A Hybrid Image Captioning Architecture with Instance Segmentation and Saliency Prediction

Chantelle Saliba

Supervisor: Dr. Dylan Seychell

June, 2022

Submitted in partial fulfilment of the requirements for the degree of M.Sc. in Artificial Intelligence.

L-Università ta' Malta
Faculty of Information &
Communication Technology

Acknowledgements

I would like to express my deepest gratitude to my supervisor Dr Dylan Seychell for his guidance, invaluable advice and continuous support throughout my academic journey. His knowledge and experience have encouraged me to push my limits and realize my potential. Moreover, I also thank my family and closest friends, especially my mother for their endless support, unceasing encouragement and understanding and without whom this journey wouldn't have been possible.

Abstract

In recent years, image captioning has increased in its populace as identified by the surge of research in this area amongst the Artificial Intelligence community. Recognising its potential as an assistive technology, a novel framework is being presented that makes use of a hybrid architecture compromising of a convolutional neural network in addition to novel image and language transformers. Reviewing the current state-of-the-art technologies a rich encoder was constructed aiming to extract information at both object and scene level. This was achieved by an amalgamation of an instance segmentation technique and a saliency predictor to identify objects within a visual scene in addition to a scene classifier to determine environmental factors. Features extracted from the concatenation of a vision hybrid transformer used for the former and a convolutional neural network used for the latter are then progressed through a dedicated image-to-sequence language transformer for the construction of the architecture. The pipeline presented influenced by rich literature is constructed argumentatively and utilises a modular framework, therefore providing an opportunity for modernisation and improvement of results. Furthermore, the discussed pipeline facilitates the future explainability of image captioning architectures in addition to focusing on a more efficient training strategy. This novel architecture was benchmarked on the Flickr8K and the Flickr30K and has managed to achieve comparable and even-so exceeding results on several metrics with the current state-of-the-art architectures while attaining the above advantages. This research strives to contribute to the improvement of image captioning and review current state-of-the-art techniques such as instance segmentation and scene classification whilst also identifying the potential of saliency prediction as an attention mechanism, in addition to focusing on the readability of the sentences generated.

Contents

List of Figures **vii**

List of Tables **ix**

1 Introduction **1**

1.1 Problem Definition . . . 1

1.2 Motivation . . . 2

1.3 Aim and Objectives . . . 2

1.4 Proposed Solution . . . 3

1.5 Contribution . . . 4

1.6 Document Structure . . . 5

1.6.1 Part 1: Background Research and Literature Review . . . 5

1.6.2 Part 2: Methodology . . . 6

1.6.3 Part 3: Evaluation . . . 6

1.7 Conclusion . . . 7

2 Background **8**

2.1 Visual Impairment and Readability . . . 8

2.2 Visual Saliency . . . 10

2.2.1 Saliency Detection Techniques . . . 10

2.2.2 Saliency Ranking . . . 13

2.3 Conclusion . . . 15

3 Literature Overview **16**

3.1 Image Captioning . . . 16

3.2 Instance Segmentation . . . 22

3.2.1 Mask Region-Based Convolutional Neural Network (Mask R-CNN) . 22

3.2.2 Yolact (You Only Look At CoefficienTs) . . . 24

3.2.3 Yolact ++ (You Only Look At CoefficienTs ++) . . . 24

3.2.4 CenterMask . . . 24
3.2.5 TensorMask . . . 25
3.3 Scene Classification . . . 25
3.4 Dedicated Systems . . . 27
3.5 Conclusion . . . 28

4 Methodology 29
4.1 System Overview . . . 29
4.1.1 Methodology Structure . . . 32
4.2 Instance Segmentation Analysis Procedure . . . 32
4.2.1 System Overview . . . 32
4.2.2 Datasets . . . 34
4.2.3 Metrics . . . 36
4.3 Saliency Prediction for Weighted Masks Analysis Procedure . . . 37
4.3.1 System Overview . . . 37
4.3.2 Metrics . . . 38
4.4 Scene Classification Analysis Procedure . . . 39
4.4.1 System Overview . . . 39
4.4.2 Dataset . . . 41
4.4.3 Metrics . . . 41
4.5 Image Captioning Architecture . . . 41
4.5.1 The Encoder - System Overview . . . 42
4.5.2 The Decoder - System Overview . . . 47
4.5.3 Datasets . . . 51
4.5.4 Metrics . . . 51
4.5.5 Training and Evaluation . . . 53
4.6 Conclusion . . . 55

5 Evaluation 56
5.1 Evaluation Overview . . . 56
5.1.1 Evaluation Structure . . . 56
5.2 Image Captioning Model Encoder . . . 58
5.2.1 Instance Segmentation Review . . . 58
5.2.2 Saliency Prediction Review . . . 65
5.2.3 Scene Classification Review . . . 72
5.3 Image Captioning Model . . . 73
5.3.1 Text Encodings . . . 74

5.3.2 Model Hyper-Parameters 76
5.3.3 Results 80
5.4 Sentence Generation and Readability 84
5.4.1 Exploring the Sentence Length 84
5.4.2 Impact on the Image Captioning Architecture 87
5.5 Conclusion 89

6 Conclusion **91**
6.1 Summary 91
6.2 Achieving the Objectives 93
6.3 Future Work 94

References **95**

List of Figures

2.1 Variations of Saliency Detection Algorithms 12
2.2 Saliency Ranking using Sara algorithm Seychell and Debono (2018) 14

3.1 Encoder-Decoder Architecture of Deep Image Captioning (Hrga and Ivašić-Kos, 2019) 17
3.2 Image Transformer Architecture proposed by Dosovitskiy et al. (2021) 18
3.3 Language Transformer Architecture proposed by Vaswani et al. (2017) 20
3.4 Mask Captioning Network proposed by Lim and Chan (2019) 21
3.5 The Evolution of Object Detection from Image Classification to Segmentation Techniques (Wilson, 2019) 23

4.1 A High-Level Overview of the Proposed Architecture 29
4.2 Block Diagram of the Proposed Architecture 31
4.3 Block Diagram of the Instance Segmentation Analysis 33
4.4 Distribution of the MS-COCO17 Dataset 35
4.5 Distribution of the Tiny Pascal VOC Dataset 36
4.6 Block Diagram of the Saliency Prediction Analysis 38
4.7 Block Diagram of the Scene Classification Analysis 40
4.8 Block Diagram of the Encoder 42
4.9 Difference between utilising a Saliency Ranking algorithm and the Confidence Score of an Instance Segmentation algorithm as an Attention Mechanism . . . 44
4.10 Image Captioning Encoder - Object Layer Processing 46

5.1 Block Diagram of the Evaluation Plan 57
5.2 Sample Instance Segmentation Inference Images 59
5.3 Comparing Instance Segmentation Architectures on the MS-COCO17 with the Tiny Pascal VOC 60
5.4 In-depth Evaluation of Instance Segmentation Architectures per class using the MS-COCO17 Val dataset 62

5.5 In-depth Evaluation of Instance Segmentation Architectures per class using the Pascal Tiny VOC Training dataset 63
5.6 A graph of the Average AP score against the Average Inference Time 64
5.7 A Sample Non-complex Image with its Binary Mask and Saliency Ranking Result 66
5.8 A Sample Non-Complex Image with its Saliency Map, Weighted Mask and Weighted Image for each considered Saliency Prediction Algorithm 67
5.9 A Sample Complex Image with its Binary Mask and Saliency Ranking result . . 68
5.10 A Sample Complex Image with its Saliency Map, Weighted Mask and Weighted Image for each considered Saliency Prediction Algorithm 69
5.11 Investigating Loss of Data caused by the Saliency Prediction Algorithms 71
5.12 Evaluation of Scene Classification Architectures 73
5.13 A Graph of Training loss against Epochs for each Encoding 75
5.14 A Graph of Validation loss against Epochs for each Encoding 75
5.15 Loss against Epochs for the Flickr8K with a 0.1 Dropout 77
5.16 Loss against Epochs for the Flickr8K with a 0.5 Dropout 77
5.17 Loss against Epochs for the Flickr30K with a 0.1 Dropout 78
5.18 Loss against Epochs for the Flickr30K with a 0.5 Dropout 78
5.19 Sample Images captioned using the Architecture presented 83

List of Tables

5.1 Instance Segmentation Evaluation on the Validation Set of the MS-COCO17 and the Training Set of the Tiny Pascal VOC 60

5.2 Evaluation of Inference Time per Image for each dataset for Instance Segmentation Architectures 63

5.3 Comparing the percentage of non-black pixels within the binary mask generated by an Instance Segmentation (IS) algorithm, the Saliency Predictor map (SP) and the Weighted Mask (WM) 70

5.4 Comparing the Number of Contours within the Binary Mask generated by an Instance Segmentation (IS) algorithm, the Saliency Predictor (SP) map and the Weighted Mask (WM). 71

5.5 Evaluation of the Pre-Trained Scene Classification Models on the Validation Set of the Places-365 Standard 72

5.6 Comparing the Performance of the Image Captioning Architecture on different Embeddings 75

5.7 Comparing the Performance of the Image Captioning Architecture with different Dropout rates on the Flickr8K Dataset 77

5.8 Comparing the Performance of the Image Captioning Architecture with different Dropout rates on the Flickr30K Dataset 78

5.9 Comparing the Performance of the Image Captioning Architecture with varying Layers on the Flickr8K Dataset 79

5.10 Comparing the Performance of the Image Captioning Architecture with varying d_model and dff on the Flickr8K Dataset 80

5.11 Comparing the Performance of the Image Captioning Architecture on the Flickr30K dataset on the Bleu (B), Meteor (M), Rouge-L (R-L) and CIDEr (C) metrics 81

5.12 Statistics Gathered on the Target Captions of the Flickr8K and Flickr30K Datasets 85

5.13 The Effect of the Vocabulary and the Maximum Sentence Length on the Generated Captions on the Flickr8K using a Batch Size of 64 85

5.14 The Effect of the Vocabulary and the Maximum Sentence Length on the Generated Captions on the Flickr30K using a Batch Size of 64. 86
5.15 Comparing the Proposed Image Captioning Performance on the Flickr8K trained with a Batch Size of 64 when varying the Vocabulary Size and the Training Caption Length. 87
5.16 Comparing the Proposed Image Captioning Performance on the Flickr30K trained with a Batch Size of 64 when varying the Vocabulary Size and Training Caption Length. 88

Chapter 1

Introduction

1.1 Problem Definition

Image captioning is a revolutionary field in Artificial Intelligence incorporating two major areas of this field: Artificial Vision and Natural Language Processing. Researchers have explored image captioning for numerous years developing a variety of architectures to contribute to this growing field. Recognising its potential in intriguing applications such as accessibility for the visually impaired, this research focuses on achieving scene understanding through the use of image captioning techniques by exploring the latest state-of-the-art instance segmentation (Hafiz and Bhat, 2020) techniques and a hybrid architecture consisting of transformers (Dosovitskiy et al., 2021; Vaswani et al., 2017) and pre-trained convolutional neural networks (CNN).

Although research on image captioning is abundant, few works have delved into instance segmentation as an attention mechanism. In fact, Lim and Chan (2019) were the first researchers to introduce instance segmentation to image captioning as an attention mechanism achieving comparable results to the current state-of-the-art. Since then, there has been no recognisable work that exclusively used instance segmentation as an attention mechanism for image captioning. This research bases its core methodology on the works of Lim and Chan delving argumentatively into any potential shortcomings and discussing solutions and alterations to this methodology through the implementation of the presented image captioning architecture. Through these arguments, saliency prediction algorithms are being introduced as a further enhancement to instance segmentation to form part of the attention mechanism.

Moreover, image transformers (Dosovitskiy et al., 2021) in the context of the encoder mechanism in the field of image captioning are fairly unexplored. The image transformer (Dosovitskiy et al., 2021) was firstly introduced in the International Conference on Learning Representations in 2021 and research suggests that the image transformer holds immense unknown potential. In the area of image captioning Liu et al. (2021) were the first researchers to leverage transformers to develop a fully transformer-based architecture in which they furtherly concluded the simplicity and exceptional potential of image transformers. This study explores a hybrid technology architecture for the encoder and the decoder consisting of a blend of transformers and traditional CNNs.

Inspired by the potential of this area for accessibility, this research also considers the possibility of minor alterations to the image captioning architecture to facilitate the understandability of captions generated. According to the World Health Organization (2019) there are at least 2.2 billion individuals that have a form of visual impairment with cases constantly raising due to an ageing population, population growth, urbanisation, as well as lifestyle changes. One of the most common feats experienced by the visually impaired is gathering an understanding of their surroundings, a feat that can be aided through image captioning architectures. This dissertation explores techniques based on work on readability, minor modifications such as varying training hyper-parameters.

1.2 Motivation

The motivation behind this research can be analysed from two perspectives: the evolution of image captioning methodologies and identifying the potential of image captioning as an assistive technology to enhance readability for individuals with special needs. Artificial intelligence is a growing field with novel state-of-the-art technologies constantly being introduced. Therefore, one of the motives behind this study is to exploit these developments, mainly instance segmentation and image transformers, for the evolution of image captioning architectures. In addition, this study strives to implement a modular framework in which any module could be replaceable with novel technology for improved results. Moreover, artificial intelligence has improved drastically the quality of life of individuals. Therefore a further motive of this study is to analyse how image captioning techniques can improve readability for people with special needs particularly those with a form of visual impairment by potentially performing minor alterations to the vocabulary and the training hyper-parameters.

1.3 Aim and Objectives

This research aims to contribute to the field of image captioning by developing a modular hybrid image captioning architecture that is constructed argumentatively and is influenced by current research with the intent to improve the known black box state of the art architecture by providing explainability and an efficient training strategy.

To achieve this aim the following objectives have been set:

1. Evaluate current state-of-the-art instance segmentation techniques and saliency prediction algorithms to explore their potential contribution as an attention mechanism for image captioning.

2. Explore research in the field of image captioning that utilises segmentation as an attention mechanism, identify any potential shortcomings and discuss improvements in reference to available research.

3. Implement a hybrid image captioning architecture that consists of an encoder and decoder that incorporates image and language transformers and CNNs while evaluating it against the current state of the art techniques.

4. Explore the impact of the vocabulary size and the maximum trained sentence length on the performance of the image captioning architecture.

1.4 Proposed Solution

The proposed solution for this project scrutinises current research and performs reviews to implement a hybrid image captioning architecture that bases its attention mechanism on instance segmentation and saliency prediction. This study begins with a review of current state-of-the-art instance segmentation algorithms as well as current saliency prediction techniques. This research furtherly delves into the relationship between the best performing segmentation algorithm and the different saliency algorithms considered to generate images with attention distribution based on saliency. Following, a review of scene classification models is conducted to provide an insight into this area and its potential contribution to providing significant information about visual imagery. The image captioning architecture proposed is rooted in the conclusions drawn from these reviews in addition to being influenced by current work in this field particularly the Mask Captioning Network introduced by Lim and Chan (2019). The proposed architecture is built on an encoder-decoder structure with a hybrid encoder consisting of an object layer and a scene layer. The former layer gathers information regarding the entities making up the image whilst the latter layer highlights the surroundings and atmosphere of the visual image. This technique inspired by Lim and Chan gives a holistic representation of the image presented. This implementation strays from the Mask Captioning Network by introducing saliency prediction to aid in the visual attention mechanism whilst also utilising dedicated scene classification models. Moreover, this architecture dabbles in hybrid vision transformers for feature extraction in addition to employing a transformer decoder. As part of this research, the vocabulary size of the corpus and the maximum sentence length the image captioning model is being trained on are being varied to survey the impact of these variables on the performance of the model on metrics such as Meteor (Denkowski and Lavie, 2014), Rouge-L (Lin, 2004), Cider (Vedantam et al., 2015) and Bleu (Papineni et al., 2002).

1.5 Contribution

The main contributions of this research can be summarised in the following points:

1. An Image Captioning Model - This research, based on the previous works introduces a novel hybrid image captioning model that bases its attention mechanism on instance segmentation and saliency prediction in addition to utilising novel technologies such as the hybrid vision transformer. This model manages to exceed the performance of current research on most metrics whilst achieving a comparable result to the Mask Captioning Network (Lim and Chan, 2019) with some metrics favouring the implemented architecture. In addition, this framework focuses on providing a modular structure for future innovation and explainability.

2. A Review on Instance Segmentation Architectures - In this research, a review of current state-of-the-art instance segmentation techniques including the Mask R-CNN (Lim and Chan, 2019), Yolact (Bolya et al., 2019), Yolact++ (Bolya et al., 2020b), TensorMask (Chen et al., 2019) and CenterMask (Lee and Park, 2020) is being implemented comparing these architectures in terms of performance and inference time.

3. A Review of pre-trained Scene Classification Models - This research presents a review of pre-trained scene classification models on the Places365 Dataset (Zhou et al., 2017) consisting of AlexNet (Krizhevsky et al., 2012), ResNet-18, ResNet-50 (He et al., 2016) and DenseNet-161 (Huang et al., 2017) measuring the model's performance and the inference time.

4. Usability of Saliency as an Attention Mechanism - This research explores saliency algorithms mainly the EML-Net (Jia and Bruce, 2020), Deep Gaze II (Kümmerer et al., 2016), Pyramid Feature Attention Network for Saliency Prediction (Zhao and Wu,

2019) in addition to the traditional non-deep saliency algorithm proposed by Itti *et al.* (Itti et al., 1998) in relation to the masks generated by the instance segmentation algorithm to distinguish the importance of objects within an image and as a byproduct act as an attention mechanism.

1.6 Document Structure

This research is constructed of three main parts. The first part presents previous work conducted in the main areas this research is developed on. In the second part, the methodology that is based on the research explored in the first part is analysed in detail. The third and final part consists of the evaluation of the system with a discussion of takeaways and potential future work.

1.6.1 Part 1: Background Research and Literature Review

The first part of this research provides a background for the work being conducted and highlights influential work developed by other researchers in the main components of the proposed architecture. The main contributing areas explored in this area consist of image captioning techniques, instance segmentation algorithms, saliency prediction and scene classification architectures.

1. Visual Impairment and Readability - This section provides an overview of the definition of a visual impairment and the social and economic repercussions that stem from it. This chapter delves deeper into guidelines for creating digital content that is accessible and more importantly measures to make this content more understandable and readable.

2. Saliency - In this section, the definition of saliency is provided along with an introduction to attention mechanisms. Research related to different visual attention mechanisms along with the progression of saliency algorithms is presented in this chapter.

3. Image Captioning - This section explores the image captioning architecture and explores the most distinguishable research in this area with special consideration to research incorporating transformers and segmentation.

4. Instance Segmentation - This section defines instance segmentation as an advanced object detection methodology and explores state-of-the-art algorithms such as Mask R-CNN, CenterMask, TensorMask, Yolact and Yolact++.

5. Scene Classification - In this section, scene classification is presented as a niche of image classification, exploring relevant related research in terms of datasets, architectures and models.

6. Similar Systems - This section delves into systems targeted at people with a form of visual impairment that aids accessibility.

1.6.2 Part 2: Methodology

The second part of this document discusses the proposed architecture based on the research conducted in the previous section. This system is not only developed on previous research but also performs reviews on current state-of-the-art architectures presented.

1. Instance Segmentation Review - This section discusses the current state-of-the-art instance segmentation algorithms and the methodology employed to compare the architectures. The best performing architecture will then be used in the encoder of the image captioning architecture.

2. Saliency Prediction Review - Complementary to the instance segmentation algorithm is the saliency prediction algorithm to generate weighted masks. Therefore, this section presents the methodology of the review of some saliency prediction algorithms for the application of generating weighted masks to act as an attention mechanism.

3. Scene Classification Review - In this section, scene classification is explored in relation to four different pre-trained models on a dedicated dataset. The best overall model will be used to construct the encoder of the image captioning model.

4. Image Captioning Architecture - This section explores the architecture being implemented based on previous research with argumentative discussions on the improvements being proposed. Here, the encoder and the decoder are delved into in detail.

1.6.3 Part 3: Evaluation

The final section of this research focuses on the evaluation of the reviews and the architecture discussed in the methodology as well as proposes future improvements. This section concludes the research conducted by drawing results from the figures generated and discussing the findings. The evaluation also targets the readability component of this architecture by discussing the effect of the vocabulary size and the sentence length of the generated captions.

1. Image Captioning Encoder - This section presents the evaluation of the reviews conducted during the construction of the encoder of the architecture. Therefore, this section contains the findings and conclusions drawn from the reviews of the instance segmentation algorithms, saliency prediction and scene classification models.
2. Image Captioning Architecture - In this section, the evaluation conducted from the training of the image captioning architecture will be presented and discussed in comparison to researched architectures.
3. Image Captioning Readability - This section is dedicated to the readability component of this research exploring the performance of the image captioning model when varying the vocabulary size and the sentence length of the generated captions.

1.7 Conclusion

This chapter introduced the problem targeted in this dissertation along with the motivation behind this work. To begin this research the aims and objectives were presented in addition to providing an overview of the proposed framework. Furthermore, this section presented the contributions together with an overview of the chapters found in this dissertation.

Chapter 2

Background

This chapter introduces the context this research is based. One of the motivations behind this research is the accessibility that the proposed image captioning architecture would be able to provide to people with a visual impairment. Therefore, the first part of this chapter is dedicated to defining a visual impairment, providing an overview of the different variations and discussing also methodologies to overcome these limitations in technology. Furthermore, research conducted on readability is discussed. This chapter then continues to introduce visual saliency providing an overview of saliency detection techniques followed by an introduction to saliency ranking.

2.1 Visual Impairment and Readability

A visual impairment (Naipal and Rampersad, 2018) can be described as any form of reduced visual performance that cannot be corrected through the use of medical procedures or else through standard refractive correction. According to the World Health Organization (2019) there are at least 2.2 billion individuals that have a form of visual impairment with cases constantly on the rise due to an ageing population, population growth, urbanisation, as well as lifestyle changes.

Visual impairment can be of varying degrees and is generally classified as mild, moderate, severe or blindness (Naipal and Rampersad, 2018) according to the level of severity of the impairment. Visual impairment is a broad term that comprises different variations such as blurred vision, colour vision, contrast and light sensitivity and loss of vision (World Wide Web Consortium, 2016). Visual acuity describes the sharpness of vision and this type of impairment can sometimes be corrected through the use of refractive correction, however, in some cases, an individual's vision might remain unfocused. Another form of visual impairment is light sensitivity in which bright light makes it difficult for individuals to read or focus. Differently, contrast sensitivity is when one finds it challenging to distinguish between contrasting areas for example distinguishing the text from the background. Visual field loss impairment is then experienced when an individual does not have a full field of vision. Different variations of this impairment consist of central visual field loss, tunnel vision, scattered patches of vision and left or right visual field loss. Colour vision, commonly referred to as colour blindness, is when an individual is unable to interpret and see certain specific colours.

A visual impairment has implications that are beyond physical. In fact, research shows that visual impairments have a social and economic impact (Naipal and Rampersad, 2018; World Health Organization, 2019). Vision impairment in young children generally results in a delay in development which can lead to future implications (Naipal and Rampersad, 2018). An adult with a vision impairment experiences a higher probability of developing mental health disorders mainly depression and anxiety whilst older adults are more likely to be subjected to social isolation. Economically, visual impairments also pose a significant burden (World Health Organization, 2019). To overcome the obstacles and challenges posed by visual impairments, accessibility measures are generally proposed and encouraged to be adhered to.

The World Wide Web Consortium (2016) proposed a set of guidelines to introduce accessibility to technologies and electronic content for individuals with a visual impairment. Some of the techniques that can be utilised to overcome these challenges include consideration to the text size, font family, style, capitalisation and interface. Spacing also plays an important role in perceiving individual words and sentences from each other, therefore factors such as line spacing, letter spacing, word spacing, element spacing, text alignment and margins should also be important considerations. Further contributing factors are brightness and contrast which variations aid people with visual impairments, particularly individuals with light and contrast sensitivity. Moreover, an application should not rely on colour information since colour is not perceived the same by people with visual impairments. Additional considerable features consist of the line length and hyphenation. Preferably, an individual should be able to adjust and change these features in order for the text and non-textual information to be better represented and perceived.

Apart from making content more accessible, measures can be applied to make it more understandable. Kadayat and Eika (2020) delved deeper into this concept by analysing the impact of the sentence length on the comprehensibility of users with a form of visual impairment that depend on the use of screen readers to get information. From this research, they concluded the significant impact of the sentence length with the optimal sentence length being between 16 and 20 words observing the lowest workload and highest understanding.

2.2 Visual Saliency

Visual saliency (Borji, 2018; Borji et al., 2014; Le Meur et al., 2006; Seychell, 2021) is a deeply researched area in the field of computer vision which aims to identify the most prominent segments in a scene. This section firstly provides an overview of different saliency detection mechanisms discussing how these algorithms generate an 8-bit saliency map to provide a level of saliency for every pixel in the input image. Following, this section continues to delve into saliency ranking to which the focus shifts to distinguishing the rank of the pixels in an image with respect to their level of saliency.

2.2.1 Saliency Detection Techniques

Visual saliency has its roots in the human visual system and bases its mechanics on how humans perform visual attention. Visual attention describes how cognitively the human brain reduces the information being received from the visual system by focusing on a single small area whilst blurring its surroundings (Seychell, 2021). Attention mechanisms, inspired by the human visual system can be classified as either utilising a task-agnostic or else a task-specific approach (Borji, 2018). A task-agnostic attention mechanism also referred to as the bottom-up approach refers to features that involuntary catch the viewer's attention (Le Meur et al., 2006). Differently, task-specific approaches also referred to as the top-down approach refer to tasks driven by some type of recognition goal (Borji et al., 2014).

Most works in the area of visual attention focus on bottom-up saliency with the work of Itti et al. (1998) inspiring the initial surge of research on visual saliency. Following the bottom-up approach, Itti et al. (1998)'s saliency prediction algorithm focuses on low-level features to distinguish the foreground from the background. This algorithm considers intensity, colour and orientation features which when combined and normalised allow for the extraction of simple feature maps. Following a linear combination, a preliminary saliency map is generated to which a winner-takes-all neural network is used for the processing of the saliency map. Due to its logical and parallel structure, this algorithm provides a saliency score to every pixel efficiently which as will be identified in succeeding sections, particularly Section 5.2.2, makes it the ideal algorithm for this research. Other well-renowned classic architectures consist of the Boolean Map based Saliency (BMS) (Zhang and Sclaroff, 2013) and EYe MOvement Laws (EYMOL) (Zanca and Gori, 2017). BMS is a bottom-up saliency model developed by Zhang and Sclaroff (2013) that generates the saliency map of an image based on the topological structure of Boolean maps. Despite its simplicity, BMS has also proven to be useful in salient object detection.

Meanwhile, EYMOL (Zanca and Gori, 2017) focuses its attention mechanism on the law principles of the movement of the human eye. Zanca and Gori (2017) discussed that humans are attracted by regions within an image that contains intricate details in addition to brightness invariance which they analysed leads to fixation and motion tracking. This theory represented mathematically focuses on generating a model that represents eye movements and calculates the saliency map as a byproduct of this theory.

With the revolution of convolutional neural networks (CNN) (Bezdan and Bacanin, 2019), saliency models have seen noticeable improvements over the classic saliency models. Researchers have analysed trained CNNs on scene recognition as well as introduced novelties to develop enhanced saliency predictors. Generally, these architectures are trained on large-scale datasets and are then fine-tuned on smaller specialised datasets such as eye-movement or click datasets (Borji, 2018). Some of the more well-known architectures are the DeepGaze II (Kümmerer et al., 2016), Expandable Multi-Layer NETwork (EML-Net) (Jia and Bruce, 2020), Saliency Attentive Model - ResNet (SAM-ResNet) (Cornia et al., 2018) and the Pyramid Feature Attention Network for Saliency Prediction Zhao and Wu (2019). The DeepGaze II (Kümmerer et al., 2016) is based on its precedent the DeepGaze I (Kümmerer et al., 2014) and utilises transfer learning to adjust a VGG-19 (Simonyan and Zisserman, 2015) neural network trained to perform object detection to perform saliency prediction. Some important features of this architecture are that the features of the VGG are not retrained as well as the architecture is trained in a probabilistic framework. Jia and Bruce introduce the EML-NET (Jia and Bruce, 2020) which consists of an encoder-decoder architecture in which the encoder and the decoder are separately trained. In addition, the encoder can have multiple CNN models to extract the features with models being able to have different architectures and be trained on different datasets. The SAM-ResNet architecture developed by Cornia et al. (2018) incorporates neural attentive mechanisms through the use of a convolutional Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). Differently, the Pyramid-Feature Attention Network presented by Zhao and Wu (2019) exploits a top-down approach and the technology of feature pyramids to extract high-level semantic feature maps and low-level spatial structural features at different scales. An overview of the discussed algorithms could be analysed in Figure 2.1 to which a demonstration of the techniques is provided by showing the generated saliency map for each technique.

To analyse the progression of saliency models, benchmarking evaluation is carried out on new saliency models. For image-based saliency models, the most common are the MIT (Bylinskii et al.) and the Saliency in Context (SALICON) (Jiang et al.) benchmark. The former is currently considered the gold standard and bases its evaluation on eye movements on the MIT300 (Judd et al., 2012) and the CAT2000 (Borji and Itti, 2015)

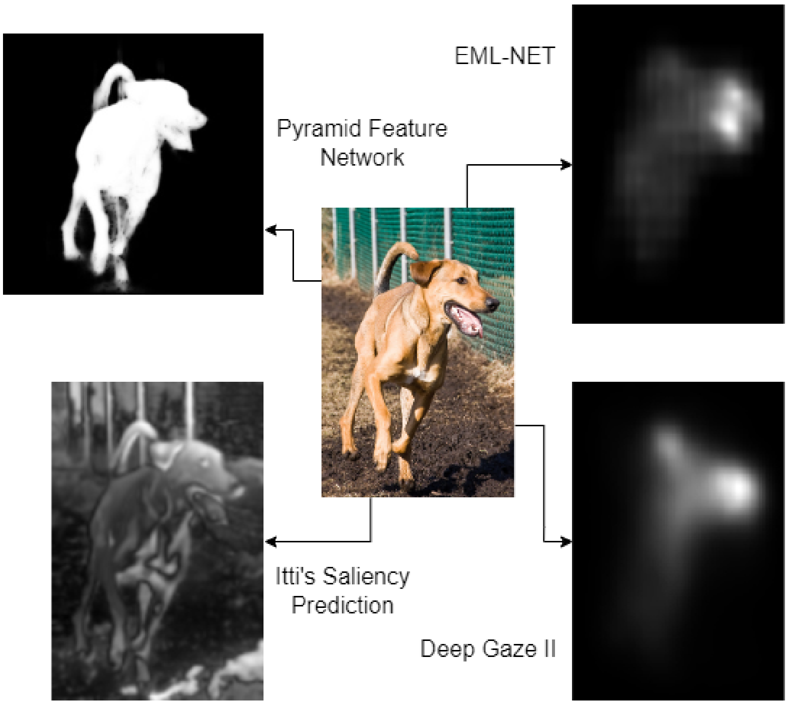


Figure 2.1: Variations of Saliency Detection Algorithms

datasets. The MIT300 (Judd et al., 2012) dataset consists of 300 natural images with recorded eye movements from 39 different observers. Whilst, the CAT2000 (Borji and Itti, 2015) dataset consists of 4000 images distributed evenly over 20 categories covering scenes such as indoor, outdoor, art and cartoons with the eye-tracking data of 120 observers. The latter benchmark, the SALICON (Jiang et al.) benchmark mainly considers the SALICON (Jiang et al., 2015) dataset. The SALICON (Jiang et al., 2015) dataset consists of saliency annotations on the Microsoft Common Objects in Context (MS-COCO) (Lin et al., 2014) collected through mouse movements using a neurophysiological and psychophysical paradigm referred to as the mouse-contingent multi-resolutional paradigm that is based on research on the human visual system.

