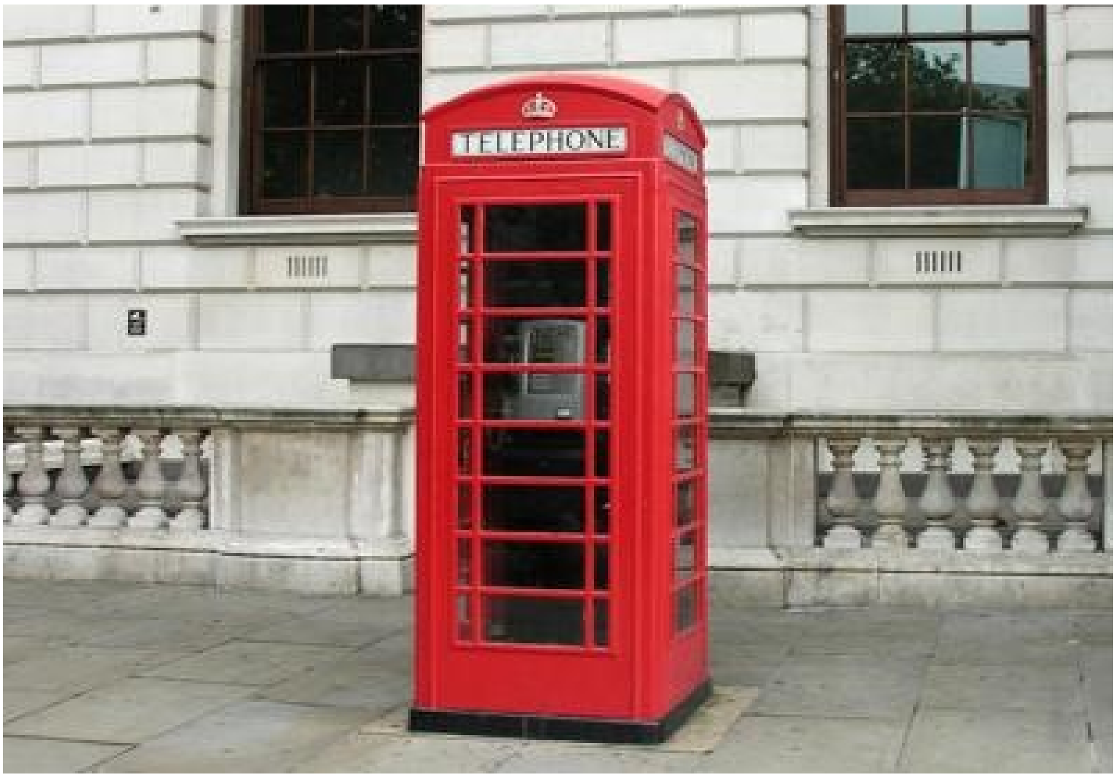


Figure 3.2: Actual Image

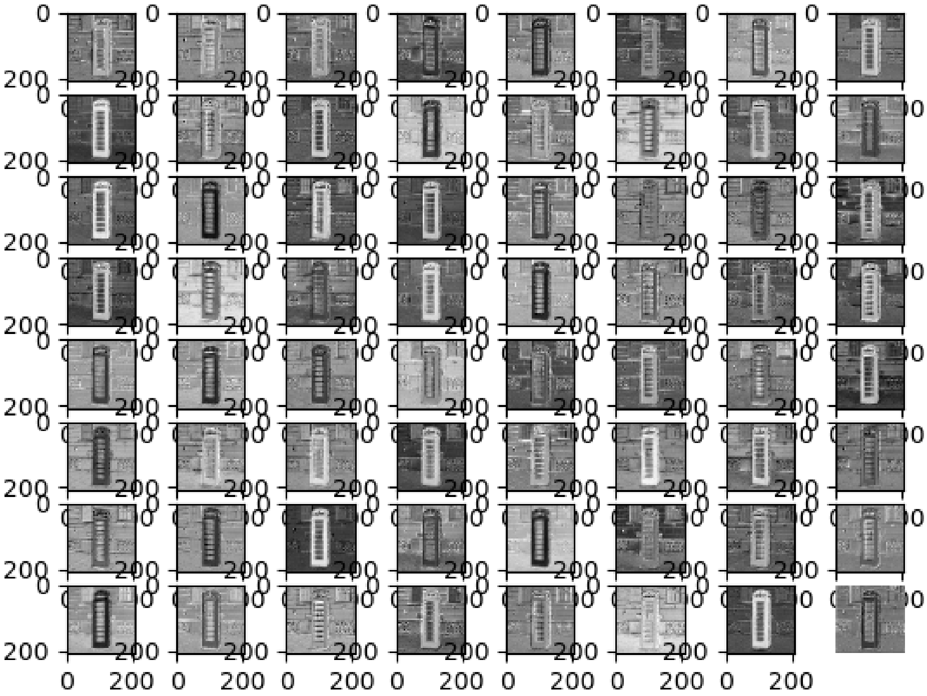


Figure 3.3: Local Feature Map of Figure 3.2

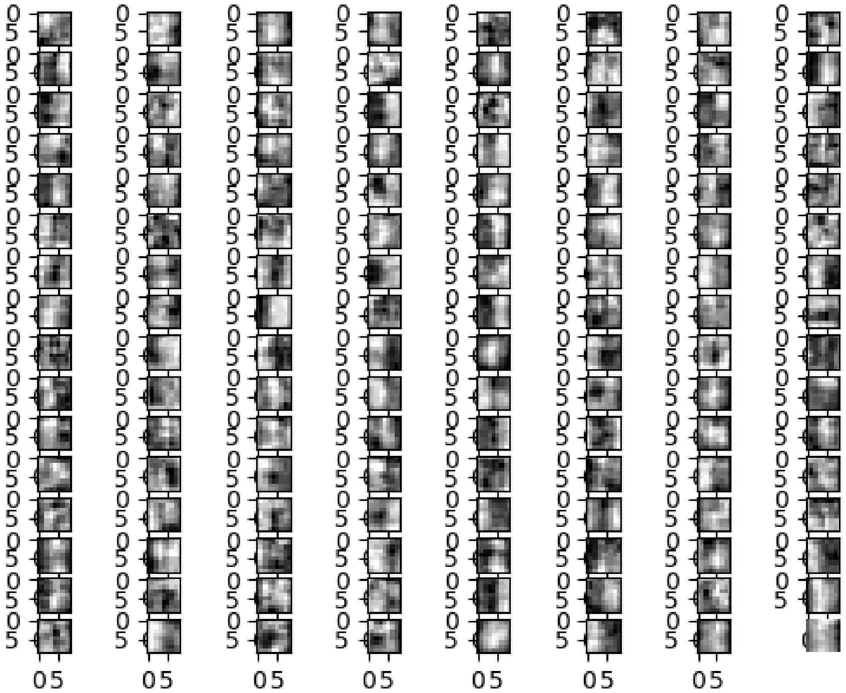


Figure 3.4: Global Features Extracted From Figure 3.2

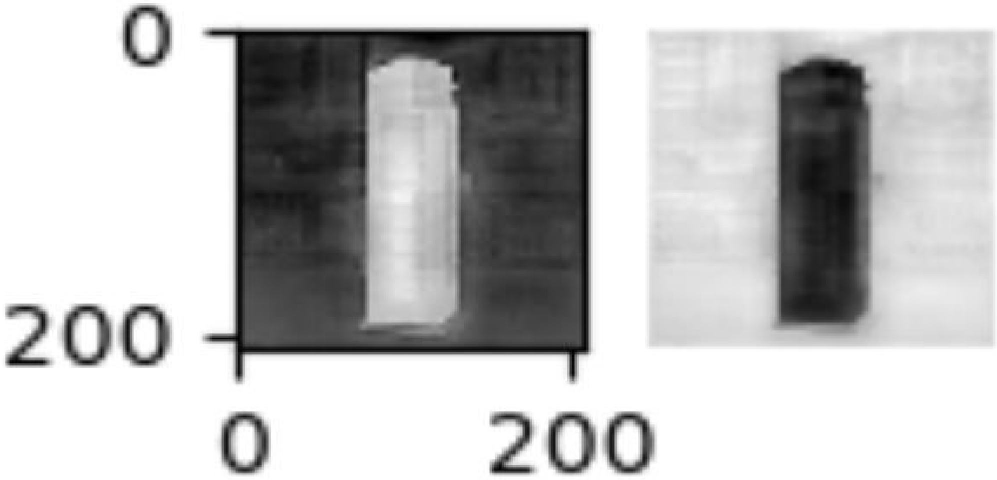


Figure 3.5: Predicted Saliency Map by Combining Local and Global Features

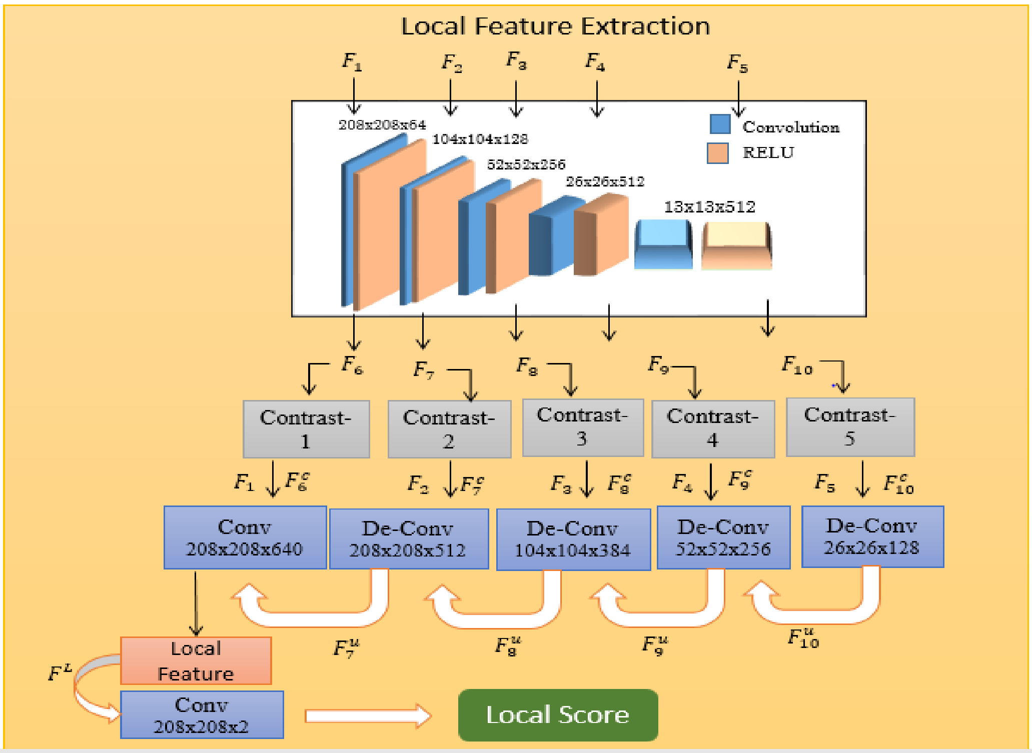


Figure 3.6: Extraction of Local Features

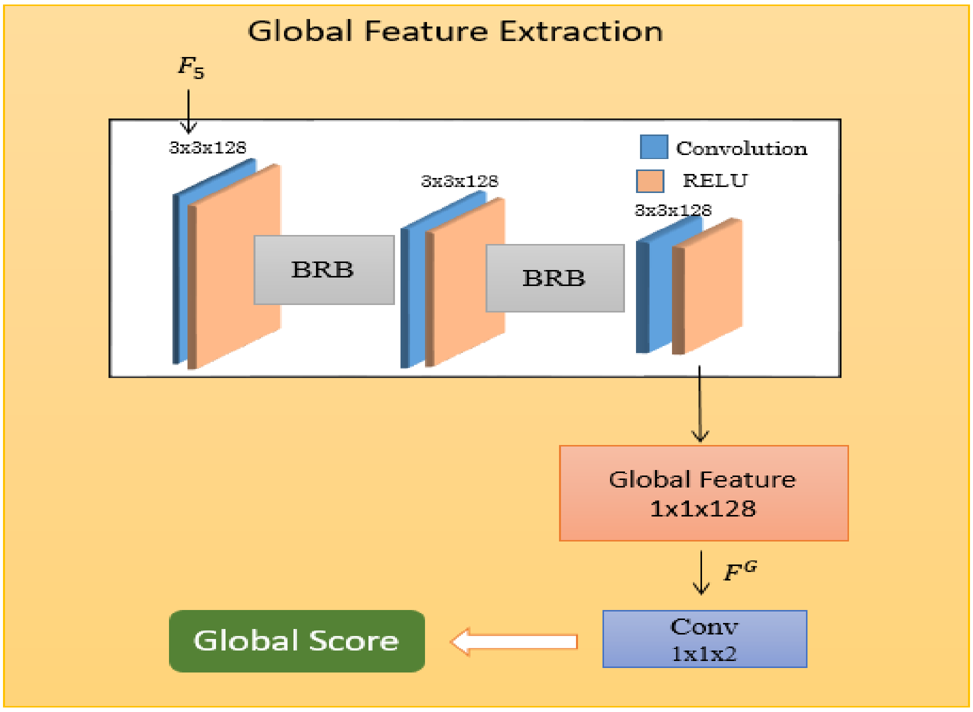


Figure 3.7: Extraction of Global Features

## 3.3 Global Convolutional Block

The refinement of network is achieved by using global convolutional block to efficiently exploit accessible visual information in low contrast images. GCB seeks to extend receptive region of feature maps and to introduce dense connections between features and classifiers, which helps the model to identify most observant areas and collect semantic information of object with very little extra cost. Structure of Global Convolutional Block is shown in Figure 3.3.

