An Efficient Saliency Driven Approach for Image Manipulation

Dylan Seychell

Supervisor: Prof Ing. Carl J. Debono

Department of Communications and Computer Engineering

University of Malta

October 2021

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Faculty of ICT

Declaration

I, the undersigned, declare that the dissertation entitled:

An Efficient Saliency Driven Approach for Image Manipulation

submitted is my work, except where acknowledged and referenced.

Dylan Seychell

October 2021

MALTA GOVERNMENT
MGSS
SCHOLARSHIP SCHEME

Dylan Seychell was supported by
the Malta Government Scholarships Scheme for his PhD Studies.

Dedicated to Christine, who was upgraded from girlfriend and fiancée to wife throughout the writing of this thesis.

Acknowledgements

This journey bears the name of one author. However, it was with the support of many others that I managed to complete it. I dedicate these two pages to all those who made a difference in this journey of discovery in science and life.

I wish to start by thanking the true north of this journey, my supervisor Prof Ing Carl J. Debono. Navigating the research space of computer vision over the past eight years was not easy. With his remarkable level of professionalism, Prof Debono enlightened the way, even when the situation seemed unreasonable. His understanding and guidance taught me a lot and positively influenced my career.

When venturing into less defined topics, bouncing off random ideas to push limits is of the essence. For this, I am thankful to my friend, colleague and sounding board, Mark Bugeja. He was always there to challenge me and support me in ventures that, at first, seemed far less possible than others.

The design and development of the COTS dataset was a challenge within itself. Its successful and complete development would not have been helpful without the help of Mr Matthew Sacco. He assisted me in the process of setting up the environment. Thanks go to Ryan Azzopardi, the professional photographer. He assisted in setting up the studio and providing lighting equipment for the COTS dataset setup. I am also very grateful for the help of Mr Jeremy Borg, who assisted me in the data collection process. I am also thankful to all the anonymous 2254 participants of the online tests. They dedicated time to the process and provided such precious feedback and information.

I am also appreciative of all my current and past colleagues and students at the University of Malta and Saint Martin's Institute of Higher Education for their support and understanding during this journey. I particularly thank the academic and administrative staff in the department of Artificial Intelligence for their support. This learning experience was also enhanced by the Google Developers Group that I was selected to start in Malta in 2015. I am thankful to this entire global

community, mentors at Google and the local organisers: Mark Bugeja, Prof Alexiei Dingli and Dr Claudia Borg.

I wish to thank all those present in the last stages of this process and preparation for the defence. These are Prof Matthew Montebello, Prof John Abela, Dr Kristian Guillaumier, Dr Colin Layfield, Dr Claude Bajada, Chantelle Saliba and Neil Micallef.

Last but definitely not least, I am grateful for the love and support of my family and close friends. They patiently supported me in yet another challenging venture. In particular, special thanks go to my parents, Rita and Charles, and my brother Aldrin. They were there for me through all the phases of my life. Outstanding support also came from my in-laws, Charlie, Theresa and Caroline Debono. I am also thankful to the friends who took a particular interest in this venture, namely Mandy, Matthew, Neil, Mariah, Jessica, Kieran and Conrad.

The person without whom this journey would not start is my wife and travel companion, Christine. Besides letting me include the tokens we acquired during our unconventional travels in the COTS dataset, she patiently supported me throughout my studies. At the same time, I literally took the writing of this thesis to every corner of the world.

Abstract

With the increasing availability of low-cost high quality cameras, embedded vision systems, advanced computer vision algorithms, and proliferating solutions based on image/video data, the volume of visual content that is being captured, stored and transmitted is on the rise. Moreover, the image capturing hardware on mobile devices is also improving with a wide range of devices also housing a multiview camera setup. When combined with today's user experience expectations, this poses a challenge to the editing process from which users expect more efficient results in the most automated possible way. Image editing is a multistage process that spans from the choice of the object or target region in the image for editing to the actual manipulation.

We introduce a novel saliency-driven image content ranking approach that allows for automatic selection of objects without the need of training the model. Regions in an image can be selected according to the desired rank. This approach was compared with human behaviour when choosing the most salient object in an image within experiments that involved 2254 participants. The results obtained by the algorithm matched the behaviour of 91% of the human participants. The technique also scored a F_β measure of 0.84 on the MSRA10k dataset and compares to normal saliency detection models that, unlike this technique, do not rank saliency. We also demonstrate how our saliency ranking model can be combined with segmentation techniques. The combined result of our saliency-driven ranking approach of segmentation masks compared well with the current deep learning state of the art methods that rank segmented objects.

Once an object is selected for editing, users expect an efficient way to accurately manipulate images. This fundamental stage is explored in our work where we demonstrate the importance of object inpainting. The main challenge of image inpainting is its objective evaluation and this work presents a new structured approach to objectively evaluate inpainting algorithms.

User studies with 2254 participants demonstrated that, on average, users take 3.67s to choose an object for editing in a screen. The combined saliency-driven image manipulation framework takes advantage of this physical limitation and efficiently pipelines processes to deliver an accurate and efficient result in image manipulation tasks such as attention re-targetting. A multi-purpose dataset was designed and built to serve all these functions and is also presented in this work.

Contents

List of Acronyms

AI Artificial Intelligence

BR Blending Result

CNN Convolutional Neural Network

COTS Common Objects of a Travelling Scientist

DFD Data Flow Diagram

DS Depth Score

EBI Exemplar Based Inpainting

FASA Fast, Accurate and Size-Aware Salient Object Detection

FMM Fast Marching Method

FPN Feature Pyramid Network

GANs Generative Adversarial Networks

GMM Gaussian Mixture Model

GT Ground Truth

IOU Intersection Over Union

IR Inpainting Result

MAE Mean Absolute Error

mAP Mean Average Precision

MOS Mean Opinion Score

MRF Markov Random Fields

MSE Mean Squared Error

MR Minimum Rank

PDE Partial Differential Equation

PFAN Pyramid Feature Attention Network

SARA Saliency-Driven Object Ranking

SegDim Segment Dimension

SM Saliency Map

SOR Salient Object Ranking

SSD Sum of Squared Differences

RG Region Growing

R-CNN Region-based CNN

RAS Rank Agreement Score

ROI Region of Interest

RPN Region Proposal Network

XAI Explainable AI

1. Introduction

> If everything seems under control,
> you are not going fast enough.
>
> Mario Andretti, Formula 1 Driver

1.1 Introduction

The volume of digital images captured in different natural environments increased over the past decade due to the availability of affordable digital cameras, particularly those embedded in mobile devices. This huge increase in content brings along with it an increase in demand on the way people interact and manipulate the same content. Moreover, the number of mobile devices equipped with multiview camera setups is also on the increase, and this will usher the era of consumer RGB-D data collection. In April 2020, the first commercial devices with depth sensing technology were also released. The content editing expectations will also evolve accordingly. However, there is a need for more tools to fulfil these expectations.

With the increase in popularity of augmented reality and virtual reality, the need for more efficient ways of handling 3D content is also on the rise. Just as today it is perfectly normal for anyone to take a 2D photograph of a scene through a mobile device, it will also be natural to take a 3D representation within the next few years. This is also coupled with the challenges faced due to the limited computational power of portable devices when compared to cloud capabilities. This limitation is particularly visible in the training and configuration of deep learning techniques. While deep learning methods work well on trained models, the quality of experience is not as high when the system is in an unknown domain or handling objects of an unknown class.

Within the context of this thesis, image manipulation or editing refers to processes that remove or add objects in an image. The crucial first step of the editing process is the selection of the target object or region in an image where the manipulation is to take place. Current methods range from a simple bounding box around the target object [2] to semi-automatic techniques that allow for a polygon to be drawn around the same object, such as [3]. This selection process may require substantial manual, human interaction. In today's expectations of needing to carry out computerised tasks in the shortest possible time, this process might appear to be lengthy within the commercial context. In the field of interaction design, there is also a push towards the idea that *"The best interface is no interface"* [4]. We kept this as a guiding principle, which also led us to successfully develop an approach that does not require any training to select content from a scene in a way that matches human behaviour.

Saliency detection is a popular way of approximating the way human fixation takes place when presented with images [5]. The current approaches that employ saliency detection to apply different forms of image manipulation [6] still depend to a certain extent on human interaction.

Moreover, these techniques interpret a greyscale saliency map resulting from a saliency detection algorithm in small patches. The main implication of this approach is that the importance of saliency can only depend on the values of the few pixels present in the processed patch. To date, saliency maps do not include any ranking or priority of salient regions in an image concerning objects found in a scene. This limitation is inherent from the fact that current saliency detection benchmarking datasets and salient object detection only deal with scenes containing single objects [7]. This gap in research motivated us to pursue the problem of devising a saliency detection approach that can rank objects based on saliency within an image to allow for the coverage of multiple objects. We are also ushering this saliency ranking technique during the time where the importance of Explainable AI (XAI) is on the rise. XAI is the field of study that aims to make conclusions of computer models understandable by humans by providing a traceable set of reasons on how a conclusion is reached [8] [9]. For this reason, the saliency-based object ranking model presented in this thesis is also designed and built to either provide an explainable output itself or when used in conjunction with other techniques.

The novel approach of ranking objects based on saliency in a scene that contains multiple objects also highlighted the limitations of current datasets [5]. This meant that these datasets were not adequate and could not be used. Therefore, the need for a new dataset tailored to fill this gap arose. The COTS dataset [10], introduced in this thesis, was designed, built and made available to the public for free to address this gap.

The second stage of the editing process involves image manipulation. Techniques such as object removal and inclusion fall in the category of image manipulation. In this thesis, object removal is referred to as inpainting and inclusion is referred to as object blending. Over the past decade, this has been a very active research area in computer vision with deep learning techniques such as convolutional neural networks, artistic neural style transfer [11] and generative adversarial networks [12] introducing a paradigm shift in the way we think about image manipulation. Throughout, we identified the lack of an objective way to evaluate the output of inpainting methods and proposed a novel technique to address this.

Having started in 2013, this research took place throughout this fast-paced development of deep learning applications in computer vision. Supervised deep learning approaches have been proved to return outstanding results in certain situations. However, this level of quality needs to be considered within the context of the computational cost and volume of data required to train such techniques.

The progress of deep learning approaches introduced a new perspective on how computer vision applications are designed and built. The AlexNet object detection model [13] in 2012 brought in this motivation by successfully demonstrating how deep learning can be used to identify and detect classes of objects in images. At that stage, object segmentation was seen as a separate problem until the successful results of Mask R-CNN [14] in 2017 where pixel level segmentation was made available for every detected object. This thesis focused on the separate effort of ranking objects according to their saliency in a scene. Post September 2019 work [1] [15] shows how the next evolution of object detection and segmentation methods is actually saliency ranking of objects. This recent work also argues how effective measurement of such methods is still an open problem. Current object saliency ranking models [1] [16] [15] rely on trained deep learning models and are also highly dependent on the training data upon which they are trained.

All these gaps combined usher the main gaps that currently exist in the emerging area of attention re-targeting. The main gaps in this area are related to the dependency on user control [6] and the lack of an objective framework to measure its effects [17]. Another challenge is that attention is subjective to users in question particularly because existing models [1] [16] [18] depend on gaze data [15]. This also introduces user subjectivity and potential bias.

From a more general perspective, it was also noted that for the evaluation of different computer vision task, one needs to use different datasets where every dataset would be dedicated to a single task. This makes the evaluation of pipelined computer vision application challenging and makes comparison difficult.

1.2 Problem Definition

Image editing software depends on two major stages. The first stage is the selection of the object that needs to be edited while the second stage is the application of some form of manipulation upon the selected object. This thesis addresses two problems related to this editing process.

The first problem is encountered in the object selection stage. When editing images, users are required to make unassisted decisions when choosing the target object. This can be a decision based on an artistic motive or a more definitive one such as removing a specific object. Such decisions are currently unassisted and the software does not provide contextual suggestions about what are the implications of choosing a specific object. The general saliency or attention of an image depends on its composition and objects. The choice of which object to edit has implication on the final image and users need assistance in choosing the most or least salient object in an image. The latest object saliency ranking models [1] [16] [15] rely on trained deep learning models and are also highly dependent on the training data upon which they are trained.

This ushers the main gaps that currently exist in the emerging area of attention re-targeting. The main gaps in this area are related to the dependency on user control [6] and the lack of an objective framework to measure its effects [17]. Another challenge is that attention is subjective to users in question particularly because existing models [1] [16] [18] depend on gaze data [15]. This also introduces user subjectivity and potential bias and introduce the need for an objective pixel-level approach.

The second problem is related to the manipulation process itself. Within the context of this thesis, image manipulation refers to the removal or inclusion of objects in an image. The field of image inpainting includes techniques that can effectively remove a region of an image, potentially containing an object, and replace it with a background consistent with the rest of the region within the image. Such techniques are generally evaluated using subjective methods and the use of full-reference metrics cannot be used for the inpainting of larger areas [19]. This is due to the lack of inpainting ground truth in datasets that would take the form of an identical image of the same scene without the inpainted object for comparison.

The gap being addressed in this thesis is the exploration of an objective metric and approach that assesses inpainting and blending given the current context where existing datasets do not have any groundtruth of the scene with or without the target object.

All these research gaps and challenges combined served as a clear basis and direction for this thesis that can contribute to improving the image manipulation experience.

1.3 Aim and Objectives

The aim of this research is to contribute to the field of computer vision by developing a saliency driven approach to rank objects or regions within a RGB-D scene upon which image manipulation operations can take place.

In order to achieve this aim, the following objectives were set up:

1. Automate object selection within images by making use of a saliency-driven method to facilitate manipulation of images.
2. Design and build a multipurpose RGB-D dataset that can be used to evaluate different computer vision applications.
3. Devise an objective approach to evaluate image inpainting techniques that allows for fair comparison of different algorithms.

1.4 Main Contributions

A Saliency Ranking Model We created a saliency-driven ranking model, (SARA), that does not require any training and can efficiently rank the visual saliency of an image, presented in Chapter 5 and published in [20]. This technique can work with or without depth information. The experiments presented in Chapter 7 demonstrate how this technique returns predictions that compare well to human behaviour. Moreover, this method ushers a new way of achieving and evaluating attention re-targeting in an image.

Ranking the Output of Segmentation Methods This thesis also demonstrates how the proposed saliency ranking model can be used to rank the masks in the output of modern segmentation methods. Experiments demonstrate that when our model is used to rank the output of Mask R-CNN, comparable results to deep learning based segmentation methods are achieved.

Intra-Object Segmentation Exploited the nature of RGB-D images to efficiently decompose the object of interest into layers. This makes it easier to manipulate 3D images in operations such as inpainting and blending. This work was published in [21].

Objective Evaluation of Inpainting The contribution of inpainting larger objects or regions in scenes that was published in MELECON16 [22] might today be overshadowed by Generative Adversarial Networks (GANs) inpainting that featured after that publication. However, the objective evaluation approach achieved through the COTS dataset still stands and can be used for any inpainting approach, whether it is Exemplar Based Inpainting (EBI) or through GANs. The objective evaluation approach was published in [23].

Multipurpose RGB-D Dataset The modular nature of the process and framework presented in this thesis highlighted the need for a multipurpose dataset that can be used to evaluate the framework throughout all its stages. For this reason, we designed and built a RGB-D dataset that serves this purpose as presented in Chapter 6, was published in [10] and is made available online to the public for free on www.cotsdataset.info.

1.5 Publications

The main milestones of this study were compiled in papers and published in international IEEE peer-reviewed conferences and journals.

The first one, *Monoscopic Inpainting Approach using Depth Information* [22], was accepted and published in the Proceedings of the 18th IEEE Mediterranean Electrotechnical Conference (MELECON) in 2016.

The second one, *Efficient Object Selection using Texture and Depth Information* [24], was accepted and published in the Proceedings of the 2016 IEEE Conference on Visual Communications and Image Processing (VCIP).

The third one, *Intra-Object Segmentation using Depth Information* [21], was accepted and published in the Proceedings of the 19th IEEE Mediterranean Electrotechnical Conference (MELECON) in 2018.

The fourth one, *Ranking Regions of Visual Saliency in RGB-D Content* [20], was accepted and published in the Proceedings of the 2018 IEEE International Conference on 3D Immersion (IC3D).

The fifth one, *An Approach for Objective Quality Assessment of Image Inpainting Results* [23], was accepted and published in the Proceedings of the 20th IEEE Mediterranean Electrotechnical Conference (MELECON) in 2020.

The sixth one, *COTS: A Multipurpose RGB-D Dataset for Saliency and Image Manipulation Applications* [10], was accepted and published in the IEEE Access Journal, Volume 9, pp. 21481-21497, 2021.

1.6 Thesis Overview

This thesis is organised into 3 main parts. The first part will present relevant literature to the reader about the key topics tackled in this thesis. The second part will demonstrate how the theory explored in the first part was brought together to tackle the above mentioned gaps. This starts with preliminary experiments that were carried out that eventually resulted in the Saliency Ranking method. The final part presents the COTS dataset together with the evaluation of the Saliency Ranking technique followed by an investigation of possible future work in this area and respective conclusions of this work.

Part 1 - Literature Review This part is organised in two main chapters covering literature and research about existing relevant topics for this thesis. These chapters cover techniques related to Visual Saliency and Image Manipulation.

1: Visual Saliency The first chapter covers the key topics and theory behind visual saliency. It starts by providing background about the human memory and visual attention, the theory upon which saliency detection techniques are based. Different types and techniques of saliency detection are then surveyed. This is followed by a survey of early attempts at ranking saliency and past interpretation of the topic. This chapter is concluded with a survey of novel saliency ranking approaches that rank different objects in a scene using deep learning models.

2: Image Manipulation The second chapter deals with background related to Image Manipulation where in the context of this thesis, it refers to the selection of an object and its removal or addition to an image. This chapter starts with an overview of segmentation methods ranging from traditional approaches to the latest state of the art deep learning methods. The second major component of this chapter provides coverage of inpainting methods. Different approaches are surveyed together with more recent generative deep learning methods.

Part 2: Methodology Following the coverage of relevant work in Part 1, the second part of this thesis deals with the application of these topics into the design and implementation of a saliency driven approach that enables image manipulation.

3: Preliminary Experiments This chapter presents two major experiments that were carried out and motivated the development of the main contributions of this thesis. The first one is a single click object selection method that uses colour and depth information from RGB-D images. The second one is an intra-object segmentation method that splits objects in different layers of colour that correspond to the depth information.

4: Saliency Ranking Approach for Image Manipulation This chapter outlines one of the main contributions of this thesis. It outlines an approach that ranks different parts of the image based on their saliency. This approach returns a ordered list of segments of the image by their saliency level, therefore indicating which parts of an image draw more or less attention. This chapter also demonstrates how the proposed saliency ranking model can be combined with any segemntation technique to rank its output masks.

Part 3: Consolidation The last part of this thesis deals with the evaluation of the saliency ranking approach together with the presentation of a new dataset for the evaluation of computer vision applications. This part also analyses the findings of this thesis and explores the potential of different saliency ranking approaches in computer vision.

5: COTS Dataset While working on different aspects of image manipulation, it was evident that the lack of datasets was hindering objective evaluation. It also meant that techniques needed to be either evaluated using subjective approaches or rely on datasets that were not intended for this purpose. In our case, this was evident when we worked on inpainting methods [22] and for that reason we designed and built the COTS dataset [10] that can be used to evaluate inpainting methods [23] and other computer vision applications.

6: Evaluation This chapter presents a detailed evaluation of the saliency ranking approach. It starts investigating key aspects of the approach such as the experiments behind the choice of the grid size and ablation studies on its key components. The technique is then benchmarked with other saliency detection techniques. Due to its novelty, user based experiments were also carried out to compare the result of the system with human behaviour. The proposed saliency ranking technique combined with Mask R-CNN was also evaluated against the comparable deep-learning based state of the art techniques.

7: Conclusion This chapter briefly reports the key topics covered in this thesis while binding them to existing research in the field. This concluding chapter then briefly goes through the details of implementation together with its respective evaluation while it is concluded by summing up all efforts in this thesis.

1.7 Conclusion

This chapter presented the motivation behind this report by giving an overview of the work presented together with the problem definition. The aim and objectives of the thesis were presented and they were followed by the main output and contributions together with an overview of the chapters found in this thesis.

2. An Overview of Visual Saliency

The true mystery of the world is
the visible, not the invisible.

Oscar Wilde

2.1 Introduction

This chapter outlines the relevance of visual saliency in the field of computer vision. It starts by providing background about human attention which provides the basis of visual saliency. This section is then followed by an overview of different saliency detection methods in computer vision. These different techniques model a colour image into an 8-bit saliency map where every pixel represents the level of saliency for the corresponding pixel in the colour image. Most of the work in this field of study focused on the accurate generation of these saliency maps. This is followed by a section that investigates earlier attempts at ranking saliency where this was interpreted as ranking the individual pixels without any particular context. This chapter is concluded with an overview of recent work that ranks objects based on saliency, which is the field to which this thesis contributes directly.

2.2 Background

As human beings, we rarely stop and take our time to appreciate the volume of information that we experience from the world around us. This large volume of information is received through our senses. Think about the smells around us, the sounds we hear while choosing to which we will listen, the temperature affecting our skin and implicitly changing our behaviour, the taste of a coffee we had few moments ago and finally all the things we see around us. Through our visual system, it is estimated that our retina is bombarded by 10 billion bits of information per second [25] [26]. Despite this large volume of information, our brain processes a smaller amount of information due to the limited capacity of the optic nerve that is approximately 6Mbits/s. Moreover, the brain is not capable of processing all this information at conscious level and ends up spreading it over the different types of memory. This results with only 100bits/s of information being processed at a conscious level [25].

The ability of making efficient use of this information is vital for survival. Furthermore, it is important that the right information is stored for retrieval [27]. The human memory model is divided into three parts: sensory, working and long term memory [28]. The information being processed through our sensory system is processed in sensory memory for a short period of time and for this reason, it is also referred to as very short term memory store. This information received from our sensory receptors can be haptic (touch) information, echoic (sound) information and, most commonly, iconic (visual) information [28]. A selection of the information at sensory level is then transferred to working memory for further processing by the brain [29]. Information is stored in working memory for a period of time that this relatively longer than that at sensory memory but it is still considerably limited. For this reason, working memory is also known as short term memory. Due to the relevance of Working Memory to this research, Section 2.2.1 is dedicated to exploring this aspect in deeper detail. Working memory is responsible for chunking information for the brain to process it accordingly. This selection is known as attention and the visual aspect of attention is explored in Section 2.2.2. Selected information that we repeatedly experience or rehearse is then stored in the last memory store known as long term memory [28].

2.2.1 Working Memory

The part of the process responsible for the preliminary processing of information after it is received from the human sensory receptors is known as working memory. The capacity of this store is important since it is used to temporarily hold information for further processing [30].

Working memory is itself a theoretical representation and due to its abstract nature, it has always been difficult to measure its limited capacity. The first theories venturing into this understanding suggested the idea of information being stored in chunks [29]. The concept of chunking states that information processing for young adults can handle seven chunks or elements, plus or minus two [31]. A chunk is a unit of information or content that a person can memorise for further processing.

All information received at sensory level can be chunked. For example, phonetic segments in a sentence can be considered as chunks and the same chunks are still processed by the brain even if spoken in a language that is unknown to the listener. Later research [32] suggested that chunks can also contain different types of information such as visual and text. This notion of chunking inspired the idea of segmenting an image in regions of interest for more efficient processing, as explained in Section 5.3.1.

The human brain processes a limited amount of information in an environment generating far more information than the same brain can handle at one point in time. The capacity of the working memory is one of these bottlenecks and the same limit on its capacity results in humans being easily distracted by other information and while needing to focus on tasks of a narrower nature [33]. Cognitively, our brain filters visual information around us by having our attention diverted to specific objects that are more salient than others as explored in more detail in Section

2.2.2.

2.2.2 Visual Attention

The selection of visual information is carried out through a set of cognitive mechanisms processing signals throughout the visual system. This is known as visual attention [34]. Cognitively, the purpose of visual attention is to reduce the volume of information that needs to be processed. The processing of visual data includes the binding and enhancement of features that are used in the recognition process [34]. The brain processes visual information in two stages. In the first stage, an entire scene is processed, and in the second stage, the system focuses on a single area for further detailed processing.

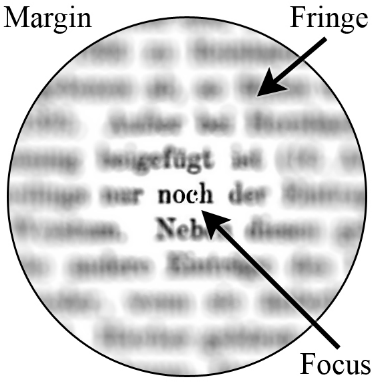


Figure 2.1: The spotlight model of visual attention [35] (Source: [36])

The first stage is also referred to as the spotlight model where the scene is organised as an area of focus at the centre and peripheral information [35] as represented in Figure 2.1. The focus at the centre is processed at the best resolution and it is the area from which the brain extracts the most information from the scene. This high-resolution area is where visual attention is focused. For a quick appreciation of how this works, one can extend his/her arm with the thumb up. Upon focusing on the thumb, one can experience a clear visual representation of that small area and the rest appears to be blurred. The blurred area is known as the margin. This is the cut-off region of visual processing. The area between the focus and the margin is known as the fringe which is a gradual change in focus. Information in the fringe is processed at a lower resolution [35]. The results presented by Judd *et al.* [37] resulting from their eye-fixation based saliency detection model reaffirms this theory. The spotlight model inspired the way the output of the saliency ranking technique presented in this chapter is composed as presented in Section 5.4.

Attention is also differently oriented over a scene through 'overt' or 'covert' orienting [38]. Overt orienting is the act of willingly and selectively focusing the visual attention onto an area of interest. Eye-tracking techniques would be visualising and representing overt attention orienting. On the other hand, covert orienting is the shifting of attention without any eye movement. In such an orienting approach, attention is attended towards a single stimulus among a selection of stimuli present in the same region. Studies [38] suggest that overt and covert orienting do not operate independently and both contribute to the focus of visual attention.

The overt and covert notion of orienting visual attention leads to the key ideas behind the way we interpret salient regions within a scene. General theories of visual attention assume bottom-up and top-down processes to converge within the same part of the brain [38]. Bottom-up processes are based on reflexive information induced by stimuli in the scene. On the other hand, top-down processes are based on voluntary focusing on a high-level specific object in a scene. For example, noticing a red area over a light background and then covering a broader area to understand that it was a person wearing a red shirt in front of the blue sky would be the result of the bottom-up process. On the other hand, the voluntary focus on a person and then deducting he/she is wearing a red shirt is the top-down process in action. This notion is the fundamental theory behind the main approaches in saliency detection techniques that are reviewed in detail in Section 2.3.

2.3 Saliency Detection Techniques

Saliency detection in computer vision can be approached through either top-down methods or bottom-up ones. This follows the same process that takes place in the biological saliency detection explored in Section 2.2. The bottom-up method is based on features belonging to a neighbourhood or small region and does not consider any object-level target. These techniques are stimulus-driven that explore low level vision features [39]. The low-level attributes and features of a scene are those that influence this approach the most [40] [41] and are normally evaluated in relation to eye-fixation models [42]. On the other hand, it is the high-level or object-level properties that drive top-down saliency detection techniques. These are based on a target interpretation of the image that is deduced following a training of a model [43] and such techniques are generally slow to train [40].

This thesis uses saliency for ranking of objects in the context of images. There are different techniques that are starting to use saliency on video content [44] [45] [46] [47] in different applications. These include the use of saliency detection in video to detect changes in the scene [46]. Another application area is 360-video content [48] [49] where saliency is used detect sections of the scene that attracts attention.

This Section reviews three saliency approaches based on eye-fixation methods, information theory approaches and deep learning approaches. Subsection 2.3.1, presents eye-fixation approaches that were motivated by the first of these methods proposed by Itti *et al.* [50] followed by other techniques [51] [52] [53]. Another set of approaches in saliency detection use information theory and statistical information such as the work presented in [37] [54] [41] surveyed in Section 2.3.2. The most recent approaches in saliency detection make use of deep learning [55] [56] [57] [58] [59] [60] [61] [62], where models are trained to generate saliency maps in colour images.

2.3.1 Eye-Fixation Saliency Detection Methods

Observation of biologically driven saliency detection inspired the first techniques. In the early work in saliency detection, the knowledge of visual attention system of primates was used to generate a saliency map [50]. It followed that by design such techniques would need to quickly detect saliency in a given scene. This results in the rapid interpretation of features within a scenes and their decomposition into a topographic map. Due to the focus on specific features within the image, saliency maps allow for a clearer distinction between the image foreground and background regions.

The original saliency detection technique was proposed by Itti *et al.* [50]. This method starts by filtering linear features in the image considering colour, pixel intensity, and orientation [50]. These are low-level features that make this technique a bottom-up approach. The second phase of this technique normalises the previously detected low-level features. These features are mapped for an across-scale normalisation process. The map normalisation operator promotes maps in which a small number of intense peaks of activity are present, resulting in the suppression of maps that contain many comparable peak responses. The feature maps are combined into three conspicuity maps. This is obtained through across-scale addition by reducing the scale of each map and point-by-point addition [50]. This results in a preliminary saliency map. The final stage of this technique uses a "winner-take-all" neural network for the processing of the saliency map built through the earlier stages.

This method gives priority to the most active regions in the image while the other less important elements in the image are suppressed and therefore end up not being prominently featured in the saliency map. A selection of examples to demonstrate the output of Itti's method is presented in Figure 2.2. Besides being the first work in this area that generated significant interest, the work in [50] is also commended for its efficiency due to its parallel implementation and its performance in complex natural scenes. This property was one of the main motivations for using

Itti *et al.* saliency map generation technique as a default in the technique presented in this thesis, as discussed in Section 5.3.1.

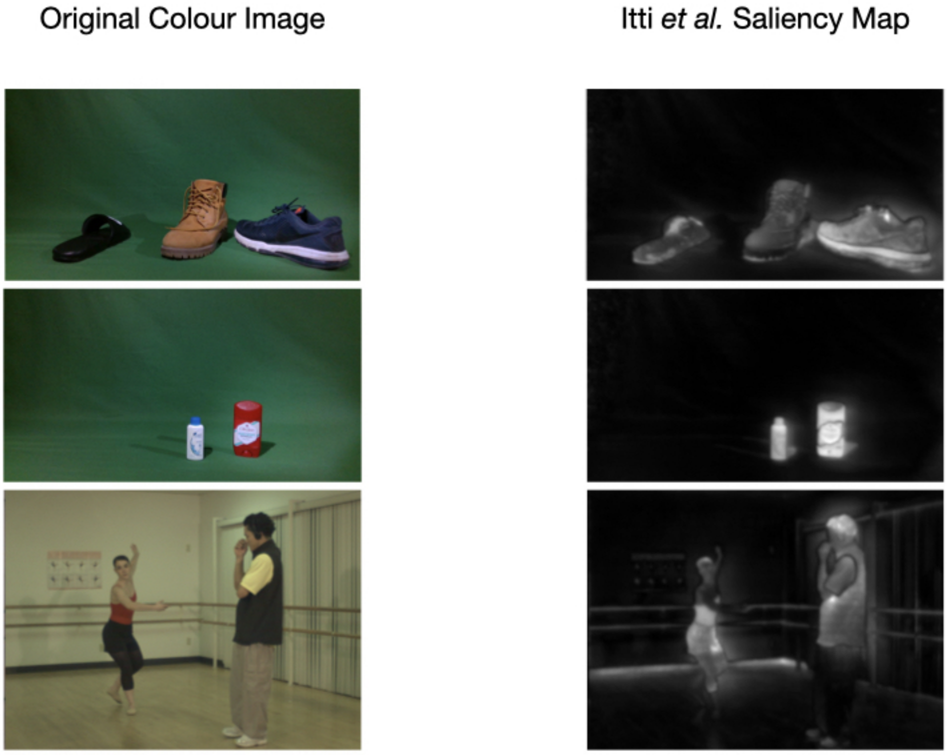


Figure 2.2: A selection of saliency maps of colour images used in this thesis. These saliency maps were generated using Itti's Method [50].

In his definition of a salient object, Borji [7] distinguishes saliency detection from eye fixation prediction. He differentiates between "where people look" (free-viewing of a scene) and "which objects stand out" (detect salient objects) [7]. The work of this thesis is placed in the latter category where the work presented models which objects people would choose. The various online tests also follow this line of thought and are aimed at testing which objects users choose.

Margolin *et al.* [53] discuss what makes a patch distinct from other patches in the same images. This approach analyses the colour and pattern attributes in an image. When processing an image, this technique first captures unique features that relate to the object structure. The second phase processes the colour information and identifies the distinct features. The concluding phase takes the two sets of features and finds what is common in both and concludes that it is the most distinct region in the image [53]. This approach enhances the saliency result by prioritising objects that are distinct in terms of colour and pattern. It does not return any form of ranking of saliency. In the few examples presented where there are multiple objects in a single image, this technique appears to makes them stand out in the saliency map depending on their colour and pattern scores.

Centre-Bias

When presented with a scene, viewers tend to fixate on the central region of the image and therefore the elements found in that area of the scene [40]. This effect is known as centre-bias. This idea is further supported by experiments where the average annotation map of a collection of saliency maps was computed and found to point towards the centre [37] and others with the mean position of object locations [42] that gave the same result. This effect is manifested in most of the datasets in this field [5], and an example can also be seen in the result presented in Figure 7.1. A possible cause behind this phenomenon can also be an ergonomic attribute and experimental constraint where the subjects would have their heads at chin-rest during the fixation experiment [51]. Another potential cause of bias in datasets could also be the result of proportions used in photography such as the rule of threes when generating or choosing images that are included in datasets [51]. Different saliency detection techniques [37] [52] also make use of centre-bias in their generation of a saliency map. Results show that a 2D Gaussian distribution models this effect in a given image and a probability of fixation can be approximated in this way. The technique presented in this thesis allows for the consideration of centre-bias in the generation of the model leading to the saliency ranking. However, given that this can bias the results, a set of experiments without this mode were also carried out and show that the technique does not rely on centre-bias as demonstrated in Table 7.1.

Viewing Proximity

Fixation is also affected by whether an object belongs to the background or foreground of an image. The detection of salient objects in scenes that contain a complex background or similar features to the target object renders the problem of saliency detection even more difficult [63]. The human visual system is sensitive to the depth of field in a scene [63] and different work shows that objects in the background are less salient than those in the foreground [40]. Thus, saliency detection algorithms should handle background and foreground information differently [43]. The results presented in this thesis show that the consideration of depth is important when ranking objects by saliency. On the other hand, this does not mean that the background needs to be ignored when detecting saliency since it might also contain objects that have a significant level of saliency. Existing saliency detection datasets reviewed in several benchmarking papers [5] [51] [39] do not include depth information and till now it is challenging to concretely correlate depth with the level of the saliency of objects.

2.3.2 Information Theory and Statistical Saliency Detection Methods

Human visual attention is influenced by regions that contain most information or rate of change [64]. This served as an inspiration for the use of information theory in the detection of saliency. This regional maximal information was found to be adequately modelled using Shannon's self-information [65].

One of the major works employing this approach [37] featured a machine learning technique that used self-information to identify features with eye-tracking data over 1003 images generated by 13 users. A model was subsequently built through this data and it managed to successfully predict human fixation in an image. A similar approach was used for identification of super-pixels in images [54]. The super-pixels were represented as nodes in a graph and the weights of the edges represented the similarity between the super-pixels. However, such approaches are affected by the cost of the generation of the super-pixels [54]. These techniques are used as an alternative method to generate a saliency map, and the output is a complete saliency map [20] rather than a specified region of interest as presented in this chapter. Probabilistic approaches were also used to generate saliency maps. One example of these methods was the Fast, Accurate, and Size-Aware Salient Object Detection (FASA) [41]. This technique quantised the colours in the image and estimated their position such that a statistical model was built to predict saliency in the image. This technique achieved significant performance when processing video content [41].

2.3.3 Deep-Learning Saliency Detection Methods

The successful results of deep learning in different computer vision techniques over the past decade inspired a number of work to use this approach in the generation of saliency maps [55] [56] [57] [58] [42] [41] [59] [60] [61] [62].

These approaches are using hand-crafted annotated datasets to train their models that make use of a convolutional neural network (CNN) architecture. In some cases, the network architecture also employs different saliency detection techniques and therefore, approximates to a more general solution [42]. Such techniques use a single network that learns crafted colour and depth features, producing a set of joint predictions [66] as shown in Figure 2.3- approach A. Another approach uses two separate CNN architectures where each architecture handles colour and depth individually and their results are combined in a final layer that returns the joint predictions [67] as shown in Figure 2.3- approach B.

The deep learning approaches vary. The extraction of salient feature vectors from scenes [57] can be used for the network to learn the interaction between saliency features and RGB-D information. The raw saliency feature vectors contain information such as the local contrast of objects, global contrast, background prior, and spatial prior. This information was then embedded in the layers of a CNN

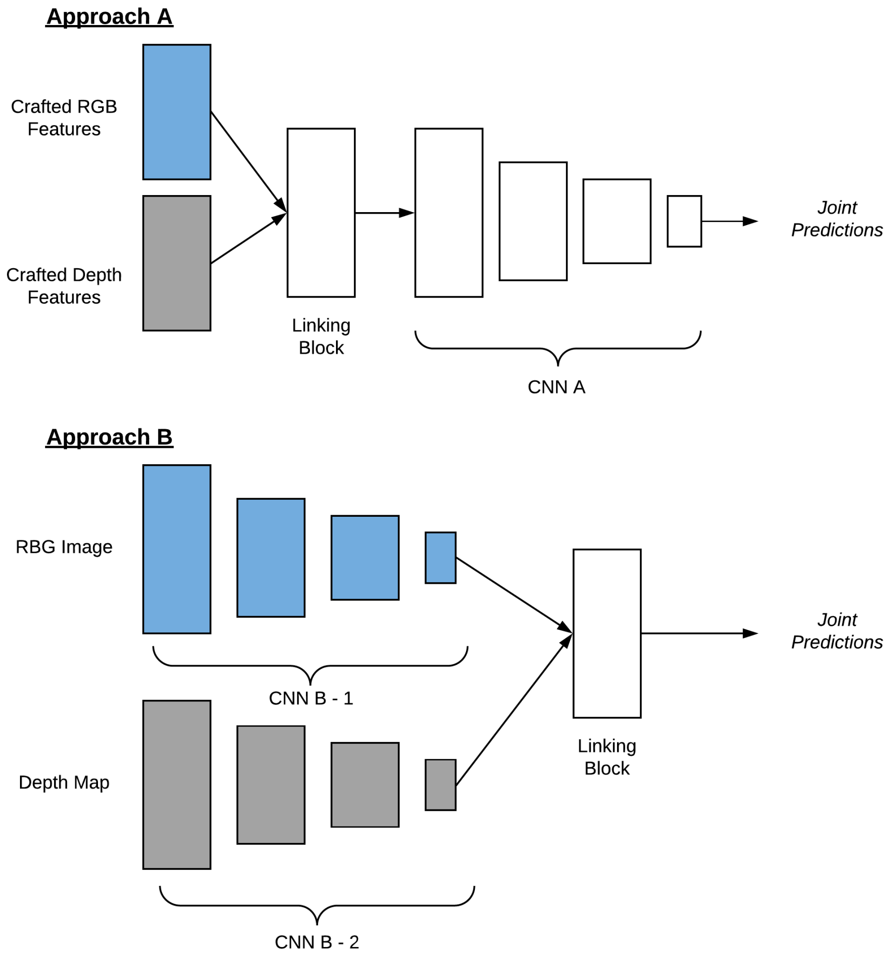


Figure 2.3: A high level illustration of the two deep learning approaches normally used in saliency map generation (Adapted from: [68])

that generated a saliency map as output [57]. RGB-D feature handling was also explored using a cross-modal approach where the color and depth stream were handled separately [68]. The work of Chen and Li [68] discusses an architecture that exploits the cross-modal nature of approximations in a progressive nature.