Borji (2018) discussed the progression of visual saliency and performed a thorough review of a considerate amount of saliency architectures, evaluating their performance over the benchmarks mentioned before. From this review on the MIT300 dataset (Judd et al., 2012) using the MIT300 benchmark (Bylinskii et al.), Borji concluded that on all evaluation measures the best five performing models were all deep visual saliency models. According to the evaluation metric, the best performing model varied however it could be concluded that EML-NET (Jia and Bruce, 2020) was the only model that achieved the best performance over three metrics whilst DeepGaze II (Kümmerer et al., 2016) and DPNSal

(Oyama and Yamanaka, 2018) achieved the best performance over two metrics. When considering traditional non-deep models, it could be concluded that the BMS (Zhang and Sclaroff, 2013) was the best performing model. This evaluation was also replicated on the CAT2000 (Borji and Itti, 2015) dataset in which it was concluded that overall the models that performed well on the MIT300 (Judd et al., 2012) dataset also performed well on this dataset. As before, the best performing model varied according to the evaluation metric considered. However, it could be identified that the SAM-ResNet (Cornia et al., 2018) architecture gave the best performing model over four measures. Considering non-deep networks, the BMS (Zhang and Sclaroff, 2013) and EYMOL (Zanca and Gori, 2017) performed the best overall out of all the other models considered. Borji also analysed some architectures over the SALICON benchmark using recent publications. This analysis was rather limited but it was concluded that the models still underperformed humans with the EML-NET (Jia and Bruce, 2020) and DeepGaze II (Kümmerer et al., 2016) having the best performance over a singular metric whilst SAM-ResNet (Cornia et al., 2018) gave the best performing model over two metrics.

Deep saliency visual models have seen great improvements over the traditional classical models because of the models' ability to extract higher-level features from images. Research has however shown that in some circumstances the traditional saliency models surpass deep saliency models highlighting how current deep models fail to explain low-level saliency. Although deep visual saliency models out-perform classical models in most cases, these nonetheless still underperform humans (Borji, 2018).

2.2.2 Saliency Ranking

Saliency ranking is a closely related field to saliency detection with the additional capability to determine the succession of the most important regions within the image which is particularly useful for multi-object images. Although saliency ranking is not directly implemented as part of this research, saliency ranking for this research is being used to drive arguments in building the logic for the methodology of the proposed architecture.

The first saliency ranking algorithm discussed is the Sara algorithm developed by Seychell and Debono (2018). This algorithm employs a grid approach with the texture image, a depth frame and the number of segments required for the grid template. Following, a saliency map using Itti's Saliency Algorithm (Itti et al., 1998) is generated from the texture image provided. This saliency map and the depth map are then segmented according to the inputted number. These segments are then processed individually to compute scores that reflect the entropy score of the saliency map, the proximity to the centre and the depth score that when combined return a single score that represents the most salient

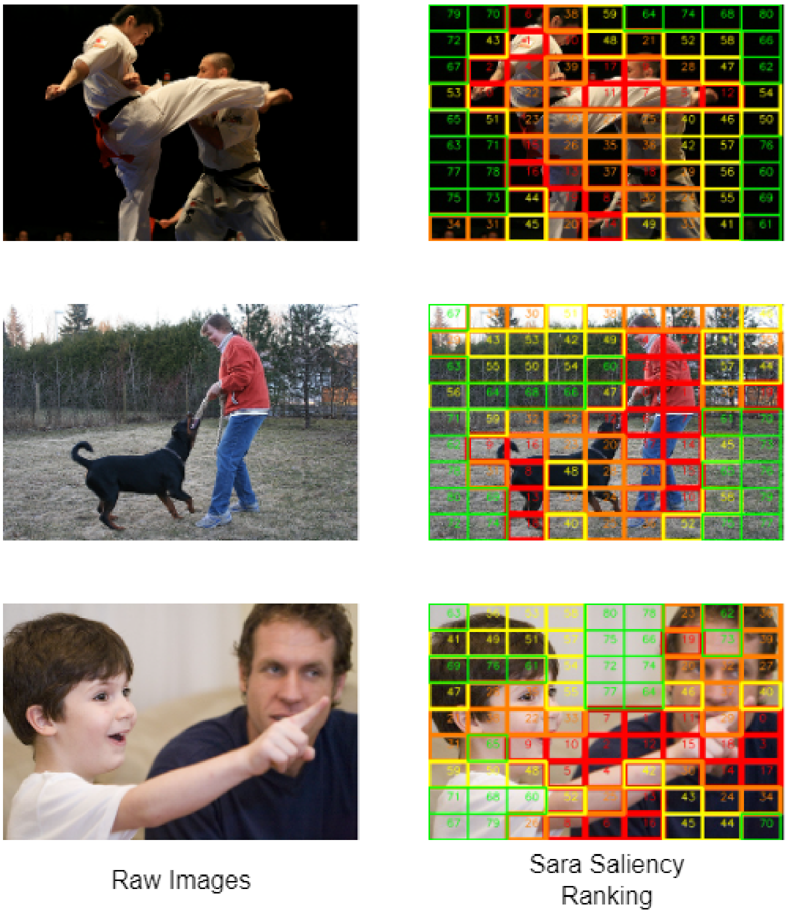


Figure 2.2: Saliency Ranking using Sara algorithm (Seychell and Debono, 2018). This algorithm highlights the most salient regions using a red to green colour scale and a ranking number. A low number and the colour red represent high saliency while a high number and the colour green represent low saliency.

segments. A demonstration of this algorithm is being provided in Figure 2.2 in which three images of varying complexities are being inferred. In separate work, Fang et al. (2021) propose a salient object ranking algorithm based on relative saliency. This architecture manages to compute instance segmentation and salient object ranking simultaneously by separating the task at hand into two branches. The first branch is dedicated to performing object detection through the use of an instance segmentation algorithm. Due to its flexible structure, the object detection algorithm can be varied however for this research, the CenterMask (Lee and Park, 2020) architecture is being used. The second complementary branch is the saliency ranking branch in which a novel dedicated module referred to as the position-preserved attention module is introduced to preserve the coordinates of the objects in the ROI pooling operation and allow for the fusion of the positional information with semantic features.

Saliency ranking is an important element in the field of visual saliency due to its capabilities in multiple object images. In this section, it was analysed how saliency detection manages to predict a saliency binary map of the saliency regions within the image which when considering multiple objects creates a challenge. Saliency ranking targets this challenge by providing the succession of the saliency of the different segments in an image.

2.3 Conclusion

This section introduced the background on which this research is based starting with an overview of visual impairments, providing a formal definition and its implications on the individuals. Furthermore, different saliency detection algorithms were discussed providing insight into their progression from non-deep visual models to deep saliency algorithms. This section is then followed by a description of saliency ranking as an extension to visual saliency.

Chapter 3

Literature Overview

This chapter discusses the rich literature on which this research is based. Starting with an overview of different image captioning techniques, this section introduces the traditional encoder-decoder structure and delves into its different variations. Different attention mechanisms, the use of segmentation and transformers in the field of image captioning are also introduced serving as the foundation of the proposed architecture. Instance segmentation is then discussed focussing on current state-of-the-art architectures. This section is then followed by literature related to scene classification and different researchers' findings in the area and ends with an overview of current systems dedicated to individuals with a form of visual impairment.

3.1 Image Captioning

In recent years artificial intelligence has seen surpassing advancements in computer vision and natural language processing allowing researchers to develop applications such as object detection (Jiao et al., 2019; Zhao et al., 2019), sentence generation (Iqbal and Qureshi, 2020) and image captioning (Anderson et al., 2018; Cai et al., 2020; Elamri and de Planque, 2016; Hrga and Ivašić-Kos, 2019; Lim and Chan, 2019; Liu et al., 2021; Makav and Kılıç, 2019; Pal et al., 2020; Tavakoli et al., 2017; Xiao et al., 2019).

Image captioning is the process of generating textual descriptions of a given image from the objects detected. As discussed by Hrga and Ivašić-Kos (2019), image captioning has seen rapid improvements from template-based models to advanced models utilising deep neural networks. In recent years, the encoder-decoder architecture (Hrga and Ivašić-Kos, 2019; Pal et al., 2020) seen in Figure 3.1, has increased in its populace in which an encoder is used to map the input into a vector representation and then utilising a decoder to generate an output based on this representation. An attention mechanism is generally

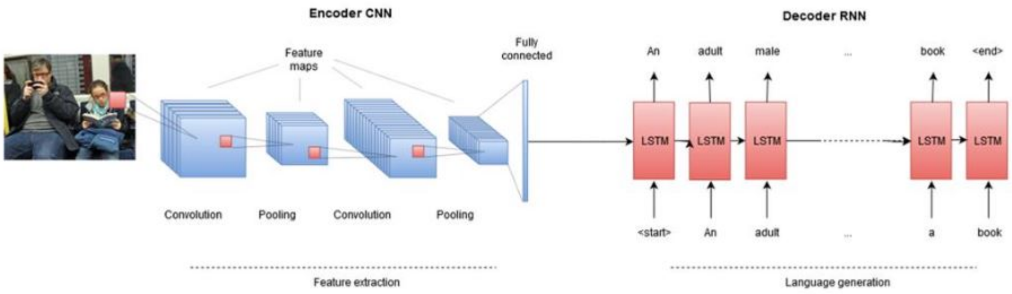


Figure 3.1: Encoder-Decoder Architecture of Deep Image Captioning (Hrga and Ivašić-Kos, 2019)

also implemented as an extension to this framework enabling the model to focus on the most salient regions within an input image to generate the next word of the output.

The encoder of this architecture (Pal et al., 2020) acting as a feature extractor is generally a CNN that is pre-trained on a large dataset for classification such as the ImageNet dataset (Deng et al., 2009). The architecture of the CNN chosen varies but typical architectures consist of the Alexnet (Krizhevsky et al., 2012), VGG (Simonyan and Zisserman, 2015), ResNet (He et al., 2016) and GoogLeNet (Szegedy et al., 2015). Alexnet developed by Krizhevsky et al. (2012) is one of the most influential architectures in image classification being the first CNN that managed to perform accurately and efficiently on the ImageNet dataset (Deng et al., 2009). This architecture introduced features such as the rectified linear units (ReLU), local response normalisation, overlapping pooling, augmentation and dropout. Following this success, Simonyan and Zisserman (2015) developed the VGG architecture which focuses on increasing the depth of the network and using very small convolutional filters with the introduction of the VGG-16 and the VGG-19 consisting of 16 and 19 layers respectively. He et al. (2016) took the research of the VGG (Simonyan and Zisserman, 2015) even deeper by developing the ResNet architecture, a deeper network with up to 152 residual layers. Variations consist of the ResNet-50, ResNet-101 and ResNet-152 consisting of 50, 101 and 152 layers respectively. The GoogLeNet (Szegedy et al., 2015) is a deep CNN that contains 22 layers and makes use of twelve times fewer parameters than the AlexNet (Krizhevsky et al., 2012) whilst increasing its accuracy. Improvements on the GoogleNet consist of the Inception-v2 (Szegedy et al., 2015), Inception-v3 (Szegedy et al., 2015) and Inception-v4 (Szegedy et al., 2016).

A recent addition in the area of image classification is the use of image transformers (Dosovitskiy et al., 2021) introduced at the International Conference on Learning Representations (ICLR) 2021, in which CNNs are eliminated. Dosovitskiy et al. (2021) inspired by the success of transformers in the area of natural language processing strived to create a similar architecture in which transformers are applied directly to an image. This was attained by dividing the image into patches and performing linear projection on every patch and appending it to its position. The sequence of vectors generated is then treated as if it were a sequence of words and inputted to the transformer encoder. The architecture of the adapted transformer as shown in Figure 3.2 (Dosovitskiy et al., 2021) is inspired by the traditional language transformer developed by Vaswani et al. (2017). The image transformer proved to match or better yet excel the current state-of-the-art image classification techniques while being comparatively cheaper to train.

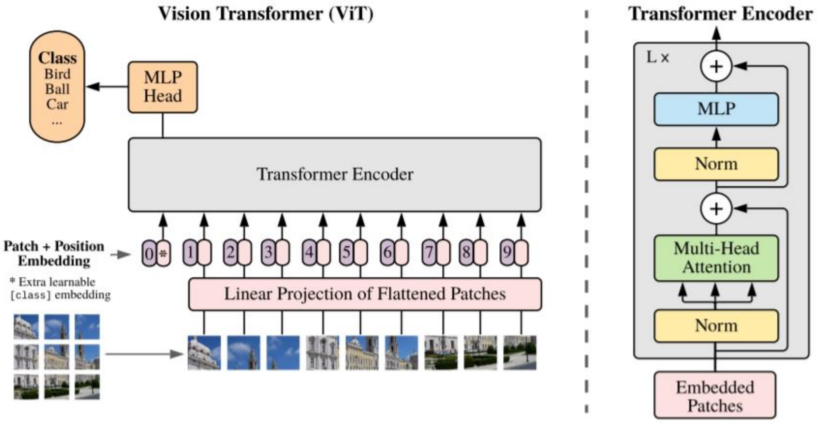


Figure 3.2: Image Transformer Architecture proposed by Dosovitskiy et al. (2021)

The second complementary part of the image captioning model consists of the decoder component (Hrga and Ivašić-Kos, 2019) which is generally an RNN (Du and Swamy, 2014) or an LSTM (long short-term memory) (Hochreiter and Schmidhuber, 1997) network. An RNN (Du and Swamy, 2014) is a type of artificial neural network that is able to retain past information through feedback loops connecting its relevance to the present. Given its architecture, however, the RNN is prone to the vanishing gradient problem in which information from the past and its relevance in the present creates a knowledge gap. The LSTM (Hochreiter and Schmidhuber, 1997) network addresses this challenge in addition to allowing for longer input sequences. Similar to the RNN, the LSTM has a chain-like structure with repeating modules, taking into consideration outputs of previous timesteps utilising three gates: the forget, input and output gate to determine the flow of information. The forget gate is responsible for forgetting information from the current cell state. Differently, the input gate is responsible to determine what new information is going to be added to the cell state. The final gate, the output gate determines how much of the updated cell state should be given as output assigning an importance level to regulate the data.

The language transformer (Vaswani et al., 2017) is another architecture that can handle sequence to sequence tasks at even longer range dependencies without making use of any convolutions or recurrences depending entirely on attention mechanisms to compute representations. The architecture follows the traditional sequence transduction models with an encoder and decoder but utilises stacked self-attention and fully-connected layers as analysed in Figure 3.3 (Vaswani et al., 2017). The encoder and decoder are constructed with 6 layers each with the encoder consisting of a multi-headed self-attention mechanism and a fully connected feed-forward network with a residual connection around each followed by a normalisation layer. The decoder differently consists of a multi-headed self-attention mechanism, a fully connected feed-forward network and a masked multi-headed attention mechanism with residual connections around each followed by a normalisation layer. The structure of this architecture allows for more parallelisation and faster training times.

Different researchers have looked at image captioning from varying perspectives and have proposed different architectures. Pal et al. (2020) analysed the effect of the encoder on the image captioning task by experimenting with different encoders while keeping a constant decoder. Of the different encoders considered the Inception-V3 (Szegedy et al., 2015) provided the most optimal results. The presented research highlighted the importance of the encoder component with varying results generated for different encoders. Liu et al. (2021) leveraged on the transformers to develop an architecture that is fully dependable on transformers. These researchers concluded that making use of a transformer dependent architecture is simple yet more effective exceeding the performance of traditional image captioning architectures.

An enchantment to image captioning architectures is attention mechanisms allowing the model to focus on the most salient parts of the images. Anderson et al. (2018) proposed a combination of a bottom-up (Le Meur et al., 2006) and top-down (Gao et al., 2009) attention mechanism aiming to generate more human-like captions by capturing how viewers perceive different image segments with varying degrees of attention. The bottom-up attention mechanism was used to provide a set of salient image regions through the use of Faster R-CNN architecture (Ren et al., 2015) whilst the top-down attention was used to predict the attention distribution over the image regions using LSTM (Hochreiter and Schmidhuber, 1997) layers. Differently, Tavakoli et al. (2017) after conducting an investigation of how humans perceive and describe visual scenes proposed a saliency-boosted image captioning model. This architecture makes use of the VGG-16 (Simonyan

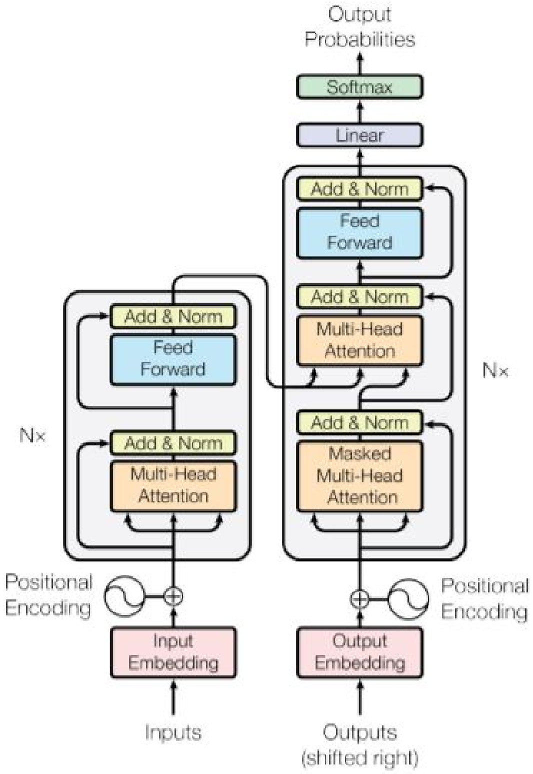


Figure 3.3: Language Transformer Architecture proposed by Vaswani et al. (2017)

and Zisserman, 2015) network as the backbone to perform feature extraction and LSTM (Hochreiter and Schmidhuber, 1997) model. Saliency is computed from the features extracted from the encoder through the use of an ensemble of saliency predictors. From this research, it was distinguished that generally, humans tend to talk about objects that are most salient early on in their description.

Lim and Chan (2019) introduced instance segmentation to image captioning as an attention mechanism to detect salient regions at a pixel level. Here as seen in Figure 3.4, the researchers employed an encoder-decoder architecture with the encoder consisting of two different layers: a mask layer to detect objects and a background layer to determine the scene. The mask layer leverages the Mask R-CNN (He et al., 2017) instance segmentation algorithm to produce a set of binary masks with their respective detection score. Depending on the confidence score, weighted masks are calculated to identify good masks by performing element-wise multiplication. Although the researchers are exploiting the confidence score as an indicator for the validity of the masks, this might not be ideal. As this research will argue in Section 4.5.1, other solutions such as utilising visual saliency could potentially result in a more suitable identifier for the mask validity. Feature extraction using a ResNet-50 (He et al., 2016) pre-trained on the ImageNet (Deng et al., 2009) dataset is then performed on the weighted masks and concatenated with features extracted from the scene layer using an identical network. Argumentatively as will be discussed in Section 3.3, ideally dedicated CNNs are utilised for the extraction of scene features and not generic CNNs pre-trained on ImageNet (Deng et al., 2009) dataset. As a decoder, this architecture employs an LSTM (Hochreiter and Schmidhuber, 1997) network. The researchers concluded that this approach outperformed baseline models and generated comparable results to the current state-of-the-art architectures.

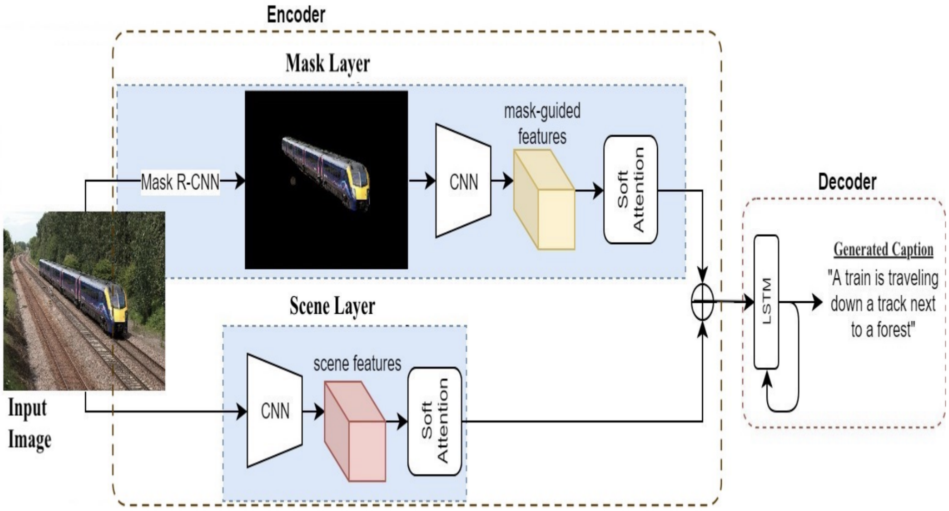


Figure 3.4: Mask Captioning Network proposed by Lim and Chan (2019) consisting of an encoder and a decoder framework. The encoder encompasses two separate layers: a mask layer utilising instance segmentation and a scene layer. Both layers use a CNN pre-trained on the ImageNet dataset for feature extraction. As a decoder, an LSTM network is being utilised.

Similarly, Cai et al. (2020) discussed the use of panoptic segmentation which is a blend of instance and semantic segmentation as an attention mechanism along with a dual-attention module. In this architecture, the instance segmentation algorithm Mask R-CNN (He et al., 2017) was used to generate masks for objects while the semantic segmentation algorithm (Chen et al., 2016) was used to describe the context of the image. The resultants from the segmentation algorithms were then merged for panoptic segmentation for feature extraction using a pre-trained ResNet-101 (He et al., 2016). As a decoder an LSTM (Hochreiter and Schmidhuber, 1997) was utilised for sentence generation. This research proved to achieve competitive results with current state-of-the-art techniques.

Specialisation has also been introduced to the area of image captioning, finding research targeted at people with a form of visual impairment. Makav and Kılıç (2019) dedicated their research to implementing a dedicated image captioning model. In this architecture, a simple VGG-16 (Simonyan and Zisserman, 2015) encoder and a decoder utilising the Stanford CoreNLP model (Manning et al., 2014) was used. Similarly, Elamri and de Planque (2016) implemented a dedicated image captioning model utilising a simple pre-trained VGG-16 (Simonyan and Zisserman, 2015) encoder and decoder architecture to generate captions with experimentation being conducted on the decoder. From this research, it was concluded that an LSTM based captioner performed slightly better than that based on an RNN.

3.2 Instance Segmentation

Object detection (Jiao et al., 2019; Zhao et al., 2019) has seen rapid improvements throughout the years with the introduction of segmentation techniques primarily semantic and instance segmentation (Hafiz and Bhat, 2020) achieving to determine the precise location of an object through the use of masks. Semantic segmentation (Li et al., 2018) performs image classification at a pixel level striving to output a class label for every pixel. Differently, instance segmentation (Hafiz and Bhat, 2020) is an evolved image segmentation technique that provides a different label for separate instances of objects belonging to the same class as shown in Figure 3.5 (Wilson, 2019). The evolution of deep learning resulted in the origin of several instance segmentation frameworks, some of which will be discussed here-under.

3.2.1 Mask Region-Based Convolutional Neural Network (Mask R-CNN)

The Mask R-CNN architecture proposed by He et al. (2017) is based on the evolution of the the R-CNN architecture (Girshick, 2015; Girshick et al., 2014; Ren et al., 2015). The R-CNN proposed by Girshick et al. (2014) was one of the first architectures that analysed the use of CNNs for object detection (Hafiz and Bhat, 2020). This architecture utilises selective search to determine regions of interest for feature extraction and a Support Vector Machine (SVM) for classification. Bounding box regression is also used to determine the bounding box coordinates for each region. This technique proved to be relatively slow due

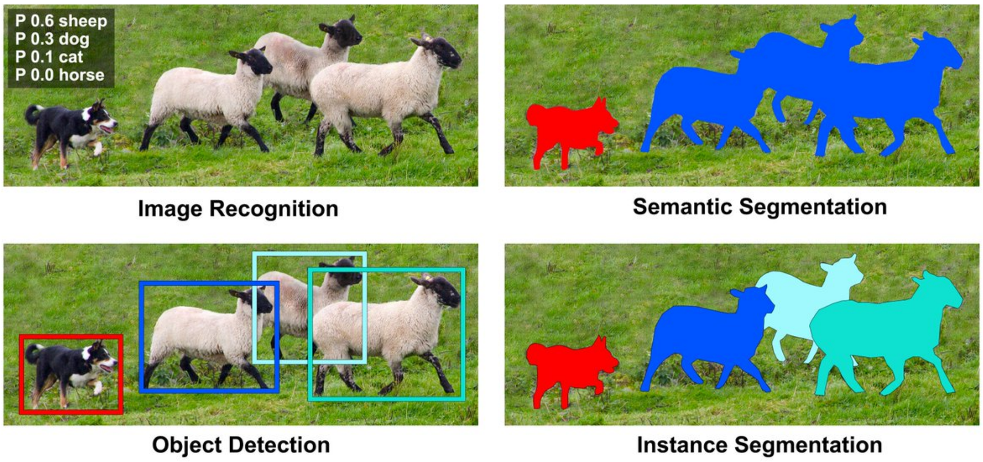


Figure 3.5: The Evolution of Object Detection from Image Classification to Segmentation Techniques (Wilson, 2019). Traditional image recognition algorithms (Top Left) allowed researchers to solely identify the objects in the image. A progression to these algorithms were the targeted object detection algorithms (Bottom Left) to which the algorithms were given the additional capability to identify the location of the objects using bounding boxes. A leap from these algorithms were segmentation algorithms that allowed for the identification of precise locations using masks. Semantic segmentation (Top Right) managed to perform pixel-level classification, classifying objects of the same category as a unit. Instance segmentation (Bottom Right) was an evolution of the previous algorithms being able to distinguish multiple objects of the same category as independent instances.

to the number of models involved as well as computationally expensive due to feature extraction being performed on every image region. An improvement is the Fast R-CNN (Ren et al., 2015) which inputs the image directly to the backbone of the architecture to generate the regions of interest. A region of interest (ROI) pooling layer is also introduced in this architecture. The Faster R-CNN implemented by Ren et al. (2015) addresses the selective search of the Fast R-CNN by replacing it with a region proposal network (RPN) to produce object proposals. Although this model is the most accurate of its precedents, it is still not optimal since it focuses on parts of the image sequentially. This architecture inspired He et al. (2017) with the implementation of the Mask R-CNN which is an extension to the Faster R-CNN (Ren et al., 2015) by introducing an object mask prediction branch on each RoI that works in parallel to the object bounding box recognition. In this architecture, ROIAlign was implemented instead of the ROI pooling technique used in the Faster R-CNN (Ren et al., 2015) to preserve the exact spatial location of the regions.

Similar to its precedent, the researchers made use of the RPN as the backbone of the architecture accompanied by a ResNet101 (He et al., 2016) instead of a VGG16 network (Simonyan and Zisserman, 2015) used in the previous architecture.

3.2.2 Yolact (You Only Look At CoefficienTs)

Yolact was introduced by Bolya et al. (2019) and it consists of a simple and fast instance segmentation model with a fully convolutional topology. This architecture is based on the RetinaNet (Lin et al., 2017b) using the ResNet-101 (He et al., 2016) and a Feature Pyramid Network (FPN) (Lin et al., 2017a) as its backbone. This model divides the main segmentation problem into two parallel sub-tasks. The first task involves generating a set of prototype masks through the use of a fully convolution network (FCN) and the second task consists of predicting per-instance mask coefficients for each anchor. Instance masks are generated by linearly combining the prototype masks to the mask coefficients that pass successfully through the Non-Maximum Suppression.

3.2.3 Yolact ++ (You Only Look At CoefficienTs ++)

The Yolact (Bolya et al., 2019) architecture was improved upon by Bolya et al. (2020b) with the implementation of the Yolact++ aiming to increase the model's performance while retaining its real-time application. The first improvement inspired by the Mask Scoring R-CNN architecture (Huang et al., 2019), consists of a network that performs re-ranking of mask predictions according to their mask quality. Secondly, deformable convolutions were used within the backbone network to generate more precise mask prototypes by utilising free-form sampling instead of the traditional grid sampling. Finally, the researchers also made use of multi-scale detection anchors per FPN level to create a more optimised prediction head. The Yolact (Bolya et al., 2019) and the Yolact++ (Bolya et al., 2020b) can perform real-time instance segmentation due to their parallel structure and lightweight assembly.

3.2.4 CenterMask

CenterMask is the first anchor-free one-stage instance segmentation technique developed by Lee and Park (2020) that is inspired by the high accuracies achieved by the Mask R-CNN (He et al., 2017) and the high-speed performance of the Yolact (Bolya et al., 2019). The CenterMask architecture is composed of three components: the backbone, the fully convolutional one-stage object (FCOS) (Tian et al., 2019) detection head and the mask head. This architecture makes use of the VoVNetv2 (Lee and Park, 2020) along with an

FPN to extract features which are then processed through the FCOS (Tian et al., 2019) detection head which is an anchor-free and proposal-free object detector. Given its anchorless properties, this network avoids complex computations reducing computational costs while increasing efficiency. The objects detected are then used by the SAG-Mask which is a spatial attention module that predicts a segmentation mask for each Region of Interest (RoI).

3.2.5 TensorMask

Chen et al. (2019) present fairly new research in the under-explored area of dense sliding-window for instance segmentation through their TensorMask architecture. The ideology behind this architecture is to analyse instance segmentation as a prediction task over 4D structured tensors by capturing geometry and enabling novel operators. Chen *et al.* developed a pyramid structure for this architecture over a scale-indexed list of the 4D tensors consisting of a pyramidal shape in both relative mask position and object position which grow in opposite directions. This reflects how large objects have high-resolution masks with coarse spatial localisation while contrarily small objects have low-resolution masks with fine spatial localisation. The TensorMask architecture also consists of a mask prediction head and a classification head to perform object detection and generate masks in sliding windows. To perform the predictions, the architecture utilises an underlying FPN backbone with ResNet-50 (He et al., 2016) similar to RetinaNet (Lin et al., 2017b).

