Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

# Salient Object Detection using AttentionMask

**Master's Thesis**
at Research Group Computer Vision
Prof. Dr. Simone Frintrop

Department of Informatics
MIN-Faculty
Universität Hamburg

submitted by
**Gitanjali Nair**
Course of study: Intelligent Adaptive Systems
Matrikelnr.: 6885261
on
19.12.2019

| | |
|---|---|
| Examiners: | Prof. Dr. Simone Frintrop |
| | Dr. Mikko Lauri |
| Adviser: | Christian Wilms |

# Abstract

Salient object detection is a key area of research in the field of Computer Vision. Several models with state-of-the-art results have been developed over the years to improve the performance of the salient object detection task. However, there is still room for improvement. As this task has real world applications such as in the health and automobile industries, these models must aim for minimal to zero failure rate. Therefore, in this thesis, a method to further improve the performance of the salient object detection task is proposed.

AttentionDSS, a novel architecture that incorporates objectness attention information into the salient object detection task is proposed. Objectness attention indicates the probability that a pixel in an image belongs to an object. The proposed network is composed of the state-of-the-art salient object detection model, DSS (Deeply supervised salient object detection with short connections) by Hou et al. [13] and the Scale-specific Objectness Attention Modules (SOAMs) of the AttentionMask object discovery system by Wilms and Frintrop[48]. The thesis aims to address the limitations of the DSS model. The DSS model fails to completely segment the most salient object in low contrast and cluttered images. This thesis answers whether objectness attention information could improve the recall value of the DSS model such that the complete salient object is retrieved, while the false positives are kept low. In AttentionDSS, the objectness attention maps from the SOAMs are fused with the saliency maps from the DSS model to produce an enhanced saliency map.

From the experiments conducted, it can be seen that the objectness attention information helps to improve the recall for saliency maps where the DSS model does not predict any false positives. However, for images where the saliency maps obtained from the DSS model have false positives, the objectness attention information causes the quality of the maps to further deteriorate. Based on the cases where AttentionDSS performs better than the DSS model, it can be inferred that objectness attention has the potential to assist the salient object detection task. Different approaches to incorporate this information into the salient object detection task can be explored in the future for improved results. Experimental results show that the SOAMs can be trained for the objectness attention prediction task by using a base network optimized with weights for the salient object detection task. This shows that objectness attention prediction and salient object detection are related tasks.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Human beings have the incredible ability of perceiving, processing and interpreting visual information with the help of their optic and nervous systems. In the age of automation and artificial intelligence, many industries are aiming to incorporate human like abilities into machines, vision being one of them. The selective processing of elements within a scene to analyse a given surrounding and comprehend a situation is an innate ability of human beings. This ability develops over time as the human engages in interactions with various environments comprising of the contextual arrangement of different elements. For instance, in a kitchen environment, one would expect to locate the oven below the stove or the exhaust hood above the stove. In the field of computer vision, the ability to detect the visually significant elements in an image is termed as Salient Object Detection. An object is considered to be salient if it is visually distinctive due to its shape, size or striking contrast with the background causing the viewer to focus attention on this object in order to draw an understanding of the scene. Examples of images marked with the most salient object can be seen in Figure 1.1

Typically research in this field has been conducted with the assumption that there exists only one salient object in a scene. However recent papers [27], [3] argue that there could be multiple salient objects in a scene, in which case they are ranked in order of their saliency. There exists several manual and deep learning approaches to enable machines to learn the salient object detection task. The deep learning approaches have outperformed hand-crafted rule based methods. However, the state-of-the-art deep learning approaches for salient object detection have not acquired performance accuracy and robustness that would match human abilities. There are several complex cases such as low contrast and cluttered scenes that these methods are unable to process accurately. In this thesis, an approach to improve the performance of the salient object detection task is proposed.

## 1.1 Problem Statement

Salient object detection has applications in tasks such as object detection, object discovery, image segmentation, content-based image retrieval, video summarization

Figure 1.1: Images from the MSRA10K dataset [6]. (first row) Bounding boxes around the most salient object as ground truth annotation, (second row) Pixel precise segmentation of the most salient object as ground truth annotation.

and visual tracking [4]. Some of these tasks have a crucial role in real world domains such as the medical and self-driving car industry as well as in human-robot interactions. It is therefore important that we strive to achieve the best performance for the state-of-the-art salient object detection systems. Borji et al. [4] state that a good saliency detection model must have high recall and at the same time have a low false positive rate, the generated saliency maps must locate and fit the salient object accurately and that the efficiency of such a system must be high as well. One such state-of-the-art salient object detection model is the 'Deeply Supervised salient object detection with Short connections', the DSS architecture proposed by Hou et al. [13].

Although the DSS model performs well for both simple as well as complex datasets, there is still room for improvement. The failure case analysis of the model by Hou et al. shows three difficult cases for which the DSS model is unable to accurately segment the most salient object. They are shown in Figure 1.2. The first case is that the most salient object in an image is not segmented completely. For example, in row 1 of Figure 1.2, the body of the cat is not entirely recognised as salient as it has low contrast with the background. The second case is that the main body of the salient object is not segmented or non-salient objects are recognized to be salient. In a cluttered scene shown in row 2 of Figure 1.2, non-salient objects like the chair, some wall hangings and items on the table are highlighted in the saliency map. This could owe to the possibility that there are in fact multiple salient objects in the image which could be ranked in order of their saliency. However, the objects highlighted as salient are not recovered in their entirety, i.e. certain portions of the objects are blurred into the background and their contours are not sharp. The third case is the one in which transparent objects are segmented partially, particularly when they overlap other objects. Row 3 in Figure 1.2 shows

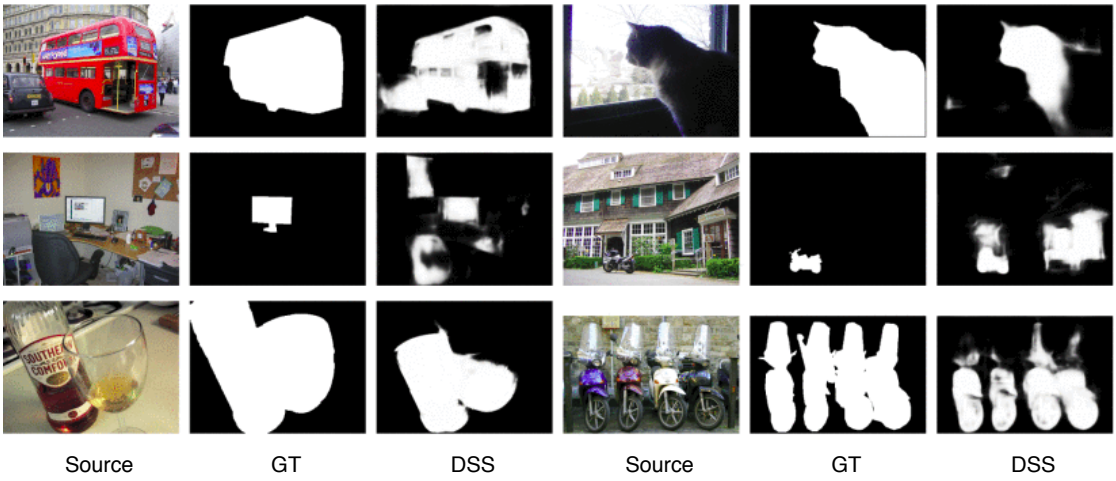



Figure 1.2: Sample images for which the DSS model could not accurately segment the most salient object (Recreated from [13]). Low contrast scenes, cluttered backgrounds and transparent objects are challenging for the DSS model.

that transparent objects such as the windshields of motorbikes are not extracted as salient although they are a part of the salient object, the motorbikes. The human eyes can recognise the windshields as part of the motorbike and hence comparing the network's predicted result with a human annotated ground truth map would result in a lower recall value for the model. A common factor in the three cases is that the most salient object is not segmented completely and some portion of the object is left out. This would reflect as a low recall value in the performance metrics. The focus of the thesis is to handle the low recall problem while keeping the number of false positives low. In other words, the thesis aims at strengthening the probability of segmenting the most salient object precisely and in its entirety.

## 1.2 Research Question

In this thesis, the utilization of objectness attention for improving the performance of the salient object detection task is studied. The objectness attention value of a pixel in an image indicates its likelihood of being part of an object. The indication of an object's presence at a pixel could strengthen the probability of segmenting that pixel if it is part of the most salient object. AttentionMask by Wilms and Frintrop [48] is a state-of-the-art class agnostic object discovery system. It contains Scale-specific Objectness Attention Modules (SOAMs) which identify regions from an image scale that have a higher probability of containing an object. The SOAMs produce attention maps that indicate these promising regions using objectness attention values. The higher the attention value of a pixel, the greater is the probability of that pixel belonging to an object.

An approach to incorporate objectness attention information into the salient object detection task for enhancing performance accuracy is proposed in this the-

sis. The SOAMs of AttentionMask, which predict objectness attention maps are integrated into the DSS model to form the novel AttentionDSS architecture. While AttentionDSS is a salient object detection system, its SOAMs must be trained for producing objectness attention maps. The SOAMs form an integral part of the object proposal generation model AttentionMask [48] and object proposal generation is highly related to salient object detection [4]. Thus, it is proposed that objectness attention prediction and salient object detection are related tasks. Therefore, the two tasks in AttentionDSS are trained in a multitask learning environment such that they share low level features. As there is no common dataset with annotations for both these tasks, they must be learned using two different datasets. The SOAMs are learnt using the COCO dataset [28], while the salient object detection task is learnt using the MSRA-B dataset [16].

## 1.3 Contribution of the Thesis

With an aim to improve the performance, a deep learning approach that utilizes objectness attention information for salient object detection is proposed in this thesis.
The following are the contributions of the thesis:

- an architecture for salient object detection using objectness attention

The following questions are answered in the thesis:

- whether objectness attention information improves the performance of the salient object detection task
- whether objectness attention prediction and salient object detection are related tasks

## 1.4 Organisation of the Thesis

The rest of the thesis is organised as follows. Chapter 2 discusses research work that has been conducted so far in the areas of object discovery, saliency and salient object detection. The different multitask learning approaches in deep neural networks are also explored in this chapter. Chapter 3 contains background theory that forms the basis of the proposed method. It lays a brief foundation for neural networks and convolution neural networks. The DSS model and AttentionMask, which form the core components of the proposed model, AttentionDSS are described in this chapter. The proposed method for performing the salient object detection task by utilizing objectness attention is elaborated with the detailed architecture of AttentionDSS in chapter 4. AttentionDSS is tested for its performance using the MSRA-B dataset and the qualitative and quantitative analysis of the results are discussed in the 'Experiments and Results' chapter. Certain ablation study results are also reported in this chapter. The final chapter concludes the thesis with the

inferences from the conducted experiments and what future work can be done in this direction.

# Chapter 2

# Related Work

The proposed thesis utilizes objectness information to improve the performance of salient object detection and this section therefore talks about associated topics. Section 2.1 of this chapter briefly covers object discovery methods that are useful for obtaining objectness information. Section 2.2 discusses saliency and fixation prediction methods that form the basis for salient object detection. The evolution of salient object detection methods from basic to state-of-the-art are discussed in section 2.3. Section 2.4 states the existing methods and neural network architectures used for multitask learning.

## 2.1 Object Discovery

Objects are defined as 'standalone things with a well-defined boundary and center, such as cows, cars, and telephones, as opposed to amorphous background stuff, such as sky, grass, and road' by Alexe et al. [2]. Object discovery could be defined as class-agnostic object proposal generation which is a step necessary for most object detection methods [48]. Object discovery methods discover all candidate objects in a scene or an image by either drawing bounding boxes around them or by segmenting them with pixel-precise accuracy. Object discovery could be done using hand-crafted features as proposed in [12, 37, 45, 47] or by using deep learning approaches such as those proposed in [35, 36, 38, 14].

#### Hand-crafted features based approaches

Uijlings et al. [45] propose a selective search object proposal method in which image segments are grouped based on multiple similarity measures, seed points as well as colour spaces. The diverse grouping strategies help improve the recall by enabling the capture of rigid, non-rigid and amorphous objects. As proposed in [12], candidate objects can be generated from a cluttered scene using multi-scale saliency computation combined with superpixel segmentation. In this approach, the image is segmented into superpixels (coherent image parts) using the segmentation algorithm by Felzenszwalb and Huttenlocher [7] which is based on the Gestalt principles: proximity and similarity of pixels. The superpixels are then selected

and combined to form candidate objects based on whether the region they belong to in the image is salient or not. The attention or saliency is computed using a scale space structure for color and intensity features of the image. The authors Horbert et al. [12] have applied this approach to frames of a video and tracked candidate regions to discard those with inconsistent boundaries over time. Werner et al. [47] propose an object discovery method in which salient regions computed in an image form seed segments and surrounding segments are added to them based on their similarity in order to form candidate objects. The candidate objects are further filtered using measures based on Gestalt principles such as symmetry, contrast and good continuation. Pont-Tuset et al. [37] propose a Multiscale Combinatorial Grouping algorithm in which segmentations obtained at multiple scales of the image are aligned to one scale and combined, thus making the segmentation of multi-sized objects possible.

### Deep-learning based approaches

Pinheiro et al. [35] propose DeepMask, a convolutional two-branched neural network to simultaneously predict a segmentation mask for an image patch and a score representing the likelihood of the image patch to fully contain an object. The deep learning method used outperformed the then state-of-the-art object proposal generation methods as the features learnt were not limited to hand-crafted ones. The score for an image patch was higher if the object was placed centrally and was fully contained in it. Pinheiro et al. [36] propose SharpMask, a successor of DeepMask that generates finer and more accurate object segments by introducing a top-down refinement network. The top-down network comprises of refinement modules that merge segment masks of each layer with their corresponding feature map obtained from the original bottom-up network. This top-down approach results in the successive upsampling of the segment mask to the size of the original image while incorporating finer spatial information from the lower layers into the coarser segment masks from the deeper layers. Qiao et al. [38] propose ScaleNet, a deep neural network which predicts a finer scale range based on the object size range for a specific application such as supermarket products. The input images are scaled to the predicted scales and fed as input to object proposal networks such as SharpMask, which makes the object proposal system independent of fixed scales and would thereby contribute to better recall. In FastMask, Hu et al. [14] propose one-shot learning for multi-scale object segment proposal. In the one-shot learning approach, the expensive convolutional operation on a multi-scale image pyramid is replaced by convolutional operation on a multi-scale feature pyramid. The overall architecture comprises of a body (extracts semantic features from the original image), residual necks (creates a feature pyramid through pooling and convolutional layers) and an attentional head (computes attention maps of sliding windows obtained from the feature pyramid maps and generates a segment mask). The one-shot learning makes the proposal recall significantly fast (2 to 5 times) and thus real-time. The multi-scale feature pyramid makes the segmentation of different sized objects possible.

## 2.2 Saliency

The 'Feature Integration Theory of Attention', an influential psychological model on visual attention by Treisman and Gelade [43] proposed in 1980 states that objects are mostly detected by the human visual system with the help of both focused attention and top-down information. Focused attention is directed serially to spatial locations in a scene so that separable features characterising an object in that location may be processed in parallel and integrated to identify or localise that object. In redundant or familiar environments past experience or knowledge acts as top-down information or top-down features whose presence is checked for in the image in order to identify an object. Focused attention also known as bottom-up attention corresponds to saliency. A salient object is often the most conspicuous object in a visual scene that is attractive due to its significant visible features. Detection of the most salient object is done using bottom-up cues such as the object's uniqueness and contrast with its neighbours, which makes it an automatic and purely image-driven process. Top-down attention on the other hand is used to shift attention based on previous knowledge, task information such the task goal or the expectation of a certain outcome. Often bottom-up and top-down attention work together to identify significant objects in a scene.

The first computational attention model was proposed by Koch and Ullman [21] in 1987 and is based on the 'Feature Integration Theory of Attention'. A computational attention system consists of a bottom-up attention module (saliency model), top-down information and a model to find the Foci Of Attentions (FOAs) from a saliency map. A Winner-Take-All (WTA) network, also known as a neural maximum finder is used to compute the FOAs or the most active elements (neuron or pixel) in a locally connected parallel architecture [21]. On serial machines the WTA may be replaced by a serial maximum finder. Tsotsos et al. in 1995 through their work in [44] emphasized the significance of selectively tuning a visual search model with the help of top-down task-specific attentional bias and a WTA network. This biologically inspired work contributes towards optimising the search procedure, particularly in robotic vision applications by reducing the number of candidate image and feature subsets.

Itti et al. [15] in 1998 published a computational system for bottom-up attention which follows the model structure proposed by Koch and Ullman [21]. This model is one of the best known approaches in computational visual attention. In this model, each feature is considered independently and a scale space representation i.e. a Gaussian pyramid is computed for each feature channel. The features are the color, intensity or orientation values in an image. Multiple centre-surround contrast feature maps are computed for each feature channel. Across-scale addition is used to combine the feature maps within each feature channel, the resulting maps are the conspicuity maps for each feature channel. The conspicuity maps are normalised and summed to form the final saliency map. The key element in this saliency model is the centre-surround contrast within a scale-space representation of the image features. The lower scales highlight the finer details in an image while the higher scales show coarser details such as contours of larger objects. A centre-

surround contrast computes the difference between a finer (centre) scale and a coarser (surround) scale in a scale-space representation of a feature. This contrast computation helps detect locations that stand out from their surrounding i.e. it extracts salient objects.