These deep learning saliency detection techniques can be classified as top-down approaches where, in some cases, it also includes low-level features. The main challenge in the use of deep-learning approaches is the necessity for an extensive labeled dataset that is uniformly sampled for training together with the computational resources required for such training [43].

More recent state of the art methods in this category combine salient object detection with object segmentation. BASNet [60] uses deep learning to generate boundary aware salient object detection and SCRN [62] similarly refines the salient object detection based on the edges of the salient object. These recent methods also give significant consideration to the performance of salient object detection with CPD-R [61] and BASNet also dedicating significant attention to the effect of the technique on the frame-rate when processing video or streamed content. S4Net [59] combines instance segmentation with saliency maps, stopping short from suggesting a rank for the results of the instance segmentation.

2.4 Early Attempts at Ranking Saliency

The review of saliency detection techniques presented in Section 2.3 reflects the significant effort carried out in this research area. These approaches provide several different saliency maps where the value of every pixel in the saliency map reflects the level of saliency of the same pixel in the original image. Within the context of the assumption that the original image contains a single object, such an approach may suffice. However, this is not always the case, and such models generally assume a single object in the scene [7].

The focus on scenes with single objects has proved to be useful in the development of these models and their respective benchmarking. The current challenge in the field of saliency detection is its use in more complex tasks such as scene description [5]. In order to be able to carry out such tasks, techniques that can rank objects within a scene according to saliency are essential.

In this Section, the current techniques that explore the idea of ranking saliency in an image are explored. While the technique presented in this thesis provides an approach that effectively ranks regions in the image from a global perspective, the techniques being reviewed tackle saliency ranking from pixel level.

2.4.1 Ranking Saliency

In their paper "*Ranking Saliency*", Zhang *et al.* [69] propose a bottom-up approach that detects salient objects in images. This work models saliency detection as a manifold ranking problem along with a cascade scheme of refined saliency generation represented as graph labeling. The final output is still a saliency map in the same format of the other techniques reviewed in Section 2.3. This work assumes that the image content is split in two where it can either be the salient object or the background [69]. It also follows the trend of using images with single objects. Only 4 out of 37 scenes presented in [69] contain more than a single object. In this 10% of the scenes, both objects are ultimately considered as a single object in the resultant output of the technique.

The graph-based manifold ranking is spread over several layers in the process of image analysis. Every node is considered as a query, and the links to the other nodes are ordered according to their rank with respect to the original node. A node is, therefore, a region from the background or salient foreground [69].

The first stage of this technique involves the use of boundary priors to obtain specific queries from the input image. The saliency of the nodes is computed as relevance/ranking to the background nodes. The saliency ranking score $\mathbf{f}^*$ for each node is computed as presented in Equation 2.1 where $\mathbf{D}$ is the degree matrix, α is a function used to control the balance of smoothness and $\mathbf{W}$ an affinity matrix. Matrix $\mathbf{A}$ is the learned optimal affinity matrix while $\mathbf{y}$ is a binary indicator vector [69].

$$\mathbf{f}^* = \mathbf{A}\mathbf{y} \quad \textit{where} \quad \mathbf{A} = (\mathbf{D} - \alpha\mathbf{W})^{-1} \tag{2.1}$$

This process results in the generation of four preliminary saliency maps. These maps are then integrated into a single saliency map S_{bq} using Equation 2.2. At this stage, every node in the combined map has a probabilistic measure to the other nodes. These measures are obtained and ranked over different iterations until a final refined map is generated [69].

$$S_{bq}(i) = S_t(i) \circ S_b(i) \circ S_l(i) \circ S_r(i), \tag{2.2}$$

where $\circ$ is an integration operator such as ×, +, *min* or *max*. $S_t(i), S_b(i), S_l(i)$ and $S_r(i)$ are the saliency maps with top, bottom, left and right boundaries as queries respectively. The super pixel being ranked is denoted by i.

This technique, therefore, uses background and foreground queries to generate a set of saliency maps. The components of these maps are iteratively ranked in a bottom-up approach in order to generate a final saliency map. The output map is a greyscale map with pixel level indication of saliency. The ranking is therefore used to generate the final saliency map that does not indicate any form of ranking within the scene by itself. The result of this technique can also serve as input to the ranking solution presented in this thesis such that the different segments of the saliency map can be ranked according to importance.

2.4.2 Other Approaches

The emerging importance of ranking saliency is also evident in other work and applications. Saliency provides an opportunity in data compression, and it can be used to guide progressive coding techniques. Rahul and Tiwari propose a JPEG framework based on saliency enabled compression [70]. This work makes a similar assumption to the other reviewed work by making use of images containing only a single object and categorizing the image into salient and non-salient regions. However, [70] introduces the idea of working with multiple ROIs.

The work of Rahul and Tiwari [70] generates a multi-level saliency map and segments the image into several salient classes. Their class variances are then processed, and every class contains pixels that are randomly distributed along with the image [70]. Each class is then ranked by a weighted variance of the pixels in the class. This work uses the JPEG 8×8 blocks where the blocks are then ranked using a probability mass function.

In separate work, Zhu *et al.* propose a saliency detection framework for complex scenes [63]. This work makes use of RGB-D information and identifies the most salient object in a scene with a complex background. This process is organized into three stages. The first stage pre-processes the scene and extracts the centre-bias, the second stage makes use of colour and depth features to generate a detection map and finally, the third stage fuses the detection map with center-bias to generate a final saliency map [63]. The result of this work still returns a saliency map that highlights the most salient object in the scene, and whenever there are multiple salient objects, they are still presented as a single object.

The model developed by Judd *et al.* [37] performs well when compared to where people look in an image. This approach is a supervised model that is trained on eye-fixation data. In certain instances, this work applies a heuristic of choosing the brightest top 10% pixels of the designated saliency map for comparative purposes on their resultant map generated through a support vector machine. This heuristic is used to compare the results of their technique with other techniques such as Itti's [50]. It therefore implies that the most salient regions are the brightest 10% of the pixels in the saliency map.

2.5 Object Level Saliency Ranking

The aim of salient object detection techniques focused on accurately measuring or detecting objects or regions in images that detect visual attention. The previous section surveyed the major work that tried to tackle the ranking of saliency in images from a point-processing perspective. Most work in saliency detection focused on datasets containing single objects as discussed above while a few ventured in datasets that contained more than one object. However, studies on how saliency changes or shifts between different objects in the same scene is still in its early stages [15].

This section surveys two very recent methods in Object Level Saliency Ranking. The first one was published in June 2020 and focuses on inferring attention shift rank [1] while the second one introduces the idea of Relative Saliency between objects [15], published in January 2021.

2.5.1 Inferring Attention Shift Ranks

This work of Siris *et al.* [1] focuses on replicating the human ability to shift attention from one part of an image to another as discussed in Section 2.2.2. The initial application of the work of Itti *et al.* [50] applied saliency maps to scene analysis. However, their main contribution was the actual generation of the pixel-level saliency map rather than the scene analysis aspect.

Siris *et al.* [1] proposed a deep learning network that is trained on human gaze data to model attention shift. Their contribution also includes a new dataset for salient object ranking based on the COCO [71] and SALICON [18] datasets. This augmented dataset was built by studying eye-fixation patterns of 11 subjects to determine shift rank order and is presented in more detail in Section 7.6.1. Based on this study, the ground truth of their dataset was based on distinct objects that were fixated one after the other while ignoring any objects that appeared more than once. The rank of saliency of an object was calculated in relation to such score across all observers [1].

The attention shift rank deep learning model, shown in Figure 2.4, was based on bottom-up features and inspired by Itti's work [50]. This model's backbone network is based on the Mask R-CNN [14] segmentation network and therefore only considers objects that are detected by this network. This network also uses a sliding-window to decide whether objects are in the foreground or background of an image. The Spatial Mask Module at the centre of the architecture considers the masks of different object proposals in order to make sure that even important smaller objects are considered. This module is reported to make use of the Centre-

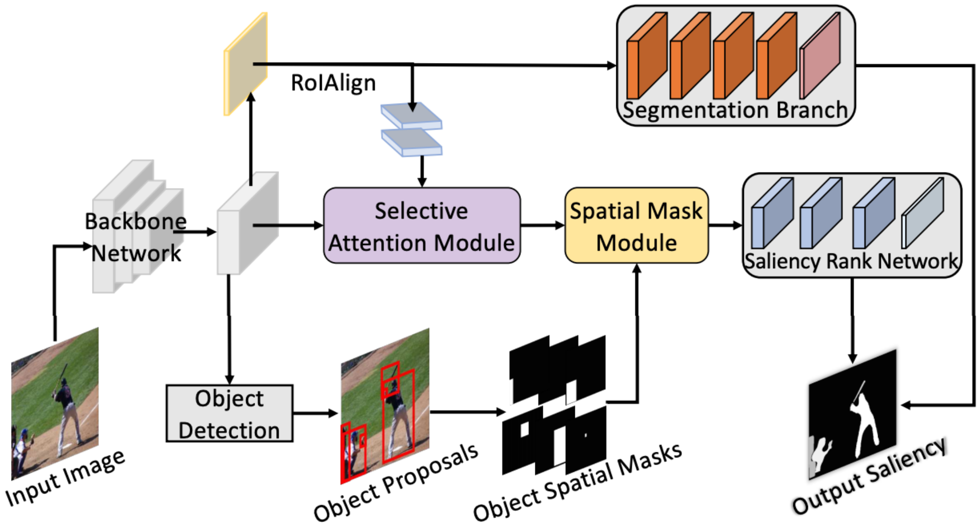


Figure 2.4: The network architecture for Attention Shift proposed by Siris *et al.* using Mask R-CNN [14] as the Backbone Network (Source: [1])

Bias principle [1] reported in Section 2.3.1.

The attention shift rank order at the latter parts of this architecture takes place in the Saliency Rank Network and returns a rank on the object proposal generated by the Mask R-CNN. The result of this architecture is limited to the first 5 objects and the authors consider higher ranks as future work [1]. It was reported that this model took 6 hours to train on a Tensorflow implementation and using an NVIDIA GTX 1080 Ti GPU.

This work was evaluated using the Salient Object Ranking (SOR) metric that was originally proposed by [16] and developed further in [15]. This metric is investigated in detail in Section 7.5.1. Siris *et al.* used the SOR metric to evaluate their model against a number of saliency detection methods although they also report that these other methods were not built to predict object level segmentation and respective detection. These methods, as reported in Section 2.3, produce a single saliency map where in some cases it is also a binary map [1].

Later in this thesis we demonstrate how our approach returns comparable results to this technique when combined with Mask R-CNN without the need of any training to rank the salient objects.

2.5.2 Relative Saliency

The work by Islam *et al.* [16] introduced the idea of ranking saliency by revisiting the concept of salient object detection and recently publishing a more detailed version of the same study [15]. Their work introduced a deep learning model that proposes a relative rank. This work was published at the same time that we published our initial work in saliency ranking [20] that is expanded further in this thesis.

One of the main contribution of [16] is the introduction of a stacked representation of the ground truth in saliency maps that paves the way for saliency ranking. The ground truth saliency map is defined as $\mathcal{G}_m$ where N maps of different participants are stacked as $(\mathcal{G}_i,\ \mathcal{G}_{i+1},\ldots,\mathcal{G}_N)$. This is visually presented in Figure 2.5. The nested nature of the ground truth is defined by $\mathcal{G}_{i+1} \subseteq \mathcal{G}_i$ and this results in new objects being presented in every new instance of the stack.

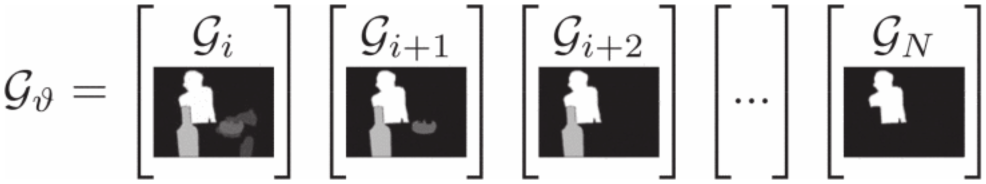


Figure 2.5: The stacked representation of ground truth of saliency maps as proposed by Islam *et al.* (Source: [16] [15])

This work [16] [15] presents a feed-forward deep learning network named RSDNet that is based on ResNet-101 [72] trained on a version of the Pascal-S [51] dataset. The output of this network trained on the stacked ground truth returns multiple salient objects. Another significant contribution of this work was that it started including multiple objects in the salient object detection output, something that was not being done by other algorithms. This work, nonetheless, does not provide an indication of the training time for their proposed network.

The evaluation of RSDNet was carried out as a comparison with state of the art salient object detection methods that were not returning any rank for the detected objects. In order to evaluate the matter of rank, the authors carried out a qualitative study by presenting visual results. In the absence of a metric to evaluate saliency ranking, they proposed the Salient Object Ranking (SOR) metric that is discussed in Section 7.5.1. In their recent update [15] to their original work [16], the authors still claim that the main challenge for saliency ranking is the lack of universally agreed metrics for evaluation.

2.6 Conclusion

This chapter demonstrated how throughout the past couple of decades, visual saliency witnessed an evolution from pixel level classification of saliency to ranking objects by their salient value.

3. Image Manipulation

> Do not try and bend the spoon,
> that's impossible. Instead, only try
> to realize the truth...there is no
> spoon. Then you'll see that it is
> not the spoon that bends, it is
> only yourself.
>
> Spoon Boy, The Matrix movie

3.1 Introduction

This chapter aims to provide background on the fundamental topics that are used within this thesis to demonstrate a selection of methods for image manipulation. In order to focus the scope of this thesis, Image Manipulation is understood to refer to the removal of objects from an image or the inclusion of a new object in the image.

This chapter starts by providing background on segmentation techniques. Segmentation results in the identification of objects within the image upon which the above mentioned manipulation can be applied. This chapter subsequently covers background related to inpainting which is the process of removing specific regions of an image in the least noticeable manner.

3.2 Segmentation Techniques

Image segmentation is the process of separating objects in an image into multiple regions or partitions [73]. Segmentation can be achieved through a variety of techniques and regions are partitioned depending on different attributes such as colour, intensity or other features [74]. The main motivation behind this process is to be able to perform smoother and more efficient analysis of the image [75]. Segmentation is employed within different techniques presented in this thesis, such as for example the distinction between foreground and background in an image. This section surveys a selection of segmentation techniques such that a context for the chosen techniques is given accordingly.

Traditional segmentation techniques can be grouped in four main categories, namely: edge-based, fuzzy, thresholding, and region-based segmentation [75]. There is also a selection of deep learning techniques that segment an image by objects belonging to specific classes upon which the deep learning model is trained.

Edge detection techniques such as Canny or Sobel detection can be used to identify the edges in an image and act as basis for segmentation [75]. This approach requires significant pre-processing for the properties of edges to be fully revealed. Moreover, the discovery of edges needs to be then complemented by contour detection. The main advantage of this approach is its fast performance, although it performs weakly when processing images with higher noise [75].

This section presents the other techniques that are more relevant to this thesis.

3.2.1 Levels of Segmentation

When dissecting a scene into partitions, segmentation can also be seen through different perspectives. This subsection explores these perspectives, particularly in the context of the intra-object segmentation approach presented in Section 4.4 within this thesis.

Foreground Segmentation

Objects in the foreground are generally of highest interest and are therefore targeted for segmentation. When depth information is available, foreground extraction is carried out first by processing the depth information, followed by image thresholding to reduce the grayscale image into a binary image for faster processing [73] as presented in Section 3.2.2. Another approach is by applying a graph-cut algorithm such as GrabCut [3] as explored in more detail in Section 3.2.3. Level Sets are also used in segmentation to exploit the identified contours in an image. Early methods [76] used a variational framework on active contours to enable segmentation of foreground objects. Level set segmentation was later improved with consideration of probabilistic formulation for 3D segmentation [77].

Depth information can also be used for foreground object segmentation to extract regions [78]. This approach distinguishes objects in foreground with respect to the texture and depth and also their trajectory in subsequent video frames [78]. The scenario presented in this thesis uses a single image/frame which makes the approach more challenging since there is no information from neighbouring frames.

Object Level Segmentation

Object-level segmentation identifies regions within the image that correspond to an object belonging to a class or label. These objects are then labelled semantically [79]. Deep neural networks, particularly CNNs, are the most popular approaches used to generate this type of segmentation [80] [81]. These models are trained using a labeled dataset that defines different objects in an image. The model will then be able to segment a given image into objects. The accuracy depends heavily on the type of model used together with the hyperparameters of the network.

Part Segmentation

Part Segmentation goes a level deeper than object level segmentation. Such models identify parts of an object and the relationship between the parts belonging to the object [79]. CNNs are also the most common approach for this type of segmentation [82]. On the other hand, [82] demonstrates unsupervised learning can also be used to tackle this problem. Markov random fields (MRF) [83] are also a method used to generate part-segmentation. The example presented in [84] uses the clique/neighbour approach in graph theory to part label the pixels in the image. Pixel proximity is used to guide this approach and if a pixel label is only related to the label value in its adjacent location, the set of the labels in this set of grid points is considered as a Markov random field.

Co-Segmentation

While the previous methods handle segmentation in one image at a time, co-segmentation results in the segmentation of an object that is found in two or more images [85]. Originally proposed by [86], co-segmentation generates simultaneous segmentation proposal in a set of images. This uses a generative model based on a MRF and computes the similarity between segmented objects in the respective images.

Semantic and Instance Segmentation

Semantic segmentation involves pixel-level extraction of objects belonging to the same class or type. Due to its effectiveness, semantic segmentation has been applied to a variety of fields of study [87] [88]. Instance segmentation address the challenge left behind by semantic segmentation. This is organised into two parts. The first part is similar to semantic segmentation where objects belonging to specific classes are detected. The second part differentiates between different objects that belong to the same class. Unlike semantic segmentation, in instance segmentation each object is extracted separately. Instance segmentation has also been applied to different research areas [89] [90] [91] and these typically include tasks that need to distinguish between objects belonging to the same class.

3.2.2 Segmentation by Thresholding

The main motivation in the use of this approach is to analyse pixel values with respect to a pre-defined threshold. In the processing procedure of the depth information, image thresholding was carried out to reduce the grayscale image into a binary image for faster processing. The white pixels on the binary image represent the foreground. Otsu's method assumes that the image contains two classes of pixels following a bi-modal histogram [92]. This results in the separation of foreground pixels and background pixels without the need of carrying out more complex operations for segmentation. The main challenge with this approach is the process of finding the most optimal threshold that can vary from image to image [75].

3.2.3 Graph-Cut Segmentation

Graph-cuts, proposed by Boykov and Jolly [93], are based on graph theory with a variety of applications in computer vision. In the context of this thesis, they are used for image segmentation and the separation between foreground and background in an image. In this approach, every pixel in the image is represented by a node in a graph, assuming a 2D image [93].

Two separate nodes are added; the Source Node that in this context will represent the foreground and the Sink Node that in this context will represent the background [93]. The source and sink nodes are connected to all other nodes in the graph and the weight of the edges will represent the probability of whether the node is either foreground or background. The pixels in the graph are also connected to their immediate neighbours, either 4-neighbourhood or 8-neighbourhood. The different choice of neighbourhood resolution reflects the level of detail in the interconnecting weights. This inter-neighbourhood connectivity assists in evaluating the similarity between pixels in a neighbourhood with their classification being also assisted by this difference in weight. Once the weights within the neighbourhoods are computed based on a measure of similarity of choice, a minimal-cut is computed.

The minimum cut will traverse the edges and reject the edges with a low weight as background, thus separating them from the foreground. This minimal-cut returns the global minimal energy in the graph and can be achieved in polynomial time when using algorithms such as the Stoer-Wagner algorithm [94].

The GrabCut foreground extraction technique [3] was designed to be an interactive approach where the user marks selected regions of the image as foreground, assuming that the rest is background. The GrabCut algorithm combines graph-cut techniques with statistical models to estimate the weight between background and foreground.

The initial step of the GrabCut algorithm is when the user selects the object of interest through a rectangular bounding box. The pixels inside the bounding box are more likely to be foreground while the rest of the image is more likely to be background. Grabcut introduces an improved modelling of the background as an extension of the original graph-cut segmentation discussed above. The bounding box data is used to create K components of a multivariate Gaussian Mixture Models (GMM). A total of two GMMs are generated, K variables for the foreground and another K variables for the background. Techniques such as K-means clustering can be then employed to compute the variance between each pixel. This is used to enhance the inter-neighbourhood similarity between the nodes in the graph. Pixels are then either assigned to the background GMM or foreground GMM respectively. This process of assigning pixels to GMMs is carried out iteratively as the GMMs learn what is background and foreground through a graph-cut approach. This process is terminated when the classification of background and foreground converges [3].

The GrabCut technique was subsequently modified to make use of depth information when available [95]. In this improvement, depth values were used as the fourth channel that would infer the opacity α of pixels on the case whether they are labelled as foreground or background.

3.2.4 Deep Learning Approaches

The general interest in Deep Learning came after the results of the AlexNet model [13] in the ImageNet image classification competition of 2012. This successful implementation of a CNN brought stronger interest in deep learning and its application in object detection and classification. The first generation of such networks focused on correctly localising an object in an image and this was denoted by a bounding box around the object. These techniques made use of a one-stage detector where proposals were made on a number of predictions on a grid.

The Region-based CNN (R-CNN) [96] introduced the notion of a two-stage detector. This model was built by first extracting potential regions from the image using a selective search technique [96]. The selective search is a heuristic based search that assumes that similar pixels belong to the same image. These regions of similar pixels are then merged and bounded by padding. The convolution features are then extracted and eventually fed into fully-convolutional layers that result in the classification. The main limitation of this model was that, due to selective search, most CNN proposals were not really objects and the network could not detect negative classes [96]. The major drawback of the R-CNN was its speed. Its successor was adequately named Fast R-CNN [97] that reduced multiple convolutions on region proposals. The Fast R-CNN model processes the entire image through the CNN instead of selected parts. However, it introduced the idea of generating a feature map upon which selective search is applied. This rendered the Fast R-CNN faster since it did not need to feed around 2000 proposals to the network. The newly added feature proposal eventually resulted in being a bottleneck for the Fast R-CNN and motivated the introduction of its successor named Faster R-CNN [98]. The Faster R-CNN eliminated the selective search and the network learnt from the region proposals through a Region Proposal Network (RPN).

As deep learning models that were dealing with object detection improved, their application to pixel-level Image Segmentation was the next natural step. The Mask R-CNN [14] segmentation model published in 2017 is one of the most popular and successful approaches for instance segmentation. The Mask R-CNN is an extension of the Faster R-CNN[98], Fast R-CNN[97] and R-CNN[96]. The main useful and novel aspect of Mask R-CNN was the introduction of a new branch that predicts pixel-level segmentation masks on each region of interest. While Faster R-CNN has two outputs (bounding box and its class label), the Mask R-CNN has three (bounding box, class label and an object mask).

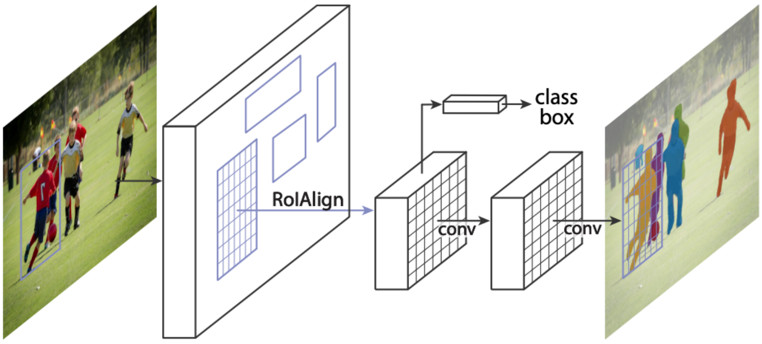


Figure 3.1: High-level framework of the Mask R-CNN model (Source: [14])

The Mask R-CNN is a two stage object detection and segmentation framework. The first stage analysis the input image and generates a set of area proposals where each area would have its own likelyhood of containing an object. The second stage is composed of a classification process of the proposal that ends in generating bounding boxes and masks.

The first component of the Mask R-CNN is the head-architecture that passes the image through a CNN that extracts its feature maps. This makes use of a Feature Pyramid Network (FPN), originally proposed by Lin *et al.* [99], that propagates high level features to lower layers and every layer would have access to the lateral higher and lower level features. Within Mask R-CNN this is achieved through a ResNet [72] and ResNeXt [100] networks with a depth of 50 or 101 layers together with an FPN as backbone. Using the Region Proposal Network (RPN) introduced in the Faster R-CNN, multiple ROIs are proposed based on the object proposals resulting from the FPN. This is accompanied by a lightweight binary classifier that predicts whether the ROI includes an object or not.

The Mask R-CNN also proposes the ROIalign layer that aligns the extracted features with the input image [14]. These proposals are then refined after being passed through a number of fully-connected layers that return the bounding box and class prediction through a regression model.

The last part of the Mask R-CNN deals with the generation of pixel-level segmentation masks. This are generated through an additional network branch that generates a mask for each of the objects detected by the RoI classifier. This branch includes a stack of consecutive convolutional layers that output the mask.

Mask R-CNN is given focused attention in this thesis particularly because of its use in the current state of the art in saliency ranking. To date, it is also generally considered as the best instance segmentation method. There are other noteworthy methods such as the Fully-Connected Network (FCN) [101] which is an extension of the CNN and DeepLab [102] semantic segmentation network that encodes multi-scale contextual information. Segmentation is also widely used in medical applications with specialised networks such as U-Net [103] and U-Net++ [104] demonstrating significant efficacy in medical tasks such as tumour segmentation [105] [106].

3.3 Inpainting Techniques

Inpainting is the process of removing a region in an image and replacing the removed region with texture that renders the end result as undetectable as possible [107]. This is sometimes referred to as image completion. The area covering the removed object is filled with texture that surrounded the original region. Another approach is the use of generative methods where the image is filled through techniques such as GANs that are explored in Section 3.3.2. The ever-improving results of inpainting methods renders this technique very popular in most packages. Different applications range from image restoration, visual editing to object removal [108] [109].

3.3.1 Traditional Approaches

One of the first successful inpainting methods was proposed by Bertalmio *et al.* [110]. This approach diffuses texture from around the target region into the missing area. This is achieved by making use of a variational method and partial differential equation (PDE). This was inspired by the fluid-dynamics principles in physics. Bertalmio *et al.* [110] designed their approach on the similarity of the problem of image intensity's relevance in inpainting to the the stream function in a 2D incompressible fluid. The stream Ψ is parallel to the Image Intensity I. The fluid velocity $v = \nabla^{\perp}\Psi$ is the foundation for the isophote direction $\nabla^{\perp}I$ where $\nabla^{\perp}$ is the perpendicular gradient [110].

In fluid dynamics, the principle of Vorticity is a pseudo-vector that models the tendency of something to rotate in a continuum around a point [111]. The stretching of vorticity due to flow compressability is modelled using the Navier-Stokes equation for continuity. The vorticity $\omega = \Delta\Psi$ is used to model the inpainting smoothness $w = \Delta I$ where Δ is the Laplace operator [110]. This analogy allows for a gentle continuity in the image intensity and its isophote directions across the boundary of the target region [110] resulting in an effective and fast method to approximate inpainting.

Chan *et al.* [112] applied a similar method that uses a PDE and the Euler-Lagrange equation to propagate the diffusion inwards towards the target. The change of information happens perpendicular to the edge of the mask towards the centre of the region to be inpainted [22]. The information used for this diffusion is based on lines of equal grey values, known as isophotes [109]. The information from extracted isophotes enable such techniques to efficiently preserve any structural information in the missing region.

Telea [113] eventually proposed an improved method that uses fast marching. This method is an improvment on the PDE approaches [109] since it addresses the computational overheads related to the propagation are removed. The Fast-Marching Method (FMM) proposed in Telea's method removed the need for the

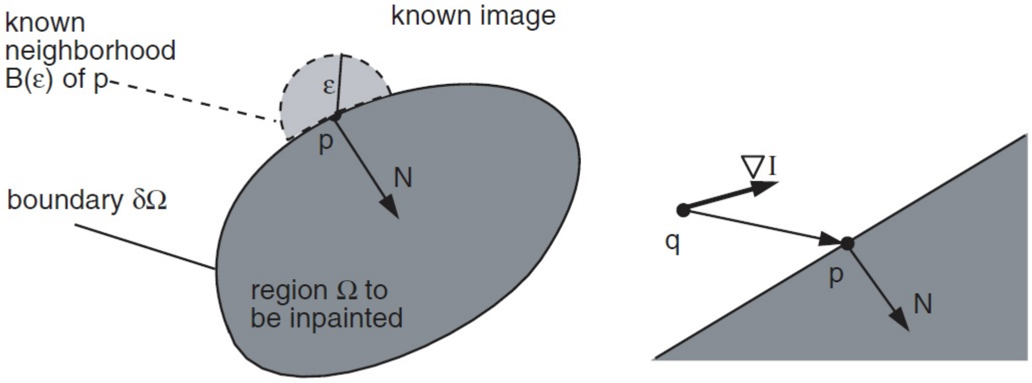


Figure 3.2: The PDE-based inpainting principle as illustrated by Telea [113]. The figure on the left presents the general inpainting problem and the figure on the right presents the propagation process.

numerically unstable and complex methods like diffusion in Bertalmio *et al.* fluid-dynamics based method [113]. The FMM produces a number of points between the known image and the region to be inpainted as illustrated in Figure 3.2. A set of distance maps are computed and inpainting is calculated based on these distances. Such methods inpaint the target region with content from the known image depending on the distance between q and p [113]. An advantage of the FMM is that the *narrow band* is maintained and this allows for a clear separation of the pixels in the known image and those in the target image along the boundary $\delta\Omega$ [113].

The FMM algorithm proposed by Telea maintains the *narrow band* along the $\delta\Omega$ where a set of values is stored for every pixel along this line. These include the pixel value T, its grey value I and a flag f. Flag f can have three values:

1. BAND: denoting pixels falling on $\delta\Omega$, requiring T to update;
2. KNOWN: denoting pixels falling in the known image where both T and I would be known;
3. INSIDE: denoting pixels within the target region where T and I would be unknown.

The FMM is initalised by setting T to zero on $\delta\Omega$ and on the known image. On the other hand, T is set to a large value inside the target region [113]. The flag f is initialised for the entire image with the values described above. The propagation is set over four steps. The first step extracts the smallest number of T. The second step starts the marching process by moving inwards by adding new points to the boundary. The third step carries out the inpainting and propagates the values of T to its neighbours [113].

This method gives a result faster. However, its limitation is based on the blurry effect that it leaves after the inpainting [109] [113]. The same effect was also obeserved in the results of inpainting experiments presented in this thesis.

The idea of using patches to tackle inpainting emerges from the contribution of Efros and Leung [114]. Their method is based on an approach that recursively fills an empty region inwards from the boundary [115]. The neighbourhood of a pixel p is considered and similar pixels are selected using the sum of squared differences. The main limitation of this method is the lower quality of the output when larger areas are inpainted [115].

Criminisi *et al.* [116] later built on the previous and introduce two main improvements. The first one is based on the filling order. The was changed from the original onion-peel to a priority scheme. Secondly, the entire patch was being copied instead of taking single pixels [22] .

This approach, illustrated in Figure 3.3, is based on filling a target region Ω with information from a source region Φ [116] such that the target region Ω does not stand out among its surroundings once filled [115]. In the context of monoscopic inpainting, Ω is treated as a hard constant. On the other hand, in stereoscopic inpainting other information is available from the other view, therefore reducing the strict boundary $\delta\Omega$. The idea of the importance of structure was further enhanced by Sun *et al.* [117] when they proposed an improved approach by considering the entire context of the image rather than the neighbourhood of the target region. The main subsequent developments are based on these approaches and address the

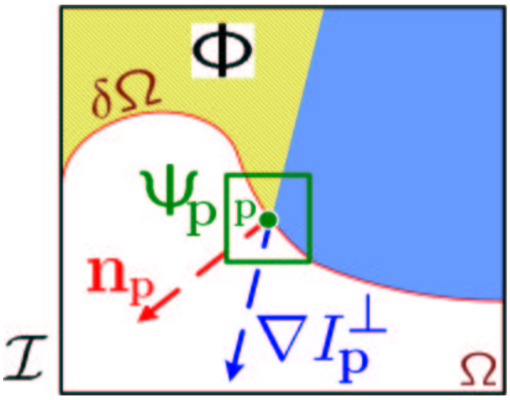


Figure 3.3: An illustration showing the notation and approach used by Criminisi *et al.* [116]

issues of performance and quality of the inpainted regions [108] [109]. The main weakness of these approaches remains the lack of semantic knowledge or context of the target region. For example, if the target region is in the middle of a human face covering entirely the nose, the inpainting approaches surveyed in this section would not be able to reconstruct the nose. This problem is solved with deep learning approaches surveyed in Section 3.3.2.

3.3.2 Deep Learning Approaches

The restriction on semantic knowledge of the domain inherent in traditional approaches is also their limitation. This was witnessed first hand in our earlier work published in [22]. The results of this work demonstrate the same limitations since it was published before the deep learning techniques analysed in this section. This limitation is directly addressed by deep learning approaches that use models trained over a number of datasets. Deep learning models related to inpainting can be organised into two. These are namely CNNs or GANs. These models are surveyed below.

CNN approaches start by first filling the target region with placeholder values or noise. These regions are then fed as input to pre-trained layers of the model that contain low level or medium level features [118]. This approach would first use content encoders [119] that feed the target region and then decode the feature space [23]. This fills the target region with upsampled regions and the result is of a relatively low quality. Yang *et al.* [120] proposed an approach that uses the output of the content encoder as input propagates texture information onto it from its trained model. The main limitations of these techniques are generally related a lower quality output with diffusion of colour or blurriness as witnessed in the traditional techniques [23].

NVIDIA's Liu *et al.* [118] proposed a method that addresses the limitations of CNN based techniques. This novel method uses cost functions that are commonly found in artistic style transfer in neural models as originally proposed by Gatys *et al.* Since this method balances between content and style, it uses a cost function for each. The pre-trained model iteratively assess the quality of the output, hence inpainting. It starts from a low quality image proposal being given as input and assessed by the cost functions. These functions compare the quality of the inpainted result until a desired level is met allowing the network to converge. The results of this paper state that a result is achieved in 0.029s. It is important to note that this result is achieved when running the model on a NVIDIA V100 GPU [118]. The original paper also explains the considerable effort in training required for the model training. It is reported [118] that the model was trained on 55,116 masks and tested on 24,866. The image size used was 512 × 512 and this was also a limitation in itself. A NVIDIA V100 GPU (16GB) with a batch size of 6 was used for training and this process took 3 days to train on CelebA-HQ while the training on Places2 and ImageNet took 10 days [118].

Generative Adversarial Networks Approach

Generative Adversarial Networks (GANs) are a framework of two neural networks. One of the networks generates new content while the other one discriminates the result of the former for the improvement of results [12]. GANs involve the training of two separate models, the Generator G and the Discriminator D. Initialised by random noise z, G evolves the meaningfulness of the content, where in this case it is an image. For every single output x from G, the discriminator D gives an output score in the form of probability of whether $\boldsymbol{x}$ is a real artefact or a fake one from the output of G. The GAN framework aims to optimise $V_{\text{GAN}}(D,\ G)$ as a two-player minimax zero-sum game between the two networks [12] as denoted in Equation 3.1.

$$\begin{aligned}\min_{G}\ \max_{D} V_{\text{GAN}}(D,\ G) = {} & \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] \\ & + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].\end{aligned} \tag{3.1}$$

In practice, GANs are composed of a generative network approximating content in a way that the adversarial discriminative network is not able to distinguish between the content generated by the generative network and real content. Various studies [121] [122] investigated the different applications and evolution of GANs since the original proposal [12]. In computer vision, GANs have been used for a variety of applications such as super-resolution [123] [124], face image synthesis [125] [126], text-to-images [127] [128] together with other image editing applications.

GANs were initially introduced in 2016 and in 2017 they were applied to the inpainting problem [122] [129] [130]. I has been demonstrated that GANs can be used to solve the inpainting problem by making use of a semantic context [130]. Our inpainting approach [22], was published in 2016 before the adaptation of GANs for inpainting. In this work, the feedback received during the user interviews where subjects remarked about the importance of the preservation of visual structure when inpainting images. Further progress in the application of GANs [131] [132] [133] extends the importance of context to result in a sharper inpainted region that blends with the surroundings of the inpainting region.

The quality of inpainting results when using GANs is considerably high however, this approach still faces certain difficulties. Its natural dependency on training data is also its main limitation. This is particularly felt on the training of the generator model [133]. The adversarial loss convergence can also be a challenge with specific training datasets [130]. The dependency on datasets where they include contexts that are not clearly featured in the training set result in the GAN inpainting method to be restricted to a selection of domains or situations. The training time of both networks in a GAN also provides potential challenges [132]. The choice between GANs or traditional approaches for inpainting depend on the balance that needs to be struck between quality and acquisition of results [23].

3.3.3 Saliency Driven Approaches

Due to their relevance to this thesis, a selection of inpainting approaches that make use of saliency maps and salient information of images are reviewed in this section. The approach proposed by Li *et al.* [134] takes advantage of the saliency map information when carrying out an inpainting operation on an image. This approach therefore considers the values of the saliency map together with colour and structure information and computes a priority score that guides the patch size of the patch-matching based inpainting algorithm. Mechrez *et al.* [6] recently proposed an improved method based on the same idea, this time within the context of an image manipulation framework. This method introduced a level of control to the user and also covered substantial evaluation over different datasets. The main limitations of these techniques is the lack of semantic information when carrying out such image manipulation.

Saliency maps can also used for the evaluation of inpainting algorithms [19]. The idea behind such an approach is to ensure that the visually important regions of an image are given due importance when evaluating inpainting. The main limitation of this approach is its computational cost due to patch similarity computation. Moreover, this technique still deserves further investigation on larger datasets. The saliency ranking approach presented in this thesis does not require such an evaluation approach since the regions being inpainted will be the same regions that scored a high saliency score.

3.4 Conclusion

This chapter surveyed a selection of traditional techniques that enable image manipulation. This focused on segmentation as the stage for the generation of the mask of the target object and image inpainting as a manipulation tool for removing the target object from an image.

The first part of this chapter explored different segmentation techniques. These methods were based on pixel-level information that varied from thresholding pixel values to graph-cut approaches. As deep learning approaches improved object detection in images, segmentation was also benefiting from this progress. With the development of current state of the art techniques, such as the Mask R-CNN [14], pixel-level segmentation of objects belonging to classes known to the model became possible and efficient. These models allow for pixel-level segmentation to be used as demonstrated in this thesis.

