Performance Evaluation of Object Proposal Generators for Salient Object Detection

by

Sai Prajwal Kotamraju

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2018 by the
Graduate Supervisory Committee:

Lina Karam, Chair
Hongbin Yu
Suren Jayasuriya

ARIZONA STATE UNIVERSITY

May 2019

ABSTRACT

The detection and segmentation of objects appearing in a natural scene, often referred to as Object Detection, has gained a lot of interest in the computer vision field. Although most existing object detectors aim to detect all the objects in a given scene, it is important to evaluate whether these methods are capable of detecting the salient objects in the scene when constraining the number of proposals that can be generated due to constraints on timing or computations during execution. Salient objects are objects that tend to be more fixated by human subjects. The detection of salient objects is important in applications such as image collection browsing, image display on small devices, and perceptual compression.

This thesis proposes a novel evaluation framework that analyses the performance of popular existing object proposal generators in detecting the most salient objects. This work also shows that, by incorporating saliency constraints, the number of generated object proposals and thus the computational cost can be decreased significantly for a target true positive detection rate (TPR).

As part of the proposed framework, salient ground-truth masks are generated from the given original ground-truth masks for a given dataset. Given an object detection dataset, this work constructs salient object location ground-truth data, referred to here as salient ground-truth data for short, that only denotes the locations of salient objects. This is obtained by first computing a saliency map for the input image and then using it to assign a saliency score to each object in the image. Objects whose saliency scores are sufficiently high are referred to as salient objects. The detection rates are analyzed for existing object proposal generators with respect to the original ground-truth masks and the generated salient ground-truth masks.

As part of this work, a salient object detection database with salient ground-truth masks was constructed from the PASCAL VOC 2007 dataset. Not only does this

dataset aid in analyzing the performance of existing object detectors for salient object detection, but it also helps in the development of new object detection methods and evaluating their performance in terms of successful detection of salient objects.

In memory of my grandmother, Smt. Janaki

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere and heartfelt thanks to Dr. Lina J. Karam for giving me an opportunity to do M.S. thesis under her supervision. Her constant guidance and support have been of immense help to me in my research journey and I have learnt a lot from her intelligent insights and critiques. I am also very grateful to her for being patient with me and encouraging me during testing times. I also want to thank her for giving me the opportunity to be R.A. for the FALL'17 and SPRING'18 semesters as it has been one of the most enriching experiences of my life.

I would like to acknowledge the support provided by all the members of the Image, Video and Usability (IVU) lab. I would specially like to acknowledge the extremely valuable assistance provided by Tejas Borkar and Samuel F. Dodge. I would like to make a special mention of my close friends Rithin, Avinash, Ravi, and Anitha for their constant support and encouragement throughout my journey. I would also like to thank my ASU family Rakshith, Pratyusha, Navya, Keerthy, Sai Kiran, Kalyan, Karthik, and Nanda Kishore for standing with me during my tough times. Last but definitely not the least, I would like to express my utmost gratitude to my parents and my sister Srujal for their strong love and support throughout my journey. Without you guys, I wouldn't have come this far.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

This chapter presents the motivation behind the work in the thesis and briefly summarizes the contributions and organization of the thesis.

1.1 Motivation

Object detection has seen a great progress from the success of object proposal methods (e.g., [3]), which aim at generating region proposals to cover most of the observable objects in a given frame. A region proposal is typically described using a bounding box that encloses the detected object. The bounding box is represented by the location (coordinates) of one of its corners (typically top left corner) together with its length and height in pixels. A good object proposal generator is expected to efficiently generate as few bounding boxes as possible to reach a sufficiently high detection rate. Although object proposal generation is the primary focus of this thesis, it is part of a bigger problem which is object detection. Object detection involves both localization as well as classification of the objects in a given frame. While Edge Boxes [3] is solely an object proposal generation method based on structured decision forests, Faster-RCNN [4], and SSD [5] train on the ground-truth data of a given object detection dataset to learn the parameters required to generate object proposals and to subsequently classify them.

All the above methods achieve high recall at the cost of sampling a large number of candidate boxes, which prevents computationally expensive classifiers to be applied in the subsequent process of object detection. For example, Edge Boxes [3] requires one of its design parameters, α, to be high in order to have a better recall at more

Figure 1.1: Images from the PASCAL VOC 2007 Test Dataset [2] with Actual and Salient Ground-Truths Indicated by Green and Red Bounding Boxes, Respectively.

challenging IoU (Intersection over Union) thresholds. But, if α is increased, the density of the sampling is increased, resulting in more candidate boxes being evaluated and slower runtimes [3]. Hence, the detection of all the objects, in a given frame, at higher IoU thresholds requires a significantly much larger number of object proposals to be generated and is also computationally expensive.

One way to overcome this problem, while ensuring that the most salient objects in the scene are detected, is to design saliency-enhanced object proposal generation methods that are capable of detecting salient objects in the scene with a limited number of object proposals. Salient objects are those objects, in any given frame, that attract more visual attention or are more fixated by human subjects than the rest. For example, people focus most on the baby shown in Figure 1.1 as compared to other objects in this scene. In many applications, such as image display on small devices [15], and image collection browsing [16], it is enough to generate object proposals to detect salient objects in that frame and to subsequently classify them rather than aiming to detect all the objects.

1.2 Contributions

In this thesis, a novel saliency-enhanced evaluation framework is proposed to analyze the performance of object proposal generators in terms of their ability to successfully detect salient objects in the scene while restricting the number of object proposals that can be generated. For this purpose, as part of this work, a benchmark database with ground-truth data corresponding to the salient object locations is generated from the PASCAL VOC 2007 test dataset. Not only will such a database help in finding the best performing object proposal generators for the task of salient object detection, but it can also be used for the development and evaluation of newly introduced object proposal generators. Using the proposed evaluation framework and constructed dataset, the performance of popular existing object proposal generators and object detectors including Edge Boxes [3], Faster RCNN [4] and SSD [5] are analyzed for the task of salient object detection.

1.3 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 provides background material for different object proposal generation methods, the visual saliency models used to generate saliency maps, and performance evaluation measures. Chapter 3 describes the proposed framework and the constructed dataset. The performance evaluation results of popular existing object proposal generation and object detection techniques are also presented in Chapter 3 using the proposed saliency-based evaluation framework. Finally, in Chapter 4, the contributions of this work are summarized and directions for future work are outlined.

Chapter 2

BACKGROUND

This chapter provides background material that is needed to better understand the contributions of this work. It starts with a general background on image classification followed by a description of the state-of-art object detection methods evaluated in this thesis, in addition to select visual saliency methods and performance evaluation measures that are used in the proposed framework. Section 2.1 provides an introduction to convolutional neural networks and their applications. Section 2.2 describes the different object proposal generation methods which have been evaluated in this thesis for the task of salient object detection. Section 2.3 introduces various bottom-up saliency models which have been considered for saliency map generation in the proposed framework. Section 2.4 presents the performance evaluation measures, such as IoU and detection rate, that are used to analyze the performance of state-of-art object proposal generators for the task of salient object detection.

2.1 Introduction to Neural Networks and CNNs

Object recognition prior to the introduction of deep learning used to be a two-step process. First, necessary features are extracted from the image and then a classifier is trained with those features to recognize the object. Most of the variants of object recognition were based on the type of features and the classifiers used. Some of the most common features include histogram of oriented gradients (HOG) [17], scale invariant feature transform (SIFT) [18] and its variants, and bag-of-visual-word features [19] while the classifiers varied from naive-bayes [20], to Support Vector Machines (SVMs) [21], to logistic functions and many more.

Histogram of Oriented Gradients (HOG) was proposed by Dalal and Triggs in [17]. The idea was that the shape and the appearance of an object can be described by having a histogram of the intensity gradients. So, the HOG descriptor was formed by computing the histogram of edge orientations by dividing the image into smaller regions. The combination of all these histograms forms a descriptor for the image. This method was used for human detection in [17].

SIFT has been one of the most widely used feature descriptor for object recognition. This method by Lowe [18] turns images into a collection of local feature vectors which are invariant to translation, rotation and scaling. Keypoints are detected in the scale-space domain. Gradients and the dominant gradient are computed for each of these keypoints over a neighbourhood of 16 x 16 and used to form a keypoint descriptor.

SIFT produces multiple keypoints resulting in a relatively large number of keypoint features to train a classifier. To overcome this problem, a bag of words can be used to compress the features. For this purpose, in [22], the keypoint features undergo a pre-processing step before being used to train a classifier. All keypoint features are collected and a clustering algorithm such as K-means is applied. Once the clustering is complete, a histogram is computed for each image to indicate the number of features in each cluster. The classification step operates on the histogram produced post the clustering step. Standard classifiers like SVM [21], [20] are used to classify the images based on the obtained histogram.

SIFT, its variants, bag of words and HOG have all been very useful in object classification. However, there are a few major issues involved in using them. For example, SIFT would be a weak choice for classification of circular objects. Also, all the aforementioned methods tackle the object recognition problem as a two-step process of feature extraction and then classification. A more robust approach would

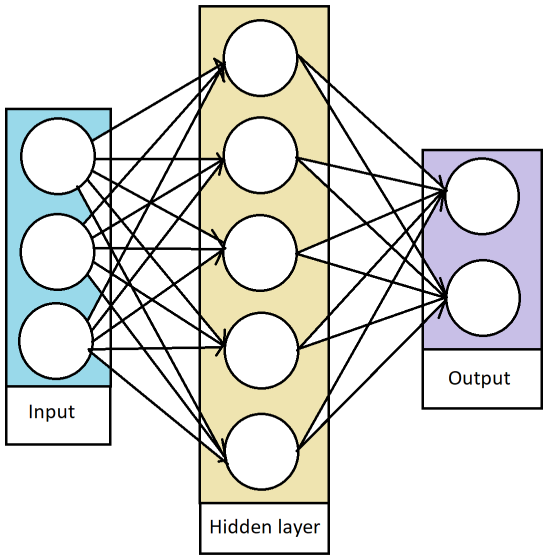

Figure 2.1: Typical Architecture of a Feed-Forward Neural Network with One Hidden Layer.

be using an end-to-end algorithm which trains both the feature extractor as well as the classifier simultaneously. This can be performed by training a neural network for example.

A Convolutional Neural Network (CNN) is an architectural extension of the feed-forward multi-layer perceptron neural network. A feed-forward neural network typically has an input layer, a few hidden layers, and an output layer, wherein the number of hidden layers and the number of units in each hidden layer are parameterizable as shown in Figure 2.1. Each unit (artificial neuron) in the hidden layer and the output layer behave like a biological neuron. The output of a single unit can be expressed as:

$$a = f(\mathbf{x}\theta) \tag{2.1}$$

where the function $f(x)$ is a non-linear function called Activation Function and $\mathbf{x}$ is an input row vector which is augmented with 1 for the bias term, and θ is a column vector representing the weights.

To train a neural network, the backpropagation algorithm [23] is used by altering the weights and biases of the neural network based on the gradient of cost function with respect to the weights and biases. Given a classification task, the input vector is propagated through the hidden layers to the output layer using Equation (2.1) for each neuron in each layer. When propagated to the output layer, the predicted class is compared to the actual class and the error obtained is backpropagated to the previous layers [7].

Although it is possible to reshape an image into a single-column vector and use it as input vector to the feed-forward neural network, such approach is not proven to be effective because of a lot of factors. Some of them include increase in number of weights resulting in increase in computational burden, and loss of spatial connectivity of data due to reshaping. In order to overcome this problem, Convolutional Neural Networks (CNNs) were introduced and were inspired by studies of the visual cortex. CNNs were also among one of the first networks to solve some of the most important commercial applications and remain at the forefront of deep learning solutions offered today [24]. In the 1990s, AT&T's neural network research group developed a CNN for check reading [23]. Soon by the end of the decade, this system was reading 10 percent of all the checks in the United States.

CNNs were some of the first working deep networks trained with back propagation [24]. Also because of their computationally efficient architecture compared to fully connected neural networks, it is easier to run multiple experiments with them and tune their implementation and hyper parameters. With the modern hardware and high power GPUs, large CNN architectures can be trained on very large datasets

and achieve results which are on par with human vision in object detection and segmentation tasks [25].

2.1.1 Visual Cortex Inspiration

Hubel and Wiesel conducted a series of experiments on the receptive fields of cats [26] and monkeys [27] by stimulating their retinas with spots and patterns of light. They found out the receptive fields defined by different cells. Simple cells have receptive fields that gave a distinct on and off areas separated by parallel straight lines for a stimulus of a spot of light. These cells were found out to be position dependent. Then there are complex cells which responded when a specific orientation of light shined on the field. These also had different responses to slits, edges and dark bars. Hypercomplex cells on the other hand have a different response such that the response drops off once the line gets off the activation area of the field. There have been cells with specific color response, cells lacking the orientation specificity, and cells with concentric fields.

Another interesting finding of these experiments is the architecture features of the visual cortex. Layering of the visual cortex involves aggregation of different cell types from simple cells, to complex cells, to lower order hypercomplex cells, to higher order complex cells.

2.1.2 Neocognitron

The study by Hubel and Wiesel became the basis for Fukushima's Neocognitron [28] consisting of a cascade connection of modular structures. These structures are made up of cascade of two structures - the S-cells and the C-cells which show the similarities to simple cells and complex cells, respectively. Each of these layers have multiple planes with different cells. The synapse for each receptive field in the S-cells

is modifiable and is subjected to change while training. Also, all the cells in each cell plane have the same distribution of input synapse.

Fukushima didn't employ the backpropagation procedure for training this network. Instead the network utilized an unsupervised competitive learning method using reinforcement for cell planes that were selected among representative planes.

