Attention in Computer Vision

Sen He

Department of Computer Science
University of Exeter

Supervisor: Dr. Nicolas Pugeault

This thesis is submitted for the degree of
Doctor of Philosophy

College of Engineering, Mathematics,
and Physic Sciences

July 2021

To my parents for their love and support

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Sen He
July 2021

Acknowledgements

First of all, I would like to thank my supervisor, Nicolas Pugeault, for his supervision and support in the last three years. To me, he is the best supervisor on the world, he can always guide me in right research direction. I very much enjoyed the three years' PhD study under his supervision. In the meanwhile, I want to thank all the members in the innovation center A1 and M1, University of Exeter. You gave me great help. I had a wonderful memory in the last three years to be with you in the office, talk about our research ideas, future plans, amusing incidents of every day's life and dine together in the Chinese restaurant. I also quite enjoyed the time every day in the gym with Dr. Binling Cheng. After one day's work in the lab, I feel refreshed with one hour's exercise in the gym. I was lucky enough to attend the computer vision summer schools in the last three years, where I met a lot of good friends, Yanling Qian, Hongxiang Tang, etc. I still remember we swam in the Mediterranean sea, played volleyball on the beach, enjoyed the local Sicilian food and wine. In the last summer of my PhD study, I was lucky to do a summer internship at Cambridge Toshiba Research, Computer Vision Lab, where I worked with Dr. Chao Zhang, Dr. Stephan Liwicki, etc. Thank you for giving me the opportunity to study the interesting spherical CNN. I want to thank to my collaborators, Ali Borji, Hamed Tavakoli and Wentong Liao, during my PhD study. I had a great experience working with you together, discussing our ideas and polishing our paper. We always working together to the last minute of the deadline. I wish we can keep our collaboration to do more influential works. Last but not least, I thank my parents for embracing me with love.

Abstract

Thanks to deep learning, computer vision has advanced by a large margin. Attention mechanism, inspired from human vision system and acts as a versatile module or mechanism that widely applied in the current deep computer vision models, strengthens the power of deep models. However, most attention models have been trained end-to-end. Why and how those attention models work? How similar is the trained attention to the human attention where it was inspired? Those questions are still unknown to us, which thus hinders us to design a better attention model, architecture or algorithm that can further advance the computer vision field. In this thesis, we aim to unravel the mysterious attention models by studying attention mechanisms in computer vision during the deep learning era.

In the first part of this thesis, we study bottom-up attention. Under the umbrella of saliency prediction, bottom-up attention has progressed a lot with the help of deep learning. However, the deep saliency models are still a black box to us and their performance has reached a ceiling. Therefore, the first part of this thesis aims to understand what happened inside the deep models when it is trained for saliency prediction. Concretely, this thesis dissected each individual unit inside a deep model that has been trained for saliency prediction. Our analysis discloses the secrets of deep models for saliency prediction as well as their limitations, and give new insights for future saliency modelling.

In the second part, we study top-down attention in computer vision. Top-down attention, a mechanism usually builds on top of bottom-up attention, has achieved great success in a lot of computer vision tasks. However, their success raised an interesting question, namely, "are those learned top-down attention similar to human attention under the same task?". To answer this question, we have collected a dataset which recorded human attention under the image captioning task. Using our collected dataset, we analyse what is the difference between attention exploited by a deep model for image captioning and human attention under the same task. Our research shows that current widely used soft attention mechanism is different from human attention under the same task. In the meanwhile, we use human attention, as a prior knowledge, to help machine to perform better in the image captioning task.

In the third part, we study contextual attention. It is a complementary part to both bottom-up and top-down attention, which contextualizes each informative region with attention. Prior contextual attention methods either adopt the contextual module in natural language processing that is only suitable for 1-D sequential inputs or complex two stream graph neural networks. Motivated by the difference of semantic units between sentences and images, we designed a transformer based architecture for image captioning. Our design widens original transformer layer by using the 2-D spatial relationship and achieves competitive performance for image captioning.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Vision, as a primary modality, on which humans heavily rely to perceive the world. However, the visual information enters into our vision system at about 10^8 to 10^9 bits per second [Borji et al., 2013]. From a search perspective, handling vast amount of visual information is computationally intractable [Tsotsos, 1992], as the processing complexity will exponentially increase with respect to the input visual information [Tsotsos, 2011]. ***Visual attention***, a mechanism embedded in human vision system that selectively focus on salient regions of our surrounding environment, is a solution for the human vision system to process the visual information in real time. As illustrated in Figure 1.1 for image or scene understanding, a human observer only need to focus on few salient regions in the image, e.g., computer and man, to understand the gist of the scene and then describe the image as "a man in front of a computer". Fundamentally, attention mechanism helps human vision system to: (1) automatically and rapidly find out most informative or salient regions in the image. (2) more accurately recognize the salient objects in the scene. (3) more precisely understand the semantic and/or spatial relationships between salient objects or stuff. Thanks to visual attention, humans can efficiently perceive the visual world.