3.3 Scene Classification

Scene classification (Lazebnik et al., 2006; Zeng et al., 2021; Zhou et al., 2015; Zhou et al., 2017) is a niche of image classification which is considered to this day a fundamental challenge within computer vision. Scene classification is the process of identifying the environment in which a particular image is located. Identifying the scene is as important as identifying the contents of an image as this aids in the understanding and interpretation of the context of an image. For instance, a man with a firearm in a shooting range and a man with a firearm in a bank are two different situations with distinct repercussions.

The main challenges (Zeng et al., 2021) that stem from scene classification are related to large intraclass variation and semantic ambiguity. Intraclass variation refers to images belonging to the same scene category that however consist of varying backgrounds, objects and imaging conditions. Differently, semantic ambiguity refers to images of different scene categories that share similar objects and backgrounds. To address these challenges, researchers have focused their efforts on developing a set of specified datasets for scene classification. One such example is the Places dataset (Zhou et al., 2015; Zhou et al., 2017) that consists of 10 million scene photographs belonging to 434 different scene categories which amount to roughly 98% of the places one generally encounters in a lifetime. Subsets of this dataset consist of the Places205 and the Places88 (Zhou et al., 2015) which consist of 205 and 88 scene categories respectively. Other variations consist of the Places365-Standard and Places365-Challenge (Zhou et al., 2017) which both contain an equal amount of categories with the training set of the latter being significantly larger. Other commonly used datasets consist of the smaller dataset Scene15 (Lazebnik et al., 2006), the MIT67 dataset (Quattoni and Torralba, 2009) and the Scene UNderstanding (SUN397) dataset (Xiao et al., 2010).

Researchers have exploited the improvements of deep learning algorithms to perform scene classification enabling the models to learn representations directly from large datasets (Zeng et al., 2021). Zhou et al. (2017) discussed the use of three different well-known architectures: the Alexnet (Krizhevsky et al., 2012), VGG16 (Simonyan and Zisserman, 2015) and GoogLeNet (Szegedy et al., 2015) and trained them on their published datasets the Places205 (Zhou et al., 2015) and the Places365-Standard (Zhou et al., 2017) dataset with the addition of the ResNet152 being trained solely on the Places365-Standard Dataset. The researchers concluded that the GoogLeNet and the VGG outperformed the AlexNet by a large margin on the Places205 whilst on the Places-365 dataset, the VGG and the ResNet had similar performances outperforming the GoogLeNet and the AlexNet. A different approach is designing specific deep learning models to perform scene classification. Liu et al. (2018) inspired by dictionary learning implemented a modified CNN that experiments with nonlinear discriminative dictionary learning layers aiming to enhance sparse representation while simultaneously maximising its discriminative capabilities. Hayat et al. (2016) discussed that unlike in object classification, in scene classification an image generally consists of multiple distinct objects of various sizes spread across diverse spatial locations. Therefore, Hayat et al. (2016) introduced a modified CNN with an added specified layer to handle spatial layout deformations as well as a pyramidal image representation to handle scale variations.

As discussed by Hayat et al. (2016), scene classification is a distinguishable task from general object classification and therefore a CNN trained for object classification whilst producing outstanding results on an object-orientated dataset will not give the same results on scene-orientated datasets and vice versa. This was also concluded by Zhou et al. (2015) when comparing a CNN trained on the Places dataset (Zhou et al., 2017) and an identical CNN trained on the ImageNet (Deng et al., 2009) dataset. The aforementioned was furtherly highlighted by Herranz et al. (2016) in which they identified CNNs for the ImageNet dataset and CNNs for the Places dataset are tuned for different scale ranges.

3.4 Dedicated Systems

The evolution of AI has enabled researchers to implement systems for the visually impaired and provide them with an enhanced quality of life. Generally, this is achieved through a combination of sensors, AI algorithms and wearable devices such as glasses (Khan et al., 2020; Yang et al., 2014), belts (Katzschmann et al., 2018), shoes (Patil et al., 2018), walking sticks (Shandu et al., 2020; Villanueva and Farcy, 2012) and gloves (Yelamarthi and Laubhan, 2015). In addition, some researchers have also opted to develop mobile applications (Awad et al., 2018; Felix et al., 2018; Kiruthika and Sheela, 2016).

The systems implemented have different functionalities depending on the information gathered by the integrated sensors. AI techniques are then used on the acquired data. The most common functionality implemented is object detection and obstacle avoidance for path-finding (Khan et al., 2020; Patil et al., 2018; Shandu et al., 2020; Yelamarthi and Laubhan, 2015) using models such as the pre-trained TensorFlow Lite single-shot detection (SSD) model (Khan et al., 2020) or TensorFlow's deep learning CNN model (Shandu et al., 2020) to perform object detection. The most commonly used integrated sensors for data extraction consist of depth and ultrasonic sensors (Patil et al., 2018), time-of-flight distance sensors (Katzschmann et al., 2018), 3D depth sensors (Yelamarthi and Laubhan, 2015) and cameras (Khan et al., 2020) for obstacle avoidance and infrared LEDs and photodiodes (Villanueva and Farcy, 2012) for path-finding. Moreover, some researchers also opted for global positioning systems (GPS) for real-time navigation (Shandu et al., 2020). Other implemented functions include integrated text readers (Khan et al., 2020; Shandu et al., 2020). Here, technologies such as the Tesseract v-4 for text recognition and eSpeak, a text to speech engine (Khan et al., 2020) or Google's Cloud Speech API (Shandu et al., 2020) are being used. Another functionality proposed was a clothes choosing assistant (Yang et al., 2014) implemented to recognise clothing patterns and colours These systems communicate with the users through a variety of outputs such as auditory feedback (Khan et al., 2020; Shandu et al., 2020; Yang et al., 2014) or vibrations (Villanueva and Farcy, 2012; Yelamarthi and Laubhan, 2015).

Furthermore, some researchers have also introduced assistive mobile applications. Here we can analyse systems that enable light and colour detection, object recognition and banknotes recognition (Awad et al., 2018). Some mobile applications are also personal assistants (Felix et al., 2018) while others make use of external sensors such as temperature sensors, ultrasonic sensors, acceleration sensors and GPS to enable navigation (Kiruthika and Sheela, 2016).

3.5 Conclusion

This chapter represented the rich literature on which this research is based starting with a review of current state-of-the-art image captioning architectures presenting similar work in this field delving into the traditional framework and its variations as well as the introduction of segmentation and transformers to this area. This section then proceeds to discuss different instance segmentation algorithms focusing on current state-of-the-art and novel research. Scene classification methodologies are then discussed focussing on dedicated datasets and research in this area. This section concludes with an analysis of systems dedicated to individuals with visual impairments.

Chapter 4

Methodology

Inspired by the current state-of-the-art technologies and the research conducted by Lim and Chan (2019), a novel dedicated image captioning architecture is proposed that utilises instance segmentation and saliency as an attention mechanism along with a hybrid encoder and a transformer-based decoder. In addition, the architecture explores dedicated variables aiming to generate sentences that are more readable and therefore more accessible. The proposed architecture will be delved into further details in this section, discussing the implementation of the various components within the system.

4.1 System Overview

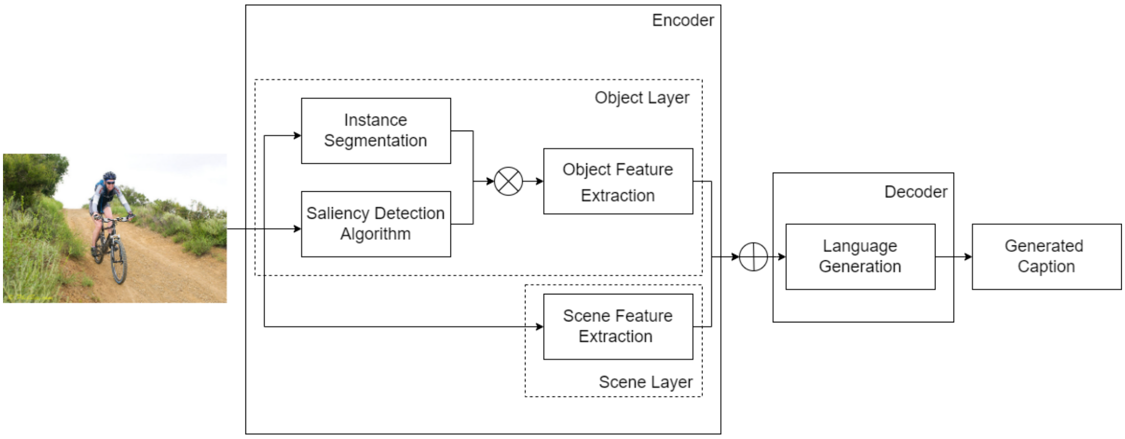


Figure 4.1: A High-Level Overview of the Proposed Architecture

The proposed architecture presented in Figure 4.1 makes use of an encoder-decoder framework with the encoder component following the concepts introduced by Lim and

Chan (2019) and utilises two separate layers: an object layer and a scene layer. The main aim of the object layer is to identify objects in the image and provide a level of saliency to the object. This is being achieved through a combination of instance segmentation and saliency detection algorithms. Object features are extracted from a combination of these two algorithms using an object feature extraction algorithm. Differently, the scene layer is used to extract features related to the environment of the image using a dedicated scene feature extraction model. The concatenated resultant is then used by a language generation model forming part of the decoder to train and generate the related captions.

An in-depth overview of the proposed methodology is being provided in Figure 4.2 which explains the holistic processes utilised throughout this study. Starting with the object layer, Lim and Chan (2019) utilised a Mask R-CNN model (He et al., 2017) to generate binary masks. Differently, in this research, an analysis is being performed to distinguish whether Mask R-CNN is still the current state-of-the-art architecture. To achieve this, an evaluation of pre-trained models of the Mask R-CNN (He et al., 2017), Yolact (Bolya et al., 2019), Yolact++ (Bolya et al., 2020b), TensorMask (Chen et al., 2019) and CenterMask (Lee and Park, 2020) are being performed on two different datasets: the MS-COCO17 (Lin et al., 2014) validation dataset and the Tiny Pascal VOC (Everingham et al., 2010) training dataset with the model achieving the overall highest performance to be utilised for the object layer. Binary masks are then to be generated from the chosen model and weighted masks are then to be calculated. Differently from Lim and Chan (2019), the architecture will not depend on the confidence score of the model but will rather generate and utilise a saliency map to determine the most salient regions of the image, therefore basing the attention mechanism on saliency rather than the assertiveness of the instance segmentation model. Saliency algorithms considered for this application consist of the renowned saliency algorithm presented by Itti et al. (1998), EML-Net (Jia and Bruce, 2020), Pyramid Feature Attention Network (Zhao and Wu, 2019) and Deep Gaze II (Kümmerer et al., 2016) which will be evaluated with the aid of contours and saliency ranking. Features are then to be extracted from the weighted images through the use of a pre-trained vision transformer.

The second layer within the encoder is the background or scene layer. This layer is used to gather information about the general context of the image. Here Lim and Chan (2019) utilised a ResNet-50 (He et al., 2016) model pre-trained on the ImageNet (Deng et al., 2009) dataset. Following the research of Hayat et al. (2016) this does not generate optimal results for scene-related information. Thereby, a new model that is pre-trained on the Places dataset is proposed to be utilised for feature extraction. Zhou et al. (2017) trained and made available the AlexNet (Krizhevsky et al., 2012), ResNet18, ResNet50 (He et al., 2016) and DenseNet161 (Huang et al., 2017) on the Places365 (Zhou et al.,

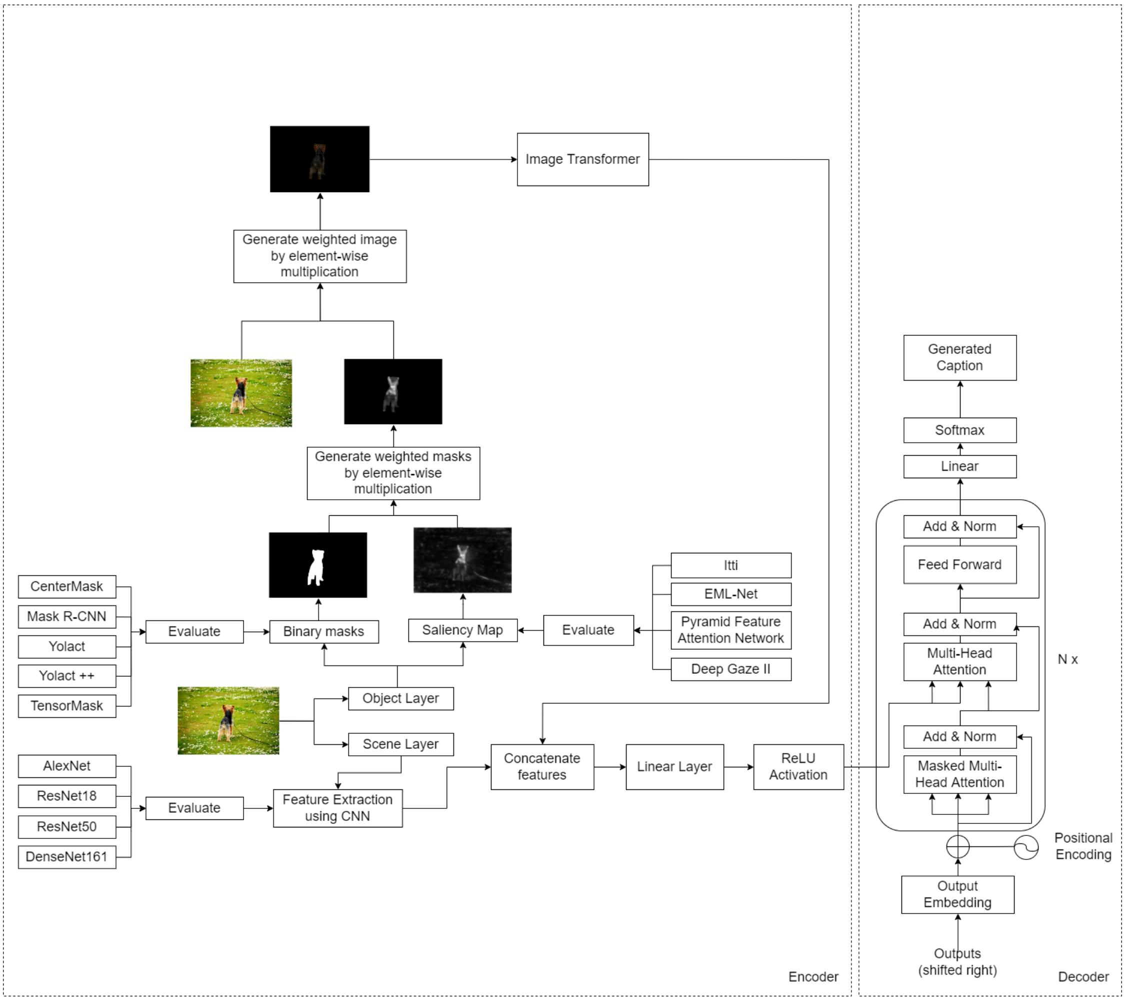


Figure 4.2: Block Diagram of the proposed Architecture consisting of a hybrid encoder consisting of a scene and an object layer each utilising dedicated researched deep learning models and a transformer decoder.

2017) dataset. After evaluating the above models, the most optimal model is to be used to perform feature extraction of the images.

The features extracted from both layers are then to be concatenated together and passed through the decoder of the architecture: a language transformer (Vaswani et al., 2017) that differs from the opted LSTM (Hochreiter and Schmidhuber, 1997) decoder by Lim and Chan (2019). A further consideration of the architecture is accessibility and readability. This is to be accomplished by a dedicated experimentation with the padding of the sentences during the pre-processing of the training captions. Furthermore, the vocabulary size will be varied to determine its impact on the general performance of the image captioning model and its influence on readability.

4.1.1 Methodology Structure

In the subsequent sections, the methodology of the processes mentioned above will be delved into furtherly following the chronological order being provided below:

1. Instance Segmentation Analysis Procedure (Section 4.2) - This section discusses the research conducted on determining the leading instance segmentation architecture for the mask layer of the encoder.

2. Saliency Prediction for Weighted Masks Analysis (Section 4.3) - This section discusses the methodology adhered to determine the ideal saliency prediction algorithm for the attention mechanism of the mask layer of the encoder.

3. Scene Classification Analysis Procedure (Section 4.4) - This section represents the methodology of the scene classification which forms part of the scene layer of the encoder.

4. Image Captioning Architecture (Section 4.5) - This section holistically discusses the methodology of the image captioning architecture starting from the encoder of the model built argumentatively and the decoder. This section continues to discuss the datasets and metrics and then follows with information regarding the training and the evaluation of the models with the hyper-parameters utilised.

4.2 Instance Segmentation Analysis Procedure

As part of this research, a review of current instance segmentation architectures is being conducted to analyse the best model for the object layer of the image captioning encoder. Here along with the Mask R-CNN (He et al., 2017), four other recently developed architectures are being considered. This section provides information regarding the methodology of this review to ensure a fair and equal comparison between the architectures.

4.2.1 System Overview

To review current instance segmentation architectures, five different architectures the Mask R-CNN (He et al., 2017), Yolact (Bolya et al., 2019), Yolact++ (Bolya et al., 2020b), TensorMask (Chen et al., 2019) and CenterMask (Lee and Park, 2020) are being evaluated on two different datasets: the validation set of the MS-COCO17 (Lin et al., 2014) and the training set of the Tiny Pascal VOC (Everingham et al., 2010). Each model used is going to be pre-trained on the MS-COCO17 training dataset.

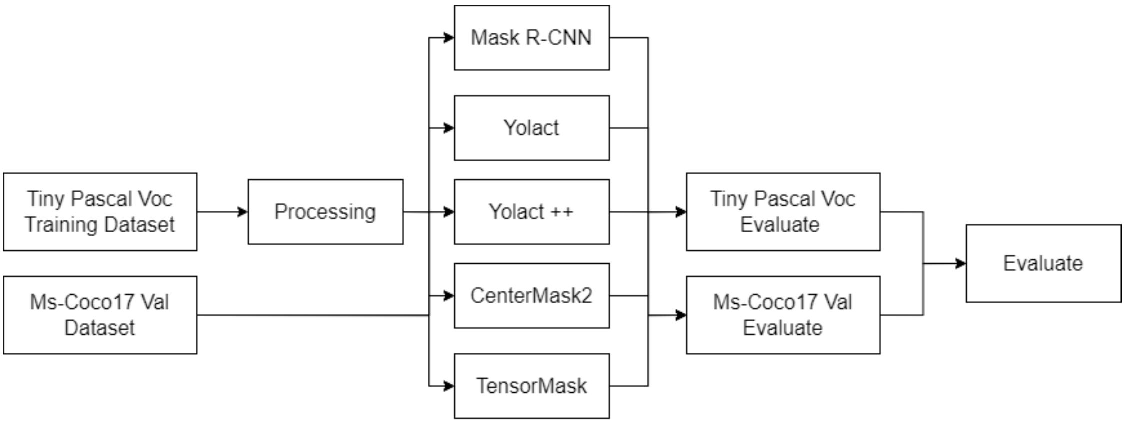


Figure 4.3: Block Diagram of the System utilised for the Instance Segmentation Analysis

As shown in Figure 4.3 the procedure required for this part of the research is rather simple. Each architecture will be exposed to both datasets and an evaluation will be carried out on each dataset individually and then compared together to draw a solid conclusion. Since the architectures are pre-trained on the MS-COCO17 there is no pre-processing required prior to performing the inference on the MS-COCO17 validation set. On the other hand, the Tiny Pascal VOC is an external dataset to which these models were not exposed to prior this inference. Therefore, processing was required to determine whether additional training is required. By comparing the categories of both datasets, it was concluded that the 20 categories of the Tiny Pascal VOC are found within the 80 categories of the MS-COCO17, some under a different label. Consequently, transfer learning or further training was not required however, labels had to be processed to match the labels found in the MS-COCO17 dataset. An example of a variation is the label *airplane* in which in the Tiny Pascal VOC dataset is referred to as an *aeroplane*. To handle such cases, a natural language processing library, the *nltk* was used to generate synonyms of the labels and perform the necessary mappings.

The pre-trained models and the architectures are being retrieved from two different sources: Github[1] and Detectron2 (Wu et al., 2019). Detectron2 (Wu et al., 2019) is a Facebook AI Research Project that contains an assortment of state-of-the-art object detection and segmentation models. The Mask R-CNN architecture used for this review was retrieved through Detectron2 and ModelZoo[2]. Various Mask R-CNN architectures are available on this platform however the model opted for was that closest to that introduced by He et al. (2017) in their research and therefore utilises the ResNet-101 with an

[1] GitHub is a code hosting platform used for version control and collaboration.

[2] Model Zoo is a machine learning model deployment platform.

FPN as the backbone. The TensorMask architecture was also retrieved from Detectron2 since this is available as one of the research projects in this library. The variation of the model being used is that that consists of the ResNet-50 with an FPN as its backbone. CenterMask2 is another research project that is dependable on the Detectron2 library however it is not currently available in this library. Therefore it was retrieved from the researchers' Github repository (Lee and Park, 2020). The model being evaluated utilises the VoVNetV2 with an FPN as its backbone. Differently, the Yolact and Yolact++ do not make use of nor are dependable on the Detectron2 library and therefore were retrieved directly from the researchers' Github (Bolya et al., 2020a). For both models, the baseline was used for the review, therefore with the ResNet-101 and an FPN as both of their backbones.

The evaluation of the models is being carried out based on the COCO evaluation metric discussed furtherly in this section. For the Mask R-CNN, TensorMask and CenterMask the evaluation was carried out through the Detectron2 library whilst for the Yolact and the Yolact++ the evaluation was conducted through the scripts implemented by the researchers of the architecture with some minor modifications.

4.2.2 Datasets

The MS-COCO (Lin et al., 2014) dataset is a well-renowned dataset that contains 330,000 images with 1.5 million object instances and 80 object categories. The applications of this dataset vary from object detection to segmentation as well as image captioning. This dataset is also one of the most commonly used benchmarking datasets allowing researchers to compare their architectures with other works. The MS-COCO 2017 dataset consists of 118,000 images for training and 5,000 images for validation. This dataset also contains a test set in which annotations are not made publicly available but a model's performance on this can be measured through the COCO Evaluation Server.

The Tiny Pascal VOC is a subset of the Pascal VOC dataset (Everingham et al., 2010) which contains 1349 training images and 100 test images out of the 11,530 images found in the original dataset containing 20 object categories. Similar to the MS-COCO dataset, this dataset is famous for object recognition and detection tasks in the fields of computer vision and machine learning. Previous to the MS-COCO dataset, the Pascal VOC dataset was widely accepted as a benchmark for object detection and similarly contains an Evaluation Server.

For this review, the validation set of the MS-COCO17 is being used as well as the training set of the Tiny Pascal VOC with their respective segmentation masks data. The classes and the distribution of both these datasets can be analysed in Figure 4.4 and Figure

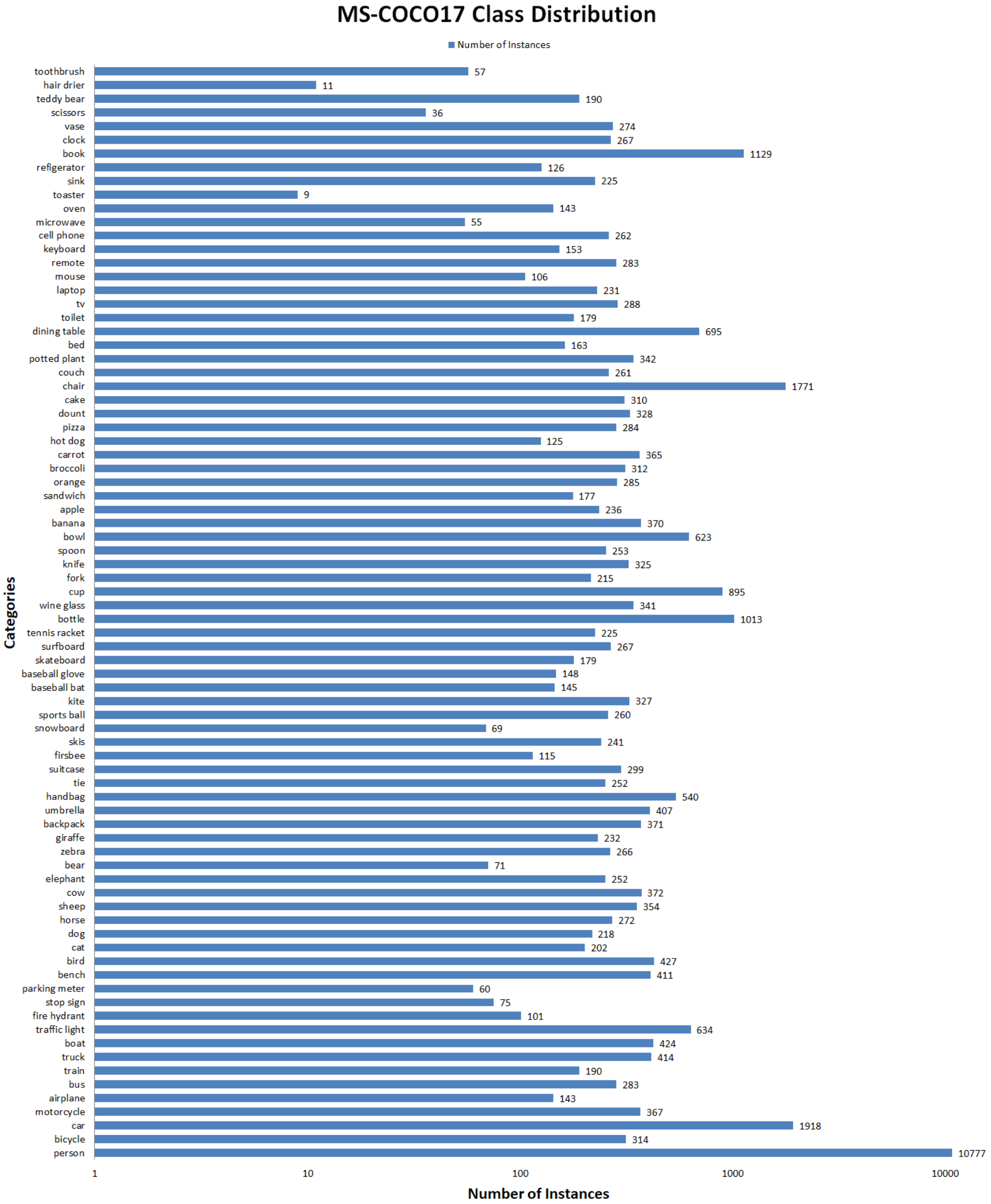


Figure 4.4: Distribution of the MS-COCO17 dataset scaled to a logarithmic base 10 highlighting that the dataset is unbalanced with multiple instances of the class *person*.

4.5 scaled to logarithmic base 10. From both these distributions, it could be analysed that the class *person* is the most commonly occurring class by a large margin, while the other

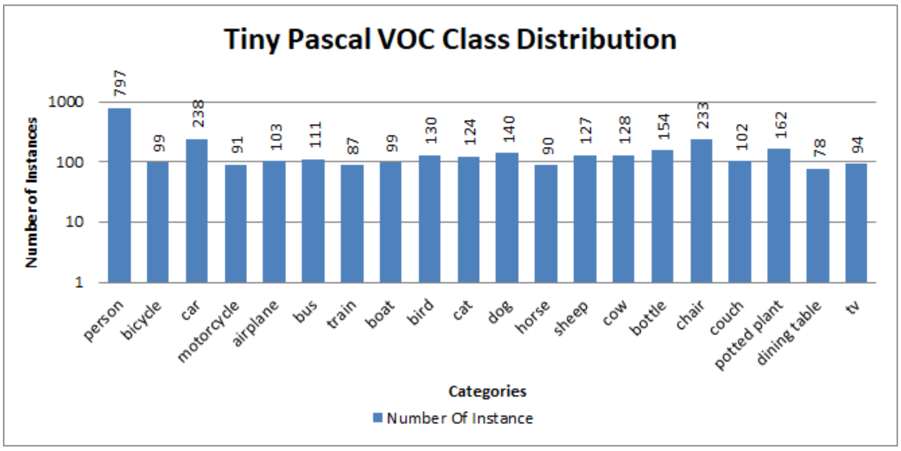


Figure 4.5: Distribution of the Tiny Pascal VOC dataset scaled to a logarithmic base 10 showing a slightly unbalanced dataset with the class *person* occurring more often.

classes appear with a frequency that is relatively similar to each other. One can also analyse that the Tiny Pascal VOC contains the categories that are most frequent in the MS-COCO17.

4.2.3 Metrics

Instance Segmentation algorithms are generally evaluated through the use of the Average Precision (AP) and the mean Average Precision (mAP) (Padilla et al., 2020). This is determined based on the intersection over union (IoU) which is the area of overlap between two segmentation masks calculated by the following equation (Equation 4.1):

$$IoU = \frac{Area\,of\,Overlap}{Area\,of\,Union} \tag{4.1}$$

Using a pre-established threshold of this value, the evaluation metric will be able to determine if a prediction should be classified as positive or negative. This will establish the true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn) which are then used to calculate the precision and recall. Precision is defined as the percentage of correct positive predictions made amongst all the predictions while recall is defined as the percentage of correct positive predictions made among all positive instances. The average precision is then calculated by the area under the precision-recall curve.

Researchers can utilise a single threshold or else a range of thresholds to evaluate their architectures (Padilla et al., 2020). The MS-COCO researchers (Lin et al., 2014) introduced an evaluation technique that makes use of the latter, and therefore makes use of a range of

IoU threshold values between 0.5 to 0.95 with a step size of 0.05 generating 10 precision-recall pairs. The mAP is then calculated by averaging over the IoU at each threshold in the following way (Equation 4.2):

$$mAP_{COCO} = \frac{mAP_{0.5} + mAP_{0.55} + ... + mAP_{0.95}}{10} \tag{4.2}$$

Using such a technique, threshold bias is eliminated since different thresholds are considered and given a different weighting. For this review, the evaluation technique presented by the MS-COCO researchers is being utilised.

4.3 Saliency Prediction for Weighted Masks Analysis Procedure

Continuing with the investigation of the ideal models for the object layer, a review of saliency prediction algorithms is being performed to distinguish the best complementary algorithm. As part of this review, both deep learning models, as well as traditional algorithms, are being considered. This section provides information about the methodology employed for this review.