Inpainting also went through a similar path of progress. With the advent of GANs in late 2016, computer vision was introduced to the new era of generative content approaches. One of these approaches was inpainting. In this case, both traditional and deep-learning approaches still have their own relevance and distinct limitations. The challenges in the evaluation of such methods also motivated us to design and build the COTS dataset [10] presented in Chapter 6 that facilitates the evaluation of inpainting techniques as we also demonstrated in [23].

4. Preliminary Experiments

> There are only two mistakes one can make along the road to truth; not going all the way, and not starting.
>
> Buddha

4.1 Introduction

In the early stages of this thesis, we were exploring efficient ways of selecting objects or part of an image. This earlier work served as a motivation for research from which saliency ranking emerged. These preliminary experiments resulted in two effective methods for selecting an object or part of it. The first one was a single click object detection [24] and the second one used depth information to break down an object in smaller components resulting in intra-object segmentation [21].

This chapter gives an account of these two experiments. The data chosen for these experiments was RGB-D video sequences from which single frames were chosen for processing. This enables future work to extend these techniques to video content.

4.2 Single-Click Object Selection

This technique was a user driven approach that enabled users to select parts of the image through a single click. This method exploited colour and depth information in RGB-D datasets to extract the object selected by the user. It involved the user clicking on an object in the input image. The single click selection technique is initialised by receiving the colour and depth information at the source click coordinate (x_s, y_s) as input. The region around the source click is stored for reference and denoted as the reference patch Ψ_{ref}. In this technique, a patch Ψ is defined as the region of pixels $k \times k$ having the click coordinate as its centre. The width and height of Ψ are manually configurable depending on the set value of k.

When the user clicks a target object, the pixel values in the reference patches Ψ_{refC} and Ψ_{refD} for colour and depth respectively are stored. A median filter is used to reduce variations in pixel differences while still maintaining the key features of the image. The median filter was preferred over the mean filter since outlying pixels do not effect the median value and therefore preserves a level of detail that can be used in this technique. This technique proceeds by traversing the image by striding a query patch Ψ_i looking for other similar patches to Ψ_{ref}. The sum of squared differences is computed using threshold θ_C and θ_D to identify patches that are similar to the reference patch. Experiments were carried out to fine-tune the threshold parameters θ_C and θ_D for the algorithm such that they return the best results over different scenes. The limit N relates to the coverage of the reference patch as it acts as a sliding window on the image, depending on the stride.

$$SSD = \sum_{i=1}^{N} (\Psi_{ref} - \Psi_i)^2 \tag{4.1}$$

Ψ_{refC} and Ψ_i are then compared by computing the sum of squared differences as shown in Equation 4.1. This computation allows for the traversal of Ψ_i with intervals equal to the width and height of the patch along colour and depth images [24].

The result of $SSD(\Psi_{refC}, \Psi_{iC})$ and $SSD(\Psi_{refD}, \Psi_{iD})$ and then compared to threshold values θ_C and θ_D respectively. The result of SSD indicates the difference between the two patches being compared. Therefore, the pixel coordinates that generated patches with ssd_C and ssd_D results which are less than θ_C and θ_D are stored in a list of colour patches L_C and and list of depth patches L_D respectively.

The only parameter that affects the performance of this technique is the size Ψ together with the size of the input image. On the other hand, the threshold parameters affect the quality of the resultant matching coordinate sets. The smaller the size of Ψ the more computationally expensive the process is. Moreover, smaller values may also result in less distinguishable features that would result in a higher number of true positives. On the other hand, a larger size of Ψ results in bigger patches being processed with significantly varied features. This results in a smaller number of matches that render the technique ineffective.

4.3 Results of Single-Click Object Selection

The single click selection technique was evaluated using Microsoft's *Ballet* sequence [135], Nayoga's *Balloons* sequence [136] and Berkeley's *3-D Objects* dataset [137]. These datasets were used to visually demonstrate the results obtained from this technique.

Figure 4.1 shows an example of use of this technique. The red dots represent the coordinates in L_D while the small blue dots represent the coordinates in L_T. The blue dots with a red border represent the coordinates in the matched coordinates list, hence coordinates present in both L_D and L_T. The green dot indicates the coordinate pair (x_s, y_s).

The *Ballet* is a nicely balanced scene with multiple objects and characteristics that became an interesting case for the evaluation of this technique. Figure 4.2 presents the result of the technique when the user clicks on three distinguishable objects. These objects had similar colour but different depth values.

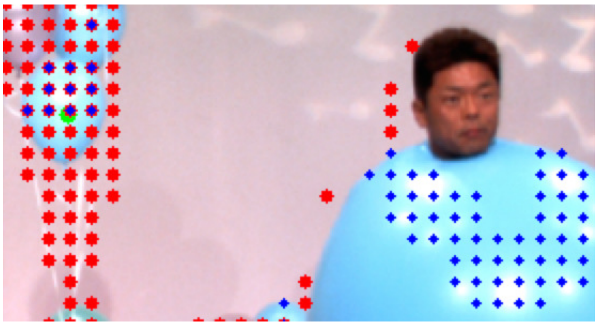


Figure 4.1: Demonstration of the single click object selection output. The red dots represent coordinates in L_D, the blue dots represent the coordinates in L_T and the blue/red points indicate a match in both. In this example, the balloons on the left hand side were selected and point x_s, y_s is marked with a green dot.

The *3D-Object* scene was used to demonstrate further results of this technique. These results are shown in Figure 4.3 and demonstrate situations where the user clicks on objects that are represented by similar colour and depth information.

The main limitations of the single click object selection technique are exposed when the user clicks on larger objects in the scene. Such objects would contain variations in colour values and also variations in depth. Failure cases are presented in Figure 4.4. The top two illustrations in this figure show how two runs of this technique have to be carried out in order to cover the entire background wall of the *Balloons* scene. The same behaviour is also demonstrated in the bottom left illustration.

On the other hand, the background wall in the bottom right illustration was entirely covered since there was a match in texture and depth. Nevertheless, part of the adjacent wall also resulted in a match. In such a case, this technique can be used to reduce the search space for other techniques mentioned in [138] and [139], such as for example Random Hough Transforms for finding dominant lines in the scene that may provide homographic information for better building of the plane. [139] also dedicates an entire category of a classifier for perspective training and this proposed technique can act as a discriminatory tool on the entire image so that such classifiers can be more efficiently trained [24].

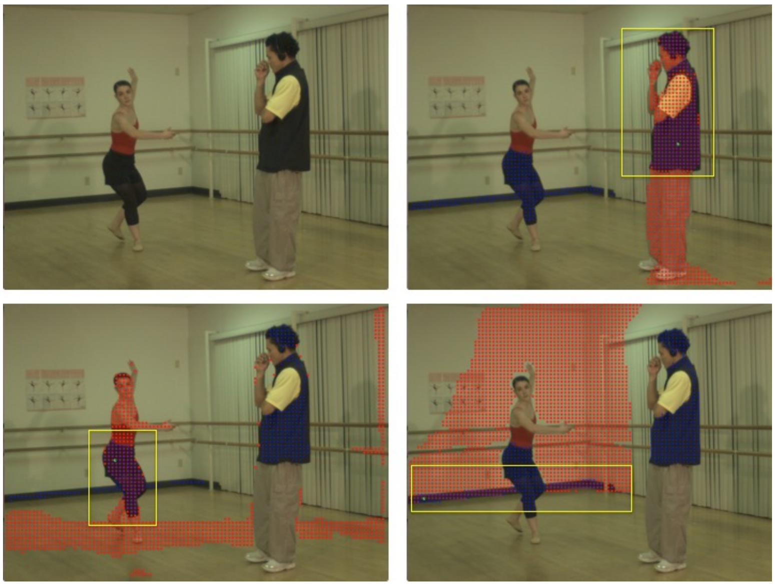


Figure 4.2: Result of the proposed technique. Top left: Original image. Top Right: Man's jacket selected. Bottom Left: Ballerina skirt selected. Bottom Right: Floor skirting selected.

4.3.1 Algorithmic Performance

The computational complexity of the single click object selection technique depends on the dimensions of the image and the size of the patch Ψ. The complexity can therefore be represented as $\mathcal{O}(n)$ where n denotes the number of pixels in the image. This is due to the reason that although I is traversed in intervals of dimensions of Ψ, the SSD computation would still require all the image pixels to be visited [24].

A set of performance experiments were carried out on a machine with a 2.6 GHz Intel Core i5 processor and 8 GB 1600 MHz DDR3 memory running a MacOSX El Capitan v10.11.2 operating system. On average, the program returned a result in 0.048 seconds. The worst time was recorded when a large object (over 85K pixels) is selected and this was 0.074 seconds. On the other hand, when the sub-object was smaller (less than 10K pixels) the program returned a result in 0.012 seconds [24].

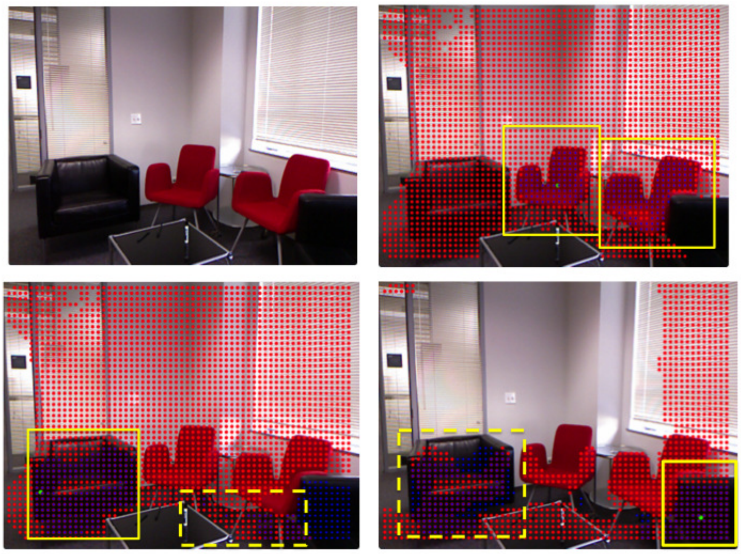


Figure 4.3: Result of the proposed technique. Top left: Original image. Top Right: Red chair selected. Bottom Left: Black armchair selected with incorrect matches in the dashed box. Bottom Right: The other black armchair selected with incorrect matches in the dashed box.

4.4 Intra-Object Segmentation

In the process of exploring inpainting and blending manipulation techniques on RGB-D images we realised that plenty of opportunities were being missed since most image manipulation techniques assume a single colour image with no depth information. Moreover, emerging technology, such as augmented reality, also provide a demand for accurate selection of objects in 3D. Most techniques handle a scene in 2D and it often results in poorer user experience.

This section presents a technique that takes advantage of the colour and depth information in order to select and dissect an object within a scene. This results in further segmentation of the selected object, referred to as 'intra-object segmentation', based on the depth information [21]. The output of intra-object segmentation is an object that is split in layers which facilitates the handling of the object in different manipulation approaches. The intra-object segmented output can be used as input to blending methods, such as inpainting, where these can be applied on each layer or just a selection of layers separately, allowing more control on the

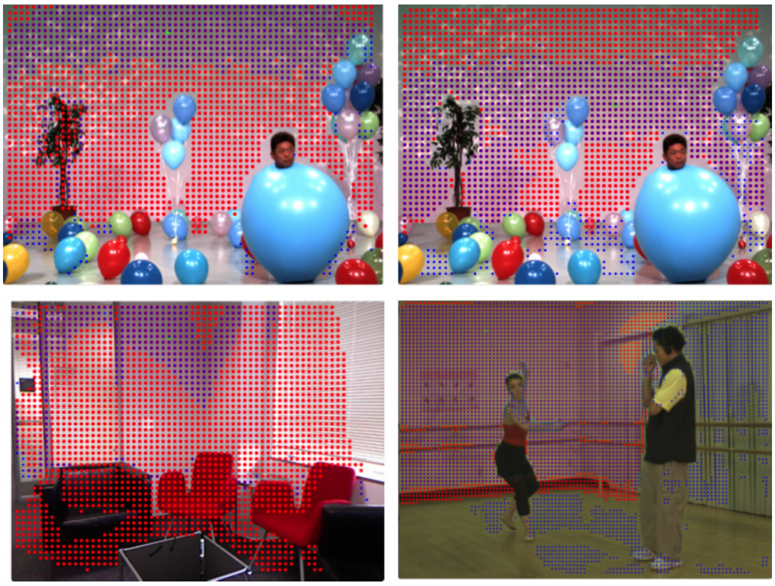


Figure 4.4: Partial limitations of the proposed technique. Top left: Partial background wall matched in the *Balloons* scene. Top Right: Another region of the wall matched in the *Balloons* scene. Bottom Left: Partial wall in the *3D Objects* scene. Bottom Right: Entire wall match but with over-spilling on second wall in the *Ballet* scene.

process.

This technique was also designed to perform efficiently within a context that would need to assign resources to other demanding processes. Other related work, such as [82] [79], contributing to object level segmentation or part segmentation depend on deep feed-forward networks that require substantial training [21]. On the other hand, intra-object segmentation is an efficient approach to achieve object level segmentation without the need of training data and only using RGB-D information.

4.4.1 Intra-Object Segmentation Methodology

For simplification and focus on this technique, this section assumes a bounding-box selection of the target object. The output of the Target Mask Generator can also be used to identify the region in which this technique is carried out. The approach presented in this section and published in [21] uses Grabcut [3] together with enhanced depth information. The depth information is optimised by processing its histogram and the optimised result is then used to partition the colour pixels with respect to the depth values.

Following the initialisation of this technique based on the selection of the region of interest, high level object segmentation is carried out. The output of the first phase is a single-channel binary image that will act as a guiding mask in the second phase. The first phase of this technique is visually illustrated in Figure 4.5.

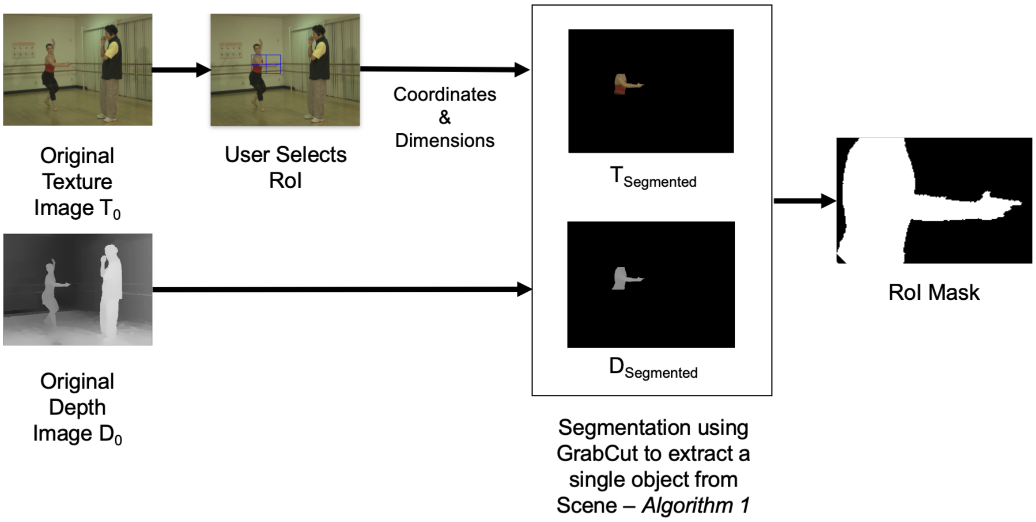

Original Texture Image T0
User Selects RoI
Coordinates & Dimensions
TSegmented
DSegmented
RoI Mask
Original Depth Image D0
Segmentation using GrabCut to extract a single object from Scene – Algorithm 1

Figure 4.5: Selection and initialisation. This technique is initialised by identifying a specific region of interest. The initial segmentation that extracts the object from the rest of the scene is based on this selection. The resultant mask is then processed for intra-object segmentation. This example is based on a frame from the *Ballet* sequence.

The process is initialised by receiving the RoI together with the original texture and depth images, T_0 and D_0 respectively, as parameters. This phase then builds the output mask μ by running the Grabcut [3] algorithm on D_0 and then carrying out a bitwise AND operation on μ and D_0.

The subsequent part of this technique processes the histogram H of the image it receives as a parameter. All the bins of the histogram are traversed, filtering out the non-zero bins that are then stored in the list $shadesNZ$. The ordered list spans

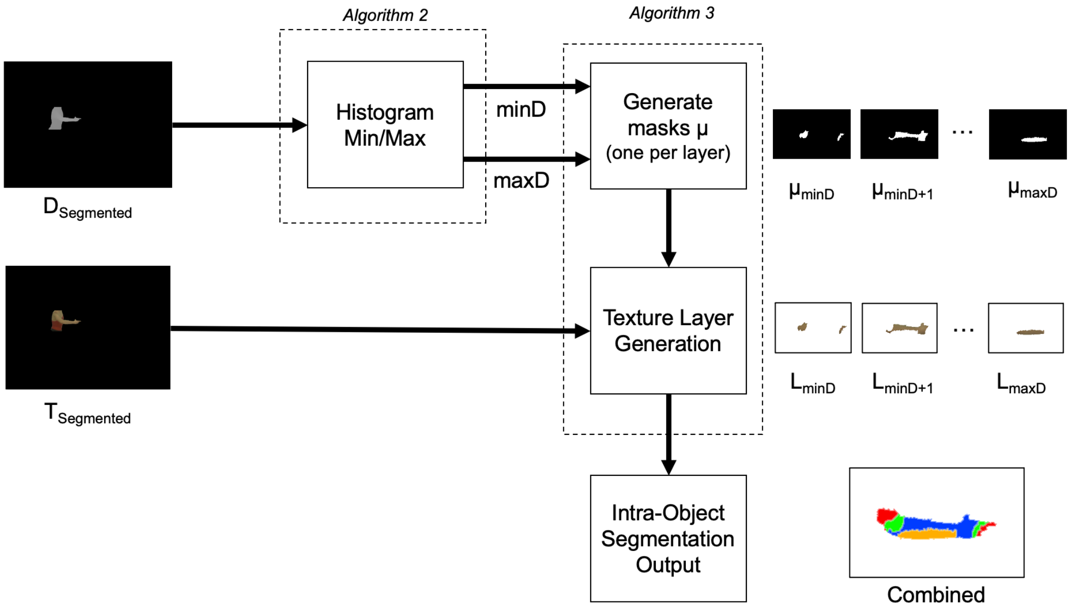


Figure 4.6: Intra-object segmentation process. The second part of the solution makes use of the segmented texture and depth images presented in Figure 4.5. The histogram of the segmented depth map is processed and subsequently used to generate the masks that are used to extract the segmented layers. In this example, we are demonstrating the procedure on the *Ballet* sequence.

from the minimum shade at its start to the maximum shade as the last element. Every shade in the list corresponds to a depth value. The minimum value $minD$ represents the layer that is farthest away in the scene while $maxD$ the closest layer [21].

Meanwhile, the $minD$ value is then used as a threshold value while μ is traversed. During this traversal, the pixels that are greater than $minD$ are set to white (255) or set to black (0) if less. Two other bitwise AND operations are used on the colour and depth map such that their corresponding segmented images are generated.

The second component, $histMinMax$, processes the output depth map $D_{segmented}$ and returns a normalised histogram. This facilitates the process of mapping the depth values to the z-values of the scene, where z represents the corresponding depth value. In this context, $minD$ and $maxD$ provide the range of z-values of the selected object.

The first phase dealt with the depth-assisted segmentation of the objected in the user selection. This is then followed by the last stage that deals with the extraction of layers. The segmented colour and depth images are used as a source to generate the laters and the process is guided byt the contents of the list of shades corresponding to the depth map in the range between $minD$ and $maxD$. This process traverses every shade and extracts every non-zero shade is stored in a separate list. A new mask is generated for every shade in the list. Each mask is built by traversing every pixel of $D_{segmented}$ and the pixels corresponding to the current depth value i are set to white (255) while the others are set to black (0) [21].

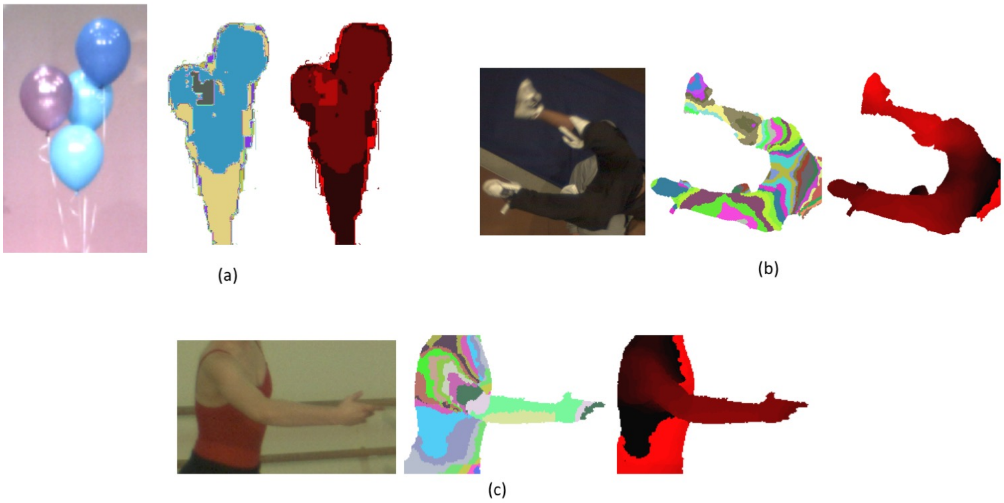


Figure 4.7: An illustration of the actual image, result of intra-object segmentation and intra-object segmentation with respect to depth for the the *Balloons* sequence (a), *Breakdance* sequence (b) and *Ballet* sequence (c).

The following step in the process takes every mask in the list relating to a level of depth and applies it on $T_{segmented}$. When the mask originating from a depth level is applied to the colour image, it extracts the colour pixels that correspond to its respective depth value. This results in list L_{layers} that contains different segments of texture for each level of depth. The output of the third part results in the data structure L_{layers}. This stores all object data in the form $[z, (x, y, (R, G, B))]$. Its design allows for a neat way of organising the image colour pixels by depth. This paves the way for the potential of more immersive content. The output is presented in Figure 4.7 and a further analysis is given in Section 4.5.

4.5 Results of Intra-Object Segmentation

4.5.1 Intra-Object Segmentation Output

The *Breakdance* and *Ballet* sequence [135] together with Nagoya University's *Balloons* sequence [136] were used to study the performance of this technique. Furthermore, they also have varying quality in their depth map and therefore providing an opportunity to explore the limitations of this solution.

This section starts by exploring the limitation of the developed intra-object segmentation technique based on the quality of the depth map. This is done by a visual comparison of the output of the technique on different datasets.

The *Balloons* sequence was chosen to evaluate how the technique performs on a low quality depth map. Frame 291 of this sequence was selected and the process of choosing the RoI and the resultant segmented layers is presented in Figure 4.8. There are 6 layers of depth between the layer containing the strings and the one having the balloons as segmented by the system. This number is based on the histogram values of the depth map. Such a result can also be interpreted from a physical perspective since the balloons would take more physical space in the z-dimension. However, the result also shows that the majority of the balloons area was captured in a single layer. One can easily argue that a depth map with finer lever of detail would have returned many more different layers for the same RoI. This also demonstrates how the technique is sensitive to the quality of the depth map. A comparison is presented in Figure 4.9 where the depth map of the *Ballet* sequence is compared to that of the *Balloons* sequence.

Another experiment was then carried out on the *Ballet* sequence based on the

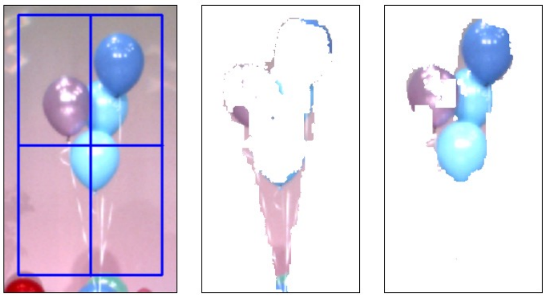


Figure 4.8: Output of the selection of the floating balloons in the *Balloons* sequence. The left image presents the selection of the balloons while the centre image shows how the strings holding the balloons were detected as a separate layer from the balloons that were then presented in the image on the right. The technique generated a total 38 layers for this object.

RoI demonstrated in Figure 4.9-a. Intra-object segmentation was computed on this RoI and it returned 30 layers of information. These layers are presented in Figures 4.10, 4.11 and 4.12 respectively.

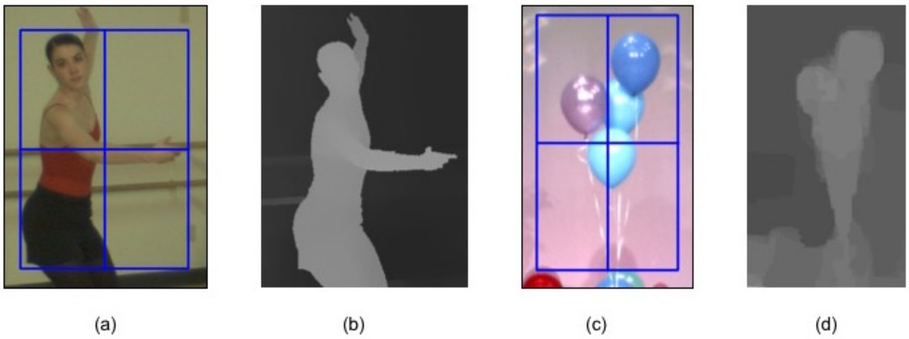


Figure 4.9: Comparison of the results of the *Ballet* and *Balloons* sequences. The difference in the quality of the depth maps in (b) and (d) strongly affects the results.

4.5.2 Algorithmic Performance

This technique was implemented in Python 3.6 and OpenCV 3.0 as a proof of concept. Performance testing was carried out on a machine with a 2.6 GHz Intel Core i5 processor and 8 GB 1600 MHz DDR3 memory running a MacOSX El Capitan v10.11.5 operating system [21]. Performance evaluation assumes a single

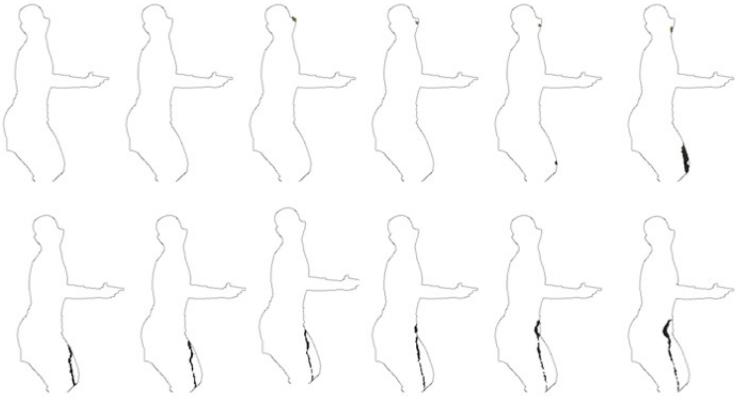


Figure 4.10: The first set of 12 layers extracted from the *Ballet* sequence. These include the layers that are most distant from the viewer.

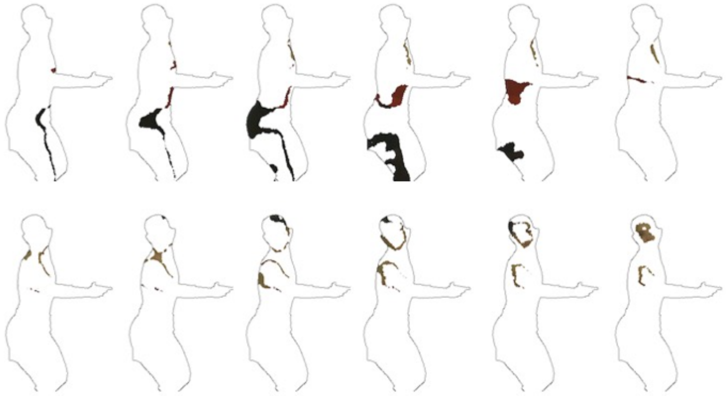


Figure 4.11: The second set of 12 layers extracted from the *Ballet* sequence

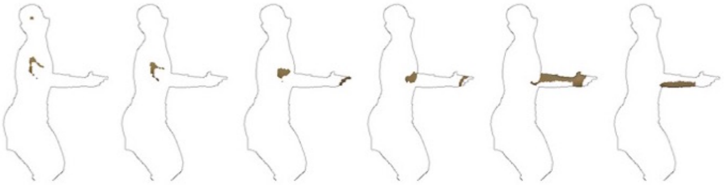


Figure 4.12: The third and last set of 6 layers extracted from the *Ballet* sequence. These are the layers that are closest to the viewer.

view such that focus is placed on the novel contribution of this thesis. On average, the program returned a result in 2.04s. The worst time was recorded when a large object (85K pixels) is selected and this was 3.34s. On the other hand, when the sub-object was smaller (9K pixels) the program returned a result in 1.29s.

Computationally, this technique is directly dependant on the size of the RoI. The number of bins in the histogram also contributes the performance. However, this parameter does not effect the computational complexity since it is a constant at worst case. In practice this is either 256 for the commonly available 8-bit depth maps or 65,536 for the less popular 16-bit depth maps. The complexity of the algorithm for the layer generation of a single image is therefore $\mathcal{O}(RoI_w * RoI_h)$ where w and h represent the width and height of the region of interest, respectively. This can also be represented as $\mathcal{O}(n)$ where n denotes the number of pixels in the RoI.

4.6 Conclusion

This chapter presented the two early experiments that took place in this research process that motivated the main contributions of this thesis.

The single-click object selection indicated the need for techniques to be able to efficiently select objects with limitations depending on where the user clicks. This ushered the way towards a saliency driven approach presented in the next chapter where selection would take place on the salient objects. Chapter 7 later confirms that the Saliency Ranking technique presented in Chapter 5 matches human clicking behaviour. Moreover, the same data used to evaluate these experiments is also used in other parts of the thesis in order to facilitate further integration in future work.

The need for segmentation was also clear in early stages of this research. For this reason, the notion of intra-object segmentation was explored. This was published [21] only a few months after the publication of the Mask R-CNN [14], the current instance segmentation state of the art. While Mask R-CNN performs significantly better in the localisation and pixel-wise segmentation of objects in a scene, our technique returns detail within every layer of depth and therefore providing numerous opportunities for image manipulation.

4.6.1 Contributions Summary

The main contributions of this chapter are the following:

1. Design and implementation of a single-click technique that exploits depth information developed before the current state of the art segmentation methods;

2. Implementation of a segmentation technique that uses depth information to extract different parts of an object developed in parallel to the current state of the art deep learning based part-segmentation techniques.

5. A Saliency Ranking Approach for Image Manipulation

5.1 Introduction

While over the past couple of decades, there were a number of saliency detection models being developed, the recently structured problem of ranking objects by saliency [20] [16] [1] remains open [15]. This chapter presents a novel saliency-based object ranking model that does not require any training. It starts by demonstrating how this technique is modelled and is followed by its implementation. The chapter is concluded by showing how our novel saliency ranking technique can be used in conjunction with the Mask R-CNN object segmentation state of the art model.

5.2 Modelling Saliency Ranking

The proposed method generates a ranking order of the areas of a colour image C that is w pixels wide and has a height of h pixels based on saliency. If available, a corresponding depth map of the colour image denoted as D with the same dimensions as C can also be used to enhance the ranking result since it would correspond to how human attention reacts when objects are closer to the viewer.

A saliency map for C is generated using any of the techniques that were discussed in Section 2.3 and is denoted by $SM_T(C)$, where SM is the saliency map of C generated using technique T. In the implementation presented in this thesis, T is set to Itti's method [50] that was discussed in Section 2.3.

The input image C is divided into a grid of segments G where k is the number of segments in the grid. Every segment in the grid has the address $G(i,j)$ where i is the row and j is the column. The relative pixel-level address of $G(i,j)$ with respect to coordinates (x,y) is given by:

$$G(i,j) = \left\{ (x,y) \;\middle|\; \begin{array}{c} i\frac{w}{k} \leq x < (i+1)\frac{w}{k}, \\ j\frac{h}{k} \leq y < (j+1)\frac{h}{k} \end{array} \right\} \tag{5.1}$$

such that

$$i = \left\lfloor \frac{xk}{w} \right\rfloor,\ j = \left\lfloor \frac{yk}{h} \right\rfloor \tag{5.2}$$

It follows that x falls in the range $x = [0, w)$ and y in the range $y = [0, h)$ [20] and the segment address $G(i,j)$ can be calculated using Equation 7.2.

The saliency map $SM_T(C)$ generates a pixel level value for saliency of the whole image where the intensity value of the pixel corresponds to how salient that pixel is according to technique T. The method developed in this thesis generates a relative ranking of saliency for every grid segment in G. The ranking is achieved after a score S is calculated for every segment in the grid. The score $S_{(i,j)}$ for the grid segment (i,j) is given by:

$$S_{(i,j)} = w_H H_{(i,j)} + w_{CB} CB_{(i,j)} + w_{DS} DS_{(i,j)} \tag{5.3}$$

where $H_{(i,j)}$ is the entropy of the grid segment, $CB_{(i,j)}$ is the centre-bias and $DS_{(i,j)}$ is the depth score. Each of these scores are weighted by a weight w. These weights introduce the possibility of giving different importance to each component, however, in the context of this thesis, the three weights are set to 1.

The range of values for each score component depends on the grid size and the resolution of the image. The values of each component were studied using a 9×9

grid size across the RGB-D images used in the evaluation presented in Chapter 7. The values of the H component range between 2.134 and 7.191. The values of the CB component are based on the 2D Gaussian Distribution. The maximum value is at the central cell and the smallest value is at the corners of the distribution, and for this grid size, the maximum value is 1 and the minimum value is 0.412. The range of the depth score depends on the representation of depth through the depth maps. In a scenario where 8-bit depth maps are used, the minimum value is 0 and the maximum is 4.628.

5.2.1 Entropy

This technique was inspired by the way the biological visual attention system tends to be affected by regions containing visual differences. From a computer vision perspective, this means a level of difference or chaos in regions of pixels [20]. This observation lead us towards the application of information theory to saliency detection in a novel approach. Information theory has been used as an alternative to generate saliency maps as described in Section 2.3.2 however, in this case it is being used differently to analyse and rank regions in a saliency map.

The higher the differences among the pixel values in any grid segment in $SM_T(C)$, the more salient the given segment is. Shannon's Entropy H presented in (5.4) is used to generate an entropy score for every segment.

Let p_m be the pixel at $(x_{i,j}, y_{i,j})$ corresponding to the value of pixel (x, y) in the segment (i, j). It follows that entropy score H is therefore computed accordingly:

$$H_{(i,j)} = -\sum_{m=1}^{|t|} P(p_m) log_2(P(p_m)) \tag{5.4}$$

where $P(p_m)$ is the probability of finding the pixel value of pixel p_m in the designated segment (i, j) [20]. The upper limit of the summation denoted by t corresponds to the total number of pixels in a segment that is equal to $\mid \frac{w}{k} \times \frac{h}{k} \mid$.

The effect of this approach is the calculation of entropy in every grid-segment.

In this case, H is representing the rate-of-change or chaos in a given grid-segment [20]. This computation provides a good ground for a saliency ranking score since higher values of H indicate grid-segments that have significant chaos with respect to pixel values in the saliency map. It also follows that the same grid-segments on the original colour image contain features that draw attention [20].

5.2.2 Centre-Bias

The proposed saliency ranking model also allows for the consideration of centre-bias in the generation of the saliency-driven ranking score. The Centre-Bias score CB of a segment is modelled using of a 2D Gaussian distribution to simulate the effect of centre-bias. Equation (5.5) is used to generate a distribution of weights, where every segment in the grid is assigned a value depending on its position, with x and y being the dimensions of the 2d-array and σ is the effective radius of the distribution. In this implementation, the full-width-half-maximum is used as a value for σ. The segment at the centre carries the maximum weight while the segments that are furthest away are assigned a minimum value [20].

$$CB_{(i,j)} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{i^2+j^2}{2\sigma^2}} \tag{5.5}$$

5.2.3 Depth Score

In the processes of ranking saliency in a scene, one should also consider the proximity of objects to the user as explored in Section 2.3.1. Generally, objects that are closer to the user are more salient than those in the background [20]. Since saliency maps are usually generated without any knowledge of depth, the concept of calculating a depth score from the depth map D of the same image is proposed.

The Depth Score DS for each segment in G rewards segments that have pixels that are closer to the viewer. The histogram of D, denoted by $Hist_D$, is used to generate this score. The mid-point mid of the histogram's range of values is given by:

$$mid = \frac{max(hist_D) + min(hist_D)}{2} \tag{5.6}$$

where in an 8-bit depth map D, $max(hist_D) \leq 255$ is the highest bin value and $min(hist_D) \geq 0$ is the lowest bin value. The depth score DS of segment (i, j) is therefore the sum of all bins in $hist_D$ whose value is greater than mid and denoted by:

$$DS_{(i,j)} = \sum_{m=mid}^{max(hist_D)} b_m \tag{5.7}$$

where b_m is the bin with value m.

5.2.4 Saliency Ranking Score

The number of grid segments in G is $n = k^2$ and every segment has a resultant score $S_{(i,j)}$ as given by Equation 5.3. The set of all saliency scores is denoted by $\mathbb{S} = \{S_1, S_2, ..S_n\}$ where S_n is presented in Equation 5.8.

$$S_n = H_n + CB_n + DS_n \tag{5.8}$$

The elements of $\mathbb{S}$ are then sorted by their absolute value for the score of each segment. The segment with the greatest value of S is the most salient segment in image C and the order of the elements in the sorted set represents the saliency ranking for the same image.

5.3 Implementation

The aim of this technique is to efficiently rank the salient regions of a given colour image without the need of training. It is also designed to exploit the usage of depth information, if available, to enhance the quality of ranking. The DFD of this technique is illustrated in Figure 5.1. The process is also pictorially presented in Figure 5.2.

The main idea is to divide the input image into a grid of segments. The saliency of every segment is then processed separately. A model is built for every image based on the entropy of the saliency of every segment, depth information and position. The score generated by the model is then used to rank the segments.