GCB enhances the classification efficiency of proposed model by taking into account the dense connections among classifiers and feature maps, which enables the network to handle different type of transitions. Besides that, GCB's large kernel is useful to encode more spatial knowledge from feature map, which increases the precision in location salient object.

GCB has two sub divisions, both of them consist of two convolutional blocks. Left block has 7x1 convolutional layer after which comes another convolutional layer of 1x7. Right one has 1x7 convolution followed by 7x1 convolution. These two sub

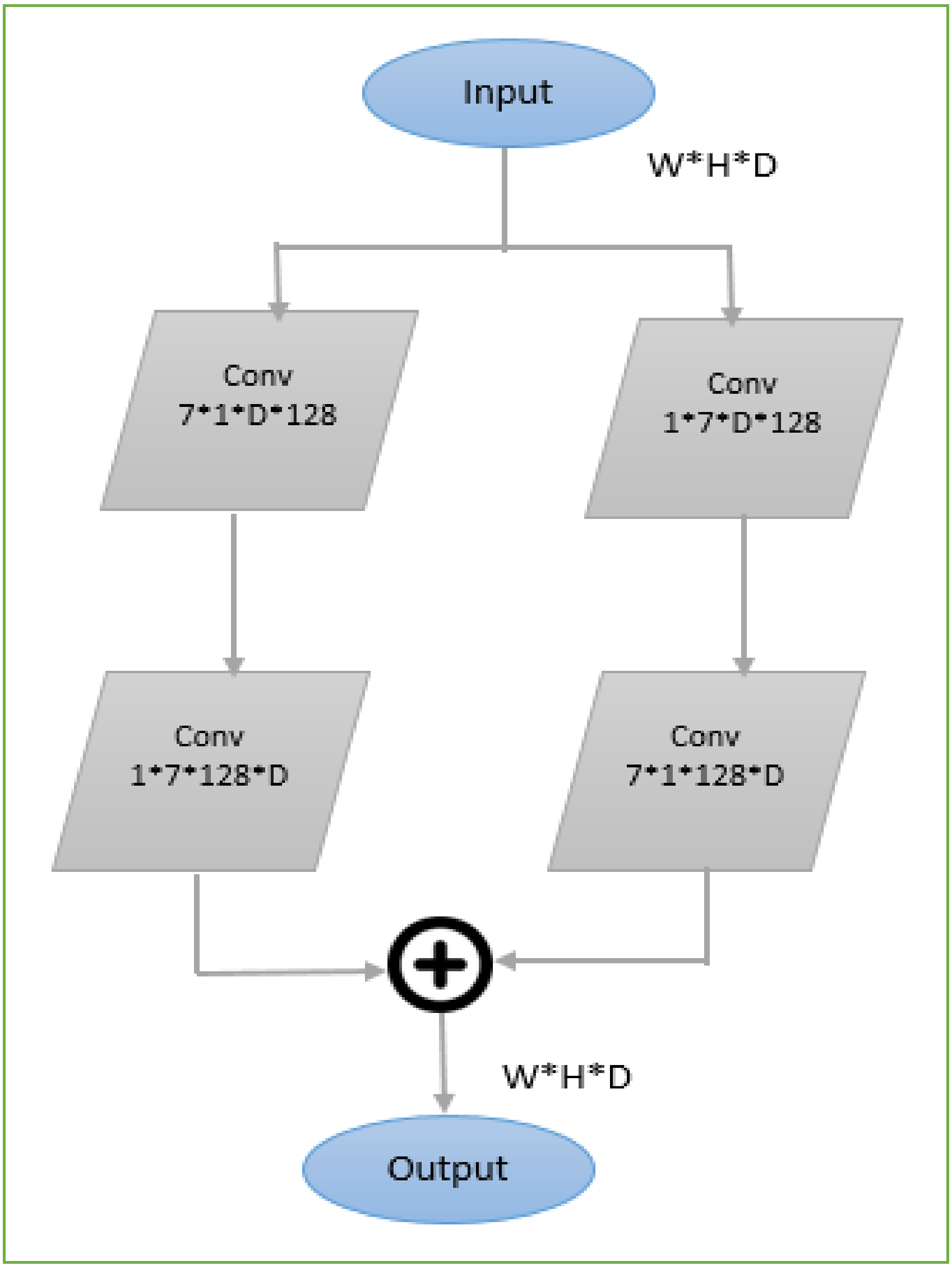


Figure 3.8: Structure of Global Convolutional Block

divisions are combined to allow dense connections which enhances receptive field's validity. GCB's computational cost is fairly low which makes it more practice.

## 3.4 Boundary Refinement Block

The second module which is included in the network is BRB. It is a residual framework, implanted after first five convolutional layers to maintain the boundary information of object. This block is added for the refinement of feature maps and to enhance the accuracy of object's spatial location. As we have already mentioned in section 3.2 that this helps to reserve the information present in initial layers. BRB is included to improve localization near boundaries, which can significantly maintain boundary knowledge in training phase. As shown in Figure 3.9 its one part directly connects input and output without performing any action and other part is residual net, comprising of two convolutional layer with kernel size 3x3. These two parts are merged by short connection which is useful in learning of boundary knowledge, which enables refinement of boundary pixel.

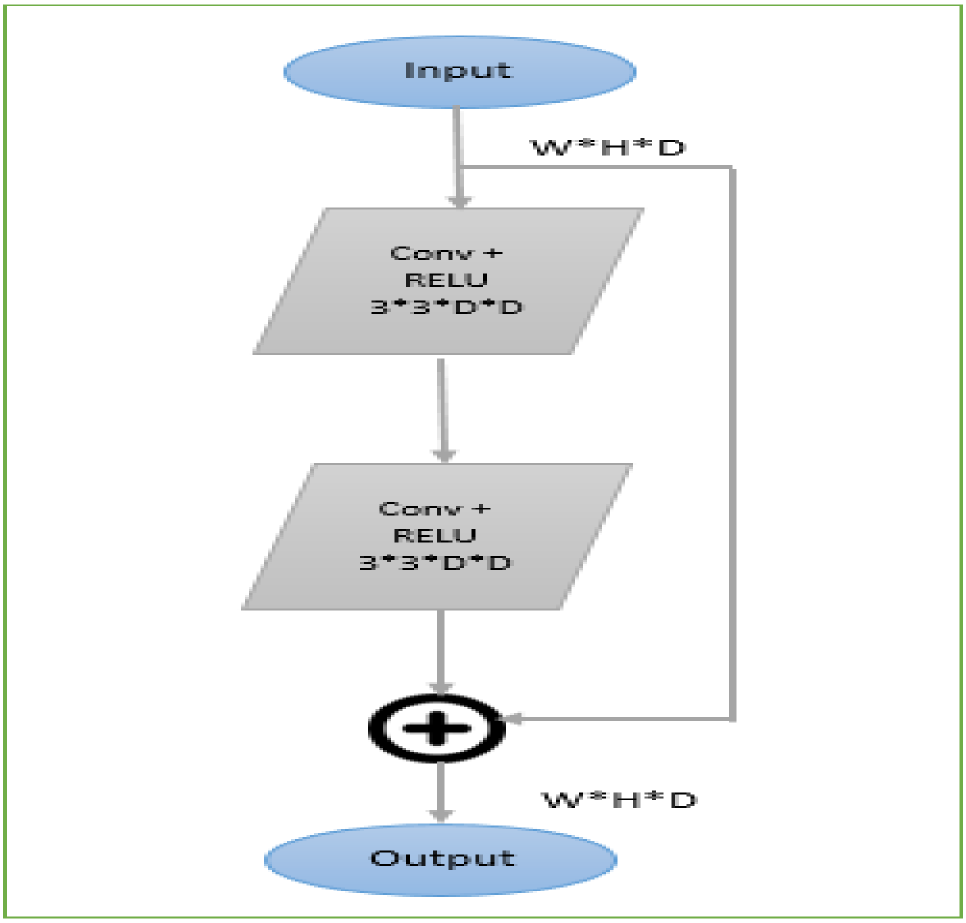


FIGURE 3.9: Structure of Boundary Refinement Block

## 3.5 Refinement Module

This module is added for overall refinement of saliency map. This module is different from BRB which we have added in initial layers as that block was used to refined the feature maps and to preserve the rich spatial information present in initial layers, the purpose of this module is to refine the predicted saliency map to enhance accuracy of model. Predicted saliency map $S_P$ is input for this refinement module. Typically, a refinement module is built as a residual block that purifies input saliency map through learning residual among saliency map and its ground truth.

We have embedded a refinement module to improve both the region and the boundary limitations in coarse saliency map [52]. Coarse saliency map refers to blurriness of boundaries and un even prediction of probabilities in regions. As explained in Figure 3.11 Refinement module consists of input layer followed by an encoder, a decoder, a link stage between encoder network and decoder and output layer. Both the encoder and decoder networks comprises of four stages. Every stage consists of one convolutional layer with 64 filters 3x3 size. After each convolutional layer batch normalization and RELU function is performed. Link stage also contains a convolutional layer with same filter and size after which normalization and RELU is performed. Decoder follow same structure as encoder.

To down sample in encoder max pooling is used and for up sampling bilinear interpolation is used in decoder. The resulting map is the final Saliency map $S_F$. Refined feature map is shown in Figure 3.10.

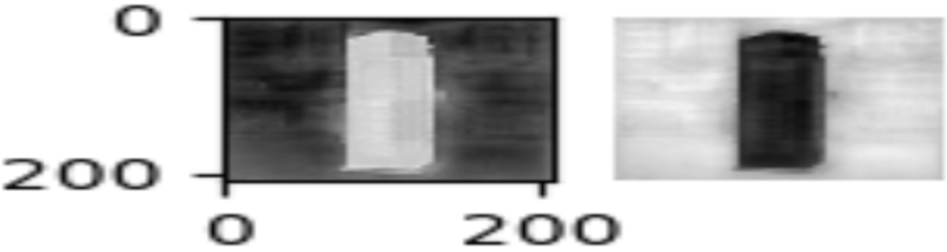


FIGURE 3.10: Predicted and Refined Saliency Maps

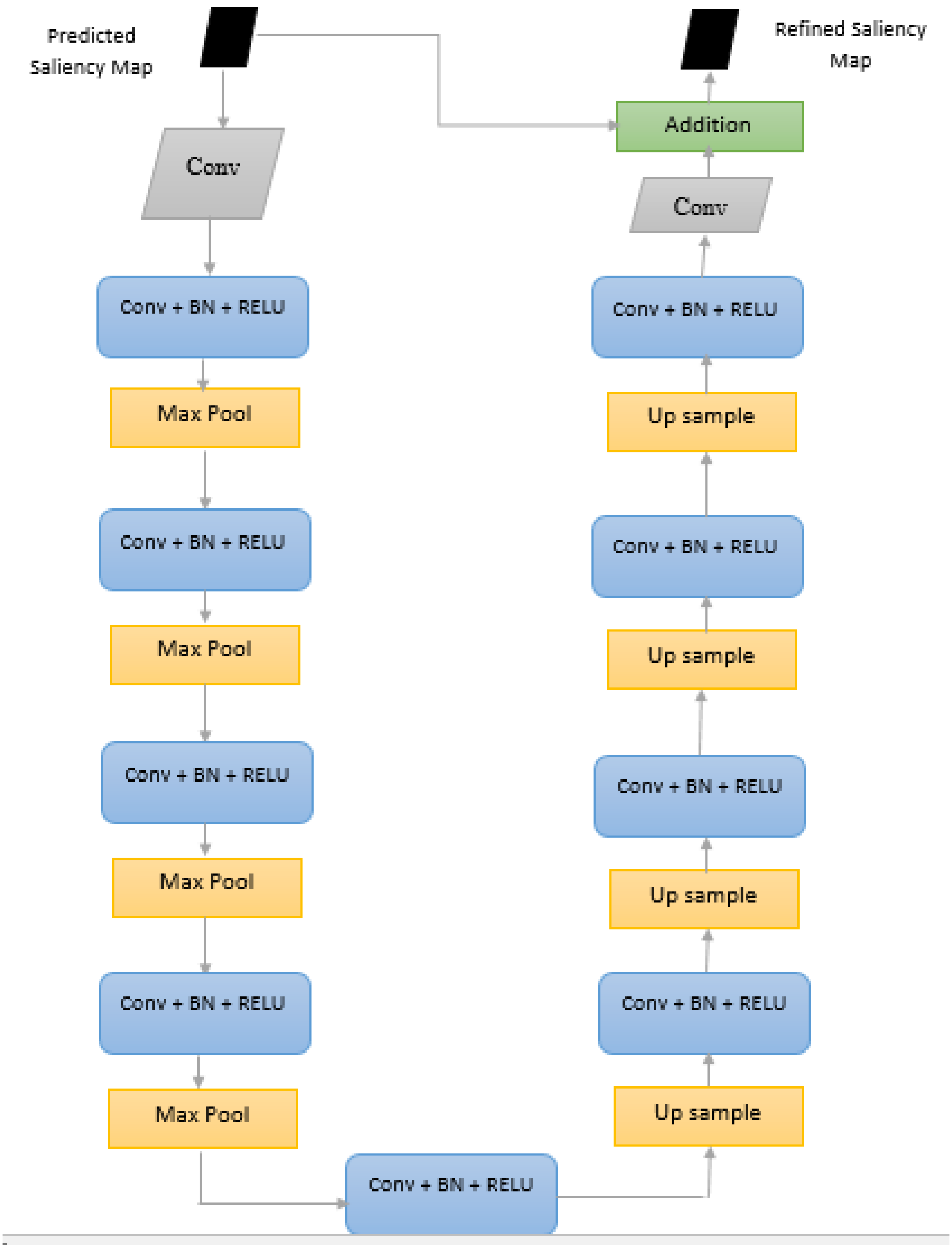


Figure 3.11: Structure of Refinement Module

TABLE 3.1: Results With and Without Refinement Module

| | | **Precision** | **MAE** |
|---|---|---|---|
| Without Refinement | Image 1 | 0.741 | 0.085 |
| | Image 2 | 0.930 | 0.047 |
| | Image 3 | 0.935 | 0.030 |
| | Image 4 | 0.880 | 0.024 |
| With Refinement | Image 1 | 0.773 | 0.069 |
| | Image 2 | 0.966 | 0.034 |
| | Image 3 | 0.950 | 0.026 |
| | Image 4 | 0.9305 | 0.013 |

To show the effectiveness of refinement module in network we have tested some images with refinement module and similar images without refinement module. We have also computed precision and mean absolute error of these images which are shown in Table 3.1. For better understanding, saliency maps of images with or without refinement module is also presented in Table 3.2.