The period between 2000 and 2008 saw the development of several saliency systems that were based on the Itti-Koch model [15] such as the 'Visual Object detection with a CompUtational attention System' (VOCUS) by Frintrop [8]. The VOCUS system incorporates top-down attention into the bottom-up Itti-Koch saliency model for a goal-oriented object search. The top-down information are the properties of the target and the background which are learned in order to direct focus only on the potential regions of interest. In 2009 Achanta et al. [1] proposed a model in which the most salient region in an image was detected by identifying the most frequent content in an image. The most frequent content was computed by taking the difference between the image average and the Gaussian blur of the image. Although the approach is simple and gives quite precise saliency maps, it is not suitable for cases where the salient object is not different from the image average such as a grey element on a black and white background. The model considers only luminance and color features and misses to incorporate features such as motion and orientation.

The Bonn Information-Theoretic Saliency (BITS) saliency model by Klein and Frintrop [19] is based on the bottom-up Itti-Koch attentional model wherein the center-surround contrast is determined by computing the Kullback-Leibler Divergence (KLD) between the centre and surround regions' feature distributions. The Continuous Distributions (CoDi) saliency model by Klein and Frintrop [20] is an extension of the BITS model wherein the discrete histograms (feature distributions) are replaced by continuous normal distributions and the KLD is replaced by the Wasserstein distance. Zhu et al. [52] compute local and global saliency in an image by taking the centre-surround contrast between superpixels. The most salient region in an image can be computed in an information-theoretic manner. Entropy is one way to measure the amount of information content in an image, the lower the amount of information content in an image, the lower is the entropy. Entropy can be represented as a probability distribution of a feature (e.g.intensity) in an image. The Kullback-Leibler Divergence (KLD), also known as information divergence, information gain or relative entropy computes the difference between the entropy of an image region to that of its surrounding. This concept corresponds to the basic principle of the centre-surround contrast used to compute saliency. In 2015, Frintrop et al. proposed VOCUS2 [9] which is an improved version of the VOCUS system wherein twin pyramids i.e. a centre Gaussian pyramid and a surround Gaussian (larger sigma) pyramid pair is computed for each feature. The centre-surround contrasts for each feature are computed between the corresponding scales from the centre pyramid and the surround pyramid of that feature.

## 2.3 Salient Object Detection

Itti et al. [15] compute the several salient regions in an image by highlighting local regions with high contrasts to its immediate surrounding. Differing slightly from this idea, salient object detection is a method in which the most salient object(s) in an image is extracted out from its background by pixel-precise segmentation or by drawing a bounding box around the object. Salient object detection is also known as salient object segmentation when the resulting output of the task is a binarized saliency map that highlights the most salient object with pixel-precise boundaries. Achanta et al. [1] explain salient object segmentation as an application of salient region detection and thereby propose two approaches to achieve this. The first approach involved the binarization of saliency maps for all the saliency values ranging from 0 to 255 as threshold values. The precision recall curve for each of these threshold values are compared to select the best binarized saliency map. The second approach is that of finding an adaptive threshold value that is dependent on the saliency value of an image. This value corresponds to twice the average saliency value of the image. Liu et al. [30] in 2011 proposed a model for salient object detection where each pixel is labelled as salient or not by combining several local and global features using Conditional Random Field (CRF) learning. The features such as multi-scale contrasts and centre-surround histograms are used to describe a salient object. The detected salient object is indicated by drawing a bounding box around it.

From the year 2014, saliency systems based on Convolutional Neural Networks (CNNs) started being developed and has been gaining momentum ever since. In [51] by Zhao et al., local and global context in an image obtained using deep CNNs are combined in order to obtain an enhanced saliency map that highlights the most salient object. The local context is useful in cases where low-level saliency cues do not result in accurate saliency maps for images with low-contrast backgrounds. The Deep Contrast Network by Li and Yu [25] proposes a two stream architecture for salient object detection. One stream is a multi-scale convolutional network that uses pixel level contrast to produce a saliency map, this map however is blurry at the object contours. The saliency map from the second stream, a spatial pooling stream that operates at superpixel or segment level better models object boundaries. Maps from both streams are combined. The resulting map has finer boundaries and the end-to-end model therefore shows how segment features can help extract finer contours of salient objects.

The salient object detection model, DeepSaliency by Li et al. [26] is composed of a multi-task Fully Convolutional Neural Network (FCNN) and a saliency refinement component. The FCNN is trained for two tasks - saliency prediction as well as semantic object segmentation. The resulting maps from the two task branches of the FCNN are fused to form a coarse-grained saliency map. As the model learns the object segmentation task, it has better object perception abilities which makes the saliency map more accurate. The refinement component is a Laplacian regularised non-linear regression scheme that enhances the saliency map by producing a more detailed map with finer object contours. In [46] Wang et al. propose a salient

object detection model that leverages a Recurrent Fully Convolutional Network (RFCN) architecture to generate a saliency map and refine it by correcting errors using the recurrent connection from previous outputs. Lee et al. [23] combine high level features obtained from a deep CNN with a distance map that encodes distances between superpixels for low level features such as color and orientation. The combination of high level and low level features contributes towards saliency maps that extract the most salient object with better precision. This method is especially useful for images with low contrast and cluttered backgrounds. Liu and Han propose Deep Hierarchical Saliency Network (DHSNet) [29] for salient object detection which follows a similar principle as stated above, wherein global and local context are combined to result in a saliency map. In the DHSNet, a global saliency map is obtained using a deep CNN and this map is then refined by integrating local features in hierarchical manner. The hierarchical architecture presents better performance as well as efficiency in generating saliency maps. Hou et al. [13] extend the existing Holistically-Nested Edge Detector (HED) [49] for salient object detection by introducing short connections from shallower to deeper layers in the network. This architecture is hierarchical and utilises global and local features to obtain precise saliency maps. As shallower layers in the deep CNN lack regularity i.e. they miss details and deeper layers are messy due to excessive fine details, the short connections between these layers take the advantage of both global information such as object locations or large object contours from the shallower layers as well as local information such as finer object details from the deeper layers, thereby resulting in a more accurate saliency map. The model is state-of-the-art and works well for both simple and cluttered image datasets.

Chang et al. [5] proposed a computational mode that uses saliency to detect objects and exploits object detection to estimate saliency. The graphical model integrates objectness and saliency to improve both saliency prediction as well as object detection. The predicted saliency map indicates the most salient object by drawing bounding a bounding box around it. Jiang et al. [17] propose a salient region detection algorithm that effectively combine three complementary image features: Uniqueness, Focus and Objectness (UFO) to generate precise saliency maps. Uniqueness is the visual contrast in an image which is a feature that most saliency algorithms rely on, focus is the degree of visual sharpness in an image region and objectness is the likelihood of an entire object being present in an image region centred around a pixel. The resulting map that detects the salient region is binarised using adaptive thresholding to extract the most salient object. Srivatsa and Babu [42] generate saliency maps that highlight the most salient object by assigning saliency values to foreground regions obtained from object proposal maps. Ye et al. [50] use a graph based method to integrate a saliency map with its objectness map to achieve salient object segmentation. The above models that bind objectness with saliency have shown consistent improvement in salient object detection and segmentation results. The failure cases in these models pertain to either the quality of the objectness measure, the quality of the saliency system or that of the fusion method. The analysis of these works therefore suggest that a effective fusion of the best objectness measure and the best saliency system

would generate a fine salient object detection/segmentation system. Therefore, a salient object segmentation model that integrates the state-of-the-art saliency system by Hou et al. [13] with the objectness generating Scale-specific Objectness Attention Map (SOAM) component of the state-of-the-art object detection system AttentionMask by Wilms and Frintrop [48] is proposed in this thesis.

## 2.4 Multitask Learning

If trained on a large dataset for several epochs, a neural network would tend to overfit on that particular dataset. One way to ensure that a network generalizes well to other datasets is by using Multitask Learning. As stated by Goodfellow et al. [10], in a typical multitask learning scenario, two or more tasks are learnt on the same network with the assumption that the tasks are related and share the same input. One could make the assumption that two tasks are related based on the knowledge that they are drawn from the same distribution or by the logical analysis of the given tasks.

The typical neural network architecture for a multitask learning set up is as shown in Figure 2.1. The tasks share a common base network, while each having its individual network branch that learns task-specific features. The sharing of the base network parameters enables the tasks to learn common low level features from the input. This is known as hard parameter sharing. Every task can have an influence on the parameter values of the base network, allowing the network to generalize well. It is also possible to initialize commonly used base networks such as the VGGNet [41] or ResNet [11] with their pre-trained weights and fine tune their last few layers. This would allow a smooth transition of the learning between the base network and the task branches. This is known as transfer learning. Soft parameter sharing is another approach of multitask learning wherein each task has its own model, but the base networks of the models are connected such that the relationships between the tasks are learnt and the parameters are adapted to be similar.

Deep Relationship Networks [31] is an approach that follows hard parameter sharing wherein certain convolution layers are shared by tasks while each task has its specific branch of fully connected layers. In addition, the task specific branches are connected with each other by matrix priors that learn the relationship between these tasks. This can be considered as an example of soft parameter sharing. Cross-Stitch Networks [33] use soft parameter sharing wherein cross stitch units are used to learn the relationship between tasks by taking the linear combination of the layers in the two task models. Certain networks such as the SLUICE network [40] have the ability to decide which parameters of the network model must be shared and to what extent. This is particularly useful in case of loosely related tasks where it is hard to determine which parts of the network can be shared. Fully adaptive feature sharing [32] is another approach that allows the dynamic grouping of similar tasks during the training process. Kendall et al. [18] train a multitask network using a single loss function that incorporates all the task losses

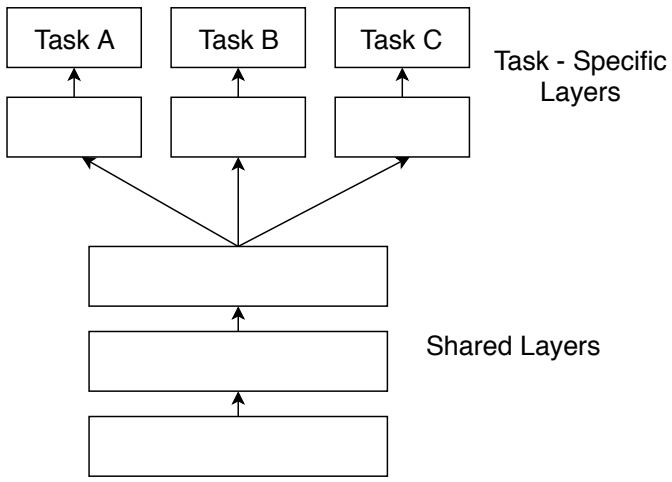



Figure 2.1: Hard parameter sharing in deep neural networks (Recreated from [39]).

by weighing them based on their uncertainty values.

# Chapter 3

# Background Theory

Chapter 2 reviewed the works that have been done so far in the fields of object discovery, salient object detection and multitask learning using neural networks. This chapter explains the core concepts and systems that form a part of the method proposed in this thesis.

## 3.1 Artificial Neural Network

An artificial neural network, simply known as a neural network is a deep learning model that is loosely inspired by the biological nervous system. It consists of neurons that are connected with one another such that data flowing through them undergoes meaningful transformations in order to learn a concept. The learnt model can be used to make predictions on previously unseen data.

Neural networks follow a supervised way of learning, i.e. they learn from data that has been annotated with the correct label. For example, a neural network model could be trained on several images of streets annotated with bounding boxes around pedestrians in order to be able to detect pedestrians crossing streets. Such a model could have an application in self-driving cars. A typical neural network architecture is shown in Figure 3.1. It consists of an input layer, one or more hidden layers and an output layer. Each layer consists of computational units or neurons that receive inputs from previous layers and produce a certain output value. All the neurons in a layer put together would thus resemble a vector of values. The data on which the network is trained is fed to the input layer. This data undergoes transformations in the hidden layer units while they are transported through the network to the output layer. The output layer gives the final predictions of the network.

Given a certain labelled dataset, a neural network learns a mapping between the data $x$ and its associated label $y$ in such a way that when the network is provided with previously unseen data it would predict the appropriate label. In order to be able to learn the mapping between the data and their associated labels, a neural network must learn weights $\theta$ that satisfy the mapping function $f$. The

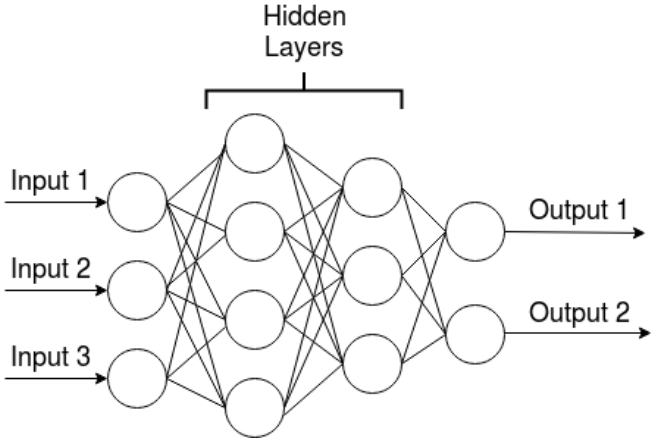



Figure 3.1: A typical neural network architecture.

mapping function is defined as

$$y = f(x; \theta) \tag{3.1}$$

$\theta$ consists of $W^T$, the transposed weight matrix and $b$, the bias. The bias is usually a small constant that is used to ensure that the activation $z$ of a neuron is never zero. The typical linear mapping function learnt by a hidden unit of a neural network is defined as

$$z = W^T x + b, \tag{3.2}$$

In order to solve non-convex problems, non-linear mappings are required. In neural networks linear transformations in the computational units are made non-linear by using activation functions. Some of the commonly used activation functions are the Rectified Linear Unit (ReLU) and the Sigmoid functions as shown in equation 3.3 and 3.4, respectively.

$$g(z) = max(0, z) \tag{3.3}$$

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{3.4}$$

Another commonly used activation function particularly for predicting the probabilities of a multinomial distribution at the output units of a network is the softmax function. The softmax activation at the $i$th output unit is

$$softmax(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{3.5}$$

A neural network trains on the data for one or more epochs, i.e. until the task has been learnt. In every epoch, the entire training data is passed through the network once. This is done in iterations, where in each iteration, one or more data samples would be passed forward through the network. At the end of the forward pass phase of an iteration, the loss is computed. This loss represents the difference between the network's predicted output for the training sample $x^{(i)}$ and its true label $y^{(i)}$, for

the network weights $\theta$. The cost function, also known as the optimization function of a model is often calculated as the average of the per example loss. It is defined as

$$J(\theta) = \frac{1}{m} \sum_{1}^{m} L(x^{(i)}, y^{(i)}, \theta), \tag{3.6}$$

where L is the per example loss. The per example loss could be a negative log likelihood which can be defined as

$$L(x, y, \theta) = -\log p(y|x; \theta) \tag{3.7}$$

After the loss has been computed, each weight in the network is updated by a certain value. This value is the derivative of the loss with respect to each of the weights. It is known as the gradient. The gradient computation is done by the backpropagation algorithm, its name indicating the task it does, i.e. computing the gradient of the loss with respect to all the weights and propagating it backwards through the network [10]. The weights of the network $\theta$ are updated with the computed gradients $\nabla_\theta J_\theta$ using an optimization algorithm such as Stochastic Gradient Descent (SGD). The update function is as given below

$$\theta = \theta - \epsilon \nabla_\theta J_\theta, \tag{3.8}$$

where $\epsilon$ is the learning rate. In order to successfully train a neural network, the provided labelled dataset is split into train, validation and test sets. The training data is used by the network to learn a task. If the networks fits to the training data perfectly, it would be unable to generalize well for unseen data. In order to ensure that the network does not overfit on the training data, several regularization methods could be used. One such method is early stopping, in which a network is tested on the validation dataset after every epoch of training and the the model where the error on the validation set is the least is chosen as the final model. The prediction made by the chosen model on the unseen test data split determines the network's performance. In the proposed thesis, the optimal weights are chosen by validating the network performance on the validation data split. Another method used to reduce generalization error is to add a regularization term to the cost function. The cost function with the regularization term is

$$J(\theta) = \frac{1}{m} \sum_{1}^{m} L(x^{(i)}, y^{(i)}, \theta) + \lambda \sum_{1}^{m} \theta^2 \tag{3.9}$$

When the regularization term is the sum of square of all network weights as shown in the above equation, it is known as the L2 regularization method. The $\lambda$ parameter must be chosen optimally such that it regulates the weights to smaller values which would in turn lead to a simpler non-overfitting hypothesis. Learning multiple related tasks simultaneously such that they share a common base network and its parameters is known as multitask-learning. It is a kind of regularization method where the base network can generalize to multiple related tasks, thereby avoiding

overfitting while improving space and time efficiency. Multitask learning is key in the proposed thesis, where a salient object detection system and an objectness attention generation system form the two related tasks that share a common base network.