In computer vision, a long standing goal is to build an artificial vision system that is able to efficiently understand its input visual information and accurately perform some vision tasks (e.g., recognition and detection). Due to the limited computation and sub-optimal architecture's design compared to human vision system, it is thus crucial to equip the artificial vision system with an ***artificial attention algorithm*** that helps it to efficiently process and understand the raw input pixels' information in real time. For example, in a holistic scene understanding problem, an artificial vision system should be able to rapidly obtain the gist of the scene. This requires an attention algorithm, like human visual attention , which can efficiently find out the main objects and background of the scene, the inter-object interactions, etc. The ultimate goal of research about attention in computer vision is to design a system

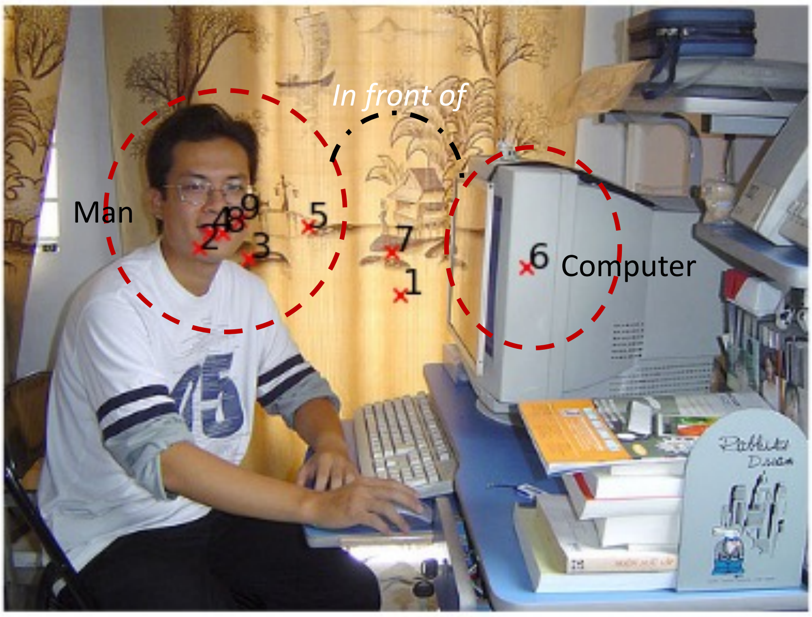


Fig. 1.1 A concrete example of human attention in image understanding. Red crosses indicate eye fixations when the human observer look at the image. Digits represent the order of eye fixations.

with attention mechanism that can help to close the gap between human vision system and machine vision system. Achieving this goal can help us to enter the era of artificial intelligent.

1.1 Problem Statement

This thesis studies three attention related problems in the context of computer vision: (1) Understanding deep learning based saliency prediction models; (2) Analysing humans' attention in image captioning task and comparing it with machine attention in the same task; (3) Designing effective context module for region contextualization. These problems cover three types of attention mechanism in computer vision, namely, **bottom-up attention, top-down attention, and contextual attention**.

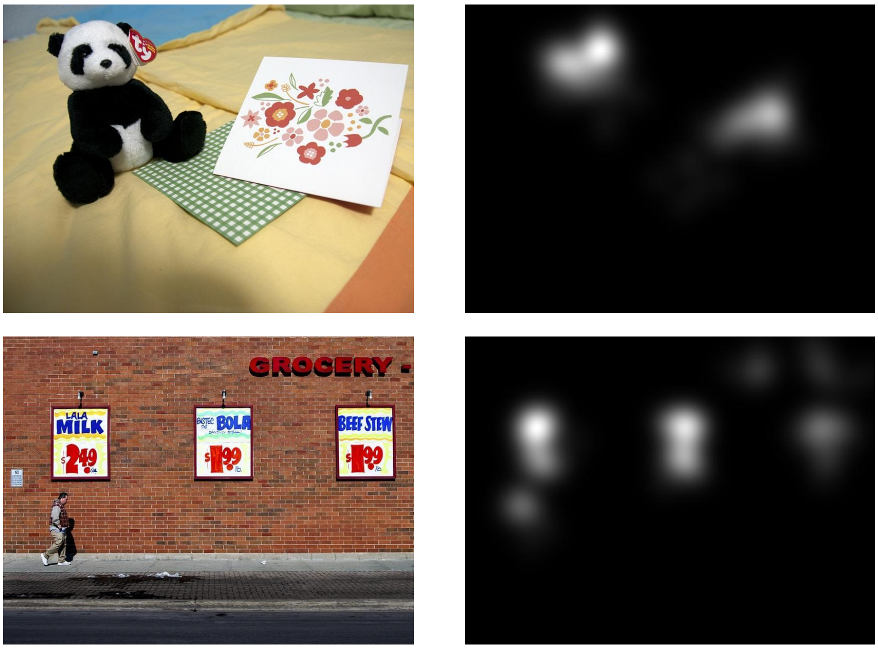


Fig. 1.2 Example of saliency prediction. Left: image, right: saliency map. The ultimate goal of saliency prediction is to predict a heat map in which the value of each pixel indicates the likelihood of that pixel would be fixated.

1.1.1 Bottom-up attention

Early research on attention in computer vision is bottom-up attention. Concretely, researchers try to build a model that predicts where human would focus on in an image, namely, saliency prediction [Borji and Itti, 2013] as illustrated in Figure 1.2. As those salient regions usually carry key information about the image. Therefore, the ability to predict those salient regions would benefit a model for fast content understanding and then efficiently perform the downstream tasks. Classical saliency prediction models [Itti et al., 1998] borrow clues found in psychology [Treisman and Gelade, 1980] and modeling it with feature engineering. Recent advances in saliency prediction [Huang et al., 2015; Jetley et al., 2016] heavily rely on deep architectures and large scale annotated dataset. Although great success has been achieved in recent deep learning based saliency prediction, some problems have occurred. First, deep saliency models are a black box, what makes it good for saliency prediction is still unknown to us. Second, the performance of deep saliency prediction models have already reached a ceiling. What are the limitations of deep models for saliency prediction and when they will fail to the classical models? The study of bottom-up attention in this thesis tries to address those problems.