From Table 3.1 it is clearly shown that refinement module has effectively improved the results and the produced maps are more refined and clear in structure.

## 3.6 Training Loss

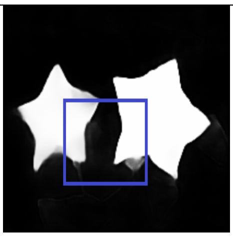
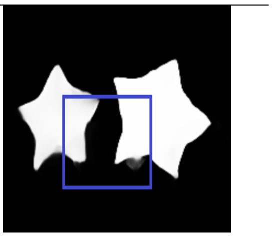
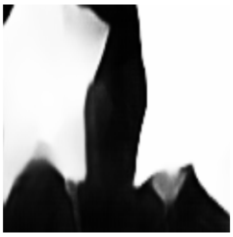
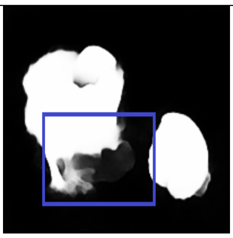
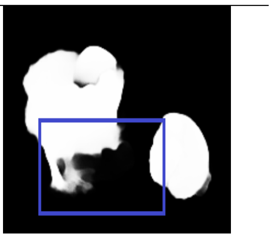
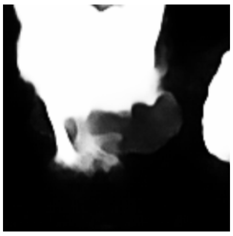
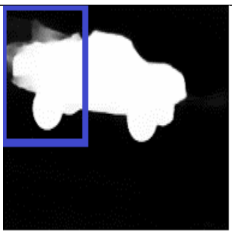
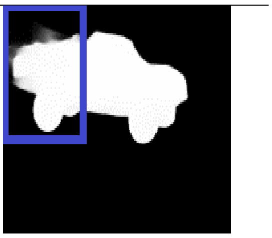
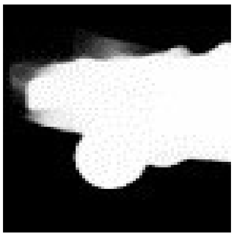
The refined final saliency map $S_F$ is computed by the refinement of predicted saliency map. Softmax Function computes probability that a pixel p in feature map belongs to salient object or not [51].

$$S^M(p) = P(G^T(p) = l) = \frac{e^{w_l^L F^L(p) + v_l^L + w_l^G F^G + v_l^G}}{\sum_{l' \epsilon (0,1)} e^{w_{l'}^L F^L(p) + v_{l'}^L + w_{l'}^G F^G + v_{l'}^G}} \tag{3.4}$$

Where ( $W^L$ , $v^L$ ) and ( $W^G$ , $V^G$ ) are linear operators. The loss function is the combination of Cross Entropy Loss and Boundary Loss.

The Cross Entropy Loss among predicted saliency map and ground truth for a

Table 3.2: Saliency Maps With or Without Refinement Module.

| **Without Refinement** | **With Refinement** |
|---|---|
|  |  |
|  | |
|  |  |
|  | |
|  |  |
| |  |

region can be calculated as:

$$Loss^{CE} = -\frac{1}{N}\sum_{j=1}^{N}\sum_{L\epsilon(0,1)}(G^{T}(p) = l)(log(S^{M}(p_{j} = l))) \tag{3.5}$$

The boundary Loss can be calculated as:

$$Loss^{B} = 1 - \frac{2|B_{r}^{T} \cap B_{r}^{M}|}{|B_{r}^{T}| + |B_{r}^{M}|} \tag{3.6}$$

These two losses are combined to train the proposed model. So, the parameters utilized for SOD can be optimized.

## 3.7 Differences with other Frameworks

Proposed method is based on extraction of local and global features with boundary refinement and some existing works like [67], [68] and [69] are also based on extraction of local and global features, but our learning strategy is extremely different. Zhang et al., proposed a Background-based map with new similarity metric is merged with centre-prior and Objectness measure to form a global saliency map, integrating low-level and high-level features with coding-based methods. The locality-based coding approach is then suggested for obtaining a local saliency map [67].

Wang et al., have produced a local feature map by combining the background saliency map generated by the multi-feature mode and the foreground saliency map generated by the contrast method of the foreground area. The deep convolution neural network (CNN) was then used to train the global image with the super pixel block centre, whose mark is the ground truth of the super pixel centre. Thus the saliency map of the global dimension was acquired. The final saliency map was presented by merging the local integrity and the global satisfaction map [68].

Wang et al., proposed a model in which area dependent local and global saliency is determined, the local saliency is computed by multi-scale neighbourhood comparison, and the global saliency is measured according to global spatial distribution and inter-region isolation of features.The final saliency can be obtained by the weighted combination of them on the basis of local saliency and global saliency [69].

All of these models are using combination of low and high level features and are extracting features by segmentation image in different parts. Unlike these methods proposed model introduce following novel contributions:

- Generation of local and global feature maps by using fully convolutional networks which helps our model to learn detailed structure of objects.
- Focus on detection of objects in low contrast images by incorporating contrast feature and a global convolutional module which enhances the classification ability by introducing dense connections between features and classifiers.
- A Boundary refinement module to preserve the boundary information present in initial layers.

## 3.8 Algorithms for Feature Extraction and Refinement

Algorithm 1 describes the algorithm for refinement module which takes predicted map as a input and perform functions on it and then return refined saliency map. The detailed algorithm of feature extraction is presented in algorithm 2, which shows the extraction of local and global features and calculation of final saliency score.

Algorithm 2 shows the flow of feature extraction from images, it takes images as a input and extract features from them and at the end it return added local and global features.

**Algorithm 1** : Refinement Module

**Require:** *Predicted saliency map*

**Ensure:** *Refined Saliency map*

```
function REFINEMENT(predicted map)
    $ref\_conv0 \leftarrow Conv(input\_features)$                    ▷ Conv=Convolution
    $ref\_conv1 \leftarrow Read\,Conv(ref\_conv0)$
    $bn1 \leftarrow batchNormalization(ref\_conv1)$
    $relu1 \leftarrow Relu(bn1)$
    $Pool1 \leftarrow max\_pool(relu1)$
    for $i \leftarrow 1$ to 5 do
        $ref\_conv[i] \leftarrow Conv(pool[i-1])$
        $bn[i] \leftarrow batchNormalization(ref\_conv[i])$
        $relu[i] \leftarrow Relu(bn[i])$
        $Pool[i] \leftarrow Read\,max\_pool(relu[i])$
    end for
    for $i \leftarrow 4$ to 1 do
        $ref\_conv\_d[i] \leftarrow Conv(concat(Upsamp(relu[i+1], relu[i])))$
        $bn\_d[i] \leftarrow batchNormalization(ref\_conv\_d[i])$
        $relu[i] \leftarrow Relu(bn[i])$
        $relu\_d[i] \leftarrow Read\,Relu(bn\_d[i])$
    end for
    $conv\_d0 \leftarrow Conv(relu\_d[1])$
    return $input - features + conv\_d0$
end function
```

FIGURE 3.12: Refinement Module

**Algorithm 2** : Feature Extraction

**Require:** *Images*

**Ensure:** *Learned Feature Score*

```
function $buildModel(images)$
    for $i \leftarrow 1$ to 5 do
        $Vgg.pool[i] \leftarrow GCB(vgg.pool[i])$ ▷ GCB=Global Convolutional Block
        $Vgg.pool[i] \leftarrow BRB(vgg.pool[i])$ ▷ BRB=Boundary Refinemnet Block
        $F^G \leftarrow RELU(Conv(vgg.pool5))$
        $F^G \leftarrow BRB(F^G)$
    end for
    for $i \leftarrow 5$ to 1 do
        $F[i] \leftarrow RELU(Conv(vgg.pool[i]))$
        $F_i^c \leftarrow Read\,Contrast(F[i])$
    end for
    for $i \leftarrow 5$ to 2 do
        $F^u[i] \leftarrow Upsamp(CAT(F[i], F_i^c))$
    end for
    $Score \leftarrow Local\_Score + Global\_Score$
    return $Score$
end function
```

Figure 3.13: Feature Extraction

# Chapter 4

# Result and Evaluation

This chapter contains details that how the proposed model is evaluated, datasets used for evaluation, metrics for evaluation and comparison of result with other SOD models. We will also discuss experimental setup used for evaluation and computational cost of model.

## 4.1 Experimental Set-up

The proposed model is based on deep learning which requires good computational power for both training and testing of model. Training of model requires more computational power as compared to testing of model. Specifications of machine used in training are given in Table 4.1 and specs of machine used for testing are shown in Table 4.2. For implementation of network we have used PyCharm framework and model is implemented in tensor flow.

Table 4.1: Specification of System Used for Training

| | |
|---|---|
| OS | Linux: Ubuntu 16.04 .6LTS |
| GPU | Nvidia GeForce GTX 1080 (8119 MiB) |
| RAM | 62 GB |
| Cuda Version | 10.1 |
| Nvidia driver version | 418.43 |
| Interpreter | Python 3.7 |

Table 4.2: Specification of System Used for Testing

| | |
|---|---|
| CPU | Intel (R)Core$^{TM}$ i5-5300U CPU @ 2.30GHz |
| RAM | 8.00 GB |
| OS | Windows 10 Pro |
| System Type | 64 bit |
| Storage | 256 GB SSD |
| **Tool Used For Implementation** | |
| Framework | Py Charm |
| Interpreter | Python 3.7 |

## 4.2 Dataset for Training

We have trained our model on MSRA-B dataset which contains 5000 images [54]. Computational cost of model training is given in coming section. To better explain model accuracy and loss while raining we have added some graphs which shows the accuracy and loss of model after 100 steps and also shows average loss and accuracy of model.

## 4.3 Computational Cost

This section explains the computational complexity of proposed model in terms of training time, number of parameters used in model, floating point operations and number of convolutional blocks.

### 4.3.1 Training Time

It is the total time required by the model for complete training. For training we have used MSRA-B dataset which contains 2500 imges for training set and 500 images for validation set.

Our model took about 12.5 hours for complete training in 10 epochs on Intel core i-7 3.20GHz CPU with Nvidia GTX GPU, which makes 75 minutes per epoch for 5000 images.

### 4.3.2 FLOPs

FLOPs are number of floating point operations required from the input layer till the output layer during inference/prediction. Proposed model took 381349256782 FLOPs for complete execution. The details are given in Figure 4.1.

### 4.3.3 Convolutional Blocks

These are number of convolutional layers used in model, this also loosely defines the depth and hence the time complexity of proposed model. Proposed model have 21 convolutional layers with varying filter sizes. The details are given in Table 4.3.

### 4.3.4 Parameters

Number of parameters used in model defines the size of model on disk and in memory as well, along with loosely defining the compute time required. Total number of trainable parameters used in model are 60400106. The details are given in Table 4.5.

## 4.4 Comparison with other Recent Deep Learning Model

To compare complexity of model we have compared proposed model with recent deep learning based saliency detection models GCBR [74] and NLDF [70], in Table 4.4 comparison shows that our model has got second place, number of parameters used in proposed mode are less than GCBR which mean that it consumes less space and it is efficient than GCBR in term of computation and produces much better results than these two models which shows the effectiveness of proposed model.