## 3.2 Convolutional Neural Network (CNN)

Neural networks in which a neuron of a certain layer receives a weighted input from every neuron in the previous layer are known as fully connected networks. In such networks, the output generated by the previous layer is matrix multiplied with the weights so as to form the input to the next layer of neurons. Neural networks in which a convolution operation is done instead of a matrix multiplication is known as a Convolutional Neural Network (CNN). Given a two dimensional image $I$ and a two dimensional weight matrix $K$, also known as the kernel or filter, the result of the convolution operation at the $ith$ row and $jth$ column of the resulting map $G$ can be calculated as

$$G(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)) \tag{3.10}$$

Unlike the matrix multiplication operation where matrices must comply with the size requirements, the weight matrix or the kernel in a convolution operation can have a size smaller than that of the input. In this case the weights can be shared across the input matrix by sliding the weight matrix across the input by a certain step size known as the stride. As a result of this property, fewer weights need to be learnt by the network, thereby improving the computational efficiency in terms of space and time. The sharing of parameters across the input matrix makes the network translation invariant. For example, if a kernel that filters or identifies vertical edges is shared across an input image, it would identify all vertical edges appearing in the image irrespective of their location. This is because the kernel would learn weights for vertical edge detection with the help of immediate local spatial information and not the information from the complete input space.

Pooling is another operation used in CNNs for making the network invariant to small translations. A pooling operation summarizes an image region to single value, as a result of which the network does not fit exactly to the input image region, it rather learns a generalized representation of the features in that region. For example max pooling represents an image region by selecting the the maximum value in an that region. Average pooling is another commonly used pooling operation. As the pooling results in a concise representation of an image region, it reduces computational time and space complexity.

A typical convolutional neural network consists of alternating convolution and pooling layers. The result of convolving a filter with an input matrix is an activation map which forms the input to the next layer. The architecture is as shown in Figure 3.2. CNNs are commonly used for image processing tasks such as image classification, object discovery and object recognition.

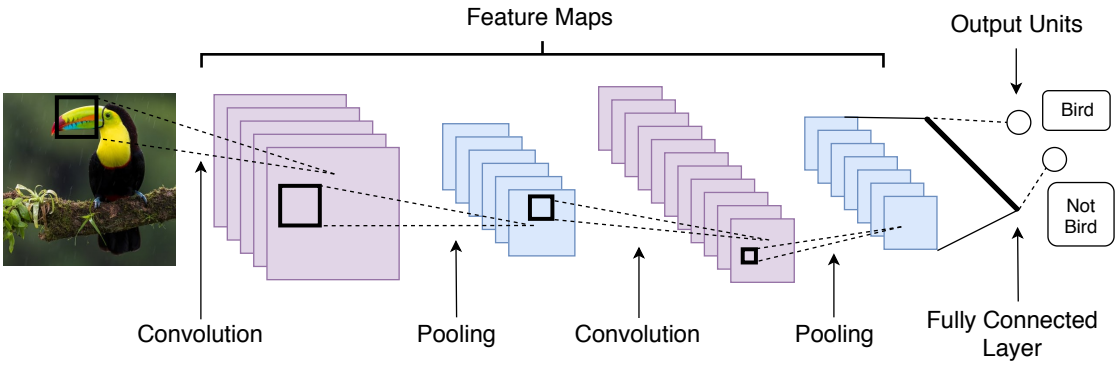



Figure 3.2: A typical CNN architecture. In this example, an image is classified into one of the 2 classes: 'Bird' or 'Not Bird'.

## 3.3 VGG Net

In 2014, Simonyan and Zisserman [41] proposed a very deep convolutional neural network architecture known as the VGG Net. They demonstrated that increasing the depth of convolutional neural networks by adding more convolution layers in the network corresponds to higher performance accuracy. They also state that a deeper network has the ability to generalize better to different tasks and datasets. In order to make deeper convolutional networks computationally feasible and efficient, smaller convolution filters are used. VGG-16 and VGG-19, variants of the VGG network comprising of 16 and 19 convolution layers, respectively are two of the architectures used commonly as base networks for large scale image related tasks. An overview of the VGG-16 and VGG-19 network architectures is shown in Table 3.1.

## 3.4 FastMask

Hu et al. [14] in 2017 proposed the FastMask architecture for CNN-based object proposal generation. It has a hierarchical structure that enables the utilisation of features from multiple scales of the input image for generating object proposals. As a result, objects of different sizes within an image can be detected. Most object proposal systems before FastMask used a multi-shot paradigm in order to segment objects of different scales within an image. In a multi-shot paradigm, an input image is scaled to different sizes and a model trained to extracts objects in an image is applied to each image scale. The trained model comprises of a body network followed by a head unit. The body network consists of convolution layers that compute feature maps from each image scale and the head unit segments objects from the generated feature maps. In such an architecture, applying the computationally expensive convolution operation on each scale of the input image is a redundant operation. FastMask therefore uses a one-shot paradigm, in which the body network is applied to the input image once in order to generate feature

| ConvNet Configuration | | |
|---|---|---|
| 16 weight layers | 16 weight layers | 19 weight layers |
| conv3-64 | conv3-64 | conv3-64 |
| conv3-64 | conv3-64 | conv3-64 |
| maxpool | | |
| conv3-128 | conv3-128 | conv3-128 |
| conv3-128 | conv3-128 | conv3-128 |
| maxpool | | |
| conv3-256 | conv3-256 | conv3-256 |
| conv3-256 | conv3-256 | conv3-256 |
| conv1-256 | conv3-256 | conv3-256 |
| | | conv3-256 |
| maxpool | | |
| conv3-512 | conv3-512 | conv3-512 |
| conv3-512 | conv3-512 | conv3-512 |
| conv1-512 | conv3-512 | conv3-512 |
| | | conv3-512 |
| maxpool | | |
| conv3-512 | conv3-512 | conv3-512 |
| conv3-512 | conv3-512 | conv3-512 |
| conv1-512 | conv3-512 | conv3-512 |
| | | conv3-512 |
| maxpool | | |
| FC-4096 | | |
| FC-4096 | | |
| FC-1000 | | |
| soft-max | | |

Table 3.1: The variants of the VGGNet (Recreated from [41]).

maps. A neck module is introduced by FastMask which scales each feature map to different sizes so as to form a scale pyramid. A head module is used to segment objects from the several scales of each feature map's scale pyramid.

The FastMask architecture as shown Figure 3.3(b) mainly comprises of three components: a bodynet or basenet, a feature pyramid and an attentional head. The input image is first fed through the basenet, which is ResNet in the case of FastMask. The downsized feature map obtained from the basenet is downsampled further to several scales by the neck module in order to form a feature pyramid. The neck module comprises of a residual component ($3 \times 3$ conv followed by $1 \times 1$

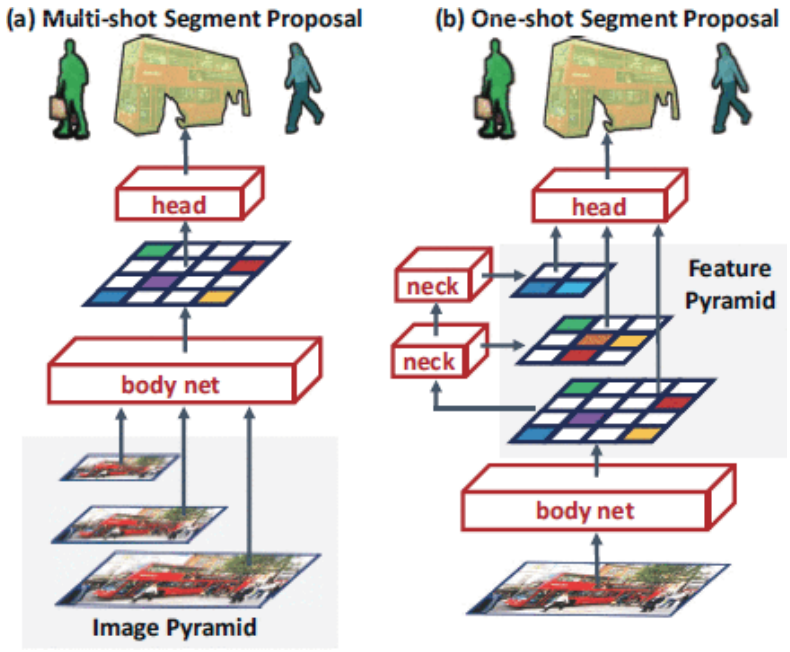



Figure 3.3: The multi-shot and one-shot object discovery paradigm [14].

conv) followed by an average pooling layer. A window of a fixed size is sampled over the scales in the feature space. This makes it possible to detect objects of several sizes, i.e. the system is scale invariant. The attentional head computes attention maps for each of the sampled windows. The best $K$ windows are selected based on the likelihood of an object being fully contained and centred in the window. Irrelevant parts of the selected windows are pruned out to ensure that the object is centred in the window. These windows are then subjected to a segmentation module that extracts the object in the window. AttentionMask uses FastMask as its core with the key difference between the two object proposal systems being that in AttentionMask only the most promising windows are sampled from the feature pyramid, resulting in fewer resource consumption and hence, increased efficiency.

## 3.5 AttentionMask

AttentionMask by Wilms and Frintrop [48] is an object proposal generation system that uses attention to increase computational efficiency and utilise the saved resources to instead focus on the detection of very small objects. AttentionMask is modelled after FastMask which uses a one-shot paradigm to generate object proposals from an image. The AttentionMask architecture is shown in Figure 3.4. It uses ResNet as a basenet. The last feature map obtained from the basenet is downsampled to several scales forming a feature pyramid. The downscaling is done with the residual neck components used in FastMask. Each of the obtained scale-specific feature maps form the input to the attention modules known as the Scale-specific Objectness Attention Modules (SOAMs). Each SOAM computes a scale-specific

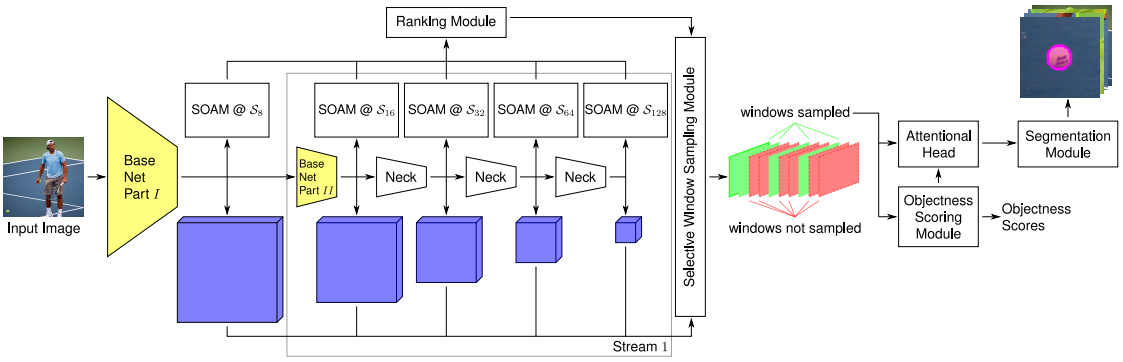



Figure 3.4: The AttentionMask architecture (Image from [48]).

attention map that highlights regions of the feature map that are likely to contain objects. The attention map denotes regions that have a higher probability of containing an object with a higher attention value. Fixed sized windows are sampled by sliding across objectness attention maps of all scales. However, unlike FastMask, not all windows are selected for further processing. Only promising windows, i.e., windows with a higher probability of containing an object (based on their attention value) are selected. This window selection step based on associated attention value discards windows not likely to contain objects, thereby saving time and memory that would have otherwise been necessary for their further processing. Objectness scores are computed for each of the selected windows using a sub-network from FastMask. The objectness score indicates the likelihood of an object to be fully contained within the window. Windows with objectness score higher than a threshold value are retained. The attentional head module from FastMask is used to prune out background from the retained windows. A pixel-precise segmentation module extracts the object from the pruned windows.

The SOAM is the key feature of AttentionMask. It uses attention to highlight image regions that have the likelihood of containing objects. As each SOAM is trained for a specific scale, it is possible to determine the likelihood of the presence of objects of different sizes. The SOAM architecture is as shown in the Figure 3.5. Each SOAM comprises of a $3 \times 3$ convolution layer with 128 channels and a ReLU activation, followed by a $4 \times 4$ convolution layer with 2 channels. A softmax layer is then applied to determine the probability of a pixel belonging to an object at that scale. The resulting output is a scale-specific objectness attention map that highlights the presence of the detected object. Pixels in the feature map with a higher probability of belonging to an object at that scale are highlighted and thus have a higher attention value. These attention values are used to select only promising windows for object segmentation in AttentionMask, thereby saving computational resources. The spared resources are used to add an additional SOAM for detecting very small objects such as tennis balls in a match scene or small glasses in a cluttered kitchen scene. Such small objects, particularly in cluttered backgrounds are usually not detected by most object proposal systems. This is a another special feature of AttentionMask.

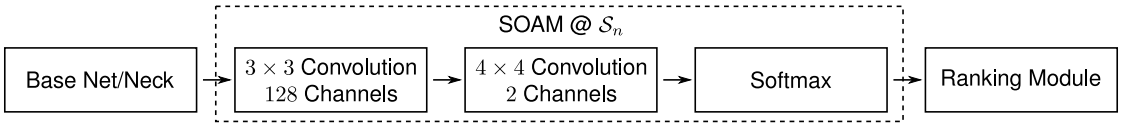



Figure 3.5: The SOAM architecture (Image from [48]).

In order to train the SOAMs, a binary cross entropy loss between the predicted objectness attention map $y'$ and the ground truth map $y$ is computed as follows:

$$L(y, y') = y. - log(\sigma(y')) + (1 - y). - log(1 - \sigma(y')), \tag{3.11}$$

where $\sigma$ is the sigmoid function. The ground truth map for each SOAM is a binary map that matches the SOAM's scale size and contains a value of 1 for every pixel that belongs to an object of that scale. The background pixels are set to a value of 0. There is a class imbalance for certain SOAM scale ground truths such as $S_8$, where the number of negative pixels (non-object pixel) is significantly larger than the number of positive pixels (object pixel). A simple binary cross entropy learning would result in a sub-optimal performance. Therefore, for every positive pixel, three negative pixels are sampled to form a collection of class balanced ground truth pixels $S$. Therefore, the loss for a SOAM is

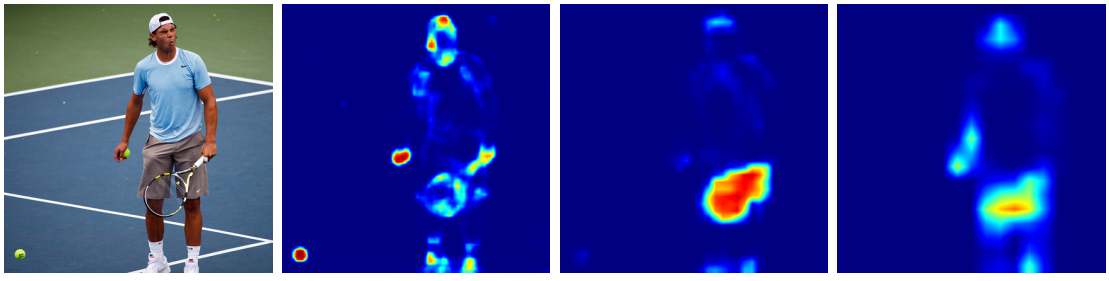
$$L_{att(a,a')} = \frac{1}{|S|} \sum_{(x,y)\epsilon S} L(a_{(x,y)}, a'_{(x,y)}), \tag{3.12}$$

where $L(a_{(x,y)}, a'_{(x,y)})$ is the binary cross entropy loss computed using Equation 3.11 between the ground truth scale-specific objectness attention map $a$ and the predicted objectness attention map $a'$.

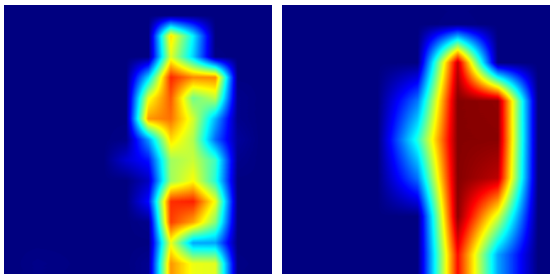
AttentionMask is a state-of-the-art object proposal generation system outperforming earlier object discovery models such as DeepMask [35], SharpMask [36] and FastMask [14] in terms of average recall as well as computational efficiency with respect to time. As shown in Figure 3.6, the attention map generated by the SOAM at scale $S_8$ highlights the smallest objects in the given input image such as the the tennis ball and the hand of the player. Scale $S_{128}$ highlights the largest object in the image, which is the tennis player. In this thesis, attention values at all positions of scale-specific attention maps are used as weights to reinforce the accuracy of the salient object detection task.

## 3.6 Deeply Supervised Salient Object Detection with Short Connections

Deeply Supervised Salient Object Detection with Short Connections (DSS) is a salient object detection architecture proposed by Hou et al. [13]. The architecture utilizes features extracted at multiple scales of an image in order to detect the most salient object. In CNNs, shallow layers detect fine patterns in images while deeper



(a) Input (b) $S_8$ (c) $S_{16}$ (d) $S_{32}$



(e) $S_{64}$ (f) $S_{128}$

Figure 3.6: The objectness attention maps (b), (c), (d), (e), (f) generated by the SOAMs of AttentionMask for the input image (a) (Images from [48]).

layers can locate object boundaries. This inspired the drawing of short connections from deeper to shallower layers in the DSS model. As a result, the final saliency map comprising of features extracted at different scales.