4.3.1 System Overview

Saliency prediction algorithms are being analysed with respect to the masks generated by an instance segmentation algorithm to determine the masks' validity. For this analysis, four different algorithms are being considered. The first two algorithms, the EML-Net (Jia and Bruce, 2020) and the DeepGaze II (Kümmerer et al., 2016) are algorithms that as discussed by Borji (2018) are two of the current best performing deep saliency models. In addition, to these algorithms the traditional saliency algorithm developed by Itti et al. (1998) is also being experimented with, to compare its performance for this application as a non-deep visual saliency model. The final algorithm considered is the Pyramid Feature Attention Network for Saliency Prediction which is an algorithm presented by Zhao and Wu (2019) which exploits a top-down approach.

To analyse these systems as shown in Figure 4.6, the same procedure that shall be followed during the feature extraction of the object layer is being followed. Therefore, binary masks are firstly being generated from the Mask R-CNN architecture and element-wise multiplication is being calculated with a saliency mask produced by each of the above saliency prediction algorithms. This will determine the weighted mask that will be used to distinguish the mask's relevance in the scene. Element-wise multiplication is then also

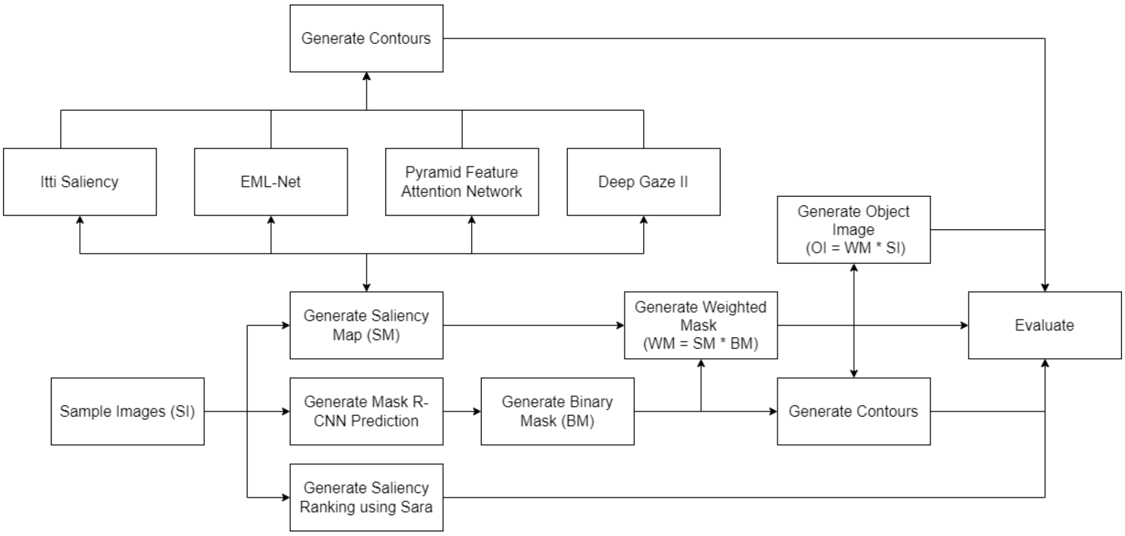

Generate Contours
Itti Saliency
EML-Net
Pyramid Feature Attention Network
Deep Gaze II
Generate Object Image (OI = WM * SI)
Generate Saliency Map (SM)
Generate Weighted Mask (WM = SM * BM)
Evaluate
Sample Images (SI)
Generate Mask R-CNN Prediction
Generate Binary Mask (BM)
Generate Contours
Generate Saliency Ranking using Sara

Figure 4.6: Block Diagram of the System utilised for the Saliency Prediction Analysis

being computed between the weighted mask and the image to consider the final object image. As shown, contours are also being generated for each binary mask, saliency map and weighted mask to identify whether the saliency predictor has generated a salient score for all the objects determined by the instance segmentation algorithm. In addition, saliency ranking is being identified through the use of the Sara algorithm (Seychell and Debono, 2018). Saliency ranking in addition to contours and other related metrics mentioned below will be used to evaluate and compare these saliency algorithms for this particular application.

For this analysis, a random sample of 100 images is being retrieved from the Flickr8K dataset (Rashtchian et al., 2010) considering a variety of different images and categories. This sample contains uni-class and multi-class images with some instances of overlapping categories, in addition to images of different orientations, sizes and manipulations. The algorithms and pre-trained models were retrieved from open-source code generally through the researchers' official Github[2] repositories.

4.3.2 Metrics

To evaluate the performance of the algorithms, contours are playing an important role in identifying any loss of object data. This is being achieved by computing a count of the contours as well as the number of masks identified by the Mask R-CNN to compare the algorithms. In addition, a percentage of non-black pixels within the binary mask, saliency map and weighted mask is being calculated using the below equation (Equation 4.3):

$$\frac{n}{h * w} * 100 \tag{4.3}$$

where:

n = number of non-black pixels
h = height of the image
w = width of the image

The output of this equation will represent the percentage of present objects in an image before and after applying the element-wise multiplication between the instance segmentation algorithm and the saliency map. Ideally, this percentage is equal to both the Mask R-CNN binary mask and the weighted mask since this would conclude that no masks from the former algorithm were lost and given no salient weighting.

4.4 Scene Classification Analysis Procedure

The scene layer is constructed of a single deep learning model that performs dedicated feature extraction to extract information regarding the environment of the image. A further review is being conducted to analyse the best trained deep learning model for this application. This section provides further information on the methodology implemented for this part of the review.

4.4.1 System Overview

The procedure implemented to analyse scene classification is similar to what was implemented to analyse instance segmentation. Instead, however, four different CNNs: the AlexNet (Krizhevsky et al., 2012), ResNet-18, ResNet-50 (He et al., 2016) and DenseNet-161 (Huang et al., 2017) are being explored on the validation set of the Places-365 Standard Dataset. Every model being used is pre-trained on the Places-365 Standard training dataset by Zhou et al. (2017), the researchers that gathered and created this dataset.

The approach for this analysis as shown in Figure 4.7 identifies that the validation set of the Places-365 Standard dataset is being inferred by each model with the performance being evaluated individually and then analysed in retrospective to the other models, determining the best performing model for this application. Since the architectures are pre-trained on the same dataset and therefore utilising the same classes, no transfer learning is required.

The pre-trained models and datasets being used are retrieved from the researchers' Github[2] (Zhou et al., 2017). The researchers offer a variety of different models mainly

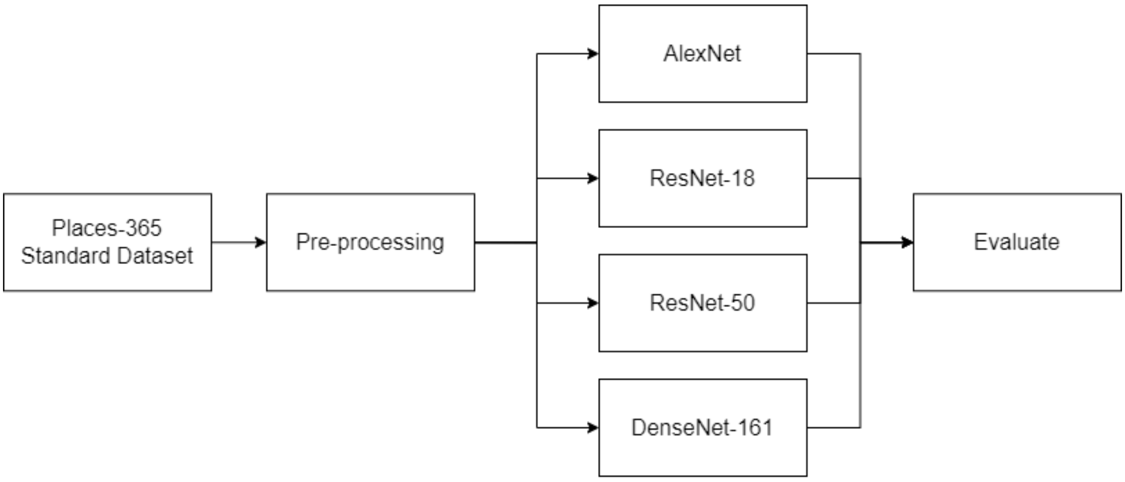


Figure 4.7: Block Diagram of the System utilised for the Scene Classification Analysis

trained or converted either for Caffe (Jia et al., 2014) or PyTorch (Paszke et al., 2019) use. The Caffe library (Jia et al., 2014) is a deep learning framework developed by Berkeley AI Research and community contributors in 2014. Although this library is still widely used and respected, it has in recent years been overshadowed by libraries such as Tensorflow (Abadi et al., 2015) and PyTorch (Paszke et al., 2019). Therefore, for this review, the PyTorch architectures are being used. In addition to offering the architectures, the researchers also offer an easy-to-use dataset that is targeted specifically for training the architectures using PyTorch. Before performing inference on the images to gather statistics, pre-processing was required on the images. In this phase of the evaluation, the images are being resized to 256 by 256 whilst the images are also being cropped at their centre to a size of 224 by 224. The image is then being transformed into a tensor (Channel x Height x Width) and normalised.

The models are being evaluated using a batch size of 16 since evaluating at a batch size of 64 or 32 results in the GPU encountering memory insufficient errors when inferring the DenseNet-161 model given that this algorithm is heavy on resources. Although the batch size does not have a direct impact on the accuracy of the model, for a fair comparison, especially in terms of inference time all the models are being evaluated using a batch size of 16. The evaluation of the models is being carried out based on the top-1 and top-5 accuracy as suggested by Zhou et al. (2017) to better determine the performance of scene classification models.

4.4.2 Dataset

As stated before Zhou et al. (2017) published their own dataset for scene classification: the Places dataset. The Places dataset is the largest current dataset compiled for scene classification consisting of a range of indoor, urban and natural environments estimated to amount to up to 98% of the places one will likely encounter throughout their lives. Different variations of this dataset exist varying in the number of categories such as the Places-88 and the Places-205 which contain 88 and 205 categories respectively (Zhou et al., 2015). The dataset being used for this review is Places-365 which consists of 365 categories. For this number of categories, there are two different versions varying in the size of the training set: the Places365-Standard and Places365-Challenge (Zhou et al., 2017) with the training set of the latter being significantly larger. For this review, having a larger validation set is more important for the evaluation, therefore, the Places365-Standard is being opted for. This dataset was downloaded from the researchers' Github (Zhou et al., 2017).

4.4.3 Metrics

To evaluate the scene classifier, general metrics used to measure the accuracies of CNNs are being used. Therefore, the predictions generated by the models are being compared with the ground truth. For this evaluation, as discussed by Zhou et al. (2017) the top-1, as well as the top-5 accuracies, are being considered due to scene ambiguity. Therefore the top five predictions are also being compared with the ground truth. These comparisons will determine the true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn). The accuracy is then calculated based on these values by identifying the ratio between the number of correct predictions over all the total predictions to describe the general performance of the models across all categories.

4.5 Image Captioning Architecture

The image captioning model being proposed for this research as discussed in Figure 4.2 consists of two complementary components: the encoder and the decoder. The encoder is responsible for extracting features from the training images which will then be provided to the decoder of the architecture to handle caption generation. In this section, implementation details will be discussed in further detail along with information regarding the hyper-parameters and training methodology.

4.5.1 The Encoder - System Overview

This research, based on the work conducted by Lim and Chan (2019), proposes a rich encoder that considers object and scene data for its feature extraction process. As shown in Figure 4.8, the training data is being progressed through two separate streams: the object layer and the scene layer each utilising a separate dedicated deep learning model. The object layer is responsible for extracting data related to the objects within the image whilst the scene layer aims to extract information regarding its environment. The features extracted from each layer are then being concatenated together to generate rich features for each image. The algorithms that are being used throughout the architecture are based on the in-depth research conducted in the previous sections.

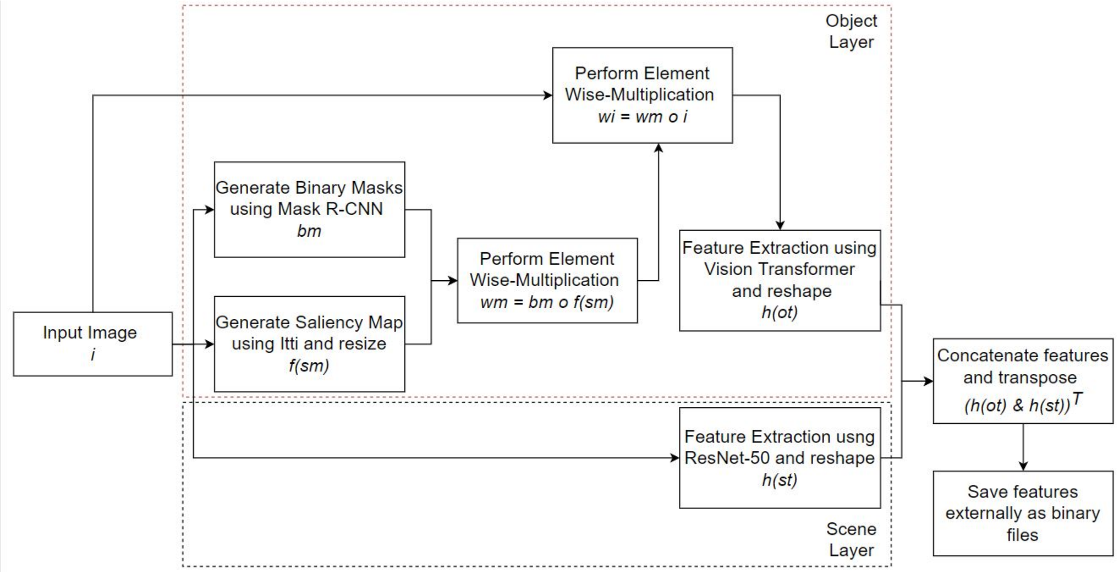

Object Layer
Perform Element Wise-Multiplication
wi = wm o i
Generate Binary Masks using Mask R-CNN
bm
Perform Element Wise-Multiplication
wm = bm o f(sm)
Feature Extraction using Vision Transformer and reshape
h(ot)
Input Image
i
Generate Saliency Map using Itti and resize
f(sm)
Concatenate features and transpose
(h(ot) & h(st))^T
Feature Extraction usng ResNet-50 and reshape
h(st)
Save features externally as binary files
Scene Layer

Figure 4.8: Block Diagram of the Encoder consisting of two layers: the object layer and the scene layers

Commencing with the mask layer, a process is being implemented to generate the weighted images as presented in Equation 4.4.

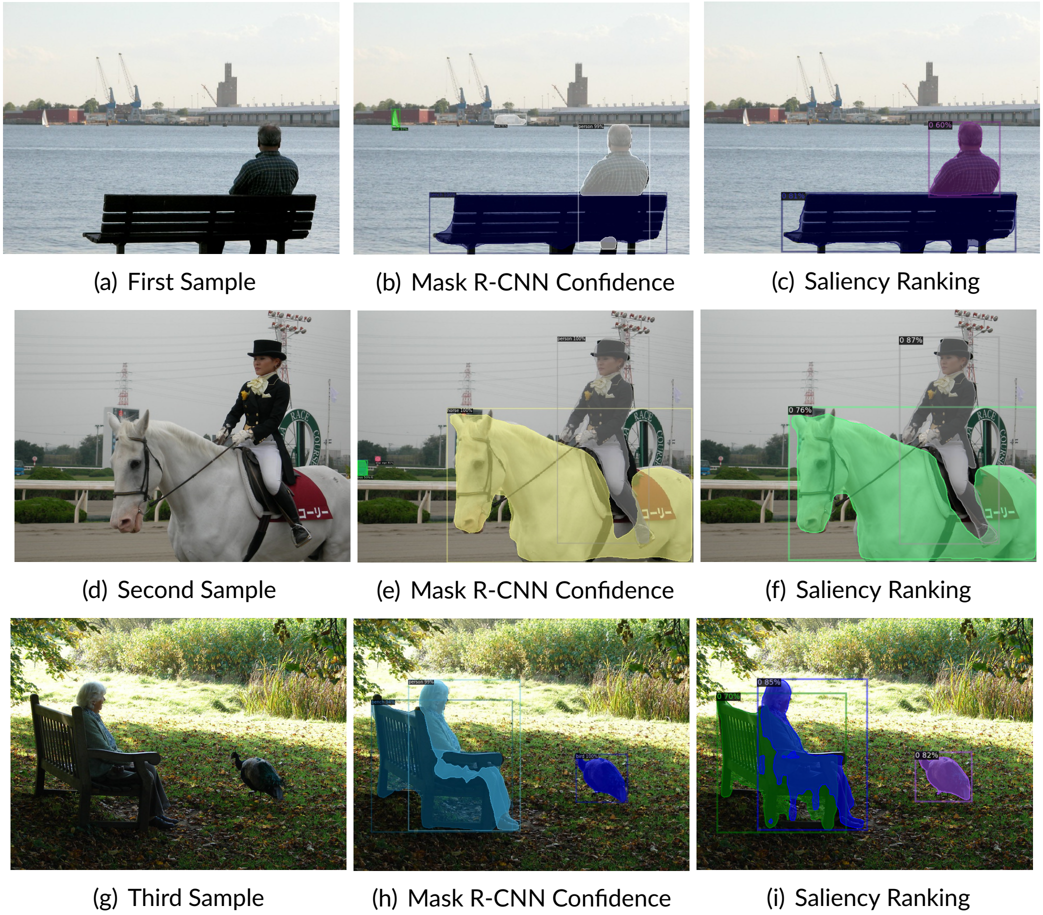
$$wi = \sum_{i=1}^{n} b_i \odot f(sm) \odot i \tag{4.4}$$

where:

$$\begin{aligned}
wi &= \text{weighted image} \\
i &= \text{original image} \\
n &= \text{single binary mask} \\
b &= \text{binary masks} \\
sm &= \text{saliency map} \\
f &= \text{resize function}
\end{aligned}$$

Firstly, binary masks *b* are being generated utilising an instance segmentation architecture, particularly the Mask R-CNN with a threshold of 0.5. Each mask instance *n* outputted is then being concatenated to a single image and converted to a binary mask *bm* setting each pixel belonging to the mask as white. In parallel, a saliency map *sm* is being generated using the traditional Itti's saliency prediction algorithm. The saliency map generated is then reshaped and converted to an RGB image corresponding to the shape and channel format of the images of the binary masks ensuring that image dimensions are of the same size. After normalisation element-wise multiplication is calculated between the two generated images. This results in the generation of weighted masks *wm* that show the distribution of the attention of the masks based on the saliency. It could be identified that this approach differs from that proposed by Lim and Chan (2019). These researchers consider the confidence level of the masks predicted by the instance segmentation algorithm as a good indicator of the saliency of the object and base the architecture's attention mechanism on this variable to identify mask features by computing element-wise multiplication between the binary masks and their respective generated confidence score. The architecture proposed for this research argues with this concept by highlighting that the confidence level outputted by the segmentation algorithm does not necessarily highlight the importance and the saliency of an object within the image but rather the model's assertiveness in outputting the correct mask with its corresponding class label. To drive this argument, a saliency ranking algorithm proposed by Fang et al. (2021) is being consulted to determine if the confidence level of an instance segmentation algorithm can be a strong basis for an attention mechanism. Utilising this algorithm to generate saliency ranking for a couple of images as well as exploiting the confidence level of the Mask R-CNN architecture to compare its effectiveness it could be identified that the confidence level is in fact not a strong basis for saliency ranking.

Firstly, it could be identified that in some instances such as for Figure 4.9(a), the Mask R-CNN manages to distinguish and predict a label with a high confidence objects that are not considered salient by the saliency ranking algorithm. In fact, the instance segmentation algorithm identifies two objects in the background with a confidence score of 97% and 51%. This contradicts the saliency ranking algorithm in which no saliency rank was



(a) First Sample (b) Mask R-CNN Confidence (c) Saliency Ranking

(d) Second Sample (e) Mask R-CNN Confidence (f) Saliency Ranking

(g) Third Sample (h) Mask R-CNN Confidence (i) Saliency Ranking

Figure 4.9: Difference between utilising a Saliency Ranking algorithm and the Confidence Score of an Instance Segmentation algorithm as an Attention Mechanism

outputted since these objects are not located in the foreground of the image and are not the main focus of the image but form part of the scenery. Moreover, it also contradicts the definition of visual attention in which it describes how cognitively the brain reduces the information received for processing by focusing on a single area for a more detailed evaluation approximately as large as an outstretched thumb (Seychell, 2021). In this case, elements from different areas of the image are being given an equal weighting of attention which is not aligned with the true definition of saliency and visual attention. A similar circumstance can be identified in the second sample Figure 4.9(d) in which objects from the background are being identified by the Mask R-CNN and given a high saliency weighting. A further consideration is the level of saliency. In this sample image, the instance segmentation algorithm identified the main two objects of the image with definite certainty of

100%. Differently, the saliency ranking algorithm provided both objects with a different standing rank of saliency highlighting the most salient object in the image and its succedents. Finally, as shown in Figure 4.9(g) although different confidence is outputted by the instance segmentation algorithm this is not reflected by the ranking provided by the dedicated saliency ranking algorithm since a different rank standing is being outputted by this algorithm. Following these arguments, in the architecture of this research, a saliency algorithm is being utilised to generate a salient value for every object detected by the Mask R-CNN algorithm to better interpret the visual attention distribution of the image. As explained before, Itti's saliency algorithm is being utilised given that it provides a saliency score for every pixel within the image ensuring that any mask generated by the Mask R-CNN is given a weight and therefore no data is lost.

The final object-orientated image *wi* that will be used for feature extraction for the object layer is then computed by performing element-wise multiplication between the normalised generated weighted masks image and the raw image *i*. As shown in Figure 4.10, the weighted image highlights the most relevant and salient sections of the images serving as an attention mechanism. To perform object feature extraction, a pre-trained vision transformer (Dosovitskiy et al., 2021) is being utilised instead of the ResNet-50 utilised by Lim and Chan (2019). The vision transformer is a novel architecture that manages to supersede current state-of-the-art architectures in computational efficiency and accuracy whilst demonstrating great scalability. For these reasons, the vision transformer was introduced to the object layer for feature extraction allowing for a deeper exploration of the capabilities of this novel architecture. Dosovitskiy et al. (2021) in addition to the architecture of vision transformers explored a hybrid vision transformer that combines vision transformers with CNNs replacing the image patches with feature maps extracted by the CNN and applying the patch embedding projection to the patches extracted from the feature maps. The researchers concluded that hybrids slightly outperformed vision transformers at computational costs with the difference gradually diminishing for larger models. For this extraction, a hybrid vision transformer referred to as the R50+ViT-L/S32 hybrid pre-trained on the ImageNet dataset is being explored. This architecture utilises 24 layers with a ResNetV2 baseline down-sampled by a ratio of 32 and trained on 384 by 384 sized images. This architecture trained on the ImageNet dataset achieved a top-1 accuracy of 97.86% and a top-5 accuracy of 99.67%. Before passing the images through the transformer for feature extraction, pre-processing is being performed on the images. The inputted images are being resized to a size of 384 by 384 with a bicubic interpolation and cropped at the centre by the same size. Normalisation is then being performed with a mean of (0.5, 0.5, 0.5) and a standard deviation of (0.5, 0.5, 0.5). After pre-processing, the images are being loaded using a custom data loader at a batch size of 64. The features extracted of shape (1024,) are being saved externally as binary files.

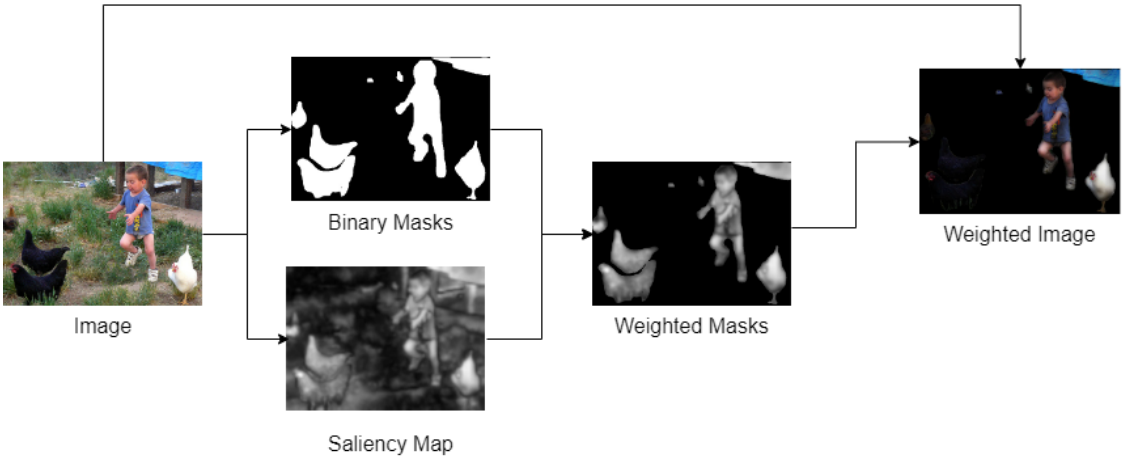


Figure 4.10: Image Processing Technique used to create the weighted images used by the Object Layer in the Image Captioning Encoder. The weighted image as observed focuses on the main objects in the image with a sharpness that corresponds to its saliency.

The scene layer aiming to perform feature extraction at a scene level follows a simple procedure of performing the necessary pre-processing required by the pre-trained scene classifier. Following the research conducted, the ResNet-50 pre-trained on the Places-365 dataset is being utilised. Although the same architecture considered by Lim and Chan (2019) is being utilised, the model being used for this research has been pre-trained on the Places365 dataset instead of the ImageNet dataset as proposed by the researchers. This alteration spurs from the argumentative discussion carried out by researchers such as Hayat et al. (2016) and Zhou et al. (2015) in which it was analysed that scene classification is a distinct task from object classification and therefore requires separate training material, models and in some cases also different architectures. These researchers highlighted that a CNN trained on object-orientated datasets such as that of the ImageNet dataset although provides excellent results for object classification does not reach up for the task of scene classification. Zhou et al. (2015) enhanced this discussion by comparing a CNN trained on the ImageNet dataset with that trained on the Places365 dataset coming to an identical conclusion. One of the key causes of this as discussed by Herranz et al. (2016) is that these CNNs trained on different datasets are tuned for different scale ranges that fit the requirements of object and scene classification. Following these discussions, a ResNet-50 pre-trained on the Places-365 dataset is being utilised. As required by this model, the images are being sized to a shape of 256 by 256 and cropped to the centre by 224. Normalisation is then being performed with a pre-defined mean and standard deviation of (0.485, 0.456, 0.406) and (0.229, 0.224, 0.225) respectively. The images are being loaded using a custom data loader at a batch size of 64 and are being inferred by the deep learning model without its two final layers, therefore, omitting the final fully-connected layer and transforming the classification model into a feature extraction model. The features extracted are being reshaped to (49, 2048) and stored as binary files.

The features extracted *f* from the object layer *ot* and the scene layer *st* are then being concatenated together using a simple concatenation function as displayed in equation 4.5.

$$f = (h(ot) \,\&\, st)^{T} \tag{4.5}$$

where:

ot = object tensor
st = scene tensor
h = reshape function
$\&$ = concatenation
T = transpose

Since the shapes of the features generated differ from one another along both axes, reshaping is required. To achieve this the maximum shape of both features is being computed by adding a dimension to the object extraction tensor and retrieving the maximum dimensions of both axes. After reshaping, both features are being concatenated together and transposed generating the final features of shape (49, 4096). These features are being stored as binary files to be utilised by the decoder of the architecture.

4.5.2 The Decoder - System Overview

The image captioning decoder is making use of a transformer that is targeted at performing image captioning. The transformer as discussed before refrains from utilising any convolutions and recurrences and instead employs stacks of self-attention layers. The architecture of the transformer being used for this research follows the architecture of the work conducted by Vaswani et al. (2017) on the language transformer and is also influenced by the research conducted by Xu et al. (2015) and Zhu et al. (2018) on image captioning.

The architecture of the image captioning as analysed in Figure 4.2 consists of an encoder and a decoder. Starting with the encoder, it could be analysed that this architecture strays from the architecture of the original language transformer. This is due to the language transformer being developed for sequence-to-sequence tasks and therefore focuses on transforming an input sequence from one domain to an output sequence in a different domain for example translating a sentence from Maltese to English. Image captioning is a different application to the traditional transformer to which an image-to-sequence task is required. To accommodate this functionality, the transformer was developed from first principles with an encoder inspired by the work of Xu et al. (2015), Lim and Chan (2019) and Zhu et al. (2018). The encoder of the image captioning architecture as discussed in the previous sections consists of two separate layers: the object layer and the scene layer which both contain distinguishable processes and deep learning models for feature extraction. The concatenation of these features is being used as input to the language transformer aiming to provide a soft attention mechanism and global image information. As part of the encoder, the features extracted are being passed through a fully connected linear layer and a ReLU activation function to obtain a $\mathrm{d}_{\mathrm{model}}$-dimensional image where $\mathrm{d}_{\mathrm{model}}$ represents the dimensionality of the representations expected in the second sub-layer of the transformer's decoder. The encoder consists of 6 identical layers with each layer performing the above-mentioned functionality.

The transformer decoder architecture implemented follows the general structure of the traditional language transformer. The target sentences associated with each image are being represented as numerics or vectors and passed as inputs to the transformer. Prior to this transformation, the target sentences are being pre-processed by firstly filtering out any punctuation and numerics. For training and validation target sentences start and end tokens are being prepended and appended to aid the model in distinguishing the beginning and the end of the captions. The architecture's dictionary is being determined through the frequency of the words in the target sentences with the highest 5,000 or 10,000 words identified constituting the dictionary. This selection is based on each caption being tokenised to gather single words which are being converted to lowercase and re-filtered for any punctuation preserving special characters such as "<" and ">". In addition, the captions are being padded to a specified number of words, varying the sentence length to between 20 and 50. For this research, the vocabulary size and the maximum length of the sentences with which the model is being trained are being considered as contributing factors that potentially enhance readability and therefore will be varied as part of the experimentation of this research.