Table 4.3: Convolution Blocks in Model

| Convolution | Filter size | Dilation |
|---|---|---|
| Conv 1 | 3 x 3 | - |
| Conv 2 | 3 x 3 | - |
| Conv 3 | 3 x 3 | - |
| Conv 4 | 3 x 3 | - |
| Conv 5 | 3 x 3 | - |
| Conv (GCB) | 7 x 1 | - |
| Conv (GCB) | 1 x 7 | - |
| Conv (GCB) | 1 x 7 | - |
| Conv (GCB) | 7 x 1 | - |
| Conv (BRB) | 3 x 3 | - |
| Conv (BRB) | 3 x 3 | - |
| Conv 6 | 3 x 3 | - |
| Conv 7 | 3 x 3 | - |
| Conv 8 | 3 x 3 | - |
| Conv 9 | 3 x 3 | - |
| Conv 10 | 3 x 3 | - |
| Conv 11 | 1 x 1 | - |
| Conv 12 | 3 x 3 | 2 |
| Conv 13 | 3 x 3 | 2 |
| Conv 14 | 3 x 3 | 2 |
| Conv 15 | 1 x 1 | - |
| Total Layers: | 21 | |

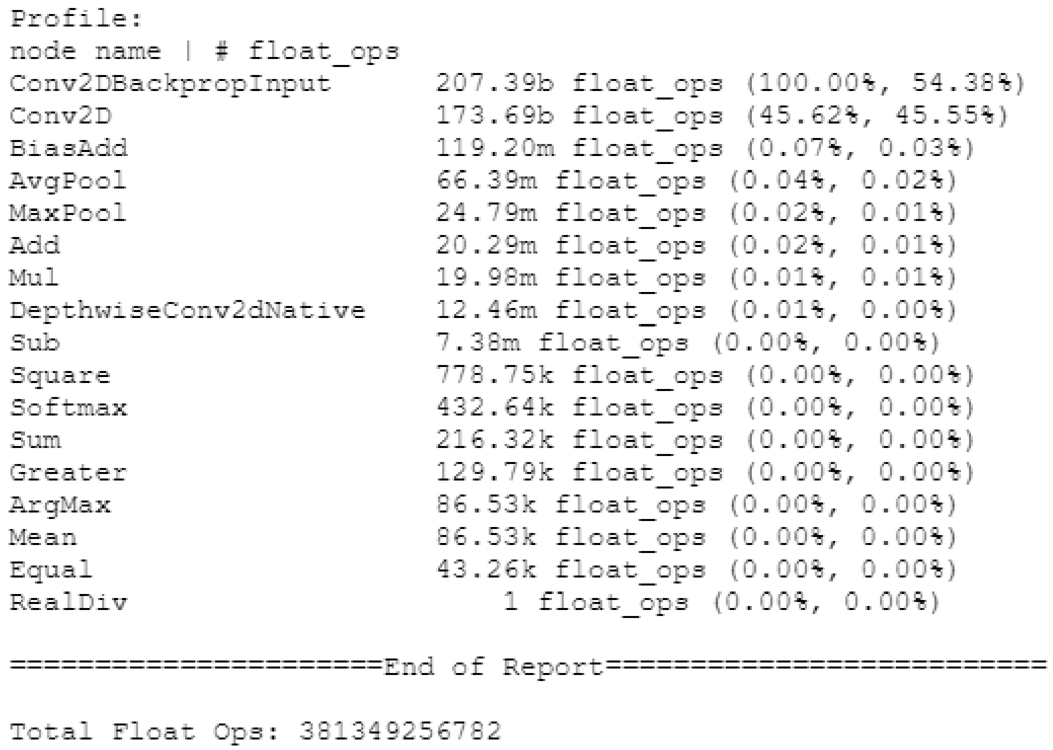
Table 4.4: Complexity Comparison With Other Models

| Metrics | SODL (proposed) | GCBR[74] | NLDF[70] |
|---|---|---|---|
| No. of Parameters | 60400106 | 71650368 | 35485928 |
| Avg Execution Time | 8.5 | 13.5 | 4.0 |

## 4.5 Model Accuracy and Loss

To show the model accuracy and loss during training we have constructed some graphs, Figure 4.2 shows the current accuracy of model after every 100 steps. Here, X-axis shows the number of steps and Y-axis shows accuracy. The graph shows that our model has even achieved accuracy of 0.99 for some steps. The accuracy of model after every 100 steps is gradually increasing which means that learning ability our model is getting better.

Figure 4.3 shows average accuracy of model throughout the training process after every 100 steps. To give the better view of model's accuracy we have plotted



```
Profile:
node name | # float_ops
Conv2DBackpropInput       207.39b float_ops (100.00%, 54.38%)
Conv2D                    173.69b float_ops (45.62%, 45.55%)
BiasAdd                   119.20m float_ops (0.07%, 0.03%)
AvgPool                   66.39m float_ops (0.04%, 0.02%)
MaxPool                   24.79m float_ops (0.02%, 0.01%)
Add                       20.29m float_ops (0.02%, 0.01%)
Mul                       19.98m float_ops (0.01%, 0.01%)
DepthwiseConv2dNative     12.46m float_ops (0.01%, 0.00%)
Sub                       7.38m float_ops (0.00%, 0.00%)
Square                    778.75k float_ops (0.00%, 0.00%)
Softmax                   432.64k float_ops (0.00%, 0.00%)
Sum                       216.32k float_ops (0.00%, 0.00%)
Greater                   129.79k float_ops (0.00%, 0.00%)
ArgMax                    86.53k float_ops (0.00%, 0.00%)
Mean                      86.53k float_ops (0.00%, 0.00%)
Equal                     43.26k float_ops (0.00%, 0.00%)
RealDiv                       1 float_ops (0.00%, 0.00%)

======================End of Report==========================

Total Float Ops: 381349256782
```

FIGURE 4.1: Floating Point Operations of Model

TABLE 4.5: Parameter Detail of Model

| **Activation Shape** | **Parameters** |
|---|---|
| (3, 3, 3, 64) | 1728 |
| (3, 3, 64, 64) | 36864 |
| (3, 3, 64, 128) | 73728 |
| (3, 3, 128, 128) | 147456 |
| (3, 3, 128, 256) | 294912 |
| (3, 3, 256, 256) | 589824 |
| (3, 3, 256, 512) | 1179648 |
| (3, 3, 512, 512) | 2359296 |
| (7, 1, 512, 128) | 458752 |
| (7, 1, 256, 128) | 229376 |
| (7, 1, 128, 128) | 114688 |
| (7, 1, 64, 128) | 57344 |
| (3, 3, 512, 512) | 2359296 |
| (3, 3, 256, 256) | 589824 |
| (3, 3, 128, 128) | 147456 |
| (3, 3, 64, 64) | 36864 |
| (3, 3, 512, 128) | 589824 |
| (3, 3, 64, 2) | 1152 |
| Total Parameters: | 60400106 |

average accuracy of model after every 100 steps, which shows overview of model accuracy during each step. Our model has shown great improvement in accuracy by time.

Figure 4.4 plots current loss of model after every 100 steps, on X-axis loss is given and on Y-axis steps are given. It shows that loss of our model has decreased by the time and has reduced to 0.3, as the accuracy of model is increasing loss after every 100 step is decreases which shows the improvement in our model's learning during training.

Figure 4.5 shows plot of average loss of model throughout the training process after every 100 steps, it shows that loss of our model has gradually decreased throughout the network. To give the overview of training loss after every 100 steps we have plotted average loss after every 100 steps during the training of model.

In Figure 4.6 we have plotted average accuracy of model for current epoch after every 100 steps, our model take 10 epochs for whole training and average accuracy of current running epoch is plotted in this graph which give the better understanding of model average accuracy for running epoch. Similarly, in Figure 4.7 we have plotted average loss of model for current epoch after every 100 steps which shows that the loss of model has gradually decrease after every epoch and accuracy has efficiently increased.

## 4.6 Datasets for Evaluatsion

We have performed testing on five benchmark datasets:

- ***MSRA-B*** [54] comprises of 5000 images. Contains single object mostly around center position along with bounding box label.

- ***DUT-OMRON*** [33] includes 5168 images with complex background and variety of content. Pixel wise ground truth annotations are also available.

- ***PASCAL-S*** [58] dataset contains 850 complex images. Along with eye fixation records non binary and pixel wise annotations are also available.