1.1.2 Top-down attention

With bottom-up attention, a model can find out most informative regions in the image. While, for a specific vision task, not every regions are equal. Top-down attention [Rao and Ballard, 1999], an algorithm that selects the most relevant regions in image for a task, has gained much attention in the deep learning era. Although many top-down attention algorithms have been proposed and they have shown amazing qualitative results [Xu et al., 2015]. A question raised is how can we trust those top-down attention algorithms or are they similar to human attention under the same task? If not, can human attention guide or help a model to do a task?

1.1.3 Contextual attention

The informative regions founded by bottom-up attention algorithm are independent from each other. Therefore, a key characteristic for each region is missing, namely, context information. Contextual attention [Torralba, 2001], which tries to contextualize each informative region with attention algorithm, has been very popular in both computer vision and natural language processing [Vaswani et al., 2017]. However, current contextual attention algorithms in computer vision are either ineffective or not well suit to the image's structure. A fundamental problem is that those methods did not consider the natural structure of images. Can we design an contextual attention mechanism that is both effective and well suited to images?

1.2 Motivation

What should be machine attention composed of? As the example illustrated in Figure 1.1, humans can effortlessly describe an image with only a few fixations on the main objects and stuff in the image. Intrinsically, there are two attention processes that happened in human observer's vision system, i.e., overt attention (eye movements) and covert attention (recognition process) [Ahmad, 1992]. Can a machine vision system also do that like humans with the help of attention? Essentially, machine attention requires three distinct but cooperative attention mechanisms, i.e., bottom-up attention, top-down attention and contextual attention. Bottom-up attention, which finds out all informative regions from the visual input, is the first step of a machine vision system and plays an important role for the down stream tasks. Top-down attention, which selects most task-relevant regions from bottom-up attention, is a crucial part for model's decision making. Contextual attention, which contextualizes each informative region through attention, advanced a machine vision system to perform more high level tasks, e.g., image captioning and visual question answering. Those three attention mechanisms cooperate with each other, and enable a machine vision system to achieve human level performance. Studying those three attention mechanisms is the priority in the research about attention in computer vision.

Indeed, there have been many works about those attention mechanisms in computer vision. Early research focus bottom-up attention modelling [Itti et al., 1998], which progressed a lot with the help of deep learning and large scale eye-tracking dataset [Jiang et al., 2015]. More recent advancements have applied top-down and contextual attention into deep models for a lot of down stream tasks [Anderson et al., 2018; Vaswani et al., 2017]. Thanks to deep learning techniques and large scale annotated dataset, all attention models have achieved great success [Huang et al., 2019; Pan et al., 2016; Wang et al., 2017a]. However, deep learning based attention models are still mysterious to us:

- **What makes deep models good at bottom-up attention modelling? And what is the limitations of deep models for attention modelling such that it has already reached a ceiling at the moment?**

- **Top-down attention mechanism, originally motivated from human vision system and widely applied in artificial machine vision system at the moment. Are the attention mechanism exploited in the artificial machine vision system similar to human attention?**

- **Many contextual attention mechanisms are borrowed from natural language processing community. What is their limitations when it is applied to computer vision problem? How can we develop more advanced contextual attention algorithm for computer vision?**

Answering those questions can help us to understand deep attention models such that we can design a better attention model for computer vision. Concretely, (1) with a better understanding to the deep saliency prediction models, we can develop an advanced saliency model that can find out all salient regions in the image, which is a crucial part for bottom-up attention. (2) By comparing machine's top-down attention and human attention, we can first realize the reliability of current attention models. We can then develop a better top-down attention algorithm by comparing the difference between human attention and machine attention. (3) With a better contextual attention algorithm, the machine vision system will be advanced to process more high level tasks and more complex scenarios, such that it is more close to human level intelligence.

1.3 Approach

Having discussed the problems to be explored, this thesis proposes novel approaches to tackle those problems:

- **Visualizing deep saliency models.** Deep models are intrinsically black boxes. To understand deep models for saliency prediction, the best way is to look inside the deep models trained for saliency prediction. To this end, we first annotate all salient regions in OSIE [Xu et al., 2014] based on some salient region categories [Bylinskii et al., 2016]. We then use this dataset to compute the correlation between ground truth salient regions and the activcation maps of a model trained for saliency. In this way, we can understand what has been learned of a model trained for saliency prediction, and what is still missing.

- **Collecting human attention data.** How to trust an attention algorithm applied in a deep model? A straightforward method is to compare it with human attention under the same task, i.e., whether they attend to the same region in decision making. However, previous dataset do not allow us to do that, as most of them are collected for free-viewing saliency research. To this end, we take image captioning task as example, and collect human attention data when they were asked to describe an image. With the collected data, we can compare the difference between human attention and machine attention in the same task.

- **Integrating spatial attention for context modelling.** A fundamental difference between text and images is the 2D spatial organisation of the latter. To simply borrow the powerful self-attention [Vaswani et al., 2017] in sentences for image context modelling is inadequate. We propose to combine relative spatial relation and self-attention for better image context modelling.