2.1.3 LeNet5

The work of LeCun et al. [7] has been inspired mainly by the work of Fukishima [28]. LeNet5 has been the basis for the type of CNNs we see today. The main difference between the work of LeCun et al. [7] and Fukishima's work [28] is the use of the backpropagation algorithm. The work of LeCun et al. [7] is based on three main ideas of local receptive fields, shared weights, and spatial subsampling [7] to make sure that the features captured would be invariant to shift, scale and rotation. These three ideas have been realized by implementing an architecture, as shown in Figure 2.2, with alternate convolutional layer C and a subsampling layer S. The input image is convolved with three trainable filters and biases to generate three different feature maps at layer C1. These maps are then subsampled by grouping 4 pixels where they are added, weighted, and combined with a bias, followed by an activation function to generate 3 feature maps at S2. After this, another convolution operation takes place at C3 followed by a subsampling operation at S4. The resulting feature maps are reshaped into a single column vector that is fed to a conventional feed-forward neural network represented by the NN block in Figure 2.2. The convolution and subsampling processes are illustrated in Figure 2.3.

The CNN architecture is parameterized by the number of layers, depth or the number of maps generated per layer, size of the convolutional kernel, size of the subsampling window, and the stride sizes. In [7], the text recognition task was addressed

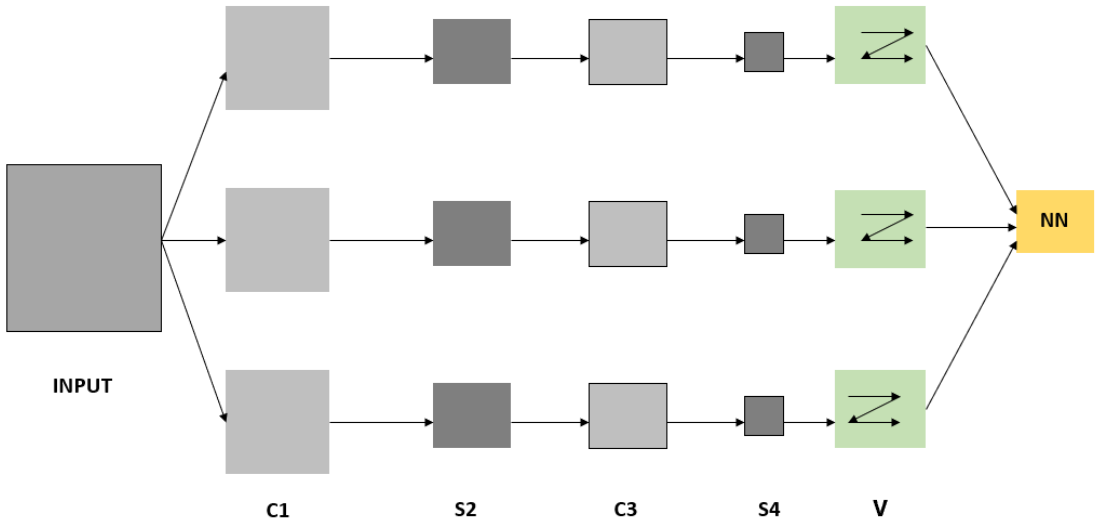

Figure 2.2: Conceptual Example of a CNN. C and S Refer to a Convolutional Layer and a Subsampling Layer, Respectively [6].

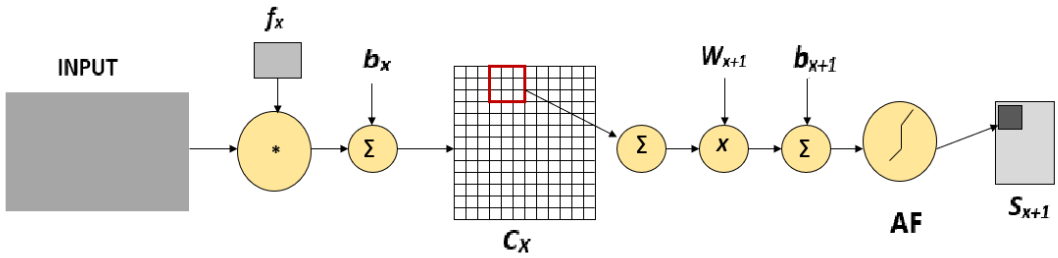

Figure 2.3: Convolutional Maps and Subsampling Details. AF Stands for the Activation Function.

with the architecture shown in Figure 2.4. The first layer is the input layer. The input image could either be a single-channel grayscale image or it could have 3 channels to accommodate color. In Figure 2.4, a grayscale image is used as input to the network. The second layer is composed of 6 feature maps. These feature maps are each the result of a convolution operation between the input layer and a kernel along with the addition of bias and the application of a non-linear function such as a sigmoid function as shown in Figure 2.3. The kernels are randomly initialized and later updated after

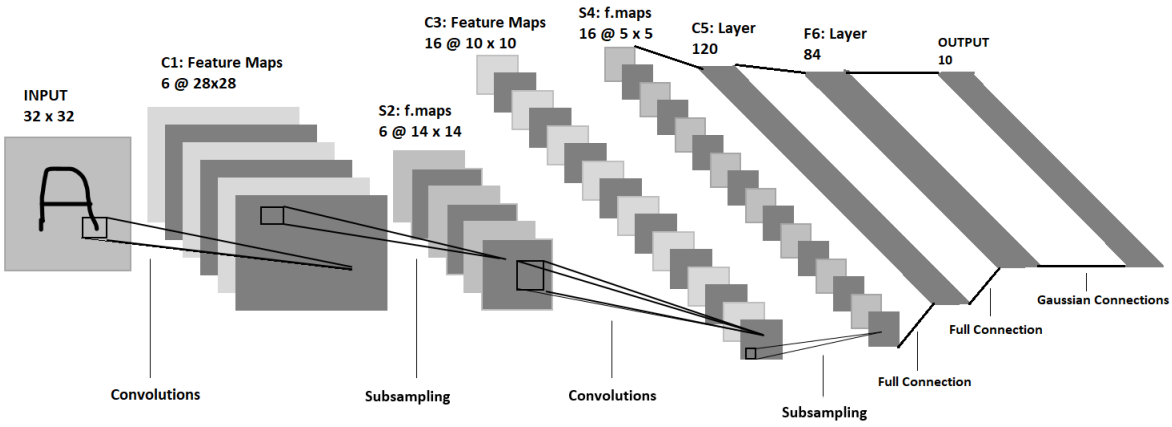

Figure 2.4: LeNet5 Architecture [7].

each pass of the backpropagation algorithm. Each kernel is different from one map to the next but the same kernel is used to generate a single feature map. This method of convolving with a kernel mimics the implementation for receptive fields and shared weights. After the convolutional layer, interest points or features are detected whose exact location is not needed. Thus, a subsampling layer, which computes the average around 4 pixels, is used. The average is then multiplied by a trainable coefficient, added with a bias term, and finally passed through an activation function.

The same combination of alternate convolution and subsampling is repeated for the next two layers with an increase in the number of feature maps. Finally, towards the end, a fully connected network is applied to have 10 output units predicting the probabilities of the image having corresponding numerical value. Obtaining the optimal weights for the kernels and biases is done using the backpropagation algorithm after every pass.

2.1.4 Variants

CNNs have been used in many applications and there have been variants in the general architecture. The first variant is the number of maps used at every convolution layer. This is sometimes referred to as the depth of the layer.

Another variant hardwires the values for some of the filters. For example, Kwolek's work on face detection [6] utilizes a 6-layer CNN which does not learn the convolution kernels for the first layer. Instead, the first layer is hardwired to be a Gabor filter so that the network could skip learning the basic edge detectors. Mutch and Lowe adopted a similar technique for object classification on the Caltech 101 image dataset [29]. The use of Gabor filters was perceptually motivated to mimic a primate's visual cortex.

One of the problems using CNNs is that they may require a large amount of training examples per class. In some applications, the number of available training images per class might not be sufficient. So, another variant pretrains the network with other types of images from a larger dataset like ImageNet [30]. Once the CNN model is pre-trained with images from ImageNet [30], it can then be fine-tuned [31] with the images from the task-specific dataset. This process aids in faster learning and it has been proven to be more effective than training the model from the scratch.

Finally, one of the problems with using CNNs is that, with the increase in the number of layers, the amount of time needed to train the network increases drastically. Usually the training time can be handled through the design of the network architecture, size of the filters, and the choice of connectivity and/or by using graphics processing units (GPUs).

2.1.5 *Applications*

CNNs have been used for various tasks. In this section, some of the applications and the datasets used are briefly described.

The MNIST digit dataset [32] has been one of the early datasets used to benchmark the performance of a CNN for object recognition. In the work of LeCun in [7], a 7-layer CNN was used to perform classification. In [33], the MNIST dataset was used to demonstrate best practices for CNNs and in [34], multi-column deep neural networks were used to classify the digits in the MNIST dataset using GPUs.

The work of Ciresan *et al.* in [35] dealt with the classification of German traffic signs. One contribution of this latter work apart from the classification of traffic signs is that it uses a fully parameterizable GPU implementation of the CNN. Convolutional layers could also be parameterized to skip convolution for some select units. A recognition rate of 99.15 percent, which is better than the human recognition rate of 98.98 percent, was reported. The work of Kang in [36] uses a convolutional neural network to classify documents. Popular CNNs that achieved top object classification performance on ImageNet [30] include AlexNet [37], VGG-16 and its variants [10], GoogleNet [25], ResNet18 and its variants [38].

2.2 Object Proposal Generation

The primary aim of an object detection system is to determine whether an object exists in a provided image and if so, where in the image it occurs. The dominant approach to this problem, for the past decade, has been the sliding window paradigm in which object classification is performed at every location and scale in that image [17, 39, 40]. But more recently, an alternative framework, referred to as object proposal generation, was proposed in which a set of object-bounding box proposals

are generated aiming to reduce the set of positions that need further analysis. This framework was adopted in the literature [3, 37, 41–47] and led to the discovery that object proposals can be accurately generated in a manner that is agnostic to the type of object being detected [3].

The field of object detection has seen a great progress from the success of these object proposal generation methods, which aim at generating an optimal number of region proposals to cover most of the observable objects in a given image or video frame. High recall and efficiency are important for an object proposal generator, i.e., an effective object proposal generator should be able to obtain a high detection rate using a relatively modest number of candidate bounding boxes.

Object proposal generation is a subset of a bigger problem called object detection in which the system is supposed to localize as well as classify the objects in the input frame. While Edge Boxes [3] is solely an object proposal generation method based on structured decision forests, Faster-RCNN [4], and SSD [5] train on the Ground-Truths of a given dataset to learn the parameters required to generate object proposals and classify them for new images. This section briefly describes these three object proposal generation methods [3–5] which are evaluated later in this thesis using our proposed evaluation framework.

2.2.1 Edge Boxes

The Edge Boxes method [3] uses edges in a given image to generate the candidate boxes. Given an image, an edge response for each pixel is computed using the Structured Edge Detector [48], [49] which has a good performance in predicting object boundaries efficiently. The single-scale variant with the sharpening enhancement [49] was utilized in Edge Boxes in order to reduce the computation time. Given the dense edge responses, Non-Maximal Suppression (NMS) is performed orthogonal to

the edge response to find the edge peaks. This results in a sparse edge map, with each pixel p having an edge magnitude m_p and orientation θ_p [3]. Edges, according to [3], are the pixels with $m_p > 0.1$ while the contour is defined as a set of edges forming a coherent boundary, curve, or a line.

When searching for the object proposals, it is important to consider the object classification algorithm applied to these proposals. Some of these may require object proposals with high accuracy while the others might be more tolerant to the errors in bounding box placement. The accuracy of these proposals is typically measured using the IoU metric described in Section 2.4.1. The IoU metric involves computing the area of intersection of the considered candidate box with a ground-truth box and dividing it with the area of their union. When evaluating an object detection algorithm, an IoU threshold of 0.5 is typically used to determine if the detection is correct.

Candidate bounding boxes are searched using a sliding window over position, scale, and aspect ratio. The step size for each is determined such that one step size leads to an IoU of α between the neighbouring boxes. The scale values range from a minimum box area of 1000 pixels to the full image. Aspect ratio varies from $1/\tau$ to τ where the value of $\tau = 3$ has been used in the implementation. Typical value of α ranges from 0.5 to 0.85. An increase in α increases the density of sampling thereby increasing the number of proposals. This leads to higher detection rates at the cost of evaluating a large number of candidate boxes.The likelihood of the candidate box is based on the number of contours that are wholly contained in it. This approach uses a simple box objectness score that measures the difference between the number of edges that exist in the box and those that are members of contours that overlap the box's boundary.

2.2.2 *Faster-RCNN*

Object proposal generation, which has been the focus element in this thesis, is a subset of a larger problem of object detection where the task involves classification of generated object proposals. In contrast with an image classification problem, the object detection problem poses a need to identify a variable number of objects in an image.

Out of the pool of proposed solutions, two classical approaches have been used extensively. The first one is the Viola-Jones framework [39] which is relatively fast and simple, and its algorithm aided in point-and-shoot cameras that implement real-time face detection with little processing power. This approach works by generating thousands of simple binary classifiers using Haar features. These classifiers are assessed with a multi-scale sliding window in cascade and are dropped early in case of a negative classification. Another traditional approach [17] uses Histogram of Oriented Gradients (HOG) features and Support Vector Machine (SVM) for classification. Although this method is superior to Viola-Jones, it is much slower.

After the evolution of deep learning techniques that are backed up by high performance computational resources, deep learning models outperformed traditional models with respect to image classification and object detection. One of the first advances in this area was OverFeat [50] which proposed a multi-scale sliding window algorithm using Convolutional Neural Networks (CNNs). Just after OverFeat, the Regions with CNN features or R-CNN method [8] was published and involved a region proposal method [41] to extract possible objects followed by computing CNN-based features, for each region, and using the features to perform classification using SVMs (Figure 2.5). While this achieved significantly better results when compared to the OverFeat approach [50], the computational time was still high because of CNN feature extrac-

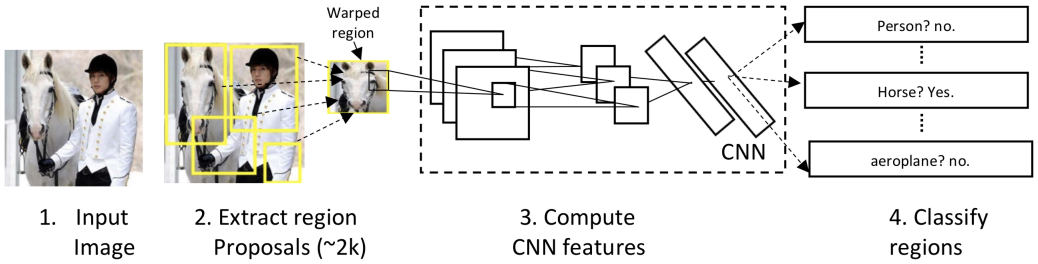

Figure 2.5: Regions With CNN Features (R-CNN) [8].