The DSS architecture is as shown in Figure 3.7. It comprises of a VGGNet extended with an additional pooling layer (pool5) as the base network. A side branch is drawn from each of the 6 scales (including the 5 conv layers and the added pooling layer) in the extended basenet. Each side branch generates a saliency map that highlights the most salient object at that scale. Short connections are drawn from the deeper side branches to the shallower side branches of the network.

The structure of the short connections is as shown in Figure 3.8. The output from a deeper side branch is upsampled and concatenated with the output of the shallower side branch. The map obtained after the concatenation operation is further weighted using a conv layer to give the final saliency map at that scale. The shallowest side branch receives the most number of maps from the deeper layers for the concatenation operation. As a result, the shallowest side branch generates a saliency map that incorporates features from all the scales in the network. The saliency maps obtained from each of the side outputs is fused to form the final saliency map.

The activation $R_{side}^{(m)}$ produced at the $m$th side output is given by

$$R_{side}^{(m)} = \begin{cases} \sum_{i=m+1}^{M} r_i^m R_{side}^i + A_{side}^m & \text{,for } m = 1, ..., 5 \\ A_{side}^m & \text{,for } m = 1 \end{cases} \tag{3.13}$$

$M$ stands for the total number of side branches and $A_{side}^m$ is the activation of side

branch $m$. $R^i_{side}$ is the activation of the side branch that is deeper than the $m$th side branch and is weighted with a value $r^m_i$ to form a short connection with the $m$th side branch.

Each side branch's loss $l^m_{side}$ is computed as follows:

$$\begin{aligned} l^m_{side}(W, w^{(m)}) = -\sum_{z_j \epsilon Z} z_j \log Pr(z_j = 1|X; W, w^{(m)}) \\ + (1 - z_j) \log Pr(z_j = 0|X; W, w^{(m)}) \end{aligned} \tag{3.14}$$

Here, $W$ stands for the base network weights while $w$ are the weights of the $m$th side branch. The standard pixel-wise cross-entropy loss is computed between the ground truth binary map $Z$ and predicted saliency map for the training image $X$.

The total loss from all the side branch outputs is computed as

$$L_{side}(W, w, r) = \sum_{m=1}^{M} \alpha_m l^{(m)}_{side}(W, w^{(m)}, r), \tag{3.15}$$

where $\alpha_m$ is the weight for each side branch loss $l^{(m)}_{side}$.

The loss computed at the fusion layer is the cross entropy (denoted by $\hat{\sigma}(.,.)$) between the ground truth map $Z$ and the fused predictions. It is given as

$$L_{fuse}(W, w, f, r) = \hat{\sigma}(Z, \sigma(\sum_{m=1}^{M} f_m R^{(m)}_{side})), \tag{3.16}$$

where $f_m$ is the fusion weight for the $m$th side branch activation $R^{(m)}_{side}$. $\sigma(.)$ denotes the sigmoid operation. The final loss is summation of the total side loss $L_{side}$ and the loss computed at the fusion layer $L_{fuse}$. It is defined as

$$L_{final}(W, w, f, r) = L_{fuse}(W, w, f, r) + L_{side}(W, w, r) \tag{3.17}$$

To further improve the quality of the obtained saliency map, a fully connected conditional random field method known as PyDenseCRF [22] is employed during the test phase of the network.

The network is trained on 2500 images of the MSRA-B dataset. The DSS architecture outperforms the previous state-of-the-art models such as DRFI [16], MDF [24], DS [26] and DCL$^+$ [25] with an F-measure of 0.927 when tested on the MSRA-B dataset. The DSS model also outperforms the previous architectures on the multiple salient object datasets PASCAL-S [27] and SOD [34] with an F-measure of 0.83 and 0.842, respectively. Figure 3.9 show the side output maps as well as the fused map 'Results' of the DSS architecture for a sample image.

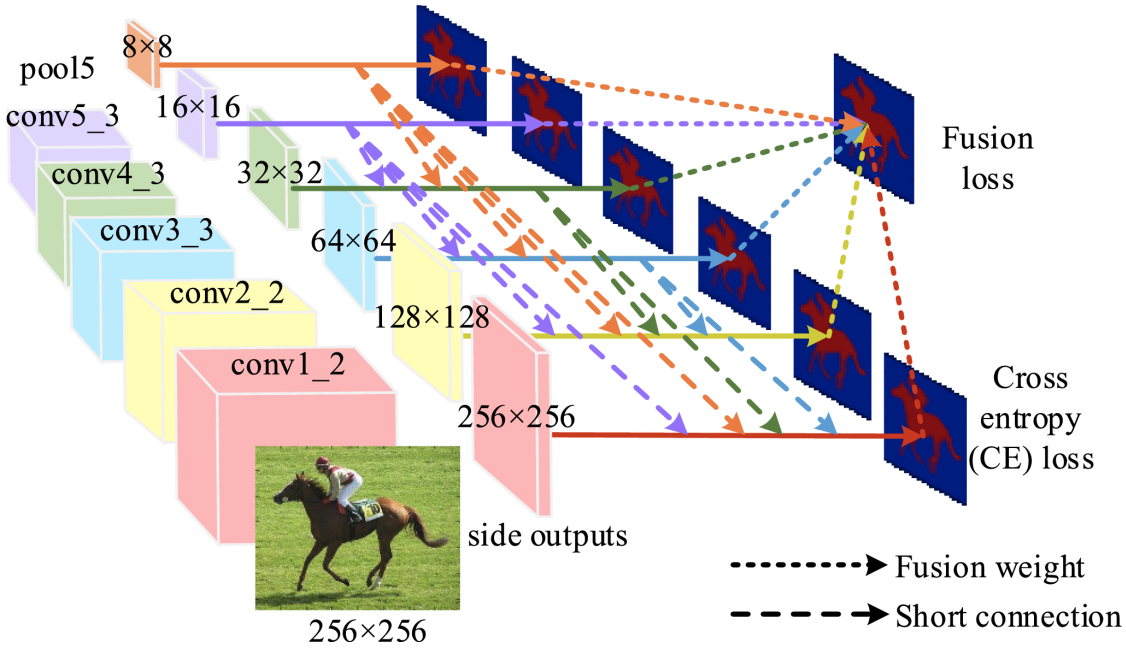



Figure 3.7: The DSS Architecture [13]. It comprises of the VGGNet as the base network and multiple side branches, each producing a saliency map as output. Short connections are drawn from the deeper side branches to the shallower side branches. The side output maps are fused to obtain the final saliency map.

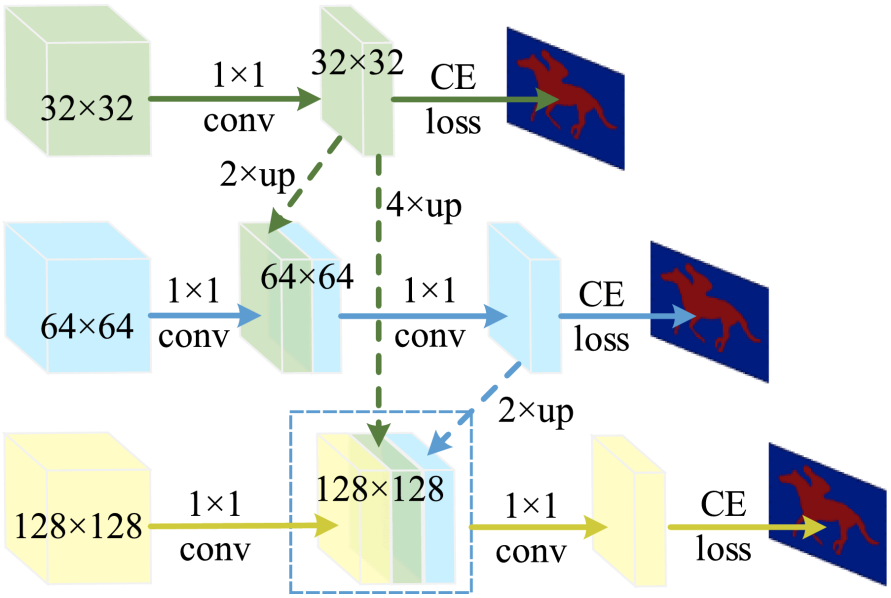
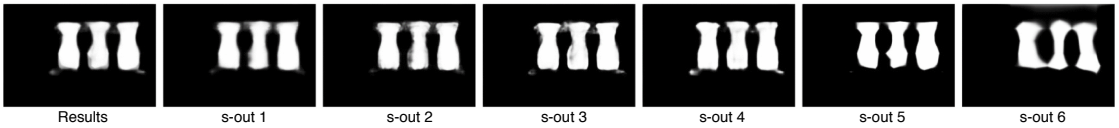



Figure 3.8: Short connections [13].


(a)




(b)

Figure 3.9: (a) The image and its ground truth saliency map. (b) The side outputs of the DSS architecture and the result of their fusion. [13]

# Chapter 4

# Salient Object Detection using Objectness Attention

Combining objectness attention with saliency prediction in order to improve the performance of the salient object detection task is proposed in this thesis. Therefore, the state-of-the-art salient object detection model, DSS by Hou et al. [13] is integrated with the Scale-specific Objectness Attention Modules (SOAMs) of the state-of-the art object discovery model, AttentionMask by Wilms and Frintrop [48].

The DSS model as described in section 3.6 detects and segments the most salient object in a given image. The model produces multiple saliency maps, each map being the output of a network branch that processes the image feature map of a specific scale size. The deeper network branches produce saliency maps from small scale image feature maps while the shallower network branches produce saliency maps from large scale image feature maps. The deep layers locate the objects well, while shallower layers detect the finer object details. Short connections are drawn from the deeper to the shallower branches of the network. This enables each scale's saliency map to contain the features extracted by the deeper branches, resulting in more accurate and finer saliency maps. The maps obtained from all the branches are fused to form the final saliency map that segments the most salient object in the given image.

The DSS architecture performs well for several benchmark datasets. However, the model fails to handle images with low contrast and cluttered backgrounds. It is also unable to accurately segment transparent objects, particularly those which overlap other objects in the image. In the cases mentioned above, the model either detects non-salient objects as salient or performs only a partial segmentation of the most salient object in the image. The complete object is not recognised as salient and this results in a low recall value for the model.

Indicating the presence of an object at a pixel location in an image could strengthen the probability of segmenting that pixel if it belongs to the most salient object. The SOAMs compute scale-specific maps containing objectness attention values. These values indicate the probability of an image pixel belonging to an object of a specific scale size. Integrating objectness attention information into the

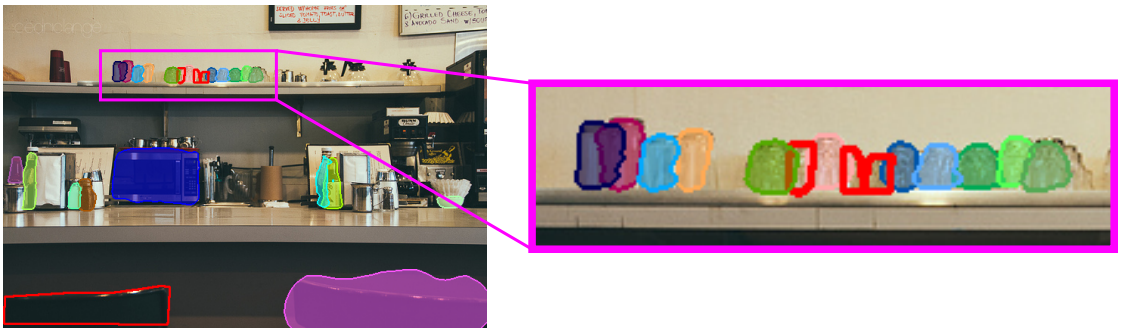


Figure 4.1: AttentionMask detects small, transparent and overlapping objects in a cluttered environment. The red contours indicate the missed object. (Image from [48].)

DSS model could be beneficial for the performance of the salient object detection task, particularly with respect to its recall value. As can be seen in Figure 4.1, AttentionMask, which relies on the SOAMs for its object discovery task, has the ability to detect very small as well as transparent objects in a cluttered arrangement. It makes the SOAMs better suitable for overcoming the drawbacks of the DSS model.

Hou et al. [13] suggest potential solutions to overcome the limitations of the DSS model which include providing prior knowledge at segment level, training on combinations of simple and complex datasets, as well as the usage of more advanced architectures. Taking into account these suggestions and the success of utilizing objectness to predict saliency as shown in [5, 17, 42, 50], a novel salient object detection architecture AttentionDSS is proposed in this thesis. The proposed model integrates the SOAMs of AttentionMask into the DSS architecture in order to produce saliency maps whose quality is enhanced by using objectness attention. AttentionDSS aims at taking care of the failure cases of the DSS architecture by generating saliency maps which segment the most salient object in its entirety and with precise contours with the help of objectness attention information.

## 4.1 The AttentionDSS architecture

The proposed AttentionDSS architecture takes an RGB image as input and produces a grey scale saliency map as an output. The saliency map highlights the most salient object in the given image with brighter pixels (values close to 1). The characteristic feature of the model is the utilization of objectness attention information to enhance the accuracy and quality of the produced saliency map. The AttentionDSS network architecture comprises of three modules: the salient object detection module (SODM), the Objectness Attention Module (OAM) and the combination module. The modules are arranged in a way that the SODM and the OAM share a common base network, VGGNet [41] and a common input image. A high level organisation of the AttentionDSS modules is shown in Figure 4.2. The

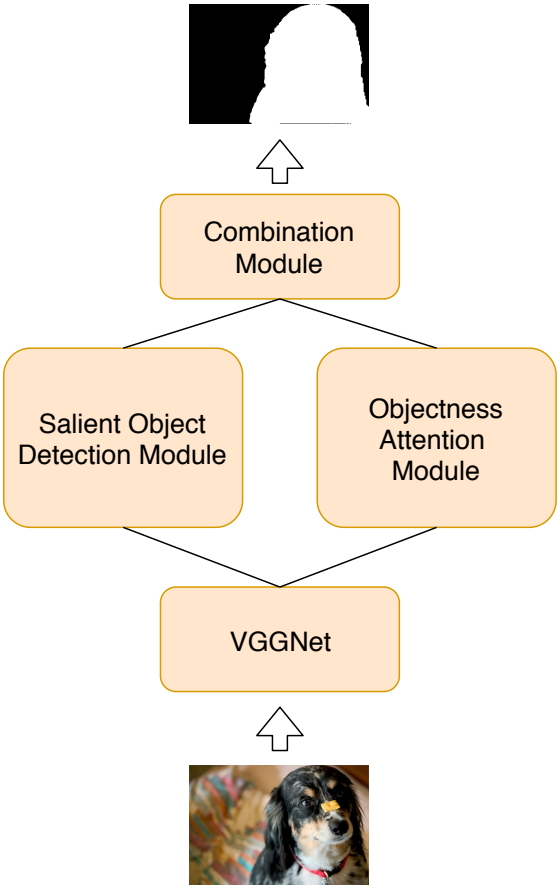



Figure 4.2: A high-level view of AttentionDSS. The VGGNet provides low level image features that are shared by all the modules. The Objectness Attention Module (OAM) comprises of Scale-specific Objectness Attention Modules (SOAMs) that generate objectness attention maps as output. The Salient Object Detection Module (SODM) as well as the Combination Module generate saliency maps as outputs.

three modules are described in detail below.

### The Salient Object Detection Module

The DSS model by Hou et al. [13] consists of a VGGNet [41] base network and side branches that generate saliency maps from image feature maps of different scales. Short connections are drawn from deeper to shallower side branches. The outputs of the side branches are fused to form the model's predicted saliency map. The side branches of the DSS model, along with the short connections between them form the Salient Object Detection Module (SODM) of AttentionDSS. The VGG base network is shared between the SODM and the Objectness Attention Module (OAM). The detailed architecture of the DSS side branches and the short connections between them have been described in section 3.6. The SODM segments the most salient object in the given image.

### The Objectness Attention Module

The Objectness Attenion Module (OAM) comprises of the Scale-specific Objectness Attention Modules (SOAMs) in AttentionMask [48]. Each SOAM generates an objectness attention map for a particular scale of the given image's feature map. The greater attention value for a pixel in the map, the better is the probability of an object's presence at that location. Each map highlights pixel locations of objects that match the particular SOAM scale. SOAMs for scales 8, 16, 32, 64 and 128 have been integrated into AttentionDSS. Scale 8 detects the smallest objects in the image, while scale 128 detects the largest object in the image. As the OAM share the VGG base network parameters with the SODM, both the modules learn their respective tasks in a multitask learning environment. The detailed description and architecture of the SOAM of AttentionMask is given in section 3.5.