1.4 Contributions and Publications

In this section, we summarize the main contributions of each chapter in this thesis as follows:

- In Chapter 3, we study bottom-up attention in computer vision. Concretely, we study and interpret the inner representation learnt by deep saliency prediction models. Compared to other deep learning based saliency prediction works that focus on improving the performance of saliency prediction. Our work tries to interpret the deep models for saliency prediction. Our study in chapter 3 uncovers the secrets inside the deep models for saliency prediction. Our experiments show that a model trained on large scale eye-fixation dataset learned some useful representations for saliency prediction; Our analysis reveals the shortcoming of data-driven saliency prediction methods; And our further comparison also tell us that deep saliency prediction models fail to the classical models in some special cases. Our work in chapter 3 gives many insights to the community for future deep learning based saliency research. Our work in chapter 3 has been published in :

Sen He, Hamed R. Tavakoli, Ali Borji, Yang Mi, and Nicolas Pugeault. Understanding and visualizing deep visual saliency models. International Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

- In Chapter 4. We study top-down attention. Concretely, we investigate the attention mechanism in image captioning. Compared to other works that try to design an automatic attention mechanism for image captioning. Our work tries to understand whether the automatic attention mechanism learnt from data is the same thing as human attention under the same task? To this end, we collected a dataset which recorded human attention under the image captioning task. With the collected data, we compare the difference between human attention and machine attention. Our experiment found that human attention and machine attention under image captioning are totally different. Our dataset and experiments in chapter 4 give many new insights to the community about the machine attention and human attention. The work in chapter 4 has been published in:

  Sen He, Hamed R. Tavakoli, Ali Borji, and Nicolas Pugeault. Human Attention in Image Captioning: Dataset and Analysis. International Conference on Computer Vision (ICCV), 2019.

- In Chapter 5, we study contextual attention. More concretely, we investigate how an artificial vision system can have an attention mechanism that automatically contextualizes each informative region in the image through attention. Compared to prior works that heavily reply on the original context modelling in natural language processing or complex graph neural networks. Our work proposed an efficient and effective contextual attention mechanism based on the 2D spatial relationships between informative regions in image. Our proposed method works well for image captioning and achieves promising results. The work in chapter 5 has been published in:

  Sen He, Wentong Liao, Hamed R. Tavakoli, Michael Yang, Bodo Rosenhahn, and Nicolas Pugeault. Image Captioning through Image Transformer. Asian Conference on Computer Vision (ACCV), 2020

In addition to the above contributions which address the main research problems stated in Section 1.1. The author of this thesis also made contributions to the following publications. Within these publications, some of them are related to the work this thesis (work in chapter 3), but will not be included.

- ***Sen He, Nicolas Pugeault. Deep saliency: What is learnt by a deep network about saliency? 2nd Workshop on Visualization for Deep Learning in International Conference on Machine Learning (ICML), 2017***

- ***Sen He, Ali Borji, Yang Mi, and Nicolas Pugeault. What Catches the Eye? Visualizing and Understanding Deep Saliency Models. Arxiv, 2018***

- ***Chao Zhang*, Sen He*, and Stephan Liwicki. A Spherical Approach to Planar Semantic Segmentation. British Machine Vision Conference (BMVC), 2020.***

1.5 Thesis Structure

The reminder of this thesis is organized as follows:

- Chapter 2 will present a comprehensive history for early research on bottom-up attention, evaluation of saliency prediction, top-down attention and contextual attention in deep learning era.

- Chapter 3 will introduce our work on bottom-up attention, deep learning based saliency prediction models, as well as their limitations.

- Chapter 4 will present our work on top-down attention and a comprehensive comparison between human attention and machine attention under the image captioning task.

- Chapter 5 will introduce our work on contextual attention. We proposed an effective contextual attention method for image captioning and achieved competitive results.

Chapter 2

Background

In this chapter, we review the backgrounds of three types attention models. First, we briefly introduce how attention mechanism works in human vision system, as well as the related psychological research on human attention. Then, we review early bottom-up attention modelling in computer vision and its evaluation. After that, we introduce top-down and contextual attention that are widely applied in deep computer vision models.

2.1 Attention in Human Vision

Neuroscientists have already tried to investigate the attention mechanism in human vision system. Their research suggests a prototype for the attention process in human vision system as illustrated in Figure 2.1. During visual information processing, the raw visual information is first processed by the retina, and then fed into the Lateral Geniculate Nucleus (LGN). After that, the processed information goes through our visual cortex, which is then divided into two parallel hierarchical streams, namely, ventral stream and dorsal stream. Ventral stream is believed to be responsible for the recognition of the objects in the scene while the dorsal stream is believed to provide the control information of the eye movements. The processed information is then fused at the prefrontal cortex which sends the signal to the superior colliculus for the new eye-movement. Our eye movements (overt attention) are a proxy variable for monitoring attention. The revealed attention process in human vision system has been a guidance in early computational attention modelling [Ahmad, 1992; Welleck et al., 2017].

However, due to the complexity of human brain and human vision system, the detailed interaction between each unit in human vision system is still unknown to us, which impedes human-like attention modelling in computer vision. That being said, human attention mechanism is still worth to explore. Once the secrets in human vision system are unraveled,