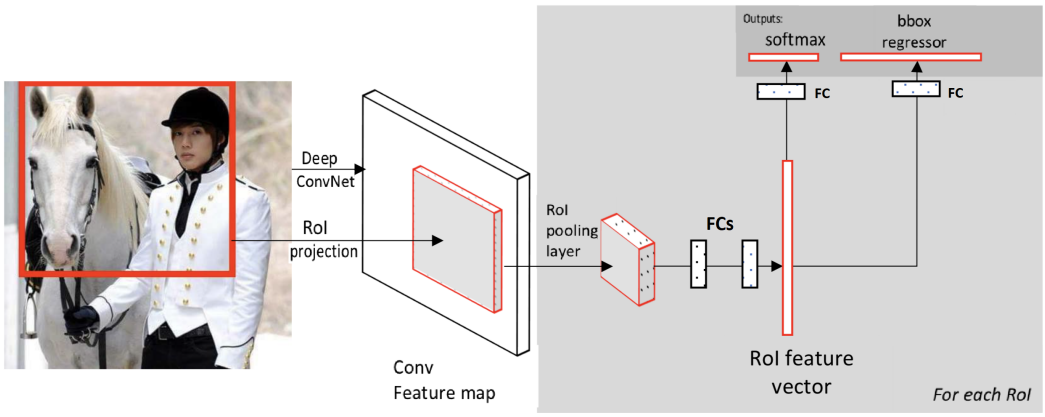

Figure 2.6: Fast R-CNN [9] Architecture.

tion for each and every predicted region. A few years later, Fast RCNN [9] adopted an approach similar to R-CNN, by using region proposals to identify prospective object locations, but applied the CNN on the whole image and used ROI pooling on feature maps with a final feed-forward network for classification and regression (Figure 2.6). The biggest problem of this framework was the use of an object proposal generator which became a bottleneck in computational time.

Subsequently, Faster-RCNN [4], the third instalment of the R-CNN series achieved a better performance than Fast R-CNN [9] by eliminating the use of an external object proposal generator and introducing a CNN based Region Proposal Network (RPN)

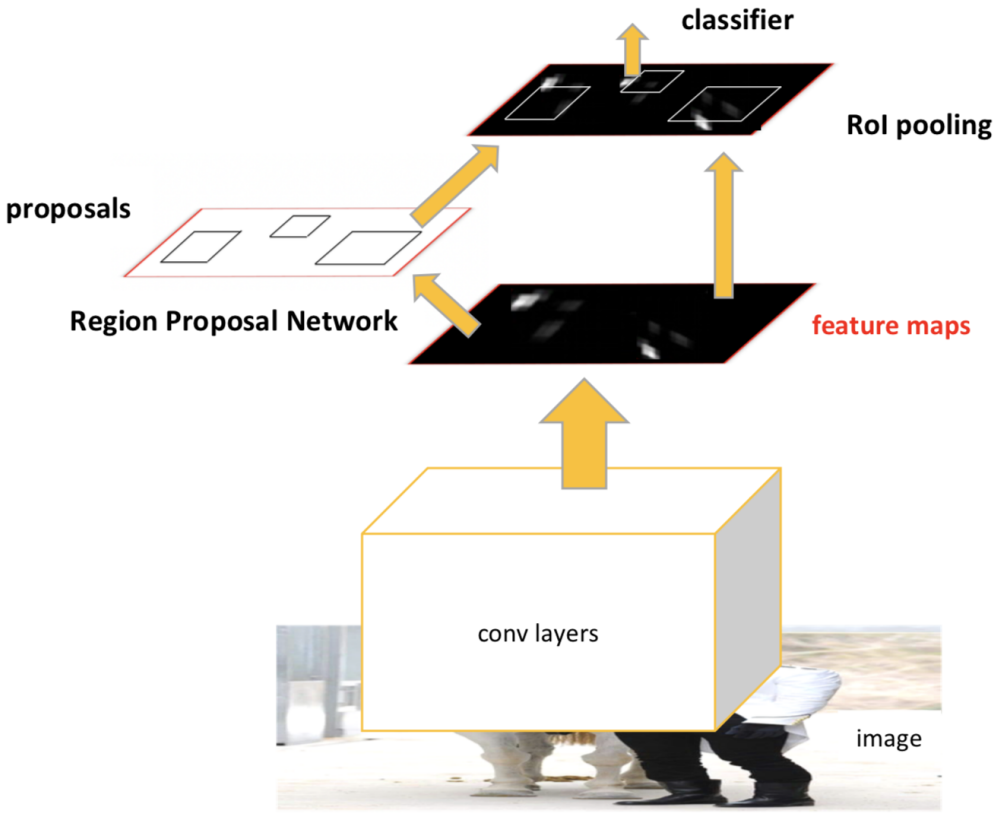

Figure 2.7: Faster R-CNN [4] Architecture.

to make the model trainable end-to-end. RPNs output object proposals based on an objectness score and the resulting object proposals are further processed by the ROI pooling and fully connected layers stages for classification. Figure 2.7 illustrates the architecture of Faster-RCNN. The input images are represented as $Height$ x $Width$ x $Depth$ tensors and are passed through a pre-trained CNN which acts as a feature extractor, until an intermediate layer generating convolutional feature maps. This technique is very commonly used in the context of transfer learning to train a

classifier on a relatively small dataset using feature maps that are obtained from a CNN that has been pre-trained on a larger dataset like ImageNet [30].

Using the features computed by CNN, an RPN predicts a predefined number of candidate boxes which may contain objects (others are designated as background). In order to tackle the problem of variable-length list of bounding boxes, RPN uses anchors: fixed size reference bounding boxes that are placed uniformly throughout the original image as shown in Figure 2.8. Instead of detecting where the objects are, the problem is modelled into two parts. For every anchor, the two following questions are posed: (1) Does the anchor contain a relevant object? (2) How much does it have to be adjusted to better fit the relevant object? After localizing the relevant objects, RoI Pooling is applied and the features corresponding to the relevant objects are extracted into a new tensor. In the final step, the R-CNN module classifies the content in the bounding box and adjusts the bounding box coordinates if it is not labelled as background.

In this thesis, the primary focus is on the object localization part of Faster R-CNN instead of the whole object detection framework. A pre-trained VGG based Faster R-CNN architecture [4, 51], trained on the PASCAL VOC 2007 training dataset, is used to predict the bounding boxes corresponding to objects in the PASCAL VOC 2007 test dataset.

2.2.3 Single-Shot Multibox Detector

Although every subsequent R-CNN version has advantages over its predecessors, these versions have a few collective disadvantages which motivated for research on new object detection frameworks. Some of those problems are as follows [52]: (1) the training process is unwieldy and takes a lot of time; (2) multiple phases are involved in the training process (e.g., training the RPN and the classifier in multiple steps in

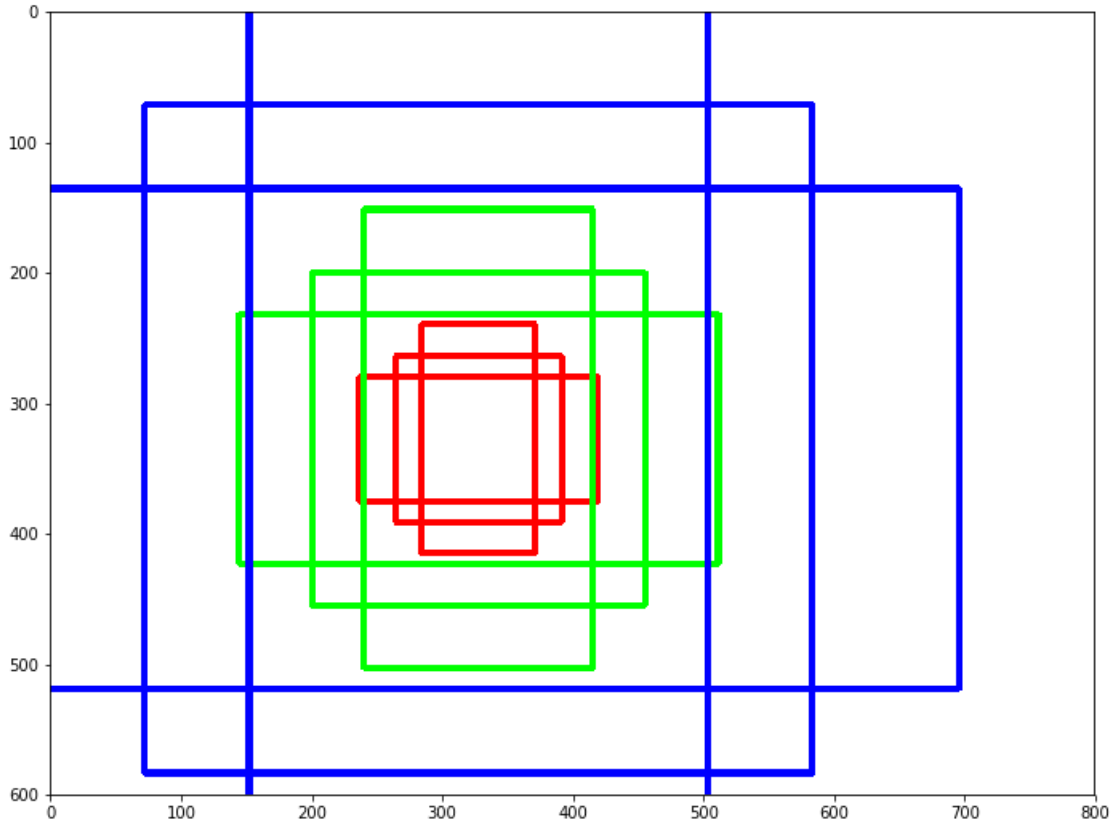

Figure 2.8: Anchors at (320,320) for a 600×800 Image.

the case of Faster R-CNN); (3) the network is slow at inference/test time. The speed at test time is a major concern as none of the aforementioned techniques managed to create a real-time object detector.

Single Shot Multibox Detector (SSD) [5] was released at the end of November 2016 and has successfully created new records in terms of performance and precision for object detection tasks with a Mean Average Precision (mAP) of 74 percent and operating speed of 59 frames per second on standard datasets like Pascal VOC [2] and COCO [53]. SSD executes the tasks of object localization and classification in a single forward pass of the network by utilizing the MultiBox technique [54] for bounding box regression.

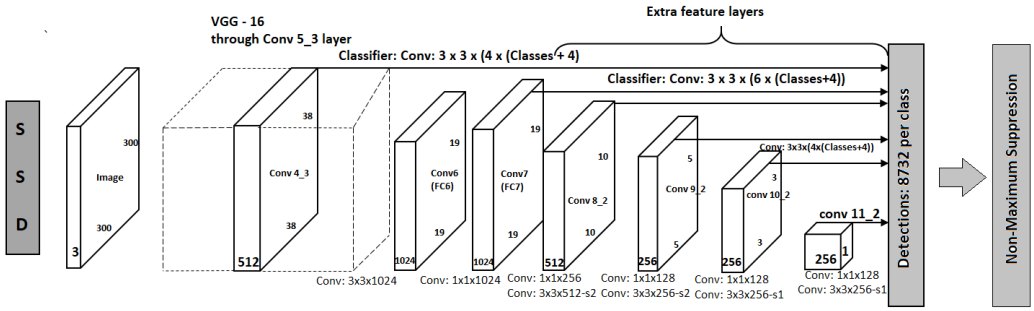

Figure 2.9: SSD Framework [5].

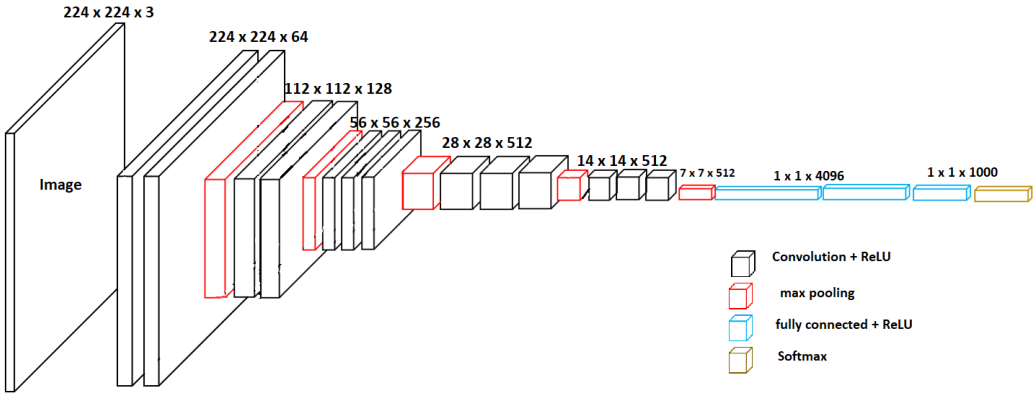

Figure 2.10: VGG-16 Architecture [10].

Figure 2.9 shows the SSD framework. It can be seen that the SSD architecture builds on the popular VGG-16 architecture [10] (Figure 2.10) but discards the final fully connected layers. VGG-16 [10] has been used as the base network because of its strong performance in high quality image classification tasks and also because of the ease in transfer learning to improve results. Instead of the original VGG fully connected layers (Figure 2.10), a set of auxiliary convolutional layers (conv6 onward) were added to enable extraction of features at multiple scales and progressive decrease in the size of the input to subsequent layers.