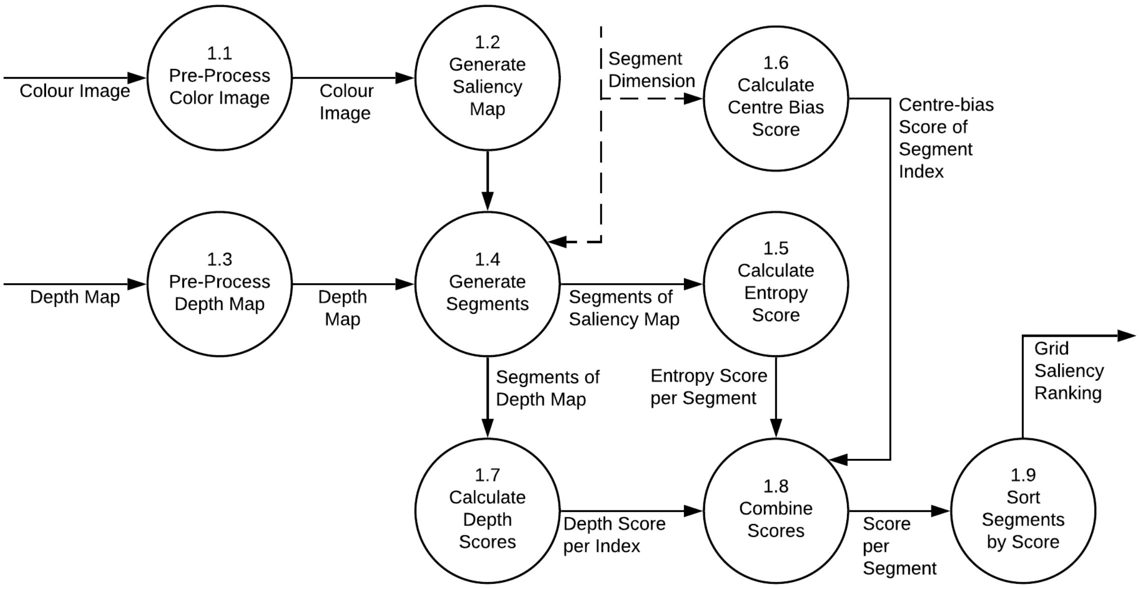
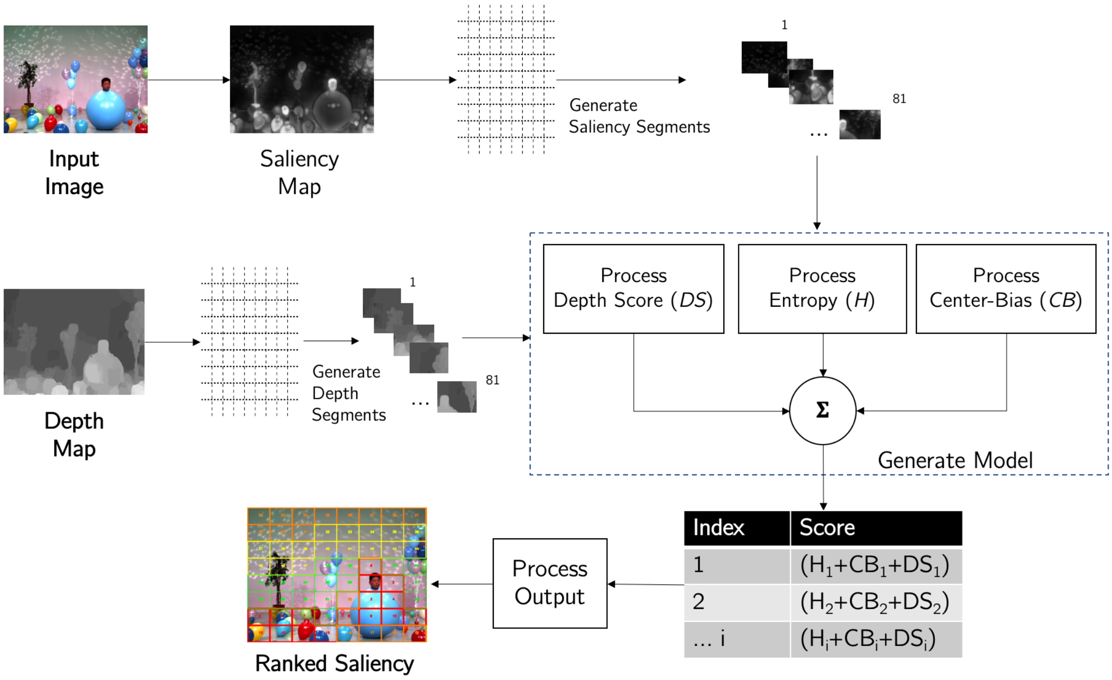


Figure 5.1: Dataflow representation of the saliency ranking technique.

The properties of the input images are processed and checked in the first stage of the process. The proposed technique can work with or without a depth map. If the depth map is not available, Modules 1.3 and 1.7 of the DFD are not be used while the rest of the modules remain unaffected in processing the saliency ranking. Following the input process, the saliency map of the colour image is generated. This technique works independently of any specific saliency map generation technique or approach as long as the saliency map is a greyscale image where the pixel values represent the respective level of saliency. The generation of results presented in this work make use of the implementation of Itti *et al.* saliency detection technique [50] implemented and freely distributed by [140]. The resultant saliency map is then split into 81 segments, and the reason behind using this number of segments follows the experiments in [20].



Index	Score
1	$(H_1+CB_1+DS_1)$
2	$(H_2+CB_2+DS_2)$
... i	$(H_i+CB_i+DS_i)$

Figure 5.2: The architecture of our saliency ranking model. This shows how the input colour and depth images are processed to generate the model that enables the ranking of saliency. This figure also shows a visualisation of the output with red segments ranking high followed by orange, yellow and green.

The model upon which the saliency ranking is based is built by processing each segment. The entropy of pixels in every segment is calculated, and this is then combined with a centre-bias score, as explained in Section 5.2.2. The resultant model maps the score of each segment with its index. This allows for a simplified and efficient manner of deciding upon the main areas of interest in the image, as discussed in Section 5.4.

5.3.1 Configuration

This section briefly outlines the main parameters required by the saliency-driven ranking algorithm, namely the grid-size and the saliency map. Furthermore, the main design decisions related to these parameters are explained.

Saliency Map

The saliency map of the input image is the enabler of the model. For the implementation and analysis of this saliency ranking technique, the Itti *et al.* [50] method, presented in Section 2.3.1, was used to generate the saliency map [20].

This method was chosen since it was the first technique of the kind and is currently one of the most popular saliency detection techniques that is used for comparative purposes. Today, different saliency techniques perform better [5], yet Itti *et al.* method was used to show that this technique performs well, even with the earliest saliency map algorithms.

The model of our technique is strongly affected by the entropy in every segment of the saliency map as explored in Section 5.2.1. The choice of the saliency map generation approach is seen as a configuration parameter since the system works independently of this choice. On the other hand, there is potential for future research that explores the enhancement or choice of saliency map that returns an improved saliency ranking [20].

Grid-Size

One of the key parameters needed is the grid size G that translates to the number of segments being extracted from the saliency map for further processing. This size, $k \times k$ also affects the size of the output since the number of scores at the end is equal to the number of segments. The grid-size can also be seen as the resolution of the model.

Our previous work in [20] presents a series of experiments that were devised to identify the ideal grid size for our technique. These experiments showed that there is no effective difference, other than the resolution of output, taking place when the grid segment dimension exceeds a factor of 8.45 [20]. This value indicates that the grid dimension can be either set to 8 or 9. The value of 9 is recommended when priority needs to be given to the resolution such as in experiments where the result of the algorithm needs to be compared to human clicks as presented in [20]. On the other hand, in cases where the number of cells affects the time performance of an algorithm, the value of 8 can also be used.

The grid size can also be used in the calculation of the optional score that models centre-bias. In this case, the 2D Gaussian Distribution discussed in Section 5.2.2 is modelled upon the 9×9 grid discussed in this section.

In their work, Kalash *et al.* [15] used a grid size or factor k of 8 for their feature map. This was because they used a ResNet-101 in DeepLab [102] for their implementation and this parameter value for k provided a balance between the semantic context and fine details in the network [102] [141].

Depth Score

One of the optional scores for saliency ranking is the depth score. Algorithm 1 is used to generate a score for depth for each of the grid segments $G(i, j)$ based on the model presented in Section 5.2.3. The depth score returns a value for each segment and grid segments with objects closer to the user are given a higher score. Depth maps from different datasets vary for different shades used to represent depth. For this reason, the histogram of the entire depth map is processed before the processing of each grid segment for calibration. The histogram H_D of the entire depth map is processed with the first and last non-zero bin indexes being recorded. The midpoint is then stored in mid, as shown in Algorithm 1.

The depth map is also split to match the same number of segments $T.size$ used in the saliency map and in the generation the 2D Gaussian Distribution. The histogram H_S is computed for each segment, and its bins' values traversed. The target of this algorithm is to supply a list of scores and to reward the segments that have objects that are closer to the camera. Hence, the bins that present an index larger than mid are considered. The summation of the bin values is then considered as the depth score for the given segment. This depth score is then appended to a list of scores whose index relates to the same index of the segment t in T as explained in Section 5.2. This list of depth scores is returned as output

Algorithm 1 CalculateDepthScores

```
Input: $DepthMap, T.size$
Output: $DepthScores$
  $H_D \leftarrow DepthMap.histogram$
  $mid \leftarrow (H_D.maxBin + H_D.minBin)/2$
  $Segments \leftarrow splitIntoSegments(DepthMap, T.size)$
  for all $segment\colon s \in Segments$ do
    $H_S \leftarrow s.histogram$
    $tempScore \leftarrow 0$
    for all $bin\colon b \in H_S$ do
      if $b.index > mid$ then
        $tempScore \leftarrow tempScore + b$
      end if
    end for
    $DepthScores.append(tempScore)$
  end for
```

by Algorithm 1.

5.4 Output

The output of the proposed saliency-driven ranking algorithm is represented by a list of segments, sorted by a the saliency ranking score S_n. The output is stored in a data structure that contains the index of the segment, the segment itself in RGB together with the top-left and bottom-right coordinates and the respective score [20]. This simplified data structure makes it easier to integrate this algorithm with other techniques. It also offers the additional capability of sorting the model in ascending or descending order, therefore indicating the less or most salient regions respectively with the most salient region being found in the segment with the index 0. A high-level illustration of the output is given in Figure 5.3. Moreover, a sample output of the proposed technique on the *Ballet* sequence [135] is presented in Figure 5.4 and three further samples are given in Figure 5.5. Based on the results of Section 7.3.2, this thesis assumes that the object that is covered by this first segment is the most salient object. The flexible data-structure also provides various further research opportunities in the traversal of the ranking with respect to the objects in a scene for alternative ways of interpreting the saliency ranking data structure output. An example of how this output can be merged with that of Mask R-CNN is given in Section 5.5.

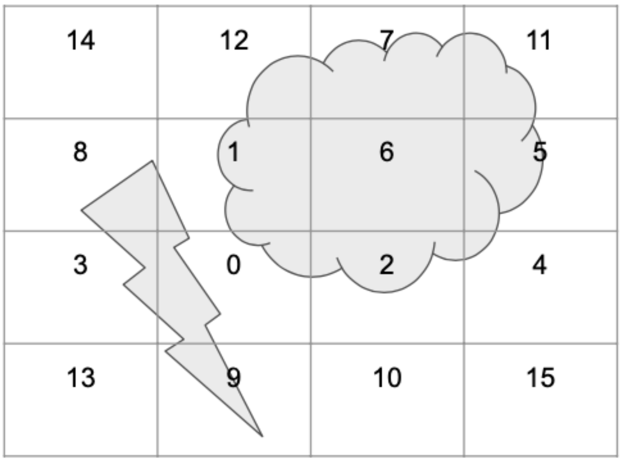


Figure 5.3: A simplified illustration of the visualised output of our Saliency Ranking model. The visualised output consists of a grid of segments that make up the entire image. Different objects are represented in different grid segment. Every grid segment has a number that indicates the segment's rank by saliency. The lower the number, the more salient the segment is.

Since this saliency-driven ranking algorithm can also be combined with deep learning techniques, its output also follows the principles of explainable AI (XAI). This is the field of study that aims to make conclusions of computer models understandable by humans by providing a traceable set of reasons on how a conclusion is reached [8]. One of the approaches, known as Local Interpretable Model-Agnostic Explanations (LIME) explains how a classifier reached its conclusion [9]. While the output of the segments in an image based on saliency ranking technique is not a classifier, it can also be used in conjunction with other techniques to classify the most salient objects. While the ranking of saliency in itself can be seen as an explainable interpretation of saliency in an image, the model behind the ranking is also further explainable in itself. The model is built by three separate scores resulting from different components as presented in Section 5.2. In practice, in understanding of why a segment was ranked in a particular way, analysis of the model

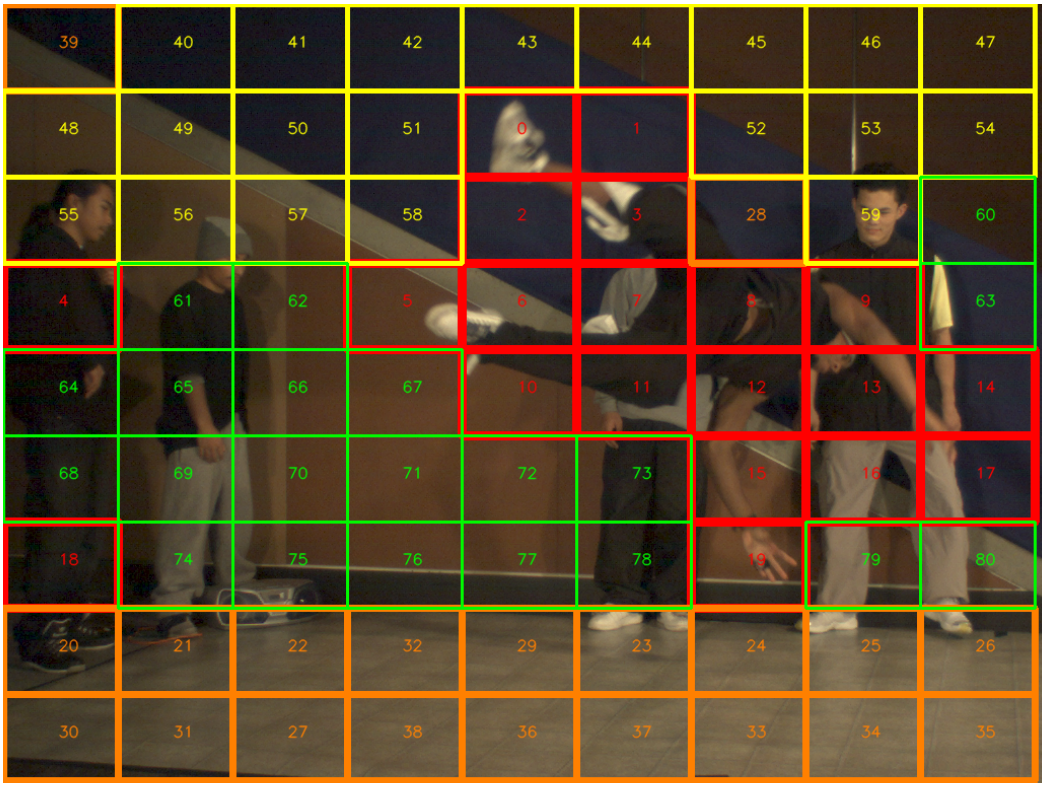


Figure 5.4: An example of the grid saliency ranking output when a frame from the *Breakdance* sequence [135] is processed using our saliency-driven ranking algorithm.

can provide the required explanation behind the decision. In this case, the explanation will include the visual artefacts and the model scores. The visual artefact includes the segment colour instance, its saliency map and the depth map segment, if made available in the first instance. The model score would then shed light on how the conclusion was reached in terms of the entropy in the segment's saliency map, its position in the image or the depth score. This approach in the output was taken since while the priority is to minimise the time in editing by shifting the manual task of object selection from the user to the model, the framework still needs to ensure that explanation about how such decisions were reached is made available to the user.

Moreover, the ranking of segments in an image using these reproducible methods provides an efficient way of evaluating attention re-targeting in an image. The output from the saliency-driven ranking algorithm clearly shows where the attention

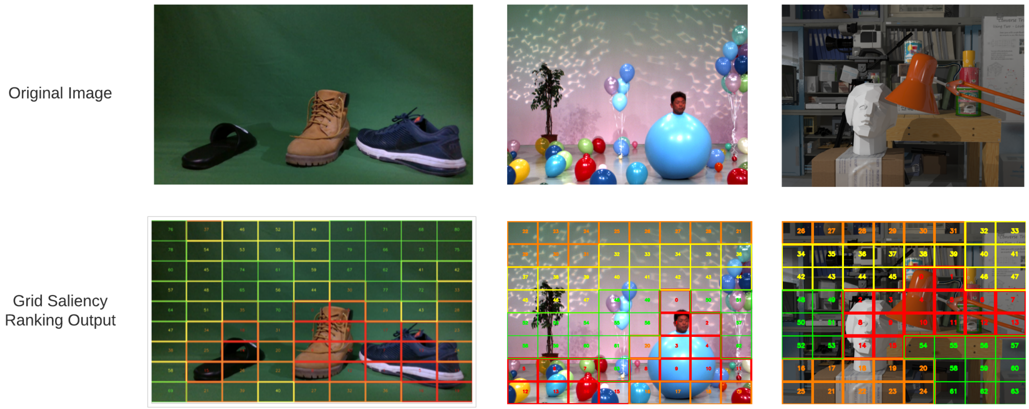


Figure 5.5: A selection of three images from different datasets together with the grid saliency ranking on each image that indicates the most salient regions in an image.

is in that current state of the image. It follows that the same saliency ranking technique can be used again on another manipulated version of the image. The saliency ranking of the manipulated image can be compared with the saliency ranking of the original image and therefore objectively measure the attention re-targeting.

5.4.1 Measuring Attention Retargeting

This section presents the saliency ranking technique in a retargeting application since it exploits its full functionality. A selection of scenes from the COTS dataset were used to generate these examples and the results are presented in Figure 5.6.

Consider an image manipulation framework where the saliency ranking of objects in a scene is measured before and after the manipulation. Images containing more than one object, denoted as S2, are fed as input to the framework, configured for inpainting. The saliency-driven ranking algorithm (SaRa) is computed on the input image S2 and the most salient segment with rank 0 is noted. Inpainting is carried out on the most salient object in S2. The saliency-driven ranking algorithm is now computed on the inpainted result and the most salient segment is now representing another object. This means that the attention of the image is retargeted to another object. The COTS dataset includes different instances of the

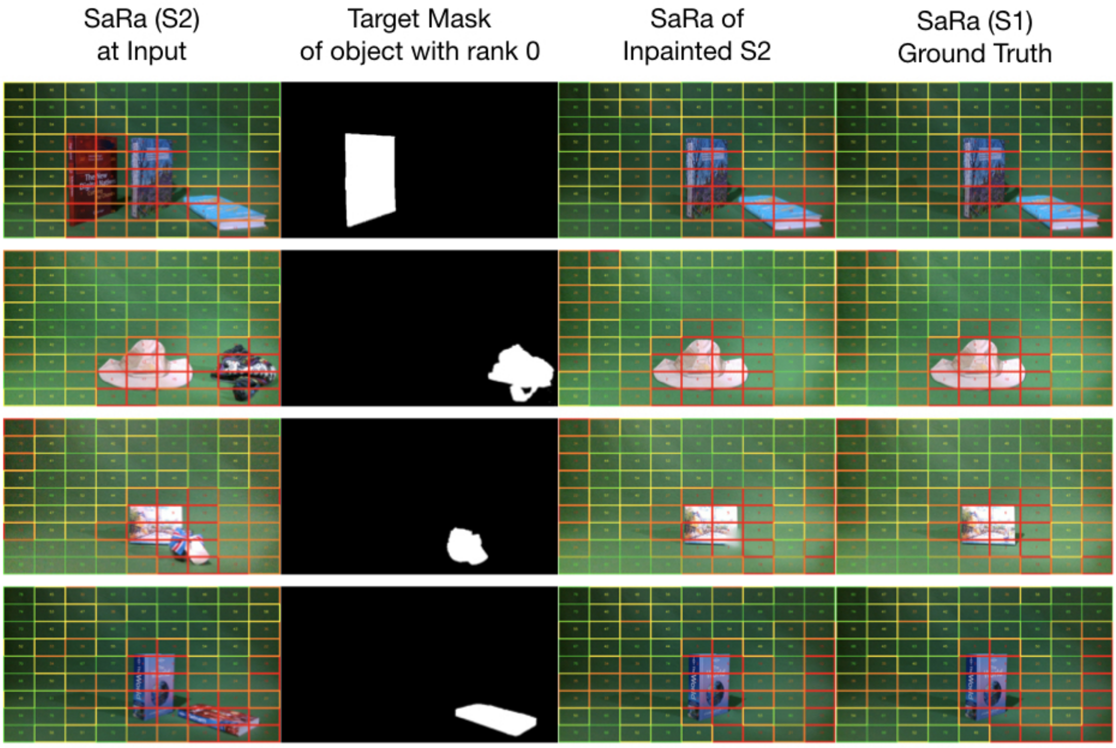


Figure 5.6: An example of using our framework for retargeting. In the first column, Saliency Ranking (SaRa) is carried out on an image with more than one object. The mask represents the object that corresponds to the most salient segment. The saliency ranking is then computed after the most salient object is inpainted and the most salient segment is retargeted to another object. The retargeting is confirmed on the saliency ranking of the same image without the inpainted image, serving as ground truth.

same scene with incremental objects as will be demonstrated in Section 6.5.2. This time, this feature of the dataset is used to track retargeting. The instance less the target object, denoted as S1, is used as ground truth. This allows us to compare the retargeting on the inpainted result against S1. In all instances, the most salient segment for both the inpainted result and S1 is found exactly in the same position, showing that the framework successfully carried out this task.

5.5 Saliency Ranking and Segmentation

Recent object detection and segmentation methods such as Mask R-CNN [14] provide pixel-level segmentation of different objects and their instances within an image. This section demonstrates how the saliency ranking method presented in this chapter can be used to rank segmented objects in a scene by their saliency.

The output of our saliency ranking technique presented in Section 5.4 can be used to rank different object masks by the level of saliency of their corresponding objects. The approach presented in this section can be used in conjunction with the inference results of any segmentation technique that returns a set of masks for the objects it detects in the input image. This is illustrated in Figure 5.7.

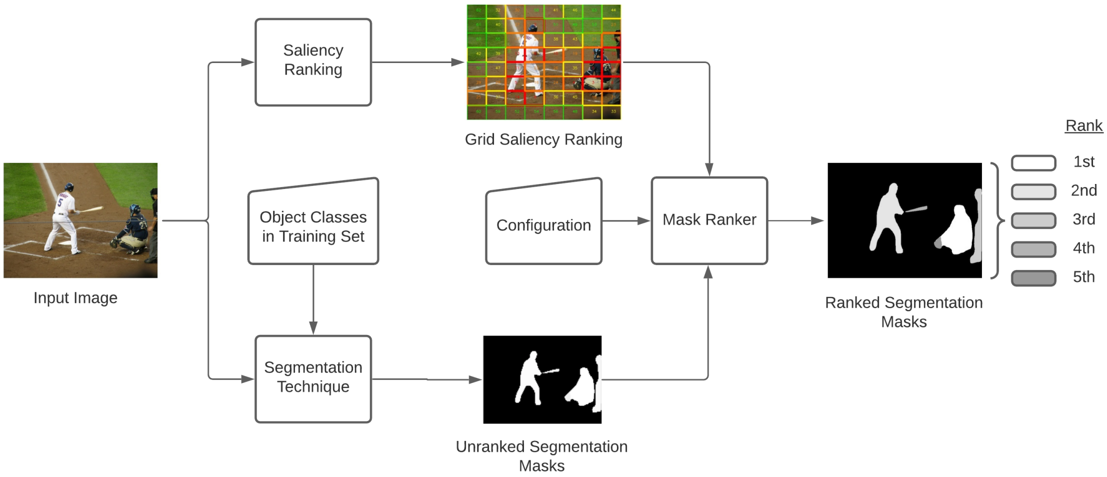


Figure 5.7: A block diagram demonstrating how our saliency ranking approach can be combined with a segmentation technique that outputs a set of unranked masks. The ranks of the grid segments are then used to calculate the rank of the generated masks.

Mask R-CNN was chosen as the model to return the segmented masks to demonstrate how our technique can be pipelined with a segmentation model. This model was chosen due to major claims that it is the state of the art in segmentation, its popularity, wide use and particularly since it was used as a backbone for the Inferring Shift Ranks method [1]. The work of Siris *et al.* [1] is a deep learning saliency ranking approach that is also accompanied by a dataset that enabled a comparative evaluation in Chapter 7.

5.5.1 Implementation

The Matterport implementation [142] of Mask R-CNN was used as the basis for this setup as the segmentation method of choice. This instance is a pre-trained ResNet-101 [72] on the COCO dataset [71]. This implementation is based on Tensorflow and uses a resized version of the images to 1024 × 1024 pixels. While it is not explicitly indicated, it appears that the work of Siris *et al.* [1] uses the same implementation of Mask R-CNN for the backbone of their saliency ranking model.

The proposed saliency-driven ranking method was applied to the inferred output of this implementation of Mask R-CNN. For this part of the experiment, the grid-size factor k was set to 8 in order to match the same properties of the implementation in [1].

5.5.2 Applying Ranks to Masks

Given that the output of our method returns a grid and Mask R-CNN returns a pixel-level mask for each object, we needed to devise an approach that passes the rank value of the grid cell to the corresponding mask resulting from the segmentation technique.

The output of the segmentation model consists of a set of masks $\mathcal{M}$ where $m \in \mathcal{M}$. The entire area of the image is covered by an 8×8 ranked grid based on the output of our saliency ranking technique, where the i^{th} grid segment is denoted by G_i. A mask M_n of an object can overlap one or more grid segments. It follows that pixel $p \in G_i$ would also be $p \in M_n$. The saliency rank of M_n therefore depends on the rank values of the grid segments in which the mask falls. The pixels that are both in G_i and M_n make up the area of coverage of a grid segment by a mask and is denoted by $G_i \cap M_n$. When the area of coverage is greater than a threshold $T\%$, the rank of grid segment G_i is attributed to mask M_n and stored in MR_n.

The allocation of grid segment ranks to a corresponding mask is represented in Equation 5.9. Experiments were carried out to determine the value of $T\%$ and these are presented in Section 7.5.2 where we demonstrate that the best value for T is 90%.

$$MR_n = rank(G_i); (G_i \cap M_n) \geqslant T\% \tag{5.9}$$

$$M_n = argmin(MR_n) \tag{5.10}$$

The list of corresponding ranks MR_n related to mask M_n then need to result in a relative saliency rank for every mask in $\mathcal{M}$. The experiment illustrated in Section 7.5.2 demonstrates that the minimum rank in MR_n gives the best saliency rank for M_n as presented in Equation 5.10. This methodology ultimately results in a translation of rank from the grid of our saliency-driven ranking algorithm to the mask in the output of the segmentation technique.

5.6 Conclusion

This chapter presented one of the main contributions of this thesis. This is the saliency-driven ranking technique that splits the image into a number of cells or segments and applies a saliency ranking score to every segment. In this chapter we demonstrated how this technique was mathematically modelled and subsequently implemented. The method and implementation use a limited grid size to facilitate benchmarking evaluation of the technique. Future work should investigate the effect of different grid sizes with respect to the detection and ranking of objects that have different sizes. The chapter is concluded by showing how our novel saliency ranking model can be seamlessly pipelined with the Mask R-CNN segmentation model.

5.6.1 Contributions Summary

The main contributions of this chapter are the following:

1. Mathematical modelling of the saliency-driven ranking model;
2. Implementation of the saliency-driven ranking model;
3. Application of the saliency-driven ranking algorithm output to a combined use with Mask R-CNN.

6. The COTS Evaluation Dataset

> If the structure does not permit dialogue the structure must be changed.
>
> Paulo Freire

6.1 Introduction

This chapter presents the COTS dataset that was specifically designed to provide an objective way to evaluate different computer vision applications. This chapter outlines how this RGB-D dataset was designed and constructed. A series of experiments to demonstrate the use of the COTS dataset on different computer vision applications such as salient object detection, segmentation and inpainting are also included in this chapter. This dataset was also published in [23] and [10].

6.2 Existing RGB-D Datasets

Over the years, an extensive list of RGB-D datasets were developed. These were surveyed by Firman [143] and were organised into different applications. The need for RGB-D images extends throughout different fields of research and applications. This ranges from object detection, segmentation and classification and visual saliency. The area or scale of the environment being captured in the dataset also varies. There are datasets, such as COTS, that feature small objects and others are designed to capture larger scenes such as a room or even outdoor environments [143]. This section surveys the RGB-D datasets that focus on small objects since these are the only ones comparable to the dataset being introduced in this thesis and published in [10]. Furthermore, datasets that have been traditionally used to benchmark saliency detection techniques will also be surveyed due to the specialised application of this dataset. Below follows a list of the datasets considered in this thesis and a set of visual samples from these datasets is given in Figure 6.1:

OBJS : RGB-D Object Dataset [144]

BBIR : Bigbird Dataset [145]

SCAN : A large dataset for object scans [146]

OSEG : Object Segmentation Dataset [147]

GLHY : Global Hypothesis for Verification for 3D Object Recognition [148]

SSEG : RGB-D Semantic Segmentation Dataset [149]

Existing datasets focus on a more holistic 3D reconstruction of the objects being captured where a DSLR camera was used together with a PrimeSense Carmine to construct a point-cloud representing the objects [144] [146]. RGB-D datasets can also be captured using different technology. Microsoft Kinect used to be a popular device to generate such datasets [143]. However, since its discontinuation in October 2017, it left a gap in the future of capturing RGB-D datasets. In this context, we use the Intel RealSense D435 Depth Camera [1] to construct this dataset.

The *RGBD Object Dataset* (OBJS) [144] was constructed indoors, within a what is claimed to be a controlled environment. The single objects were constructed after

[1] The full documentation of the Intel RealSense Camera can be found on: https://realsense.intel.com

Dataset	**Categories**	**Images**
OBJS	51	250000
BBIR	1	100
SCAN	1	10000
OSEG	1	111
GLHY	35	50
SSEG	16	NA
COTS	*29*	*120*

Table 6.1: Every dataset has a different number of images organised in a number of categories.

Dataset	**OBJS**	**BBIR**	**SCAN**	**OSEG**	**GLHY**	**SSEG**	***COTS***
Masks	Yes	Yes	No	Yes	No	No	*Yes*
Setup Info Available	No	Yes	Yes	No	No	No	*Yes*
subject Interaction Data	No	No	No	No	No	No	*Yes*
Semantic Segmentation	Yes	No	No	Yes	Yes	Yes	*Yes*
Controlled Lighting	No	Yes	No	No	Yes	Yes	*Yes*
Pointcloud or 3D Mesh	No	Yes	Yes	Yes	Yes	Yes	*No*

Table 6.2: An overview of the different types of data available in datasets.This includes the availability of object masks, setup information, semantic segmentation data and point-clouds. This information is also supplemented with an indication whether controlled lighting was used or not.

the RGB-D image of a room was captured. This was followed by an approach that used a mask, the colour image and depth to extract objects of interest. Due to this reason, one cannot guarantee a constant preservation of attributes such as shadows and lighting across the dataset. This was one of the priorities in the construction of our COTS dataset and the detailed process is outlined in Section 6.3. On the other hand, the *Bigbird Dataset* (BB) [145] was constructed in a very structured manner. Objects were placed onto a turn table that also included a calibration check-board. Two pairs of DSLR cameras and a corresponding Carmine 1.09 sensor for each camera were placed in front of the turn table. In this case, 600 RGB-D frames

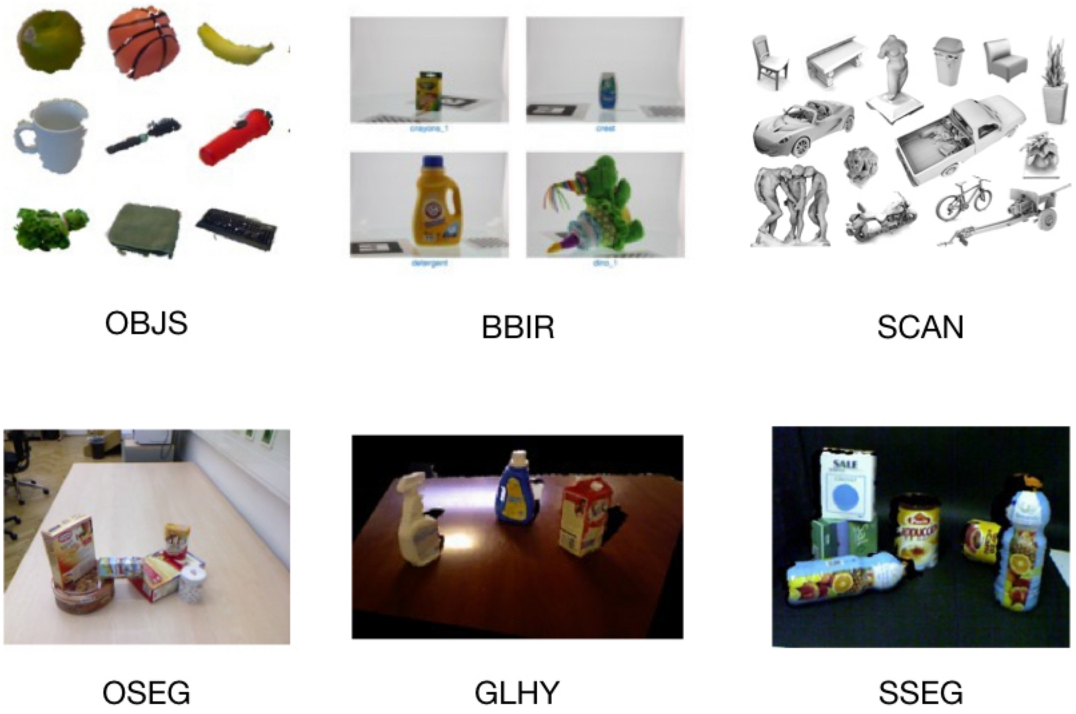


Figure 6.1: A selection of similar RGB-D datasets also reviewed in [143]

were captured through this approach [145]. The *Large Dataset of Object Scans* [146] was also captured using the Carmine sensor. In this case, the priority of the dataset was the acquisition of a large number of images and the process was crowd-sourced to non-professionals. Every object was captured using a video footage and its motion was used to construct a point cloud. This however hindered image attributes across the dataset from being constantly preserved.

The remaining datasets [149] [148] [147], as mentioned, included small objects thata were captured using the Kinect v1. These small objects were placed on a table during the acquisition process. This setup is very similar to the COTS dataset. However, these papers do not document the setup used to capture the data and the lighting conditions while in other situations it was subject to variations. Such an approach makes the dataset suitable for segmentation applications, however it makes it difficult to use the dataset for objective evaluation of image manipulation methods.

All the datasets surveyed include individual static scenes and their categories and number of images is given in Table 6.1. A summary of the key features of the surveyed datasets is presented in Table 6.2. On the other hand, the COTS dataset was designed and built to progressively include objects in a scene while leaving the previous objects in the scene. The objects already in the scene would remain unmoved and the lighting conditions also remaining the same. This incremental approach is presented in Figure 6.11.

6.2.1 Datasets for Visual Saliency

Saliency detection is the problem of detecting the parts of an image that attract more attention than others [7] [5]. There are different approaches to detect saliency. These include the ones similar to the first method proposed by Itti *et al.* [50] that was highly influenced by the human way to visually perceive an environment. There are also more modern deep learning approaches [55] [57] [58]. Itti's approach is a bottom-up approach that exploits features and visual attributes of a single image. On the other hand, deep learning approaches detect patterns that are attributed to visual saliency after training on extensive datasets. For this reason, some datasets used in saliency detection benchmarking include a large number of images. Among the different datasets used for this purpose, one finds the MSRA10K [150] containing 10,000 images that are accompanied by a binary masks and the CAT2000 [151] dataset containing 4,000 images. These datasets can be split in training and testing sets respectively, where the former set is used to train machine learning models. The dataset proposed in this thesis is not designed for the training of such learning-based techniques.

On the other hand, the COTS dataset is intended to usher the way towards tackling the upcoming challenges in saliency detection. The current saliency datasets contain single objects and this is evident in various benchmarking exercises [7] [5] [39]. Moreover, the next challenge in saliency detection is the ranking of saliency in images that contain more than one object [7]. Siris *et al.* recently proposed their own dataset that is specifically labelled for saliency ranking. This dataset is different from COTS since it is an adaptation of the COCO [71] and SALICON [18] datasets and is intended to be used for the training of machine learning models.

The challenge of ranking saliency in images has been tackled at pixel level in some work [69] [70]. These techniques attempt to rank saliency in an image based on the weight of saliency at pixel level. The limitation of such an approach is that values pertaining to a single pixel would be out of context if considering the image at object level, where every object would be made up of hundreds or thousands of pixels. The alternative approach that tackles this problem dissects the image into segments or regions and processes the weights of the segments as a hole before sorting the regions by their level of saliency [20].

The currently commonly used datasets for saliency detection do not offer any depth information. The need to study saliency detection algorithms in relation to RGB-D content is seen as a current challenge in the area [39] and the proposed dataset aims to provide a way to explore this further.

6.3 Constructing the Dataset

The process of constructing the COTS dataset is presented in this section. The dataset was made available for free on http://cotsdataset.info. The motivation for the construction of the COTS dataset was to have a single dataset that can be used for the evaluation of different stages of a computer vision application.

The first part consists of images of single objects located on a green surface while also having a green background. The second part uses the same setting but consists of scenes containing objects belonging to a theme.

This second part of the COTS dataset contains 27 scenes, with every scene containing a set of objects that were incrementally included. An example is presented in Figure 6.11. On average, every scene has three incremental instances. This excludes the blank scene, denoted as instance 0, since it is the same for every case. There are 88 instances in total, organised into two parts as explained below.

The dataset has a travelling theme it contains objects organised by categories related to travel and technology. The theme provided easier semantic categorisation of content while also providing the opportunity of future expansion of the dataset. The scenes were also carefully crafted to include occluded objects for every aspect for the evaluation of techniques that are affected by occlusion.

The choice of objects was also based on their material and reflective properties. The materials include polished glass (tagine, mug and statues of Genisha and Buddha), transparent glass (shooter glass), matt paper (Google Cardboard and most of the books), textile (Daydream VR headset, shoes and headgear), metal (Macbook and travel-mug) and plastic (headphones, washing containers). Objects also vary significantly in size and range from a small shooter glass to a tagine and a laptop [10].

6.3.1 Data Collection

A dedicated controlled environment was used to construct this dataset. The setup is visually presented as a plan elevation in Figure 6.2. The image acquisition process was carried out indoors, in a room without external lighting. The only light on the objects was from two auxiliary LED lighting with diffusers. The auxiliary scene lighting was carefully chosen to ensure that it was not generating infra-red noise that negatively affects the quality of the captured depth map. A designated region on the surface was clearly marked and the objects were placed within this region that fell precisely within the camera's field of view. The configuration of the scene was measured and recorded. The setup was kept constant throughout the scene capturing process [10].