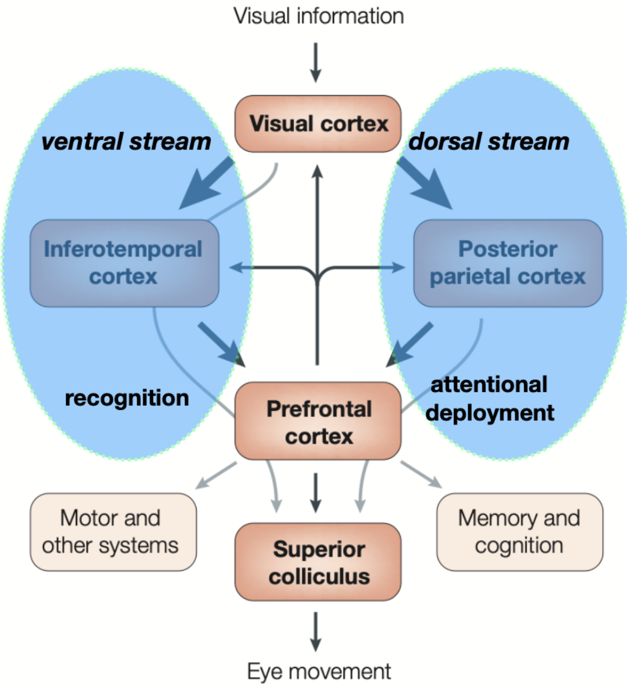


Fig. 2.1 The overview of visual attention process in visual cortex in human vision system [Itti and Koch, 2001].

it can be used to measure the differences between human and machine, which is a crucial part for the interpretability of machine intelligence. This thesis will study the human attention on image captioning task and compare it with machine attention on the same task.

2.2 Attention in Psychology

In psychology, people are more interested in finding out what is attracting human visual attention. Yarbus [2013] studied the role of eye movements in vision. Treisman and Gelade [1980] proposed the Feature Integration Theory, where they assume that visual scene is initially coded along a number of separable dimensions, such as color, orientation, etc. To recombine those separate representations and to ensure the correct synthesis of features for each object in a complex environment, stimulus locations are processed serially with focal attention. On the other hand, Wolfe et al. [1989] proposed Guided Search, which claims that early stages of visual system processes all locations in parallel but are capable of extracting only limited amount of information from the visual input, and guidance signal is used to deploy the focal attention to the location of interest. There are no contradiction between the above mentioned two theories, the difference between them is that feature integration theory is bottom-up attention based theory, and the guided search is top-down attention theory. Bottom-up attention freely explore the visual environments without any bias to a specific stimulus while the top-down attention is guided by a task and biased to some stimulus interested to the task.

Those proposed theories and prototypes about visual attention in psychology have been used in early bottom-up attention modelling (saliency prediction) [Itti et al., 1998; Ouerhani and Hugli, 2000]. However, (1) many of those psychological theories are built on controlled experiments with synthetic images; (2) the prevalent cues for attention modelling are variations of central surround differences. When it is used to modelling attention on **natural images**, the developed bottom-up attention models usually have poor performance that is far behind current deep learning based models [Bylinskii et al., 2012]. On the contrary, this thesis compares between those classical models based on psychological clues and data-driven deep models on **synthetic images** in section 3. Surprisingly, classical models outperform all state-of-the-art deep models. This result suggests that the psychological clues and theories are still useful for attention modeling for synthetic images.

2.3 Early Bottom-Up Attention in Computer Vision

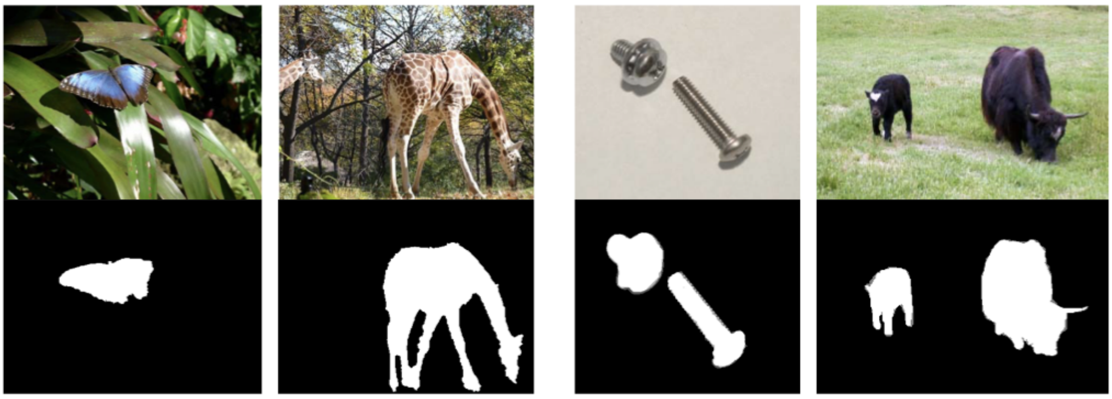


Fig. 2.2 Example for salient object detection [Borji et al., 2015].

In early days, most researches about attention in computer vision focus on bottom-up attention. In general, there are two subtasks for bottom-up attention in computer vision, namely, saliency prediction and salient object detection [Borji et al., 2015]. Saliency prediction aims at predicting a heat map which indicates the probability a region human may attend to in an image, while salient object detection is to segment out most salient objects (foreground segmentation) in the scene (binary segmentation , example in Figure 2.2). As the research of this thesis focus on saliency prediction, we will not discuss salient object detection in the following part (we refer the reader to Wang et al. [2019a] for detailed introduction of salient object segmentation).

Automatically predicting regions of high saliency in images is a crucial step for founding out most informative regions, it is also useful for applications including content-aware image re-targeting, image compression and progressive transmission, object and motion detection, image retrieval and matching, etc.

Human eye fixations on the image during free-viewing is often regarded as ground truth of image saliency. Computational models producing a saliency value at each pixel of an image are referred to as saliency prediction models. In the rest of this section, we will give a brief introduction of the data collection for saliency research, classical computational saliency modelling as well as the evaluation metrics for saliency models.