The bounding box regression technique of SSD is inspired from MultiBox [54], a method known to predict fast class-agnostic box coordinate proposals. In [54], an

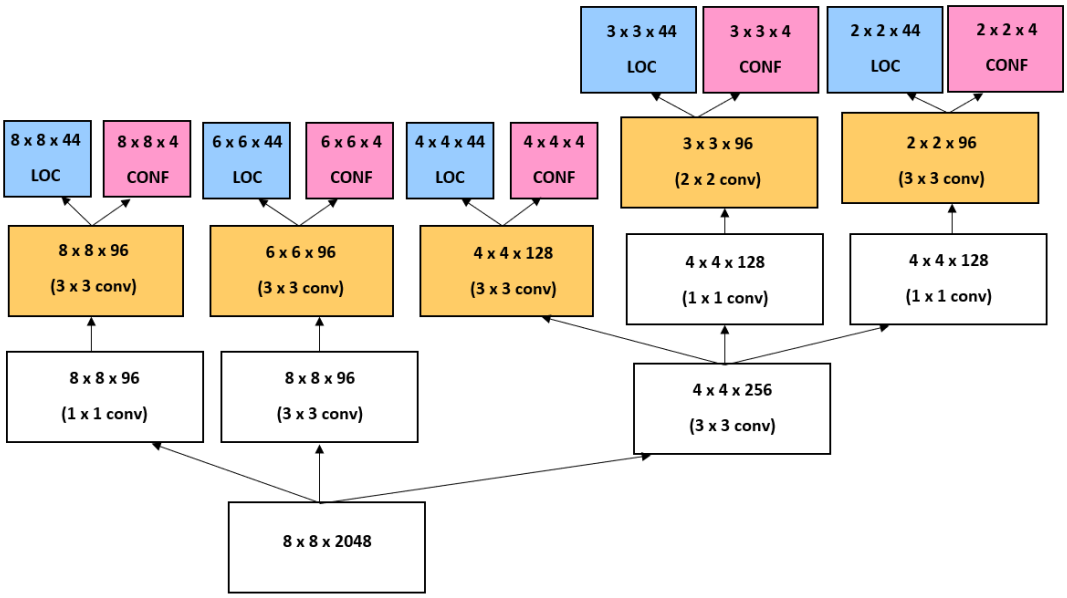

Figure 2.11: An Illustration of the Multi-Scale Convolutional Prediction of the Locations and Confidences for Multibox.

Inception-style convolutional network has been used as shown in Figure 2.11. The MultiBoxs loss function combined two critical components which have been incorporated into the SSD architecture: (1) confidence loss: measures how confident the network is about the objectness of the computed bounding box; the loss is computed using the categorical cross-entropy function in this case; (2) location loss: measures how far are the ground-truths from the bounding boxes predicted by the network; the L_2 norm is used for this purpose. Similar to the anchors in Faster R-CNN, mentioned in the previous sub-section, MultiBox regression utilizes anchors such that the IoU between consecutive anchors is greater than 0.5.

In the SSD architecture, a few changes were made in order to make the network even more capable of localizing and classifying the objects. Unlike [54], every feature map cell is associated with a set of manually chosen default bounding boxes of different dimensions and aspect ratios. Also, SSD uses the L_1 norm to calculate the localization

loss instead of the L_2 norm used in [54]. Finally, apart from MultiBox's object proposal generation, SSD has an ability to classify the predicted objects just like Faster-RCNN [4]. Although SSD tackles the whole problem of object detection, this thesis aims only at the evaluation of SSD as an object proposal generator and does not delve into the classification part of it.

2.3 Visual Attention Models

A large number of visual attention models compute the saliency of a pixel giving a measure of how much that pixel stands out from its surroundings and, as a result, produce a 2D topological map, called saliency map, which gives the relative importance of each pixel in the given image [1]. Lots of studies [55, 56] exploit the idea of a two-component framework for explaining how attention is deployed. Per this framework, visual attention (VA) mechanisms can be classified into bottom-up and top-down components.

Bottom-up attention usually occurs in the pre-attention stage and is highly influenced by center-surround operations on basic features extracted in pre-attentive stage like colour, orientation, motion, to name a few, while the top-down component is highly task-dependent. In general, the top-down component is not totally independent of the bottom-up component, and the VA mechanism is considered to be the result of an interplay of both these components [1].

Most of the VA models developed over the past decade have been targeted at modelling the bottom-up component of visual attention because of the top-down component being highly task-dependent. Also, in general, modelling the top-down component of VA requires some sort of supervised learning which, in turn, requires massive amounts of data to train. Hence, in this thesis, VA models focusing on the bottom-up approach are chosen to generate the saliency maps which are further used

to generate salient ground-truths. The top four bottom-up saliency models, as per the subjective experiments from [1], are introduced in the following sub-sections.

2.3.1 Fast and Efficient Saliency

In this model, proposed by Tavakoli et al. [11], a Bayesian framework based center-surround approach is adopted. Salieny at a point, for a given image, is considered to be a binary random variable having the value 1 if the point is salient, and 0 otherwise. The probability of a pixel being salient given the feature values is considered as the saliency.

For an image I, each pixel is defined as $x = (\bar{x}, f)$ where $\bar{x}$ represents the coordinate of the pixel x in image I, and f is the feature vector for that coordinate. f can be a colour vector or any other desired feature like SIFT, Gabor, LBP, LSK etc. Assuming there exists a binary random variable H_x that defines pixel saliency, it can be defined as follows:

$$H_x = \begin{cases} 1 & \text{if } x \text{ is salient} \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

The saliency of pixel x can then by computed using $P(H_x = 1|f) = P(1|f)$. From Bayes rule, it can be expanded as follows:

$$P(1|f, \bar{x}) = \frac{P(f|\bar{x}, 1)P(1|\bar{x})}{P(f|\bar{x}, 1)P(1|\bar{x}) + P(f|\bar{x}, 0)P(0|\bar{x})} \tag{2.3}$$

The FES method [11] adapts a kernel density approximation method to compute the feature distribution and estimate $P(f|\bar{x}, 1)$ and $P(f|\bar{x}, 0)$. It utilizes a multi-scale approach where saliency of a centre pixel is computed at different scales by changing the radius and number of samples involved and taking their average.

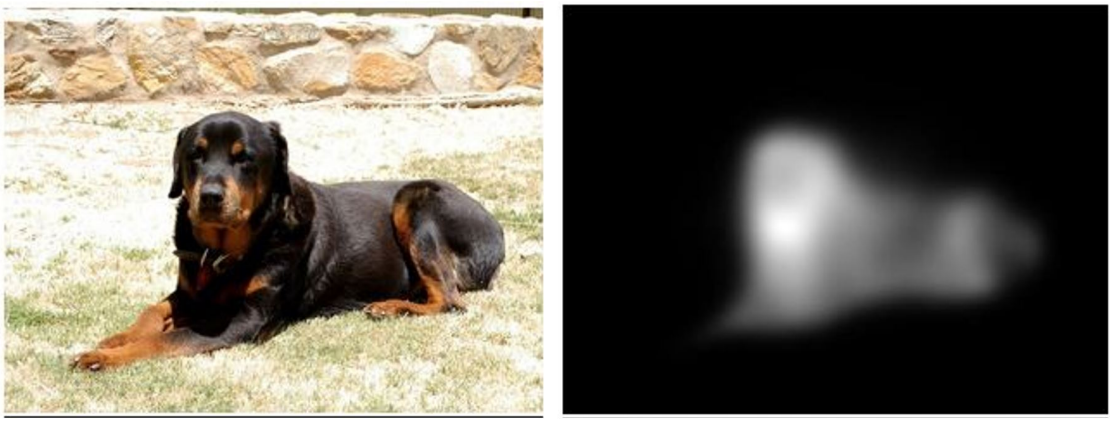

Figure 2.12: FES Saliency Illustration. (Left) Sample Image from the PASCAL VOC 2007 Test Dataset. (Right) Saliency Map Generated Using the FES Saliency Method.

Figure 2.12 shows the saliency map generated using FES [11] for a sample image from the PASCAL VOC 2007 test dataset [2]. The higher the value of the pixel in the saliency map, the higher the saliency of that pixel.

2.3.2 Boolean Map Saliency

In the Boolean Map Saliency (BMS) approach proposed by Zhang and Scarloff [13], the Gestalt principle of surroundedness is used to compute the saliency. In this method, colour-based feature maps are thresholded by varying the thresholds in order to obtain Boolean maps. These maps are in turn used to obtain connected regions. A binary value of 1 is assigned to the regions which exhibit surroundedness or are closed, and a value of 0 is assigned to the rest. Maps for a given threshold are normalized using the L_2 norm and then the normalized maps are averaged over different thresholds and different features to get the final saliency map.

For a given image I, a set of Boolean maps $B = B_1, B_2, B_3, B_n$ are generated by thresholding with various threshold values. The influence of a Boolean map B_i on visual attention is represented by an attention map $A_i(B_i)$, which highlights regions in

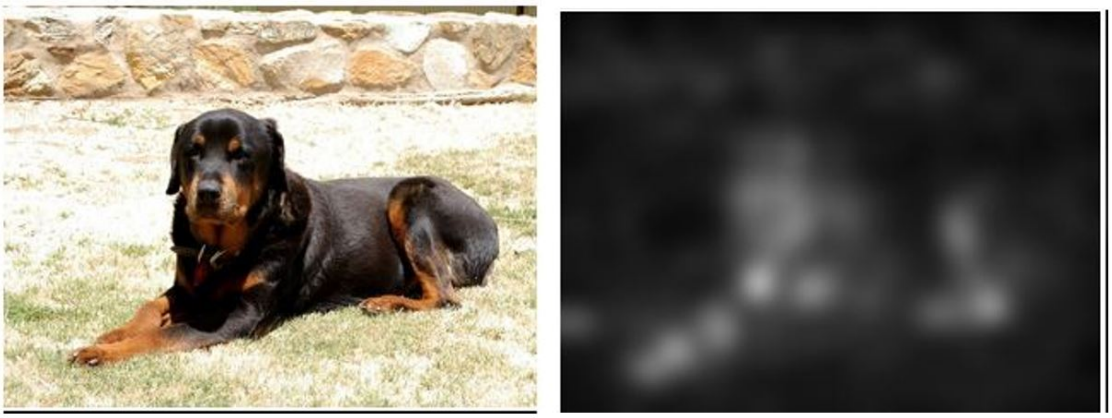

Figure 2.13: BMS Saliency Illustration. (Left) Sample Image from the PASCAL VOC 2007 Test Dataset. (Right) Saliency Map Generated Using the BMS saliency Method.

B_i that attract visual attention [13]. Saliency is then modelled by the mean attention map $\bar{A}$ over the randomly generated Boolean maps. This mean attention map can be further post-processed to form the final saliency map S for the task of salient object detection.

Figure 2.13 shows the saliency map generated by the BMS saliency method for a sample image from the PASCAL VOC 2007 test dataset. The pixel values close to 1 indicate higher saliency while the ones close to 0 indicate least salient pixels.

2.3.3 Covariance Saliency

The Covariance Saliency (CS) method proposed by Erdem et al. [12] uses covariance matrices of simple image features as region covariance descriptors in order to capture the local image structures and provide non-linear integration of the features. In this approach, saliency is obtained by finding the distance between the covariance matrices of a central region with its surrounding neighbourhood regions using a non-Euclidian distance measure based on the eigen values and eigen vectors of the covariance matrices. For an image I with F being the feature extracted, a region R

Figure 2.14: Covariance Saliency Illustration. (Left) Sample Image from the PASCAL VOC 2007 Test Dataset. (Right) Saliency Map Generated using the Covariance Saliency Method.

inside F is represented with a dxd covariance matrix C_R of feature points as shown below:

$$C_R = \frac{1}{n-1}\sum_{i=1}^{n}(f_i - \mu)(f_i - \mu)^T \tag{2.4}$$

where $(f_i)_{i=1,\ldots,n}$ denotes the d-dimensional feature points inside R and μ denotes the mean of these points.

The distance between two covariance matrices C_1 and C_2 is given by:

$$\rho(C_1, C_2) = \sqrt{\sum_{i=1}^{n} \ln^2 \left[\lambda_i(C_1, C_2)\right]} \tag{2.5}$$

where $(\lambda_i(C_1, C_2))_{i=1\ldots n}$ are the generalized eigen values of C_1 and C_2.

The CS method utilizes a multi-scale approach to compute the final saliency map for a given image. Figure 2.14 shows the saliency map that is generated using the CovSal saliency method on a sample image from the PASCAL VOC 2007 test dataset.

Figure 2.15: SigSal Saliency Illustration. (Left) Sample Image from the PASCAL VOC 2007 Test Dataset. (Right) Saliency Map Generated using the Image Signature Saliency Method.

2.3.4 Image Signature Saliency

This approach by Hou et al. [14] develops an image signature based on the sign function of the discrete cosine transform (DCT) which is used to predict visually noticeable pixels in a given image. Let I be the considered image. The saliency map is computed as follows:

$$\hat{I} = sign(DCT(I)) \tag{2.6}$$

$$\bar{I} = IDCT(sign(\hat{I})) \tag{2.7}$$

$$m = g * (\bar{I}.\bar{I}) \tag{2.8}$$

where g is the impulse response of a two-dimensional lowpass smoothing filter and m represents the final saliency map obtained by smoothing the squared reconstructed image $\bar{I}$. Figure 2.15 shows the saliency map generated using the Image Signature (SigSal) method for a sample image from the PASCAL VOC 2007 test dataset.

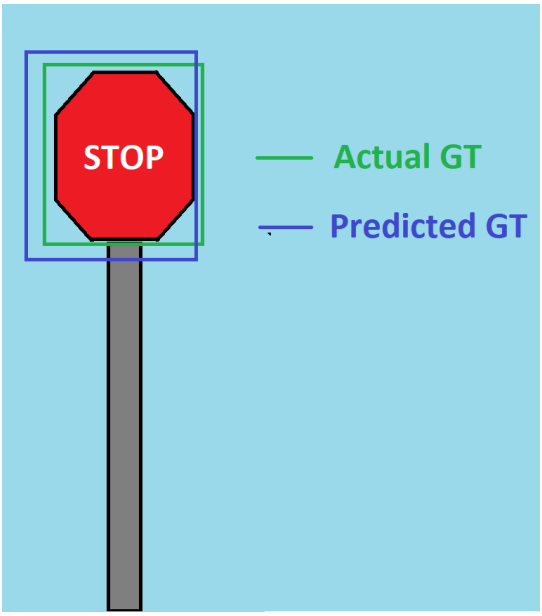

Figure 2.16: An Illustration of Detecting a Stop Sign in a Given Image.