The Intel RealSense depth camera D435 which uses active IR stereo technology was used to capture the images presented in this dataset. The realsense-viewer tool in the official SDK was used to initially calibrate the camera settings and then for recording. The recordings were taken indoors, in a room without external natural

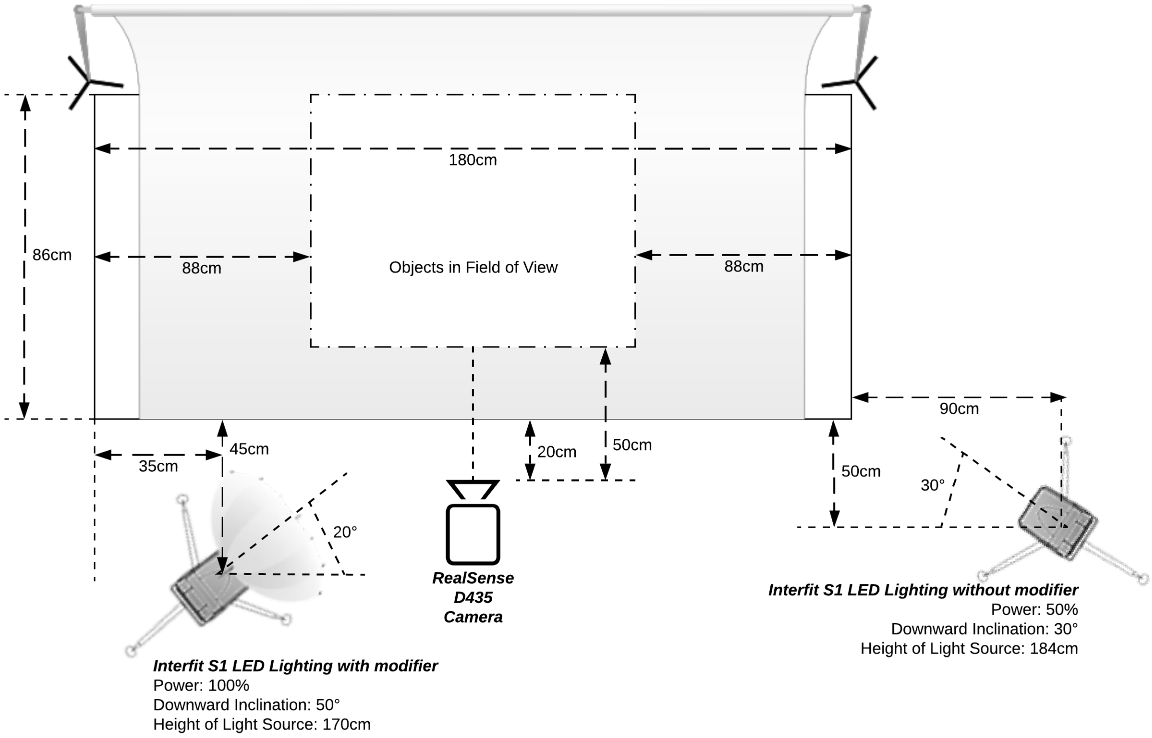

180cm
86cm
88cm
Objects in Field of View
88cm
90cm
45cm
35cm
20cm
50cm
50cm
30°
20°
RealSense D435 Camera
Interfit S1 LED Lighting without modifier
Power: 50%
Downward Inclination: 30°
Height of Light Source: 184cm
Interfit S1 LED Lighting with modifier
Power: 100%
Downward Inclination: 50°
Height of Light Source: 170cm

Figure 6.2: A plan elevation of the studio layout used for the data collection. This diagram is not to scale.

lighting as shown in Figure 6.3. Static scenes were recorded as a 6s video sequence and saved as a Robot OS (ROS) .bag file that contained all the raw data streams. This method was used so that all raw data is preserved and could then be exploited for colour/depth alignment, depth measurements and hole filling algorithms. This also provides subjects of the COTS dataset with the opportunity to perform a more refined selection of the frame if need be.

Various reasons led to the decision of using the Intel D435 for this particular task. Apart from being an affordable model, the camera offers one of the highest resolutions with high-accuracy depth reading within the recommended range of 0.2-7m and has a depth field of view of 87° × 58°. Its properties, presented in Table 6.3.1, also provide the required flexibility for such a task. Stereo cameras were impractical and usually disregarded due to the weak operation in low-texture

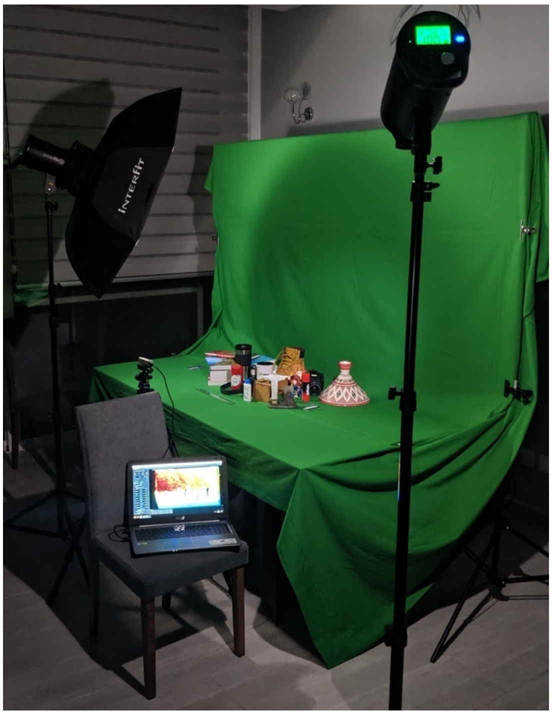


Figure 6.3: The environment where the COTS datset was captured.

Image resolution	1280×720 pixels
Video frequency	30 Hz
Extracted scene files	Colour image (jpeg), 16-bit depth map (png), 8-bit depth map (png), Raw ROS .bag file
Intrinsic Parameters	ppx: 623.328, ppy: 361.712, fx: 924.744, fy: 925.107
Depth scale (only used for 16-bit images)	0.001

Table 6.3: Intel RealSense Camera Properties

scenes. However, the D435 is equipped with an active stereoscopic IR projector which transmits its own pattern, at up to 90fps depth refresh rate, allowing for information to be gathered even on low-textured surfaces. The dedicated ASIC chip within the camera is conducting the necessary edge computing for the dense image registration problem required with all stereo technology. The latter is done in real-time and thus the whole camera system outputs directly the depth information, allowing for minimal computation on the host platform. Finally, the Intel open-source community is one of the most resourceful on various platforms and hence it is easier to get started and progress during development. The algorithm introduced below is built-in the D435 SDK which is updated on a regular basis.

Hole-Filling

During the extraction process of the aligned depth frames from the recording it was noticed that the depth map contained "holes" - which represent missing information. In a stereo system, there are a number of reasons for this artefact as portrayed in [152]:

1. **Occlusions** – This occurs when the right and left images do not include the same scene or object due to coverage or shadowing. The left view is usually used as reference and hence the occlusions effect is noticed on the left side of objects and along the left edge of the image;

2. **Low-texture** - The basis of stereo matching algorithms depends on the texture matching in the right and left images. In texture-less surfaces like a flat white surface, the depth estimation provides challenges and for this reason, texture is generated using an active projector;

3. **Multiple matches** - It may be the case that during the matching process, more than one block is found to match equally with the reference one. This is common when the scene comprises a uniform periodic structure;

4. **Signal** - Whenever the images are under or over exposed there is a lack of signal, hence no information;

5. **Out of range** - The search range of the algorithm is exceeded when the object is too close to the camera. Objects need to be more distant than the minimum distance Z from the capturing device, to be detected and recorded.

For certain applications, especially those that require real-time processing, sometimes it is better to not deal with holes since the processing might be too intensive. However, in cases where depth-enhanced output is desired, a “best guess” is better than no guess [152].

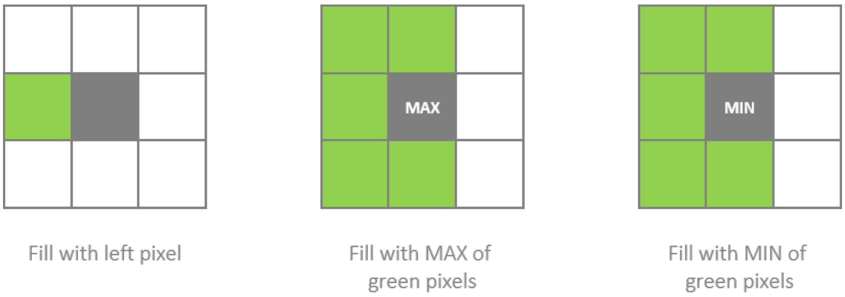


Figure 6.4: Different hole filling methods

The algorithm used for this task falls under spatial filtering. This simple algorithm uses the neighbouring pixels (left or right) within a specified radius to fill the blank pixel with the results being presented in Figure 6.5. This technique is used in various literature as a baseline for comparing new hole filling methodologies [153]. For the D435 camera, the left neighbouring pixel is taken since the left camera is the reference. Within the Intel SDK, three methods are available:

1. Left valid pixel value;

2. The biggest (farthest away) among the valid five upper left and down pixel values (used for depth map) as shown in Figure 6.4;

3. The smallest among the valid five upper left and down pixel values (used for disparity map).

6.4 COTS Web Test

One of the main aims of this dataset is its use in the evaluation of visual saliency techniques and frameworks. This needed the colour images and the related depth

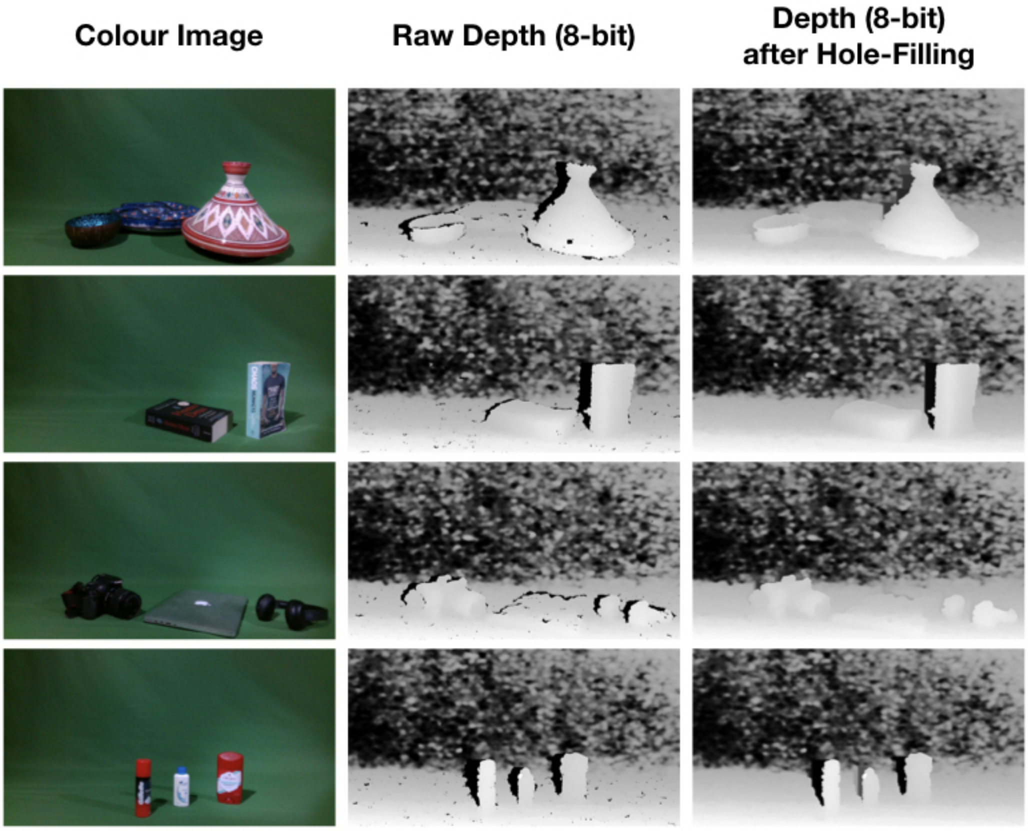


Figure 6.5: A sample of depth maps (raw) as captured by the camera (middle images) and their output of the hole-filling method (right images). The colour images (left images) are also shown for reference.

information and object masks to be included with the data that illustrates how humans relate to the content of this dataset. In order to achieve this objective, an online test was designed and built.

The choice of deploying this test through a website was based on its potential for applications and scalability. This experiment successfully attracted 1267 respondents through the sharing of the experiment URL. A website with its respective backend was specifically built for this experiment. This online test was designed to allow subjects to be presented with colour images on a web-page from the dataset while Javascript was used to collect usage data in the background in order to minimise the impact on the user experience. No sensitive subject data was collected in this experiment. The usage data was stored in a hosted database. Google Analytics were also deployed to monitor usage activity of the experiment website.

Figure 6.6: A set of separator images were chosen to avoid visual bias from previous images.

Subjects were presented with batches of 10 images, displayed one at a time. The main challenge was that the dataset contains 84 images that each needed to be presented to a subject or another. A sequential approach was not deemed to be feasible since it would skew the data towards the first occurring images in the dataset. A specifically designed load-balancing algorithm was implemented in the backend in order to evenly distribute the images in the dataset [10]. This algorithm is based on two requirements. The first requirement is that images from all across the dataset needed to be featured evenly. The second requirement was that subsequent images from the same scene were not to be displayed in sequence. The latter requirement emerged from preliminary laboratory testing of the experiment where it was noticed that when subjects were presented with incrementing objects of the same scene, such as the instances in Figure 6.11, they were more prone to click on the new object rather than considering what is actually more salient [10]. As a further precaution, three separating images were displayed before every dataset image presented to the subject in an effort to minimise the visual bias from the preceding image. A random separating image from those presented in Figure 6.6 was loaded. The visual inconsistency of these images made them ideal for this purpose. This algorithm was successfully deployed and every single image in the dataset was clicked by 213 unique subjects.

The focus of the exercise was to present a colour image from the dataset to the participating subjects and see where they think is the most salient region. A single image was loaded on every screen and for each colour image, the subjects were asked the following question:

Task:

> *Click/Tap on the point that attracts your attention when you first see the image. The point can be anywhere and includes persons or other objects.*

The subject click or tap coordinates were preserved for every image. These were then used to create a heatmap of the clicks/taps. This dataset includes a CSV file with the click coordinates for every image. When the experiment was carried out through a mouse-enabled device, such as laptop or a desktop computer, the movement coordinates were also collected. The time between the loading of the image and the subject click/tap was also measured. This allowed for a better understanding of subject behaviour, since subjects clicking in shorter periods of time were more likely to be impulsive and click on what they deemed to be more salient.

6.4.1 Web Test Implementation

The implementation of the online data collection method is outlined in this section. The Data-Flow Diagram presented in Figure 6.7 shows the main modules. The two main data collection components are related to the image selection (Module 1.0) and user handling (Module 1.1).

The Image Selection module (1.0) spawns the images made available to the subjects as explained in Section 6.4. Module 1.0.1 selects the images from the COTS dataset (1.0.3) and prepares a subset of images (1.0.2). These selected images is then presented to the subjects on the HTML web page [10].

The User Handler module (1.1) handles the presentation of the chosen subset of images to the subjects (1.1.0) and the resultant activity information collection (1.1.1). The user handling module stores the information related to the image, such as the click coordinates and cursor movement when available. On the other hand, it also stores other information such as the time of interaction and the type of device that was used for the test [10].

This architecture enabled a scalable dissemination of the online test and the successful completion of the test by the subjects.

This section provides an analysis of the subject interaction on the dataset during the online test. Out of the total 1267 respondents, 77% used a hand held mobile device to do the test and 6% used a tablet. The remaining 16% used a desktop or laptop computer. This means that 83% of the respondents were tapping the images and therefore no cursor movements could be collected in these cases.

A total of 1690 persons visited the site and 1267 of them completed the test. This meant a bounce rate of only 25%.

Figure 6.8 presents the average annotation map the clicks and taps gathered during the online test. This shows a balanced overall distribution along the entire area where objects were placed. Furthermore, it also shows a reduced center-bias.

Evaluating whether the amount of subjects participating in the study is enough to warrant further evaluation is not a straightforward task. In the area of com-

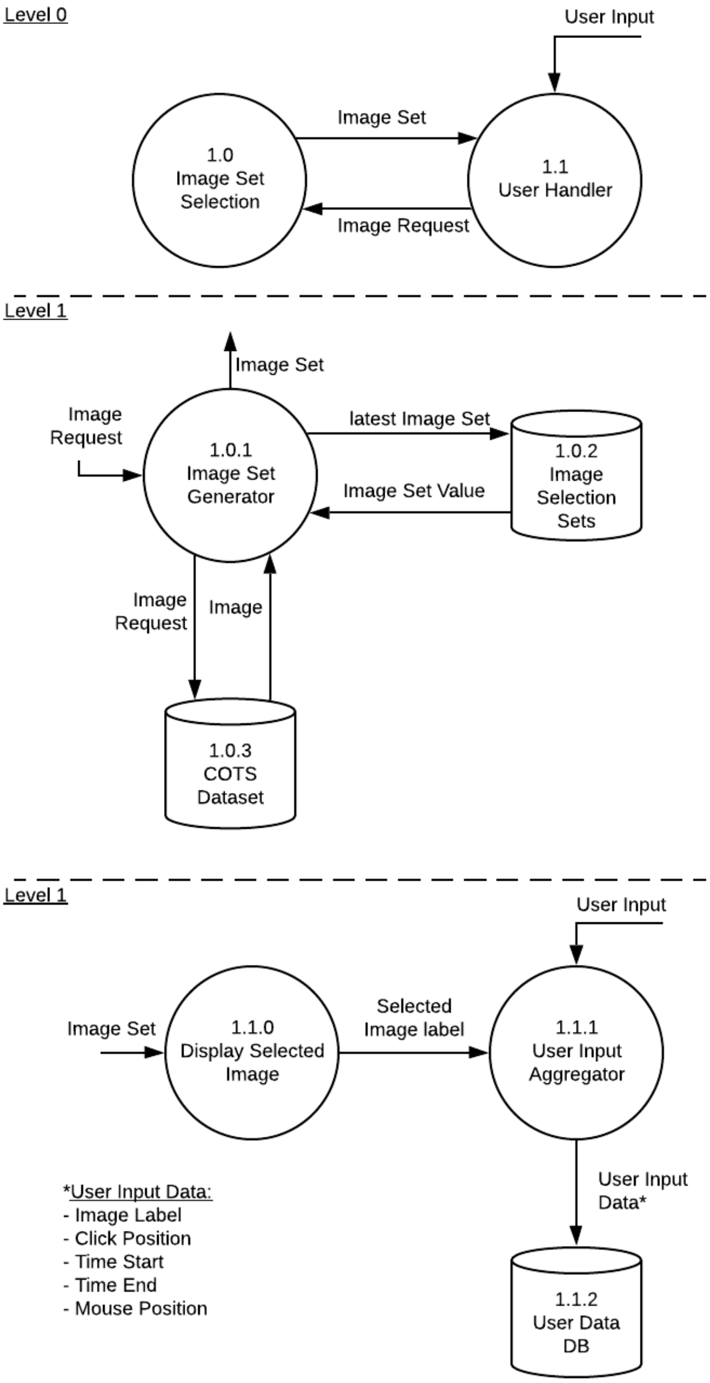


Figure 6.7: A dataflow diagram of the online test architecture

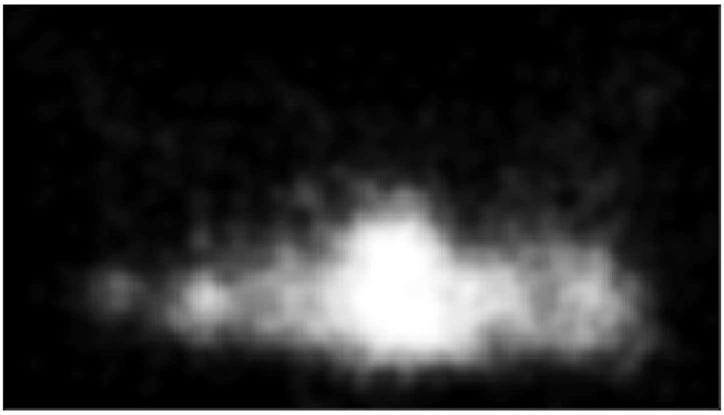


Figure 6.8: The average annotation map resulting from the taps or clicks gathered from the web test.

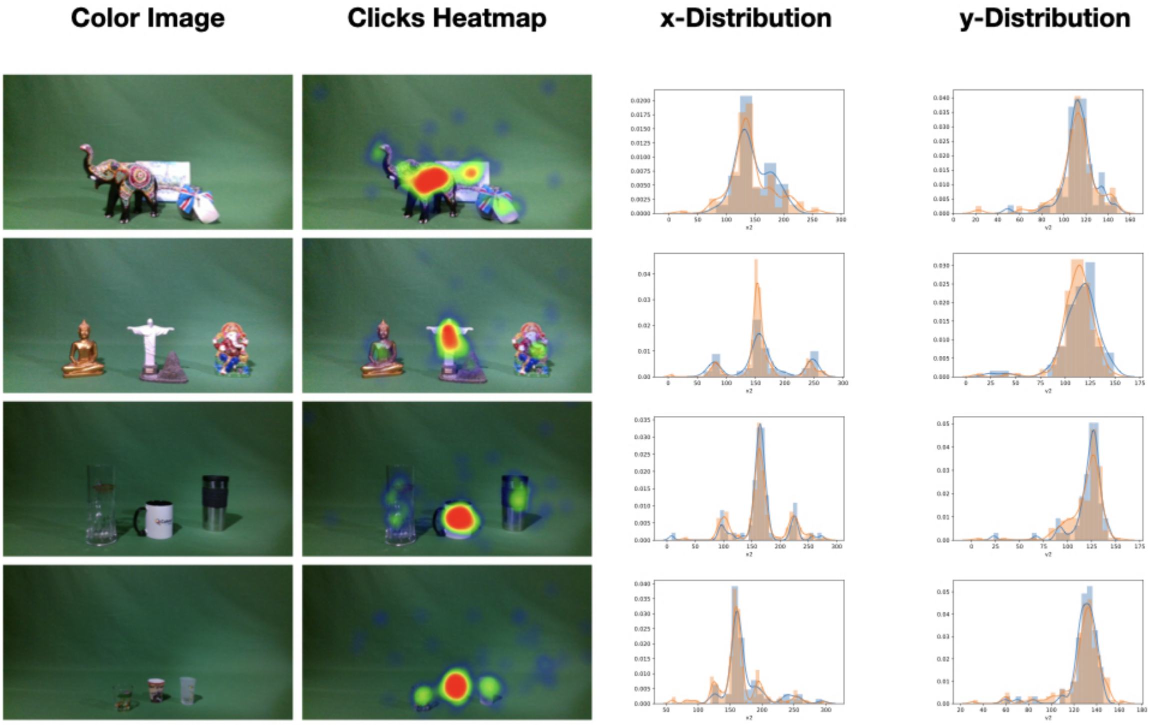


Figure 6.9: A selection of heatmaps created from the subject interaction data gathered during the online test. The distribution testing of the x and y coordinates is also presented on the right.

puter vision, especially, traditional statistical techniques that calculate a minimum threshold for a population are inconsistent in such a case, since through this experiment, we do not measure opinion. Our approach tests statistical validity by dividing the total number of mouse clicks into two groups divided into a ratio of 3:7. The distribution of the x and y coordinate values between the two groups was compared. Figure 6.9-left images describe some of the results gathered from the study. Visually, the heatmap already gives a strong indication of the expected result given the tendency of the clusters to focus on specific points in the image. Furthermore, the similarity of the x and y distribution curves across both groups emphasis our methodology as seen in Figure 6.9-right images. To consolidate the process, a t-test was performed and it compared the x and y distribution of clicks across the two groups. The null-hypothesis was defined accordingly:

$Hypothesis H_0$**:**

> The distribution of the clicks (x and y dimension) on the smaller sample size is similar to the distribution of the clicks on the larger sample size.

A t-test was carried out for each image. In the cases where results showed the p-value to be higher than 0.05; therefore, we cannot reject the null hypothesis. This means that there was no significant difference between the two distribution and the click or tap coordinates settled to specific regions.

6.4.2 Annotation Process

Single channel binary masks corresponding to an object in a coloured image are a common way of annotating datasets in computer vision. These masks are black and white images representing the target object, represented by white pixels. These masks are also important for the evaluation of other techniques that generate a mask from the depth information [22].

These masks were created using the LabelMe tool[2]. The masks used in the

[2] `http://labelme.csail.mit.edu`

COTS dataset were generated by three third party annotators. A mask was generated for every object in all of the 28 scenes. The third party annotators were not part of the project. Every annotator created a mask for each object.

The next step was the inter-annotation agreement between the three masks. There are different approaches of choosing the final mask. These vary from choosing either the smallest or the larger masks or even an average. We felt that such an approach might introduce certain bias and from experiments it was noticed that they were also introducing scattered white pixels in the output masks. Such an effect is undesired. Therefore, a more conservative approach was followed. The approach used outputs a white pixel on the final mask only if there is a white pixel in all the three masks in the corresponding position [10].

6.5 Usage of Dataset

When designing the COTS dataset, different applications of computer vision were considered. These were explored in detail in the journal paper outlining the dataset itself [10]. This section demonstrates how the COTS datsaset can be used in applications that are directly related to this thesis. These include saliency detection together with inpainting and blending applications. The last part of this section discusses how pipelined computer vision modules can also be evaluated using this dataset.

6.5.1 Visual Saliency

Visual saliency is an intriguing topic in computer vision that motivated a number of researchers to explore and develop techniques to generate more accurate saliency maps. A range of techniques that include eye-fixation models and deep learning models are used to generate a map that approximates human visual attention [5] [7]. Saliency techniques can be organised into two categories namely those based on the prediction of fixation and Salient Object Detection. The process of salient

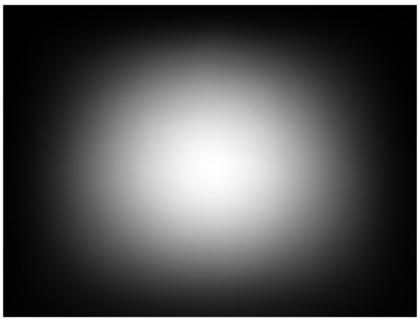


Figure 6.10: The MSRA10K [150] annotation map indicates the centre bias in this dataset. It shows that most salient objects are found towards the centre of the image.

object detection is a detection or segmentation process that is initiated by the detection of the salient objects in an image [5].

The COTS dataset is designed to facilitate the evaluation of saliency-based methods while also studying saliency detection results. The first part of the dataset consists of images that contain a single object together with the respective binary-image covering the object and serving as ground truth. This is similar to what one finds in the ECSSD [154] and MSRA10K [150] datasets. The first instance of every set in the second part of the dataset can also be used for this purpose. This instance contains the single object and its ground truth and can be therefore considered as an extension of the first part of the dataset. Saliency datasets are also prone to centre-bias. In order to demonstrate this, the average annotation map of the MSRA10K dataset is presented in Figure 6.10.

A selection of 8-bit and 16-bit depth maps of each incremental scene are also an integral part of the COTS dataset [10]. This introduces an opportunity for the investigate and potential link between saliency and depth information allowing for the validation of earlier work [155]. Important saliency datasets include the JuddDB [37], DUT-OMRON [156], Pascal-S [51], ECSSD [154] and MSRA10K [150]. These datasets are more commonly used to benchmark and evaluate saliency techniques. However, they do not include any depth maps to accompany the colour images. The COTS dataset addressed this gap and provides depth maps for further exploration of this research topic.

Benchmarking Saliency

In order to validate the COTS dataset, a benchmarking exercise was carried out based on the methodology and source code provided by Borji *et al.* [5]. This study investigated 41 saliency models and these were benchmarked extensively. This experiment was extended to demonstrate how the COTS dataset can also be used to benchmark saliency models. This extension of the original study [5] includes seven models that were selected based on the availability of the source-code within the work of Borji *et al.* [5]. This techniques included a selection of Fixation Prediction methods (SeR [157]) SIM [158] SR [159] COV [160]) and Salient Object Detection models (SEG [161] SWD [162] FES [163] CA [164]). Besides the above mentioned methods, the state of the art saliency detection technique Pyramid Feature Attention Network (PFAN) [165] was also included. In this experiment, the PFAN model gave an F_β of 0.957 on the COTS dataset as listed in Table 6.4 and a reported 0.931 on the ECSSD dataset [165].

The source-code provided with each technique was used for the comparative analysis of the selected techniques on COTS together with the ECSSD [154] and MSRA10K [150] datasets. In order to test the source-code, the results obtained by these seven techniques on the ECSSD [154] and MSRA10K [150] datasets were reproduced and confirmed that they match with those in the work of Borji *et al.* [5] and listed in Table 6.4. The reported comparison presents on the F_β metric on both Fixed Thresolds and Adaptive Threshold (IDAT). The β value was set to 0.3 so that more importance is given to precision rather than recall [10].

$$F_\beta = \frac{(1+\beta^2)\cdot \text{Precision}\cdot \text{Recall}}{\beta^2\cdot \text{Precision} + \text{Recall}} \tag{6.1}$$

The implementation of these techniques was verified in the first part of the experiment. Once that the performance of the models was confirmed by comparing them to the other models, the second part of the experiment introduced the

Table 6.4: The results of the selected saliency models on the COTS, MSRA10K and ECSSD datasets. This table presents the extension of the work by Borji *et al.* [5] by also including the current state of the art technique, the Pyramid Feature Attention Network (PFAN) [165]. This was evaluated on the COTS dataset with results illustrated in the last row.

	COTS		**ECSSD**		**PASCAL-S**		**DUT-OMRON**		**MSRA10K**	
	Fixed	**IDAT**	**Fixed**	**IDAT**	**Fixed**	**IDAT**	**Fixed**	**IDAT**	**Fixed**	**IDAT**
			Results from [5]							
FES [163]	0.812	0.692	0.655	0.645	0.619	0.605	0.520	0.555	0.717	0.753
SR [159]	0.676	0.507	0.385	0.381	0.447	0.442	0.298	0.363	0.473	0.569
SIM [158]	0.699	0.625	0.391	0.433	0.434	0.407	0.358	0.402	0.689	0.705
SWD [162]	0.785	0.702	0.624	0.549	0.577	0.523	0.478	0.506	0.498	0.585
CA [164]	0.766	0.587	0.515	0.494	0.489	0.472	0.435	0.458	0.621	0.679
COV [160]	0.628	0.541	0.641	0.677	0.589	0.604	0.486	0.579	0.667	0.755
SEG [161]	0.951	0.941	0.568	0.408	0.534	0.344	0.516	0.450	0.697	0.585
SeR [157]	0.722	0.488	0.419	0.391	0.433	0.406	0.385	0.411	0.542	0.607
			Results from [165]							
PFAN [165]	0.957	0.842	0.931	N/A	0.892	N/A	0.856	N/A	N/A	N/A

COTS dataset. The F_β statistic was calculated for the FES [163], SR [159], SIM [158], SWD [162], CA [164], COV [160], SEG [161] and SeR [157] models. Table 6.4 presents the results that show how the COTS dataset can also be used for the benchmarking of saliency models just as other datasets allow. This extended benchmarking enables further prospects for research by also considering the depth information available in the COTS dataset.

Saliency and Multiple Objects

The COTS dataset was also enriched with data related to the subject interaction in relation to every image as collected through the online test described in Section 6.4. The saliency-driven ranking technique presented in this thesis was evaluated using the COTS dataset. A detailed account of this evaluation process is presented in Chapter 7.

6.5.2 Inpainting

The removal of an object from a scene is referred to as inpainting. Different techniques were surveyed in Chapter 3. Techniques range from traditional methods

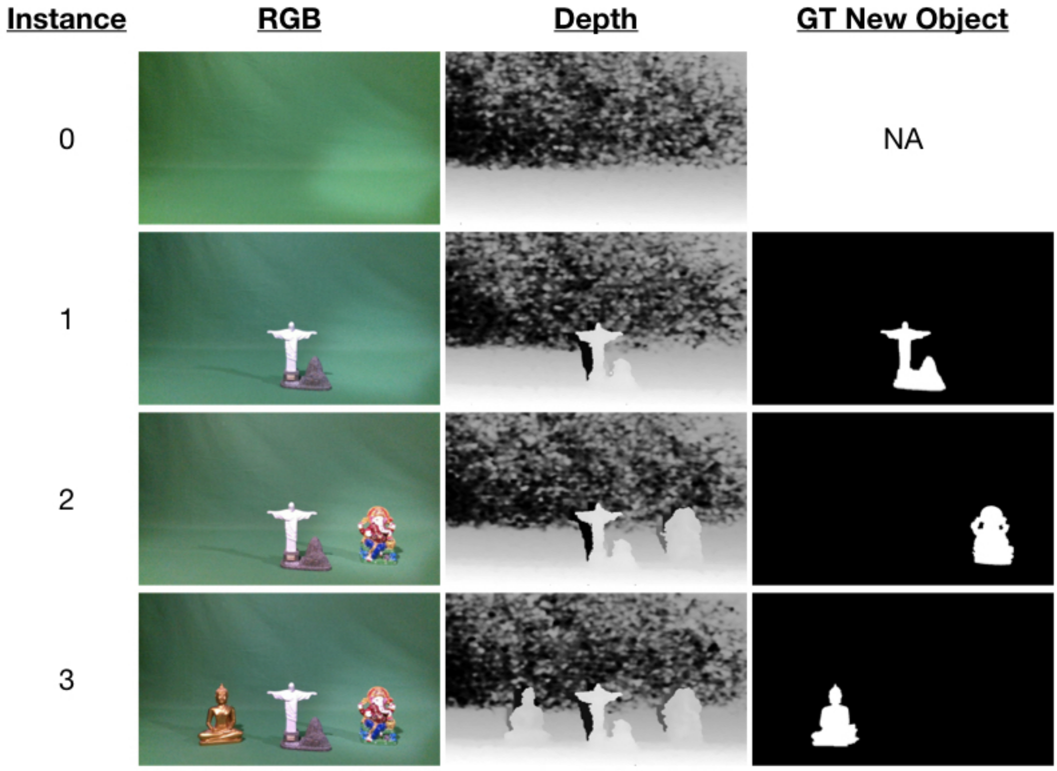


Figure 6.11: Incremental scenes are one of the distinctive features of the COTS dataset. These are ideal for the evaluation of inpainting techniques. Every row corresponds to an instance in the scene with a new object being introduced in every scene. Every instance includes an RGB image, an 8-bit depth map and a binary mask.

[113] [110] [166] to generative deep learning models such as GANs [167].

Inpainting situations where objects of larger sizes need to be removed, are generally evaluated using a Mean Opinion Score (MOS). Opinion scores are indicative in the evaluation of subject's perception, however, they do not provide objective insight into the efficacy of the inpainting approach being used [23].

The COTS dataset is designed to address this limitation. Figure 6.11 presents an example of how each of the 23 instance in the second part of the dataset is organised in instances. Every instance is an increment to the previous instance that has an object that was not present in the one before it. The new instance has a single new object included with no other modification in the image. This

Table 6.5: The results for each scene where the inpainting result was compared to S1 using the MSE metric. The maximum error related to the MSE reading when S2 was measured against S1, hence comparing the scene without the object with the scene including the target object.

		Mean Squared Error (MSE)			
	Occlusion	**Ours + Telea**	**Ours + NS**	**Deep Learning**	**Max Error**
Statues	Yes	369.10	452.39	455.79	1139.27
Shooters	Yes	57.20	68.17	72.11	83.09
Academic	Yes	384.76	488.48	484.78	1990.00
Footwear	No	58.64	69.12	124.73	1617.40
Mugs	No	79.31	101.61	108.91	407.76
Tech	No	112.46	153.91	142.79	570.52

set of consistent increments provide the desired setup for the objective evaluation of inpainting techniques. Inpainting techniques remove an object from a specific instance (n) and the instance before it ($n-1$) can be therefore used as ground truth as it will be missing the removed object and has the ideal background [10].

The ground truth for every new addition is provided with every instance and this allowed us to facilitate the evaluation of inpainting algorithms. The binary mask can be used during implementation to guide the inpainting method being evaluated. This reduces the need of creating a mask through segmentation method that can render any comparison challenging. A potential experimental setup that uses the COTS dataset is that uses our published framework [22] is presented in this section. A sample setup for the evaluation of inpainting is illustrated in Figure 6.13. This shows the target object to be inpainted as the red deodorant in Scene 2 denoted as S2. Inpainting using Telea's method [113] was carried out using the [22] framework. Scene 2 (S2) is the same as S1 but has the additional red deodorant. This allows S1 to be used as ground truth for the inpainting of the target object found in S2. The inpainting result is hence compared to S1 using a full-reference metric. In this experiment, the MSE metric was used.

Inpainting Experiment

The COTS dataset was designed to compare different inpainting methods. This section outlines an experiment to demonstrate how this can be achieved. Six scenes from the COTS dataset were chosen for this experiment and were split into two sets. The first set includes scenes in which some objects occlude others. The other set includes objects that do not cause any occlusion to the others. The results are presented in Table 6.5.

The evaluation process discussed above was applied to all six scenes. The S2 scenes are the ones containing an object for inpaiting being also represented by a binary mask. S1 is the instance of S2 without the object. This instance was a specific feature of the COTS dataset since it enables objective evalation. This framework was used to demonstrate how the COTS dataset can be used to evaluate three inpainting approaches. The first two inpainting methods evaluated are the ones presented in [22]. These are namely Teala's [113] method and the Bertalmio *et al.* [110] method. This experiment also included a GANs based method, namely NVIDIA's approach by Liu *et al.* [118].