After pre-processing of the sentences, text encoding is being conducted. Text encoding is an important process for any architecture to help preserve the context of the words and to aid the architecture to distinguish patterns. Vaswani et al. (2017) in the implementation of the language transformer do not specify the embedding utilised for their system. Therefore a variety of techniques such as Index-Based Encoding, One-Hot

Encoding, Glove Encoding and Bert Encoding are being considered for this architecture. Index-Based Encoding is the most simple encoding in which each word is given a unique number identifier, therefore, converting the textual sentences to numerical representations. Differently, in one-hot-encoding, each word is transformed into a unique vector consisting of only binary digits in which a word in the corpus is represented as 1 whilst any other word is represented as 0. A sentence is therefore represented as an array of vectors with each element representing a single word with the binary digit 1 at the location of the word. The drawback of using such a technique is that with a bigger corpus, the feature space grows drastically resulting in possible high memory consumption. A more advanced system is the Global Vectors for Word Representation (GloVe) (Pennington et al., 2014) developed by Stanford University which through an unsupervised learning algorithm generates embeddings by identifying the relationship between the words from statistics gathered from the corpus. The Bidirectional Encoder Representations from Transformers (Bert) (Devlin et al., 2018) is a recently developed revolutionary transformer that has managed to achieve state-of-the-art performance in various NLP applications. One of its many applications is to produce embeddings to be used as input to other architectures. Due to its dynamic capabilities, Bert has an advantage over systems such as GloVe (Pennington et al., 2014) or Word2Vec (T. Mikolov et al., 2013) since it can produce embeddings for each word based on its context, having the additional capability to target homonyms. For this architecture, the above-mentioned word encodings are being considered as will be furtherly analysed in the Evaluation Section. From these encodings, One-Hot-Encoding was deemed not feasible due to memory usage constraints given that each target sentence is being represented using an array of vectors. Therefore, the main contenders consisted of the Index-Based, GloVe and Bert Encoding to which it was concluded that comparable results were generated with the Index-Based encoding generating a very slight increase in performance. This could be due to the decoder of the transformer consisting of an embedding layer that allows the architecture to learn its word embeddings in addition to calculating its positional encoding aiming to analyse the context of the word within the sentence. Therefore using a simple untrained encoding might give a slight competitive advantage due to its simplicity.

As discussed the encoded target sentences are being processed through the embedding layer to represent each word in $\mathrm{d}_{\mathrm{model}}$-dimensional space. Positional encodings are then calculated and summed to the word embeddings. Given that the transformer does not contain any recurrent or convolutional layers positional encodings are crucial since they provide the model with information regarding the relative position of the words in the sentence. Similar to the original research the positional encodings are calculated as shown in Equation 4.6.

$$\begin{aligned} PE_{pos,2i} &= sin(pos/10000^{2i/d_{model}}) \\ PE_{pos,2i+1} &= cos(pos/10000^{2i/d_{model}}) \end{aligned} \tag{4.6}$$

where:

i = individual dimension of embedding
pos = position of word in sequence
d = size of word embedding
d_{model} = d-dimensional space

Having gathered information regarding the target sentence, the architecture progresses to the first decoder layer with this vector as input. The decoder consists of 6 identical layers similar to that in the original research paper. The first module in the layer consists of the Masked Multi-Head Attention Module. This module performs self-attention generating an attention vector for every word to identify the relationship between each word to every other word within the same target sentence. Referring to the name of this module, it could be identified that this module utilises a masking technique to hide from the model future tokens within the target sentence. Therefore, the model only has visibility to the encoder vectors and previous words of the target sentence whilst predicting the next word. This allows the model to be trainable as well as facilitates parallelisation. These generated vectors along with the vectors generated from the encoder layer are then passed through the second module referred to as the Multi-Head Attention Module. This module is the second attention mechanism within this architecture that focuses on identifying relationships between each word in the target sentence with the features extracted from the input images passed through the encoder. It is at this phase that the mappings between the images and the target sentences are established. The outputted attention vector representing these relationships is forwarded to a feed-forward network preparing the output vector for the next iteration of the decoder layers or the final linear layer. After every sublayer, it could be observed that there is a residual connection around to avoid the vanishing gradient problem followed by a normalisation layer. The final linear layer is a fully connected dense layer used to expand the dimensions to the size of the target vocabulary with a final softmax layer used to convert it into a probability distribution.

As an optimisation, the transformer utilises the Adam optimiser with a customised learning rate. For this architecture, experimentation was carried out on the hyper-parameters with the optimal parameters found to be a $\mathrm{d_{model}}$ set to 512 with the dimension of the feed-forward network model set to 2048. The number of heads set for the multi-head attention module is 8 whilst the number of encoder layers and the number of decoder layers are set to 6 each. Given its image-to-sequence applicability, different transformer architectures could not be explored since to date, dedicated pre-defined image-to-sequence transformers are not readily-made available in libraries such as HuggingFace [3] and therefore each architecture is required to be built from scratch.

4.5.3 Datasets

Dedicated datasets have been developed for the implementation of image captioning models. The most commonly used datasets consist of the Flickr8K (Rashtchian et al., 2010), Flickr30K (Young et al., 2014) and the MS-COCO (Lin et al., 2014) captions dataset, all of which are considered as benchmark datasets for image captioning. The Flickr8K (Rashtchian et al., 2010) and the Flickr30K (Young et al., 2014) are constructed from the social network and photo-sharing platform Flickr which datasets consist of 8,091 and 31,783 images respectively. Each image is being mapped to 5 unique captions, therefore, amounting to 40,455 and 158,915 different captions for the images available. The MS-COCO Captions dataset (Lin et al., 2014) is large scale dataset consisting of over 330,000 images with 5 human-generated captions for each image, amounting to 1,650,000 different captions. Considering the computational power available and the feasibility of this research, the image captioning model proposed is being trained on the Flickr8K and the Flickr30K.

4.5.4 Metrics

Metrics have been developed for the evaluation of image captioning models focusing on the readability and human-like attributes of the captions generated. The Bilingual Evaluation Understudy Score (BLEU) (Papineni et al., 2002) metric was one of the first metrics developed to evaluate the generated sentences in relation to a provided set of reference sentences. This metric as shown in Equation 4.7 considers the counts of matched n-grams between the generated sentence and the reference sentence providing a score between 0 and 1 depending on the similarity.

$$BLEU = BP \cdot \exp(\sum_{n=1}^{N} w_n \log p_n) \tag{4.7}$$

where:

[3] HuggingFace is an open-source NLP library that provides models for a variety of applications available at https://huggingface.co/

p_n = modified precision for n-gram
w_n = weight between 0 and 1 for log p$_n$
BP = Brevity Penalty for short machine captions which is calculated using the below Equation:

$$BP = \begin{cases} 1 & \text{if } c > r \\ \exp(1 - \frac{r}{c}) & \text{if } c \leq r \end{cases} \tag{4.8}$$

where:

c = total length of candidate sentences
r = average length of all reference sentences

The Metric for Evaluation of Translation with Explicit ORdering (METEOR) (Denkowski and Lavie, 2014) is another metric targeted at evaluating generated captions with the aim of addressing the shortcomings of the BLEU metric. Therefore, this metric focuses on human judgement and the ability of the sentences to sound as human as possible instead of performing a high-level evaluation at corpus level. To achieve this, this metric depends on the harmonic mean of the unigram precision and recall with the recall being given a higher weighting in order to perform the evaluation. The equations implemented to compute the precision, recall and harmonic mean for the Meteor equation can be analysed in Equation 4.9.

$$\begin{aligned} PrecisionP &= \frac{m}{w_t} & RecallR &= \frac{m}{w_r} \\ F_{mean} &= \frac{10PR}{R + 9P} & Meteor &= F_{mean}(1 - p) \end{aligned} \tag{4.9}$$

where:

m = Number of unigrams in the candidate caption also found in reference
w_t = Number of unigrams in candidate caption
w_r = Number of unigrams in reference caption
p = Chunk penalty for a set of consecutive words in the candidate that map to chunks in the reference computed using the below Equation:

$$p = 0.5(\frac{c}{u_m})^3 \tag{4.10}$$

where:

c = number of chunks in candidate
u_m = unigrams in candidate

A further evaluation metric is the Consensus-based Image Description Evaluation (CIDEr) (Vedantam et al., 2015) metric that was created by Microsoft Research and forms part of the MS-COCO Caption Evaluation Server. This metric bases its evaluation mechanism on the human consensus and is computed based on the average cosine similarity between the candidate and reference sentence accounting for both precision and recall. The CIDEr metric is being described in Equation 4.11.

$$CIDEr_n(c_i, S_i) = \frac{1}{m}\sum_j \frac{g^n(c_i).g^n(s_{ij})}{||g^n(c_i)||||g^n(s_{ij})||} \tag{4.11}$$

where:

$g^n(c_i)$ = vector formed by $g_k(c_i)$ related to all n_grams of length n
$||g^n(c_i)||$ = magnitude of all the vector $g^n(c_i)$
$g^n(s_{ij})$ = vector formed by $g_n(s_{ij})$ related to all n_grams of length n
$||g^n(s_{ij})||$ = magnitude of all the vector $g^n(s_{ij})$

Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence (ROUGE-L) (Lin, 2004) is yet another metric dedicated to performing image captioning evaluation. This metric takes into consideration sentence-level structure similarity and aims to identify the longest co-occurring n-grams in sequences by making use of the Longest Common Subsequence (LCS) statistic. The Rouge-L metric is based on the F-Score constructed of the precision and recall as shown in Equation 4.12.

$$\begin{aligned} PrecisionP &= \frac{\sum_{r \in reference} |LCS_{\cup}(candidate, r_i)|}{count(candidate)} \\ RecallR &= \frac{\sum_{r \in reference} |LCS_{\cup}(candidate, r_i)|}{count(reference)} \\ Rouge\_L = max_k\{&\frac{1+\beta^2 RP}{R+\beta^2 P}\} \end{aligned} \tag{4.12}$$

where:

β = relative importance of the precision and recall

Throughout this study, the Bleu, Meteor, CIDEr and Rouge-L metrics are being utilised through the MS-COCO evaluation tools to determine the performance of the image captioning model trained on the Flickr8K and Flickr30K.

4.5.5 Training and Evaluation

The image captioning models are being trained using the well-known Karparthy split (Karpathy and Fei-Fei, 2015) to separate the datasets. The Karparthy split was created by Andrej Karpathy and since its development, it has become widely used in the training of image captioning models. Following the works of other researchers, the Flickr8K and the Flickr30K are being divided using this split, therefore dedicating 1,000 images for validation, a further 1,000 images for testing and the rest for training.

Utilising the annotation file provided, a dictionary is being created and used throughout this study as a point of reference containing information regarding the split and respective five captions of a given image using its filename as a unique identifier. An input pipeline for the architecture is being set up considering all the necessary optimisation techniques to load data most efficiently. Features extracted from the encoder stored externally as binary files are being mapped and loaded in parallel, shuffled and batched using a buffer size of 1000 and a batch size of 64. The prefetch technique is also being utilised aiming to reduce extracting time by overlapping the retrieval and the model execution. Therefore, while executing the current training step, this technique in parallel is reading the data that will be used for the subsequent training step. In the case of validation data, given that this split is relatively small, this is being cached in memory for easy retrieval. Furthermore, the data files are being ensured to be within the same environment of the executing script to reduce the reading time between different mediums drastically reducing the training time by more than half whilst also making the most of the GPU resources.

Throughout the training of the image captioning model, a loss function utilising the Sparse Categorical Cross Entropy is being used to compute the cross-entropy between the target caption and the generated caption during the training and the validation of the model. A different approach is taken for the accuracy by comparing the prediction with the target using an *and* operation for both the training and validation. However, given the output of the image captioning model, this is not an accurate representation of the performance of the model and therefore the image captioning metrics discussed before are being utilised to measure its performance and the readability. The loss and accuracy functions are being called from a customised training and validation step used to train and evaluate the transformer and update the optimiser. The model is being trained for a total of 20 epochs outputting information every 100 batches and saving a checkpoint after every epoch.

In addition, two different inference models are being developed to create captioners capable of performing image inference. The first inference model is employing a greedy search algorithm in which the model is constructing the predicted sentence gradually choosing the most optimal word available at the current step. Differently, the second inference model is using the beam search heuristic algorithm. This algorithm expands its search space to N best alternatives determined by the beam width when choosing a predicted token for a given position with the best predicted sentence being distinguished through conditional probability. For this research, the beam-width size is being set to 3. Both inference models are being exported and used to infer the testing images, saving the predictions in COCO format. The generated predictions are then being compared with the ground truth and using the before-mentioned metrics, the performance of the model is determined.

4.6 Conclusion

This chapter provided a holistic description of the methodology employed for this research. Starting with an overview of the image captioning architecture, this chapter delves into each component separately describing the techniques utilised. The review of instance segmentation algorithms is the first analysis approach explored. This sub-section contains an overview of the system considered with a description of the datasets involved and the metrics that will be used for this component. Saliency Prediction is then identified and its methodology is explored as an attention mechanism. Following the scene classification analysis describing the overall system employed in addition to the datasets and metrics used is investigated. This section then explores the image captioning architecture which is constructed from the previously conducted reviews. Starting with the encoder, this chapter explored its system methodology with an argumentative discussion presented to justify the logic of this system. The decoder is then delved into to generate textual descriptions from the features extracted from the encoder. Finally, datasets and metrics used throughout the training and analysis of the image captioning architecture are described ending with an overview of the training and evaluation employed.

Chapter 5

Evaluation

To evaluate this system, an evaluation framework is presented as described in the introductory section of this chapter. This chapter commences by considering the reviews conducted as part of the construction of the encoder and continues to evaluate the results and findings on the trained architecture performing also a comparison with other similar architectures. This chapter concludes with an evaluation on readability.

5.1 Evaluation Overview

The evaluation for this research as highlighted by the presented framework in Figure 5.1 is composed of three different phases. The first evaluation conducted is within the construction of the encoder of the image captioning model, whereas the second phase of the evaluation analyses the overall general performance of the captioning model and determines if it is competitive with the current state-of-art architectures and baseline models. The last and final phase of the evaluation identifies the model's effectiveness and whether the vocabulary size and the sentence length of the trained captioning model successfully provide an improvement to the performance and readability.

5.1.1 Evaluation Structure

In the subsequent sections, the evaluation of the methodology discussed in the previous chapter will be conducted following the order below:

1. Image Captioning Model Encoder (Section 5.2) - This section discusses the reviews conducted during the construction of the encoder, starting with a discussion of the gathered results from the instance segmentation algorithms (Section 5.2.1). This

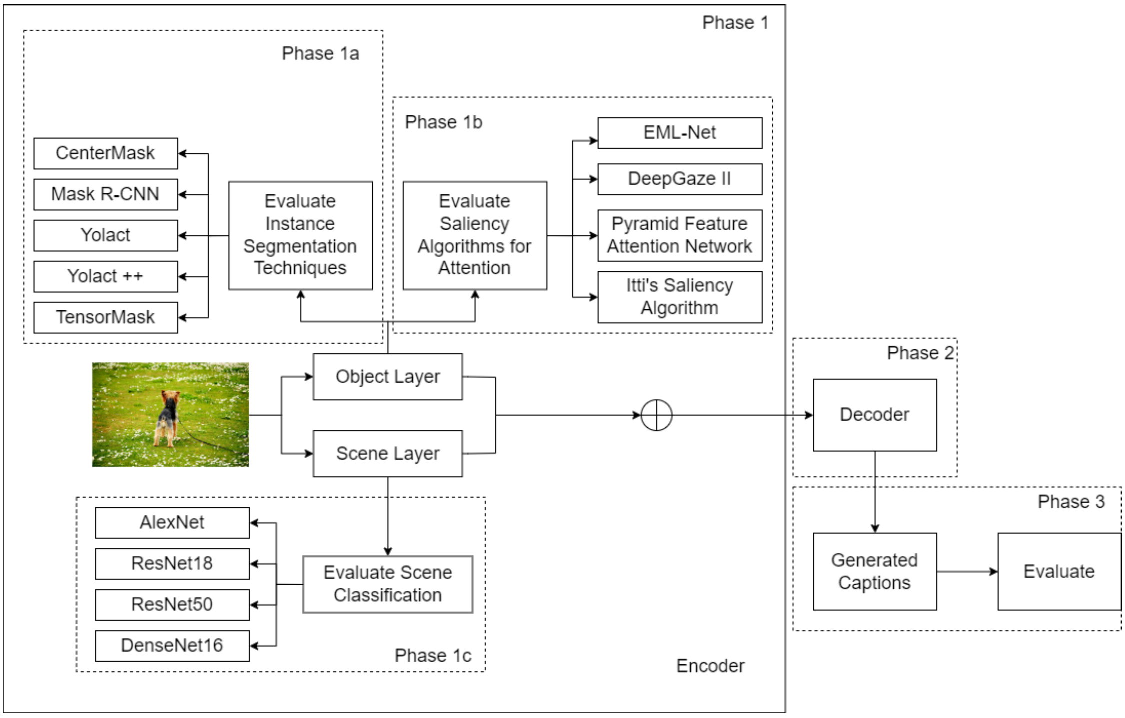


Figure 5.1: Block Diagram of the Evaluation plan consisting of three phases: the first phase is the evaluation during the construction of the architecture, the second phase consists of the evaluation of the trained models and the third phase consists of the readability aspect of the research.

section will then delve into a discussion of the use of saliency algorithms as an indicator of valid masks and discuss the results achieved from different saliency detection algorithms (Section 5.2.2). This section concludes with the conclusions drawn from the review conducted on scene classification models (Section 5.2.3)

2. Image Captioning Model (Section 5.3) - This section starts with a discussion on the experimentation conducted on the text encodings and hyper-parameters of the image captioning architecture. This section concludes by comparing the performance of the trained model with other similar architectures in addition to providing sample captions generated by the trained architecture.

3. Sentence Generation and Readability (Section 5.4) - This section handles the evaluation conducted with respect to the sentence length and whether varying hyper-parameters related to the trained and evaluated sentence length and the vocabulary size have a direct impact on the generated sentence length and the performance of the image captioning architecture.

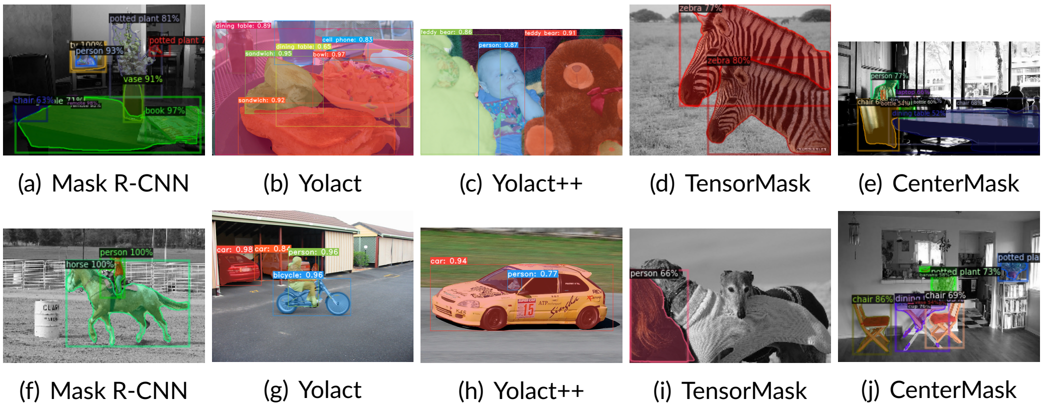
5.2 Image Captioning Model Encoder

As explained prior, the encoder of the image captioning model consists of two separate branches: the object branch that identifies items within the image through the use of an instance segmentation architecture with weightings given through a saliency prediction algorithm and the scene branch that identifies the environment of the image through a CNN. Both layers make use of trained models chosen after conducting a thorough evaluation of the currently available pre-trained models to ensure that the best performing model is used for the construction of the encoder. In this sub-section, an analysis of different instance segmentation architectures will be conducted to establish the current state-of-the-art. Moreover, different saliency prediction algorithms will be explored in relation to providing a salient weight to objects identified through the instance segmentation algorithm. Furthermore, an analysis of different neural network architectures for scene classification will be performed.

5.2.1 Instance Segmentation Review

The Mask R-CNN is still referred to by many as the current state-of-the-art instance segmentation technique. In this evaluation, four other recently developed instance segmentation techniques, the Yolact (Bolya et al., 2019), Yolact++ (Bolya et al., 2020b), TensorMask (Chen et al., 2019) and CenterMask2 (Lee and Park, 2020) will be considered to determine whether the Mask R-CNN is still the current state-of-the-art. This is being achieved by performing an analysis of these architectures on two datasets: the validation set of the MS-COCO17 (Lin et al., 2014) and the training set of the Tiny Pascal VOC(Everingham et al., 2010).

The first evaluation conducted was to verify whether the architectures were correctly loaded and configured. Therefore a couple of images were inferred using each model with a 0.5 inference threshold. Some of these samples for both datasets can be analysed in Figure 5.2. Here, one can analyse that each architecture gave clear results and managed to also output masks and labels of smaller items within the image with a good level of confidence. The architectures were then evaluated using the before-discussed evaluation metrics on the validation set of the MS-COCO17 and the training set of the Tiny Pascal VOC. The first analysis to be discussed is the evaluation of the architectures on the validation dataset of the MS-COCO17. As shown in Table 5.1 the best performing model was the Mask R-CNN with an AP score of 35.59 followed by the Yolact++ and CenterMask2 which achieved a similar AP to each other with a score of 30.57 and 30.31 respectively. The worst performance on this dataset was that of the Yolact and Tensor-



(a) Mask R-CNN (b) Yolact (c) Yolact++ (d) TensorMask (e) CenterMask

(f) Mask R-CNN (g) Yolact (h) Yolact++ (i) TensorMask (j) CenterMask

Figure 5.2: Sample Instance Segmentation Inference Images on the MS-COCO17 validation set (top) and the Tiny Pascal VOCtraining set (bottom)

Mask which achieved an AP of 24.89 and 26.02 respectively. Although the performance of the Yolact++ was significantly better than that of its precedent the Yolact, this was still not as good as that achieved by the Mask R-CNN. Bolya et al. (2019) in the Yolact architecture focused their research on real-time inference rather than the accuracy of the model with the research conducted on the Yolact++ shifting their focus on increasing the accuracy. Although a distinguishable increase in performance was noticed, this was not sufficient to compete with the famous Mask R-CNN on this dataset. The TensorMask researchers Chen et al. (2019) rather than aimed to compete with the current state-of-the-art, focused on identifying a different perspective to the instance segmentation task through the use of the dense sliding-window technique. Therefore, a lower AP score was expected.

The architectures with the same configurations were then analysed on the training set of the Tiny Pascal VOC which is a different dataset than that the architectures were exposed to during training. The performance of the architectures as can be analysed in Table 5.1 gave a rather different outcome than that established before. Differently, the Yolact++ achieved the best AP score out of all the architectures considerably with a value of 47.59. Therefore it could be concluded that on this dataset, Bolya et al. (2020b) managed to reach the aim of their research by surpassing the performance of the Mask R-CNN while possibly retaining their real-time application which will be evaluated furtherly in this section. In fact, the Mask R-CNN which gave the highest AP on the MS-COCO17 ranked second with an AP of 43.93. The Yolact and the CenterMask2 performed relatively similar to each other with a score of 40.39 and 39.98 respectively. This time, the TensorMask architecture performed the worst of all the architectures.

Table 5.1: Instance Segmentation Evaluation on the validation set of the MS-COCO17 and the training set of the Tiny Pascal VOC distinguishing the best performing models by their high scores identified by the underlining.

	MS-COCO17			Tiny Pascal VOC		
	AP	AP-50	AP-75	AP	AP-50	AP-75
Mask R-CNN	**35.59**	**54.31**	**38.67**	43.93	**72.62**	46.73
Yolact	24.89	37.60	27.15	40.39	64.09	43.17
Yolact++	30.57	45.14	33.46	**47.59**	72.59	**51.68**
CenterMask2	30.31	43.93	33.80	39.98	64.23	43.25
TensorMask	26.02	37.38	29.04	35.53	59.06	37.22

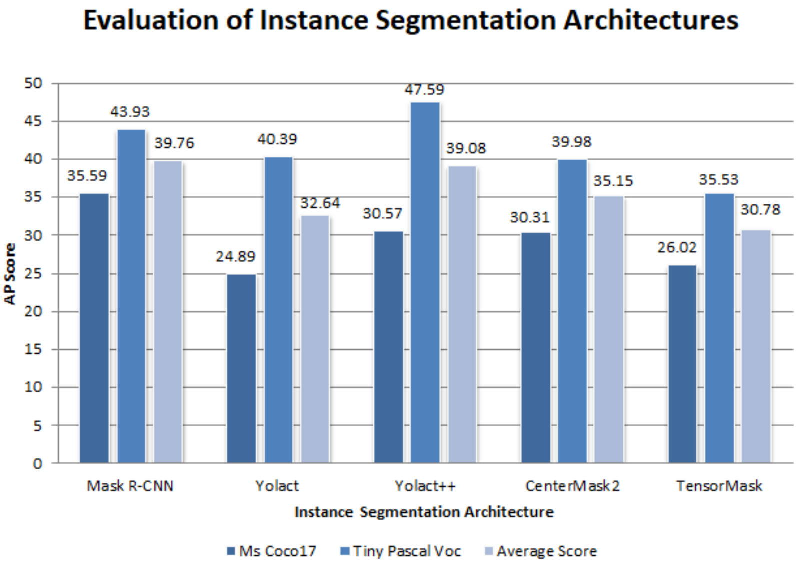


Figure 5.3: Comparing the evaluation of the Instance Segmentation Architectures on the MS-COCO17 Validation with the Training Set of Tiny Pascal VOC.

The performances of the architectures on both datasets were then compared for further evaluation as shown in Figure 5.3 generating also an average score. Overall it could be analysed that the results on the Tiny Pascal VOC were slightly higher than those produced on the MS-COCO17 for all the architectures. This could be due to the fact that the Tiny Pascal VOCdataset contains fewer categories than the MS-COCO17 dataset, including the person category which is the most occurring class. For a better comparison, an average score was calculated for each architecture giving the performance on each dataset an equal weighting. From this evaluation, it could be concluded that the Mask R-CNN and the Yolact++ gave the highest AP scores with a similar score of 39.76 and 39.08 respectively. These were followed by a slightly lower score from the CenterMask2 with an AP of 35.15. The architectures with the lowest average score were the Yolact and the TensorMask with an average AP of 32.64 and 30.78 respectively. It could be noticed that the AP score for all the architectures was relatively low. This is due to the way the AP is calculated. As discussed before, the MS-COCO evaluation takes into consideration a range of IoU threshold values between 0.5 to 0.95 with a step size of 0.05, calculating an mAP for each IoU. The final mAP is then calculated by averaging over the IoU at each threshold. Using such a technique although reducing threshold bias gives low scores since higher thresholds are also considered with an equal weighting hence reducing the average score.

Moreover, a deeper evaluation was conducted on the Mask R-CNN, CenterMask2 and TensorMask since these architectures were evaluated through the use of the Detectron2 library which allows for insights into the performance of each separate class. Starting with the MS-COCO17 dataset, an evaluation of the 80 classes present in this dataset shows that the AP for every class generally follows the same pattern with the Mask R-CNN being slightly higher than the other architectures. The CenterMask2 closely follows whilst the TensorMask generally ranks last apart from some rare cases such as the category *stop sign* and *parking meter* in which TensorMask gave slightly higher AP scores. On the other hand, there were only three categories in which a higher AP was observed from CenterMask2 over Mask R-CNN: the *snowboard*, *toaster* and *hair drier*. This analysis is reflected in the overall performance of these architectures. A comparison of the performance of these three architectures for each category can be analysed in Figure 5.4.

A similar analysis was carried out on the Tiny Pascal VOC training dataset containing 20 different classes. Similar to what was concluded in the overall analysis of the architectures on this dataset, the Mask R-CNN gave the best performance for each class except for the *horse* and *airplane* category. Between the CenterMask2 and the TensorMask, it's not always the case that the CenterMask2 outperforms the TensorMask but it is the case for the majority of the categories. The conclusions from this analysis are similar to those from the in-depth evaluation of the MS-COCO17 dataset categories.

Another important factor considered for the evaluation of the instance segmentation architectures is the inference time. Intuitively, the inference image time for each architecture should not vary greatly regardless of the dataset since this is dependable on the trained model, however, an evaluation was carried out on each dataset nonetheless. Starting with the MS-COCO17 it could be identified that the longest inference time

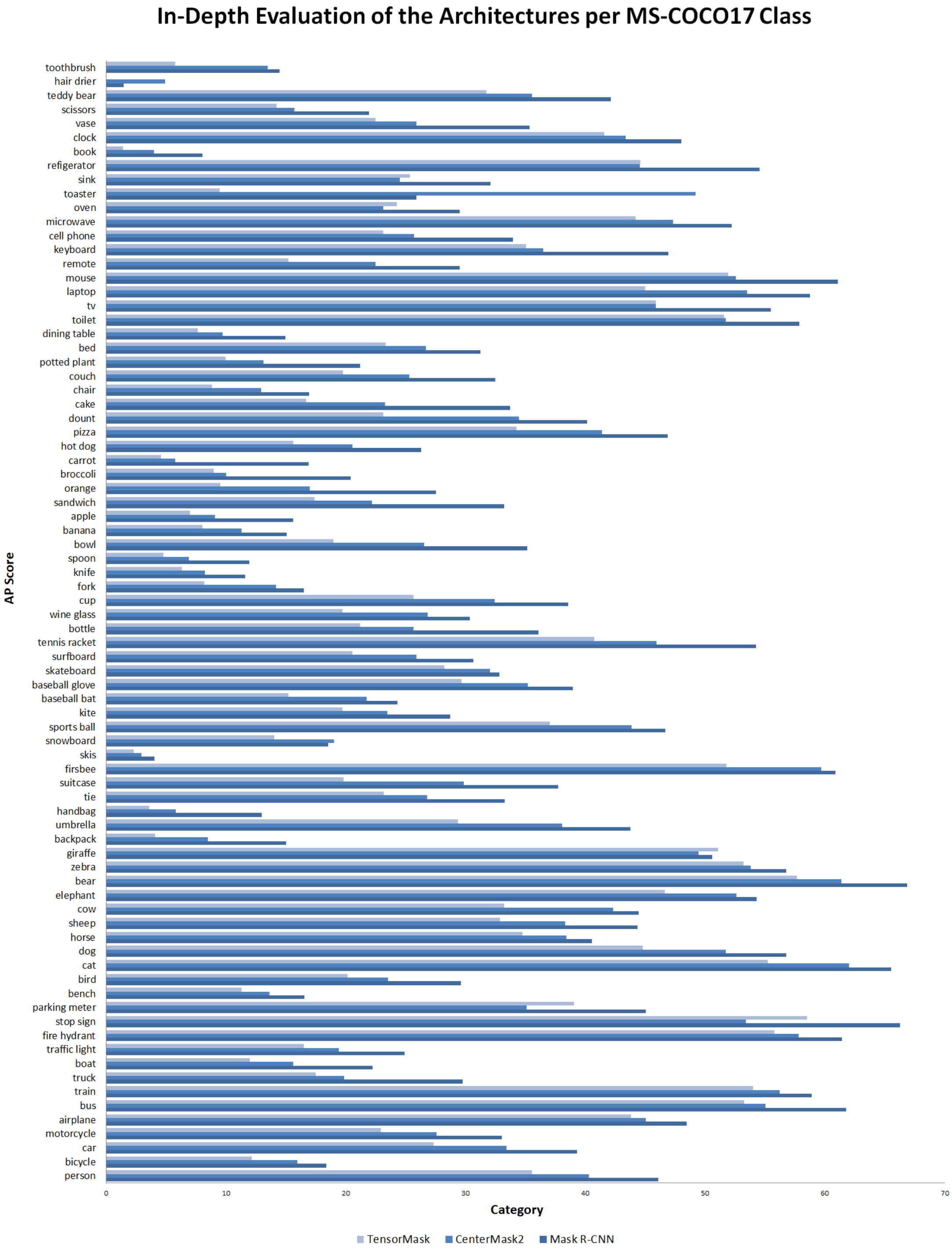


Figure 5.4: In-depth Evaluation of Instance Segmentation Architectures per class using the MS-COCO17 Validation dataset.