### The Combination Module

This module combines the outputs from the salient object detection module and the objectness attention module. The saliency maps generated by the SODM are fused with the objectness attention maps generated by the OAM. The objectness attention values act as additional features for the salient object detection task by providing information as to whether or not an object is present at a pixel location. Thus if the boundary pixels of the most salient object have a high attention value in the objectness attention maps, it would result in high weighting for those boundary pixels during the fusion of the attention maps with the saliency map. This would result in a higher recall as more boundary pixels of the salient object that were missed by the salient object detection module would be extracted. The final saliency map after the fusion of the attention maps with the saliency maps would thus highlight the most salient object in its entirety and with more precise contours. The attention maps also highlight the presence of non-salient objects. Fusing these values with the saliency map could reduce the precision of the final saliency map as the non-salient pixels with high attention values could increase the number of false positives. The combination module consists of a concatenation layer, followed by two convolution-relu blocks and one last convolution layer. The concatenation module fuses the saliency map with the upsampled objectness attention maps. The convolution-relu block comprises of 128 kernels of size $1 \times 1$ followed by a ReLU activation function. The final convolution layer consists of 1 kernel of size $1 \times 1$. All the kernels are convolved with a stride of 1. A sigmoid function is applied to the activation map obtained from the last convolution layer to give the final saliency map of the AttentionDSS model.

The detailed architecture of AttentionDSS is shown in Figure 4.3. Some additional layers are used to connect the three modules with one another in the AttentionDSS architecture. Two max pooling layers of size $2 \times 2$ with a stride of 2, followed by an average pooling layer of size $3 \times 3$ with a stride and padding of 1 are used to connect the base network with the SOAM of scale 128.

## 4.2 Learning multiple tasks

In the proposed network architecture AttentionDSS, objectness attention prediction and salient object detection are the two main tasks that share the base network in a multitask learning set up. Goodfellow et al. [10] state that in a typical multitask learning scenario, two or more tasks are learnt on the same network with the assumption that the tasks are related and share the same input. The OAM of AttentionDSS consists of the SOAMs from AttentionMask [48]. AttentionMask is an object discovery system which relies on object features such as edges, ridges and other fine patterns in images to make predictions. Like the object discovery task, low-level image features are used to detect the most salient object. One can therefore make an assumption that the two tasks are related and can be considered for multitask learning. As stated earlier, for multitask learning, the participating tasks must have the same input. However, for the proposed method, there exists no dataset which has ground truth annotations for both the tasks i.e. objectness attention prediction as well as salient object detection. Therefore, certain multitask learning experiments are conducted to find the most suitable learning approach. They have been explained in detail in chapter 5.

Sebastian Ruder [39] states that if a network has more than one loss function to be optimised, it is effectively multitask learning. In AttentionDSS, each of the two main tasks comprise of multiple subtasks. For example, although all the SOAMs in the OAM of AttentionDSS learn the same task of objectness attention prediction, their generated maps are scale-specific. Therefore, each of the SOAMs learn their task using a separate loss function. Thus, the learning by each SOAM can be considered as an individual task. Similarly, the saliency prediction task learnt by each of the side branches in the SODM of AttentionDSS can be considered as an individual task, as each side branch processes a different feature map scale. The fusion of the attention maps and the saliency maps are also learnt using a separate loss function. Thus, multitask learning takes place at several levels in the AttentionDSS architecture. The base network in AttentionDSS is shared between the two main tasks. The outcomes of all the involved tasks can be tested at once in the given multitask learning set up and the redundant traversal of the input through the base network for each individual task can be avoided, thus making the system time efficient. Due to the shared base network, fewer parameters need to be stored, thereby reducing memory consumption as well.

In order to learn the salient object detection task, the SODM is trained to minimize the sigmoid cross entropy loss between the predicted map and the ground truth map. The ground truth map is a binary mask that has value 1 for every pixel which belongs to the most salient object in the image. Naturally, the background pixels are set to value 0. For each side branch $m$ in the SODM, the pixel-wise sigmoid cross entropy loss $l^m_{side}$ between the predicted saliency map and the ground truth map is computed. The total side loss $L_{side}$ is the weighted summation of all

the side branch losses. It is represented as

$$L_{side} = \sum_{m=1}^{M} \alpha_m l_{side}^{(m)}, \tag{4.1}$$

where $\alpha_m$ is the weight for each side branch loss $l_{side}^{(m)}$.

The fusion of the side branch maps within the SODM is also learnt, for which the fusion loss $L_{fuse}$ is computed. This loss too is a cross entropy between the fused saliency map and the ground truth map. The computation of the losses $l_{side}^{m}$ and $L_{fuse}$ have been elaborated in section 3.6. The final loss of the SODM $L_{sodm}$ is a summation of the total side loss and the fusion loss. It is represented as

$$L_{sodm} = L_{side} + L_{fuse} \tag{4.2}$$

A cross entropy loss between the predicted objectness attention map and the ground truth scale-specific objectness attention map is used to train each of the SOAMs in the OAM. The total loss of the OAM is the sum of the individual SOAM losses which can be written as

$$L_{oam} = \sum_{m=0}^{M} L_{attm}(a_m, a'_m), \tag{4.3}$$

where M is the total number of SOAMs and $L_{attm}(a_m, a'_m)$ is an individual SOAM loss between the SOAM's predicted objectness attention map $a'_m$ and the ground truth scale-specific objectness attention map $a_m$. The computation of $L_{attm}(a_m, a'_m)$ is explained in section 3.5.

The combination module learns the fusion of the objectness attention maps with the saliency map. The resulting output is saliency map that segments the most salient object in the given image. Therefore, like the SODM, the combination module is trained using a cross entropy loss between the predicted saliency map and the ground truth binary mask $Z$ for a training image $X$. The computed loss at the combination module can be written as

$$L_{comb} = -\sum_{z_j \epsilon Z} z_j \log Pr(z_j = 1|X; R) + (1 - z_j) \log Pr(z_j = 0|X; R), \tag{4.4}$$

where R represents the weights of the complete AttentionDSS network. $Pr(z_j = 1|X; R)$ is the probability of the activation at the $jth$ location of the saliency map predicted by the combination module.

If all the three modules of AttentionDSS are trained simultaneously, the resulting loss of the network is the summation of all the module losses which can be written as

$$L_{attdss} = L_{sodm} + L_{oam} + L_{comb} \tag{4.5}$$

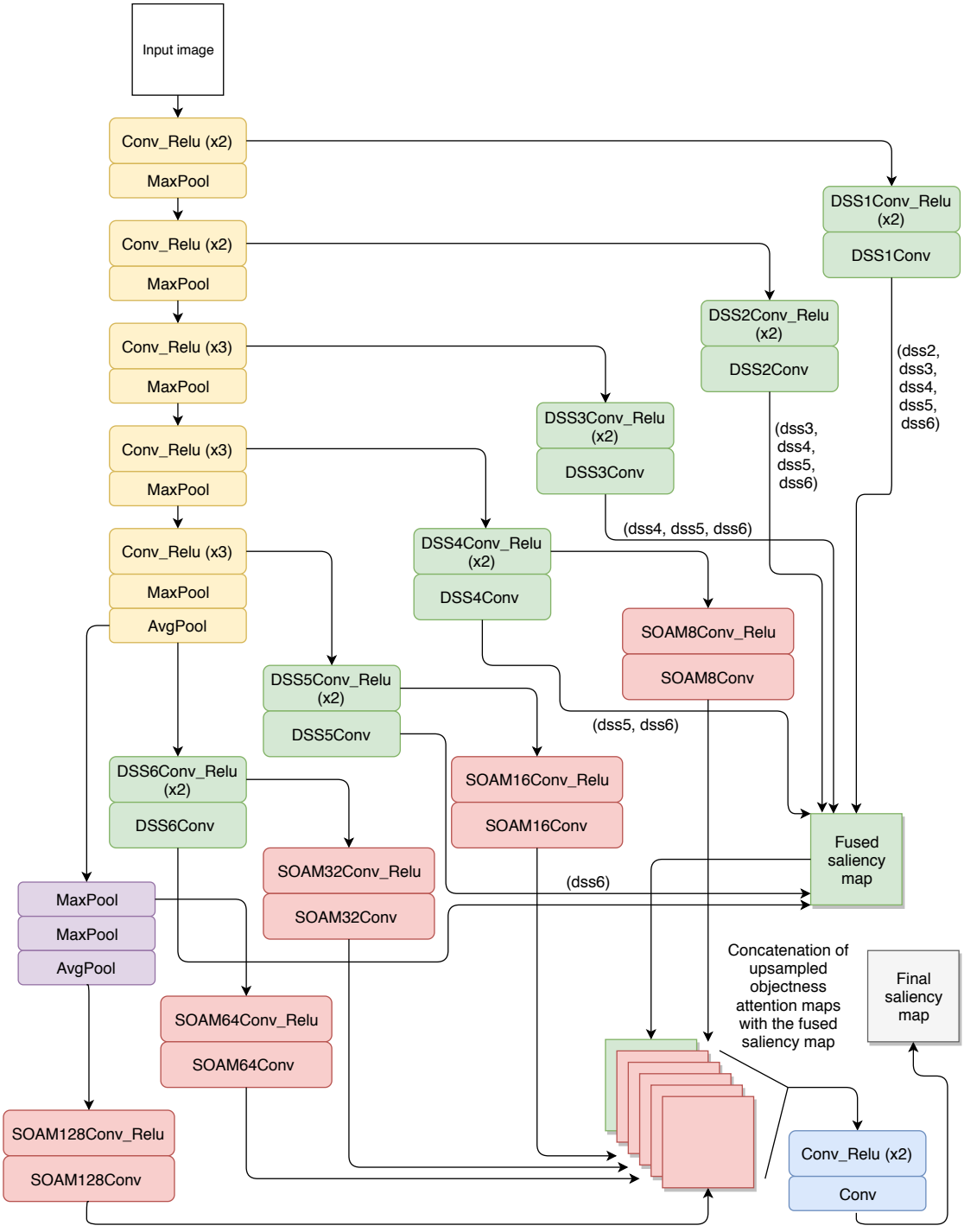



Figure 4.3: The detailed architecture of AttentionDSS. The yellow boxes form the base network (VGGNet + Average pooling layer). Each of the 6 green blocks represent one side branch of the DSS model[13], together forming the SODM. The comma-separated text along the tail of the side branches indicate the side branches from which it receives a short connection. Each of the 5 pink blocks represent a SOAM, which together form the OAM. The purple boxes are the components connecting the SOAMs of scale 64 and 128 with the base network. The blue boxes form the combination module.

# Chapter 5

# Experiments and Results

In this chapter, the experiments conducted and the results procured are reported in detail. The experimental set up including the datasets and the training approaches used are elaborated. At the end of the chapter the inferences drawn from the conducted experiments and ablation studies are discussed in detail.

## 5.1 Experimental Setup

This section describes the pre-processing of the datasets and the computational framework required for the implementation of the proposed method. The multitask learning approaches explored in this thesis are elaborated in this section.

### 5.1.1 Datasets

AttentionDSS as described in chapter 4 consists of 3 main modules: the Salient Object Detection Module (SODM), the Objectness Attention Module (OAM) consisting of SOAMs and the combination module that fuses the outputs from the SODM and OAM. The OAM is trained on the COCO [28] dataset, while the SODM is trained on the MSRA-B [16] dataset. The two datasets have been described below in detail.

#### Microsoft's Common Objects in Context (MS COCO)

The MS COCO [28] dataset as the name suggests is a large set of images capturing the complex yet contextual arrangement of objects in everyday scenes. It consists of RGB images with a maximum length 640 pixels for the longest dimension of an image. The images were annotated by means of crowd sourcing for object classification, object detection and semantic segmentation tasks. The dataset consists of 328k images covering 91 object classes and 2.5 million labelled objects. 'Things', i.e. objects which have a clear boundary and shape such as chairs or cars have been annotated in the dataset. Labelling 'stuff' i.e. non-things such as sky, grass has been proposed as future work by the authors. The object instances have entry-level category labelling such as 'dog' and not 'mammal' or 'German shepherd'.

Parts of objects such as the arms and face of a person are also labelled as object instances in the COCO dataset. The COCO images comprise of the more commonly occurring scenes and object categories with a minimum of 5000 instances per category. There could be multiple instances of the same category within an image. The number of categories per super category of objects are also balanced.

The dataset is a mixture of iconic-object, iconic-scene and non-iconic images. Iconic-object images typically contain a single large object placed centrally. Iconic-scenes are photographs that have been taken with certain intent and from specific view points, usually obtained from simple web searches. The COCO dataset contains a large number of non-iconic images which include candid shots, pictures by amateur photographers, and everyday scenes that are processed by the human eye and captured with no specific intent. The COCO dataset also contains a large number of images with small objects which are generally harder to identify without contextual information. Such a collection would help neural networks transfer the learning application to real world problems and generalise well to contextual information. The images are annotated by crowd sourcing and involved 3 main steps: specifying the object categories present in an image, locating object instances and segmenting them.

Several task-specific versions of the dataset have been released in the years 2014, 2015 and 2017. In this thesis, the 2014 split of the dataset has been used to train the network. The split consists of 82,783 training, 40,504 validation, and 40,775 testing images. Of the training data, 80,000 images have been used to train the Objectness Attention Module of AttentionDSS.

### MSRA-B

The MSRA-B dataset by Jiang et al. [16] is a subset of the larger image dataset created by Liu et al. [30]. The image dataset by Liu et al. [30] is a varied collection of 60,000+ training and 20,840 test images. These are high quality RGB images obtained from image search engines, each containing a single salient object. The images are rectangular, with the longest dimension having a maximum length of 400 pixels.The dataset was filtered to eliminate images with a single very large salient object. A bounding box around the most salient object object in an image forms the ground truth for the images in this dataset. Jiang et al. [16] select 5000 images from this dataset to form the MSRA-B dataset. They generate binary masks as ground truths for these images. The binary masks are created by segmenting out the salient object from within the bounding box, such that the segmented salient object is masked in white pixels while the background is set to black. In this thesis, 2500 images of the MSRA-B dataset are used as training data, 500 images as validation data and 2000 images as test data.

### 5.1.2 Data Pre-processing

#### MS COCO

The 80,000 training images of the MS COCO dataset are shuffled in order to eliminate any biases that could exist due to the sequential arrangement of similar images. Some of the images are randomly chosen to be flipped horizontally and some images may be extended by a small value along their longest dimension. This is done in order for the network to be accustomed to images with varied patterns and sizes while training. The images are also resized such that the height and width ratio of all images remain the same. Before the network processes each image, the pixel values along the RGB channels of the image are normalised by subtracting the dataset mean of the RGB values. This is done in order to centre the data and hence, stabilise the learning.

Pixel-precise segmentation annotations for all objects present in an image are stored in a json file. These annotations are used to create ground truths for the SOAMs in the OAM of AttentionDSS. For every training image, a ground truth binary mask is created at each scale $S_n$ corresponding to a SOAM. The ground truth masks all objects in an image that fit to a scale with white pixels, while rendering their background and other objects to black pixels. An object fits to scale $S_n$ if the bounding box lengths around the object in the image resized to scale $S_n$ are within 40% to 80% of the lengths of the fixed window size. In order to ensure that the ground truth for a scale has significant amount of relevant information to be learned by the corresponding SOAM, some filtering of ground truth maps is done. For every white pixel in the ground truth map, three black pixels are sampled, the excess black pixels are ignored in the learning process.

#### MSRA-B

The MSRA-B images are centred before the training process by subtracting the dataset mean of the RGB values from them.

### 5.1.3 Training approaches

The AttentionDSS architecture detects the most salient object in an image. The outputs from the two main components of AttentionDSS, i.e. the scale-specific attention maps from the OAM and the saliency maps from the SODM form the intermediate outputs. These outputs are fused in the third module, the combination module, to generate the final saliency map enhanced with objectness attention. In AttentionDSS, the OAM and the SODM share a common base network, the VGGNet. In order to test the feasibility of learning the two tasks associated with these modules from a common network, the following multitask learning procedures were conducted.

### Learning the three modules simultaneously

The common base network, VGGNet shared between the SODM and the OAM of AttentionDSS learns the common features of both their tasks, i.e. salient object detection and objectness attention prediction. The base network is initialised with the optimised VGG-16 weights. Due to the lack of a dataset that has annotations for both the tasks, the two modules acting as the branches of the VGGNet must be trained with their respective datasets. The OAM branch is trained on the COCO dataset while the salient object detection branch is trained on the MSRA-B dataset. The training process involves alternating between the learning of the two tasks. While learning its specific task, each branch alters the base network as well. The losses computed at the end of each of the two modules are used to update the weights of the base network. This leads to the VGGNet learning features that are common to both the objectness attention prediction task as well as the salient object detection task. It is essential that both tasks are learnt simultaneously as training the base network modules sequentially for say, task1 first and then for task2 could lead to one of the following situations:

- the base network optimises for task1, which could act as a local minimum for the complete network and hence would not further learn task2
- the network learns task1, however when further trained for task2, the base network overfits on task2

In order to alternate the learning iterations between the two tasks in a stable manner, the two datasets must be of the same size. In our case however, the two datasets are unbalanced in size. The COCO training dataset comprises of 80,000 images while the relatively smaller MSRA-B training dataset consists of 2500 images. Therefore, to stabilize the learning, every consecutive 32 training iterations for the objectness attention branch on COCO images is followed by 1 iteration for the salient object detection branch on the MSRA-B images. This results in a total of 82500 iterations for 1 training epoch, where batch size is set to 1. The combination module of AttentionDSS also learns simultaneously. This module learns the fusion of the SODM and the OAM task outputs during each MSRA-B iteration, as its target output is also a saliency map with the most salient object segmented. This module also uses the MSRA-B images as training data.