The setup of this experiment included objects being placed in front of a plain green background. This resulted in particularly interesting cases. The background of the COTS dataset exposes different strengths and weaknesses of the inpainting methods being evaluated. Traditional dispersion based methods returned results that were blurry and this matched what was already reported in previous work [22]. The output of the GAN inpainting was more crisp when objects were occluded. Moreover, the result of the GAN inpainting when the target object was not occluded, hence only having a plain background, was comparable the traditional techniques. The experiment is presented visually in Figure 6.13.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(S1_i - IR_i)^2 \tag{6.2}$$

Equation 6.2 gives the MSE metric that was used to evaluate the output of the

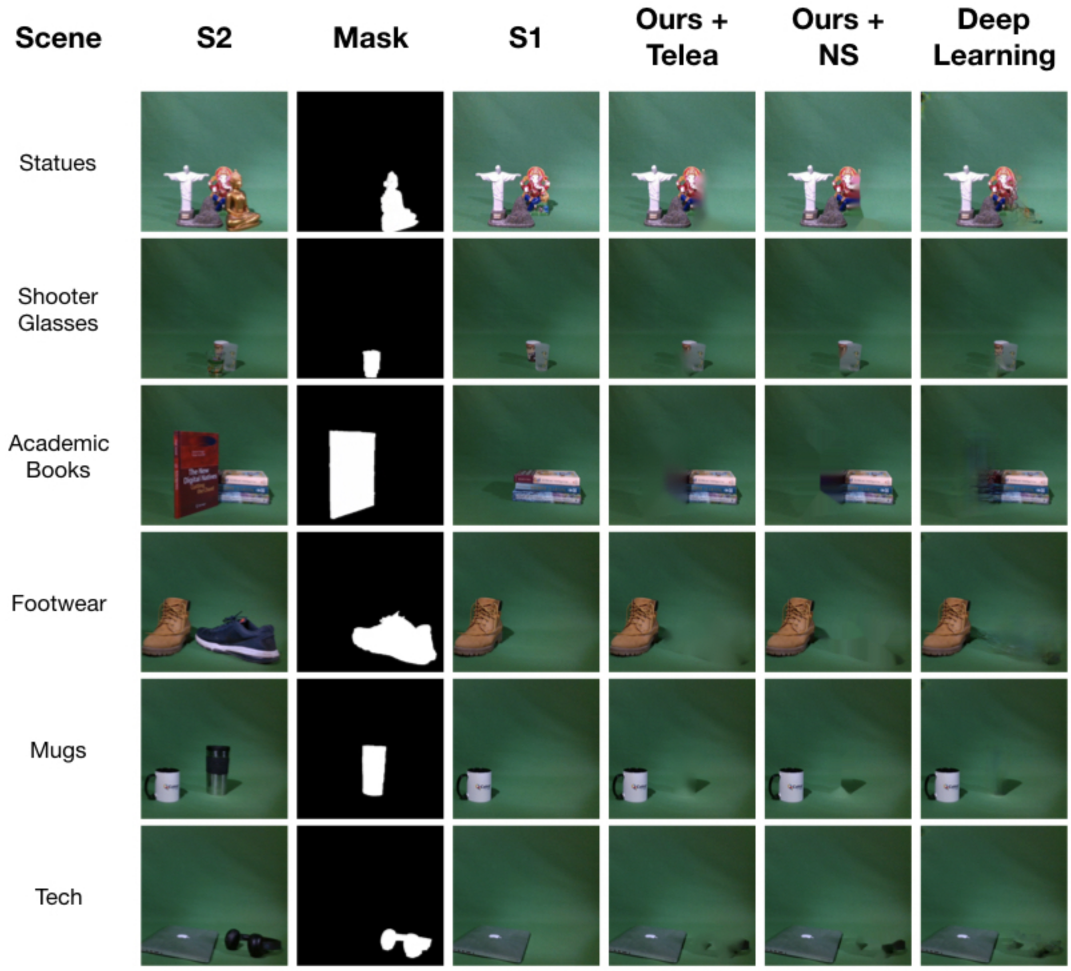


Figure 6.12: A presentation of the components in the inpainting experiment. This includes a comparative results of the objective evaluation. S2 is the original scene upon which the inpainting is carried out using the mask presented in the second column. S1 acts as the ground truth since it is the instance before S2, hence without the object that was inpainted. The results of the different inpainting techniques under evaluation are presented in the last three columns. These are namely our technique with Teala's [113] and Bertalmio *et al.* [110] and the NVIDIA deep learning method of [118] in the last column.

inpainted result IR for each of the three techniques when compared to the ground truth, S1. The comparison of the two original scenes, S1 and S2, with the target object as the only difference returns the maximum error. This is computed to put the comparative results for each of the other approaches into context. The instance without the target object, S1, is used to compare the and therefore the comparison

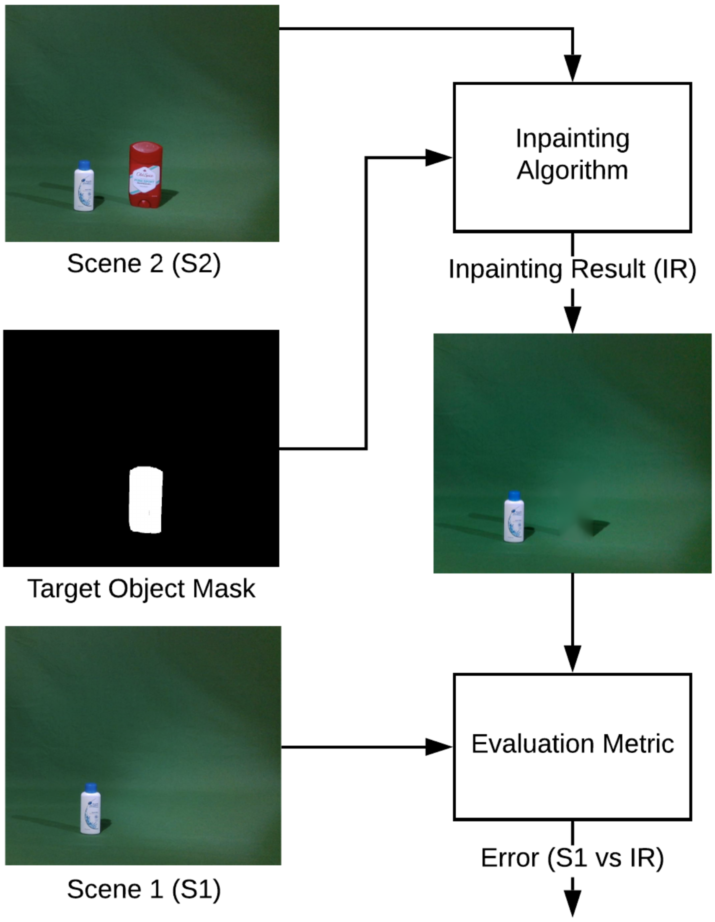


Figure 6.13: A high-level experimental pipeline for the evaluation of inpainting techniques presented in this section.

to S1 should be as far as possible from the maximum error. The results show that the Telea inpainting technique gives the best result when compared to the other inpainting methods [23].

Performance Evaluation

These techniques were also compared with respect to their time performance. The results of this experiment are in Table 6.6. The approach that we published in [22] using Telea and NS inpainting was implemented again in Python 3.6 and OpenCV

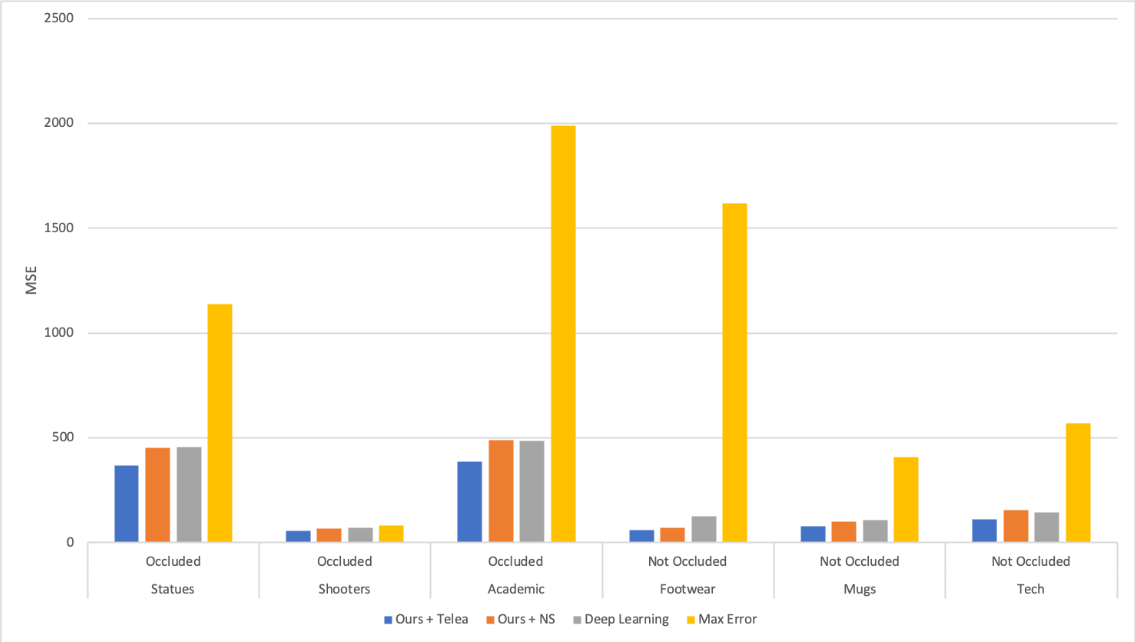


Figure 6.14: A bar graph of the comparative results. The y-axis relates to the MSE. The scenes are presented on the x-axis. The occluded scenes are presented first and followed by the other three. The Maximum error represents the error of the scene when S1 is compared with S2.

3.0. These exeperiments were run on a machine with a 2.6 GHz Intel Core i7 processor and 16 GB 2400 MHz DDR4 memory running a MacOSX Mojave 10.14.6 operating system. The deep learning technique by Liu *et al.* was accessed using the online instance provided by NVIDIA[3] and the performance evaluation and training information is directly based on the results of the same paper.

The results emerging from this experiment demonstrate that there is a relationship between the performance of this approach and the size of the target object being inpainted. The average time for every technique to compute the inpainting result was measured. Telea's method returned a result in 4.45s while NS took 3.52s. On the other hand, the deep learning approach returns the results significantly faster. However, this depends completely on the trained model. The paper specifies that a NVIDIA GPU V100 16 GB was used and training took between 3 to 10 days depending on the dataset. This means that if our technique is re-

[3] https://www.nvidia.com/research/inpainting/

	Time (Telea) [s]	**Time (NS)[s]**	**Time (DL) [s]**	**Target Object Size [%]**
Statues	3.63	2.83	0.03	6.4%
Shooters	1.51	1.06	0.03	2.0%
Academic Books	8.67	6.94	0.03	20.5%
Footwear	7.00	5.65	0.03	14.7%
Mugs	3.20	2.49	0.03	5.4%
Tech	2.70	2.20	0.03	3.9%
Average	4.86	3.53	0.03	8.8%
Processor	CPU	CPU	GPU V100	-
Training	NA	NA	3 - 10 Days	-

Table 6.6: Time analysis of the inpainting techniques in relation to the percentage area of the object to the entire scene. The information related to the deep learning technique is as specified by Liu *et al.* in [118]. While Telea's [113] and NS [110] did not require any pre-processing, the deep learning method [118] needed between 3 to 10 days of training on an NVIDIA GPU V100.

designed to exploit parallelism and run on a GPU architecture, it can potentially improve in terms in performance while still not requiring any effort in training and therefore can be used in different situations. This collection of comparative results indicates that the choice of technique should depend on the priority between time, pre-processing or training and the quality of results as demonstrated in this section.

Region Growing

The binary masks are a very important component of the techniques presented in this thesis. Masks can be either generated using segmentation techniques [92] [3] [95] as presented in Section 3.2 or using hand-crafted annotations as discussed above. The challenge with such techniques is that there is no guarantee that the mask of an object corresponds precisely to every pixel that represents the object in the colour image.

In order to mitigate this discrepancy, region growing or dilation is applied to to the mask. In each region growing or dilation pass, all eight neighbours of a foreground pixel are set to a value of 255 and they are therefore also assigned as foreground pixels. This operation increases the size of the mask and compensates for any discrepancy between the outline of the mask and the object.

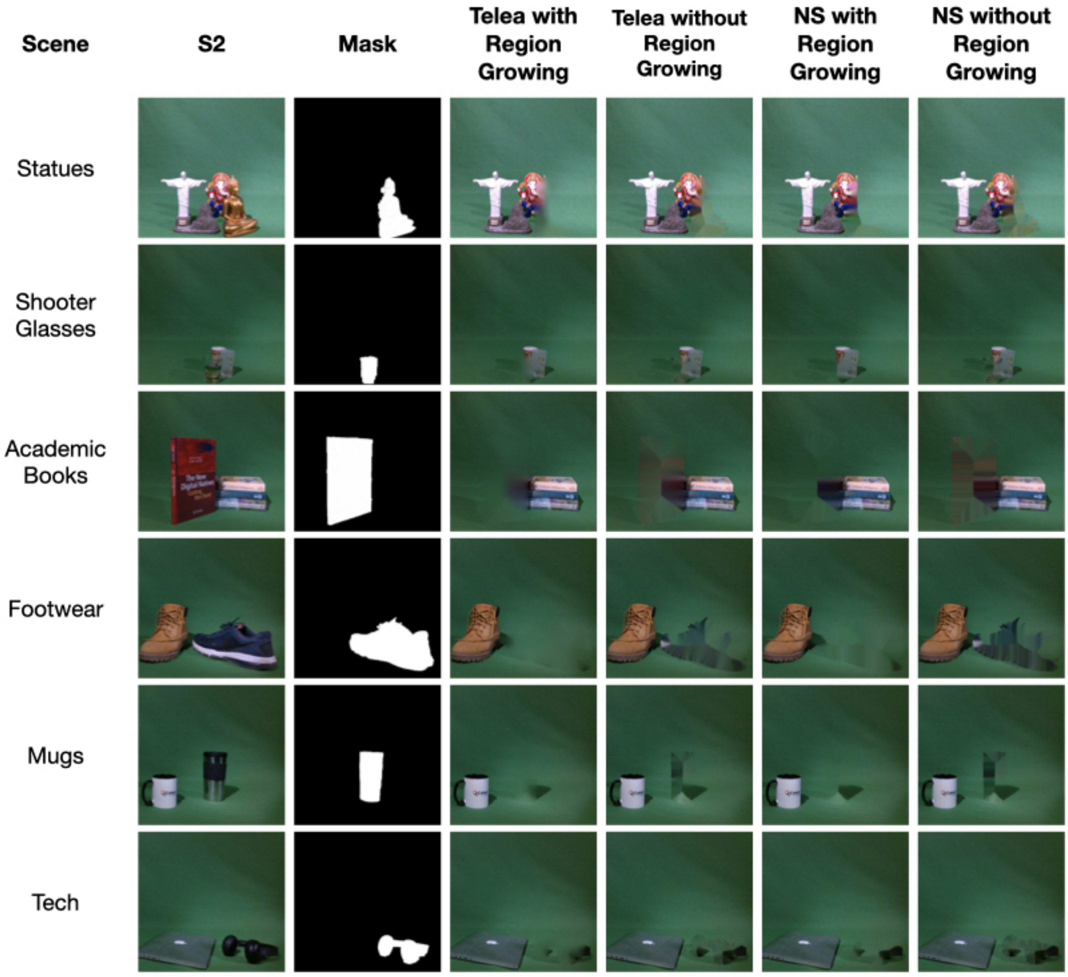


Figure 6.15: A visual representation of with Teala's [113] and Bertalmio *et al.* [110] inpainting techniques with or without using Region Growing.

Figure 6.15 presents results that demonstrate the effect of applying region growing to a mask that is used in inpainting. The experiment presented above was executed again for comparative purposes and region growing was not applied on the mask. Figure 6.15 demonstrates the visual results of this experiment. When region growing was not applied on the mask, inpainting performed poorly since it also used pixels belonging to the object to fill the space of its removal. This left traces of the object spread along its entire area. This was also objectively measured

Table 6.7: The results from the computation of the MSE of the inpainting techniques with or without region growing (RG).

		Mean Squared Error (MSE)				
	Occulusion	**Telea with RG**	**Telea without RG**	**NS with RG**	**NS without RG**	**Max Error**
Statues	Yes	369.10	427.50	452.39	460.67	1139.27
Shooters	Yes	57.20	64.31	68.17	60.43	83.09
Academic	Yes	384.76	592.35	488.48	644.65	1990.00
Footwear	No	58.64	239.06	69.12	351.48	1617.40
Mugs	No	79.31	153.28	101.61	190.01	407.76
Tech	No	112.46	203.81	153.91	229.80	570.52

by calculating the MSE of the inpainting procedure without region growing. Table 6.7 shows how the error in the images in which region growing was not applied was greater than the ones which used it. The maximum error is returned when the S2 is compared to S1. The maximum error is also presented here for comparison purposes.

6.5.3 Content Blending

Blending, or addition of an object, is the inverse process of inpainting. For this reason, the COTS dataset can also be used to evaluate such approaches. An experiment was designed to demonstrate how the COTS dataset can be used for this purpose and its process is illustrated in Figure 6.16. The experiment starts by using a binary image mask on S2 to identify the target object. This time, the target object will be included into S1. The object can be segmented using any segmentation technique or using depth information [22]. The blending result combines the newly extracted object onto S1. For the COTS dataset to be use together with a full-reference metric, the blended object needs to be placed in the same coordinates from which it was extracted. The MSE metric presented in Equation 6.2 can also be used to evaluate this method. The comparison experiment for blending includes S2 that will act as ground truth for the Blending Result BR. With its inclusion of shadows, the COTS dataset allows for their observations during this process. These shadows nonetheless make scenes more realistic and allow for fair evaluation of blending algorithms. Moreover, more advanced blending techniques that attempt to create the shadows can also be evaluated using the COTS dataset [10]. One can consider modern deep learning models [168] that detect shadows of objects and that can also facilitate this process.

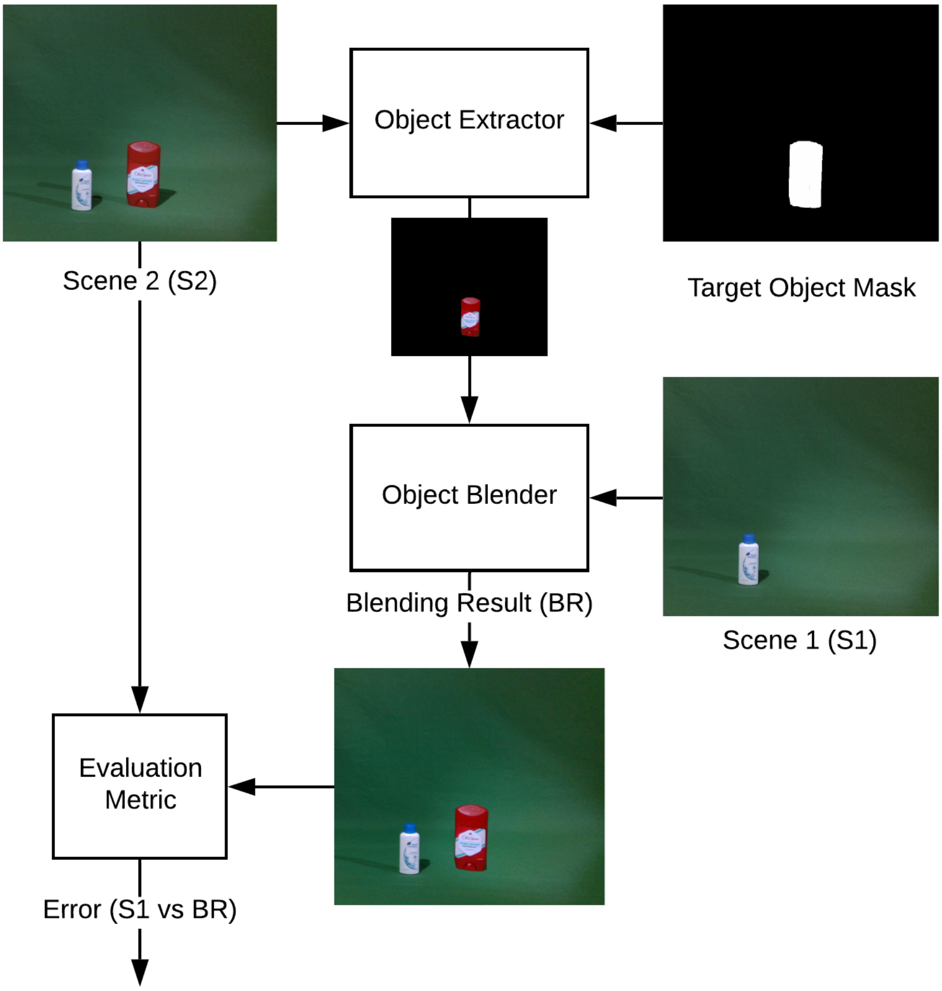


Figure 6.16: A sample experimental setup that uses the COTS dataset for the evaluation of blending techniques. An object is extracted from S2 and is blended into S1. The experiment is concluded when the blended result (BR) is compared to S2.

This experiment was implemented using the same set of scenes from COTS as the inpainting experiment found in Section 6.5.2. The visual results are presented in Figure 6.17. The column on the left shows instances of S1 from different scenes that also serve as the target image on which the object will be blended. The second column presents the mask of the object guiding the extraction of the target object from S2. S2 serves as ground truth following the object blending. The last column shows the results of the blending process. The objective results using the MSE metric to compare the blending result with S2 can be found in Table 6.8.

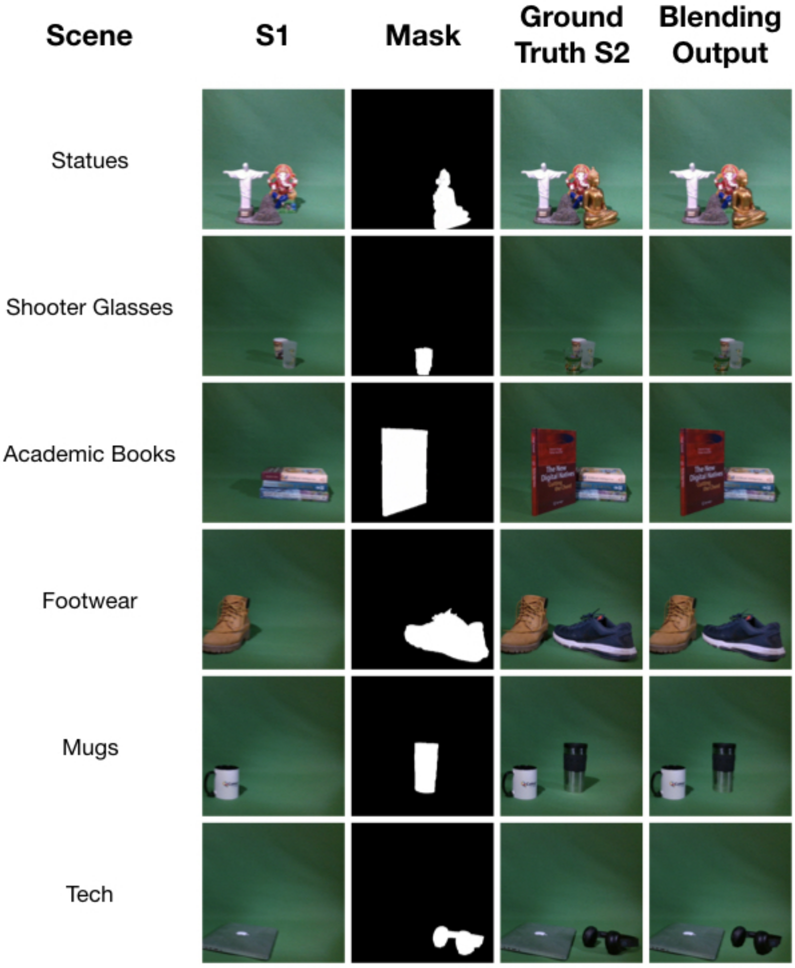


Figure 6.17: A set of visual results from the evaluation of blending techniques using the COTS dataset. The target object is extracted from S2 and blended onto S1 resulting in the blending output. In this case, S2 serves as ground truth.

Table 6.8: The comparative results following the calculation of the MSE metric when comparing the blending result (BR) is compared to S2. The maximum error returned when the S2 is compared to S1.

		Mean Squared Error (MSE)	
	Occlusion	**Error (BR vs S2)**	**Max Error (S1 vs S2)**
Statues	Yes	135.25	2664.17
Shooters	Yes	26.19	272.51
Academic	Yes	141.87	2177.99
Footwear	No	63.96	3444.17
Mugs	No	72.72	461.99
Tech	No	94.54	163.98

6.5.4 Combined Usage

This section demonstrated how the COTS dataset can be used to evaluate different applications of computer vision. Each experiment was presented in a modular fashion and therefore can be used to evaluate pipelined applications in more complex frameworks.

6.6 Conclusion

This chapter introduced the novel COTS dataset, containing 120 images accompanied by depth maps and binary ground truth images. The sets are organised in instances where each is also has a corresponding CSV file that contains the click coordinates collected from the online test in 1267 participants took part. This dataset can be used in the evaluation of different computer vision applications that span from segmentation, inpainting and blending to saliency. Such a dataset also faciltates the evaluation of pipelined computer vision applications by making use of a single dataset.

The COTS dataset was made available for free and published in an open access publication [10] for easier dissamination and use. The first version of the COTS dataset is focused on a plain green background. This was originally intended to allow for chroma-key background replacement and therefore increase the variety and complexity of the data.

6.6.1 Contributions Summary

The main contributions of this chapter are the following:

1. Design and build a multipurpose RGB-D dataset for different computer vision applications;

2. Collection of interaction data from 1267 participants on the COTS dataset showing what participants perceived as more salient objects;

3. Demonstration of the efficacy of the dataset through different benchmarking experiments;

4. Demonstration of a novel objective approach for evaluating inpainting techniques;

5. Demonstration of how the novel objective approach for evaluating inpainting can also be applied to object blending.

7. Evaluation

> Measure what is measurable, and
> make measurable what is not so.
>
> Galileo Galilei

7.1 Introduction

This chapter evaluates different aspects of the saliency-driven ranking approach presented in Chapter 5. The main decisions behind the developed algorithm are evaluated in this chapter and are organised as follows:

Ablation Study of Centre Bias The saliency-driven ranking score of our technique can consider the centre bias. An ablation study of this score was carried out to evaluate the effect of centre bias on the overall result.

Benchmarking with other saliency detection techniques The proposed technique is benchmarked against other techniques using the MSRA10K [150] dataset. This verifies whether the proposed technique performs as good as others on this extensive dataset that contains single objects.

Comparison with Human Behaviour The ultimate validation of the saliency-driven ranking technique is its match with human behaviour. This section compares in detail the results of the proposed technique with the results gathered by participants on an online test.

Performance of Saliency Ranking This section presents the performance results of the saliency-driven ranking algorithm.

Configuring Saliency Ranking with Segmentation This section presents the experiment that was carried out to fine-tune the configuration of our saliency ranking model to rank the output of segmentation techniques.

Comparative Study with Attention Shift Ranks This section compares the output of our technique with the current state of the art. This includes a quantitative comparative study between methods together with a qualitative visual evaluation of the results.

7.2 Benchmarking

A variety of saliency benchmarking experiments [5] [42] [169] [150] provide a comparable presentation of this variety of techniques that provide a saliency map. These studies investigate different aspects and biases of saliency detection techniques that would influence the resultant saliency map. Furthermore, they shed light upon influences such as centre-bias and scene complexity that would influence the training of supervised models that allow for the generation of saliency maps. The use of these datasets containing mainly single objects are the main limitation of the current research of saliency detection [7] [39] but are nonetheless the datasets used to benchmark saliency related techniques.

The saliency-driven object ranking technique developed in this thesis aims to guarantee the same quality of results when benchmarked with traditional methods, without the need of any training, while being independent of any of the above mentioned saliency generation methods or approaches. All the techniques discussed above do not provide a ranking of which region in an image is more (or less) salient than others.

The hypothesis evaluated in this section investigates how the top 50% grid segments ranked by the proposed technique compared to the ground truth of the MSRA10K [150] dataset. This dataset was chosen due to its popularity and the large number of images it contains when compared to other datasets of its sort.

7.2.1 Performance Measurements

The precision and recall are critical quantitative metrics that indicate the performance of the system when compared to the ground truth of the MSRA10K. The recall indicates how much of the ground truth was detected by the model. On the other hand, precision indicates how many of the top 50% segments successfully matched the ground truth. For this reason, the F_β-Measure, denoted in Equation 6.1 was used as a supporting metric so that a combined level of importance is given to both precision and recall.

The non-negative weight β^2 is used to prioritise either precision or recall. When β is between 0 and 1, more weight is given to precision, and any value above 1 would give more weight to recall. In this work, it was important to increase the importance of precision since a high score of recall can, for example, be achieved by setting the majority of the image as salient or foreground. This weight was set to 0.3 as used by a variety of benchmarking studies [5] [169] [170] [39] that are based on the original findings of Achanta [171].

7.2.2 Ablation Study of Centre-Bias

This section presents an ablation study on the Centre-Bias. As presented in Figure 7.1, the MSRA10K tends to be influenced by centre-bias. In order to investigate the effect of this property on the proposed model, the same experiment was repeated without including the centre-bias in the generation of the model. The results are presented in Table 7.1, and they show that the average F-Measure along all the

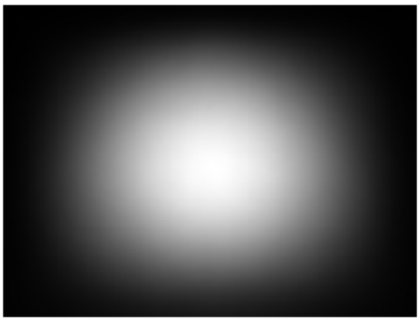


Figure 7.1: The average annotation map of the MSRA10K dataset produced by [5] showing its centre-bias.

	With Centre-Bias			Without Centre-Bias		
	Precision	Recall	F_β	Precision	Recall	F_β
Mean	0.84	0.83	0.83	0.81	0.81	0.80
Std Dev	0.15	0.15	0.14	0.16	0.16	0.15

Table 7.1: A comparative performance of the system when the system has Centre-Bias enabled or disabled.

images of the MSRA10K is only changed by 0.03 when the centre-bias is disabled. These results also follow the experimental presentation used by [1].

7.2.3 Comparative Results

The F_β statistic was used to compare the performance of this algorithm with other techniques as benchmarked in [5]. The proposed technique was compared against the worst (IT [50], SR [172] and SIM [173]) and best (ST [174] , QCt [175] and DRFI [176]) performing techniques as presented in Table 7.2. This shows that the proposed algorithm compares well with the best techniques.

	IT	**SR**	**SIM**	**ST**	**QCt**	**DRFI**	*Ours*
F_β	0.471	0.473	0.498	0.868	0.874	0.881	.828

Table 7.2: A comparison on the F_β statistic.

The results were also analysed visually against a sample from the MSRA10K dataset. The performance measurements presented in Section 7.2.1, assume a complete bi-level representation of the ground truth, reasonably matching its corre-

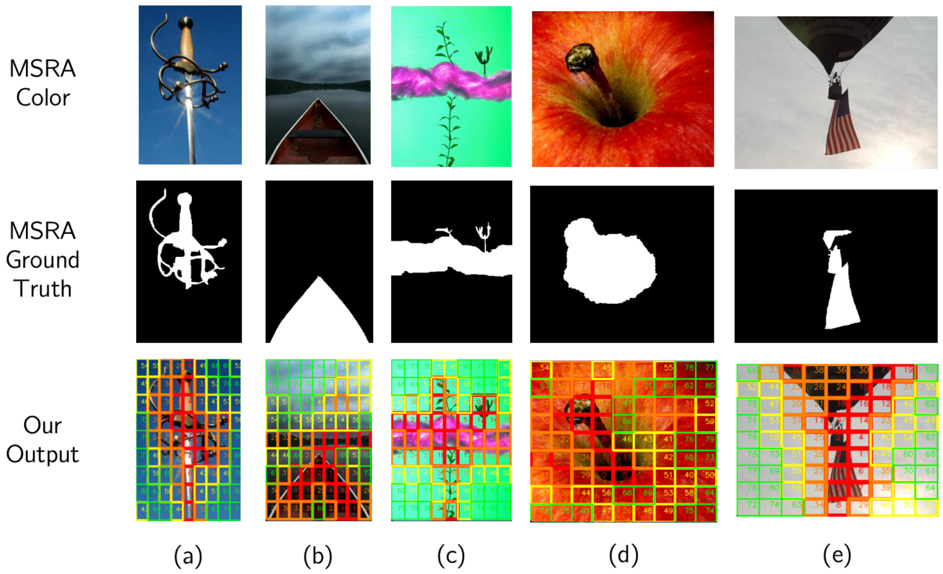


Figure 7.2: A selection of five images from the MSRA10K dataset showing how the ground truth does not accurately cover the salient region. The bottom row presents the result of the proposed technique that shows how the regions of interest are adequately detected and labeled.

sponding colour image. Figure 7.2 shows that there are cases within the MSRA10K dataset where the ground truth does not fully match the corresponding colour image such as case (a), (c) and (e) presented in the figure. In these cases, there are significant parts of the object in the image that are not marked as ground truth. For instance, in case (a), one can see that while the handle and the guard of the sword are marked as ground truth, more than half of the blade is left uncovered. Figure 7.2 also presents the results of our solution that successfully captures the entire objects. Although this is desirable, it negatively impacts the F_β score since the correctly detected segments of for example the blade of the sword in case (a), are considered by the metric as a false positive since the same parts of the blade are not covered by the ground truth.

7.3 Comparison with Human Behaviour

7.3.1 Online Test

While the other experiments showed that the proposed technique performs well on ranking saliency, it was also important to evaluate how it compares to human behaviour. Thus, a website was set up to facilitate dissemination and availability [20]. This tool allowed for a very detailed data collection that provided the required information for the comparison presented in this chapter. A website was specifically designed and built for this experiment. This presented the images to the participants through HTML and collected usage information through JavaScript. The collected data was stored in a hosted database. The usage data was collected in the background to minimise the participant interaction and keep the experience as focused as possible. This data included information about the session together with the operating system and browser being used by the participant.

Two separate online tests were carried out.

1. (987 Participants) In the first test we provided the participants with a selection of RGB-D datasets containing actual real-life scenes where no explicit object is prominent. This test aimed to evaluate the system using natural scenes. The datasets used in this test are Microsoft's *Ballet* and *Breakdance* sequence [135], Nagoya University's *Balloons* sequence [136], Fu *et al*'s *Bear* sequence [177] and Tsukuba's University stereo dataset [178]. The main results of this section are based on this test.

2. (1267 Participants) The second test used the COTS dataset and it aimed to investigate how attention on a scene changes when a new item is introduced. This test took advantage of the incremental design of the COTS dataset and the detailed metholodogy of this test to minimise visual bias is outlined in Section 6.4.

A set of images from the selected datasets were presented to the participant and for each image, the participants were asked the following question:

> *Click/Tap on the point that attracts your attention when you first see the image. The point can be anywhere and includes persons or other objects.*

The tap or click coordinates on each presented image were recorded. These coordinates were essential for the generation of the heatmap. The resultant heatmaps of the regions of interest in the images are presented in Figure 7.3. The time between the presentation of the image to the participant and the click or tap was recorded for every image. This allowed for a better understanding of the time taken by participants to choose an object from the scene while also making it easier to clean the data.

This experiment needed widespread distribution to maximise the number of participants. This was successfully achieved by engaging a social media campaign to promote the experiment URL. This was positively received with a total of 987 participants. Subsequently, the data was extracted from the online database and cleaned by the *time to click* variable. Following an inspection of the collected data, the records with time values less than 0.1s and more than 50s were identified as outliers and discarded in order to fairly represent adequate use of the test. This left 854 records for evaluation. The average time for participants to click on an object of their choice was 4.3s. The fastest participant clicked the image after 0.48s and the slowest one took 41.3s. The average x-coordinate of the clicks took place at 53% of the image width, and the average y-coordinate of the clicks took place at 51.1% of the height. This demonstrates how the subject clicks or taps reflect the effect and relevance of centre-bias [20].

7.3.2 Comparing the Algorithm against participant Clicks

The results of the saliency-driven object ranking technique presented in this thesis were studied in relation to the data collected during the online test. This part of the evaluation process compares the prioritisation of the salient regions of the images detected by the algorithm to where the participants clicked during the two distinct online tests. Such a test was important since the use of such a technique would be based on the notion of having an automatic technique that closely resembles human behaviour.

The online test provided valuable data as discussed above. This data was processed using a tailored algorithm that converted the clicks extracted from the online test to a format that is comparable to the output produced by the proposed saliency-driven ranking technique. This data processing algorithm went through the click coordinates of every participant on every image. It then returned an output image with grid-segments of a 9×9 resolution identical to that used by the algorithm.

Every click coordinate (x, y) had to be mapped with a grid segment with index (i, j). However, each image has its own width w and height h pixels, so the area covered by each grid varies according to the image size. This follows that x falls in the range $x = [0, w)$ and y in the range $y = [0, h)$ [20]. The segment $S(i, j)$ follows Equation (7.1) where D is the Segment Dimension, that gives the index of the respective cell as presented in Equation (7.2).

$$S(i,j) = \left\{ (x,y) \;\middle|\; \begin{array}{l} i\frac{w}{D} \leq x < (i+1)\frac{w}{D}, \\ j\frac{h}{D} \leq y < (j+1)\frac{h}{D} \end{array} \right\} \tag{7.1}$$

$$i = \left\lfloor \frac{xD}{w} \right\rfloor, \; j = \left\lfloor \frac{yD}{h} \right\rfloor \tag{7.2}$$

A systematic procedure was followed for each evaluated scene. This provided a fair comparison between the the participant clicks and the result of the proposed saliency ranking algorithm. The objects clicked during the online test were identified and recorded. The output of the algorithm was compared with the selection of objects done by participants. For every object in the list, the number of labeled segments, colour-coded red, orange, yellow, and green, were individually counted in the output generated by the proposed technique, and the segment-based

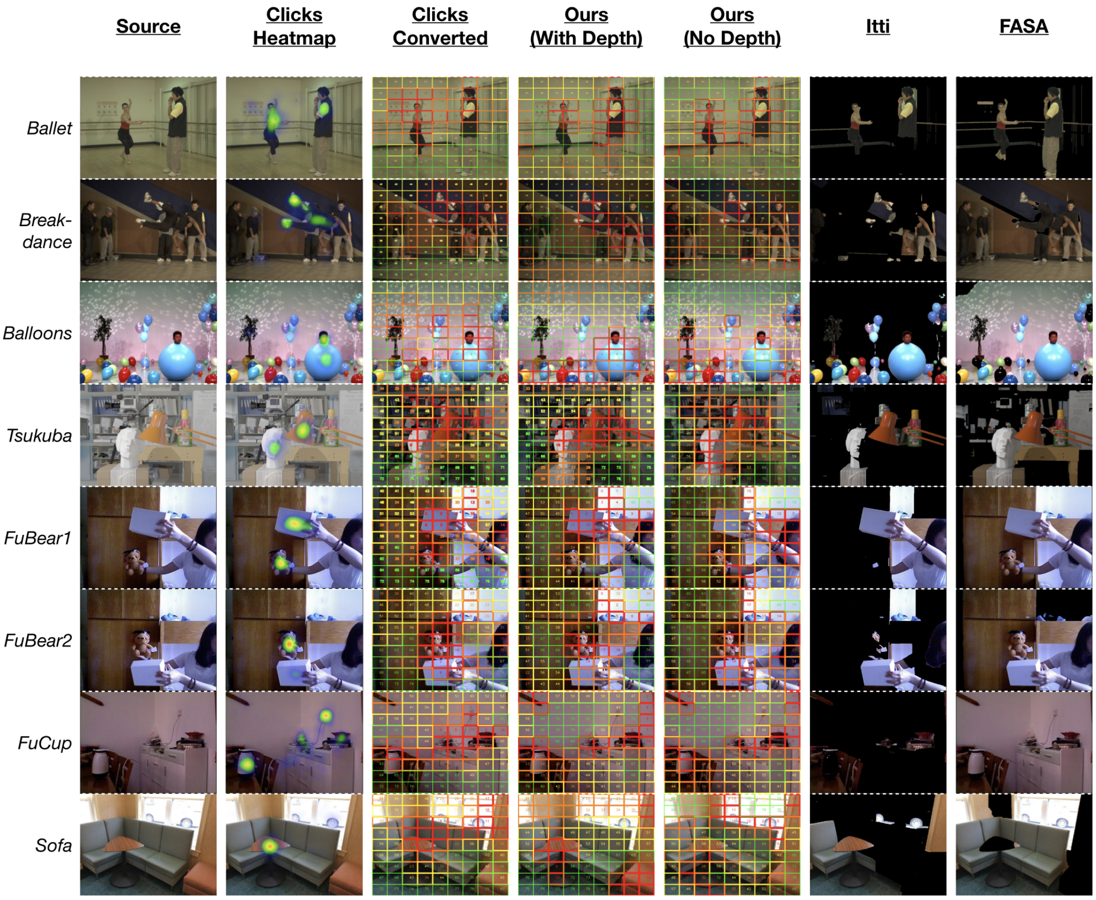


Figure 7.3: A visual comparative illustration demonstrating how the saliency-driven technique matches with clicks and taps of subjects and other techniques.

representation of the participant clicks.