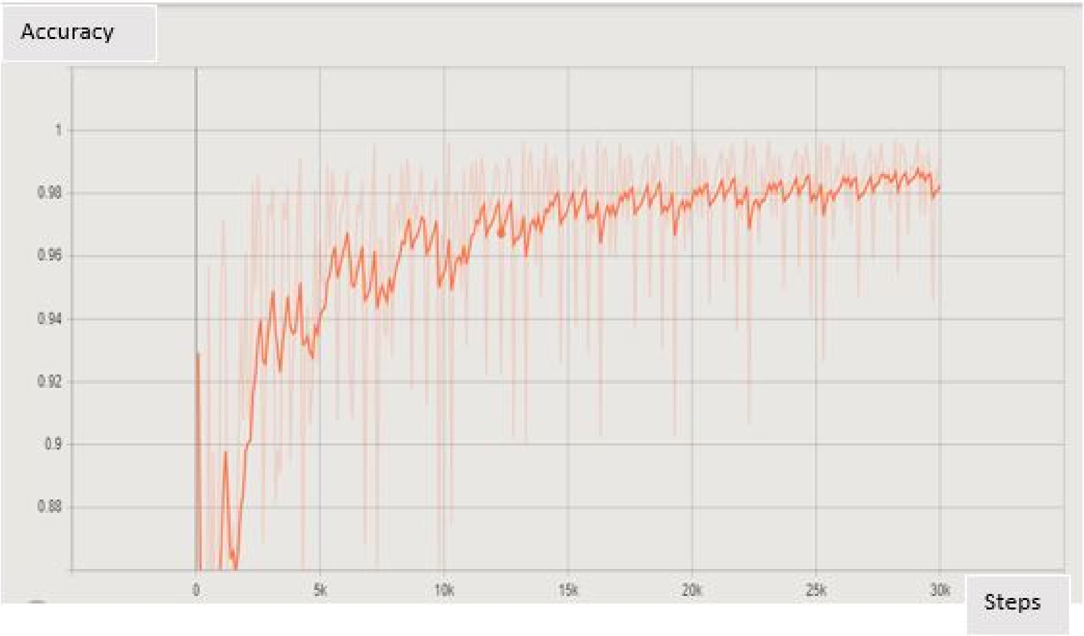


FIGURE 4.2: Plot of Accuracy vs Steps

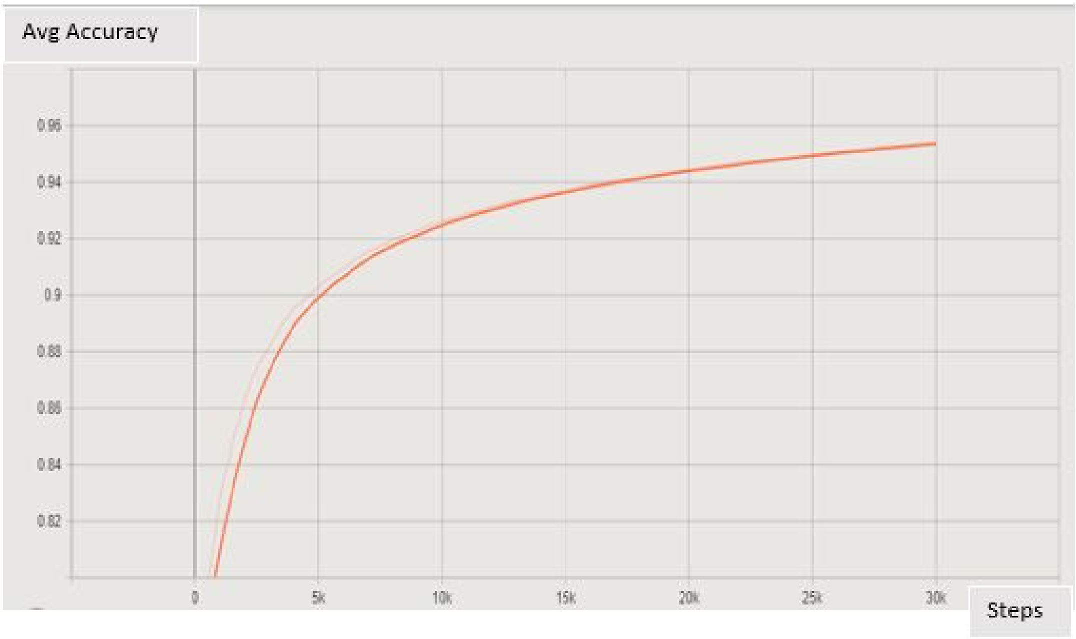


FIGURE 4.3: Plot of Average Accuracy vs Steps

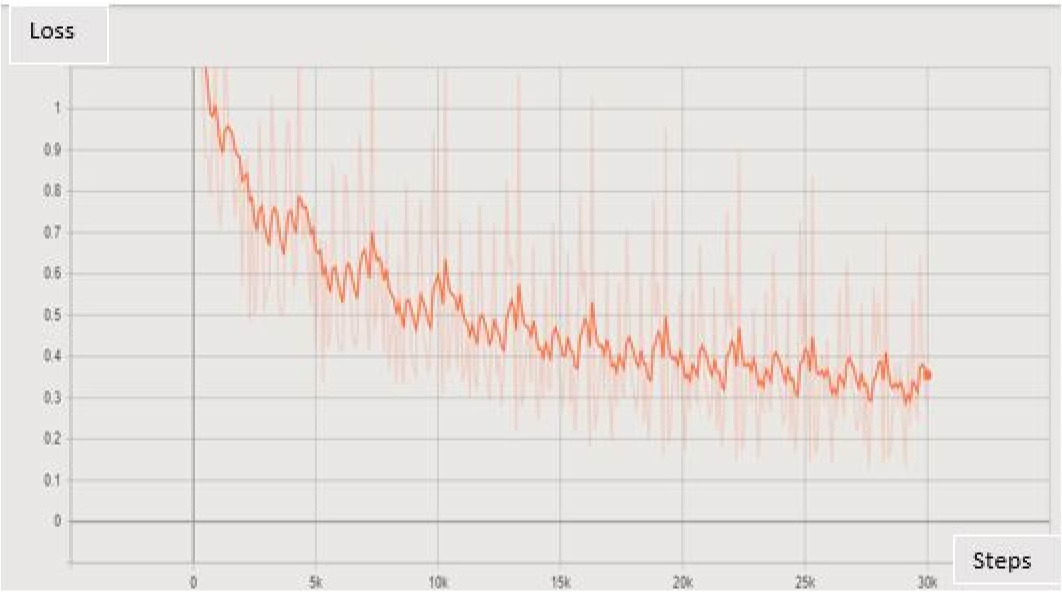


FIGURE 4.4: Plot of Loss vs Steps

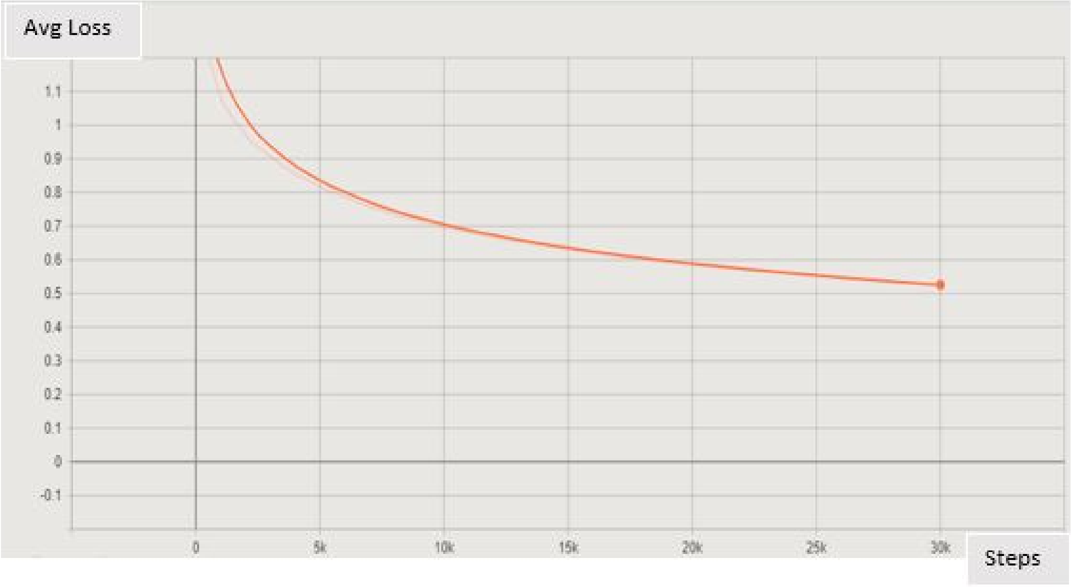


FIGURE 4.5: Plot of Average Loss vs Steps

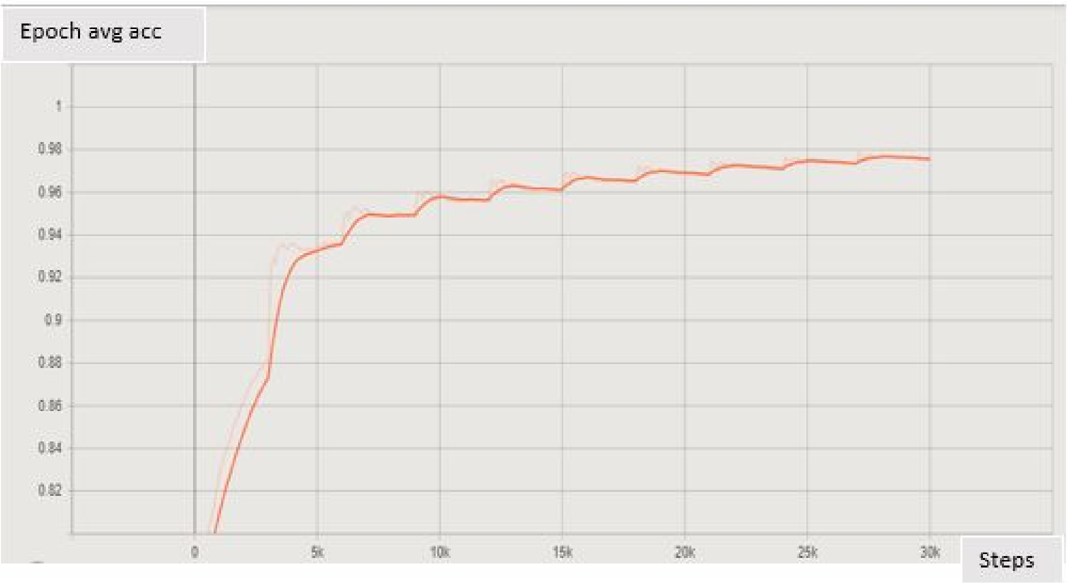


FIGURE 4.6: Plot of Epoch Average Accuracy vs Steps

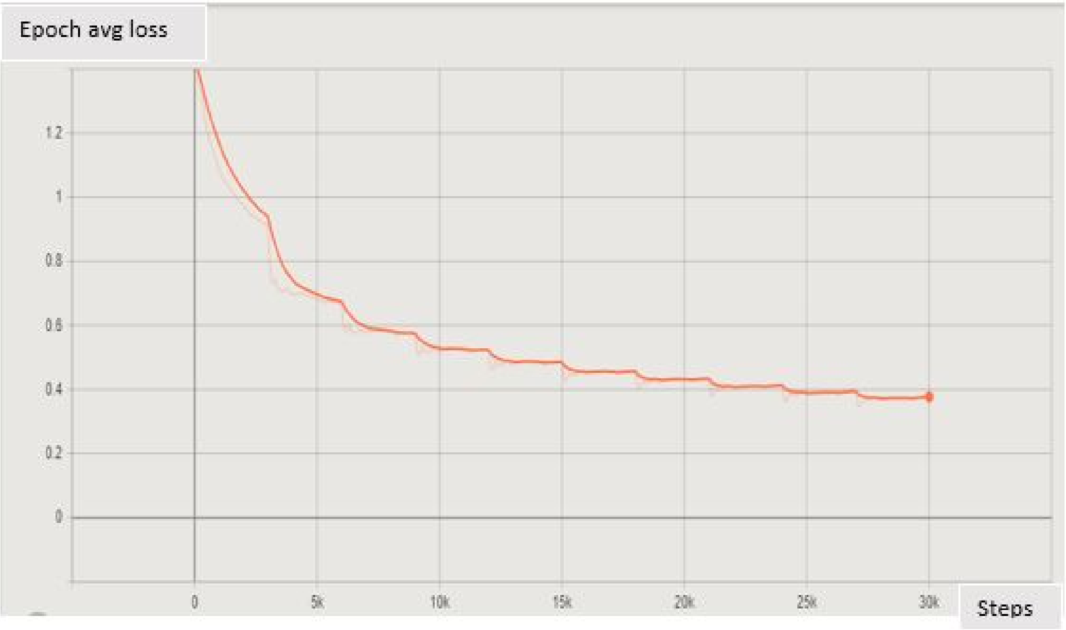


FIGURE 4.7: Epoch Average Loss vs Steps

- ***HKU-IS*** [59] comprises of 4447 multiple distant objects. In these minimum one object is present at the image boundary. Less difference between background and foreground makes these images more complex.
- ***DUTS*** [60] is very large Sod dataset, containing 5019 test images and 10553 training images. Many models use this dataset for training.

## 4.7 Metrics for Evaluation

Following evaluation measures have been used to measure the efficiency of proposed model with other models:

- ***Precision-Recall Curve*** is calculated by conversion of predicted saliency map into binarized map and ground truth. Thresholding of 0–255 is applied to produce binary map. For all saliency maps present in dataset, every binarizing threshold outcomes in a set of average precision and recall. An order of precision recall pair is generated when we vary threshold from 0 to 1, which is used to plot the PR Curve.
- ***F Measure Curve*** To provide a comprehensive analysis $F_{ß}$is calculated on both precision and recall as:

  $$F_{ß} = \frac{(1 + ß^2)Precision * Recall}{ß^2 Precision + Recall} \tag{4.1}$$

  $F^{ß}$ value is set as 0.3 to highlight precision more, this is because the rate of recall is not as significant as precision. Value of average f measure is also presented in this research. F measure curve is generated by comparison of binary map with ground truth that are obtained by changing threshold to decide if a pixel is owned by salient object or not.
- ***Mean Absolute Error*** MAE is used to correctly measure False negative pixels. It calculates pixel wise error between saliency map and Ground truth.

  $$MAE = mean(|S^M - G^T|) \tag{4.2}$$

If MAE value is less It indicates that ground truth $G^T$ and predicted saliency map $S^M$ are highly similar.

- ***Time*** (ins seconds) The average time taken by an image for execution (during testing)

- ***Area Under Curve*** it defines as the region under the ROC curve. A good model would earn 1 AUC, while random prediction will rate average about 0.5 AUC

## 4.8 Quantitative Evaluation

To determine the accuracy of the objects our model have segmented, we have tested our model SODL with six state of the art saliency detection models on five datasets. The deep saliency models include non-local features (NLDF) [70], contour to saliency (C2S) [71], Visual saliency detection based on multiscale deep CNN features (MDF) [72], Deep saliency with encoded low level distance map and high level features (ELD) [73], Salient Object Detection in Low Contrast Images via Global Convolution and Boundary Refinement [74] and Amulet: Aggregating multi-level convolutional features for salient object detection [75] and to show results we have plotted precision recall and f measure curves on each dataset. Additionally, we have also shown the results of our model SODL and in terms of MAE and weighted F measure.

### 4.8.1 PR and F Measure Curves for HKU-IS

Figure 4.8 shows PR Curves on HKU-IS dataset. The pairs of precision and recall are calculated by comparing the binary saliency maps with the ground truth to plot the PR curve, where the threshold for binarizing slides from 0 to 255. The closer the PR curve is to the upper right corner, the better the performance is. The plot of these values show that our model outperforms other models and our PR curves are significantly higher than those of other methods.

Figure 4.9 presents F measure on HKU-IS dataset, and it is clearly shown from f measure plot that our model also outperforms for HKU-IS dataset, we have calculated f measures on different thresholds varying from 0 to 1 and our model has shown much better results on various thresholds.

### 4.8.2 PR and F Measure Curves for DUT-OMRON

Figure 4.10 shows plot of PR Curve for DUT-OMRON dataset, which shows that our model SODL produce mush finer results and achieve greater precision for object segmentation. Our model has achieved mush higher precision for DUT-OMRON dataset.

Figure 4.11 shows F measure curve for DUT-OMRON where thresholds from 0 to 1 is applied and is plotted on X-axis and F measure values on different thresholds in plotted on Y-axis. Our model also outperforms for this dataset.

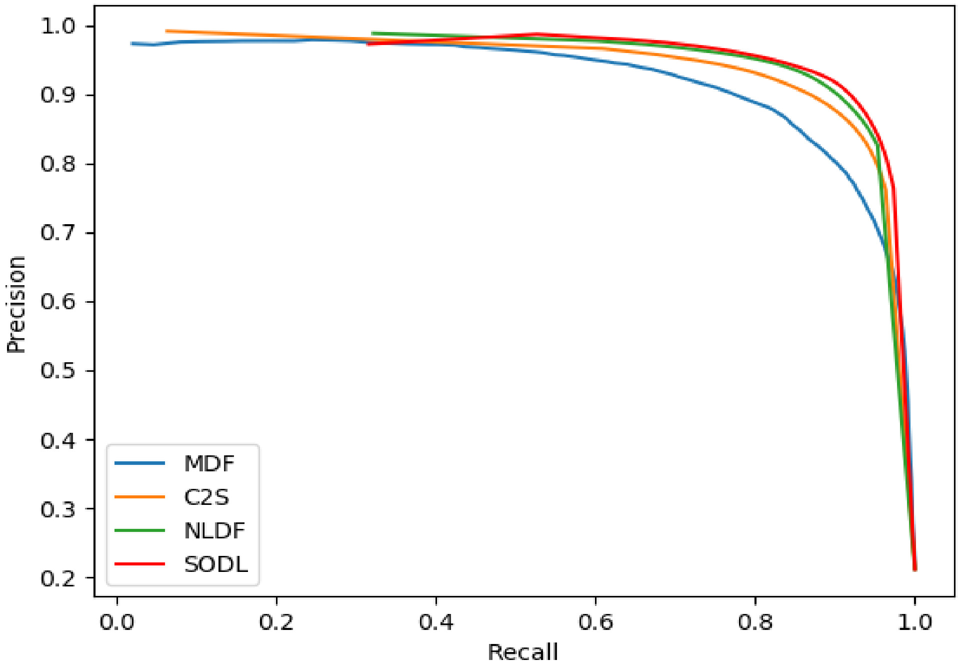


Figure 4.8: PR Curve for HKU-IS Dataset

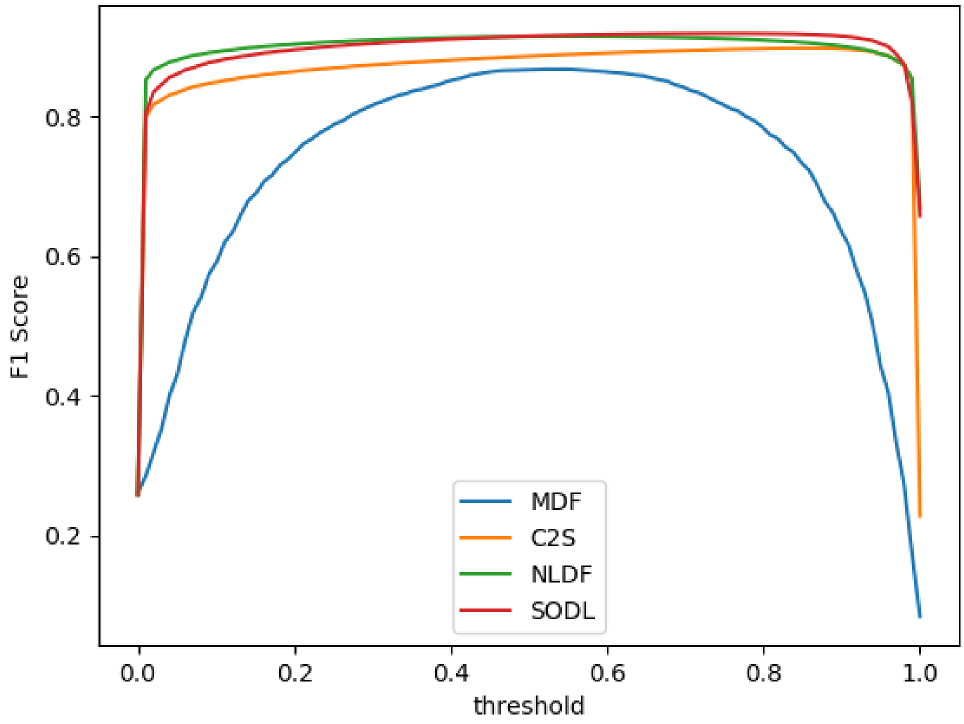


FIGURE 4.9: F measure Curve for HKU-IS Dataset

### 4.8.3 PR and F Measure Curves for PASCAL-S

Figure 4.12 shows PR Curve for PASCAL-S dataset and it is certain from the plot that our model has also outperformed for PASCAL-S dataset. Figure 4.13 shows the plot of F measure curve for PASCAL-S dataset.