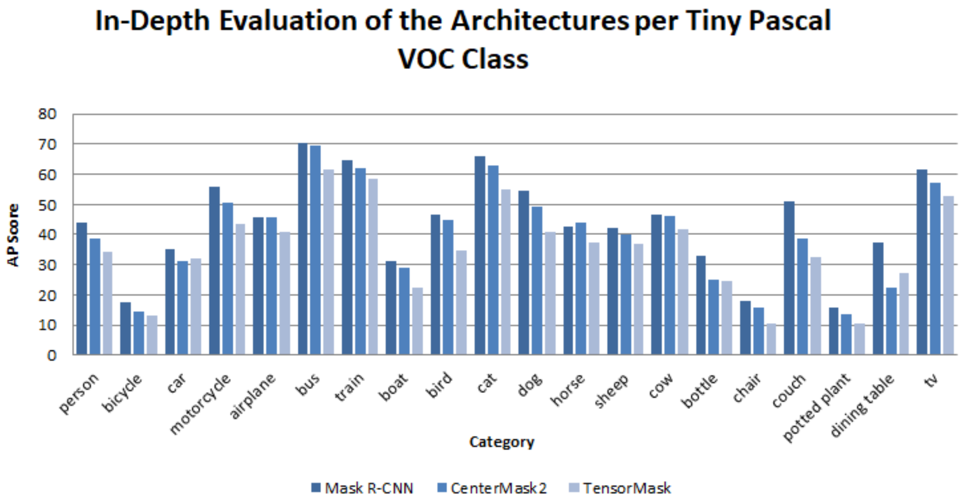


Figure 5.5: In-depth Evaluation of Instance Segmentation Architectures per class using the Pascal Tiny VOC Train dataset.

Table 5.2: Evaluation of Inference Time per image for each dataset for Instance Segmentation architectures distinguishing the best performing models in this regard by their low time identified by the underlining.

	MS-COCO17 (s)	Tiny Pascal VOC (s)	Average (s)
Mask R-CNN	0.474447	0.464071	0.469259
Yolact	0.485437	0.2114165	0.3484267
Yolact++	0.487805	0.4132231	0.450514
CenterMask2	0.653036	0.672848	0.662942
TensorMask	0.999698	0.995292	0.997495

was that of the TensorMask whilst the shortest belonged to the Mask R-CNN. The Yolact and Yolact++ gave very similar inference times that were very competitive with the Mask R-CNN. This is plausible given the real-time inference properties of the Yolact architecture (Bolya et al., 2019). A different pattern can however be analysed for the Tiny Pascal VOC training dataset with the Yolact this time giving a significant low inference time. The Yolact++ and the Mask R-CNN then follow with a similar inference time ranking second and third respectively. Similar to what was achieved on the MS-COCO17, the TensorMask and the CenterMask2 outputted with the longest inference ranking last. An average inference time was then calculated to easily compare the overall inference times by giving each performance on both datasets an equal weighting. From this analysis, the Yolact provided the fastest inference time followed by the Yolact++. As stated before, the researchers Bolya et al. (2019) focused on real-time inference and from this analysis it could be concluded that they managed to do just that with their architecture ranking first in terms of effectiveness. The Mask R-CNN achieved an overall score that was similar to the Yolact++ whilst the CenterMask2 and TensorMask performed the poorest in this regard.

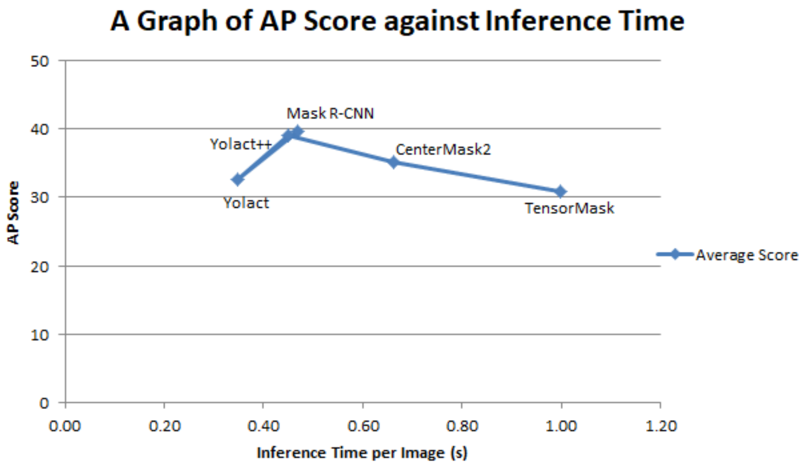


Figure 5.6: A graph of the Average AP score against the Average Inference time. The best overall performing model would be located as close as possible to the y-axis and as high on the y-axis as possible as this would signify a low inference time and a high average AP Score respectively.

To determine the best overall performing model a graph of the average AP score against the average inference time was plotted as shown in Figure 5.6. From this graph, it could be concluded that the TensorMask was the worst-performing model giving the highest inference time as well as the lowest accuracy. Chen et al. (2019) with this architecture aimed to provide other researchers with a baseline model for the development of instance segmentation techniques using dense-sliding windows. Therefore, this result does not come as a surprise. On the other hand, the Yolact architecture provided the lowest inference time and a lower than others AP score. The main aim of this research by Bolya et al. (2019) was to develop an architecture that is able to perform real-time inference. From this analysis, it could be concluded that this was achieved at the expanse of the AP score. With the research of the Yolact++, the researchers aimed to increase the performance of the AP Score of the Yolact while retaining the inference time. From this analysis, it could be concluded that the AP score improved drastically and managed to achieve a similar score to that of the Mask R-CNN although slightly lower. In terms of inference, the Yolact++ gave a slightly faster inference time to Mask R-CNN however the inference time increased drastically from its precedent. CenterMask2 although it performed better than the TensorMask did not manage to compete with the Mask R-CNN and the Yolact++. It can be analysed however that it gave a better AP score than the Yolact. The Mask R-CNN, the current state-of-the-art gave a very similar performance to the Yolact++. In fact, it gave a slightly higher score and a slightly higher inference time.

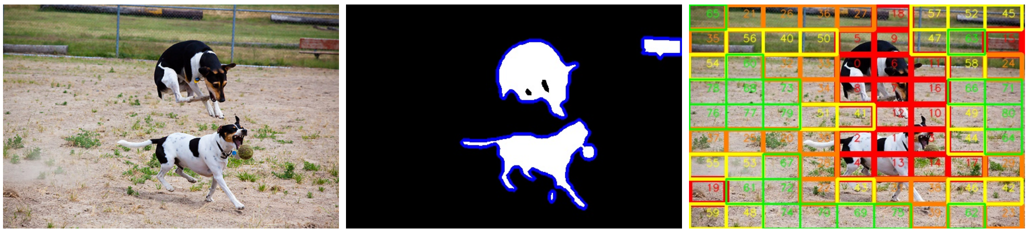
To conclude, both the Mask R-CNN as well as the Yolact++ could be considered the current state-of-the-art. This implies that other architectures are becoming more competitive with the Mask R-CNN. From this research, it could be concluded that the state-of-the-art varies depending on the use case. If a shorter inference time is more important than accuracy then the Yolact++ should be considered as the state-of-the-art whilst if the accuracy of the architecture is more important then the Mask R-CNN can still be considered as the current state-of-the-art.

5.2.2 Saliency Prediction Review

Evaluating saliency prediction for this application is a rather challenging task since the performance of the algorithms ought to be measured in relevance to their ability to identify the level of the saliency of the masks generated by the instance segmentation algorithm. In addition, the images generated from this process require to be clear and identifiable for the image transformer to perform feature extraction. The algorithms being considered for this research are the EML-Net (Jia and Bruce, 2020), DeepGaze II (Kümmerer et al., 2016), the Pyramid Feature Network (Zhao and Wu, 2019) and Itti's algorithm (Itti et al., 1998) for saliency prediction whilst the instance segmentation algorithm being used is the Mask R-CNN (He et al., 2017).

The evaluation process consists of running a hundred random samples through an identical procedure to parallelly generate binary masks from the instance segmentation algorithm and saliency maps from one of the saliency prediction algorithms. Following the procedure discussed for the encoder, element-wise multiplication is being performed on the binary masks and saliency map to generate the weighted mask image and the final weighted image. According to the methodologies applied, the performance of the saliency algorithms differs according to the complexity of the image, therefore images of varying complexities will be analysed separately for each algorithm to gather an understanding of the behaviour of the saliency predictors for this application. In addition, further analysis will be conducted on the number of contours calculated on the binary mask, saliency map and weighted mask images as well as on the percentage of non-black pixels.

The first image considered as shown in Figure 5.7 consists of an image with two main distinguishable objects of the class dog in addition to an object of type ball in possession

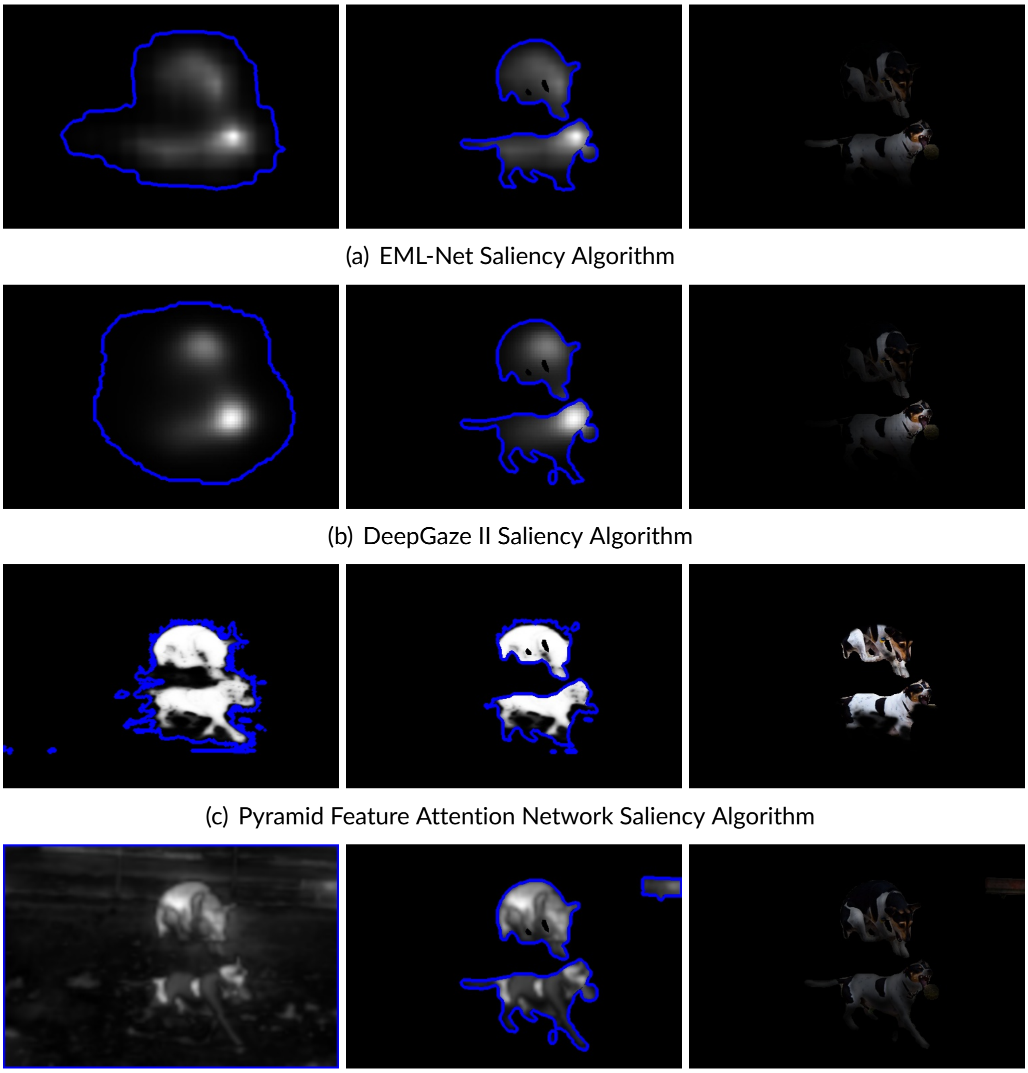


(a) Sample Image (b) Binary Mask - Mask R-CNN (c) Saliency Ranking - Sara

Figure 5.7: A Sample Image consisting of a Non-complex background with its respective Binary Mask and Saliency Ranking Result using Mask R-CNN and Sara respectively

of one of the dogs. A further object of the class bench can be identified on the far right. As can be analysed the Mask R-CNN algorithm generated four masks representing all the objects in the image. This is also reflected by the saliency ranking algorithm Sara (Seychell and Debono, 2018) which identified the main three objects at the centre as being the main focus of the image with ranks decreasing as they get progressively away from the focal point. Moreover, it also identified the fourth object at the side of the image as salient providing it with a significant score.

The saliency map of the image is then being calculated using each of the considered saliency algorithms as shown in Figure 5.8 with each row representing an algorithm. As can be examined, the EML-Net algorithm highlighted that the most salient part was the face of the lower dog. The contour drawn on the image, showing which segments contained a pixel value greater than 0, showed that the central area of the image was the most salient therefore providing a score to three out of the four objects in the image, discarding the fourth. This was also reflected in the weighted mask and the final weighted object image. Similarly, the DeepGaze2 algorithm provided a score to three out of the four objects however the salient map generated covered a larger area managing to retain the full shape of the lower dog. A similar result was achieved by the Pyramid Feature Attention Network however rather than highlighting a particular point within the objects it classified the entire objects as salient. Consequently, the final weighted image consisted of distinguishable objects rather than segments of the objects which would be essential during feature extraction. The final algorithm considered is Itti's saliency algorithm. This algorithm produces a saliency score for every pixel in the image therefore, all the masks generated by the instance segmentation algorithm would be classified with a level of saliency ensuring no loss of data. In fact, in the final image produced it could be identified that all four objects recognised by the Mask R-CNN are in the image with varying levels of attention.



(a) EML-Net Saliency Algorithm

(b) DeepGaze II Saliency Algorithm

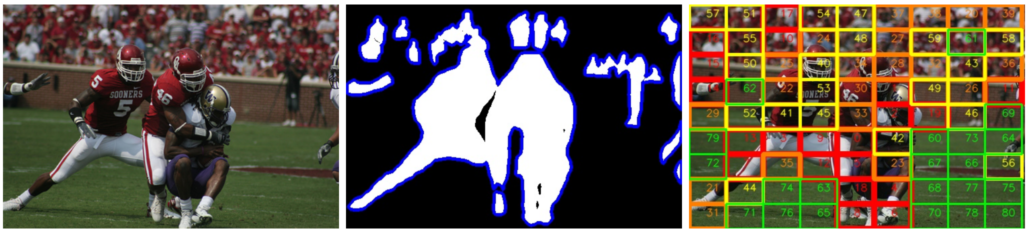
(c) Pyramid Feature Attention Network Saliency Algorithm

(d) Non-deep Itti Saliency Algorithm

Figure 5.8: A Sample Image consisting of a Non-complex background with its Saliency Map (left), Weighted Mask (middle) and Weighted Image (right) for each considered Saliency Prediction Algorithm

The second image considered as shown in Figure 5.9 contains a more complex background as well as a good number of present objects. In fact, the instance segmentation algorithm identified 23 objects with the majority located in the background. The saliency ranking algorithm highlighted that the centre of the image is the main focus however the top side of the background, the area with the stalls, is also salient.

Similar to the previous analysis, the saliency map, the weighted mask and the weighted image generated were computed for each saliency algorithm as shown in Figure 5.10.

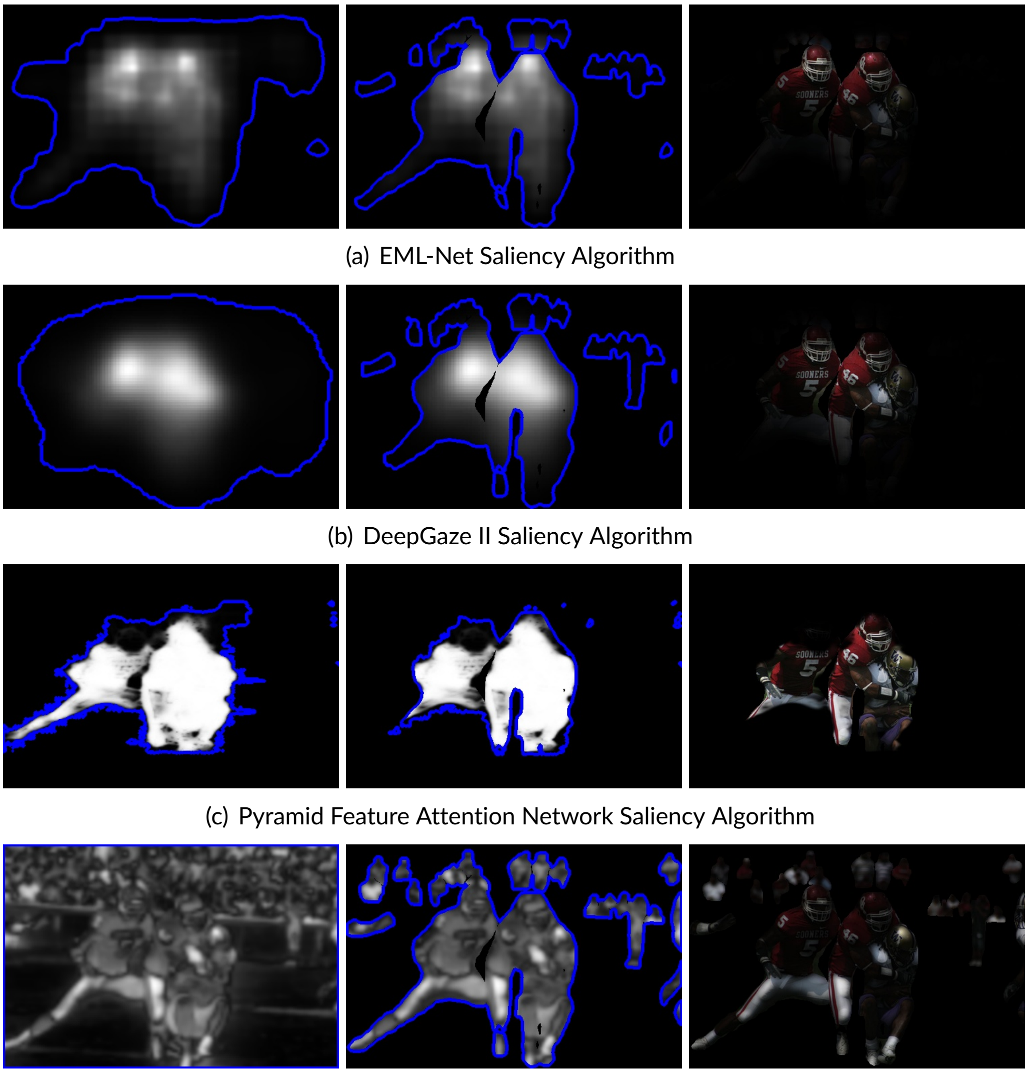


(a) Sample Image (b) Binary Mask - Mask R-CNN (c) Saliency Ranking - Sara

Figure 5.9: A Sample Image consisting of a Complex background with its respective Binary Mask and Saliency Ranking using Mask R-CNN and Sara respectively

Starting with the EML-Net algorithm, it could be analysed that this algorithm identified that the most salient regions of the image are the faces of the players. However, the contours drawn highlight that a saliency score was given to a large portion of the image. In fact, as shown from the weighted mask some masks in the background were retained even though the saliency is low and shown as faint in the final image generated. A similar conclusion can be deciphered for DeepGaze II however a bigger area was identified as salient therefore as shown in the weighted mask, more objects from the background were identified. These masks are rated at an even lower saliency score and therefore appear even fainter in the final image generated. Differently, the Pyramid Feature Attention Network algorithm only identified the players as salient and didn't identify any objects from the background. In addition, the entire objects were identified as salient and not only parts of it as deduced in the previous algorithms. Since the Itti saliency algorithm provides a salient score for every pixel in the image, all the masks generated by the instance segmentation algorithm were provided with a score that reflects their attention as shown in the final generated image.

As could be concluded from these two separate instances of varying complexities, a saliency score is not always provided to all the masks with the exception of Itti's saliency algorithm since a score for every pixel within the image is provided. This suggests a potential loss of information that might have been otherwise important to form part of the feature extraction. To further evaluate this concept, the percentage of non-black pixels within all the sample images was explored for all the saliency algorithms comparing the instance segmentation binary map, the saliency prediction map and the weighted mask. An equal score between the instance segmentation map and the weighted mask signifies that there has been no loss of data since all the masks in the binary map would have been provided with a salient score. From this analysis, as shown in Table 5.3, it could be concluded that as expected Itti's algorithm experienced no loss of mask data whilst for



(a) EML-Net Saliency Algorithm

(b) DeepGaze II Saliency Algorithm

(c) Pyramid Feature Attention Network Saliency Algorithm

(d) Non-deep Itti Saliency Algorithm

Figure 5.10: A Sample Image consisting og a Complex background with its Saliency Map (left), Weighted Mask (middle) and Weighted Image (right) for each considered Saliency Prediction Algorithm

the other algorithms the loss was of less than 10% with the highest being 9.05% from the Pyramid Feature Attention Network algorithm and the least being 5.46% from the DeepGaze II.

Delving furtherly into the number of contours generated as shown in Table 5.4, it could be deduced that this is not a reliable metric to determine loss of data. This is due to transforming the image into a binary image to compute the contour. As a threshold for this transformation, any non-black pixel is being set to white while black pixels are retaining

Table 5.3: Comparing the percentage of non-black pixels within the binary mask generated by an Instance Segmentation (IS) algorithm, the Saliency Predictor map (SP) and the Weighted Mask (WM). The best saliency prediction algorithm for this application would be distinguished by a high SP, an IS equivalent to the WM and a low difference as identified by the underlining.

	IS (%)	SP (%)	WM (%)	Difference (IS - WM) (%)
EML-Net	28.08	41.40	20.71	7.37
DeepGaze II	28.08	56.99	22.63	5.46
Pyramid Feature Attention Network	28.08	28.17	19.03	9.05
Itti's Saliency	**28.08**	**100.0**	**28.08**	**0.0**

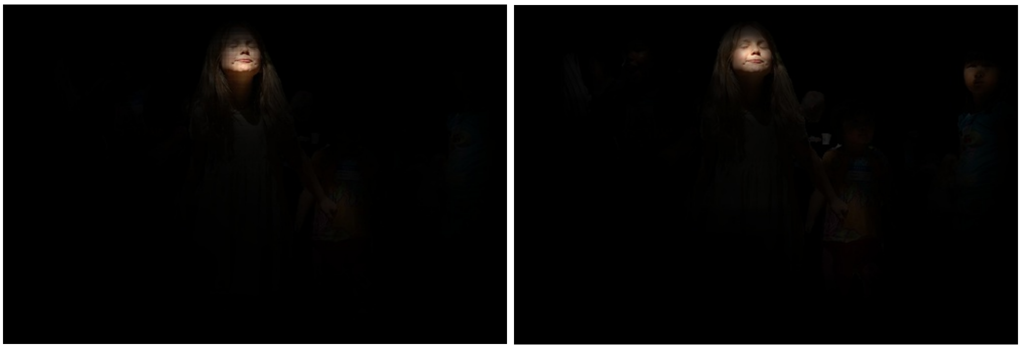
their colour. Using such a technique is reasonable to determine the salient points within an image however some algorithms such as the Pyramid Feature Attention Network give a vast number of random pixels a small score during the process of identifying the salient region generating insignificant undetectable contours contributing to the count of contours. As a repercussion, this will also contribute to a bigger count in the weighted mask image and the final image. Moreover, converting the weighted mask back to a binary image and re-generating contours does not guarantee that the same number of contours are generated since minor discrepancies in the image can cause contours to be adjoined together reducing the count. Therefore, the percentage of non-black pixels is a better metric to measure the loss of data.

As a closing argument, it could be concluded that the performance of the saliency prediction for this application depends on the required balance between the mask relevance and loss of data. Some deep saliency algorithms as analysed are sometimes prone to loss of mask data generated by the instance segmentation algorithm. In some circumstances, however, it could be identified that a saliency score is being provided but the score is extremely low hence implying that the object's relevance is insignificant and therefore this loss of data can be justified. In some situations, however, it was identified when consulting with the saliency ranking algorithm that a rank was provided, highlighting the object's importance in the image. Furthermore, some saliency algorithms such as DeepGaze II and EML-Net have sought to highlight the most salient part within the objects. Therefore in some instances such as for a person, the face is generally given a high saliency score

Table 5.4: Comparing the Number of Contours within the Binary Mask generated by an Instance Segmentation (IS) algorithm, the Saliency Predictor (SP) map and the Weighted Mask (WM). The ideal saliency prediction algorithm would contain an IS equivalent to the SP and WM as highlighted by the underlining. As discussed, this was not ideal metric for distinguishing the best saliency prediction for this application.

	IS	SP	WM	Difference (IS - WM)
EML-Net	2.98	1.3	2.43	0.55
DeepGaze II	**2.98**	**2.51**	**2.65**	**0.33**
Pyramid Feature Attention Network	2.98	77.32	6.7	-3.72
Itti's Saliency	2.98	1.0	2.49	0.49

while the rest of the body is considered insignificant such as in Figure 5.11. This although showcasing the outstanding performance and the improvement towards refinement of these algorithms is not ideal for this application since the image transformer will not be able to extract valid and reliable features.


(a) DeepGaze II

(b) EML-Net

Figure 5.11: Investigating Loss of Data caused by the Saliency Prediction Algorithms due to their Dedicated Processes of selecting the most salient regions within the objects of the image

As a result for this application, the non-deep Itti's visual saliency algorithm is going to be utilised since this algorithm provides a saliency score for every pixel, ensuring that there is no loss of data, unlike the Pyramid Feature Attention Network. Furthermore, given its simplicity unlike DeepGaze II and EML-Net, this algorithm does not provide a high saliency score to only a segment of the object but the entire object making it the ideal candidate for this application.

5.2.3 Scene Classification Review

Zhou et al. (2017) trained four different architectures: the ResNet-18 (He et al., 2016), ResNet-50 (He et al., 2016), DenseNet-161 (Huang et al., 2017) and AlexNet (Krizhevsky et al., 2012) on the Places-365 Standard dataset (Zhou et al., 2017) to perform scene classification. For this part of the evaluation, the validation set of this dataset was utilised to analyse the performance of these models by checking whether the predicted labels match the ground truth. The top-1 and top-5 accuracies were used with the latter being introduced due to ambiguity in scene classification (Zhou et al., 2017). Furthermore, the inference time was discussed since this also plays an important role in identifying the best holistic performing model.

Table 5.5: Evaluation of the pre-trained Scene Classification models on the Validation Set of the Places-365 Standard highlighting the best performing models that achieved the highest accuracy and the lowest inference time by underlining.

	Top-1 Accuracy	Top-5 Accuracy	Inference Time per image (s)
Alexnet	47.55%	77.98%	**0.00275**
ResNet-18	53.69%	83.78%	0.00300
Renset-50	54.77%	84.93%	0.00731
DenseNet-161	**56.13%**	**86.12%**	0.01963

The accuracies achieved on the validation set as can be analysed in Table 5.5 were rather similar to each other with the best performing model being the DenseNet-161 both when considering the top-1 and top-5 accuracy with a score of 56.13% and 86.12% respectively. The lowest-performing model was then the AlexNet with a top-1 accuracy of 47.55% and a top-5 accuracy of 77.98%. It could be noted that the difference between the best and worst-performing models was that of around 8.5% which is rather minimal. Furthermore, it could also be identified that each model imitates the same accuracy performance over both the top-1 accuracy and the top-5 accuracy as shown in Figure 5.12. Therefore the ordering of the best performing models remains the same for both the top-1 accuracy and the top-5 accuracy.

Differently, when considering the inference time the best performing model was the AlexNet in which the lowest inference time was observed whilst the DenseNet-161 outputted with the highest inference time making it the worst-performing model in this regard. From this analysis, it could also be observed that the higher the accuracy of the model, the higher the inference time as being conveyed in Figure 5.12. This could be due to the depth of the models. The deeper the model, the higher the accuracy and the higher the inference time. In fact, the deepest model considered in this evaluation is the DenseNet-161 whilst the AlexNet is the shallowest model.

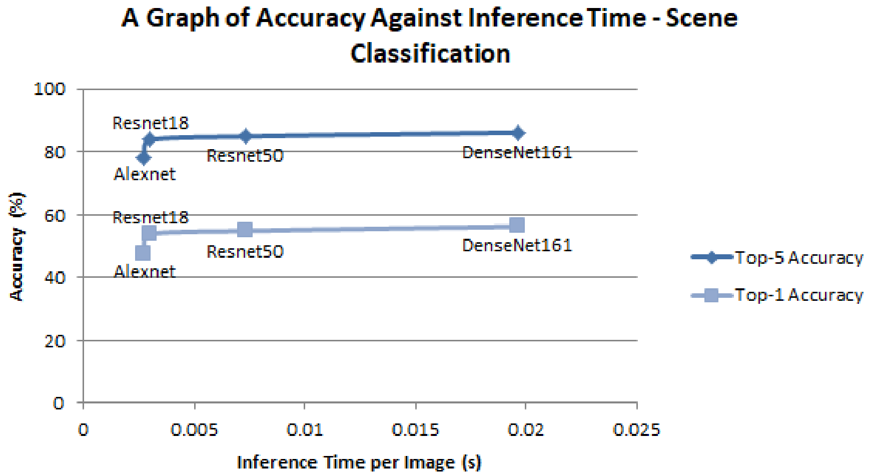


Figure 5.12: Evaluation of the Scene Classification pre-trained architectures by plotting a graph of the accuracy against the inference time. The best overall performing model would be located as close as possible to the y-axis and as high on the y-axis as possible as this would signify a low inference time and a high accuracy respectively.