2.4 Performance Evaluation Measures

This section gives a basic understanding of a few evaluation techniques that would be necessary in understanding the proposed framework. Concepts of IoU and detection rate play a key role in better understanding the obtained results in Chapter 4.

2.4.1 Intersection Over Union

The Intersection over Union (IoU) is a popular evaluation metric used in measuring the accuracy of an object detector on a given dataset. This evaluation metric is often seen in object detection challenges such as the PASCAL VOC challenge. Typically, the IoU is used to evaluate the performance of Convolutional Neural Network (CNN) based detectors such as R-CNN, Faster-RCNN, YOLO, etc. Any algorithm that predicts candidate boxes as an output can be evaluated using the IoU.

In order to apply the IoU to evaluate a given object detector, the following components are expected:

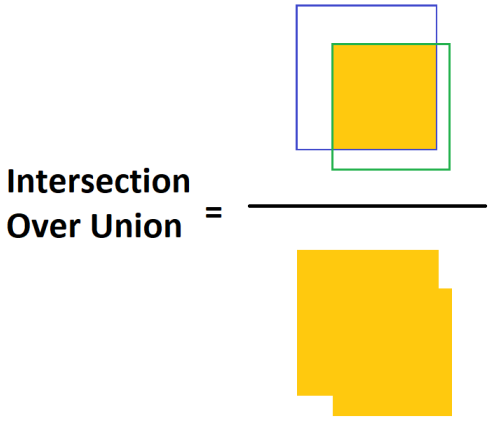

Figure 2.17: Pictorial Representation of Computing the Intersection Over Union (IoU).

- The ground-truth bounding boxes that specify where the object is for a given test image.
- The predicted bounding boxes from the model under evaluation.

In Figure 2.16, the ground-truth bounding box is represented with green color while the predicted bounding box from an object detector is drawn in dark blue. In order to compute the Intersection over Union, the ratio, shown in Figure 2.17, must be calculated.

From Figure 2.17, the Intersection over Union can be described as the ratio of area of overlap to the area of union between the ground-truth and predicted bounding boxes. Areas in case of images can be computed by counting the number of pixels in the desired region. A higher IoU value, close to 1, indicates a better prediction from the object proposal method. Figure 2.18 shows an illustration of good and bad IoU

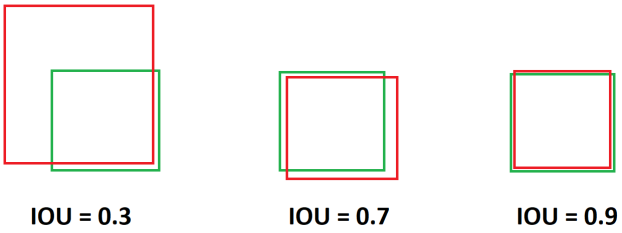

Figure 2.18: IoU Scores Between Ground-Truths (in Green) and Candidate Boxes (in Red).

scores. Typically, a score of 0.5 is used as an IoU threshold to label the object as detected.

2.4.2 Detection Rate

The detection rate, also referred to as recall, of a system is closely associated with the IoU threshold that is used to qualify the object as detected. For a given image having labeled ground-truth objects, the detection rate can be computed using the following ratio:

$$\text{Detection Rate} = \frac{\text{True Positives}}{\text{Total number of ground truths}} \tag{2.9}$$

where *True Positives* is the number of ground truths that are successfully detected by the predicted candidate boxes of the object proposal generator. It is difficult for an object proposal generator to detect the objects at higher IoU thresholds. So, the detection rates decrease with the increase in IoU threshold. Figure 2.19 illustrates this phenomenon by plotting detection rate with respect to IoU threshold. In order to compute the detection rate at a given IoU value for the entire dataset, the detection rates are calculated for all the images of the dataset and their average is computed to get the final detection rate at the considered IoU threshold.

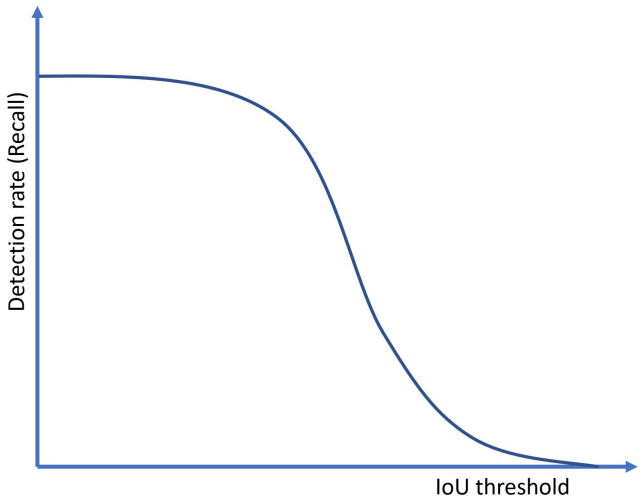

Figure 2.19: Typical Recall vs IoU Curve for an Object Proposal Generator.

2.4.3 Precision

Precision measures how accurate the predictions of the model are. It is the ratio of true object detections to the total number of objects that the classifier classifies. Mathematically, Precision can be defined as follows:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2.10}$$

If the precision score is close to 1.0, then there is a high likelihood that whatever classifier predicts as a positive detection is actually a correct prediction. On the other hand, recall measures the ratio of true object detections to the number of objects in the dataset. Average Precision (AP) is computed at a constant IoU threshold, which is 0.5 in our case.

2.4.4 Area Under the Curve (AUC)

The Area Under the Curve (AUC) is used to compute the average recall over a range of IoU thresholds. One way to compute the AUC is by approximating the area using rectangles. The total area under the curve, as shown in Figure 2.20, can be

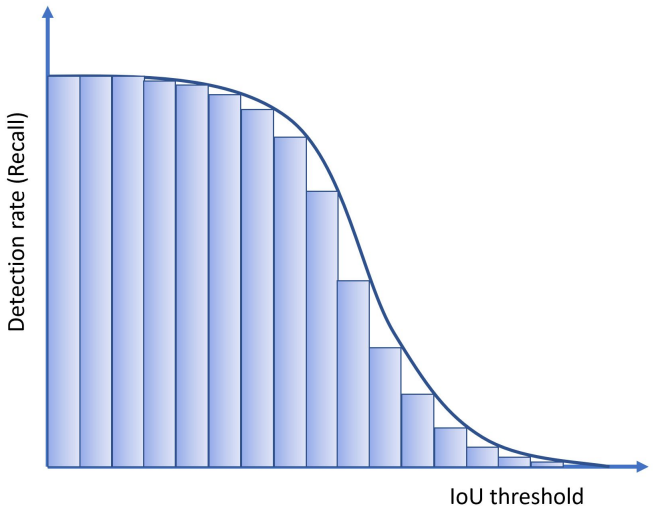

Figure 2.20: Area Under the Curve (AUC) Calculation by Right End-Point Approximation.

approximated to be the sum of individual rectangular areas. As the right end point have been used to define the height of the approximating rectangle above each sub-interval, it is called right end-point approximation for the AUC. For M sub-intevals with a Δx step size, the right end-point approximation for the AUC is given by:

$$AUC \approx \sum_{i=1}^{M} f(x_i)\Delta x \tag{2.11}$$

where $f(x_i)$ is the value of recall at an IoU value of x_i.

As the number of sub-intervals increase, the approximate value comes closer to the actual value. In this thesis, M was set to 400 and a step size of 0.0025 was used to vary the IoU threshold value from 0 to 1.

Chapter 3

SALIENT OBJECT DETECTION DATABASE AND FRAMEWORK FOR BENCHMARKING OBJECT PROPOSAL GENERATORS FOR SALIENT OBJECT DETECTION

This chapter presents a framework for comparing the performance of state-of-the-art object proposal generators for the task of salient object detection. For this purpose, a salient object detection database, referred to as SalBox database, is constructed as part of this work. The predictions obtained by EdgeBoxes [3], FasterR-CNN [4], and SSD [5] are evaluated with respect to the SalBox database. The constructed database provides an insight into which of the existing and future object proposal generators are better at detecting salient objects with a reduced number of proposals. The performance results show a significant reduction in the required number of proposals when detecting just the salient objects. These results also show that SSD [5] provides the maximum reduction in the number of required proposals while maintaining the same salient object detection accuracy as compared to [3] and [4].

3.1 Introduction

Object detection has been a key problem in the field of computer vision for a long time now. The primary aim of an object detection system is to determine whether an object exists in a provided image and, if so, where in the image it occurs. The dominant approach to this problem, for the past decade, has been the sliding window paradigm in which object classification is performed at every location and scale in that image [17, 39, 40]. But more recently, object proposal generation/detection methods

have been proposed in which a set of object-bounding box proposals are generated aiming to reduce the set of positions that need further analysis. These approaches [37, 41–47] led to the discovery that object proposals can be accurately generated in a manner that is agnostic to the type of object being detected [3]. The field of object detection has seen a great progress from the success of these object proposal generation methods, which aim at generating an optimal number of region proposals to cover most of the observable objects in a given frame. A good object proposal generator is expected to efficiently generate as few bounding boxes as possible while reaching a sufficiently high detection rate.

Object detection involves both localization as well as classification of the objects in a given frame. While Edge Boxes [3] is solely an object proposal generation method based on structured decision forests, Faster-RCNN [4] and SSD [5] train on the ground-truth data of a given object detection dataset to learn the parameters required to not only generate object proposals but to also classify those proposals. All the above methods achieve high recall at the cost of sampling a large number of candidate boxes, which leads to an increase in computational complexity, especially when classification is also performed. For example, Edge Boxes [3] requires one of its design parameters, δ, which controls the step size of the sliding window based search, to be high in order to have a better recall at more challenging IoU (Intersection over Union) thresholds. But, if δ is increased, the density of the sampling is increased, resulting in more candidate boxes being evaluated and slower runtimes [3]. Similarly, for SSD [5] and Faster R-CNN [4], an increase in the number of pivot points leads to the same problem. Hence, the detection of all the objects, in a given frame, at higher IoU thresholds requires a significantly much larger number of object proposals to be generated and is computationally expensive.

One way to overcome this problem, while ensuring that the most salient objects in the scene are detected, is to design saliency-enhanced object proposal generation methods that are capable of detecting salient objects in the scene while minimizing the number of generated object proposals. In many applications, such as image display on small devices [15], and image collection browsing [16], it is enough to generate object proposals to detect salient objects in that frame and to subsequently classify them rather than aiming to detect all the objects. Hence, it is important to evaluate the performance of these object proposal generators for the task of salient object detection. This work proposes a framework to assess the performance of object proposal generation and/or detection methods in terms of their ability to detect salient objects while constraining the number of generated object proposals or the number of non-salient object detections to a maximum allowed value. The proposed framework can also be used to evaluate and optimize the performance of newly developed object detection methods with a focus on increasing the accuracy of salient object detection.

In general, the IoU score between the predicted bounding boxes and the reference ground-truth bounding boxes is used to evaluate the performance of an object proposal generator, i.e., if the IoU score between those two bounding boxes is more than a given threshold, the proposed bounding box is labeled as a true positive. Although this method of calculating recall for varying IoU thresholds helps in analyzing the performance of an object proposal generator with respect to the ground-truth database, there are currently no databases and no evaluation framework to assess the performance of object proposal generation/detection methods in terms of their ability to detect salient objects. Hence, as part of this work, a benchmark database (SalBox database) with salient ground-truth annotations is constructed to enable such a much needed saliency-enhanced evaluation of existing object proposal generators. Not only will such a database help in finding the best performing object proposal generator

for salient object detection, but also it establishes a benchmark by which other newly introduced proposal generators can be compared.

This chapter is organized as follows. Section 3.2 describes the proposed framework and constructed SalBox database. Performance results using the proposed framework and the constructed SalBox database are presented in Section 3.3.

3.2 Salient Object Detection Evaluation Framework and Database

Although a large number of object proposal generators aim to predict the bounding boxes enclosing most of the noticeable objects in a given frame, it is important to evaluate whether these methods are capable of detecting the salient objects in the scene when constraining the number of proposals that can be generated due to constraints on timing or computations during execution. To address this issue, a salient object detection database, referred to as SalBox database, is constructed following the procedure shown in Figure 3.1 for images in the PASCAL VOC 2007 test dataset.