Before introducing the relevant literature in saliency prediction research, we first introduce three basic concepts:

Saliency Prediction: saliency prediction, in the context of computer vision, means predicting a heat map that indicates the probability of each region people may attend in a given image under free viewing condition. The free viewing condition means freely view a given image in 3 to 5 seconds without any required task, e.g., object recognition, to be finished. All the works we will present in chapter 3 are free viewing saliency prediction.

Fixation Map: fixation map for an image is a 2 dimensional binary map, which records the eye fixations from several people when they view the image. It is the raw data used in saliency research.

Saliency Map: saliency map is the continuous representation for the fixation map, it is usually obtained by convolving the fixation map with a Gaussian kernel. Saliency map is the ground truth for the saliency prediction models.

An example for fixation map and saliency map of a given image is illustrated in Figure 2.3

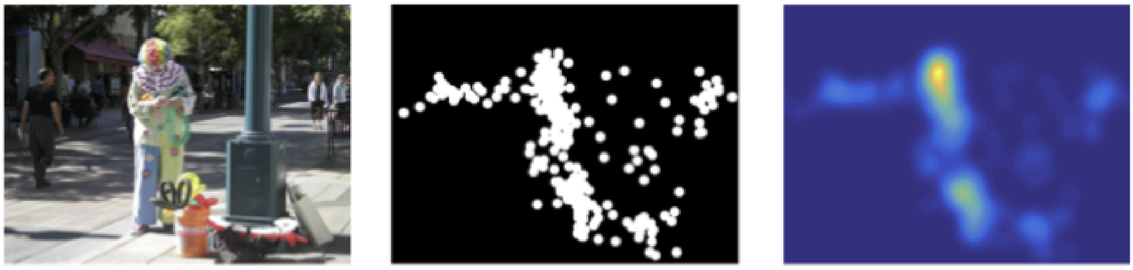
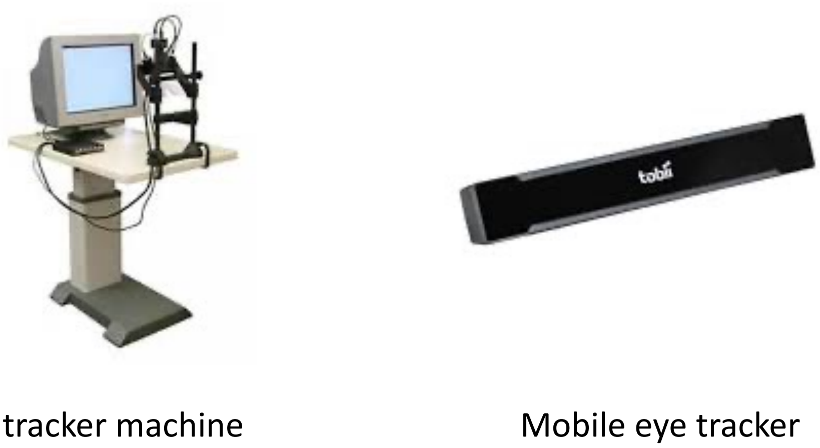


Fig. 2.3 Image and its fixation map, saliency map [Bylinskii et al., 2018].

2.3.1 Data collection for saliency prediction research—eye tracking



Eye tracker machine

Mobile eye tracker

Fig. 2.4 Illustration of Eye tracker machine (left) and mobile eye tracker (right).

To study where people will look in a static image or a video clip, the first step is to collect the data which records the human eye fixations (sometimes people refer it as eye movements). There are three commonly used methods to record the human eye fixations. Eye tracker machine [HNB-13b] (example in Figure 2.4) is the original method with high precision to record the human eye fixations. Eye tracker machine based systems usually require a controlled environments, in which a commander controls the system and an experimenter is asked to view the image or video displayed on a screen for few seconds (usually 2 to 5 seconds). Before data recording, calibration is required. Compared to the eye tracker machine, recent mobile eye tracker [Tobii] (example in Figure 2.4) is more flexible. It is now widely used in the eye movements research. Albeit it still requires calibration at the beginning of data recording, it does not require a controlled environment. And a single experimenter can handle all things in data collection after calibration. Both eye tracker machine and mobile eye tracker are expensive, and it takes a lot of time to record the eye movements. A faster method is to use the coarse mouse clicking [Jiang et al., 2015] to approximate the precised eye tracking system. Under this setting, the experimenter is asked to view the image for few seconds and then to click the positions he or she has looked at. This method allows us to collect the *approximated* eye movements data at large scale. Although the collected data cannot be used for direct model evaluation, it is good for model training [Tavakoli et al., 2017a]. Indeed, the collected data using mouse clicking has lead to the revolution of saliency prediction in the deep learning era [Kummerer et al., 2017; Wang et al., 2017a].

In this thesis, we use mobile eye tracker to collect human eye fixations when they are doing image captioning task. The collected dataset is the largest one at the moment for vision-language research.