### 4.8.4 PR and F Measure Curves for MSRA-B

Figure 4.14 shows plot of PR Curve for MSRA-B dataset and our model outperforms other models.

Figure 4.15 is a plot for F-measure Curve for MSRA-B dataset, for F measure our model also performs better as compare to others.

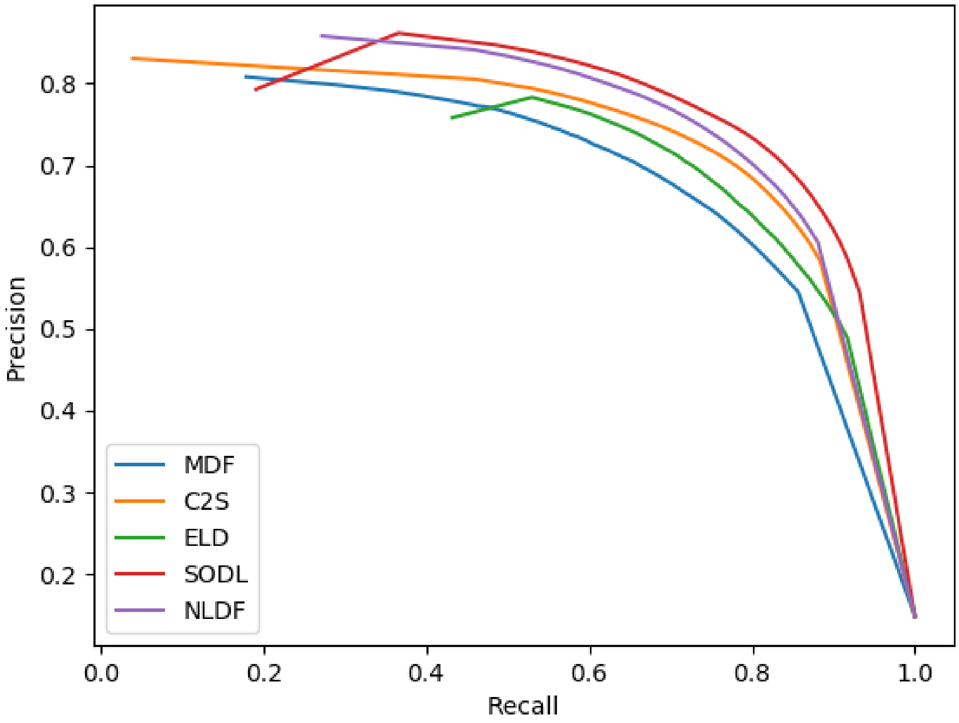


FIGURE 4.10: PR Curves for DUT-OMRON Dataset

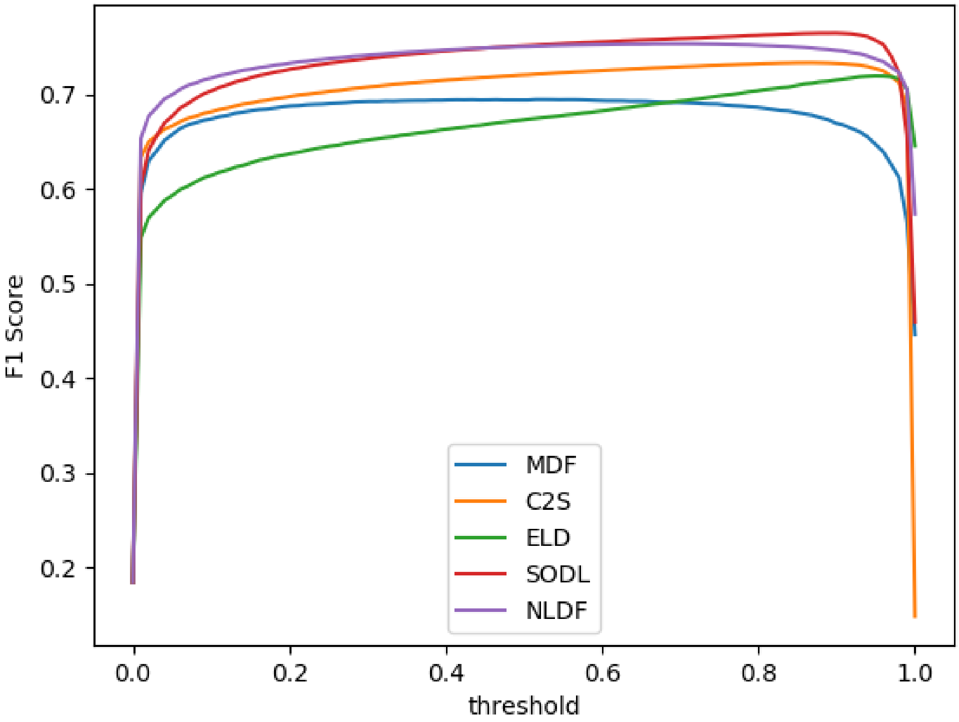


FIGURE 4.11: F Measure Curve for DUT-OMRON Dataset

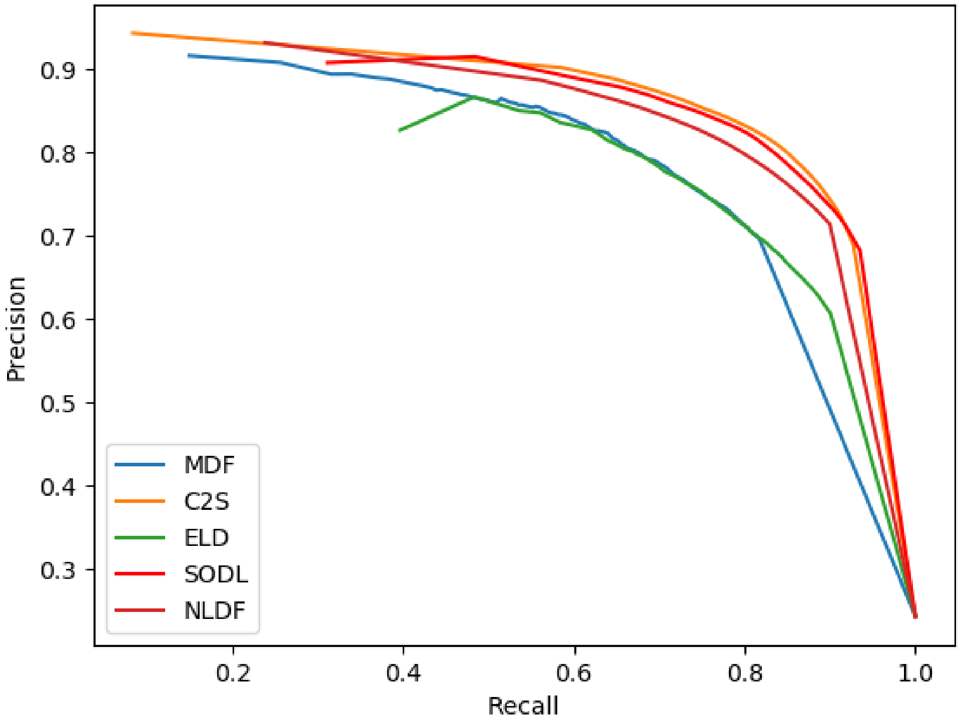


Figure 4.12: PR Curve for PASCAL-S

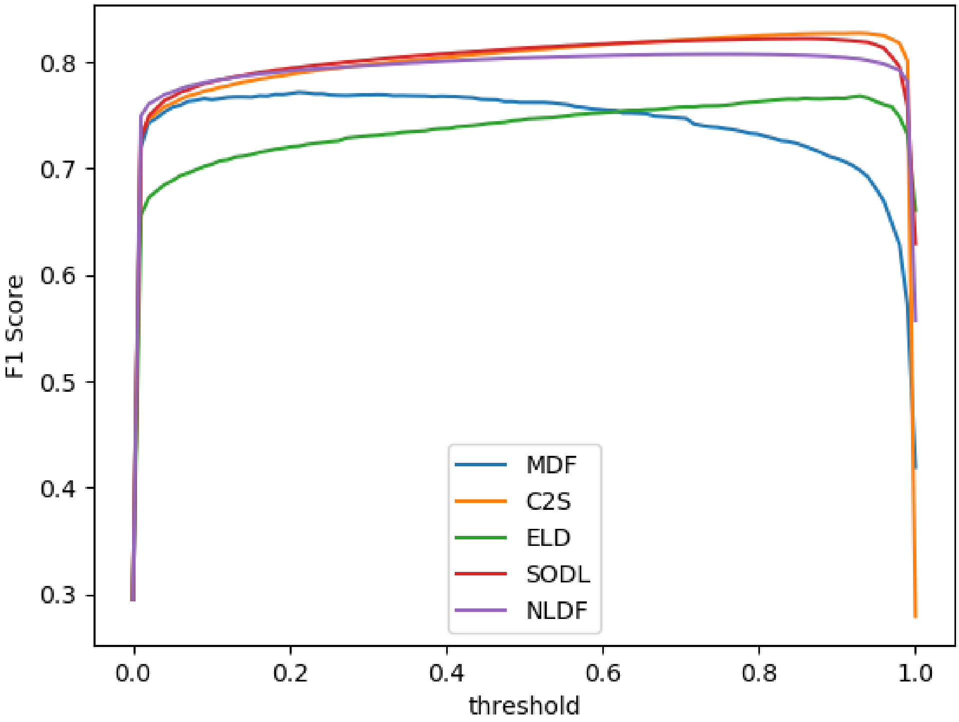


Figure 4.13: F Measure Curve for PASCAL-S Dataset

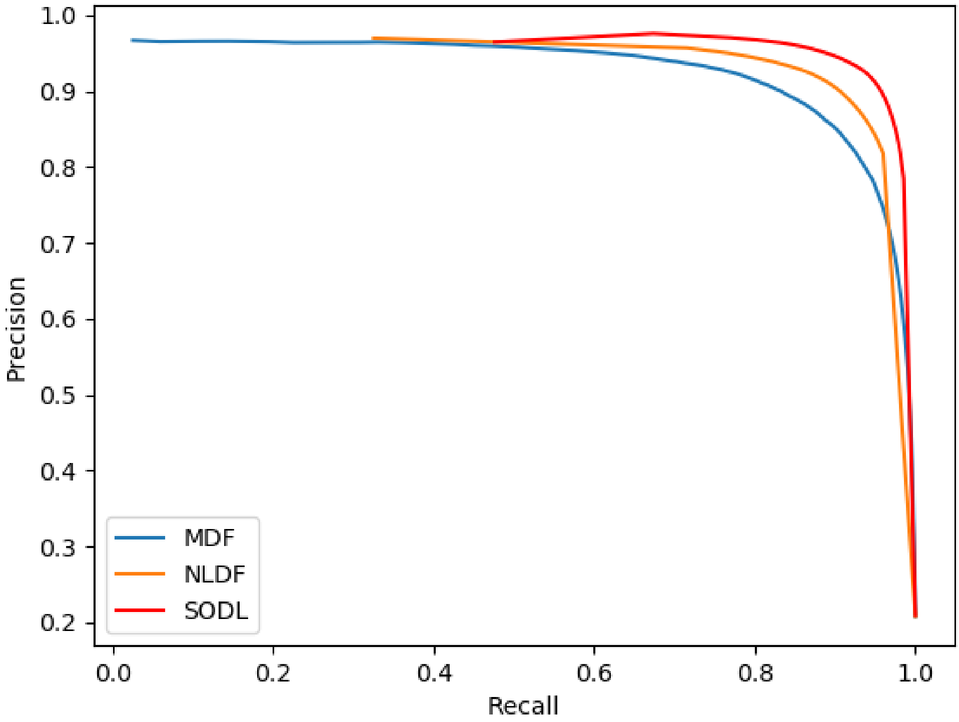


FIGURE 4.14: PR Curve for MSRA-B Dataset

### 4.8.5 PR and F Measure Curves for DUTS-TE

Figure 4.16 shows PR curve for DUTS dataset, it can be seen that our model has achieved greater precision and it's more accurate. Figure 4.17 shows F measure curve for both models.

Our approach achieves very good results on these datasets. As shown in Precision-Recall and F measure curves, our results are much flatter at most thresholds, which reflects that our prediction results are more uniform and consistent. Another quantitative comparison in term of weighted F measure and mean absolute error in Table 4.6 shows that proposed model has got 1st rank among all other models and F measure is improved by 2.5%, 1.5%, 1.1%, 0.2% and 2% for MSRA-B, DUTS, DUT-OMRON, PASCAL-S and HKU-IS datasets respectively.