The χ^2-test was used to determine whether there is a relationship between the grid-segments generated by the saliency-driven ranking algorithm and their equivalent representation of the clicks collected from the online test. Below follows the hypothesis that was selected for evaluation:

$Hypothesis H_0$:

> There is no significant difference between the algorithm ranking result and the participant clicks

When the null-hypothesis H_0 is not rejected, the predictions of the algorithm were most of the time similar to segment selected by the participants. The results of the χ^2-test with the technique is using a depth map are presented in Table 7.3.2. The null-hypothesis H_0 was rejected name in two items: the first in the *Balloons* dataset and the second case was the man in the *Ballet* dataset. While in the *Balloons* sequence, the algorithm discarded the plant and the centre balloons, the case of the man in the *Ballet* sequence deserves further investigation. The man covers 25% of the image, which is a considerable area of the image when considering and comparing to the number of pixels in the image where a participant can potentially tap or click. The saliency-driven technique successfully detected the man from top to bottom while, conversely, participants only interacted on the man's vest and head. This affected the result since when all the algorithm results of segments representing the man were counted and tested against the participant clicks, H_0 was rejected. However, considering such a scenario, one can safely consider that the algorithm correctly predicted the selection of the man. When also considering that this technique is pipelined with object-segmentation methods, such a case would also match to a correct prediction since the same label would be selected. All this results in the technique correctly detected 91.3% of the objects [20].

A closer look at the results, presented in Figure 7.3, also shows that participants were sometimes clicking on regions that would typically be classified as background. One of the most evident cases is the *Sofa* scene. The clicks heatmap and 'clicks converted' images in Figure 7.3 show that a significant number of participants were clicking on the wheel of the van outside the window of the same scene. It is interesting to note that this was not captured by the proposed technique when the depth score was used. However, when the depth score was not used, the same regions were also detected by the system. A similar effect was also manifested in the *FuCup* scene where participants were clicking on objects that were placed on the cabinet in the background and even on the sword hung in the top left corner of the scene. These would be generally disregarded as background and therefore ruled out from being a salient object, yet this experiment shows that a significant number of participants perceived these same objects as salient objects. This technique was also successful from this perspective since it accurately reflected what participants were choosing in scenes that contained multiple objects.

Another possible approach of evaluating saliency ranking is using a heuristic such as the brightest 10% pixels of a saliency map as proposed by Judd *et al.* and reviewed in Section 2.4.2. The technique presented in this thesis avoids this heuristic and instead of filtering individual pixels by brightness, it takes into consideration the context of the neighbourhood. This avoids situations where, for example, there would be a very small number of bright pixels in a region that does not contain a salient object and it therefore gives more weight to other bright pixels that are more probable to represent a salient object.

Dataset	Object	χ^2	p-value	H_0 rejected?
Ballet	Man	10.095	0.0178	Yes
Ballet	Ballerina	2.9706	0.3962	No
Ballet	Poster	1.5	0.6823	No
Balloons	Man	3.8294	0.2805	No
Balloons	Center Ballons	6.5714	0.0468	Yes
Balloons	Plant	8.0563	0.0448	Yes
Tsukuba	Lamp	3.3333	0.3430	No
Tsukuba	Head	1.1778	0.7583	No
FuBear1	Bear	3.2667	0.3523	No
FuBear1	Box	1.1818	0.7574	No
FuBear1	Woman	1.3518	0.7169	No
FuBear2	Bear	0.4	0.9402	No
FuBear2	Box	6.2667	0.0993	No
FuBear2	Woman	0.6421	0.8867	No
Sofa	Table	3.4961	0.3213	No
Breakdance	Dancer	2.5	0.4753	No
Breakdance	Radio	1.0667	0.7851	No
Breakdance	Right Man	0.7222	0.8680	No
Breakdance	Left Man	1.9762	0.5774	No
FuCup	Kettle	4.6095	0.2027	No
FuCup	Wall Socket	1.6667	0.6444	No
FuCup	Right Objects	0.66667	0.881	No
FuCup	Center Objects	1.1667	0.761	No

Table 7.3: Table showing the χ^2-square statistic and the respective p-value for the test of hypothesis H_0 for each object in the respective scenes.

While the first test extensively reported above covers real-life situations with datasets containing natural scenes, the second test focused on the incremental nature of the COTS dataset. The participant clicks on different images can also be compared with the output of the saliency ranking algorithm. Figure 7.4 presents an example of the results of our technique compared to participant clicks on the COTS dataset. In all images it is clear that the results of the algorithm match the participant clicks. From a ranking perspective, the top-ranked segment by the algorithm is only a segment away from the participant interactions. The same object is represented by both of these segments. When the objects where small, such as the last row example, the ranking is targeting different objects but in practice, the indicative segment is only a segment away. This is another motivation for further future research on the analysis of the ranking output. The saliency ranking algorithm also shows to be sensitive to other implicit features in the scene that also affected human participants. For example in the scene featured in the last row, there is a region on the left where participants were clicking, probably because of a change in shade that caught their attention. In this case, the algorithm also marked the corresponding segments as potentially salient with a ranking that is considerably close to that of the participant clicks.

7.4 Performance Measurements

The computational complexity of the proposed solution is directly related to the size of the input image. The traversal of each pixel in each image is required, and this results in $2\mathcal{O}(N) \Rightarrow \mathcal{O}(N)$ where N is the number of pixels in the image. This also means that the segment dimension does not affect the complexity of the algorithm. Thus, the proposed technique provides more flexibility in applying it to a variety of applications.

The architecture presented in Section 5.3 was implemented in Python 3.6 and OpenCV 3.0 to generate the results presented in this section. Performance testing

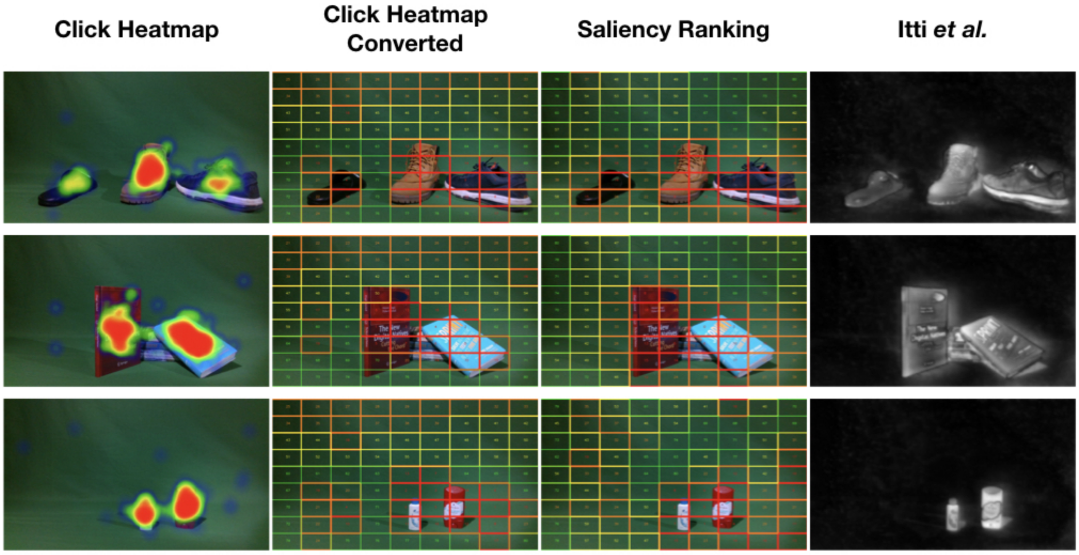


Figure 7.4: Visual representation of our saliency-driven ranking algorithm compared to participant clicks and Itti's [50] saliency detection using the COTS dataset in Test 2. The column on the left presents a the heatmaps of clicks from the online test. The second from the left column illustrates grid layout representation of the clicks such that they can be compared to our saliency-driven ranking method. The column on the right shows the same scene from the COTS dataset processed with Itti's technique [50].

Dataset	t(Depth)	t(¬Depth)	t(Itti)	t(FASA)
Ballet	0.1498	0.0530	2.4306	4.7573
Balloons	0.1264	0.0590	2.0461	4.1101
Breakdance	0.1537	0.0568	17.9918	4.8980
FuBear1	0.1086	0.0383	1.9396	1.6093
FuBear2	0.0911	0.0387	1.7726	1.6805
FuCup	0.0893	0.0327	2.1946	1.5265
Table	0.0940	0.0420	2.7215	2.8111
Sofa	0.1086	0.0390	3.1986	1.5280
Tsukuba	0.1141	0.0417	1.0810	1.5082

Table 7.4: The time results for the algorithms to execute the detection. The scene being processed is presented in the first column. This is followed by the results in seconds of the proposed algorithm with depth score enabled and disabled, respectively. The last two columns present the time taken in seconds for Itti's [50] and FASA's [41] algorithm to carry out similar result.

was carried out on a machine with a 2.6 GHz Intel Core i5 processor and 8 GB 1600 MHz DDR3 memory running a MacOSX High Sierra n v0.13.5 operating system.

An instance was also run without the depth information to understand its impact. The closest similar techniques found in literature use other saliency techniques and generate segmentation of the most salient objects using a binarised map of each technique [20]. Run-time performance of the proposed technique was carried out and the results are presented in Table 7.4.

7.5 Configuring Saliency Ranking with Segmentation

This section evaluates the results that our saliency-driven ranking technique gives when combined with object segmentation techniques. The following subsections cover the metrics that enable such evaluation together with the experiments that led to the configuration of our method to rank segmented objects.

7.5.1 Metrics for Ranking Salient Objects

The research area of saliency ranking is still in its infancy and there is still no agreement on a universal metric to measure performance within this problem [15]. For this reason, to evaluate the work presented in this thesis we use the metrics that were also used in recent comparable work [16] [1] [15] and also introduce a new metric.

Salient Object Ranking (SOR)

The Salient Object Ranking (SOR) is based on Spearman's Rank-Order correlation originally proposed in this context by [16]. Spearman's rank-order correlation ρ is a non-parametric measure of the dependence between the rank and the order of two distinct variables. Spearman's correlation is preferred over Pearson's correlation since the former can detect a correlation irrespective of whether the relationship between variables is linear or not. This is known as a monotonic relationship. Spearman's coefficient has the high value of 1 when a variable has a perfect monotonic relationship to another. When on the other hand, variables have contrasting ranks, Spearman's coefficient is at its minimum value of -1. The SOR returns a value for the rank between salient objects with a rank in the range $[-1, 1]$ [16]. In their recent publication [15], the researchers behind this metric presented three variants for the SOR namely based on the average, power and maximum as presented in Equation 7.3.

$$\text{Rank} = \begin{cases} \text{SOR}_{\text{avg}}(\mathcal{S}(\delta)) = & \frac{\sum_{i=1}^{\rho_\delta} \delta(x_i, y_i)}{\rho_\delta} \\ \text{SOR}_{\text{pow}}(\mathcal{S}(\delta); \alpha) = & \frac{\sum_{i=1}^{\rho_\delta} \delta(x_i, y_i)}{\rho_\delta^\alpha} \\ \text{SOR}_{\text{max}}(\mathcal{S}(\delta)) = & \max(\delta(x_i, y_i)) \end{cases} \tag{7.3}$$

where δ represents a particular instance of the predicted saliency map $\mathcal{S}$, the power is denoted by α, ρ_δ denotes the total number of pixels δ contains, and $\delta(x_i, y_i)$ refers to saliency score for the pixel (x_i, y_i).

The most popular variant of the SOR is the original one based on the average [16] and was also used to evaluate the work in [1]. This metric is presented in Equation 7.4.

$$\text{Rank}(\mathcal{S}_m^T(\delta)) = \frac{\sum_{i=1}^{\rho_\delta} \delta(x_i, y_i)}{\rho_\delta} \tag{7.4}$$

where δ represents an instance of the predicted saliency map $\mathcal{S}_m^T(\delta)$ generated using technique T, ρ_δ denotes the number of pixels in δ, and $\delta(x_i, y_i)$ refers to saliency score for the pixel (x_i, y_i).

Since the SOR is based on Spearman's coefficient, it does not cater for situations where there are no common objects between rank variables such as the case where the method detects salient objects that are not present in the ground-truth [1]. This limitation of the metric motivated Siris *et al.* to also report the number of images upon which it was calculated in order to improve the reliability of the SOR.

Mean Absolute Error (MAE)

In there work, Siris *et al.* [1] expressed reservations about the SOR and also used the Mean Absolute Error (MAE) as a metric in their evaluation. This metric, presented in Equation 7.5, measures the average pixel per pixel difference between the predicted saliency map and the ground truth. This metric was introduced since it also returns a value when there is disagreement between the set of objects in the ground truth and the predicted objects. The lower the value of the MAE, the more similar the predicted saliency map is to the ground truth.

$$MAE = (\frac{1}{n}) \sum_{i=1}^{n} \left| p_i^{\delta} - p_i^{\mathcal{G}} \right| \tag{7.5}$$

where p_i^{δ} is the pixel in the predicted saliency map being compared to the corresponding pixel $p_i^{\mathcal{G}}$ in the ground truth and n is the number of pixels in either the predicted saliency map or the ground truth where it is assumed that both images are of the same size.

Rank Agreement Score (RAS)

The SOR and MAE metrics look at the problem of saliency ranking from a bottom-up perspective by assessing and comparing the values of pixels in the predicted saliency map and the ground truth. In itself, this highlights an opportunity of looking at the problem from a top-down approach when assessing saliency ranking. For this reason, we propose a new metric, the Rank Agreement Score (RAS), that compares the predicted masks as a whole in both images. This metric also compares the rank of the predicted mask with the rank of the corresponding mask in the ground truth.

Effectively, the RAS counts how many predicted masks were matched to the ground truth together with a matching rank. Consider a scene where there are five objects in the ground truth, each with a particular rank. In this case the saliency ranking technique would detect four out of the five masks and three of them would also have the same rank as their corresponding mask in the ground truth. In this case, the Rank Agreement Score would return the value of 0.6, indicating that there was an agreement of 60% between the prediction and the ground truth.

7.5.2 Configuration Experiments

This section presents the experiments that were used to determine the percentage threshold $T\%$ that denotes the coverage of a mask in a grid segment in our saliency-driven ranking technique presented in Section 5.5. A mask features in different grid segments and every grid segment would have a rank. The rank of the mask therefore needs to be representative of the grid segments' ranks that cover it. This experiment also investigates which rank from the grid is to be assigned to the mask. The alternative modes are the minimum rank, the maximum rank or the average.

Out of the 2418 images in the validation set proposed by [1], three sets of 100 images each were randomly generated for this experiment. For each set of 100 images, our predicted saliency rank applied to the Mask R-CNN masks was evaluated using the three metrics discussed above. The threshold was incremented in units of 10 for the minimum, maximum and average rank respectively. For each set, the experiment was carried out four times, once for every metric, namely: RAS, SOR, MAE (saliency map) and MAE (binary map). Using this methodology, the results of the first part of the experiment where results were generated for three random sets are presented in Tables A.1, A.2, A.3 and A.4. These initial four tables do not provide clearly identifiable trends. However, throughout all results, it is clear that the *Min Rank Value* column gives the least standard deviation across all results. This is a preliminary indication stating that the *Min Rank Value* is returning the most stable set of results.

Once that the individual results were generated for each set, the averages of these results were aggregated in a table for each metric. These aggregated average results are presented in Tables A.5, A.6, A.7 and A.8 found in Appendix 1. These results start to indicate how the best values are mostly occurring at the 60% and

90% thresholds and this indicated that these needed further investigation prior to concluding the configuration.

Based on the previous sets of results, the last phase of the experiment investigated in more detail the behaviour of the 60% and 90% thresholds. This was done by generating instances for these thresholds for every single image in the image set and the results are presented in Table A.9. The results for maximum and average rank values of the 60% threshold were not generated since previous results showed that this threshold was not performing well on these modes. The final results presented in this table clearly indicate that the best performance is achieved at 90% threshold on the minimum rank value. The worst performance was in both MAE metrics. This was not given much weight since the pixel to pixel accuracy depends on the similarity in the shape of the mask in the ground truth and predicted image and does not have a direct effect on the ranking of saliency.

7.6 Comparative Study with Siris *et al.* [1]

This section presents a comparative study with the recently published "Inferring Shift Ranks" method [1]. The first part introduces the dataset that was proposed in this same publication were we also give an outline of how the authors constructed it. This is followed by a section that evaluates how our saliency-driven ranking module performs with other methods when used to rank the masks of Mask R-CNN. These results include a quantitative and qualitative evaluation.

7.6.1 Saliency Ranking Dataset

One of the major contributions of [1] was the annotated dataset that provided the first form of ground truth for saliency ranking. This dataset combines the COCO dataset [71] with the SALICON dataset [18]. The COCO dataset is widely used for object detection and segmentation applications. On the other hand, the SALICON dataset was built on top of COCO to provide mouse trajectory based

	RAS		
Threshold	*Min Rank Value*	*Max Rank Value*	*Average Rank Value*
60	21.83%	-	-
90	**31.96%**	21.33%	21.09%

	SOR		
Threshold	*Min Rank Value*	*Max Rank Value*	*Average Rank Value*
60	0.710	-	-
90	**0.716**	0.715	0.714

	MAE (Binary Saliency Map)		
Threshold	*Min Rank Value*	*Max Rank Value*	*Average Rank Value*
60	0.121	-	-
90	0.119	**0.117**	0.118

	MAE (Binary Mask)		
Threshold	*Min Rank Value*	*Max Rank Value*	*Average Rank Value*
60	0.121	-	-
90	0.119	**0.118**	0.119

Table 7.5: This table presents the result of the metrics on the entire 2418 images in the validation set for the 60 and 90 threshold since they were the best performing thresholds in previous experiments. The results in these tables clearly indicate that the 90% threshold returns the best result. Moreover, the RAS metric suggests that the minimum rank value should be used.

fixations. These fixations are categorised into two. The first one provides fixation point sequences and the other provides fixation maps for each image.

The saliency ranking dataset by [1] built upon these datasets by generating three approaches to generate rank proposals based on the mouse fixation trajectories in the SALICON dataset. The first approach focused solely on the fixation points and sequences where they used four methods to generate the score for each object based on fixation data. These four methods were namely: *average*, *maximum*, $average + maximum$ and $average \times maximum$. The second approach focused on how distinct objects were fixated in sequence while ignoring repeated objects. In the third and last approach the fixation maps were used and the four methods used in approach one were also included. Each of these approaches generated different results that were then evaluated in a participant study that involved 11 participants in an effort to investigate which approach delivers the most stable ranking ground truth based on human judgement. This experiment concluded that the best approach was the second one where ranks were generated from the order in which observers fixated on objects. The ground truth of this dataset was also based on this approach [1].

7.6.2 Results

This section presents the comparative result of our technique combined with the output of Mask R-CNN with the state of the art [1] and other comparable techniques, where applicable. The first part of this section presents the Quantitative Results using the metrics used in similar work [1] [15]. The second part of this section presents the Qualitative Results of this evaluation by discussing visual examples of our method in comparison to the ground truth and the state of the art.

Method	MAE ↓	SOR ↑	#Images used ↑
RSDNet [16]	0.139	0.728	2418
S4Net [59]	0.150	0.891	1507
BASNet [60]	0.115	0.707	2402
CPD-R [61]	0.100	0.766	2417
SCRN [62]	0.116	0.756	2418
Siris *et al.* [1]	0.101	0.792	2365
Mean	*0.124*	*0.770*	*2232*
Ours + Mask R-CNN	**0.119**	**0.716**	**2370**

Table 7.6: Comparison of our technique combined with Mask R-CNN against the current state of the art on the Attention Shift Rank dataset [1]. The results of the other techniques are reproduced from [1]. In case of the MAE, the lower the result the better and in the other cases, the higher the better. The maximum number of images that can be used is 2418.

Quantitative Evaluation

In their work, Siris *et al.* [1] compared their deep learning saliency ranking model with five other techniques using the dataset discussed in Section 7.6.1. For this evaluation, the MAE and SOR metrics were used to compare the techniques. Since the SOR rejects cases where the proposals do not match the ground truth, the number of images used was also reported for a fair comparison. The techniques included in this comparative study constituted of the RSDNet proposed by Islam *et al.* [16] that initially suggested the idea of saliency ranking using deep learning together with four state of the art saliency detection techniques. These salient object detection techniques, [16] [59] [60] [61] and [62], return binary saliency maps.

For comparison purposes, we evaluated our saliency-driven ranking model combined with Mask R-CNN that was presented in Section 5.5 on the same dataset and using the same metrics and methodology used by [1]. Our experiment presented in this section compares saliency ranking results with a technique that returns a saliency ranking proposal together with the other state of the art salient object detection. The results of this comparative study are presented in Table 7.6.

The results presented in Table 7.6 show how our saliency-driven ranking method combined with Mask R-CNN performed similar to most of the other techniques. The MAE compares the pixel to pixel accuracy between masks. While our technique performs better than the average MAE, the same cannot be reported on the SOR. However, the number of images used for the SOR is higher than average. The number of images used needs to be also taken into consideration. For example, while the S4Net obtained the highest SOR value of 0.891, thus raising the average, it used 911 less images than the maximum number of available images. The main reason why some of the other methods performed better is due to the fact that they are deep learning models that were trained and tuned to detect such patterns in the data while ours ranks the output of Mask R-CNN without training.

Qualitative Evaluation

The qualitative results showcased in Figure 7.5 show how our technique compares on visual level with the ground truth and the results of [1]. Our saliency ranking grid output is presented by (d) in each figure while (e) illustrates the result when our saliency ranking model was used to rank the masks of Mask R-CNN. The ground truth of the dataset discussed in Section 7.6.1 is presented in (b) and the result of the state of the art is presented in (c). In all images where masks are ranked, the grey value of the mask corresponds to the rank of the same mask. The higher (or brighter) the grey value of the mask, the higher its rank is.

In general, an image may contain a number of objects in set $\mathcal{O}$ of which a number of them found in set $\mathcal{S}$ are detected by a segmentation technique such that $\mathcal{S} \subseteq \mathcal{O}$. One main limitation in all of these results was the limit of 5 detectable objects set by [1] that was also reflected in the set of objects in the ground truth $\mathcal{G}$ such that $\mathcal{G} \subseteq \mathcal{S}$. When our saliency-driven ranking method is combined with a segmentation technique, its best performance is as good as the number of objects in $\mathcal{S}$. However, the saliency ranking grid itself, operating at pixel level, also ranks regions that contain objects that do not belong to $\mathcal{S}$. An example of this is the sample presented in Figure 7.7 where the building itself is not classified as an object

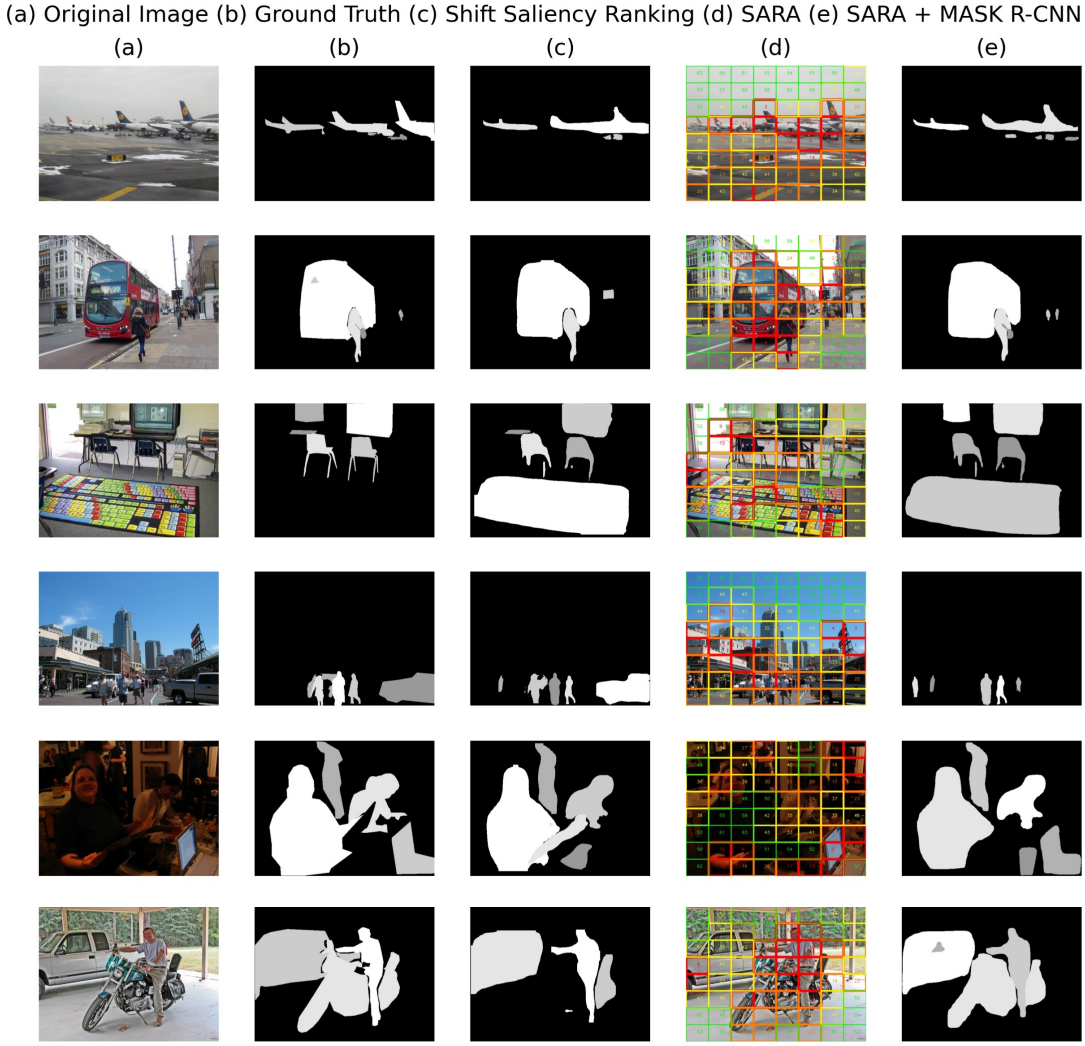

(a) Original Image (b) Ground Truth (c) Shift Saliency Ranking (d) SARA (e) SARA + MASK R-CNN
(a)
(b)
(c)
(d)
(e)

Figure 7.5: Comparison of the proposed technique (d) and (e) with the original image (a), the ground truth (b) and state of the art [1] (c). The masks of the detected objects are coloured by rank. The higher the grey value of the mask the higher the rank of the same mask.

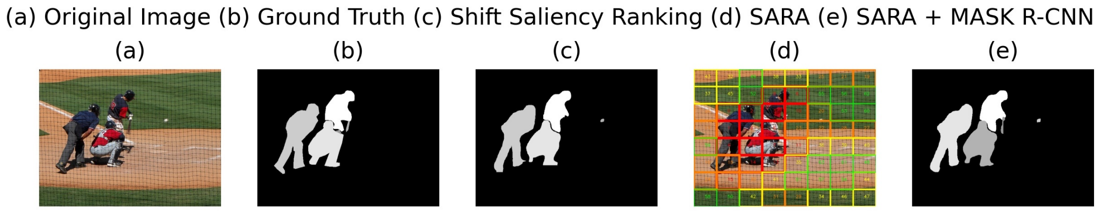

(a) Original Image (b) Ground Truth (c) Shift Saliency Ranking (d) SARA (e) SARA + MASK R-CNN
(a)
(b)
(c)
(d)
(e)

Figure 7.6: An example where there is agreement between our result and that of [1]. Moreover, this case shows how the ball was not featured in the ground truth but was detected by the other techniques.

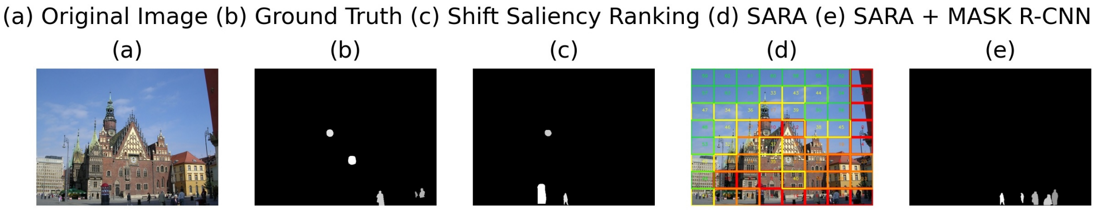

(a) Original Image (b) Ground Truth (c) Shift Saliency Ranking (d) SARA (e) SARA + MASK R-CNN
(a)
(b)
(c)
(d)
(e)

Figure 7.7: An example where there is a predominant salient object in a scene that is not classified by segmentation methods. In this case, the photo features Riga's town hall and while all segmentation methods detected people, an important object class in datasets, they did not detect the building that is the subject of the image. On the other hand, our saliency ranking method clearly indicated that the building is a very salient object of interest in the image.

but it is nonetheless featured by our grid method. This means that if the training of the segmentation method is improved to also detect such objects, our method would be capable of ranking it right away. This is also similar to another case where an object contains smaller regions that are themselves more salient but are not detected by the segmentation model. We present an example of such a case in Figure 7.8 where the yellow button on the remote control attracts attention while the remote control itself contains less salient regions.

7.7 Conclusion

This chapter presented all the experiments and the evaluation procedures that were carried out to evaluate our proposed saliency-driven ranking method. The other techniques that perform a comparable task to our method use a deep learning

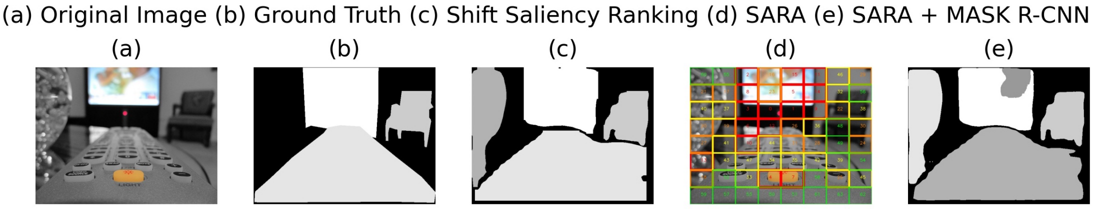

(a) Original Image (b) Ground Truth (c) Shift Saliency Ranking (d) SARA (e) SARA + MASK R-CNN
(a)
(b)
(c)
(d)
(e)

Figure 7.8: This example presents a case where there is a large object, the remote control, in the image that is not the most salient object but contains a particular salient region. All segmentation based techniques would detect the object correctly, including ours together with Mask R-CNN. However, such techniques are not capable of identifying salient regions within it while the saliency grid indicated the yellow button on the remote control.

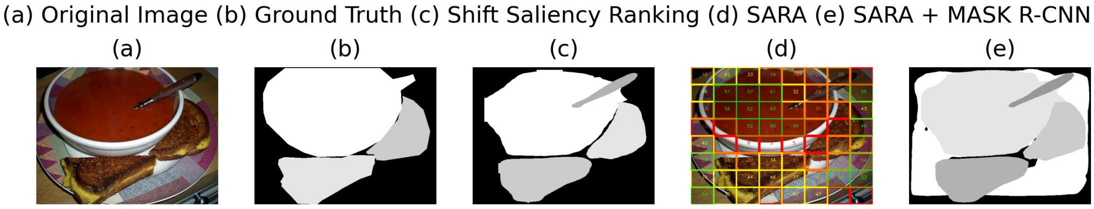

(a) Original Image (b) Ground Truth (c) Shift Saliency Ranking (d) SARA (e) SARA + MASK R-CNN
(a)
(b)
(c)
(d)
(e)

Figure 7.9: The input image in this example features a plate with food. Our technique detected the same objects as the ground truth and the state of the art, although it ranked them differently. In this case, it is interesting to note that our method also detected the plate underneath the food and gave it a higher rank. The plate was not detected by the other technique and the ground truth.

approach. On the other hand, our technique generates scores onto a grid based on pixel-level information that does not require any training. For this reason, we needed to first evaluate the configuration of the model in itself, starting with an experiment to establish the optimal grid size across different datasets. Another component of this model is the use of centre bias in an image. In order to ensure that the results do not rely on centre bias, we carried out an ablation study on this component that resulted in minimal improvement of the final result and therefore confirm that the model does not solely depend on it. Our saliency ranking model was then compared with other benchmarked techniques that detect single salient objects in an image. This experiment also demonstrated that our model returns very close results to the best saliency detection models, even if it serves a different purpose of detecting multiple objects in an image. Another experiment that was carried out was the comparison of our saliency ranking model with human behaviour. An experiment that involved 1267 participants demonstrated that our saliency-driven ranking model successfully matched human behaviour since the click patterns matched the same ranking of our model.

The last part of this chapter investigated the effectiveness of our saliency ranking when used to rank the masks of segmentation models. This included an experiment to confirm the configuration of our method with segmentation techniques and was then followed by a comparative study with the state of the art. These results show that our technique performs very closely to the deep learning based state of the art technique without the need of any training, reaches high performance and is adaptable with any segmentation technique.

7.7.1 Contributions Summary

The main contributions of this chapter are the following:

1. Presentation of an experiment to establish the optimal grid size for saliency-driven ranking;

2. Ablation study on the center bias to investigate its effect on the saliency-driven ranking algorithm score;
3. Benchmarking of saliency ranking with other saliency detection techniques;
4. Demonstration of how the saliency-driven ranking model compares to human behaviour;
5. Identification of how our saliency-driven ranking model can be combined with any segmentation technique;
6. Comparison with the current state of the art and demonstrating how, without any training, our model performs well when compared with deep learning based methods.

8. Conclusion

Wisdom begins at the end.

Daniel Webster

8.1 Summary

The topic of saliency-based object detection is an evolving area. Since the first saliency detection technique [50], a number of other techniques that detect saliency in an image were also proposed. The emerging importance and relevance of deep learning also resulted in a number of saliency detection techniques that use deep learning, such as the current state of the art for saliency detection [165].

While saliency detection techniques were being refined to the current accuracy, object detection and segmentation was also evolving. This resulted in state of the art object segmentation [14] that returns a pixel-level representation of objects in an image.

The work presented in this thesis demonstrates how salient object detection and object segmentation have the potential to be merged as part of their evolution. The saliency of objects that are also accurately segmented can be defined but the current open question [179] [1] [15] is how the segmented objects can be ranked by their saliency.

Very few publications [1] [16] [15] have touched upon the notion of ranking objects by saliency at the same time that the work in this thesis was being finalised and initial findings published in [20]. The main difference between the technique presented in this thesis and other work is that our approach does not require any training and achieves very good results when compared to the deep learning methods. The inherent limitation of the competing techniques is that they depend on the set of classes upon which their network is trained while our method ranks regions of the image based on information from the image.

Our saliency-driven ranking approach is designed to be combined with any segmentation model. This also means that when a new segmentation model becomes available, our method can be easily combined with it. One of the main limitations of the current object segmentation techniques is that they are limited to the number of classes upon which they are trained. Results show that our model detected salient regions that include objects not yet detected by such deep learning models. Once these models will be able to detect them, our model would be able to rank them accordingly. Moreover, when our method is pipelined with segmentation models, it also renders the ranking of objects by saliency explainable since the final rank allows for tracing throughout our model to understand what caused it.

In this thesis, we demonstrated that this method matches human selection behaviour in natural images that are normally perceived as more challenging, hence meeting the first objective of this thesis. Image manipulation was used as a task that can be combined with saliency ranking. For the purpose of this thesis, image manipulation refered to the inpainting or blending of objects in a scene.

The results presented in this thesis show our saliency-driven ranking approach together with the COTS dataset can be efficiently used to explore new ways of achieving attention re-targeting and evaluate the results objectively. The extensive user research that is also included in the COTS dataset also provides further research opportunity in this field.

The lack of inpainting groundtruth in datasets rendered objective evaluation of such techniques impossible. For this reason, we made sure that the newly designed and built COTS dataset [10] addresses this gap and we also demonstrated a concrete method of how inpainting can be objectively evaluated [23] meeting the third objective of this thesis. While designing this dataset that achieves the second objective of this thesis, we also wanted to make sure that it addressed, as much as possible, different needs in the evaluation of a range of computer vision applications.

8.2 Applications of Saliency Ranking

Saliency-driven ranking provides a number of opportunities that fill the gap in the emerging trends of computer vision. Since 2012, the computer vision community witnessed how object detection evolved by making use of CNNs to detect and classify objects in an image. This brought around a number of refined methods until the next paradigm shift arrived. This paradigm shift was pixel-level segmentation and instance segmentation. This meant that objects could now be located, classified and also segmented at pixel-level for every particular instance of a class.

The current state of the art of instance segmentation therefore returns an output that provides a detailed breakdown of an image. The use of saliency to rank objects in a scene appears to be the natural way forward. In this thesis, we demonstrated how objects in natural images can be ranked according to their level of saliency and this ushers the way to a number of emerging applications. Below follows a brief list of such examples:

Image Caption Generation Such techniques make use of information related to the classes of the objects segmented and other spatial information extracted from the scene to generate captions. If the extracted objects are ranked according to their visual saliency, caption generation can be adapted to use saliency-driven ranking algorithm. This can, for example, reconfigure captions depending on the rank of objects in a scene or structure a caption to give priority to less salient objects depending on the application.

Product Placement Most product placement research is carried through subjective information or evaluation. Scenes that contain products can be investigated using the proposed saliency-driven ranking method and the rank can be used to reconfigure the scene until the desired saliency ranking effect is achieved. This means that objects can be placed in prominent places or less prominent ones while having a rapid analysis.

Accessibility Software Read-aloud software that interprets the world for the blind serves a very important role in society. The main downside of such software is that it tends to vocally report every detected object that also results in overwhelming the user. Saliency ranking can be applied to this scenario to prioritise information related to the most salient objects in front of the user and provide an ordered manner in which visual information can be accessed.

8.3 Future Work

The different experiments and their respective results presented in this thesis highlight the opportunity for a deeper exploration of the following research topics presented below:

Intra-Object Segmentation This thesis introduced the idea of extracting further information from the texture or colour layers being guided by depth information. This information can be used to enhance image manipulation, particularly if pattern recognition algorithms are employed to study any variance between layers. Moreover, this can even enhance image manipulation techniques such as re-colouring.

Inpainting Both subjective and objective results of this thesis show that there appears to be a relationship between the size of the object being inpainted and its quality. On the other hand, there is a RoI size up to which exemplar based inpainting performs well. Future research can focus on this problem and explore this threshold and its direct implication on quality particular with the novel generative deep learning methods.

Saliency Ranking This thesis demonstrated an efficient way of ranking the saliency of an image in a similar way to humans. Results show that the proposed saliency-driven ranking technique can be combined with output of segmentation models using fixed parameters. Future work can explore learning based approaches for the tuning of these parameters for improved performance.

Applications of Saliency Ranking Saliency ranking in itself is a novel topic and research area. It ushers a paradigm shift to the way we look at object segmentation. Such a paradigm shift brings along a number of research opportunities into the application of saliency ranking to different domains or applications together with a study on its impact.

Attention re-Targeting Evaluation The detailed saliency ranked output can be used to evaluate the way different algorithms are re-targeting attention in images. Current approaches [6] rely on subjective evaluation and do not provide a detailed and reproducible way of comparing results. The technique presented in this thesis can be precisely used for this purpose.

Enhanced Attention re-Targeting The approach presented in this thesis and most of the existing techniques depend on the interpretation of pixel values in saliency maps. Other attention re-targeting techniques use classifiers to identify objects in an image yet they present a number of limitations. These two approaches can be combined together by having the unsupervised ranking of saliency being assisted by the results of an object detection classifier and therefore improve the semantic quality of the image manipulation result.

COTS Dataset Future iterations of this dataset can potentially include a set of scenes with a more complex natural background that would increase the evaluation possibilities upon it.