2.3.2 Classical methods for saliency prediction

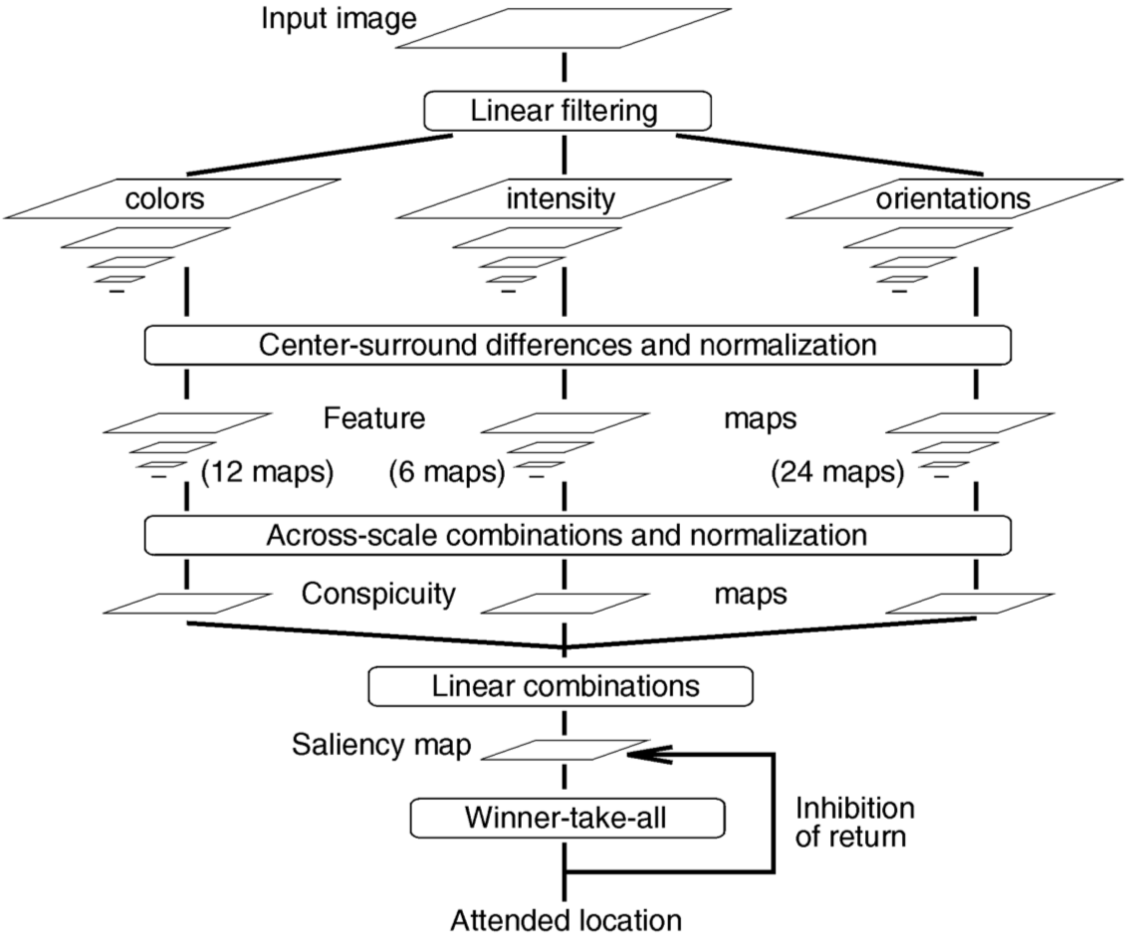


Fig. 2.5 The classical architecture for saliency prediction model by Itti et al. [1998].

Before deep learning era, saliency prediction models are totally unsupervised. The developments of those models are usually based on the *feature integration theory* and *centre-surround theory* proposed in psychological research [Treisman and Gelade, 1980] as stated in section 2.2. For example, the first computational saliency model [Itti et al., 1998] uses the color, intensity and orientations as the features (those features are found attract human attention in early psychological experiments with synthetic images), and applies the central surround theory between the local and surrounding features at multi-scale to compute the final saliency map (see Figure 2.5). Later variants on this model also add depth feature on this model [Ouerhani and Hugli, 2000]. Those classical models are computationally efficient (with only few hand crafted filters) and do not require large scale dataset for training. However, as stated in section 2.2, all the theories and clues used in classical models are based on naive experiments with synthetic images. When the developed models are applied to natural images, they often fail and cannot compete with state-of-the-art deep learning based models [Kummerer et al., 2017; Wang et al., 2017a].

Hou and Zhang [2007] extends the *centre-surround theory* into the frequency domain. Their hypothesis is that plain images, without salient regions inside it, should have a smooth spectrum. Therefore, any fluctuation in the spectrum is caused by the salient patterns in the image. Their method, although simple and effective in some cases, is applied directly on the low level pixel's information. It thus cannot handle high level semantic information.

Bruce and Tsotsos [2005] combined the *centre-surround theory* with information maximization. Different with Itti et al. [1998] that uses some pre-defined filters to extract local features, they instead use some learned filter basis to extract local information. The saliency of each region is then defined by its entropy. Their method is theoretically elegant. However, a closer look at the learned filter basis indicates that it is still using some low level feature filtering. It thus has no fundamental difference compared to Itti et al. [1998], which is limited to predict high level and semantic salient patterns.

This thesis focus more on recent deep learning based saliency prediction models (the related works will be detailed in Chapter 3). However, we also compare deep saliency prediction models with classical hand crafted models on **synthetic images**, which is a missing part in the current evaluation setting for saliency prediction models.

2.3.3 Evaluation for saliency prediction models

After the development of a saliency prediction model, we need to evaluate its performance compared to the ground truth saliency map or raw fixation map. In saliency prediction community, there are eight widely used evaluation metrics. Different evaluation metric focus on different factors of a model's prediction. In this part, we give a brief introduction for each evaluation metric and their pros and cons. For detailed comparison among those evaluation metrics, we refer the reader to Bylinskii et al. [2018].