Given an input image with corresponding ground-truth object (GTO) bounding boxes annotations, the proposed SalBox dataset is constructed by first generating the saliency map for the input image. The generated saliency map is thresholded to produce a binary mask with 1 (white pixel in Figure 3.1) and 0 (black pixel in Figure 3.1) denoting a salient and non-salient location, respectively. The salient regions are then enclosed by bounding boxes and these in turn represent the saliency map bounding boxes. The IoU scores between the GTO bounding boxes and saliency map bounding boxes are computed. GTO bounding boxes with an IoU score higher than a specified threshold, expressed as a percentage δ of the maximum IoU score in the considered image (67% in our implementation), are kept and referred to as the salient GTO (SGTO) bounding boxes. All other bounding boxes are removed from the annotations. The IoU scores determine the degree of saliency of the ground-

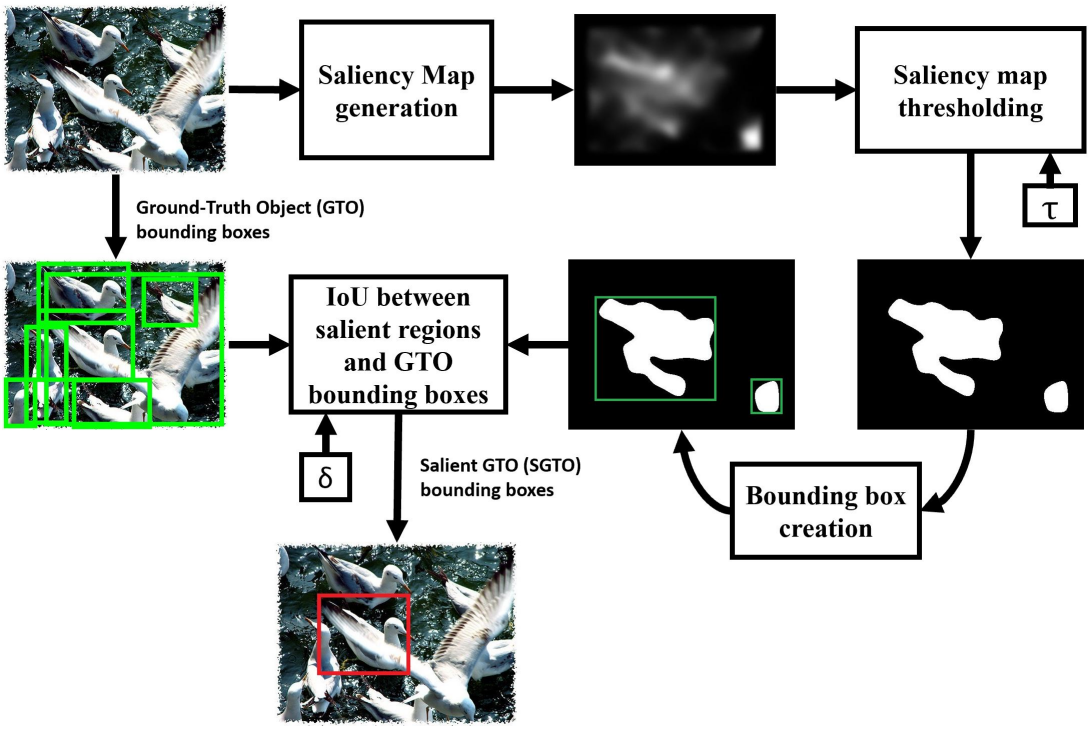

Figure 3.1: Process Involved in Generating Salient Ground-Truths from the Provided Ground-Truths for a Given Dataset.

truth objects for a given image. The higher the IoU score, the higher would be the degree of saliency of that object. The IoU parameter δ, $0 \leq \delta \leq 1$, determines the degree of overlap between a GTO bounding box and a saliency map bounding box relative to the maximum overlap present in the considered input image. If δ equals 1, the method outputs only the most salient ground-truth in that image. When it is 0, all the ground-truth objects in the given frame are output as salient ground-truths. This parameter is important in analyzing the performance of object proposal generators against objects having a certain degree of saliency. Finally, the output of the procedure shown in Figure 3.1 would be a dataset with just the salient ground-truths.

The proposed SalBox database was constructed by applying the aforementioned procedure (Figure 3.1) to all images of the PASCAL VOC 2007 test dataset. The saliency maps were generated by using the FES saliency method [11], which ranked among the top saliency prediction methods according to the performance evaluations in [1]. In our implementation, the values $\tau = 0.5$ and $\delta = 0.67$ (Figure 3.1) were used for thresholding the saliency map and the IoU scores, respectively.

Details about the generation of the saliency map and the determination of salient ground-truth object bounding boxes are provided in the following subsections.

3.2.1 Saliency Map Generation

In order to determine which saliency map prediction method to adopt, we selected the top 4 saliency prediction methods based on the work of Gide and Karam [1]. These methods are FES [11], CovSal [12], BMS [13] and SigSal [14]. These methods were ranked in [1] by conducting subjective evaluations comparing the predicted saliency map to the reference ground-truth saliency map. The subjective ratings were obtained in [1] by using a 5-point quality scale (5 being Excellent and 1 being Poor). The mean opinion score (MOS) was computed for each saliency map prediction method by averaging the subjective ratings over all the saliency maps produced by the considered method. Figure 3.2 shows the resulting MOS in decreasing order. From Figure 3.2, it can be seen that FES [11], CovSal [12], BMS [13] and SigSal [14] ranked as the top 4 methods in terms of MOS. Figure 3.3 shows the saliency maps generated by all the four VA models for two sample images taken from the PASCAL VOC 2007 test dataset. Table 3.1 gives the average running time for the selected saliency prediction methods for all the images in the Pascal VOC 2007 test dataset. All methods were run using MATLAB 2017b on a Microsoft Surface Pro 4 with Intel Core i7-6650U CPU running at 2.9 GHz clock frequency. Out of these methods, BMS [13] uses

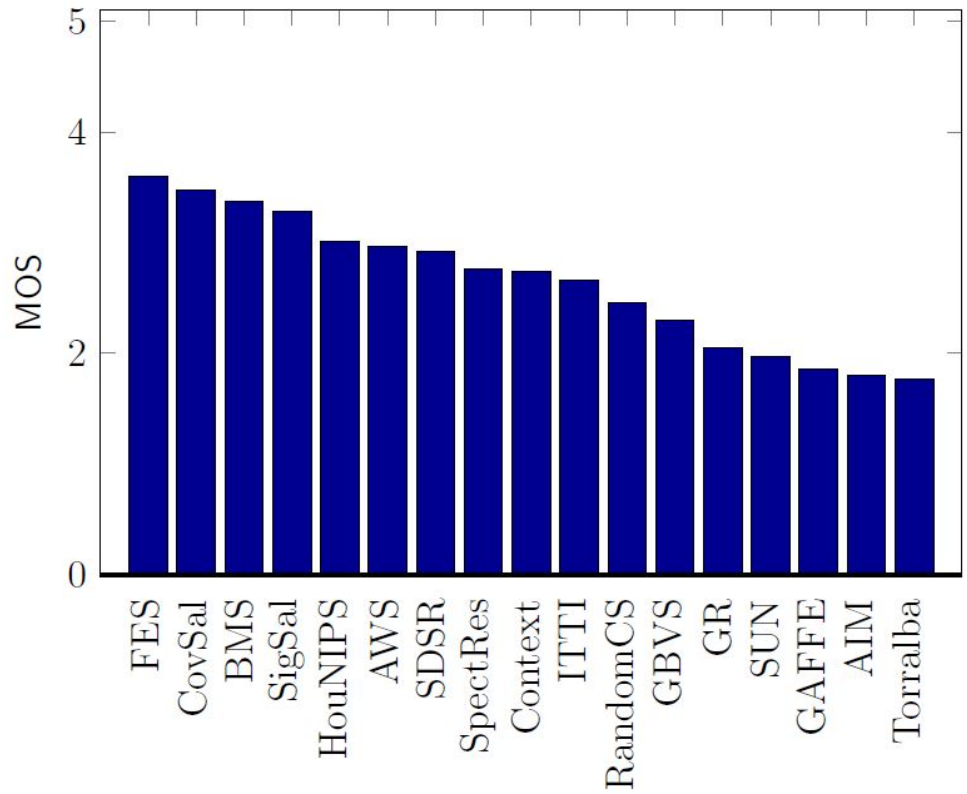

Figure 3.2: MOS Taken Over All Predicted Saliency Maps for Each VA Model and Arranged in Descending Order by Milind S. Gide and Lina J. Karam (2017) [1].

MEX routines to speed up the saliency map generation process while the rest of the methods call standard MATLAB functions. From Figure 3.2 and Table 3.1, it can be seen that the FES method [11] is not only the top performing method in terms of saliency prediction but it is also the fastest in terms of average running time when compared to the three other considered methods (CovSal, BMS and SigSal). Hence, FES [11] is chosen to generate saliency maps for all the images in the PASCAL VOC 2007 test dataset and these saliency maps are further used to generate the salient ground-truth object bounding boxes.

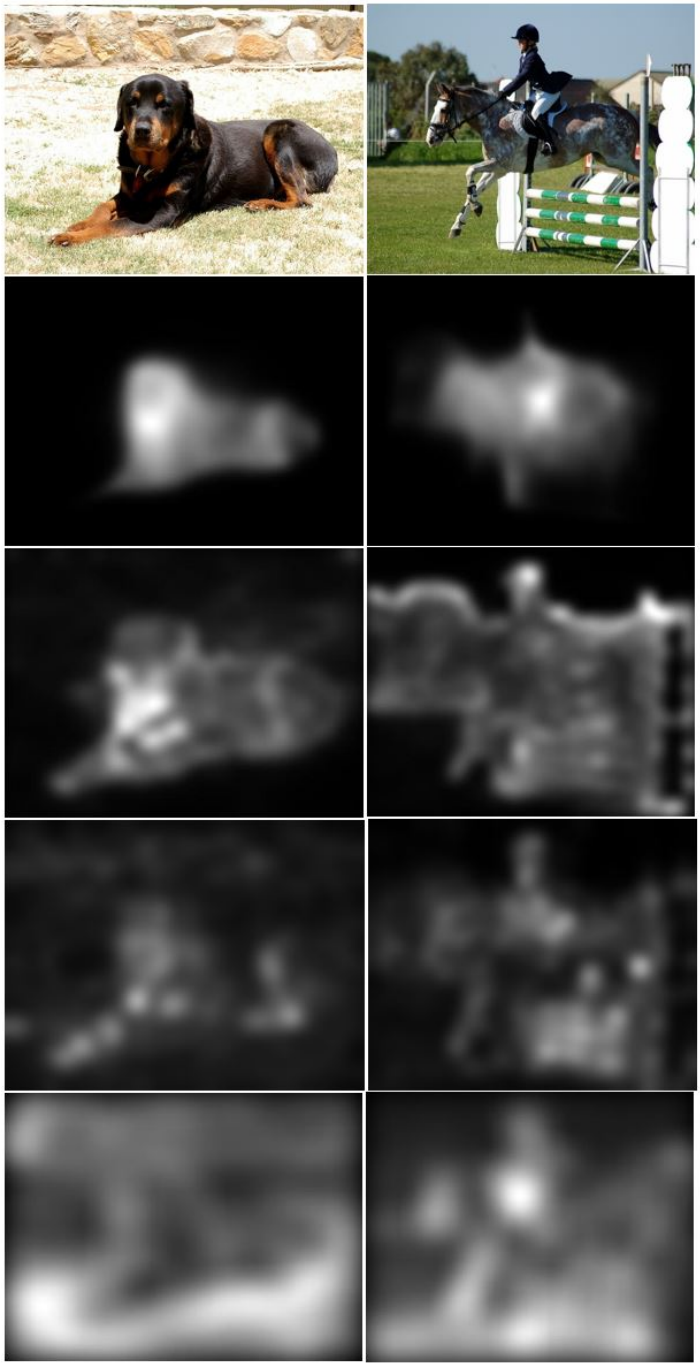

Figure 3.3: Example of Saliency Maps That Are Generated Using the Top Performing Methods. From Top to Bottom Row: Two Sample Images from the PASCAL VOC 2007 Test Dataset and Corresponding Saliency Maps Generated by FES [11], CovSal [12], BMS [13], and SigSal [14], Respectively.

Table 3.1: Average Run Times for the Top Four VA Models in [1] for the Images From PASCAL VOC 2007 Test Dataset.

Methods	Time taken (sec)
FES [11]	0.125
BMS [13]	0.147
SigSal [14]	0.496
CovSal [12]	20.046

3.2.2 Determination of Salient Ground-Truth Objects

Once the saliency maps are generated, they are thresholded to form binary masks as explained previously. After this, a recursive connected component labelling algorithm is applied to label the salient regions in each thresholded saliency map (MATLAB R2017b function *regionprops* was used for this step). For every labelled salient region, a rectangular salient region bounding box is formed by enclosing it. For every image in the dataset, IoU scores are computed between the ground-truth objects (GTO) bounding boxes and salient regions bounding boxes. Then, salient ground-truth objects (SGTO) bounding boxes are determined by retaining those GTO bounding boxes resulting in relatively large IoU scores. The remaining GTO bounding boxes with lower IoU scores are discarded. Figure 3.4 illustrates the process of finding the salient ground-truth object bounding boxes using the thresholded saliency map.

Let D represent the dataset of all the images with their corresponding GTO bounding boxes. Let GTO_k, S_k and T_k represent, respectively, the set of GTO bounding boxes, saliency map, and thresholded saliency map corresponding to the k^{th} image I_k

Figure 3.4: Generation of Salient Ground-Truth Objects (SGTOs) From Provided Ground-Truth Objects (GTOs). From Top Row to Bottom Row: Three Sample Images From the PASCAL VOC 2007 Test Dataset with Corresponding GTO Bounding Boxes; Binary Maps Generated by Thresholding the FES [11] Saliency Maps, and Corresponding Salient Regions Bounding Boxes; Images with Salient GTO Bounding Boxes.

in D. $T_k(x, y)$ is obtained by thresholding the saliency map $S_k(x, y)$ as follows:

$$T_k(x,y) = \begin{cases} 1, & \text{if } S_k(x,y) \geqslant \tau \\ 0, & \text{if } S_k(x,y) < \tau \end{cases} \tag{3.1}$$

where τ is a specified threshold. In our implementation, the value of τ was set to 0.5. Each connected region formed by the set of pixels (x, y) where $T_k(x, y) = 1$ is enclosed by a rectangular bounding box. Let R_k be the set of rectangular bounding boxes enclosing each connected region in the thresholded saliency map T_k (Figure 3.1 and middle row of Figure 3.4) and let R_{kj} be the j^{th} bounding box in the set R_k.

The next step is to identify which of the ground-truth objects fall within the salient regions and to compute their degree of saliency. Let θ_{ki} be the degree of saliency of the i^{th} GTO bounding box GTO_{ki} in the set GTO_k. θ_{ki} is given by:

$$\theta_{ki} = \max_j(IoU(GTO_{ki}, R_{kj})) \tag{3.2}$$

where $IoU(GTO_{ki}, R_{kj})$ computes the IoU score between the i^{th} ground-truth object and the j^{th} rectangular salient region bounding box.