A. Configuration Results

RAS	Set 1			Set 2			Set 3		
Threshold	Min Rank Value	Max Rank Value	Avg Rank Value	Min Rank Value	Max Rank Value	Avg Rank Value	Min Rank Value	Max Rank Value	Avg Rank Value
0	23.67%	11.17%	13.53%	17.28%	8.97%	9.30%	19.88%	8.90%	8.55%
10	28.00%	16.52%	18.30%	17.83%	10.15%	10.72%	21.67%	12.18%	11.52%
20	31.83%	24.15%	26.38%	18.17%	14.30%	13.78%	21.83%	13.68%	10.63%
30	33.90%	25.98%	30.92%	20.08%	16.13%	14.50%	22.23%	17.10%	13.73%
40	34.63%	27.20%	30.05%	21.15%	17.73%	16.62%	24.30%	18.00%	13.52%
50	34.75%	28.32%	32.92%	20.85%	18.15%	16.72%	26.05%	22.02%	17.15%
60	36.92%	32.07%	32.18%	21.05%	18.07%	18.03%	27.43%	21.82%	17.15%
70	35.75%	32.22%	32.53%	21.92%	20.68%	21.05%	25.65%	22.60%	22.03%
80	32.92%	31.62%	29.52%	21.78%	20.50%	20.92%	29.57%	26.25%	26.75%
90	39.18%	36.40%	35.65%	20.88%	21.38%	20.15%	28.63%	26.15%	26.35%
95	30.20%	25.82%	25.82%	22.62%	19.25%	20.38%	28.77%	26.73%	27.53%
100	33.40%	29.37%	30.37%	24.67%	22.33%	22.93%	30.03%	28.20%	28.80%
Average	*32.93%*	*26.73%*	*28.18%*	*20.69%*	*17.30%*	*17.09%*	*25.50%*	*20.30%*	*18.64%*
Std Dev	*4.15%*	*7.01%*	*6.41%*	*2.11%*	*4.26%*	*4.32%*	*3.48%*	*6.32%*	*7.33%*

Table A.1: Results of the RAS metric on the three randomly generated sets of 100 images. For each set, the minimum, maximum and average rank values were computed for a range of threshold values.

SOR	Set 1			Set 2			Set 3		
Threshold	Min Rank Value	Max Rank Value	Avg Rank Value	Min Rank Value	Max Rank Value	Avg Rank Value	Min Rank Value	Max Rank Value	Avg Rank Value
0	0.677	0.599	0.619	0.653	0.541	0.539	0.729	0.583	0.692
10	0.653	0.584	0.588	0.647	0.540	0.561	0.684	0.669	0.690
20	0.652	0.590	0.594	0.657	0.550	0.547	0.723	0.583	0.702
30	0.663	0.606	0.604	0.672	0.553	0.594	0.699	0.614	0.690
40	0.651	0.609	0.618	0.667	0.577	0.621	0.705	0.606	0.678
50	0.705	0.633	0.676	0.706	0.635	0.658	0.698	0.658	0.689
60	0.727	0.675	0.692	0.732	0.686	0.701	0.704	0.667	0.709
70	0.725	0.693	0.695	0.702	0.700	0.704	0.692	0.707	0.730
80	0.738	0.715	0.721	0.697	0.734	0.708	0.723	0.745	0.750
90	0.740	0.734	0.735	0.679	0.711	0.692	0.733	0.751	0.755
95	0.737	0.745	0.734	0.675	0.716	0.690	0.721	0.719	0.711
100	0.729	0.736	0.728	0.689	0.719	0.693	0.700	0.712	0.733
Average	*0.700*	*0.660*	*0.667*	*0.681*	*0.639*	*0.642*	*0.709*	*0.668*	*0.711*
Std Dev	*0.037*	*0.063*	*0.058*	*0.025*	*0.081*	*0.066*	*0.016*	*0.060*	*0.026*

Table A.2: Results of the SOR metric on the three randomly generated sets of 100 images. For each set, the minimum, maximum and average rank values were computed for a range of threshold values.

MAE (Binary SM)	Set 1			Set 2			Set 3		
Threshold	Min Rank Value	Max Rank Value	Avg Rank Value	Min Rank Value	Max Rank Value	Avg Rank Value	Min Rank Value	Max Rank Value	Avg Rank Value
0	0.145	0.168	0.149	0.142	0.176	0.172	0.159	0.187	0.182
10	0.140	0.159	0.145	0.138	0.165	0.165	0.147	0.169	0.164
20	0.134	0.149	0.131	0.134	0.157	0.156	0.145	0.151	0.152
30	0.131	0.134	0.125	0.125	0.141	0.137	0.145	0.139	0.139
40	0.133	0.139	0.132	0.124	0.133	0.129	0.139	0.139	0.137
50	0.134	0.134	0.127	0.124	0.127	0.126	0.136	0.133	0.137
60	0.130	0.132	0.127	0.120	0.116	0.120	0.138	0.126	0.138
70	0.127	0.131	0.128	0.121	0.114	0.118	0.135	0.126	0.136
80	0.126	0.127	0.127	0.120	0.113	0.119	0.132	0.128	0.133
90	0.125	0.125	0.125	0.118	0.109	0.115	0.133	0.128	0.132
95	0.124	0.122	0.123	0.118	0.108	0.114	0.135	0.132	0.133
100	0.123	0.121	0.121	0.115	0.108	0.111	0.134	0.131	0.132

Average	*0.131*	*0.137*	*0.130*	*0.125*	*0.131*	*0.132*	*0.140*	*0.141*	*0.143*
Std Dev	*0.007*	*0.015*	*0.009*	*0.009*	*0.024*	*0.021*	*0.008*	*0.019*	*0.015*

Table A.3: Results of the MAE (Binary Saliency Map) metric on the three randomly generated sets of 100 images. For each set, the minimum, maximum and average rank values were computed for a range of threshold values.

MAE (Binary Mask)	Set 1			Set 2			Set 3		
Threshold	Min Rank Value	Max Rank Value	Avg Rank Value	Min Rank Value	Max Rank Value	Avg Rank Value	Min Rank Value	Max Rank Value	Avg Rank Value
0	0.141	0.162	0.143	0.137	0.177	0.171	0.164	0.185	0.190
10	0.137	0.154	0.141	0.132	0.163	0.163	0.145	0.163	0.167
20	0.127	0.143	0.125	0.127	0.153	0.152	0.142	0.141	0.154
30	0.126	0.125	0.120	0.117	0.132	0.130	0.142	0.130	0.140
40	0.129	0.132	0.132	0.118	0.124	0.125	0.138	0.135	0.139
50	0.133	0.128	0.126	0.119	0.122	0.123	0.137	0.129	0.140
60	0.134	0.130	0.127	0.115	0.108	0.117	0.139	0.120	0.143
70	0.129	0.132	0.132	0.114	0.109	0.114	0.138	0.123	0.142
80	0.127	0.130	0.130	0.114	0.110	0.116	0.137	0.130	0.140
90	0.126	0.125	0.126	0.113	0.107	0.113	0.138	0.133	0.138
95	0.125	0.124	0.125	0.112	0.107	0.112	0.137	0.137	0.137
100	0.124	0.124	0.124	0.110	0.110	0.110	0.136	0.136	0.136

Average	*0.130*	*0.134*	*0.129*	*0.119*	*0.127*	*0.129*	*0.141*	*0.139*	*0.147*
Std Dev	*0.005*	*0.013*	*0.007*	*0.008*	*0.025*	*0.021*	*0.008*	*0.018*	*0.016*

Table A.4: Results of the MAE (Binary Mask) metric on the three randomly generated sets of 100 images. For each set, the minimum, maximum and average rank values were computed for a range of threshold values.

	RAS		
Threshold	Min Rank Value	Max Rank Value	Average Rank Value
0	20.28%	9.68%	10.46%
10	22.50%	12.95%	13.51%
20	23.94%	17.38%	16.93%
30	25.41%	19.74%	19.72%
40	26.69%	20.98%	20.06%
50	27.22%	22.83%	22.26%
60	28.47%	23.98%	22.46%
70	27.77%	25.17%	25.21%
80	28.09%	26.12%	25.73%
90	29.57%	28.01%	27.38%
95	27.19%	23.74%	24.58%
100	29.37%	26.14%	27.37%

Average	*25.99%*	*20.68%*	*20.37%*
Std Dev	*2.94%*	*5.88%*	*5.44%*

Table A.5: The aggregated average values of Sets 1, 2 and 3 for the RAS metric.

	SOR		
Threshold	Min Rank Value	Max Rank Value	Average Rank Value
0	0.687	0.574	0.617
10	0.661	0.598	0.613
20	0.677	0.574	0.614
30	0.678	0.591	0.630
40	0.674	0.597	0.639
50	0.703	0.642	0.674
60	0.721	0.676	0.701
70	0.706	0.700	0.710
80	0.719	0.732	0.726
90	0.717	0.732	0.728
95	0.711	0.727	0.712
100	0.706	0.722	0.718

Average	*0.694*	*0.642*	*0.665*
Std Dev	*0.022*	*0.064*	*0.048*

Table A.6: The aggregated average values of Sets 1, 2 and 3 for the SOR metric.

	MAE (Binary Saliency Map)		
Threshold	Min Rank Value	Max Rank Value	Average Rank Value
0	0.149	0.177	0.168
10	0.142	0.164	0.158
20	0.138	0.152	0.146
30	0.134	0.138	0.134
40	0.132	0.137	0.133
50	0.131	0.131	0.130
60	0.130	0.125	0.129
70	0.128	0.124	0.128
80	0.126	0.123	0.126
90	0.125	0.120	0.124
95	0.126	0.121	0.123
100	0.124	0.120	0.121

Average	*0.133*	*0.139*	*0.138*
Std Dev	*0.007*	*0.019*	*0.015*

Table A.7: The aggregated average values of Sets 1, 2 and 3 for the MAE (Binary Saliency Map) metric.

	MAE (Binary Mask)		
Threshold	Min Rank Value	Max Rank Value	Average Rank Value
0	0.148	0.175	0.168
10	0.138	0.160	0.157
20	0.132	0.146	0.143
30	0.128	0.129	0.130
40	0.129	0.130	0.132
50	0.130	0.126	0.130
60	0.129	0.119	0.129
70	0.127	0.121	0.129
80	0.126	0.123	0.128
90	0.126	0.122	0.126
95	0.125	0.123	0.125
100	0.123	0.123	0.123

Average	*0.131*	*0.135*	*0.137*
Std Dev	*0.007*	*0.019*	*0.015*

Table A.8: The aggregated average values of Sets 1, 2 and 3 for the MAE (Binary Mask) metric.

	RAS		
Threshold	*Min Rank Value*	*Max Rank Value*	*Average Rank Value*
60	21.83%	-	-
90	**31.96%**	21.33%	21.09%

	SOR		
Threshold	*Min Rank Value*	*Max Rank Value*	*Average Rank Value*
60	0.710	-	-
90	**0.716**	0.715	0.714

	MAE (Binary Saliency Map)		
Threshold	*Min Rank Value*	*Max Rank Value*	*Average Rank Value*
60	0.121	-	-
90	0.119	**0.117**	0.118

	MAE (Binary Mask)		
Threshold	*Min Rank Value*	*Max Rank Value*	*Average Rank Value*
60	0.121	-	-
90	0.119	**0.118**	0.119

Table A.9: This table presents the result of the metrics on the entire 2418 images in the validation set for the 60 and 90 threshold since they were the best performing thresholds in previous experiments. The results in these tables clearly indicate that the 90% threshold returns the best result. Moreover, the RAS metric suggests that the minimum rank value should be used.

References

[1] A. Siris, J. Jiao, G. Tam, X. Xie, and R. Lau, "Inferring attention shift ranks of objects for image saliency," in *Proc. of. the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," in *Proc. of ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, (New York, NY, USA), pp. 24:1–24:11, ACM, 2009.

[3] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut - interactive foreground extraction using iterated graph cuts," in *in Proc. of ACM SIGGRAPH 2004,*, Aug 2004.

[4] G. Krishna, *The Best Interface Is No Interface: The simple path to brilliant technology (Voices That Matter).* New Riders, 2015.

[5] A. Borji, M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Transactions on Image Processing*, vol. 24, pp. 5706–5722, Dec. 2015.

[6] R. Mechrez, E. Shechtman, and L. Zelnik-Manor, "Saliency driven image manipulation," in *Proc. of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1368–1376, Mar 2018.

[7] A. Borji, "What is a salient object? A dataset and a baseline model for salient object detection," *IEEE Transactions on Image Processing*, vol. 24, pp. 742–756, Feb 2015.

[8] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, Aug 13-17, 2016*, pp. 1135–1144, 2016.

[10] D. Seychell, C. J. Debono, M. Bugeja, J. Borg, and M. Sacco, "Cots: A multipurpose rgb-d dataset for saliency and image manipulation applications," *IEEE Access*, vol. 9, pp. 21481–21497, 2021.

[11] L. Gatys, A. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv*, vol. abs/1508.06576, 2015.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 2672–2680, Curran Associates, Inc., 2014.

[13] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. of the Advances in Neural Information Processing Systems (NEURIPS)* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proc. of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[15] M. Kalash, M. A. Islam, and N. D. B. Bruce, "Relative saliency and ranking: Models, metrics, data and benchmarks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 204–219, 2021.

[16] M. Islam, M. Kalash, and N. Bruce, "Revisiting salient object detection: Simultaneous detection, ranking, and subitizing of multiple salient objects," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[17] T. V. Nguyen, B. Ni, H. Liu, W. Xia, J. Luo, M. Kankanhalli, and S. Yan, "Image re-attentionizing," *IEEE Transactions on Multimedia*, vol. 15, pp. 1910–1919, Dec 2013.

[18] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "Salicon: Saliency in context," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1072–1080, 2015.

[19] T. T. Dang, A. Beghdadi, and M. Larabi, "Perceptual evaluation of digital image completion quality," in *Proc. of the 21st European Signal Processing Conference (EUSIPCO 2013)*, pp. 1–5, Sep. 2013.

[20] D. Seychell and C. Debono, "Ranking regions of visual saliency in RGB-D content," in *Proc. of the 2018 IEEE International Conference on 3D-Immersion (IC3D)*, Dec. 2018.

[21] D. Seychell and C. J. Debono, “Intra-object segmentation using depth information,” in *Proc. of the 2018 19th IEEE Mediterranean Electrotechnical Conference (MELECON)*, pp. 30–34, May 2018.

[22] D. Seychell and C. Debono, “Monoscopic inpainting approach using depth information,” in *Proc. of the 18th IEEE Mediterranean Electrotechnical Conference*, 2016.

[23] D. Seychell and C. J. Debono, “An approach for objective quality assessment of image inpainting results,” in *Proc. of the 2020 20th IEEE Mediterranean Electrotechnical Conference (MELECON)*, June 2020.

[24] D. Seychell and C. Debono, “Efficient object selection using depth and texture information,” in *Proc. of the 2016 Visual Communications and Image Processing (VCIP)*, Nov 2016.

[25] M. E. Raichle, “Two views of brain function,” *Trends in Cognitive Sciences*, vol. 14, pp. 180–190, 2010.

[26] D. Kelly, “Information capacity of a single retinal channel,” *IRE Transactions on Information Theory*, vol. 8, pp. 221–226, April 1962.

[27] M. W. Passer and R. E. Smith, *Psychology: The science of mind and behavior (4th ed.).* McGraw-Hill, 2007.

[28] R. Atkinson and R. Shiffrin, “Human memory: A proposed system and its control processes,” *The psychology of learning and motivation*, p. 89–195, 1968.

[29] S. Atkinson, S. Tomley, C. Landau, and O. S, *The Psychology Book.* Dorling Kindersley Limited, 2012.

[30] A. Miyake and P. Shah, *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control.* Cambridge University Press, 1999.

[31] J. Braun, C. Koch, J. L. Davis, and G. A. Miller, “The magical number seven, plus or minus two: Some limits on our capacity for processing information,” *Psychological Review*, 1956.

[32] E. Service, “The effect of word length on immediate serial recall depends on phonological complexity, not articulatory duration,” *The Quarterly Journal of Experimental Psychology Section A*, vol. 51, no. 2, pp. 283–304, 1998.

[33] R. Engle, S. W. Tuholski, J. E. Laughlin, and A. Conway, “Working memory, short-term memory and general fluid intelligence: A latent variable approach,” *Journal of Experimental Psychology: General*, vol. 130, pp. 169–183, 03 1999.

[34] K. K. Evans, T. S. Horowitz, P. Howe, R. Pedersini, E. Reijnen, Y. Pinto, Y. Kuzmova, and J. M. Wolfe, "Visual attention," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 2, no. 5, pp. 503–514, 2011.

[35] C. W. Eriksen and J. E. Hoffman, "Temporal and spatial characteristics of selective encoding from visual displays," *Perception & Psychophysics*, vol. 12, pp. 201–204, Mar 1972.

[36] W. Commons, "Vision spotlight diagram." `https://en.wikipedia.org/wiki/Attention#/media/File:Wikipedia-spotlight.jpg`, 2010.

[37] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proc. of the 2009 IEEE 12th International Conference on Computer Vision*, pp. 2106–2113, Sept. 2009.

[38] A. R. Hunt and A. Kingstone, "Covert and overt voluntary attention: linked or independent?," *Cognitive Brain Research*, vol. 18, no. 1, pp. 102 – 105, 2003.

[39] R. Cong, J. Lei, H. Fu, M. Cheng, W. Lin, and Q. Huang, "Review of visual saliency detection with comprehensive information," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[40] P. Sharma, "Evaluating visual saliency algorithms: Past, present and future," *Journal of Imaging Science and Technology*, vol. 59, no. 5, pp. 50501 - 1 - 50501-17, Sept. 2015.

[41] G. Yildirim and S. Süsstrunk, "Fasa: Fast, accurate, and size-aware salient object detection," in *Computer Vision – ACCV 2014*, (Cham), pp. 514–528, Springer International Publishing, 2015.

[42] A. Borji, "Boosting bottom-up and top-down visual features for saliency estimation," in *Proc. of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 438–445, June 2012.

[43] C. Zhu, K. Huang, and G. Li, "Automatic salient object detection for panoramic images using region growing and fixation prediction model," *Journal of Computer Research Repository*, 2017.

[44] A. Azaza and A. Douik, "Deep saliency features for video saliency prediction," in *2018 International Conference on Advanced Systems and Electric Technologies* $(IC_ASET), pp. 355--359, 2018.$

[45] F. Guo, W. Wang, Z. Shen, J. Shen, L. Shao, and D. Tao, "Motion-aware rapid video saliency detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 12, pp. 4887–4898, 2020.

[46] J. Li, F. Meng, and J. Mao, “Saliency detection on videos with scene change,” in *2014 International Conference on Audio, Language and Image Processing*, pp. 506–510, 2014.

[47] W. Wang, J. Shen, J. Xie, M. Cheng, H. Ling, and A. Borji, “Revisiting video saliency prediction in the deep learning era,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 220–237, 2021.

[48] M. Cokelek, N. Imamoglu, C. Ozcinar, E. Erdem, and A. Erdem, “Leveraging frequency based salient spatial sound localization to improve 360° video saliency prediction,” in *2021 17th International Conference on Machine Vision and Applications (MVA)*, pp. 1–5, 2021.

[49] R. Du and A. Varshney, “Saliency computation for virtual cinematography in 360° videos,” *IEEE Computer Graphics and Applications*, vol. 41, no. 4, pp. 99–106, 2021.

[50] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1254–1259, Nov 1998.

[51] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, “The secrets of salient object segmentation,” in *Proc. of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 280–287, June 2014.

[52] C. Yang, L. Zhang, H. Lu, X. Ruan, and M. Yang, “Saliency detection via graph-based manifold ranking,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3166–3173, 2013.

[53] R. Margolin, A. Tal, and L. Zelnik-Manor, “What makes a patch distinct?,” in *Proc. of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1139–1146, June 2013.

[54] J. Lu, X. Wen, H. Shao, Z. Lu, and Y. Chen, “An effective visual saliency detection method based on maximum entropy random walk,” in *Proc. of the 2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pp. 1–6, July 2016.

[55] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, “Amulet: Aggregating multi-level convolutional features for salient object detection,” in *Proc. of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 202–211, Oct. 2017.

[56] Y. Hu, Z. Chen, Z. Chi, and H. Fu, “Learning to detect saliency with deep structure,” in *Proc. of the 2015 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1770–1775, Oct 2015.

[57] L. Qu, S. He, J. Zhang, J. Tian, Y. Tang, and Q. Yang, “Rgbd salient object detection via deep fusion,” *IEEE Transactions on Image Processing*, vol. 26, pp. 2274–2285, May 2017.

[58] G. Lee, Y. Tai, and J. Kim, “Eld-net: An efficient deep learning architecture for accurate saliency detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 1599–1610, July 2018.

[59] R. Fan, M. Cheng, Q. Hou, T. Mu, J. Wang, and S. Hu, “S4net: Single stage salient-instance segmentation,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6096–6105, 2019.

[60] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, “Basnet: Boundary-aware salient object detection,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7471–7481, 2019.

[61] Z. Wu, L. Su, and Q. Huang, “Cascaded partial decoder for fast and accurate salient object detection,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3902–3911, 2019.

[62] Z. Wu, L. Su, and Q. Huang, “Stacked cross refinement network for edge-aware salient object detection,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7263–7272, 2019.

[63] C. Zhu, G. Li, W. Wang, and R. Wang, “Salient object detection with complex scene based on cognitive neuroscience,” in *2017 IEEE Third International Conference on Multimedia Big Data (BigMM)*, pp. 33–37, April 2017.

[64] T. Lee and S. Yu, “An information-theoretic framework for understanding saccadic behaviors,” in *Advances in Neural Information Processing Systems (NIPS)*, MIT Press, 2000.

[65] N. D. B. Bruce and J. K. Tsotsos, “Saliency based on information maximization,” in *Proc. of the 18th International Conference on Neural Information Processing Systems*, NIPS’05, (Cambridge, MA, USA), pp. 155–162, MIT Press, 2005.

[66] Q. Hou, M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, “Deeply supervised salient object detection with short connections,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 815–828, April 2019.

[67] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, “Advanced deep-learning techniques for salient and category-specific object detection: A survey,” *IEEE Signal Processing Magazine*, vol. 35, pp. 84–100, Jan 2018.

[68] H. Chen and Y. Li, “Progressively complementarity-aware fusion network for rgb-d salient object detection,” in *Proc. of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3051–3060, June 2018.

[69] L. Zhang, C. Yang, H. Lu, X. Ruan, and M. Yang, “Ranking saliency,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1892–1904, Sep. 2017.

[70] K. Rahul and A. K. Tiwari, “Saliency enabled compression in jpeg framework,” *IET Image Processing*, vol. 12, no. 7, pp. 1142–1149, 2018.

[71] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Proc. of the 2014 European Conference on Computer Vision ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 740–755, Springer International Publishing, 2014.

[72] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[73] T. Morris, *Computer Vision and Image Processing.* Palgrave Macmillan, 2003.

[74] N. Jain and A. Lala, “Image segmentation: A short survey,” in *Proc. of Confluence 2013: The Next Generation Information Technology Summit (4th International Conference)*, pp. 380–384, Sep. 2013.

[75] J. N. Chandra, B. S. Supraja, and V. Bhavana, “A survey on advanced segmentation techniques in image processing applications,” in *Proc. of the 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pp. 1–5, Dec 2017.

[76] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr, “Diffusion snakes: introducing statistical shape knowledge into the mumford-shah functional,” *Journal of Computer Vision*, vol. 50, pp. 295–313, 2002.

[77] K. Kolev, T. Brox, and D. Cremers, “Robust variational segmentation of 3d objects from multiple views,” in *Pattern Recognition (Proc. DAGM)*, vol. 4174 of *Lecture Notes in Computer Science*, pp. 688–697, Springer, Sept. 2006.

[78] H. Fu, D. Xu, and S. Lin, “Object-based multiple foreground segmentation in rgbd video,” *IEEE Transactions on Image Processing*, vol. 26, pp. 1418–1427, March 2017.

[79] F. Meng, H. Li, Q. Wu, B. Luo, and K. N. Ngan, "Weakly supervised part proposal segmentation from multiple images," *IEEE Transactions on Image Processing*, vol. 26, pp. 4019–4031, Aug 2017.

[80] M. Ramanathan, W. Y. Yau, and E. K. Teoh, "Improving human body part detection using deep learning and motion consistency," in *Proc. of the 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1–5, Nov 2016.

[81] Y. Zhang, Z. Liu, W. Zhou, and Y. Zhang, "Object recognition base on deep belief network," in *Proc. of the 2015 10th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pp. 268–273, Nov 2015.

[82] J. Wang, Z. Zhang, V. Premachandran, and A. L. Yuille, "Discovering internal representations from object-cnns using population encoding," *ArXiv*, vol. abs/1511.06855, 2015.

[83] S. Z. Li, *Markov Random Field Modeling in Image Analysis.* Springer Publishing Company, Incorporated, 3rd ed., 2009.

[84] Q. Dai, J. Qiao, F. Liu, X. Shi, and H. Yang, "A human body part segmentation method based on markov random field," in *Proc. of the 2012 International Conference on Control Engineering and Communication Technology*, pp. 149–152, Dec 2012.

[85] D. S. Hochbaum and V. Singh, "An efficient algorithm for co-segmentation," in *Proc. of the 2009 IEEE 12th International Conference on Computer Vision*, pp. 269–276, Sept 2009.

[86] C. Rother, T. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs," in *Proc. of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '06, (Washington, DC, USA), pp. 993–1000, IEEE Computer Society, 2006.

[87] J. Jiao, Y. Wei, Z. Jie, H. Shi, R. W. Lau, and T. S. Huang, "Geometry-aware distillation for indoor semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[88] T. He, C. Shen, Z. Tian, D. Gong, C. Sun, and Y. Yan, "Knowledge adaptation for efficient semantic segmentation," in *Proc. of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[89] T. Yen, H. Hsu, P. Pati, M. Gabrani, A. Foncubierta-Rodríguez, and P. Chung, "Ninepins: Nuclei instance segmentation with point annotations," *arXiv preprint arXiv:2006.13556*, 2020.

[90] A. Grenier, "Visual scene understanding for self-driving cars using deep learning and stereovision," 2019.

[91] F. Zhang, C. Guan, J. Fang, S. B., R. Yang, p. Torr, and V. Prisacariu, "Instance segmentation of lidar point clouds," *ICRA, Cited by*, vol. 4, no. 1, 2020.

[92] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[93] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *Proc. of the Eighth IEEE International Conference onComputer Vision, 2001. ICCV 2001.*, vol. 1, pp. 105–112 vol.1, 2001.

[94] M. Stoer and F. Wagner, "A simple min-cut algorithm," *Journal of the ACM*, vol. 44, pp. 585–591, July 1997.

[95] K. Vaiapury, A. Aksay, and E. Izquierdo, "GrabcutD: Improved grabcut using depth information," in *Proc. of the 2010 ACM Workshop on Surreal Media and Virtual Cloning*, SMVC '10, (New York, NY, USA), pp. 57–62, ACM, 2010.

[96] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. of the 2014 IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[97] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[98] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[99] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, 2017.

[100] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, 2017.

[101] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: object detection via region-based fully convolutional networks," *CoRR*, vol. abs/1605.06409, 2016.

[102] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[103] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, eds.), (Cham), pp. 234–241, Springer International Publishing, 2015.

[104] Z. Z., M. M. R. Siddiquee, N. T., and J. L., "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, (Cham), pp. 3–11, Springer International Publishing, 2018.

[105] N. Micallef, D. Seychell, and C. Bajada, "A nested u-net approach for brain tumour segmentation," in *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON)*, pp. 376–381, 2020.

[106] N. Micallef, D. Seychell, and C. Bajada, "Exploring the u-net++ model for automatic brain tumor segmentation," *IEEE Access*, vol. 9, pp. 125523–125539, 2021.

[107] L. Wang, H. Jin, R. Yang, and M. Gong, "Stereoscopic inpainting: Joint color and depth completion from stereo images," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2008.

[108] S. Shivaranjani and R. Priyadharsini, "A survey on inpainting techniques," in *Proc. of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 2934–2937, March 2016.

[109] S. Zarif, I. Faye, and D. Rohaya, "A comparative study of different image completion techniques," in *Proc. of the 2014 International Conference on Computer and Information Sciences (ICCOINS)*, pp. 1–6, June 2014.

[110] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Proc. of the 2001 IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 355–362, 2001.

[111] R. Barnard, *Elementary Fluid Dynamics.* Oxford University Press, 1990.

[112] T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *Journal of Visual Communication and Image Representation*, vol. 12, no. 4, pp. 436 – 449, 2001.

[113] A. Telea, "An image inpainting technique based on the fast marching method.," *Journal of Graphics, GPU and Game Tools*, vol. 9, no. 1, pp. 23–34, 2004.

[114] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. of the International Conference on Computer Vision (ICCV 99)*, ICCV '99, (Washington, DC, USA), pp. 1033–, IEEE Computer Society, 1999.

[115] M. S. Bertalmio A., Caselles V. and S. G., "Inpainting," in *Computer Vision: A Reference Guide* (I. K., ed.), Springer, 2014.

[116] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *Image Processing, IEEE Transactions on*, vol. 13, pp. 1200 –1212, Sept 2004.

[117] D. Sun, L. Yuan, Y. Zhang, J. Zhang, and G. Pan, "Structure-aware image completion with texture propagation," in *Proc. of the 2011 Sixth International Conference on Image and Graphics*, pp. 199–204, Aug 2011.

[118] G. Liu, F. A. Reda, K. J. Shih, T. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," *Proc. of the 2018 European Conference on Computer Vision (ECCV)*, Sept 2018.

[119] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, June 2016.

[120] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4076–4084, July 2017.

[121] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (gans): A survey," *IEEE Access*, vol. 7, pp. 36322–36333, 2019.

[122] K. Zhu, X. Liu, and H. Yang, "A survey of generative adversarial networks," in *Proc. of the 2018 Chinese Automation Congress (CAC)*, pp. 2768–2773, Nov 2018.

[123] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 105–114, July 2017.

[124] X. Wang and A. Gupta, “Generative image modeling using style and structure adversarial networks,” in *Computer Vision – ECCV 2016* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 318–335, Springer International Publishing, 2016.

[125] G. Antipov, M. Baccouche, and J. Dugelay, “Face aging with conditional generative adversarial networks,” in *Proc. of the 2017 IEEE International Conference on Image Processing (ICIP)*, pp. 2089–2093, Sep. 2017.

[126] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, “Towards large-pose face frontalization in the wild,” in *Proc. of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4010–4019, Oct 2017.

[127] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, “Generative adversarial text to image synthesis,” in *Proc. of The 33rd International Conference on Machine Learning* (M. F. Balcan and K. Q. Weinberger, eds.), vol. 48 of *Proc. of Machine Learning Research*, (New York, New York, USA), pp. 1060–1069, PMLR, 20–22 Jun 2016.

[128] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, “Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks,” in *Proc. of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5908–5916, Oct 2017.

[129] L. Yuan, C. Ruan, H. Hu, and D. Chen, “Image inpainting based on patch-gans,” *IEEE Access*, vol. 7, pp. 46411–46421, 2019.

[130] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in *Proc. of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, June 2016.

[131] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and locally consistent image completion,” *ACM Trans. Graph.*, vol. 36, pp. 107:1–107:14, July 2017.

[132] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, “High-resolution image inpainting using multi-scale neural patch synthesis,” in *Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4076–4084, July 2017.

[133] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, “Semantic image inpainting with deep generative models,” in *Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6882–6890, July 2017.

[134] Z. Li, H. Wang, and K. Li, “Saliency-aware image completion,” in *Proc. of the 2014 IEEE 17th International Conference on Computational Science and Engineering*, pp. 509–512, Dec 2014.

[135] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *Proc. of the ACM SIGGRAPH 2004*, (New York, NY, USA), pp. 600–608, ACM, 2004.

[136] "Nagoya university sequences." [Online]. Available: http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/. Accessed: 2015-06-1.

[137] A. Janoch, S. Karayev, Yangqing Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell, "A category-level 3-d object dataset: Putting the kinect to work," in *Proc. of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 1168–1174, Nov 2011.

[138] S. M. Rhee, Y. B. Lee, J. D. K. Kim, and T. Rhee, "Split and merge approach for detecting multiple planes in a depth image," in *Proc. of the 19th IEEE International Conference on Image Processing*, pp. 1213–1216, Sept 2012.

[139] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering surface layout from an image," *International Journal of Computer Vision*, vol. 75, pp. 151–172, Feb 2007.

[140] A. Kimura, "Saliency map implementation." [Online]. Available: https://github.com/akisato-/pySaliencyMap/ . Accessed: 2017-11-1.

[141] M. A. Islam, M. Rochan, N. D. B. Bruce, and Y. Wang, "Gated feedback refinement network for dense image labeling," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4877–4885, 2017.

[142] W. Abdulla, "Mask r-cnn for object detection and instance segmentation on keras and tensorflow." `https://github.com/matterport/Mask_RCNN`, 2017.

[143] M. Firman, "RGBD datasets: Past, present and future," in *Proc. of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 661–673, June 2016.

[144] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *Proc. of the 2011 IEEE International Conference on Robotics and Automation*, pp. 1817–1824, May 2011.

[145] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, "Bigbird: A large-scale 3d database of object instances," in *Proc. of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 509–516, May 2014.

[146] S. Choi, Q.-Y. Zhou, S. Miller, and V. Koltun, "A large dataset of object scans," *arXiv:1602.02481*, 2016.

[147] A. Richtsfeld, T. Mörwald, J. Prankl, M. Zillich, and M. Vincze, "Segmentation of unknown objects in indoor environments," in *Proc. of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4791–4796, Oct 2012.

[148] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze, "A global hypotheses verification method for 3d object recognition," in *Proc. of the Computer Vision – ECCV 2012* (A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, eds.), (Berlin, Heidelberg), pp. 511–524, Springer Berlin Heidelberg, 2012.

[149] F. Tombari, L. Di Stefano, and S. Giardino, "Online learning for automatic segmentation of 3d data," in *Proc. of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4857–4864, Sep. 2011.

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

[151] A. Borji and L. Itti, "Cat2000: A large scale fixation dataset for boosting saliency research," *Proc. of the CVPR 2015 workshop on "Future of Datasets"*, 2015. arXiv preprint arXiv:1505.03581.

[152] G. Anders and D. Tong, "Depth post-processing for intel® realsense™ d400 depth cameras." https://www.mouser.com/pdfdocs/Intel-RealSense-Depth-PostProcess.pdf. Accessed: 2019-04-20.

[153] A. Atapour-Abarghouei and T. Breckon, "A comparative review of plausible hole filling strategies in the context of scene depth image completion," *Computers & Graphics*, vol. 72, pp. 39–58, 2018.

[154] J. Shi, Q. Yan, L. Xu, and J. Jia, "Hierarchical image saliency detection on extended cssd," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 38, p. 717–729, Apr. 2016.

[155] Y. Cheng, H. Fu, X. Wei, J. Xiao, and X. Cao, "Depth enhanced saliency detection method," in *Proc. of International Conference on Internet Multimedia Computing and Service*, (New York, NY, USA), pp. 23:23–23:27, ACM, 2014.

[156] Y. Chuan, Z. Lihe, L. Xiang, and Y. Ming-Hsuan, "Saliency detection via graph-based manifold ranking," in *Proc. of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3166–3173, IEEE, 2013.

[157] H. J. Seo and P. Milanfar, "Static and space-time visual saliency detection by self-resemblance," *Journal of Vision*, vol. 9, pp. 15–15, 11 2009.

[158] N. Murray, M. Vanrell, X. Otazu, and C. A. Parraga, “Saliency estimation using a non-parametric low-level vision model,” in *Proc. of the 2011 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 433–440, 2011.

[159] X. Hou and L. Zhang, “Saliency detection: A spectral residual approach,” in *Proc. of the 2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[160] E. Erdem and A. Erdem, “Visual saliency estimation by nonlinearly integrating features using region covariances,” *Journal of Vision*, vol. 13, pp. 11–11, 03 2013.

[161] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä, “Segmenting salient objects from images and videos,” in *Proc. of the 2010 European Conference on Computer Vision ECCV 2010* (K. Daniilidis, P. Maragos, and N. Paragios, eds.), (Berlin, Heidelberg), pp. 366–379, Springer Berlin Heidelberg, 2010.

[162] L. Duan, C. Wu, J. Miao, L. Qing, and Y. Fu, “Visual saliency detection by spatially weighted dissimilarity,” in *Proc. of the 2011 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 473–480, 2011.

[163] H. Rezazadegan Tavakoli, E. Rahtu, and J. Heikkilä, “Fast and efficient saliency detection using sparse sampling and kernel density estimation,” in *Image Analysis* (A. Heyden and F. Kahl, eds.), (Berlin, Heidelberg), pp. 666–675, Springer Berlin Heidelberg, 2011.

[164] S. Goferman, L. Zelnik-Manor, and A. Tal, “Context-aware saliency detection,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2376–2383, 2010.

[165] T. Zhao and X. Wu, “Pyramid feature attention network for saliency detection,” in *Proc. of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3080–3089, 2019.

[166] A. Criminisi, P. Peréz, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *IEEE Transactions on Image Processing*, vol. 13, pp. 1200–1212, Sept. 2004.

[167] X. Wu, K. Xu, and P. Hall, “A survey of image synthesis and editing with generative adversarial networks,” *Tsinghua Science and Technology*, vol. 22, pp. 660–674, December 2017.

[168] Y. Lin, W. Chen, and Y. Chuang, “Bedsr-net: A deep shadow removal network from a single document image,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[169] L. Zhou, Z. Yang, Z. Zhou, and D. Hu, “Salient region detection using diffusion process on a two-layer sparse graph,” *IEEE Transactions on Image Processing*, vol. 26, pp. 5882–5894, Dec. 2017.

[170] C. Xia, J. Li, X. Chen, A. Zheng, and Y. Zhang, “What is and what is not a salient object? learning salient object detector by ensembling linear exemplar regressors,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4399–4407, July 2017.

[171] R. Achanta, S. S. Hemami, F. J. Estrada, and S. Süsstrunk, “Frequency-tuned salient region detection,” *in Proc. of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1597–1604, 2009.

[172] X. Hou and L. Zhang, “Saliency detection: A spectral residual approach,” in *Proc. of the 2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2007.

[173] N. Murray, M. Vanrell, X. Otazu, and C. A. Parraga, “Saliency estimation using a non-parametric low-level vision model,” in *Proc. of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 433–440, June 2011.

[174] Z. Liu, W. Zou, and O. L. Meur, “Saliency tree: A novel saliency detection framework,” *IEEE Transactions on Image Processing*, vol. 23, pp. 1937–1952, May 2014.

[175] C. Aytekin, S. Kiranyaz, and M. Gabbouj, “Automatic object segmentation by quantum cuts,” in *Proc. of the 2014 22nd International Conference on Pattern Recognition*, pp. 112–117, Aug. 2014.

[176] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, “Salient object detection: A discriminative regional feature integration approach,” in *Proc. of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2083–2090, June 2013.

[177] H. Fu, D. Xu, S. Lin, and J. Liu, “Object-based rgbd image co-segmentation with mutex constraint,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4428–4436, June 2015.

[178] M. Peris, S. Martull, A. Maki, Y. Ohkawa, and K. Fukui, “Towards a simulation driven stereo vision system,” in *Proc. of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 1038–1042, Nov 2012.

[179] J. Zhang, F. Malmberg, and S. Sclaroff, *Visual Saliency: From Pixel-Level to Object-Level Analysis.* Springer, 01 2019.