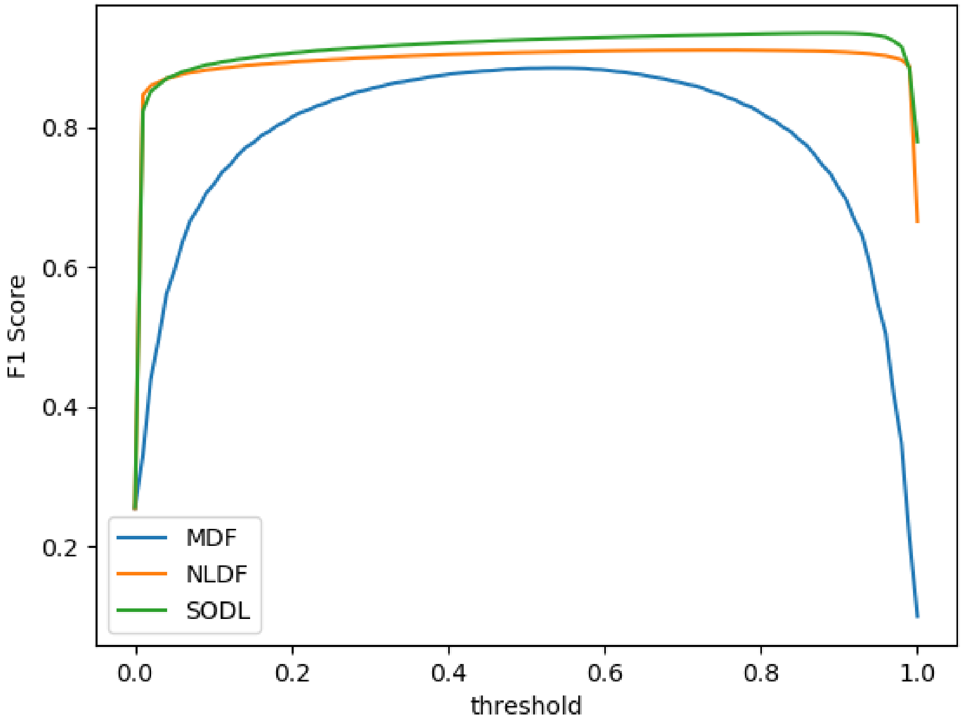


Figure 4.15: F Measure Curve for MSRA-B Dataset

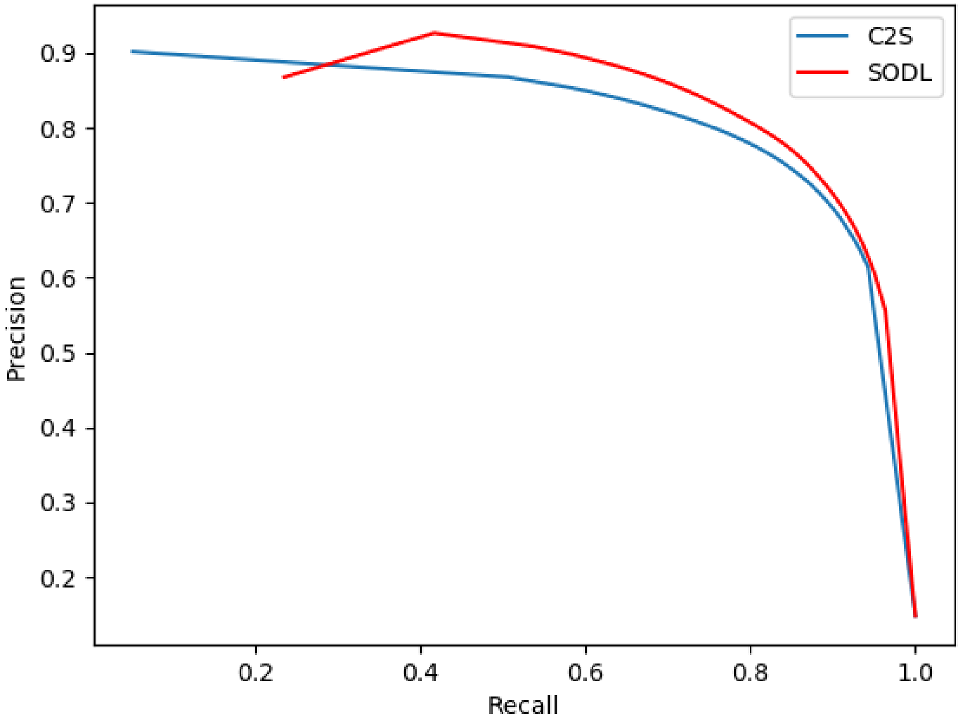


Figure 4.16: PR Curves for DUTS-TE Dataset

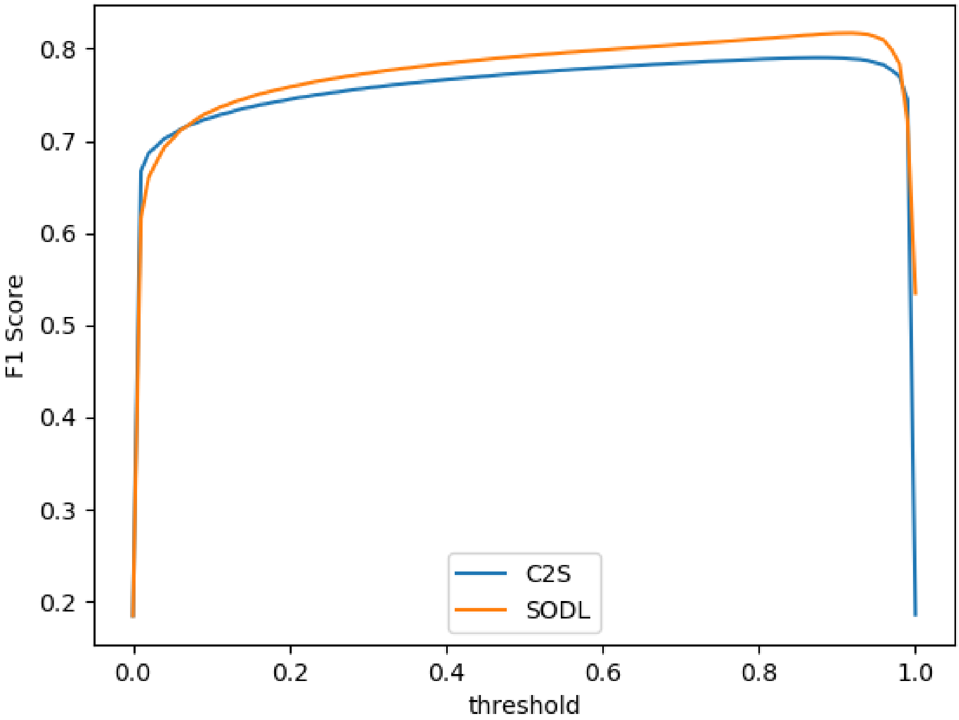


Figure 4.17: F Measure Curve for DUTS-TE Dataset

Table 4.6: Comparison of Proposed Model and Other Six Methods on Five Datasets in Terms of F-measure (larger the better) and Mean Absolute Error MAE (smaller the better). Proposed Model Ranked 1st on These Two Metrics

| **Dataset** | **Criteria** | **SODL (proposed)** | **NLDF [70]** | **CS2 [71]** | **MDF [72]** | **ELD [73]** | **GCBR [74]** | **Amulet [75]** |
|---|---|---|---|---|---|---|---|---|
| MSRA-B | WF | **0.935** | 0.910 | 0.830 | 0.885 | - | 0.8904 | - |
| | MAE | **0.035** | 0.0447 | 0.066 | 0.104 | - | 0.0373 | - |
| DUTS | WF | **0.816** | 0.812 | 0.790 | 0.730 | 0.738 | 0.801 | 0.773 |
| | MAE | **0.0653** | 0.066 | 0.066 | 0.094 | 0.093 | 0.0695 | 0.075 |
| DUT-OMRON | WF | **0.764** | 0.753 | 0.733 | 0.694 | 0.719 | 0.701 | 0.737 |
| | MAE | **0.0753** | 0.0795 | 0.079 | 0.091 | 0.090 | 0.0763 | 0.083 |
| PASCAL-S | WF | **0.829** | 0.807 | 0.827 | 0.771 | 0.768 | 0.801 | 0.826 |
| | MAE | 0.106 | 0.113 | 0.099 | 0.143 | 0.133 | **0.0356** | 0.092 |
| HKU-IS | WF | **0.918** | 0.90 | 0.897 | 0.867 | 0.839 | 0.8988 | 0.889 |
| | MAE | **0.0432** | 0.0485 | 0.052 | 0.135 | 0.074 | 0.0432 | 0.052 |

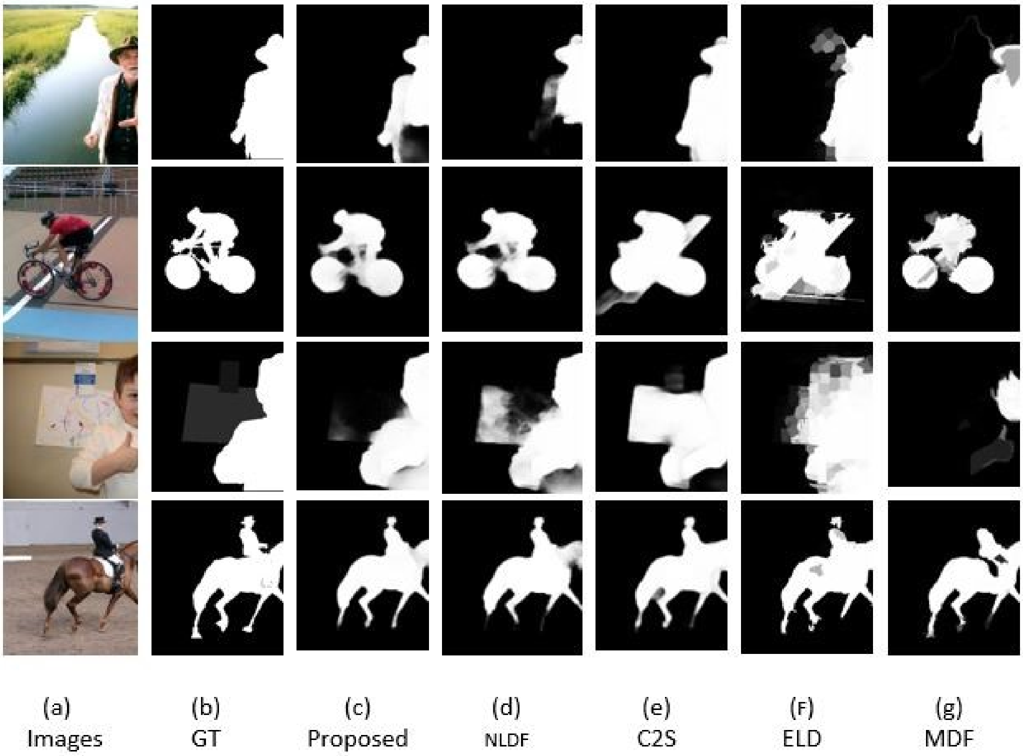
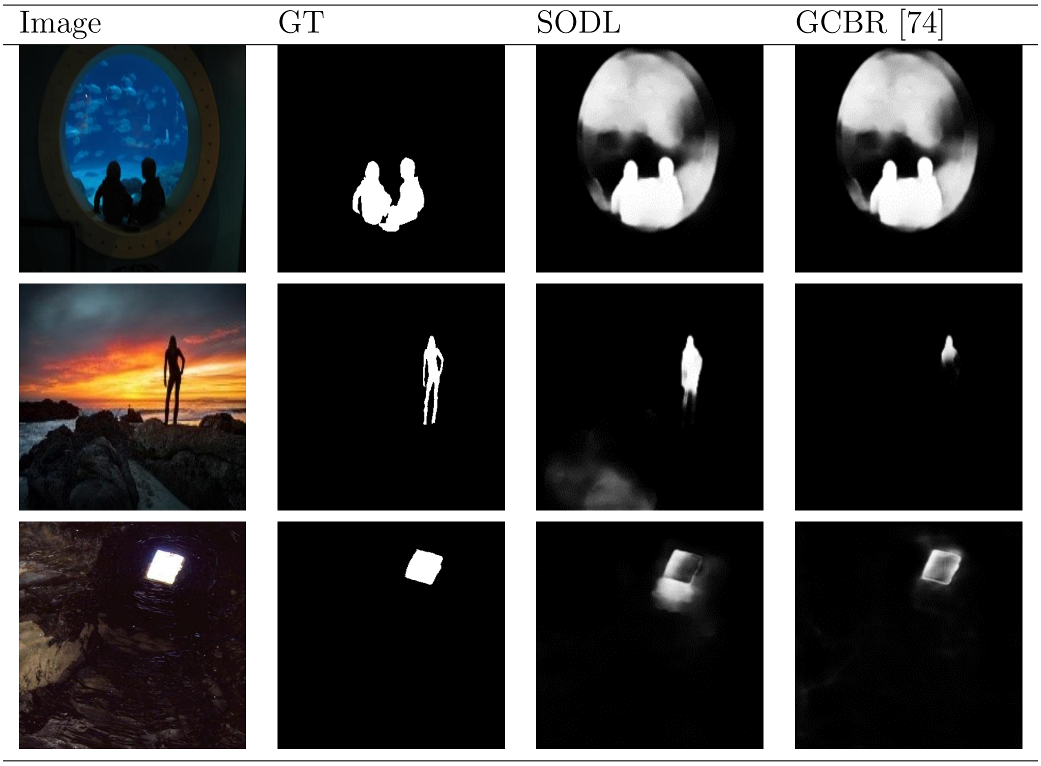


FIGURE 4.18: Visual Saliency Maps Generated by Our Method and Four Other Methods. Ours Achieves Best Results, Especially in Recovering The Spatial Details of The Salient Objects

## 4.9 Qualitative Evaluation

To further highlight the performance of proposed model and quality of saliency maps qualitative analysis of proposed model on five data sets is also performed. Figure 4.18 shows the quality of maps and segmented salient objects from five widely used datasets. Qualitative comparison of proposed model on five dataset shows that proposed model can segment object accurately under different challenging scenarios, containing images that have very less color difference from their background, objects that are large and touching image boundaries. Most strategies can produce good results for images which have simple scenarios, while the proposed model is able to achieve better results even in complex scenarios. Many deep learning models cannot segment and locate object in complex backgrounds, but proposed technique is successful in capturing of most salient regions.

Table 4.7: Salient Object Detection in Low Contrast Images



### 4.9.1 Results for Low Contrast Images

To show the performance of our proposed model on low contrast image we have taken low contrast images from 5 mentioned datasets and tested these images on our model and another state of the art model. Produced saliency maps are given in Table 4.7. Comparison results demonstrate the efficiency and robustness of proposed model in segmenting salient object from low contrast images.