Let $SGTO_k \, \epsilon \, GTO_k$ be the set of salient GTOs in image I_k given by:

$$SGTO_k = \left\{ GTO_{ki} \mid \theta_{ki} \geqslant \delta.\theta_{k,max} \right\} \tag{3.3}$$

where $\theta_{k,max} = \max_i(\theta_{ki})$, and δ is a relative saliency threshold. The value of δ was set to 0.67 in our implementation.

Using Equations (3.1), (3.2), and (3.3), salient ground-truth objects can be obtained for any given dataset D. Using these equations, the SalBox dataset was generated from the PASCAL VOC 2007 test dataset with the parameters τ and δ being 0.5 and 0.67, respectively. This dataset can be used along with our framework to analyze the performance of any newly introduced object proposal technique for the

Table 3.2: Number of GTOs per Image in the PASCAL VOC 2007 Test Dataset [2] and SalBox Dataset.

Dataset	Minimum GTOs	Maximum GTOs	Average GTOs per image	Standard deviation
PASCAL VOC 2007 test dataset [2]	1	41	3.3296	3.3895
SalBox dataset	1	9	1.2223	0.5159

salient object detection task and to compare the performance with respect to state-of-art techniques like [3], [4] and [5]. Table 3.2 shows statistics corresponding to the number of GTOs per image in the PASCAL VOC 2007 test dataset [2] and in the SalBox dataset.

3.3 Experimental Results

After the generation of salient ground-truths, the object proposal generator under evaluation is analyzed with respect to both the original ground-truths as well as the salient ground-truths. For this purpose, the detection rate (recall) vs IoU curves are plotted by varying the IoU threshold α for different N values, where N is the number of proposals that can be generated. IoU threshold α determines if the proposed bounding box, by the object proposal generator, is a true detection. If the IoU between the ground-truth bounding box and the predicted bounding box exceeds α, the predicted bounding box is counted as a positive detection. This process can be visualized with the help of the block diagram shown in Figure 3.5. Figures 3.6, 3.7 and 3.8 show the recall VS IoU curves for EdgeBoxes [3], Faster R-CNN [4] and SSD [5], respectively. The performance analysis is performed with respect to all (salient

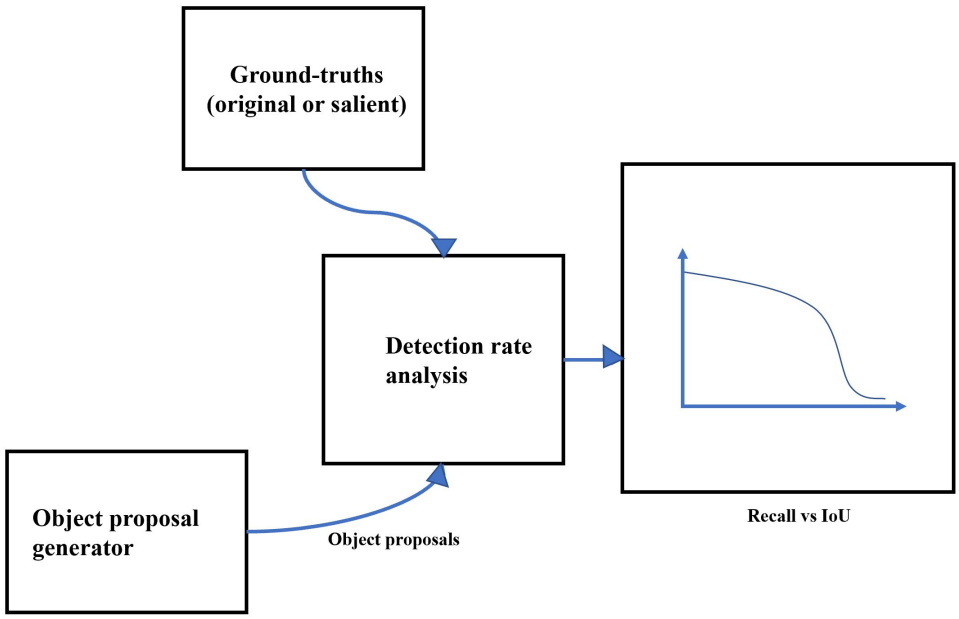

Figure 3.5: Detection Rates of the Object Proposal Generator Under Evaluation with Respect to the Original Ground-Truths as Well as the Salient Ground-Truths with Varying IoU Thresholds.

and non-salient) ground-truths as well as the salient ground-truths by varying the number of generated proposals (N).

From Figure 3.6, it can be seen that the detection rate of EdgeBoxes [3] with respect to SGTO bounding boxes with $N = 20$ proposals, at an IoU threshold of 0.5, is similar to the detection rate of EdgeBoxes with respect to all ground-truths with $N = 100$ proposals. Similarly from Figure 3.7, it can be seen that the detection rate of Faster R-CNN [4] with respect to SGTO bounding boxes with $N = 50$ proposals, at an IoU threshold of 0.5, is similar to the detection rate of Faster R-CNN with respect to all ground-truths with $N = 100$ proposals. Hence, the number of proposals generated by EdgeBoxes and Faster R-CNN can be cutdown by 80 percent and 50 percent

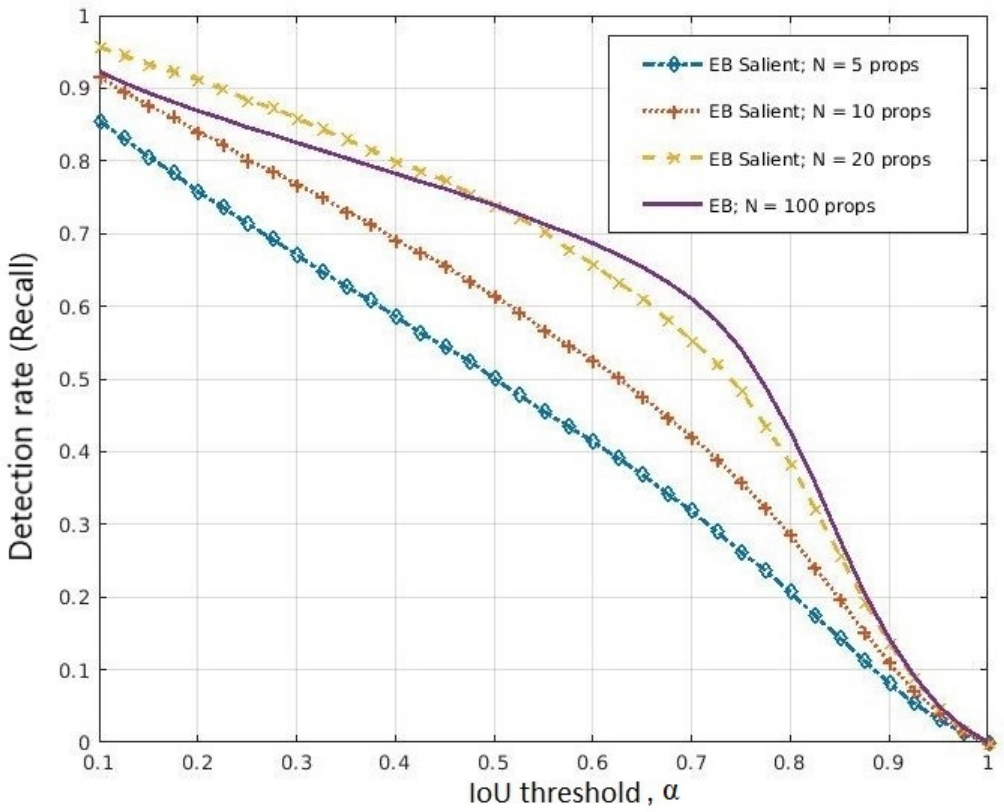

Figure 3.6: Detection Rate (Recall) VS IoU Threshold for the Proposals Generated by EdgeBoxes [3], with Respect to All Ground-Truth Objects (EB) and Salient Ground-Truth Objects (EB Salient) for Different Numbers of Object Proposals.

respectively, when detecting just the salient ground-truths without compromising the detection rate.

From Figure 3.8, it can be seen that the detection rate of SSD [5] with respect to the salient ground-truths with $N = 5$ proposals, at an IoU threshold of 0.5, is similar to the detection rate of SSD with respect to all ground-truths and with $N = 100$ proposals. This reinforces the fact that the number of proposals generated by SSD can be cut down by 95 percent when detecting just the salient ground-truths without any compromise in recall. Table 3.3 shows the AUC values, computed using Equa-

tion (2.11), for different number of proposals N and for different methods including EdgeBoxes [3], Faster R-CNN [4], and SSD [5] when detecting all GTOs and salient GTOs. It can be seen that SSD [5] has the highest AUC value for 100 proposals when detecting the salient ground-truths when compared to the rest of the cases. Table 3.4 shows that a significant reduction in the number of object proposals can be achieved for a given recall rate when the detection focuses on salient objects. The best performance is achieved by SSD [5] which requires only 5 proposals when detecting salient objects to achieve the same recall rate as compared to 100 proposals when detecting all objects.

For all the three methods [3–5], and for a fixed number of generated object proposals (N), it can be observed from Figures 3.6, 3.7, and 3.8 that the detection rates decrease with an increase in the IoU threshold for a fixed number of proposals generated. This is due to the difficulty involved in perfect localization of object at higher IoU thresholds. From Table 3.4, at a typical IoU threshold of 0.5, the detection rate is higher for SSD when compared to [3] and [4]. Also, from Table 3.5, it can be seen that the Average Precision (AP) for [3–5] increases when proposals are aimed to detect salient GTOs. The Average Precision (AP) was calculated using Equation (2.10) with the number of proposals N set to the average number of GTOs per image for the dataset under evaluation. These values are shown in Table 3.2.

More interesting results are observed when the detection rates are analyzed with respect to the following cases: 1) having only one most salient object as ground-truth for every image in dataset, 2) having only one least salient object as ground-truth for every image in the dataset. First case is obtained by choosing GTO bounding box with θ_{max} degree of saliency while the second case is obtained by choosing GTO bounding box with θ_{min} degree of saliency for each image in the dataset. From Figure 3.9, it can be noticed that the SSD [5] method's detection rate increased approximately

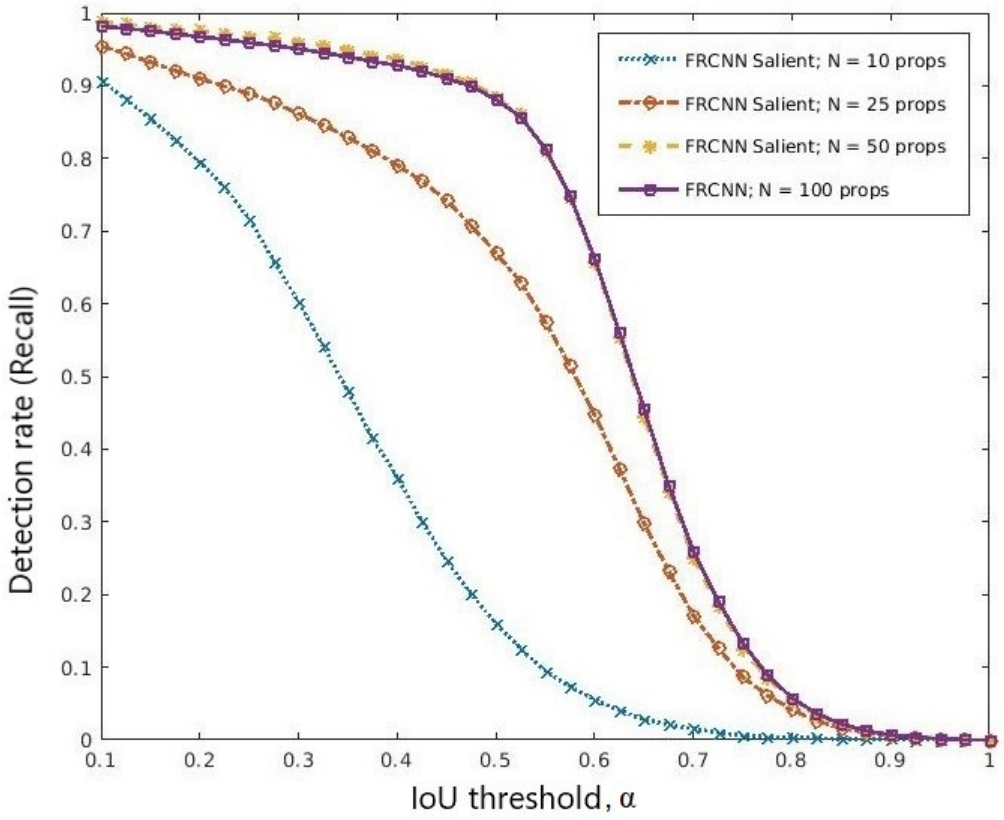

Figure 3.7: Detection Rate (Recall) vs IoU Threshold Using Faster R-CNN [4] with Respect to All Provided Ground-Truths (FRCNN) and Salient Ground-Truths (FR-CNN Salient) for Different Number of Object Proposals.

by 10 percent when detecting the most salient object, at a typical IoU value α of 0.5, as compared to detecting the least salient object. Similarly, the detection rate using the Edge Boxes [3] method with 100 proposals to detect the most salient object is approximately 12.5 percent greater than the detection rate with 100 proposals to detect the least salient object. This is an interesting finding as it clearly demonstrates the fact that the degree of saliency rather than the number of ground-truths in a given image influence the detection rate of the object proposal system.

Table 3.3: AUC Values for Various Number of Proposals Generated by EdgeBoxes (EB) [3], Faster R-CNN (FRCNN) [4], and SSD [5] with Respect to All GTO Bounding Boxes and SGTO Bounding Boxes.

Method	Ground-Truth	Number of proposals				
		5	10	25	50	100
EB [3]	All GTOs	0.3961	0.4673	0.5473	0.6026	0.6352
	SGTOs	0.4706	0.5510	0.6358	0.7075	0.7205
FRCNN [4]	All GTOs	0.2463	0.3237	0.4960	0.5965	0.6667
	SGTOs	0.2851	0.3888	0.5809	0.7088	0.7361
SSD [5]	All GTOs	0.6461	0.6786	0.6956	0.7081	0.7280
	SGTOs	0.7362	0.7523	0.7701	0.7865	0.8104

Table 3.4: Reduction in the Number of Proposals When Detecting Only Salient Objects.