Overall, it could be concluded that the ResNet-50 is the best performing model for scene classification. This is because although the DensetNet-161 gave the highest accuracy, it did this at the expanse of inference time in addition to being computationally expensive in memory. The ResNet-50 gave slightly lower accuracies than the DenseNet-161 ranking second as well as was competitive with the other architectures in terms of the inference time.

5.3 Image Captioning Model

During the construction of the image captioning model, various experiments took place to distinguish the most optimal model. The most notable experimentation revolved around the text encodings used as input for the decoder transformer and the hyper-parameters of the model. In this section, firstly these related experiments will be explored and secondly, the performance of the final optimal model will be discussed in relation to current research.

5.3.1 Text Encodings

The decoder as analysed in the methodology accepts the target sentences in their encoding representation. For this phase, four different techniques were evaluated on the Flickr8K dataset since this is the smallest available dataset making it ideal for evaluating model variations given the limitations imposed due to the lack of computational power. The techniques considered consisted of Index-Based Encoding, One-Hot Encoding, Glove Encoding and Bert Encoding.

Index-based Encoding is the most basic of the encodings considered, converting each word into a unique number identifier. Differently, one-hot-encoding is more complex in that it transforms each word into a unique vector consisting of only binary digits. Due to the size of the corpus and memory limitations, upon using such a technique the data loading processing in the input pipeline pre-longed drastically and it was opted to eliminate this encoding from the analysis. The remaining two encodings consisted of the GloVe (Pennington et al., 2014) embedding which also represents the relationships between words and the Bert (Devlin et al., 2018) embeddings which manages to produce embeddings for each word based also on the context.

To experiment with different embeddings, an architecture was trained on a 10,000 word vocabulary. Upon investigation, it was identified that the Flickr8K dataset does not contain 10,000 unique words, therefore, all the words retrieved which amount to 8,674 were used. The caption length to which sentences were reduced or padded to was set to 20 words. Following the original paper, the number of layers for the encoder and the decoder were initialised to 6 each with the d_model set to 512 and a dff of 2048. Unlike that presented in the original transformer paper, the dropout was set to 0.5 instead of 0.1 since from experimentation overfitting was observed and increasing the dropout as a regularisation technique increased the overall performance of the image captioning model.

As shown in Table 5.6, the evaluation metrics discussed in the methodology: Bleu, Meteor, Rouge-L and CIDEr were used to measure the performance of the image captioning models. The figures generated for this analysis were retrieved using a beam search algorithm with a width of 3. From this experimentation, it could be concluded that an index-based encoding generated the highest performance out of all the encodings considered. It could also be identified that the performance of the models could be described as rather comparable since the difference between the best and worst performing was constantly less than 3. The lack of noticeable improvement when using a more advanced

Table 5.6: Comparing the performance of the image captioning architecture on different embeddings highlighting the best performing model that generated the highest scores on each metric by underlining and concluding that the Index-Based Encoding showed a slight increase in performance.

	Bleu-1	Bleu-2	Bleu-3	Bleu-4	Meteor	Rouge-L	CIDEr
Index-Based	**63.4**	**45.6**	**31.3**	**21.5**	**19.9**	**46.3**	**49.0**
GloVe	61.9	43.1	28.7	18.7	19.4	45.3	46.3
Bert	61.0	42.9	28.8	19.2	**19.9**	45.1	46.7

encoding such as GloVe or Bert could be due to the transformer learning its own embeddings in addition to calculating the respective positional encoding. A simple encoding might provide the model with a cleaner and more understandable representation allowing the image captioning model to generalise and perform the necessary mappings from the target sentences more efficiently.

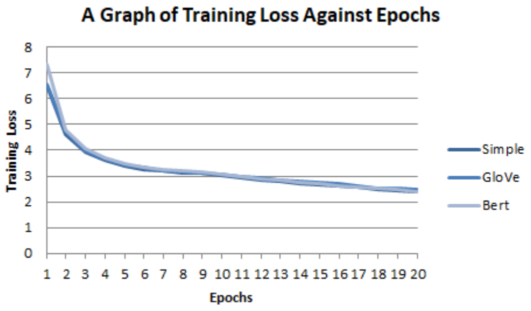


Figure 5.13: A Graph of Training loss against Epochs for each Encoding

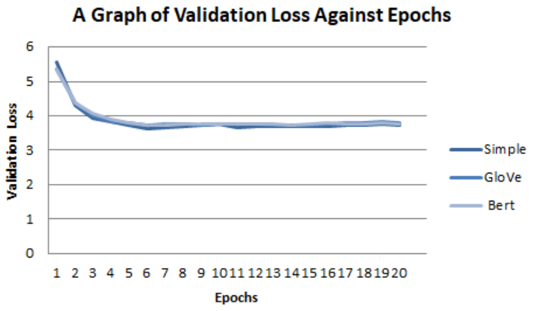


Figure 5.14: A Graph of Validation loss against Epochs for each Encoding

The effect of the encodings on the training of the image captioning model was also explored. From this analysis as distinguished in Figures 5.13 and 5.14, it could be determined that there was no effect on either the training or the validation loss. Bert encoding started at a slightly higher training loss when compared to the other two however it could be identified that the training loss decreased at a faster rate concluding with almost identical figures. For the validation loss, the index-based encoding started at a very slightly higher validation loss that was even quicker to diminish to replicate the loss of the other encodings. From these two graphs, it could also be concluded the training loss is lower than the validation loss. In fact, the training loss was getting progressively lower whilst the validation loss became rather stable. This could imply that the model is overfitting. To overcome this the dropout rate was already increased to 0.5 from 0.1 but as can be analysed slight overfitting is still being experienced. Increasing the dataset and utilising the Flickr30K might resolve the overfitting being experienced on this smaller dataset.

To conclude, the encoding opted for this research, based on this analysis was the index-based encoding provided that this is the most simple encoding in addition to having performed slightly better than the other encodings considered.

5.3.2 Model Hyper-Parameters

The hyper-parameters presented by Vaswani et al. (2017) for the transformer consisted of a 6 layers encoder and decoder with a dropout rate of 0.1, a d_model of 512 and a dff of 2048. This section presents experiments performed varying these hyper-parameters. As a standard, the sentences are being limited to or padded to 20 words with a 10,000 word dictionary. As distinguished before, the Flickr8K does not contain 10,000 unique words therefore for that dataset all the unique words which tally to 8,674 words are being used.

The first notable short-coming during the training of the models was overfitting with the training loss being significantly lower than the validation loss. To overcome this challenge, experiments with the dropout rate were performed on the Flickr8K and the Flickr30K datasets and evaluated using the above-mentioned metrics: Bleu, Meteor, Rouge-L and CIDEr using the beam search heuristic with a beam width of 3. Commencing with the Flickr8K dataset, the dropout rate was first increased to 0.3 and then to 0.5. The results generated as shown in Table 5.7 concluded that the best performing model consisted of a dropout rate of 0.5. However, it could be analysed that the improvement was rather minimal with a change of less than 3 across all metrics. Furthermore, it could also be deduced that the results are not consistent and therefore non-conclusive. This is because from the numbers gathered the worst performing model was that with a dropout of 0.3 and not with a dropout of 0.1 concluding that increasing the dropout rate does not necessarily have a direct impact on improving the model's performance.

Statistics gathered on the loss during the training of the models showed that the difference between the training and the validation loss decreased when the dropout rate increased. The relation between the training and the validation loss as shown in Figures 5.15 and 5.16 for a dropout rate of 0.1 and 0.5 respectively exhibit that although the training and validation loss follow the same formation, the spacing between the losses is smaller when using a dropout of 0.5. It is also interesting to observe that when utilising a dropout of 0.5, the training and validation loss are slightly higher. These continue to decrease with a lower dropout rate. Therefore, with a dropout of 0.3, the training and validation loss is lower and with a dropout of 0.1, the loss is at its lowest.

Table 5.7: Comparing the performance of the image captioning architecture with different dropout rates on the Flickr8K Dataset highlighting the best performing model that generated the highest scores on each metric by underlining and concluding that a dropout of 0.5 showed an increase in performance.

	Bleu-1	Bleu-2	Bleu-3	Bleu-4	Meteor	Rouge-L	CIDEr
0.1	62.1	44.0	30.0	20.2	**19.9**	46.1	48.9
0.3	60.9	42.7	28.7	19.1	19.4	44.3	47.0
0.5	**63.4**	**45.6**	**31.3**	**21.5**	**19.9**	**46.3**	**49.0**

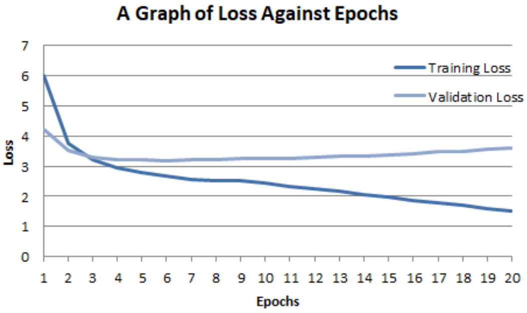


Figure 5.15: Loss against Epochs for the Flickr8K with a 0.1 Dropout

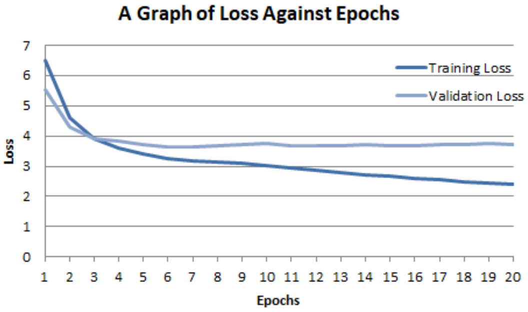


Figure 5.16: Loss against Epochs for the Flickr8K with a 0.5 Dropout

Apart from regularisation, increasing the dataset can potentially mitigate overfitting. Therefore a similar analysis was performed on the Flickr30K dataset which consists of 31,783 images. As analysed from Table 5.8, the image captioning model gave an overall higher performance when utilising a dropout rate of 0.1. The difference between the performance was once again minimal with a maximum difference of 1.2 for the CIDEr metric. In addition, for metrics such as Bleu-2, Bleu-3 and Bleu-4 a higher performance was observed when utilising a dropout rate of 0.5. This differs from the conclusions drawn on the Flickr8K dataset performance review and could therefore imply that by increasing the size of the dataset, overfitting was addressed without the need of adjusting the regularisation. Targeting overfitting is important for the model to have the ability to generalise over unseen data. However, a balance must be established to ensure that the model is not underfitting. Having a validation loss that is equal to the training loss or lower than the training loss might signify underfitting.

To analyse the impact of a larger dataset on the training, visualisations showcasing the training and validation loss throughout the duration of the training are being plotted

Table 5.8: Comparing the performance of the image captioning architecture with different dropout rates on the Flickr30K Dataset highlighting the best performing model that generated the highest scores on each metric by underlining and concluding that the Bleu metrics favoured a dropout rate of 0.5 whilst other metrics a dropout of 0.1.

	Bleu-1	Bleu-2	Bleu-3	Bleu-4	Meteor	Rouge-L	CIDEr
0.1	**63.1**	44.1	30.3	20.8	**18.6**	**44.1**	**43.1**
0.5	62.6	**44.3**	**30.6**	**20.9**	18.3	43.8	41.9

in Figures 5.17 and 5.18 showing a dropout rate of 0.1 and 0.5 respectively. From these figures, it could be identified that the discrepancies between the training and validation loss are not as wide as previously seen on the Flickr8K dataset. In fact, with a dropout rate of 0.1, the model is slightly overfitting with a disparity between the losses that is slightly less than what was analysed on the Flickr8K with a dropout of 0.5. Looking at the graph representing a dropout of 0.5, it could be concluded that the difference between the training and validation loss is even less. In general, it could also be observed that losses: in both validation and training started lower than those identified on the Flickr8K.

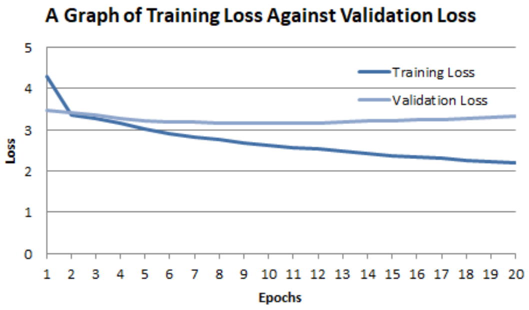


Figure 5.17: Loss against Epochs for the Flickr30K with a 0.1 Dropout

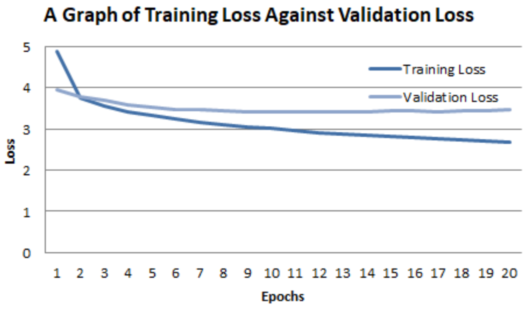


Figure 5.18: Loss against Epochs for the Flickr30K with a 0.5 Dropout

Overall from the statistics gathered it was concluded that for the Flickr8K the optimal dropout rate to aid in overfitting and to generate the best performing model was that of 0.5. Differently, for the Flickr30K dataset, it was opted to utilise a dropout rate of 0.1. This is because from the visualisation only slight overfitting was observed in addition to generating the highest performance over the metrics considered showing that the model managed to generalise over unseen data. Moreover, using a dropout rate of 0.1 is in-line with the original architecture of Vaswani et al. (2017). Furthermore, from this analysis, it was also observed that increasing the dataset helps combat overfitting.

The second set of experiments conducted revolved around the depth of the architecture. The default number of layers for the encoder and decoder set for the transformer architecture is 6. This part of the evaluation targets the depth by experimenting with a 4 and 8 layer transformer. From the experimentation carried out on the Flickr8K dataset, the architecture performed the best with 6 layers and the worst with 8 layers. In fact, it could be observed from Table 5.9 that with 8 layers the model did not train. With 4 layers the performance was almost identical to that with 6 layers with a maximum difference of 1.6. Reducing the number of layers in the architecture allows for faster training given that the model's depth is reduced. However, given the results and considering that the architecture will be used to train a model with a bigger dataset, it was decided to utilise a 6 layer encoder and decoder structure.

Table 5.9: Comparing the performance of the image captioning architecture with varying layers on the Flickr8K Dataset highlighting the best performing model that generated the highest scores on each metric by underlining and concluding that an architecture with 6 layers provided the best performance.

layers	Bleu-1	Bleu-2	Bleu-3	Bleu-4	Meteor	Rouge-L	CIDEr
4	62.7	44.3	30.1	19.9	19.7	45.7	47.7
6	**63.4**	**45.6**	**31.3**	**21.5**	**19.9**	**46.3**	**49.0**
8	11.0	0.0	0.0	0.0	3.9	14.9	0.1

A further evaluation was conducted by varying the dff and d_model variables. The first architecture shown in Table 5.10 represents the architecture presented in the transformer architecture, therefore, consisting of a dff set to 2048 and d_model set to 512. The second architecture considered is a simplified version with all the values from the original architecture divided by 4 and therefore are set to a dff of 512 and a d_model of 128. The third model experiments with different ratios and therefore divides the dff by 2 for a value of 1024 while retaining the d_model value of 512.

From this analysis, it was concluded that the best performing model followed the hyper-parameters set by the Vaswani et al. (2017) followed closely by the third architecture with almost identical metrics. This could be due to the fact that the d_model remained constant with alteration only to the dff which represents the dimension of the feed-forward network. Following the results achieved, it was opted to use the hyper-parameters as presented in the original transformer research.

In this section, an analysis of the hyper-parameters of the transformer architecture which composes the decoder structure of the image captioning model was presented.

Table 5.10: Comparing the performance of the image captioning architecture with varying d_model and dff on the Flickr8K Dataset highlighting the best performing model that generated the highest scores on each metric by underlining and concluding that an architecture with a configuration similar to that presented in the original transformer architecture showed an increase in performance over the other considered configurations.

dff	d_model	Bleu-1	Bleu-2	Bleu-3	Bleu-4	Meteor	Rouge-L	CIDEr
2048	512	**63.4**	**45.6**	**31.3**	**21.5**	19.9	**46.3**	**49**
512	128	60.1	41.4	27.7	18.3	19.1	44.2	44.1
1024	512	63.2	44.5	30.4	20.1	**20.0**	46.1	48.4

From the investigation conducted it was determined that the optimal hyper-parameters were identical to those proposed by the researchers of the transformer hence consisting of a 6 layer encoder and decoder with a dff of 2058 and a d_model of 512. The dropout rate for the Flickr8K was determined to be set to 0.5 to combat overfitting whilst for the Flickr30K, it was opted to utilise a dropout rate of 0.1 following the default parameters.

5.3.3 Results

Although the main aim of this research was not to surpass the current state-of-the-art image captioning architectures but rather to develop an explainable modular architecture that focuses on segmentation and saliency prediction as an attention mechanism, a comparison is being performed with other similar architectures in this field. The developed model trained using the architecture discussed in the methodology and enhanced through experimentation performed on the hyperparameters along with other related architectures that are also bench-marked on the Flickr-30K dataset can be analysed in Table 5.11. The developed model similar to the Mask Captioning Network (Lim and Chan, 2019) was trained using a vocabulary of 10,000 words and sentences containing up to 20 words with a batch size of 32. The evaluation technique being used follows the beam search heuristic with a beam width of 3.

From the figures generated on the before discussed metrics: Bleu, Meteor, Rouge-L and CIDEr it was identified that the implemented architecture generated comparable results to the current state-of-the-art architectures. In fact, in general, it could be identified that the architecture exceeded the performance of all the image captioning models considered except the Mask Captioning Network on all metrics except for the Bleu-1 metric. Compared with the recent architecture explored by Al-Malla et al. (2022) which similarly utilises an encoder-decoder structure that bases its attention mechanism on features

Table 5.11: Comparing the Performance of the Image Captioning Architecture on the Flickr30K dataset on the Bleu (B), Meteor (M), Rouge-L (R-L) and CIDEr (C) metrics highlighting the best performing architecture on each metric by the underlining and the colour red and concluding that our research exceeded the performance of current research on most metrics as well as achieved comparable results to the Mask Captioning Network with some metrics favouring this architecture whilst other favoured this research. Furthermore, as marked by the colour blue our architecture exceeded the calculated average metric score over the other architectures considered.

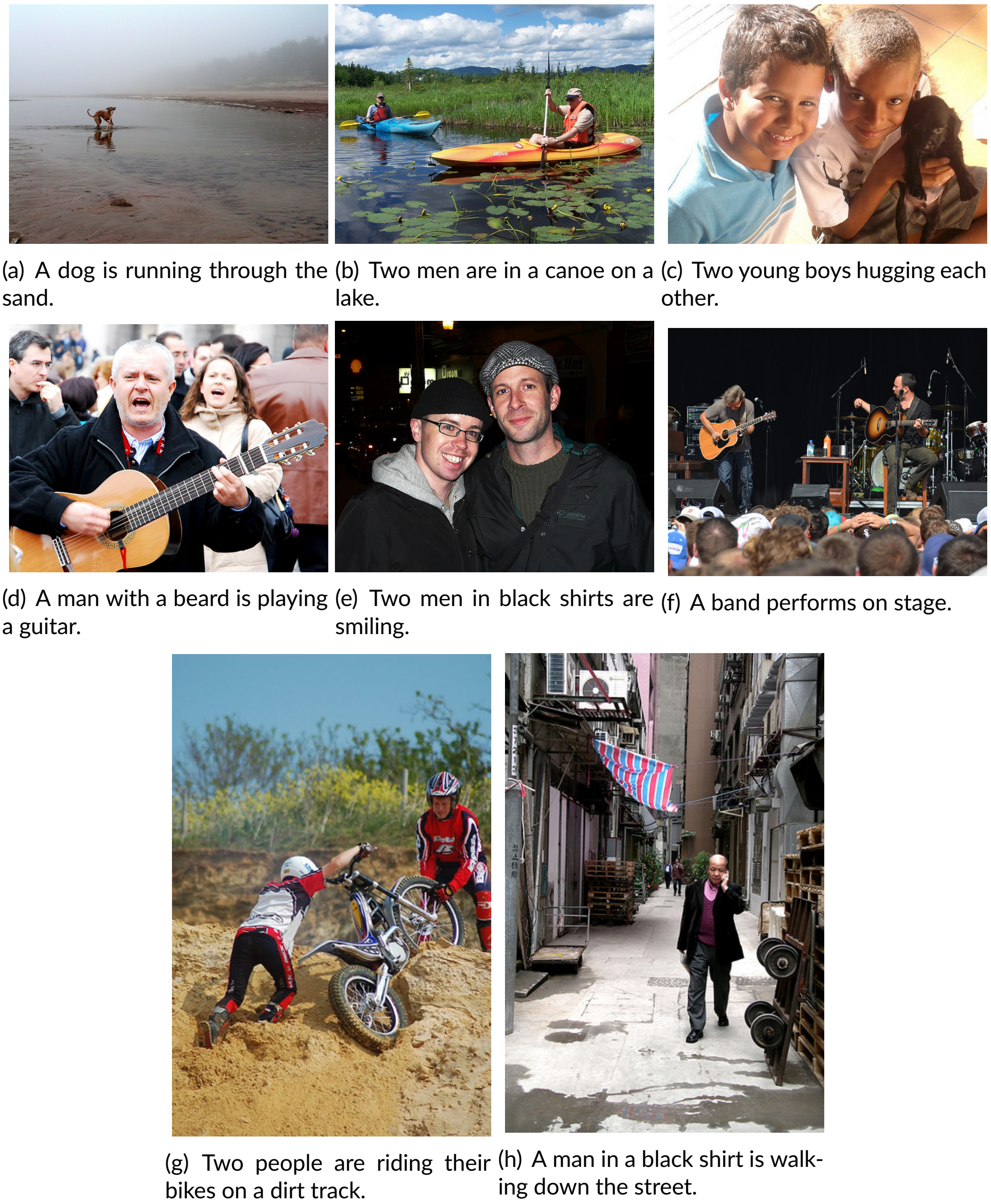
Architecture	B-1	B-2	B-3	B-4	M	R-L	C
Log Bilinear (Kiros et al., 2014)	60.0	38	25.4	17.1	16.88	-	-
Soft Attention (Xu et al., 2015)	66.7	43.4	28.8	19.1	18.49	-	-
Hard Attention (Xu et al., 2015)	66.9	43.9	29.6	19.9	18.46	-	
Google NIC (Vinyals et al., 2015)	66.3	42.3	27.7	18.3	-		
Mask Captioning Network (Lim and Chan, 2019)	64.7	46.2	32.5	22.7	18.5	45.0	43.4
Attention and Object Features Captioning Model Al-Malla et al. (2022)	39.8	22.1	11.6	6.1	12.9	29.8	15.0
Generated Average	60.7	39.3	25.9	17.2	17.0	37.4	29.2
Our Research	63.3	45.0	31.2	21.2	18.5	44.2	43.8

extracted from a pre-trained image classification model together with objects extracted from the object detection model YOLOv4, it could be concluded that the implemented architecture for this research surpassed their architecture on all the metrics considered by a large margin. Similarly, when comparing with the Log Bilinear (Kiros et al., 2014) architecture it could be identified that the implemented architecture gave an exceeding performance across all the different metrics but with a closer margin.

The Soft Attention and Hard Attention architectures (Xu et al., 2015) presented in the Show, Attend and Tell research gave a similar performance to the developed architecture. However, it could be identified that the performance of the presented architecture exceeded slightly the performance of these models except for the Bleu-1 metric to which a slightly lower score was observed. A similar conclusion can be drawn from the comparison with the Google NIC (Vinyals et al., 2015). From this observation, it could be distinguished that the Google NIC managed to transcend the presented architecture only on the Bleu-1 metric whilst it generated a lower score on the other evaluated metrics. Here, it could be analysed that the discrepancy was slightly higher than that observed on the soft and hard attention architectures (Xu et al., 2015).

A most interesting comparison is between the Mask Captioning Network and this research which has its roots in this architecture with a more explainable and modular structure. Observing these figures it could be analysed that these two architectures performed rather similar with a maximum difference of 1.5 between the metrics in favour of the Mask Captioning Network. It could be identified that the major discrepancies took place on the Bleu metric. This metric although widely used and generally referred to as the main language performance metric is also heavily criticised. This is because this metric does not measure the quality of the sentences generated but rather the string similarity by measuring the count of word overlap disregarding any linguistic elements in the process. Moreover, a further challenge of this metric is the vocabulary. If a generated sentence is composed of words that are synonyms or different to those presented in the ground truth, this metric due to its methodology performs weakly. Regarding the other metrics, it could be observed that the architectures generated an equal score of 18.5 for the Meteor score. Furthermore, it could be observed that the implemented architecture for this research generated a higher CIDEr score than the Mask Captioning Network. A final comparison is being conducted between the calculated average score for each metric considering all the architectures discussed and the generated score of this research. From this analysis, it could be identified that the proposed architecture exceeded the average score for each metric considered.

To evaluate the developed architecture on a deeper level, samples from the Karpathy test split being used to evaluate this research are being presented in Figure 5.19. The presented visualisations along with the model's generated captions show that the model manages to through the use of the scene layer also identify the environment of the image. For instance for Figures 5.19(a), 5.19(b), 5.19(g) and 5.19(h), the image captioning model managed to identify the scene and provide it as part of the image description. Another element is the subject's action identification. The trained model is identifying actions such as running, hugging, playing, smiling, walking, performing and riding in the correct grammatically form aiding to describe what is transcending in the image. A further consideration is the provided details for a richer description. For example for Figure 5.19(d), the captioning model identified a facial feature of the main subject of the image describing the man as having a beard. An additional example is Figures 5.19(e) and 5.19(h) which provided a colour adjective to the clothes of the people in the image. A different adjective



(a) A dog is running through the sand.

(b) Two men are in a canoe on a lake.

(c) Two young boys hugging each other.

(d) A man with a beard is playing a guitar.

(e) Two men in black shirts are smiling.

(f) A band performs on stage.

(g) Two people are riding their bikes on a dirt track.

(h) A man in a black shirt is walking down the street.

Figure 5.19: Images from the Karpathy Test split captioned using the Trained Model of the Architecture Presented

can be identified in Figure 5.19(c) to which the adjective young was used to describe the two boys in the image.

To sum up, this section has represented the conclusions drawn from experimentation carried out on the construction of the proposed image captioning architecture as well as presented the results achieved on language metrics. Moreover, the performance of the architecture was compared with other similar architecture deriving that the implemented architecture achieved comparable results with the Mask Captioning Network and exceeded the results of other considered architectures. Visualisations of several images inferred using the trained image captioning architecture were also discussed with consideration to the details in the descriptions and their grammatical correctness.

5.4 Sentence Generation and Readability

This section introduces the concept of readability to the image captioning architecture and aims to commence a discussion to target readability more creatively. Basing the notion of readability on the research of Kadayat and Eika (2020), it could be identified that the sentence length has a significant impact on the comprehensibility of individuals with a visual impairment with the ideal sentence length established to be between 16 and 20 words. In this section, the vocabulary size and the maximum sentence length used during the processing of the target sentences are being explored to identify whether the sentence goal length could be achieved by simply varying the configuration's parameters. The vocabulary size is being varied between 5,000 and 10,000 words whilst the target sentence length is being varied between 20 and 50 words on both the Flickr8K and the Flickr30K dataset.

5.4.1 Exploring the Sentence Length

Investigating the datasets, it could be identified that the Flickr8K contains 8,674 unique words that constitute its vocabulary with target captions containing up to 37 words. Differently, the larger dataset Flickr30K contains 19,710 distinct words and captions with up to 78 words. Further statistics gathered from the datasets as shown in Table 5.12 provide information regarding the minimum, maximum and average length of the target sentences. From these figures, it could be identified that the average length of the target sentences varied between the datasets with the Flickr8K dataset containing an average of 10.84 words and the Flickr30K dataset with an average of 12.32 words. Consulting with the work of Kadayat and Eika (2020), sentences with a length between 10 and 15 were the second category considered to be readable after the 16 to 20 words category.

In this section, it will be explored if by varying the vocabulary size and the target and evaluation sentence length, the generated average sentence length could be adjusted to fit the recommended 16 to 20 words category. Regarding Table 5.13 with results generated on the Flickr8K dataset, it could be identified that the average lengths of the gen-

Table 5.12: Statistics Gathered on the Target Captions of the Flickr8K and Flickr30K Datasets distinguishing that the Flickr30K contains longer target sentences and a larger vocabulary.

Dataset	Vocabulary	Min Caption Targeted Length	Max Caption Targeted Length	Avg Cap Targeted Length
Flickr8K	8,674	2	37	10.84
Flickr30K	19,710	2	78	12.32

erated captions were at their lowest when the models were trained or evaluated using a sentence length of 20. Contrarily, the longest average could be observed when training the model with the longest sentence length found in the dataset, that of 37 words and evaluated at 50 words. Although an average of 11.33 was observed this was still not sufficient to meet the requirements of 16 to 20 words established by Kadayat and Eika (2020) to promote readability. Moreover, from these statistics, it could also be identified that overall the average caption lengths generated are slightly lower than the average target length presented in the Flickr8K dataset.

Table 5.13: The Effect of the Vocabulary and the Maximum Sentence Length on the Generated Captions on the Flickr8K using a Batch Size of 64. Since the maximum target length of this dataset is 37, the maximum sentence length is not 50 as identified on the Flickr30K but 37. The highest average generated caption length was highlighted by the underlining.

Dataset	Vocabulary	Sentence Length	Sentences Evaluated at a Max Sentence Length	Min Cap Generated Length	Max Cap Generated Length	Avg Cap Generated Length
Flickr8K	5,000	20	20	4	20	9.732
Flickr8K	5,000	37	20	2	20	7.793
Flickr8K	5,000	37	50	2	50	**11.326**
Flickr8K	10,000	20	20	4	20	8.912
Flickr8K	10,000	37	20	5	20	9.615
Flickr8K	10,000	37	50	5	50	9.886

Exploring the above experimentation on the Flickr30K dataset as could be analysed in Table 5.14, it could be identified that similar deductions can be made. In fact, for this dataset, the models trained on 50 word sentences, generated a slightly higher average caption length. A further observation is that the average generated caption length is higher than that explored on the Flickr8K and similarly, the average caption length is slightly lower than the average target length deduced from the Flickr30K dataset. Compared with the work of Lim and Chan (2019) it could be observed that the average caption length generated by their architecture was that of 9.37. This is slightly lower than what was managed to be achieved by this research to which on the Flickr30K an average of 10.53 words per caption was observed when training on a 5,000 word vocabulary and trained and evaluated on a maximum of 50 words for each caption.

Table 5.14: The Effect of the Vocabulary and the Maximum Sentence Length on the Generated Captions on the Flickr30K using a Batch Size of 64. The highest average generated caption length was highlighted by the underlining.