While training the three modules simultaneously, but on different datasets, i.e. the SODM and the combination module on the MSRA-B dataset and the OAM on the COCO dataset, the network weights are updated back and forth for the different tasks based on losses by the different modules. The SODM losses influence the weights of the shared base network. The OAM losses are also used to update the weights of the base network. However, the losses for training the two modules are computed on different datasets, thereby resulting in incomparable losses. It is thus equivalent to making incorrect weight updates when both the module losses have an effect on the base network. The loss computed for training the combination module on the MSRA-B dataset affects the weights of the base network, the

SODM and the OAM. This loss is used to update weights of the SOAMs, whose weights are updated by its own loss computed using the COCO dataset. Thus, the SOAM weights are also updated using unrelated losses. This makes the learning of the complete network highly unstable and the network averages on both the main tasks. As neither of the tasks are optimised, the combination module fails to learn anything meaningful. This approach is therefore not considered for further experiments.

**Learning only 2 modules simultaneously**

In this method, the SODM as well as the base network, VGGnet are not trained. They are initialized with the optimised DSS [13] model weights as the DSS model comprises of a VGGNet and its side branches that form the SODM. The OAM containing the SOAMs are trained along with the combination module. During every MSRA-B iteration, loss computed at the combination module is used to update weights of the combination module as well as the SOAMs. This results in the combination module that ultimately learns for the salient object detection task to have an influence on the SOAM weights. Hence, the SOAMs apart from sharing common low-level features from the VGGNet, are also fine-tuned for the combination task. This allows the combination module to better incorporate the objectness information as the weights of the OAM and the combination module would be better related.

**Learning one module at a time**

In this training procedure, the VGGNet along with the salient object detection module are initialized with the optimised DSS model weights and not trained further. The SOAMs are trained to generate the scale-specific objectness attention maps. Once the SOAMs are trained, their weights are frozen and the combination module alone is trained for the fusion of the saliency maps from the SODM and the attention maps from the OAM.

For all the training procedures conducted, the learning rate is set to $1e-8$, unless otherwise specified. Stochastic Gradient Descent is used as the optimization algorithm.

### 5.1.4 Hardware and software requirements

AttentionDSS is implemented using the Caffe framework and the network is trained on an NVIDIA TITAN X 12GB GPU. The GPU is utilized to its capacity while training the model with a batch size of 1 image, where the upper limit on the length of the larger dimension of each image is 500 pixels. Due to this reason, validating the tasks during the training is not possible as the validation model would require another 12GB for computation. In order to accommodate both the training as well as the validation model into the memory at the same time, the

upper limit on the length of the larger image dimension must be set to 250. Resizing images such that the larger image dimension has a maximum length of 250 pixels would compromise on the quality of the learning due to loss of information. It is therefore not recommended to validate the tasks during the training process unless more computational resources are at disposal. In order to make the learning process efficient, the fetching and pre-processing of images from both datasets are carried out simultaneously along with the training iterations using Python's multiprocessing library.

## 5.2 Evaluation Methods

The AttentionDSS network is evaluated on the 2000 test images of the MSRA-B dataset for the salient object detection task. In this thesis, the evaluation tool Salmetric [13] is used to evaluate the network performance. The tool computes the standard evaluation metrics used to test the performance of the salient object detection task which include the Precision-Recall (PR) values, the F-measure and the Mean Absolute Error (MAE).

In case of the salient object detection task, the predicted saliency map is binarized to contain 0s and 1s as pixel values. The ground truth is a binary map of 0s and 1s as well. 0 indicates the negative class while 1 indicates the positive class. If a pixel in the predicted map is correctly placed into the positive class, i.e. its predicted value as well as its true value is 1, that pixel is a true positive ($tp$). If a pixel has been incorrectly placed into the positive class, i.e. its predicted value is 1 while its true value is 0, it is a false positive ($fp$). A pixel is said to be a true negative ($tn$) if it is correctly predicted to belong to the negative class, i.e. if its predicted as well as true value is 0. A false negative ($fn$) is a pixel that has been incorrectly placed into the negative class, i.e. its predicted value is 0 while its true value is 1.

**Precision** It is the ratio of the number of correct positive predictions with respect to the total number of positive predictions. The ideal precision value is 1. Precision is computed as

$$Precision = \frac{tp}{tp + fp} \tag{5.1}$$

**Recall** It is the ratio of the number of correct positive predictions with respect to the total number of expected positive predictions. The ideal recall value is 1. Recall is computed as

$$Recall = \frac{tp}{tp + fn} \tag{5.2}$$

**F-measure** Also known as F-score, is a weighted harmonic mean of precision and recall, as precision or recall alone would not be ideal measures of accuracy for a predicted saliency map. For example, the recall for a predicted saliency map

could be 1 when the complete map is set to 1s, i.e. with a lot of false positives. A precision of 1 could be achieved even if only one pixel is placed correctly into the positive class while many positive pixels are set to 0s. The formula for computing the F-measure is

$$F_\beta = \frac{(1+\beta^2)Precision \times Recall}{\beta^2 Precision + Recall}, \tag{5.3}$$

where $\beta^2$ is set to 0.3 in order to give more importance to precision over the recall. The ideal F-score is 1, i.e. when both, the precision and recall value are 1.

**Precision-Recall Curve** In Salmetric, a precision-recall pair and an F-score is computed for a saliency map by binarising the map at every threshold value ranging from 0 to 255. These values are averaged across the complete dataset, resulting in a single set of 256 precision-recall pairs and F-scores. The highest value among the F-scores is reported as the final F-score of the AttentionDSS architecture on the complete test data. A Precision-Recall curve is plotted using the 256 precision-recall pair values, each pair representing the value for a particular threshold. The 2D plot has precision values along the vertical axis and recall values along the horizontal axis, the values ranging from 0 to 1 for both.

**Mean Absolute Error** As stated in [4], the above methods do not consider the true negatives in the computation of accuracy i.e. if negative pixels have been correctly placed in the class. The MAE is a metric that computes the absolute difference between the complete predicted saliency map $S$ and the complete ground truth map $G$, therefore considering the true negatives in the accuracy computation. For this, both maps are normalized in the range [0, 1]. The MAE is computed as

$$MAE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} |S(i,j) - G(i,j))|, \tag{5.4}$$

where $H$ and $W$ represent the height and width of the ground truth map, respectively.

## 5.3 Results

The aim of the proposed method, AttentionDSS is to test if objectness attention information contributes towards improving the performance of the salient object detection task, particularly in terms of its recall value. It aimed at overcoming the poor performance of the DSS model [13] for images with low contrast and cluttered backgrounds. The model also tries to answer if objectness attention prediction and salient object detection are related tasks.

### 5.3.1 Results of training one module at a time

In this approach, the SODM along with the base network are initialized and frozen with the optimized DSS weights. Then the SOAMs of the OAM are trained to

produce objectness attention maps. After training the SOAMs, they are initialized with their optimized weights and the layers are frozen. The combination module of the network is then trained to learn the fusion of the saliency maps from SODM and the objectness maps from the OAM. The weights of a module are only updated by the losses computed for its specific task. Such a training method allows each module to retain its characteristic behaviour of learning its specific task. For example, the performance of the SOAMs do not depreciate due to the influence of the fusion task. In this approach of learning one task at a time, each module requires approximately 3 hours for training 1 epoch. The OAM is trained on 80000 COCO images and the combination module is trained on 2500 MSRA-B images. The performance results of AttentionDSS are determined on 2000 MSRA-B test images.

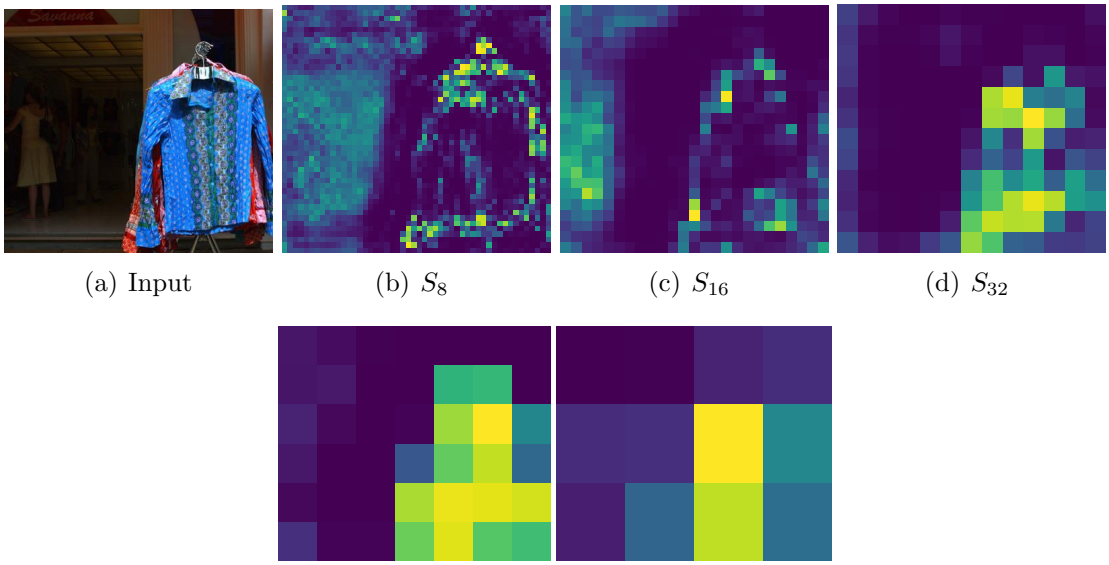
**Fusing all 5 SOAM maps with the fused saliency map obtained from the SODM**

In this experiment, the combination module learns to fuse all 5 SOAM output maps with the final fused saliency map obtained from the SODM. Objectness attention maps are obtained for the scales $S_8$, $S_{16}$, $S_{32}$, $S_{64}$ and $S_{128}$. In order to produce the objectness attention maps, the SOAMs are trained for 2 epochs. As the base network used by the OAM is already optimised, it is important to choose such as small learning rate so that the SOAMs can learn the task by taking small steps, while avoiding any gradient explosion. Scale-specific objectness attention maps obtained from the trained OAM for an example image are shown in Figure 5.1. It can be seen from the images that the SOAMs in the OAM of AttentionDSS have learnt to predict objectness attention maps for specific object sizes. For example, the SOAM for the smallest scale $S_8$ has highlighted the smaller objects in the image that matched its scale size such as the parts of the shirt collar and the hook of the clothes hanger. The SOAM for the largest image scale $S_{128}$ has highlighted the vertical patch on the left-hand side of the shirt as it is an object of a larger size. This patch may have been recognized as an object due to its striking contrast with the immediate background.

The combination module is trained for 5 epochs. The test results for the AttentionDSS models that are saved after the third and fifth training epoch are reported. It would be ccomputationally efficient to stop the training at epoch 3, as the difference in the F-measures for the AttentionDSS model at epoch 3 and epoch 5 is not significant. However, an example image analysis that follows later in this subsection suggests the advantage of choosing epoch 5 to stop the training.

**Training the combination module for 3 epochs**

Table 5.1 shows the results of AttentionDSS and DSS after training the combination module for 3 epochs. As the base network and the SODM in AttentionDSS together form the DSS model, the performance results on the fused saliency map obtained from the SODM can be termed as the DSS model result. The numbers in the table indicate that the F-measure for both the AttentionDSS model and the



(a) Input (b) $S_8$ (c) $S_{16}$ (d) $S_{32}$

(e) $S_{64}$ (f) $S_{128}$

Figure 5.1: The objectness attention maps (b), (c), (d), (e), (f) generated by the SOAMs in AttentionDSS for the input image (a). Each map highlights objects in the image that match the specific scale size.

| Results on 2000 MSRA-B test images | | | | |
|---|---|---|---|---|
| | Precision | Recall | F-measure | MAE |
| DSS | 0.868 | 0.762 | 0.841 | 0.066 |
| DSS with CRF | 0.885 | 0.783 | 0.859 | 0.060 |
| AttentionDSS (epoch 3) | 0.852 | 0.800 | 0.839 | 0.071 |
| AttentionDSS with CRF (epoch 3) | 0.862 | 0.825 | 0.853 | 0.061 |
| AttentionDSS (epoch 5) | 0.871 | 0.755 | 0.841 | 0.066 |
| AttentionDSS with CRF (epoch5) | 0.874 | 0.801 | 0.856 | 0.06 |

Table 5.1: Performance results of AttentionDSS when the fused saliency map from the SODM is combined with all the scale-specific objectness attention maps obtained from the OAM.

DSS model are almost equal. As expected the recall for AttentionDSS is higher than the DSS model with the influence of objectness attention. It has however compromised on the model's precision resulting in an unchanged F-measure in comparison to the DSS baseline. The decline in the precision could indicate the predictions of false positives that may have been introduced by the OAM.

After applying a fully connected Conditional Random Field (CRF) post-processing method from Hou et al. [13], the F-measures of both the models improve. These values can be observed in Table 5.1. The precision of the DSS model improved by 1.7% while precision of the AttentionDSS model improved by 1%. The recall for

the DSS model and the AttentionDSS model increased by 2.1% and 2.5%, respectively. Another parallel observation is that the application of the CRF reduced the MAE of the models. Thus it can be drawn that the applied CRF method is an effective post-processing tool.

### Training the combination module for 5 epochs

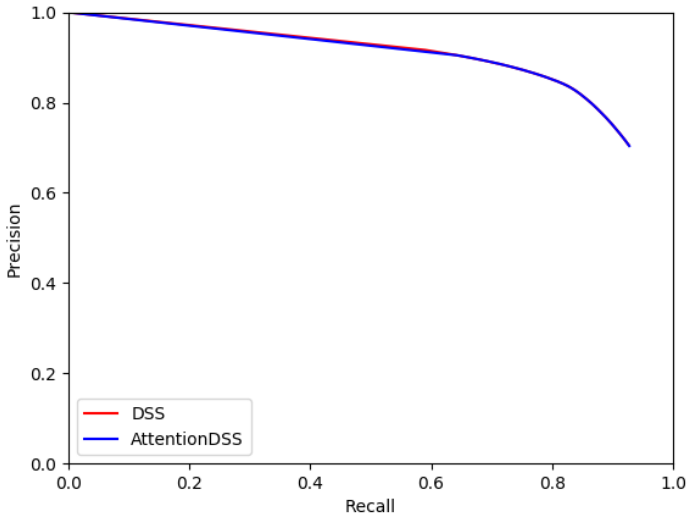
Keeping all the hyperparameters and the experimental approach the same as above, the performance of the AttentionDSS model after the combination module has been trained for 5 epochs is tested. The results are shown in Table 5.1. Although the F-measure does not differ much from the F-measure on testing the model after 3 epochs, it is equal to the F-measure of the DSS baseline model. Another behaviour observed is, that at epoch 5, the precision of AttentionDSS increases and its recall decreases in comparison with the model at epoch 3. This could indicate that the number of false positives decrease in epoch 5. On comparing subfigures (g) and (e) in Figure 5.3, one can observe that the number of false positives in subfigure (g) appear fewer as compared to subfigure (e). This further supports the possibility of the decrease in recall to be caused by the reduction in the number of false positives. It can be seen in Table 5.1, that as observed for epoch3, the performance metrics of AttentionDSS improve on applying the CRF post-processing.

Based on the results, obtained so far, AttentionDSS has the best results for epoch 5, after the CRF post-processing is applied. The Precision-Recall curves for AttentionDSS at epoch 5 is plotted in Figure 5.2.

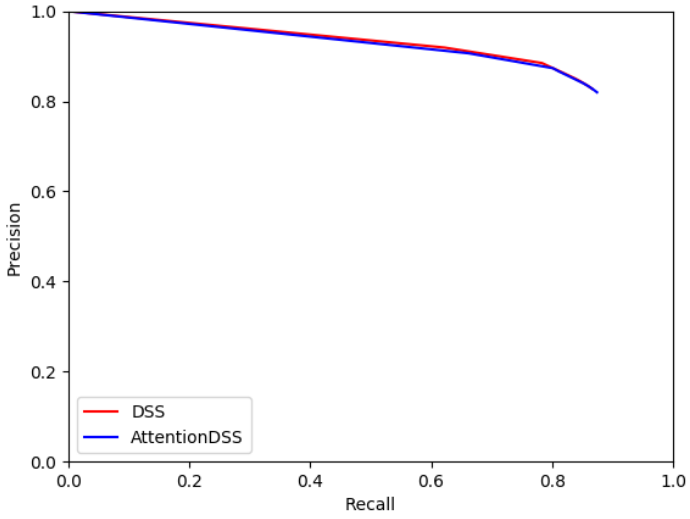
### Qualitative analysis

A qualitative analysis of the saliency maps obtained from the above experiments can be performed by taking a look at Figure 5.3. On comparing subfigure (c) and (e), one can notice that the AttentionDSS saliency map highlights the most salient object with a larger number of white pixels i.e. the recall of the map appears to be visually higher than the DSS saliency map for this image example. The handle bars of the bicycle are more clearly visible in the AttentionDSS saliency map, giving the object a better structure. This make it easier to recognise the object as a bicycle. Subfigure (d) and (f) show that applying a CRF post-processing gives sharper contours to the detected object, thereby improving the contrast between the salient object and the background. Subfigure (g) shows the AttentionDSS saliency map obtained after training the combination module for 5 epochs. The map appears to have fewer false positives when compared to the map obtained after 3 epochs of training, i.e. subfigure (e). Apply CRF post-processing to such a map results in a map as shown in subfigure (h), which has sharper contours and finer details within the object structure.