Methods	Detection rate (Recall)	Number of proposals to detect GTs	Number of proposals to detect salient GTs	Percentage of reduction in number of proposals
EB [3]	0.74	100	25	75
FRCNN [4]	0.87	100	50	50
SSD [5]	0.91	100	5	95

Table 3.5: Average Precision (AP) Values for Salient GTs and All GTs At a δ Value of 0.5. The AP Values Were Computed by Setting the Number of Proposals to be Equal to the Average Number of GTOs per Image for the Considered Datasets.

Methods	AP (All GTs)	AP (Salient GTs)
EB [3]	0.1986	0.2672
FRCNN [4]	0.4836	0.5804
SSD [5]	0.5714	0.7278

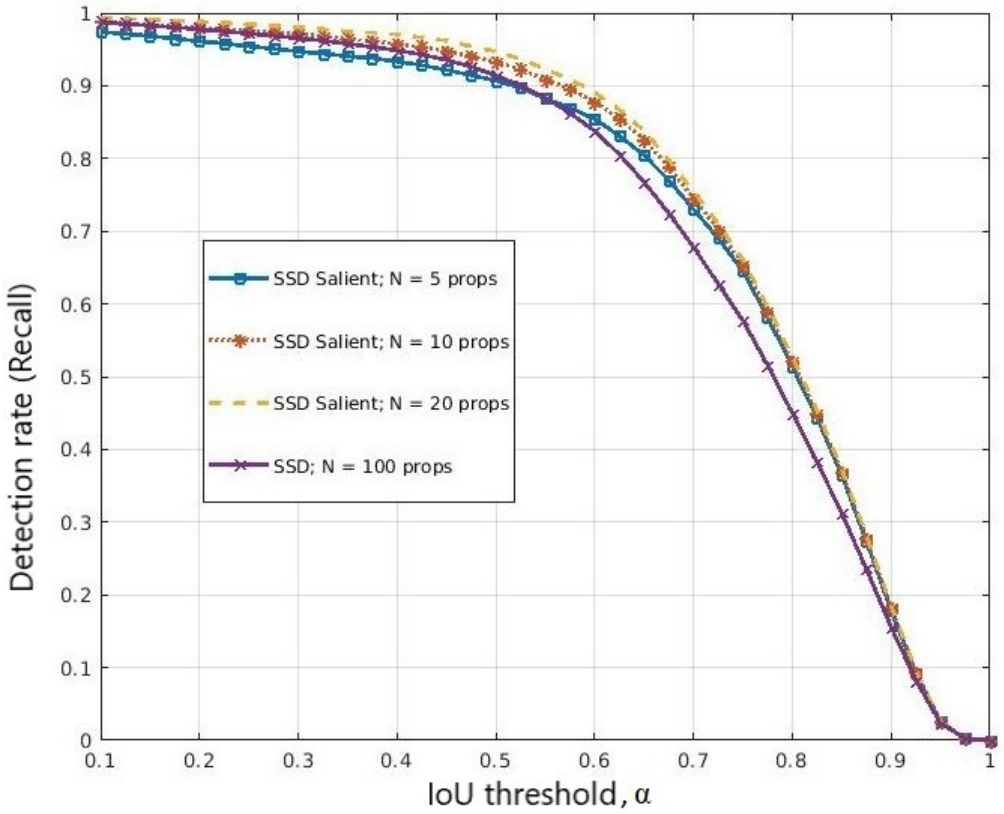

Figure 3.8: Detection Rate (Recall) vs IoU Threshold Using SSD [5] with Respect to All (SSD) and Salient Ground-Truth Objects (SSD Salient) for Different Number of Object Proposals.

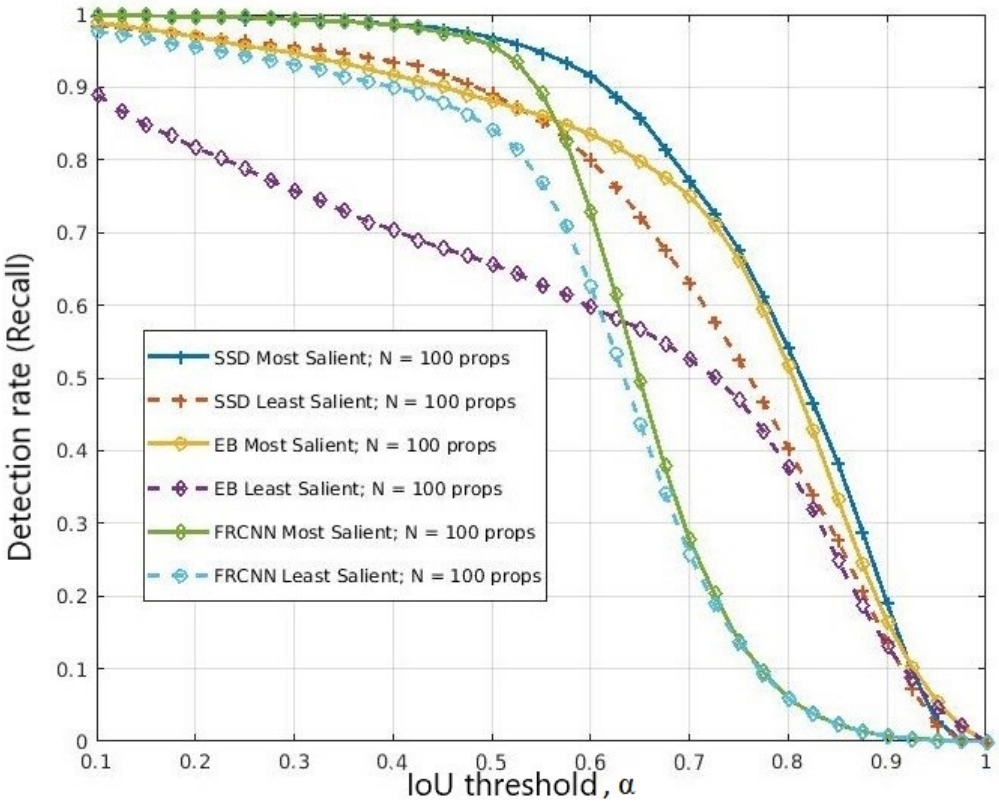

Figure 3.9: Detection Rate (Recall) VS IoU Threshold for EdgeBoxes [3], Faster R-CNN [4], and SSD [5] with Respect to Most Salient and Least Salient Ground-Truths.

Chapter 4

CONCLUSION

This thesis implements an evaluation framework for salient object proposal generation for any given image dataset. This work contributes to the field of object detection in general and to the area of salient object proposal generation in particular. This chapter summarizes the contributions of this thesis and proposes several directions for future research.

4.1 Contributions

The contributions of this thesis can be summarized as follows:

- A novel evaluation framework is presented for assessing the performance of a given object proposal generator for the task of salient object proposal generation.
- The proposed framework was used to evaluate the performance of state-of-art object proposal generators/detectors.
- Given a labeled object dataset as input, a novel algorithm is presented to construct a corresponding labeled salient object dataset.
- A benchmark dataset for salient object detection is constructed from the PASCAL VOC 2007 test dataset in order to facilitate the evaluation of any newly introduced object proposal generators for the task of salient object detection.

4.2 Future Research Directions

Possible enhancements and future directions for the proposed framework are as follows:

- An end-to-end salient object detector can be trained on the salient PASCAL VOC 2007 dataset using the SSD framework. It will be interesting to see if the resulting trained SSD network would help in generating less number of proposals when compared to the conventional object detector for the task of salient object detection.

- Recent advances in deep learning show that features obtained from initial layers of a CNN trained for object detection are very useful in other computer vision tasks. Current saliency models use either linear or non-linear combinations of low-level features to produce saliency maps. It will be interesting to see if the fusion of CNN features and the current bottom-up saliency models can lead to better saliency maps.

REFERENCES

[1] Milind S Gide and Lina J Karam. Computational visual attention models. *Foundations and Trends in Signal Processing*, 10(4):347–427, 2017.

[2] Mark Everingham, L Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. Available online at: `www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html`, Accessed on Jan. 01, 2018.

[3] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016.

[6] Roy Bayot and Teresa Gonçalves. A survey on object classification using convolutional neural networks. *Technical Report, University of Évora*, 2015. Available online at: `dspace.uevora.pt/rdpc/bitstream/10174/17508/1/CNNPaper02.pdf`, Accessed on January 01, 2019.

[7] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[9] Ross Girshick. Fast R-CNN. *arXiv preprint arXiv:1504.08083*, 2015.

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

[11] Hamed Rezazadegan Tavakoli, Esa Rahtu, and Janne Heikkilä. Fast and efficient saliency detection using sparse sampling and kernel density estimation. In *Scandinavian Conference on Image Analysis*, pages 666–675. Springer, 2011.

[12] Erkut Erdem and Aykut Erdem. Visual saliency estimation by nonlinearly integrating features using region covariances. *Journal of Vision*, 13(4):11–11, 2013.

[13] Jianming Zhang and Stan Sclaroff. Saliency detection: A boolean map approach. In *IEEE International Conference on Computer Vision (ICCV)*, pages 153–160, 2013.

[14] Xiaodi Hou, Jonathan Harel, and Christof Koch. Image signature: Highlighting sparse salient regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):194–201, 2012.

[15] Li-Qun Chen, Xing Xie, Xin Fan, Wei-Ying Ma, Hong-Jiang Zhang, and He-Qin Zhou. A visual attention model for adapting images on small displays. *Multimedia Systems*, 9(4):353–364, Springer, 2003.

[16] Carsten Rother, Lucas Bordeaux, Youssef Hamadi, and Andrew Blake. Autocollage. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 847–852, 2006.

[17] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[18] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[19] Jun Yang, Yu-Gang Jiang, Alexander G Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In *ACM International Workshop on Multimedia Information Retrieval*, pages 197–206, 2007.

[20] Kevin P Murphy. Naive bayes classifiers. *Technical Report, University of British Columbia*, 2006. Available online at: `https://www.cs.ubc.ca/~murphyk/Teaching/CS340-Fall06/reading/NB.pdf`, Accessed on Jan. 01, 2019.

[21] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.

[22] Fei-Fei Li, Rob Fergus, and Antonio Torralba. Recognizing and learning object categories. *Tutorial at IEEE International Conference on Computer Vision (ICCV)*, 2005. Available online at: `http://people.csail.mit.edu/torralba/shortCourseRLOC/`, Accessed on Jan. 04, 2019.

[23] Yann LeCun, D Touresky, G Hinton, and T Sejnowski. A theoretical framework for back-propagation. In *Proceedings of the 1988 Connectionist Models Summer School*, volume 1, pages 21–28. CMU, Pittsburgh, PA: Morgan Kaufmann, 1988.

[24] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. Available online at: `http://www.deeplearningbook.org`, Accessed on Jan. 01, 2018.

[25] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[26] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106–154, 1962.

[27] David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195(1):215–243, 1968.

[28] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and Cooperation in Neural Nets*, pages 267–285. Springer, 1982.

[29] Jim Mutch and David G Lowe. Multiclass object recognition with sparse, localized features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 11–18, 2006.

[30] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[31] Daniel Sonntag, Michael Barz, Jan Zacharias, Sven Stauden, Vahid Rahmani, Áron Fóthi, and András Lőrincz. Fine-tuning deep CNN models on specific MS COCO categories. *arXiv preprint arXiv:1709.01476*, 2017.

[32] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[33] Patrice Y Simard, David Steinkraus, and John C Platt. Best practices for convolutional neural networks applied to visual document analysis. In *IAPR International Conference on Document Analysis and Recognition*, volume 3, pages 958–962, 2003.

[34] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3642–3649, 2012.

[35] Dan Cireşan, Ueli Meier, Jonathan Masci, and Jürgen Schmidhuber. A committee of neural networks for traffic sign classification. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 1918–1921, 2011.

[36] Le Kang, Jayant Kumar, Peng Ye, Yi Li, and David Doermann. Convolutional neural networks for document image classification. In *International Conference on Pattern Recognition (ICPR)*, pages 3168–3172, 2014.

[37] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, 2012.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[39] Paul Viola and Michael J Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[40] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[41] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[42] Joao Carreira and Cristian Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1312–1328, 2012.

[43] Esa Rahtu, Juho Kannala, and Matthew Blaschko. Learning a category independent object detection cascade. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1052–1059, 2011.

[44] Santiago Manen, Matthieu Guillaumin, and Luc Van Gool. Prime object proposals with randomized prim's algorithm. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2536–2543, 2013.

[45] Ian Endres and Derek Hoiem. Category-independent object proposals with diverse ranking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):222–234, 2014.

[46] Pekka Rantalankila, Juho Kannala, and Esa Rahtu. Generating object segmentation proposals using global and local search. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2417–2424, 2014.

[47] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3286–3293, 2014.

[48] Piotr Dollár and C Lawrence Zitnick. Structured forests for fast edge detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1841–1848, 2013.

[49] Piotr Dollár and C Lawrence Zitnick. Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37 (8):1558–1570, 2015.

[50] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[51] Samuel Albanie. Faster R-CNN. Available online at: `https://github.com/albanie/mcnFasterRCNN`, 2017. Accessed on February 17, 2019.

[52] Forson Eddie. Understanding SSD MultiBox- Real-Time Object Detection In Deep Learning, 2017. Available online at: `https://tinyurl.com/y99k8q9m`, Accessed on February 17, 2019.

[53] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[54] Christian Szegedy, Scott Reed, Dumitru Erhan, Dragomir Anguelov, and Sergey Ioffe. Scalable, high-quality object detection. *arXiv preprint arXiv:1412.1441*, 2014.

[55] Jeremy M Wolfe, Kyle R Cave, and Susan L Franzel. Guided search: an alternative to the feature integration model for visual search. *Journal of Experimental Psychology: Human perception and performance*, 15(3):419, 1989.

[56] Robert Desimone and John Duncan. Neural mechanisms of selective visual attention. *Annual Review of Neuroscience*, 18(1):193–222, 1995.