Comparison results demonstrate the efficiency and robustness of proposed SOD technique in various complex scenes.

## 4.10 Ablation Study

In this part we show the effective of each component used in our model. For this we use DUT-OMRON dataset and test images by adding or removing components present in our model. First we have removed GCB from our model and then

Table 4.8: Results After Removing GCB

| | **Without GCB** | | | **With GCB** | | |
|---|---|---|---|---|---|---|
| | **MAE** | **Precision** | **Recall** | **MAE** | **Precision** | **Recall** |
| **Image 1** | 0.220 | 0.002 | 0.01 | 0.009 | 0.975 | 0.9775 |
| **Image 2** | 0.173 | 0.001 | 0.01 | 0.022 | 0.893 | 0.983 |
| **Image 3** | 0.237 | 0.002 | 0.01 | 0.01 | 0.964 | 0.969 |

we have tested our model, Table 4.8 shows the results and we can see a drastic changing in results after adding this block which validates the importance of this block in our model.

After that we have removed boundary refinement block from our model and tested similar images on model. Table 4.9 shows the results and Table 4.10 shows the saliency maps with BRB and without BRB and it is velar from their maps that this block refines segmentation.

Next we have removed our refinement block which refines whole saliency map and the results with or without refinement blocks are shown in Table 4.11 which shows

Table 4.9: Results After Removing BRB

| | **Without BRB** | | | **With BRB** | | |
|---|---|---|---|---|---|---|
| | **MAE** | **Precision** | **Recall** | **MAE** | **Precision** | **Recall** |
| **Image 1** | 0.041 | 0.884 | 0.981 | 0.009 | 0.975 | 0.9775 |
| **Image 2** | 0.062 | 0.760 | 0.987 | 0.022 | 0.893 | 0.983 |
| **Image 3** | 0.027 | 0.925 | 0.966 | 0.01 | 0.964 | 0.969 |

Table 4.10: Saliency Maps With and Without BRB

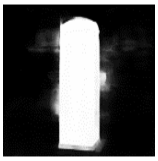
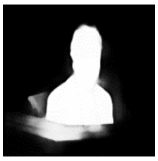
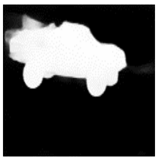
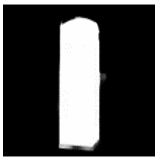
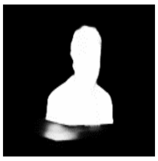
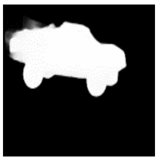
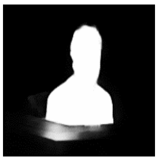
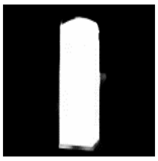
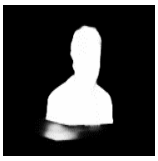
| **Without BRB** | | | **With BRB** | | |
|---|---|---|---|---|---|
| **Image 1** | **Image 2** | **Image 3** | **Image 1** | **Image 2** | **Image 3** |
|  |  |  |  |  |  |

Table 4.11: Results With or Without Refinement Block

| | **Without Refinement** | | | **With Refinement** | | |
|---|---|---|---|---|---|---|
| | **MAE** | **Precision** | **Recall** | **MAE** | **Precision** | **Recall** |
| **Image 1** | 0.015 | 0.956 | 0.98 | 0.009 | 0.975 | 0.9775 |
| **Image 2** | 0.033 | 0.857 | 0.99 | 0.022 | 0.893 | 0.983 |
| **Image 3** | 0.021 | 0.942 | 0.97 | 0.01 | 0.964 | 0.969 |

Table 4.12: Saliency Maps With and Without Refinement Block

| **Without BRB** | | | **With BRB** | | |
|---|---|---|---|---|---|
| **Image 1** | **Image 2** | **Image 3** | **Image 1** | **Image 2** | **Image 3** |
| |  | |  |  | |

the effectiveness of this block, it has improved the precision and MAE for the images which put large effect on the segmentation.

# Chapter 5

# Conclusion and Future Work

We summarize the research in this chapter by drawing the achievements of this research. It also highlights the areas for future work.

## 5.1 Conclusion

Everyday life has enormous amount of visual data and information which is accessible and generated every minute. The rise in image data has created new problems of extracting the correct information and fast processing to ease different task from searching in images to image compression and spreading of images over network.

In recent years it has been one particular problem of computer vision algorithms to find object of interest in images as its importance lies in many areas of medicine, robotics, graphics and computer vision.

Extensive research has been done on detection of salient object and produced impressive results as compare to traditional approaches but it is still a problem in case of low contrast images and fine segmentation near boundaries. Therefore, it is essential to develop a technique that is capable of fine detection of salient objects with accurate boundaries. This research presents a deep convolutional network with integration of local and global features with boundary refinement.

The combination of these features helps in accurate detection and segmentation of salient objects. The embedded Global convolutional block and Boundary refinement block helps in better feature extraction and preserves spatial location present in initial layers and are later refined by the refinement module, which results in more distinct features and accurate detection. Moreover, we examine the capability of proposed model on five large and widely used datasets. Experimental results indicate that proposed method outperform state of the art methods and has high capability for many computer vision tasks.

## 5.2 Future Work

Effective results produced by the integration of feature maps and refinement module motivated further enhancement in this work. We have interpreted some potential directions for future work. We can train the model by using some other feature extraction network. Furthermore, we can also make enhancement in loss function to increase the accuracy of model.

# Bibliography

[1] R. K. K. C. A. C. R. S. R. Z. Y. B. Kelvin Xu, Jimmy Ba, “Show, attend and tell: Neural image caption generation with visual attention,” 2015.

[2] S. G. H. Fang, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G.Zweig, “From captions to visual concepts and back,” pp. 1473–1482, 2015.

[3] M. N. A. B. N. Borji, Ali Ahmadabadi, “Cost-sensitive learning of top-down modulation for attentional control,” *Machine Vision and Applications*, vol. 22, pp. 1432–1769, 2011.

[4] A. Borji and L. Itti, “Scene classification with a sparse set of salient regions,” pp. 1902–1908, 2011.

[5] C. Kanan and G. Cottrell, “Robust classification of objects, faces, and flowers using natural image statistics,” pp. 2472–2479, 2010.

[6] C. Guo and L. Zhang, “A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression,” *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 185–198, 2010.

[7] W. Wang, J. Shen, R. Yang, and F. M. Porikli, “A unified spatiotemporal prior based on geodesic distance for video object segmentation,” 2017.

[8] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, “Pyramid dilated deeper convlstm for video salient object detection,” September 2018.

[9] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, “Object region mining with adversarial erasing: A simple classification to semantic

segmentation approach," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6488–6496, 2017.

[10] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, Y. Zhao, and S. Yan, "Stc: A simple to complex framework for weakly-supervised semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2314–2320, 2017.

[11] R. Zhao, W. Ouyang, and X. Wang, "Unsupervised salience learning for person re-identification," pp. 3586–3593, 2013.

[12] S. Bi, G. Li, and Y. Yu, "Person re-identification using multiple experts with random subspaces," 2014.

[13] D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little, and D. G. Lowe, "Curious george: An attentive semantic robot," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 503 – 511, 2008.

[14] W. Wang, J. Shen, and H. Ling, "A deep network solution for attention and aesthetics aware photo cropping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1531–1544, 2019.

[15] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," 2007.

[16] Jin Sun and H. Ling, "Scale and object aware image retargeting for thumbnail browsing," pp. 1511–1518, 2011.

[17] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li, "A user attention model for video summarization," 2002.

[18] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," pp. 1–8, 2008.

[19] A. Li, X. She, and Q. Sun, "Color image quality assessment combining saliency and fsim," 2013.

[20] H. Liu and I. Heynderickx, "Studying the added value of visual attention in objective image quality metrics based on eye movement data," pp. 3097–3100, 2009.

[21] R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk, "Salient region detection and segmentation," p. 66–75, 2008.

[22] F. Liu and M. Gleicher, "Region enhanced scale-invariant saliency detection," pp. 1477–1480, 2006.

[23] T. Liu, J. Sun, N. Zheng, X. Tang, and H. Shum, "Learning to detect a salient object," pp. 1–8, 2007.

[24] P. Khuwuthyakorn, A. Robles-kelly, and J. Zhou, "Object of interest detection by saliency learning," pp. 636–649, 2010.

[25] G. Lee, Y.-W. Tai, and J. Kim, "Deep saliency with encoded low level distance map and high level features," 2016.

[26] J. Kuen, Z. Wang, and G. Wang, "Recurrent attentional networks for saliency detection," pp. 3668–3677, 2016.

[27] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," pp. 1597–1604, 2009.

[28] Q. Zhang, J. Lin, Y. Tao, W. Li, and Y. Shi, "Salient object detection via color and texture cues," *Neurocomputing*, vol. 243, 03 2017.

[29] E. Rahtu and J. Heikkilä, "A simple and efficient saliency detector for background subtraction," pp. 1137–1144, 2009.

[30] J. Sokalski and T. P. Breckon, "Automatic salient object detection in uav imagery," 2010.

[31] M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2015.

[32] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 194–201, 2012.

[33] C. Yang, L. Zhang, H. Lu, X. Ruan, and M. Yang, "Saliency detection via graph-based manifold ranking," pp. 3166–3173, 2013.

[34] N. Imamoglu, W. Lin, and Y. Fang, "A saliency detection model using low-level features based on wavelet transform," *IEEE Transactions on Multimedia*, vol. 15, no. 1, pp. 96–105, 2013.

[35] W. Zou, Z. Liu, K. Kpalma, J. Ronsin, Y. Zhao, and N. Komodakis, "Unsupervised joint salient region detection and object segmentation," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3858–3873, 2015.

[36] N. Tong, H. Lu, X. Ruan, and M. Yang, "Salient object detection via bootstrap learning," pp. 1884–1892, 2015.

[37] G. Lee, Y.-W. Tai, and J. Kim, "Deep saliency with encoded low level distance map and high level features," 2016.

[38] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Salient object detection with recurrent fully convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1734–1746, 2019.

[39] V. K. Singh and N. Kumar, "Saliency bagging: a novel framework for robust salient object detection," *The Visual Computer*, vol. 36, pp. 1423–1441, 2019.

[40] Y. Liu, J. Han, Q. Zhang, and C. Shan, "Deep salient object detection with contextual information guidance," *IEEE Transactions on Image Processing*, vol. 29, pp. 360–374, 2020.

[41] M. Feng, H. Lu, and E. Ding, "Attentive feedback network for boundary-aware salient object detection," pp. 1623–1632, 2019.

[42] Y. Wang, X. Zhao, X. Hu, Y. Li, and K. Huang, "Focal boundary guided salient object detection," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2813–2824, 2019.

[43] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, “Multimodal deep learning for robust rgb-d object recognition,” pp. 681–687, 2015.

[44] L. Zhu, Y. Chen, A. Yuille, and W. Freeman, “Latent hierarchical structural learning for object detection,” pp. 1062–1069, 2010.

[45] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[46] X. Ding, Y. Luo, Q. Li, Y. J. Cheng, G. Cai, R. Munnoch, D. Xue, Q. Yu, X. Zheng, and B. Wang, “Prior knowledge-based deep learning method for indoor object recognition and application,” *Systems Science and Control Engineering*, vol. 6, pp. 249 – 257, 2018.

[47] Y. Zhang, X. Li, Z. Zhang, F. Wu, and L. Zhao, “Deep learning driven blockwise moving object detection with binary scene modeling,” *Neurocomputing*, vol. 168, pp. 454–463, 2015.

[48] Z.-Q. Zhao, P. Zheng, S. tao Xu, and X. Wu, “Object detection with deep learning: A review,” 2018.

[49] A. Borji, M.-M. Cheng, Q. Hou, H. Jiang, and J. Li, “Salient object detection: A survey,” *Computational Visual Media*, vol. 5, no. 2, p. 117–150, Jun 2019. [Online]. Available: http://dx.doi.org/10.1007/s41095-019-0149-9

[50] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, “Salient object detection in the deep learning era: An in-depth survey,” 2019.

[51] N. Mu, X. Xu, and X. Zhang, “Salient object detection in low contrast images via global convolution and boundary refinement,” pp. 743–751, 2019.

[52] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, “Basnet: Boundary-aware salient object detection,” pp. 7471–7481, 2019.

[53] J. Li, M. D. Levine, X. An, X. Xu, and H. He, "Visual saliency based on scale-space analysis in the frequency domain," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 996–1010, 2013.

[54] T. Liu, J. Sun, N. Zheng, X. Tang, and H. Shum, "Learning to detect a salient object," pp. 1–8, 2007.

[55] S. Alpert, M. Galun, A. Brandt, and R. Basri, "Image segmentation by probabilistic bottom-up aggregation and cue integration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 315–327, 2012.

[56] V. Movahedi and J. H. Elder, "Design and perceptual validation of performance measures for salient object segmentation," pp. 49–56, 2010.

[57] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," pp. 1155–1162, 2013.

[58] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," 2014.

[59] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," 2015.

[60] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, "Learning to detect salient objects with image-level supervision," pp. 3796–3805, 2017.

[61] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[62] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, and P. Jodoin, "Non-local deep features for salient object detection," pp. 6593–6601, 2017.

[63] S. Chen, X. Tan, B. Wang, and X. Hu, "Reverse attention for salient object detection," 2018.

[64] X. Li, F. Yang, H. Cheng, W. Liu, and D. Shen, "Contour knowledge transfer for salient object detection," 2018.

[65] Y. Zeng, H. Lu, L. Zhang, M. Feng, and A. Borji, "Learning to promote saliency detectors," pp. 1644–1653, 2018.