Figure 5.4 shows example images where the AttentionDSS saliency maps visually appear to have a better recall and fewer or no false positives when compared with their corresponding DSS saliency maps. These examples show that AttentionDSS utilizes the objectness attention information to strengthen the positive



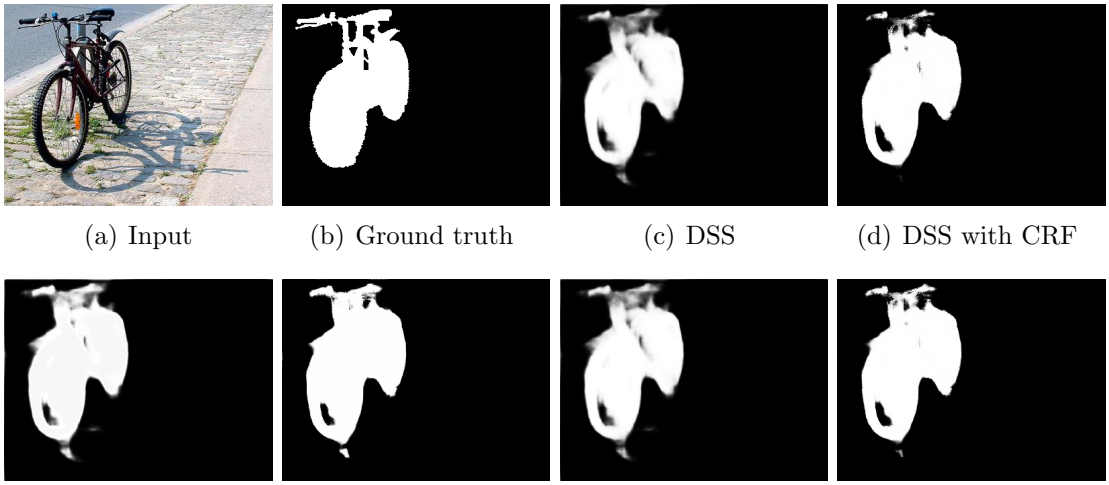

No post-processing




With CRF post-processing

Figure 5.2: Precision-Recall curve when the fused saliency map from the SODM is combined with all the scale-specific objectness attention maps obtained from the OAM. The plotted curves shows the test result of AttentionDSS when the combination module is trained for 5 epochs.

pixels determined by the DSS model. The pixels immediately surrounding the positive pixels in DSS map are highlighted to sharpen the salient object's contour and fill gaps within the salient object predicted in the DSS map. In the second and third row, although the DSS map extracts the finer details of the salient object, several true positives are not retrieved. Their corresponding AttentionDSS map although not entirely, better masks the salient object, indicating that a higher number of



(a) Input (b) Ground truth (c) DSS (d) DSS with CRF

(e) AttentionDSS (epoch3) (f) AttentionDSS with CRF (epoch3) (g) AttentionDSS (epoch5) (h) AttentionDSS with CRF (epoch5)

Figure 5.3: Comparing saliency maps obtained from the DSS model and the AttentionDSS model, with and without CRF post-processing.

true positives are predicted.

In Figure 5.5 the failure cases of AttentionDSS can be seen. The first row of the figure shows the worst case scenario for the AttentionDSS performance. One can observe that the salient object predicted by the DSS map does not match with the salient object annotated in the ground truth map. Although the DSS model has highlighted very small portions of the ground truth salient object, a larger portion of the non-salient object is segmented. This contributes to an increase in the number of false positives for the image. The predicted object too is not segmented with good precision. In such cases, where the underlying saliency prediction model predicts a non-salient object to be salient, the AttentionDSS model fails to improve the results. As can be seen in the figure, the AttentionDSS map for the image in the first row has improved the segmentation of the salient object falsely predicted by the DSS map. This has led to an increase in the recall of the predicted object, however, as the predicted object does not correspond with the ground truth salient object, the recall contributes to an increase in the number of false positives for the complete image. It therefore worsens the quality of the prediction.

In the second row of Figure 5.5, although the DSS map correctly highlights the most salient, non-salient parts of the image are also segmented. These non-salient parts are further strengthened by AttentionDSS, leading to an increase in the number of false positives, thereby worsening the prediction performance. On taking a very close look at the DSS map for this image, one can notice faint border like artifacts along the left-handside vertical axis and the bottom-left corner of the map. These artifacts too are enhanced in the AttentionDSS map. In the third row of the Figure, in comparison with the DSS map, the AttentionDSS map appears to look closer to the ground truth map, particularly with respect to the segmentation of the green game token around its neck region. However, one can notice that the

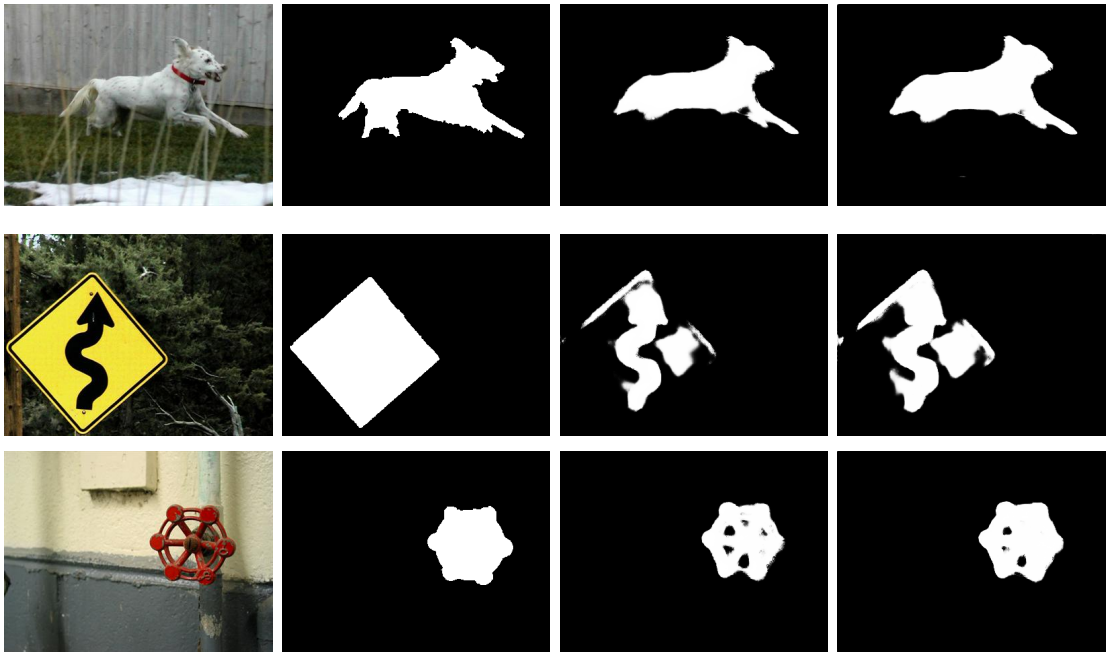



Figure 5.4: Cases where AttentionDSS produces visually better saliency maps than the DSS model. The AttentionDSS saliency maps appear to have a better recall, while keeping the false positives low.

predicted DSS map connects the game token along its base, while the ground truth annotates them as separate pieces. The pixels connecting the game pieces are false positives and the AttentionDSS model, as can been seen in the AttentionDSS map, increases the number of these connecting pixels.

### Fusing SOAM maps of scale 32, 64 and 128 with the fused saliency map obtained from the SODM

Keeping the hyperparameters the same as in the above experiments, the combination module of AttentionDSS is trained to fuse the saliency map obtained from the SODM with the objectness attention maps of the scales $S_{32}$,$S_{64}$ and $S_{128}$. The test results of AttentionDSS with the combination module trained for 5 epochs in this manner is shown in Table 5.2. In this set up, the F-measure of AttentionDSS does not show a significant difference from the F-measure when the combination module considered objectness maps of all the 5 SOAMs for the fusion. After CRF post-processing, the F-measure, as shown in Table 5.2 is identical to when all 5 SOAM maps were used in the combination module and CRF was applied. It would therefore be efficient to take advantage of objectness information from all SOAM maps and train the combination module for 5 epochs.

The performance results with and without CRF are shown in Table 5.2.

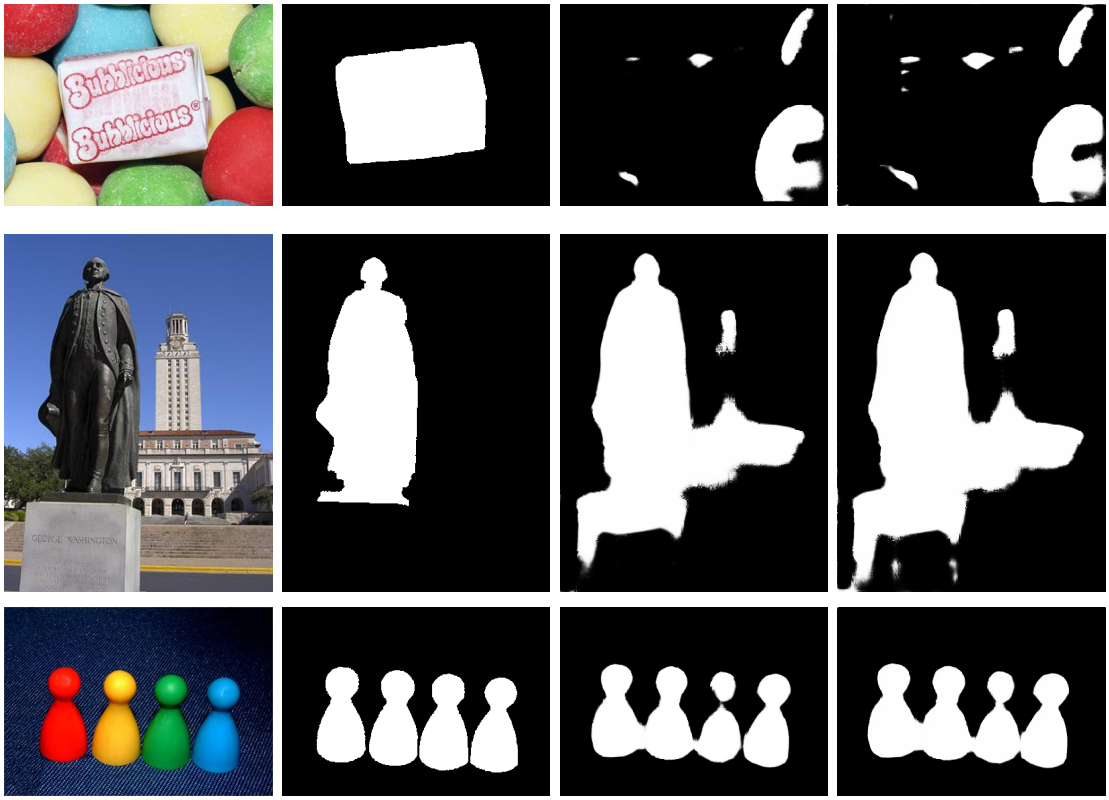



Figure 5.5: Image cases where AttentionDSS maps appear to be of poorer quality in comparison with their DSS counterparts. In these images, the AttentionDSS model has the false positives predicted by the DSS model.

| Results on 2000 MSRA-B test images | | | | |
|---|---|---|---|---|
| | Precision | Recall | F-measure | MAE |
| DSS | 0.868 | 0.762 | 0.841 | 0.066 |
| DSS with CRF | 0.885 | 0.783 | 0.859 | 0.060 |
| AttentionDSS | 0.860 | 0.783 | 0.841 | 0.070 |
| AttentionDSS with CRF | 0.872 | 0.806 | 0.856 | 0.061 |

Table 5.2: Performance results of the AttentionDSS when the fused saliency map from the SODM is combined with the objectness attention maps for scales 32, 64 and 128. The results are depicted for the AttentionDSS model when the combination module is trained for 5 epochs.

### Ablation Experiments

Certain ablation experiments were conducted where the combination module was trained to fuse activation maps obtained from the SODM and the OAM. These maps are the un-sigmoided feature maps, i.e. maps from a convolution layer before the output layer of the SODM and OAM. On fusing all 5 activation maps obtained

Input Ground truth AttentionDSS map

Figure 5.6: On fusing the activation maps obtained from the convolution layer before the output layers of SODM and OAM, a saliency map that identifies the salient object correctly is obtained. However, the map is inverted and boundaries between foreground and background are blur.

from the OAM with all the activation maps (6 side branch maps and fused map) from SODM, the combination module outputs a map which detects the salient object. The object structure appears to be detected correctly, but the boundaries between the foreground and background is blur. Combining all 5 activation maps from the OAM with only the activation map before the fused saliency map from SODM also gives similar results. During this experiment, the last convolution layer before each SOAM output layer as well as the convolution layer before the fused saliency output map in SODM were unfrozen and their weights were updated based on the losses computed during the training of the combination module. A similar pattern is obtained when the last convolution layer of the SODM and the OAM are unfrozen and learnt with the help of the losses computed during training of the combination module. However, the obtained map is inverted, i.e. the salient object is masked in black pixels while the background is in white. Sample saliency map obtained after the fusion of activation maps in the combination module can be seen in Figure 5.6. On training these models for 3 epochs, the maps showed no significant improvement in its appearance.

As part of ablation study few convolution layers were added as a bridge connecting the SOAMs of scale $S_{32}, S_{64}$ and $S_{128}$ with the base network VGGNet. While the basenetwork and the SODM were frozen with the optimised DSS weights, the SOAMs and the added convolution layers were trained. A drawback of adding the convolution layers before the SOAMs is that they were unable to learn the their task, i.e. the SOAMs of scales $S_{32}, S_{64}$ and $S_{128}$ failed to produce objectness attention maps. This resulted in the combination module fusing the objectness maps of scales $S_8$ and $S_{16}$ with the 7 saliency maps (6 side branch saliency maps and 1 fused saliency map) obtained from the SODM. The combination module was trained for 3 epochs. The F-measure shown in Table 5.3 indicates that the AttentionDSS does not perform better than the DSS model. The MAE for the AttentionDSS model is also high. However on applying CRF post processing, the MAE of AttentionDSS reduces significantly and the F-measure is higher than that of the DSS model by almost 2 percent. This can be seen in Table 5.3.

| Results on 2000 MSRA-B test images | | | | |
|---|---|---|---|---|
| | Precision | Recall | F-measure | MAE |
| DSS | 0.869 | 0.756 | 0.840 | 0.070 |
| DSS with CRF | 0.884 | 0.792 | 0.861 | 0.060 |
| AttentionDSS | 0.863 | 0.770 | 0.840 | 0.084 |
| AttentionDSS with CRF | 0.911 | 0.797 | 0.882 | 0.059 |

Table 5.3: Performance results of the AttentionDSS when all 7 saliency maps (6 side branch saliency maps and 1 fused saliency map) obtained from the SODM are combined with the objectness attention maps of scales 8 and 16. The results are depicted for the AttentionDSS model when the combination module is trained for 3 epochs.

### 5.3.2 Results of training two modules simultaneously

In this approach, the OAM as well as the combination module are trained simultaneously i.e. the losses computed by the combination module are used to update the weights of the combination module as well as the OAM. It causes the weights of the SOAMs to be better related to those of the combination module. This results in the SOAMs to incorporate certain characteristics of the fusion task, thereby making the fusion of the saliency maps and the objectness attention maps smoother. The results reported here are for the model where the convolution layers are added before the SOAMs of scales 32, 64 and 128. The combination module fuses the objectness attention maps for scales 8 and 16 with all the 7 saliency maps from the SODM. The combination module is trained for 3 epochs with a learning rate of 1e-4. This learning rate resulted in gradient explosion, hence the gradients were clipped to a value of 1 during the training. One can observe from Table 5.4 that the DSS module has a higher F-measure score than AttentionDSS. A CRF post-processing does not improve the F-measure for the AttentionDSS model. The results after post-processing can be seen in Table 5.4. A reason for the AttentionDSS model to not have a performance equal to or better than the DSS model could be that as the combination module losses have an impact on the weights of the SOAMs, the SOAMs do not optimize for their specific task. Therefore, the SOAMs in this case have either no effect or a negative effect on the performance of the AttentionDSS model.

## 5.4 Discussion

From the experiments conducted and the results reported in this chapter, several inferences can be made. They are as follows.