Dataset	Vocabulary	Sentence Length	Sentences Evaluated at a Max Sentence Length	Min Cap Generated Length	Max Cap Generated Length	Avg Cap Generated Length
Flickr30K	5,000	20	20	3	20	9.719
Flickr30K	5,000	50	20	4	20	10.520
Flickr30K	5,000	50	50	4	50	**<u>10.526</u>**
Flickr30K	10,000	20	20	4	20	9.543
Flickr30K	10,000	50	20	3	20	10.125
Flickr30K	10,000	50	50	3	50	10.216

To conclude, it could be analysed that although the vocabulary size and the sentence length were varied, it could be identified that the average caption generated by the trained model was not greatly affected. However, overall it could be observed that training and evaluating at a higher maximum sentence length resulted in a generally higher average generated caption length. Another considerable factor is the sentences the architecture is being trained with. In fact, a direct relation could be observed between the average sentence length presented in the dataset as target sentences with the generated captions of the architectures. In future work, experimentation with novel datasets that contain a longer description of the images and therefore present a higher word count could be explored to generate sentences between the 16 and 20 words as recommended by Kadayat and Eika (2020).

5.4.2 Impact on the Image Captioning Architecture

Varying the vocabulary size and the trained maximum caption length might potentially affect the performance of the image captioning architecture. In this section, the above-mentioned variations will be considered with respect to the generated results on the evaluation metrics Bleu, Meteor, Rouge-L and CIDEr.

Table 5.15: Comparing the Proposed Image Captioning Architecture Performance on the Flickr8K trained with a Batch Size of 64 when varying the Vocabulary Size and the Training Caption Length. The best performing architectures are being highlighted by the underlining.

Vocab	Sent Length	Sent Evaluated at a Max Sent Length	Bleu -1	Bleu -2	Bleu-3	Bleu -4	Meteor	Rouge -L	CIDEr
5,000	20	20	62.6	44.1	29.8	19.8	19.8	45.8	48.6
5,000	50	20	57.6	36.7	20.7	11.5	16.8	42.6	35.0
5,000	50	50	46.3	29.0	16.2	9	16.2	41.4	34.0
10,000	20	20	**63.4**	**45.6**	**31.3**	**21.5**	**19.9**	**46.3**	**49.0**
10,000	50	20	61.1	42.5	28.3	18.5	18.8	44.8	41.3
10,000	50	50	60.4	42.3	28.3	18.4	19.2	45.4	45.4

The first analysis to be performed on the Flickr8K dataset as shown in Table 5.15 shows that the performance of the trained models was rather similar. The best performing model consists of a 10,000 word vocabulary trained and evaluated using a limit of 20 words whilst the worst performing model consisted of a 5,000 word vocabulary trained and evaluated over 50 words. The second highest performance could be observed on the 5,000 word vocabulary trained and evaluated over 20 words. A similar pattern could be identified on the Flickr30K as shown in Table 5.16 in which the best performing model consists of the 5,000 word vocabulary trained and evaluated over 20 words whilst the second best performing model consists of a 10,000 word vocabulary trained and evaluated over 20 words. The worst performing architectures in the evaluation consisted of the model with a 10,000 word vocabulary trained over 50 words and evaluated over 20 and 50 words.

Table 5.16: Comparing the Proposed Image Captioning Architecture Performance on the Flickr30K trained with a Batch Size of 64 when varying the Vocabulary Size and Training Caption Length. The best performing architectures are being highlighted by the underlining.

Vocab	Sent Length	Sent Evaluated at a Max Sent Length	Bleu -1	Bleu -2	Bleu -3	Bleu -4	Meteor	Rouge -L	CIDEr
5,000	20	20	**64.0**	**45.3**	**31.4**	**21.5**	18.7	**44.5**	**44.4**
5,000	50	20	62.6	43.9	30.3	20.7	**19.0**	44.2	**44.4**
5,000	50	50	62.3	43.7	30.2	20.6	**19.0**	44.1	**44.4**
10,000	20	20	63.1	44.1	30.3	20.8	18.6	44.1	43.1
10,000	50	20	61.9	42.9	29.2	19.8	18.6	43.3	41.5
10,000	50	50	61.9	42.9	29.2	19.7	18.6	43.3	41.5

Intuitively the conclusions drawn from these results follow. This is due to the fact that the best performing architectures utilised the larger vocabulary and were trained and evaluated on the smallest sentence limit considered for this experimentation. This highlights that the models were trained on a large set of 10,000 words to which only a maximum of 20 words were used to generate the captions. Contrarily it logically follows that the worst performing model as shown for the Flickr8K dataset consists of using the smallest vocabulary size of 5,000 and training and evaluating on the longest sentence word limit considered therefore that of 50 words. Combining the two conducted evaluations it be could deciphered that the relation between the image captioning performance and the average caption lengths generated is inverse. Therefore the model that generated the longest average captions was also considered to be the model with the lowest performance measured on the considered metrics. Inversely, the image captioning model that recorded the highest performance on the consulted metrics generated the shortest average captions. Therefore, it could be identified that as of this architecture a balance should be established according to the requirements between readability and the performance.

To sum up, this section has identified the effect of two pre-processing parameters of the image captioning architecture, the vocabulary and the length of sentences the model is being trained and evaluated on, in relation to the readability as established by Kadayat and Eika (2020). Through this analysis, it was established that although slight effects were distinguished on the performance and the length of the captions this was not sufficient to reach the goal of 16 to 20 words. However, it was identified that the average sentence length generated was slightly higher than that established by Lim and Chan (2019). Moreover, it was observed that the performance of the model and the length of the generated captions were inverse. Furthermore, a relation was observed between the average target sentence length calculated from the respective datasets to the generated captions. A future improvement could be the creation of a novel image captaining dataset that provides more descriptive captions and hence longer captions that could aid in potentially increasing the details to which a trained image captioning model provides hence increasing the caption length.

5.5 Conclusion

This chapter incorporates the evaluation conducted and a discussion on the results achieved. This research contains an incremental approach to the creation of the captioning architecture starting with reviews of the modules that will be used. Therefore, the first section of the evaluation contains the results achieved from the reviews required for the image captioning model encoder. Starting with the instance segmentation, five different algorithms: the Mask R-CNN, Yolact, Yolact++, CenterMask2 and TensorMask were compared in terms of accuracy and efficiency over two datasets establishing that both the Mask R-CNN and the Yolact++ could be considered as the current state-of-the-art due to their exceeding performances on accuracy and inference time respectively. The section that follows contains an analysis and discussion of various saliency prediction algorithms to assist the instance segmentation algorithm to achieve an attention mechanism. Here, the main algorithms considered were the EML-NET, Pyramid Feature Attention Network, Itti's algorithm and DeepGaze2. As assistive algorithms for discussion, saliency ranking algorithms, one of which is the Sara algorithm were also used to back up the research conducted and validate the reasoning of introducing saliency prediction to the architecture. From this research it was concluded that saliency prediction provides a strong basis for determining the importance of an object in an image and is more logical than what is currently recognised in research, opting to utilise Itti's algorithm due to its ability to output a level of saliency for every pixel combating loss of data. The third section within the image captioning encoder evaluation consisted of the scene classifiers ResNet-18, ResNet-50, AlexNet and DenseNet-161 to which the overall best deep learning model that gave the optimal desired balance between accuracy and inference time was determined to be the ResNet-50 pre-trained on the Places-365. The second set of experiments conducted revolved around the image captioning model. Here, different text encodings for the decoder of the architecture were explored in addition to varying the language transformer's hyper-parameters to find the optimal results, concluding that an Index-Based text embedding, a dropout rate of 0.1 and a configuration similar to that provided by the original language transformer research were ideal. Benchmarking on the Flickr8K and Flickr30K the performance of the image captioning model developed for this research was compared to similar state-of-the-art architectures distinguishing that the trained architecture generated exceeding figures for most metrics when compared to similar architectures whilst was comparable to the Mask Captioning Network generating a higher CIDEr and an equal Meteor. Finally, this section concludes with an evaluation on readability to which two different hyper-parameters the vocabulary and the maximum sentence trained and evaluated with were varied to determine their effectiveness on the length of the generated captions. From this analysis, it was identified that relationships exist between the captions in the training datasets and the generated captions in addition to the performance of the captioning models to these variations. The captions generated were between 10 and 15 words which as established by research belonged to the second most readable category group.

Chapter 6

Conclusion

This chapter bring this dissertation to a close starting with a summary of what was implemented and achieved. This chapter then continues by reverting to the aims and objectives introduced at the beginning of this dissertation, discussing each objective and how this objective was met. This chapter concludes by discussing future work consisting of research that could be built to further enhance this work.

6.1 Summary

This research has presented a novel image captioning architecture that is constructed on a pipeline that is built argumentatively with a selection of explainable techniques. Furthermore, this architecture is based on current state-of-the-art techniques and is influenced by rich literature in the field. In addition, the architecture is built on a modular framework in which any module can easily be replaced by improved models and architectures, therefore, providing an opportunity for improved results and modernisation.

The image captioning architecture presented is constructed of an encoder and decoder framework with the encoder consisting of two layers: a mask layer and a scene layer. The mask layer utilises a pre-trained instance segmentation and a saliency prediction algorithm. The elements composing this layer are derived argumentatively and through conducted reviews of current state-of-the-art architectures and research. Firstly, a review of instance segmentation algorithms was considered, exploring architectures such as the Mask R-CNN (He et al., 2017), Yolact (Bolya et al., 2019), Yolact++ (Bolya et al., 2020b), CenterMask (Lee and Park, 2020) and TensorMask (Chen et al., 2019). This review concluded that the current state-of-the-art architecture consists of the Mask R-CNN with an advantage over the accuracy and a competitive inference time. Saliency was introduced to the architecture as part of its attention mechanism. The influential research of the Mask Captioning Network built its attention mechanism based on the confidence level of the instance segmentation algorithm. This research argues that the accuracy of this algorithm does not directly correlate with the importance of an object in the image. Therefore, this research presents a review of saliency prediction algorithms to detect the distinguishability of the objects within the image building the attention mechanism on a stronger argumentative discussion. Different saliency prediction algorithms considered for this aim consisted of the EML-Net (Jia and Bruce, 2020), DeepGaze II (Kümmerer et al., 2016), the Pyramid Feature Network (Zhao and Wu, 2019) and the traditional Itti's saliency algorithm (Itti et al., 1998). Following this discussion, the non-deep Itti's visual saliency algorithm was opted for due to its ability to provide a saliency score for each pixel in the image combating loss of data. The weighted image presented for the mask layer was then calculated through a simple element-wise multiplication between the binary mask generated through the Mask R-CNN, the saliency map generated by Itti's algorithm and the original image. The novel hybrid vision transformer (Dosovitskiy et al., 2021) introduced in 2021 is then being used on the weighted image to extract the mask features.

The second layer of the encoder is the scene layer which is constructed of a simple CNN to extract scene features from the images. Again through an argumentative discussion, a review of scene classification techniques is being performed considering the ResNet-18, ResNet-50 (He et al., 2016), DenseNet-161 (Huang et al., 2017) and AlexNet (Krizhevsky et al., 2012) on the Places-365 Standard dataset (Zhou et al., 2017) concluding that the ResNet-50 is the best overall performing model after establishing a balance between the highest accuracy and the lowest inference time. The features extracted from the mask layer and the scene layer are then concatenated to generate the image features to be used as input for the decoder. The decoder is based on a dedicated language transformer to perform image-to-sequence tasks. The construction of the decoder furthermore contains discussions regarding different text encodings and varied hyperparameters for the transformer architecture. The performance of the trained architecture was then compared to similar architectures in this field concluding that the model generated comparable results with the Mask Captioning Network with a higher CIDEr score while also managing to exceed the results of the other considered architectures. This research concludes with experimentation on the vocabulary size and the target sentence length parameters to analyse their effect on readability. From this analysis, it was concluded that the sentences generated were considered to belong to the second most readable category by Kadayat and Eika (2020). In addition, it was also identified that an inverse relationship exists between the performance of the image captioning model and the longest average caption length and a direct relation exists between the average target sentence length retrieved from the datasets to the generated captions. Moreover, the average length generated by this architecture was slightly higher than that established on the Mask Captioning Network.

6.2 Achieving the Objectives

Reverting to the main aim and objectives of this research it could be concluded that this research reached its aim through the specified objectives. In fact, a modular hybrid image captioning architecture consisting of a combination of transformers and CNNs was built based on an argumentative discussion that is influenced by current research in this area promoting explainability. Furthermore, the presented framework is constructed using a modular structure facilitating the training strategy. Each objective will be analysed in further detail below discussing how these were achieved:

1. Objective 1 - Reviews on state-of-the-art instance segmentation algorithms and saliency prediction and their combined contribution as an attention mechanism for the image captioning architecture were highlighted throughout this study providing an argumentative discussion built on explainability using also saliency ranking algorithms to aid in the discussion.

2. Objective 2 - This research built on rich literature discussed the currently available research in the field of image captioning particularly the Mask Captioning Network that utilises an instance segmentation architecture and depends on its confidence as saliency. The techniques employed in this architecture were discussed with its identified shortcomings used to influence the work in this dissertation.

3. Objective 3 - A hybrid image captioning architecture was implemented that makes use of an encoder and decoder framework using vision and language transformers along with CNNs. Furthermore, this architecture was bench-marked on the Flickr30K and compared to similar architectures. In fact, this architecture exceeded the results of similar image captioning architectures on most metrics and generated comparable results to the Mask Captioning Network with a higher CIDEr and an equal Meteor score.

4. Objective 4 - The vocabulary size and the trained sentence length parameters were explored in distinguishing their effect on readability as well as on the performance of the trained image captioning model.

By reaching its aim, this research strives to leave its contribution through its comparable novel image captioning architecture bench-marked on the Flickr30K dataset that is built on strong argumentative discussions with state-of-the-art methodologies that is constructed on a modular framework allowing this architecture to improve by replacing the models with novel technologies. Further contributions consist of the reviews conducted on current state-of-the-art instance segmentation architectures as well as scene classification models. Moreover, an introduction of saliency algorithms to the field of image captioning architectures was performed to improve current attention mechanisms.

6.3 Future Work

With a flexible modular structure, this research provides the opportunity to extend this work by experimenting with different modules and identifying the impact of the alterations on the performance of the image captioning architecture. Alterations to the instance segmentation architecture, scene classification model, saliency classification algorithm and the mask layer's feature extraction model are all modules that could be replaced by novel or similar architecture that might provide a boost to the performance.

Furthermore, utilising an enhanced computing system provides the opportunity to train with further data and utilise bigger datasets such as the MS-COCO dataset which consists of over 330,000 images with 5 captions per image, totalling 1,650,000 different captions. Training with such a dataset would potentially provide enhanced results in addition to better visibility of the performance of the model. Moreover, although the Flickr8K and the Flickr30K datasets are both heavily used as benchmarking datasets, the MS-COCO Captions due to its volume is more commonly used and therefore by training on this dataset, a comparison to further image captioning architectures could be performed. For this model due to hardware limitations, this was not feasible but remains a future improvement.

The field of image captioning holds immense potential for accessibility as seen for visually impaired individuals. As part of this research, a brief analysis was performed on the readability aspect of the generated captions starting the discussion on adaptability however, further work can be conducted in the area. First and foremost targeted novel caption datasets could be introduced that provide more descriptive and understandable captions. This could be in the form of lengthier captions or through exploring the generation of multiple distinct sentences instead of a single caption for the same image to provide further context. Expanding on this point, the evaluation to determine its effectiveness could then be conducted by packaging the system in an accessible application that visually impaired individuals could utilise and provide feedback.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Al-Malla, M. A., Jafar, A., and Ghneim, N. Image captioning model using attention and object features to mimic human image understanding. *Journal of Big Data*, 9(1):20, 2022.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6077–6086, June 2018. doi: 10.1109/CVPR.2018.00636.

Awad, M., Haddad, J. E., Khneisser, E., Mahmoud, T., Yaacoub, E., and Malli, M. Intelligent eye: A mobile application for assisting blind people. In *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*, pages 1–6, November 2018. doi: 10.1109/MENACOMM.2018.8371005.

Bezdan, T. and Bacanin, N. Convolutional neural network layers and architectures. In *International Scientific Conference on Information Technology and Data Related Research*, pages 445–451, January 2019.

Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J. Yolact: Real-time instance segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9156–9165, November 2019.

Bolya, D., Zhou, C., Xiao, F., and Lee, Y. Yolact. https://github.com/dbolya/yolact, 2020a. Accessed 2021-10-05.

Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J. Yolact++ better real-time instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1108–1121, 2020b.

Borji, A. Saliency prediction in the deep learning era: An empirical investigation. October 2018.

Borji, A. and Itti, L. CAT2000: A large scale fixation dataset for boosting saliency research. *arXiv preprint arXiv:1505.03581*, 2015.

Borji, A., Sihite, D., and Itti, L. What/where to look next? modeling top-down visual attention in complex interactive environments. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 44:523–538, May 2014.

Bylinskii, Z., Judd, T., Borji, A., Itti, L., Durand, F., Oliva, A., and Torralba, A. Mit saliency benchmark. http://saliency.mit.edu/. Accessed 2021-12-01.

Cai, W., Xiong, Z., Sun, X., Rosin, P. L., Jin, L., and Peng, X. Panoptic segmentation-based attention for image captioning. *Applied Sciences*, 10:391, 2020.

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016.

Chen, X., Girshick, R., He, K., and Dollar, P. Tensormask: A foundation for dense object segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2061–2069, November 2019.

Cornia, M., Baraldi, L., Serra, G., and Cucchiara, R. Predicting human eye fixations via an lstm-based saliency attentive. *IEEE Transactions on Image Processing*, 27(10):5142–5154, 2018.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848.

Denkowski, M. and Lavie, A. Meteor universal: Language specific translation evaluation for any target language. In *9th Workshop on Statistical Machine Translation*, pages 376–380, June 2014. doi: 10.3115/v1/W14-3348.

Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *2021 International Conference on Learning Representations (ICLR)*, May 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

Du, K. and Swamy, M. N. S. *Neural Networks and Statistical Learning*, chapter Recurrent Neural Network, pages 337–353. December 2014. ISBN 978-1-4471-5570-6. doi: 10.1007/978-1-4471-5571-3_11.

Elamri, C. and de Planque, T. Automated neural image caption generator for visually impaired people. 2016.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

Fang, H., Zhang, D., Zhang, Y., Chen, M., Li, J., Hu, Y., Cai, D., and He, X. Salient object ranking with position-preserved attention. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16331–16341, October 2021.

Felix, S. M., Kumar, S., and Veeramuthu, A. A smart personal AI assistant for visually impaired people. In *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1245–1250, May 2018. doi: 10.1109/ICOEI.2018.8553750.

Gao, D., Han, S., and Vasconcelos, N. Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31:989–1005, July 2009.

Girshick, R. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, December 2015. doi: 10.1109/ICCV.2015.169.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, June 2014. doi: 10.1109/CVPR.2014.81.

Hafiz, A. and Bhat, G. A survey on instance segmentation: state of the art. *International Journal of Multimedia Information Retrieval*, 9(3):171–189, September 2020. doi: 10.1007/s13735-020-00195-x.

Hayat, M., Khan, S. H., Bennamoun, M., and An, S. A spatial layout and scale invariant feature representation for indoor scene classification. *IEEE Transactions on Image Processing*, 25(10):4829–4841, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016. doi: 10.1109/CVPR.2016.90.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, October 2017.

Herranz, L., Jiang, S., and Li, X. Scene recognition with cnns: Objects, scales and dataset bias. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 571–579, June 2016.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9:1735–80, December 1997.

Hrga, I. and Ivašić-Kos, M. Deep image captioning: An overview. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 995–1000, May 2019.

Huang, G., Liu, Z., Maaten, L. V. D., and Weinberger, K. Q. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, July 2017.

Huang, Z., Huang, L., Gong, Y., Huang, C., and Wang, X. Mask scoring r-cnn. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6402–6411, June 2019. doi: 10.1109/CVPR.2019.00657.

Iqbal, T. and Qureshi, S. The survey: Text generation models in deep learning. *Journal of King Saud University - Computer and Information Sciences*, 2020. ISSN 1319-1578.

Itti, L., Koch, C., and Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998. doi: 10.1109/34.730558.

Jia, S. and Bruce, N. D. Eml-net: An expandable multi-layer network for saliency prediction. *Image and Vision Computing*, 95:103887, 2020.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In *22nd Association for Computing Machinery (ACM) International Conference on Multimedia*, pages 675–678, November 2014.

Jiang, M., Huang, S., Duan, J., and Zhao, Q. Salicon saliency benchmark. http://salicon.net/. Accessed 2021-12-03.

Jiang, M., Huang, S., Duan, J., and Zhao, Q. Salicon: Saliency in context. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., and Qu, R. A survey of deep learning-based object detection. *IEEE Access*, 7:128837–128868, 2019. doi: 10.1109/ACCESS.2019.2939201.

Judd, T., Durand, F., and Torralba, A. A benchmark of computational models of saliency to predict human fixations. In *MIT Technical Report*, 2012.

Kadayat, B. B. and Eika, E. *Universal Access in Human-Computer Interaction. Design Approaches and Supporting Technologies*, chapter Impact of Sentence Length on the Readability of Web for Screen Reader Users, pages 261–271. Springer International Publishing, 2020. ISBN 978-3-030-49281-6.

Karpathy, A. and Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, June 2015.

Katzschmann, R. K., Araki, B., and Rus, D. Safe local navigation for visually impaired users with a time-of-flight and haptic feedback device. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(3):583–593, 2018. doi: 10.1109/TNSRE.2018.2800665.

Khan, M. A., Paul, P., Rashid, M., Hossain, M., and Ahad, M. A. R. An AI-based visual aid with integrated reading assistant for the completely blind. *IEEE Transactions on Human-Machine Systems*, 50(6):507–517, 2020.

Kiros, R., Salakhutdinov, R., and Zemel, R. Multimodal neural language models. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 595–603. PMLR, 22–24 June 2014.

Kiruthika and Sheela. Developing mobile application to navigate blind people using sensors. In *2016 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC)*, pages 080–084, April 2016. doi: 10.1109/ICCPEIC.2016.7557228.

Krizhevsky, A., Sutskever, I., and Hinton, G. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Kümmerer, M., Wallis, T. S. A., and Bethge, M. Deepgaze II: reading fixations from deep features trained on object recognition. *arXiv preprint arXiv:1610.01563*, 2016.

Kümmerer, M., Theis, L., and Bethge, M. Deep gaze I: Boosting saliency prediction with feature maps trained on imagenet. November 2014.

Lazebnik, S., Schmid, C., and Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2169–2178, June 2006.

Le Meur, O., Le Callet, P., Barba, D., and Thoreau, D. A coherent computational approach to model the bottom-up visual attention. *IEEE transactions on pattern analysis and machine intelligence*, 28:802–817, June 2006. doi: 10.1109/TPAMI.2006.86.

Lee, Y. and Park, J. Centermask. https://github.com/youngwanLEE/CenterMask, 2020. Accessed 2021-10-07.

Lee, Y. and Park, J. Centermask : Real-time anchor-free instance segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13903–13912, June 2020.

Li, B., Shi, Y., Qi, Z., and Chen, Z. A survey on semantic segmentation. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1233–1240, November 2018. doi: 10.1109/ICDMW.2018.00176.

Lim, J. H. and Chan, C. S. Mask captioning network. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1–5, September 2019. doi: 10.1109/ICIP.2019.8803004.

Lin, C. Rouge: A package for automatic evaluation of summaries. In *Association for Computational Linguistics (ACL) Workshop: Text Summarization Branches Out*, page 10, January 2004.

Lin, T., Maire, M., Belongie, S., J, H., Perona, P., Ramanan, D., Dollár, P., and Zitnick, L. C. Microsoft COCO: Common objects in context. In *2014 European Conference on Computer Vision (ECCV)*, pages 740–755. Springer International Publishing, September 2014.

Lin, T. Y., Dollár, P., Girshick, R. B., He, K., Hariharan, B., and Belongie, S. J. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, July 2017a.

Lin, T. Y., Goyal, P., Girshick, R. B., He, K., and Dollár, P. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, October 2017b.

Liu, W., Chen, S., Guo, L., Zhu, X., and Liu, J. CPTR: Full transformer network for image captioning. *ArXiv*, abs/2101.10804, 2021.

Liu, Y. P., Chen, Q., Chen, W., and Wassell, I. J. Dictionary learning inspired deep network for scene recognition. In *2018 Association for the Advancement of Artificial Intelligence (AAAI)*, February 2018.

Makav, B. and Kılıç, V. A new image captioning approach for visually impaired people. In *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*, pages 945–949, November 2019. doi: 10.23919/ELECO47770.2019.8990630.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60. Association for Computational Linguistics, June 2014.

Naipal, S. and Rampersad, N. A review of visual impairment. *African Vision and Eye Health*, 77, January 2018.

Oyama, T. and Yamanaka, T. Influence of image classification accuracy on saliency map estimation. *CAAI Transactions on Intelligence Technology*, 3:140–152, 2018.

Padilla, R., Netto, S. L., and da Silva, E. A. A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242, July 2020. doi: 10.1109/IWSSIP48289.2020.

Pal, A., Kar, S., Taneja, A., and Jadoun, V. Image captioning and comparison of different encoders. *Journal of Physics: Conference Series*, 1478:012004, April 2020. doi: 10.1088/1742-6596/1478/1/012004.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. Bleu: A method for automatic evaluation of machine translation. In *40th Annual Meeting on Association for Computational Linguistics (ACL)*, page 311–318. Association for Computational Linguistics, July 2002.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

Patil, K., Jawadwala, Q., and Shu, F. C. Design and construction of electronic aid for visually impaired people. *IEEE Transactions on Human-Machine Systems*, 48(2):172–182, 2018. doi: 10.1109/THMS.2018.2799588.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *2014 Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, October 2014. URL http://www.aclweb.org/anthology/D14-1162.

Quattoni, A. and Torralba, A. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 413–420, June 2009.

Rashtchian, C., Young, P., Hodosh, M., and Hockenmaier, J. Collecting image annotations using amazon's mechanical turk. In *NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, page 139–147. Association for Computational Linguistics, June 2010.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, June 2015. doi: 10.1109/TPAMI.2016.2577031.

Seychell, D. *An Efficient Saliency Driven Approach for Image Manipulation*. PhD thesis, University of Malta, 2021.

Seychell, D. and Debono, C. J. Ranking regions of visual saliency in rgb-d content. In *2018 International Conference on 3D Immersion (IC3D)*, pages 1–8, December 2018. doi: 10.1109/IC3D.2018.8657902.

Shandu, N. E., Owolawi, P. A., Mapayi, T., and Odeyemi, K. AI based pilot system for visually impaired people. In *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, pages 1–7, August 2020. doi: 10.1109/icABCD49160.2020.9183857.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *The 3rd International Conference on Learning Representations (ICLR)*, May 2015.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015. doi: 10.1109/CVPR.2015.7298594.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

Szegedy, C., Ioffe, S., and Vanhoucke, V. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

T. Mikolov, K. C., Corrado, G. S., and Dean, J. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at International Conference on Learning Representations (ICLR)*, January 2013.

Tavakoli, H. R., Shetty, R., Borji, A., and Laaksonen, J. Can saliency information benefit image captioning models? April 2017.

Tian, Z., Shen, C., Chen, H., and He, T. FCOS: fully convolutional one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9627–9636, October-November 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *2017 Advances in Neural Information Processing Systems (NIPS)*, volume 30. Curran Associates, Inc., December 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Vedantam, R., Zitnick, C., and Parikh, D. Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, June 2015. doi: 10.1109/CVPR.2015.7299087. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2015.7299087.

Villanueva, J. and Farcy, R. Optical device indicating a safe free path to blind people. *IEEE Transactions on Instrumentation and Measurement*, 61(1):170–177, 2012. doi: 10.1109/TIM.2011.2160910.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. Show and tell: A neural image caption generator. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, June 2015.

Wilson, J. AI Pool. https://files.ai-pool.com/d/DV8TLgkWsAEGsEs.jpg, 2019. Accessed 2021-10-20.

World Health Organization. World report on vision, 2019.

World Wide Web Consortium. *Accessibility Requirements for People with Low Vision*. 2016. Accessed 2021-10-26.

Wu, Y., Kirillov, A., Massa, F., Lo, W. Y., and Girshick, R. Detectron2. https://github.com/facebookresearch/detectron2, 2019. Accessed 2021-10-03.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3485–3492, June 2010. doi: 10.1109/CVPR.2010.5539970.

Xiao, X., Wang, L., Ding, K., Xiang, S., and Pan, C. Deep hierarchical encoder–decoder network for image captioning. *IEEE Transactions on Multimedia*, 21(11):2942–2956, 2019.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In *2015 International Conference on Machine Learning (ICML)*, pages 2048–2057, July 2015.

Yang, X., Yuan, S., and Tian, Y. Assistive clothing pattern recognition for visually impaired people. *IEEE Transactions on Human-Machine Systems*, 44(2):234–243, 2014. doi: 10.1109/THMS.2014.2302814.

Yelamarthi, K. and Laubhan, K. Navigation assistive system for the blind using a portable depth sensor. In *2015 IEEE International Conference on Electro/Information Technology (EIT)*, pages 112–116, May 2015. doi: 10.1109/EIT.2015.7293328.

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics (TACL)*, 2: 67–78, December 2014.

Zanca, D. and Gori, M. Variational laws of visual attention for dynamic scenes. In *31st Annual Conference on Neural Information Processing Systems (NIPS)*, December 2017.

Zeng, D., Liao, M., Tavakolian, M., Guo, Y., Zhou, B., Hu, D., Pietikäinen, M., and Liu, L. Deep learning for scene classification: A survey. *ArXiv*, abs/2101.10531, 2021.

Zhang, J. and Sclaroff, S. Saliency detection: A boolean map approach. In *2013 IEEE International Conference on Computer Vision (ICCV)*, pages 153–160, December 2013. doi: 10.1109/ICCV.2013.26.

Zhao, T. and Wu, X. Pyramid feature attention network for saliency detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3085–3094, June 2019.

Zhao, Z., Zheng, P., Xu, S., and Wu, X. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, PP:1–21, January 2019. doi: 10.1109/TNNLS.2018.2876865.

Zhou, B., Lapedriza, , Xiao, J., Torralba, A., and Oliva, A. Learning deep features for scene recognition using places database. *Advances in Neural Information Processing Systems*, 1, May 2015.

Zhou, B., A.Lapedriza, A.Khosla, Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places365. https://github.com/CSAILVision/places365, 2017. Accessed 2021-10-09.

Zhu, X., Liu, J., Haipeng, P., and Niu, X. Captioning transformer with stacked attention modules. *Applied Sciences*, 8:739, May 2018. doi: 10.3390/app8050739.