In the AttentionDSS architecture, the base network VGGNet is shared with the Salient Object Detection Module (SODM) and the Objectness Attention Module (OAM). The base network and the SODM together form the original salient object

| Results on 2000 MSRA-B test images | | | | |
|---|---|---|---|---|
| | Precision | Recall | F-measure | MAE |
| DSS | 0.868 | 0.758 | 0.840 | 0.070 |
| DSS with CRF | 0.883 | 0.795 | 0.861 | 0.060 |
| AttentionDSS | 0.786 | 0.877 | 0.806 | 0.080 |
| AttentionDSS with CRF | 0.785 | 0.895 | 0.808 | 0.072 |

Table 5.4: Performance results of the AttentionDSS when all 7 saliency maps (6 side branch saliency maps and 1 fused saliency map) obtained from the SODM are combined with the objectness attention maps of scales 8 and 16. The results are depicted for the AttentionDSS model when the combination module and the OAM are trained simultaneously for 3 epochs.

detection model, DSS by Hou et al. [13]. The OAM comprises of the Scale-specific Objectness Attention Modules (SOAMs) of scales $S_8$, $S_{16}$, $S_{32}$, $S_{64}$ and $S_{128}$. In the experiments where one module of AttentionDSS is trained at a time, the OAM is trained by freezing the base network and the SODM with the optimised DSS model weights. As can be seen in Figure 5.1, the OAM is able to produce scale-specific objectness attention maps, in which the objects that match the specific SOAM scale are highlighted. Here, the objectness attention prediction task of the OAM is trained on a base network frozen with weights optimised for the salient object detection task. It can therefore be deduced that salient object detection and objectness attention prediction are related tasks.

The performance results of AttentionDSS on 2000 MSRA-B test images as well as the qualitative analysis of the predicted saliency maps have been reported in this chapter. On training the combination module the overall F-measure for the AttentionDSS model does not get better than the F-measure of the DSS model. For the image cases shown in Figure 5.4, the saliency maps predicted by AttentionDSS appear visually more accurate when compared with their corresponding DSS maps. AttentionDSS utilizes scale-specific objectness attention to complete the structure of the detected salient object. This would contribute to an increase in the number of true positives, thereby improving the recall value. Visually, the predicted maps appear to have few to no false positives as background or non-salient pixels are not highlighted. One could infer that combination module learns to fuse the objectness attention maps with the saliency map in a way that pixels around the positive predictions in the saliency map from the SODM are retrieved up until the nearest object contour predicted in the objectness maps. No additional false positives are introduced.

However, this behavior is only beneficial if the predictions in the saliency map by the SODM are true positives. If the positive predictions in the saliency map from the SODM are false positives, the fusion of this map with the objectness maps would lead to an increase in the number of false positives. This is caused as the combination module would strengthen the positive predictions in the saliency

map further by retrieving neighbouring pixels, irrespective of whether the positive predictions are true or false. Examples of such maps predicted by AttentionDSS were shown in Figure 5.5. The increase in the number of false positives would contribute to a lower F-score. This explains why the F-measure of the AttentionDSS does not improve over the baseline model, DSS. Another factor for the F-scores not reflecting the improved recall of the AttentionDSS model could be the value of $\beta^2$ in the formula for computing the F-measure as shown in Equation 5.3 in subsection 5.2. This value is set to 0.3 for the evaluation of the DSS baseline in order to weight the precision higher than the recall in the computation of the F-score. This value has been retained for the evaluation of AttentionDSS for better comparison with the DSS baseline.

On applying a Conditional Random Field (CRF) post processing method, the Mean Absolute Error (MAE) for the AttentionDSS model appears to decrease significantly. As the MAE is the average of the absolute difference between the predicted saliency map and the ground truth map, this value represents both, the number of false negatives and the number of false positives. A decrease in the MAE for the AttentionDSS after the application of the CRF would indicate that the CRF helps eliminate the false positives and increase the number of true positives (or reduce the number of false negatives). This is reflected in an increase in the precision and recall values, and thereby the F-score.

Combining objectness information in the manner proposed in this thesis would be beneficial for baseline salient object detection models that have very low false positive scores. It would also be advantageous to incorporate the objectness attention information at an earlier stage in a saliency model. This allow the saliency model to utilize the objectness attention information to prevent the prediction of false positives in the saliency map, as the objectness information would lay emphasis one only those locations of the image that belong to an object. It would be particularly useful for predicting saliency maps for images that mostly consist of one single object in the foreground. A weighted fusion of the saliency map and the objectness maps in the combination module of AttentionDSS, where the objectness maps are weighted higher could help incorporate more objectness information in the saliency prediction task, allowing the recall value to improve significantly. A different architecture of the combination module could be explored for better performance scores. The element-wise product or addition could also be tested for the fusion of the saliency map and the objectness attention maps.

Fusing the saliency map from the SODM with only the larger scale objectness attention maps, i.e. scale 32, 64 and 128, results in a similar F-score value as when all objectness attention maps of all scales are fused. However, the combination module must be trained for more number of epochs to get a similar result when only the larger scale objectness attention maps are used. The MAE is also higher when only the larger scale objectness attention masks are used. It is therefore beneficial to use the objectness attention maps of all scales as it is computationally efficient and has a lower MAE value.

As part of ablation studies, instead of the output layer maps, feature maps from the last convolution layers of the OAM and SODM were fused in the combination

module. Deeper layers of the SODM and the OAM were also unfrozen during the training of the combination module. The resulting saliency maps appear to identify the location of the salient object, they however fail to bring out a sharp contrast between the detected object and the background. This approach of fusing activation maps instead of output layer maps looks promising. A better choice of hyperparameters and a different architecture for the combination module could be inspected for this approach.

Using convolution layers as the connecting component between the SOAMs and the base network caused the SOAMs fail to learn their task. It indicates that the SOAMs can directly learn from the low-level features extracted by the base network which is shared with the SODM. This further supports the inference drawn earlier that objectness attention prediction and salient object detection are related tasks. In the experiment conducted, convolution layers were placed between the VGGNet and the SOAMs of scales 32, 64 and 128. It resulted in only the SOAMs of scales 8 and 16 to learn their task. On fusing the saliency map from the SODM with the objectness attention maps obtained from the SOAMs of scales 8 and 16, the F-measure for AttentionDSS model is equal to that of the DSS model. The MAE however, is much higher for the AttentionDSS model. However, on applying CRF post-processing, the MAE drops significantly and the precision increases sharply. It results in the F-measure of AttentionDSS to be higher than that of the DSS model by approximately 2 percent. It would be interesting to further research in this direction to gain insights about the usefulness of smaller scale objectness attention maps for salient object detection.

Using the same model as above, the OAM and the combination module were trained simultaneously. There was a drop in the F-measure by approximately 3.5% compared to when the OAM and the combination module are trained in isolation. Although the tasks of the two modules are related, such a behaviour can be expected as the two modules are trained on two different datasets. During the simultaneously training of the OAM and the combination module, the losses computed by the combination module based on its own data sample are backpropagated to update the weights of the OAM. As the weights of the two modules are unrelated due to the difference in the data that they are trained on, the weight updates become less stable. This causes the performance of the model to deteriorate.

For the very same reason, the model fails to learn anything meaningful when all the three modules as well as the base network are trained simultaneously. A dataset that is annotated with the ground truth for both, the salient object detection task and objectness attention prediction task would be favourable. End-to-end training of the model on a single dataset would make the computation better efficient and the learning more stable. More concrete insights can be drawn on the impact of objectness attention in improving the performance of the salient object detection task.

# Chapter 6

# Conclusion

This chapter summarizes the contributions of the thesis and the questions answered by the proposed method based on the experiments conducted. The possible directions in which research can be continued are also briefly discussed.

## 6.1 Summary of the Thesis

This thesis addresses the topic of salient object detection and how objectness attention information can contribute towards improving the quality of the predictions. The major contribution of this thesis is the proposed model AttentionDSS, that incorporates objectness attention prediction components into a salient objection detection architecture. AttentionDSS uses a state of the art architecture, DSS by Hou et al. [13] as its underlying Salient Object Detection Module (SODM). The Scale-specific Objectness Attention Modules (SOAMs) from the state-of-the-art object discovery model, AttentionMask form the Objectness Attention Module of AttentionDSS. The base network of the DSS model is shared with the OAM. The saliency map generated by SODM is fused with the scale-specific objectness attention maps generated by the SOAMs to produce a saliency map that is enhanced with objectness attention.

The DSS model has some drawbacks. It has a low recall value for low contrast images in that it does not segment the detected salient object completely. It also incorrectly predicts non-salient objects as salient, particularly in images with a cluttered background. AttentionDSS aimed to overcome these limitations by utilizing objectness attention information. Objectness attention indicates an image pixel's probability of belonging to an object in the image. The objectness attention maps generated by the OAM in AttentionDSS are scale-specific, i.e. each map highlights objects that match the specific scale size. The hypothesis of the thesis was that the indication of an object's presence at a pixel location in an image would increase the likelihood of segmenting that pixel if it belongs to the salient object. Therefore, experiments were conducted to validate the correctness of the hypothesis. Simply put, the thesis answers the question 'does objectness attention information improve the performance of the salient object detection task ?'.

The results of the experiments show that objectness attention helps improve the recall in image cases where the saliency map produced by the underlying SODM has fewer false positives. The objectness attention information contributes towards segmenting the detected object completely, thereby increasing the recall. However in cases where the saliency map predicted by the SODM has many false positives, the objectness attention considers the false positives as the true positives and retrieves pixels around them to expand and segment the object that the false positive pixels belong to. This leads to a further increase in the number of false positives. It is fair to say that AttentionDSS has the potential to solve the DSS model's limitation of a low recall value, particularly for maps with fewer false positives. However the incorrect predictions of non-salient objects as salient remains unsolved. Integrating objectness attention at an early stage in the architecture, for example, within the SODM itself could allow the objectness attention information to have a greater influence in the prediction of the saliency maps, particularly in preventing the occurrences of false positives.

The importance of particular SOAM scales is also studied in this thesis. From the qualitative and quantitative analysis of the experimental results, SOAMs of all the scales appear to have an importance in the predictions made by AttentionDSS. A Conditional Random Field post-processing method caused the AttentionDSS model to acquire an improvement in the F-measure over the baseline DSS model, when the objectness attention maps of the smaller scales, 8 and 16 were used to make the predictions.

The learning is most stable and produces the best results when each of the modules in the AttentionDSS architecture are trained in isolation, i.e all modules except for the one being currently trained are frozen. Training all the modules simultaneously results in an unstable learning process as both the tasks, salient obj detection and objectness attention prediction are trained using different datasets. A dataset with annotations for both the tasks would have facilitated the simultaneous training of all the AttentionDSS modules.

The second question that AttentionDSS answers is 'are salient object detection and objectness attention prediction related tasks?'. As the OAM in the AttentionDSS architecture shares the base network of the salient object detection model, DSS and learns the objectness attention prediction task from the low level features extracted by this base network, it can be inferred that objectness attention prediction and salient object detection are related tasks.

## 6.2 Future Work

Based on the inferences drawn from the results reported in this thesis, it can be stated that objectness attention shows potential in improving the segmentation of the salient object detected in an image. It would therefore be noteworthy to consider further research in this direction.

In the proposed model, AttentionDSS, objectness attention information is incorporated into the model only after the underlying salient object detection module

has produced a saliency map. Therefore, objectness attention has limited scope of altering predictions that have been made. For example, objectness attention is unable to mend or eliminate the false positives predicted in the saliency map obtained from the salient object detection module in AttentionDSS. It would be beneficial if objectness attention can contribute towards the extraction of the low-level features that are fundamental to the salient object detection task. Therefore, incorporating objectness information at an earlier stage in a salient object detection model would be promising. It would be interesting to develop an architecture that would allow the objectness information to assist the salient object detection task, rather than act as a post-processing method.

Different methods for the fusion of the saliency map and the objectness maps in AttentionDSS would also be worth exploring. For example, the network architecture of the module that learns the fusion could be modified or a weighted fusion of the maps could be done, where more weight is given to the objectness attention maps. As the SOAM scales 8 and 16 proved beneficial for improving the F-measure of AttentionDSS, although only after post-processing, more experiments could be made to gain insights into the behavior of the objectness attention maps of these scales during their fusion with the saliency map.

Last, but not the least, the process of testing the effect on objectness attention in the salient object detection task could be made easier with the availability of a dataset that has ground truth annotations for both the tasks. It would then be possible to train all the modules of AttentionDSS simultaneously and in a stable manner. Training all the sub-tasks within the network simultaneously would also make the learning process computationally efficient.

# Bibliography


[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Süsstrunk. Frequency-tuned salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[3] Ali Borji. What is a salient object? a dataset and a baseline model for salient object detection. *IEEE Transactions on Image Processing*, 24(2):742–756, 2014.

[4] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey, 2014.

[5] Kai-Yueh Chang, Tyng-Luh Liu, Hwann-Tzong Chen, and Shang-Hong Lai. Fusing generic objectness and visual saliency for salient object detection. In *IEEE International Conference on Computer Vision*, 2011.

[6] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2014.

[7] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

[8] Simone Frintrop. *VOCUS: A visual attention system for object detection and goal-directed search*. Springer, 2006.

[9] Simone Frintrop, Thomas Werner, and German Martin Garcia. Traditional saliency reloaded: A good old model in new shape. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[11] 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.

[12] Esther Horbert, Germán M García, Simone Frintrop, and Bastian Leibe. Sequence-level object candidates based on saliency for generic object recognition on mobile systems. In *IEEE International Conference on Robotics and Automation*, 2015.

[13] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip HS Torr. Deeply supervised salient object detection with short connections. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[14] Hexiang Hu, Shiyi Lan, Yuning Jiang, Zhimin Cao, and Fei Sha. Fastmask: Segment multi-scale object candidates in one shot. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[15] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.

[16] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[17] Peng Jiang, Haibin Ling, Jingyi Yu, and Jingliang Peng. Salient region detection by ufo: Uniqueness, focusness and objectness. In *IEEE International Conference on Computer Vision*, 2013.

[18] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018.

[19] Dominik A Klein and Simone Frintrop. Center-surround divergence of feature statistics for salient object detection. In *IEEE International Conference on Computer Vision*, 2011.

[20] Dominik Alexander Klein and Simone Frintrop. Salient pattern detection using W2 on multivariate normal distributions. In *Pattern Recognition*. Springer, 2012.

[21] Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*. Springer, 1987.

[22] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[23] Gayoung Lee, Yu-Wing Tai, and Junmo Kim. Deep saliency with encoded low level distance map and high level features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[24] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[25] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[26] Xi Li, Liming Zhao, Lina Wei, Ming-Hsuan Yang, Fei Wu, Yueting Zhuang, Haibin Ling, and Jingdong Wang. Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE Transactions on Image Processing*, 2016.

[27] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–287, 2014.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014.

[29] Nian Liu and Junwei Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[30] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[31] Mingsheng Long and Jianmin Wang. Learning multiple tasks with deep relationship networks. *arXiv preprint arXiv:1506.02117*, 2, 2015.

[32] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5334–5343, 2017.

[33] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3994–4003, June 2016.

[34] Vida Movahedi and James H Elder. Design and perceptual validation of performance measures for salient object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[35] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, 2015.

[36] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *European Conference on Computer Vision*. Springer, 2016.

[37] Jordi Pont-Tuset, Pablo Arbelaez, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[38] Siyuan Qiao, Wei Shen, Weichao Qiu, Chenxi Liu, and Alan Yuille. Scalenet: Guiding object proposal generation in supermarkets and beyond. In *IEEE International Conference on Computer Vision*, 2017.

[39] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[40] Sebastian Ruder12, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Sluice networks: Learning what to share between loosely related tasks. *stat*, 1050:23, 2017.

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

[42] R Sai Srivatsa and R Venkatesh Babu. Salient object detection via objectness measure. In *IEEE International Conference on Image Processing*, 2015.

[43] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 1980.

[44] John K Tsotsos, Scan M Culhane, Winky Yan Kei Wai, Yuzhong Lai, Neal Davis, and Fernando Nuflo. Modeling visual attention via selective tuning. *Artificial intelligence*, 1995.

[45] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013.

[46] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *European Conference on Computer Vision*. Springer, 2016.

[47] Thomas Werner, Germán Martín-García, and Simone Frintrop. Saliency-guided object candidates based on gestalt principles. In *International Conference on Computer Vision Systems*. Springer, 2015.

[48] Christian Wilms and Simone Frintrop. AttentionMask: Attentive, efficient object proposal generation focusing on small objects. In *Asien Conference on Computer Vision*, 2018.

[49] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *IEEE International Conference on Computer Vision*, 2015.

[50] Linwei Ye, Zhi Liu, Lina Li, Liquan Shen, Cong Bai, and Yang Wang. Salient object segmentation via effective integration of saliency and objectness. *IEEE Transactions on Multimedia*, 2017.

[51] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[52] Lei Zhu, Dominik A Klein, Simone Frintrop, Zhiguo Cao, and Armin B Cremers. Multi-scale region-based saliency detection using W2 distance on n-dimensional normal distributions. In *IEEE International Conference on Image Processing*, 2013.

# Erklärung der Urheberschaft

Hiermit versichere ich an Eides statt, dass ich die vorliegende Master's Thesis im Studiengang Intelligent Adaptive Systems selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel - insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Ort, Datum Unterschrift

# Erklärung zur Veröffentlichung

Ich stimme der Einstellung der Master's Thesis in die Bibliothek des Fachbereichs Informatik zu.

Ort, Datum Unterschrift