Area under ROC curve—AUC

Area under the ROC curve, referred as **AUC** ($0 < \mathrm{AUC} < 1$), is one of the most used metrics for evaluating saliency prediction models. In **AUC**, saliency map is treated as a binary classifier of fixations at various threshold values (level sets). The ROC curve is meusured by the true positive (tp) and false positive (fp) rates under each binary classifier (level set). And the **AUC** is computed by:

$$\mathrm{AUC} = \int_{\mathrm{fp}} \mathrm{tp}\, d(\mathrm{fp}) \tag{2.1}$$

Different **AUC** implementations differ in the way of calculating true and false positives. An **AUC** variant from Judd et al. [2009], referred as **AUC- Judd**. For a given threshold, the true positive rate (TP rate) is the ratio of true positives to the total number of fixations, where true positives are total number of fixated pixels whose saliency map value is above the threshold. This is equivalent to the ratio of fixations falling within the level set to the total fixations. The false positive rate (FP rate) is the ratio of false positives to the total number of saliency map pixels at a given threshold, where false positives are total number of non-fixated pixels whose saliency map values is above the threshold. This is equivalent to the number of pixels in each level set, minus the pixels already accounted for by fixations. Another variant of **AUC** by Borji et al. [2012], referred as **AUC-Borji**, uses a uniform random sample of image pixels as negatives and defines the saliency map values above the threshold within those pixels as false positives.

As **AUC** scores are a function of which level sets the false positives fall into. Models with many low-valued false positives do not incur large penalties. Therefore, saliency maps that place different amounts of density but at the correct (fixated) locations will receive similar **AUC** scores.

Shuffled AUC—sAUC

The natural distribution of fixations on images tends to include a higher density near the center of an image [Tatler, 2007]. As a result, a model that incorporates a center bias [Itti et al., 1998] into its predictions will be able to account for at least part of the fixations on an image, independent of image content. In a center-biased dataset, a center prior baseline will achieve a high AUC score. Shuffled AUC metric, proposed in Borji et al. [2013], instead of sampling negatives uniformly from the same image, which samples negatives from fixation locations from other images, is designed to diminish the model's center bias. It has the effect of sampling negatives predominantly from the image center because averaging fixations over many images results in the natural emergence of a central Gaussian distribution.

A model that incorporates a center bias into its predictions is putting density in the center at the expense of other image regions. Such a model will score worse according to **sAUC** compared to a model that makes off-center predictions, because **sAUC** will effectively discount the central predictions. Although the center bias is alleviated in **sAUC**, it still ignore the low-valued false positives.

Normalised scanpath saliency—NSS

Normalized scanpath saliency, **NSS**, was introduced to the saliency community as a simple correspondence measurement between predicted saliency maps and ground truth, computed as the average normalized saliency at fixated locations [Peters et al., 2005]. Given a model's predicted saliency map P and the binary fixation map F of the corresponding image, the **NSS** score for the model prediction is computed as:

$$\begin{aligned} \mathrm{NSS}(P,F) &= \frac{1}{N}\sum \overline{P} \times F \\ \text{where } N = \sum F, &\text{ and } \overline{P} = \frac{P-\mu(P)}{\sigma(P)} \end{aligned} \tag{2.2}$$

where, μ and σ are the mean and variance of the predicted saliency map P.

Unlike in AUC, the absolute saliency values are part of the normalization calculation. **NSS** is sensitive to false positives, relative differences in saliency across the image, and general monotonic transformations. However, because the mean saliency value is subtracted during computation, **NSS** is invariant to linear transformations like contrast offsets. Due to the normalizaion process, a few false positives will be washed out by the other saliency values and will not significantly affect the saliency values at fixated locations. However, as the number of false positives increases, they begin to have a larger influence on the normalization calculation, driving the overall NSS score down.

Pearson's correlation coefficient—CC

The Pearson's Correlation Coefficient, **CC**, also called linear correlation coefficient, is a statistical method that measures the correlation between two variables, computed as:

$$\mathrm{CC}(P,S) = \frac{\sigma(P,S)}{\sigma(P) \times \sigma(S)} \tag{2.3}$$

where $\sigma(P,S)$ is the covariance between P and S.

CC can be used to interpret the predicted (P) and ground truth (S) saliency maps, as random variables to measure the linear relationship between them [Le Meur et al., 2007].

CC is similar to **NSS**, it is thus equally affected by false negatives and positives [Bylinskii et al., 2018].

Earth mover's distance—EMD

The Earth Mover's Distance, **EMD**, measures the spatial distance between two probability distributions over a region. It was introduced as a spatially robust metric for image matching [Rubner et al., 2000]. Fundamentally, it is the minimum cost of morphing one distribution into the other. It is defined as follows:

$$
\begin{aligned}
&\mathrm{EMD}(P,S) = \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}x_{ij}, \\
&\text{subject to} \quad x_{ij} \geq 0, i = 1,\ldots,N, j = 1,\ldots,N, \\
&\qquad\qquad\quad \sum_{j=1}^{N} x_{ij} = s_i, i = 1,\ldots,N, \\
&\qquad\qquad\quad \sum_{i=1}^{N} x_{ij} = d_j, j = 1,\ldots,N
\end{aligned}
\tag{2.4}
$$

where c_{ij} is the cost to transform i_{th} unit in P to j_{th} unit in S. x_{ij} is the flow of i_{th} unit in P to j_{th} unit in S. Generally, saliency maps that spread density over a larger area have larger **EMD** values (i.e., worse scores) as all the extra density has to be moved to match the ground truth map.

In theory, **EMD** is one of the elegant way to measure the distance between the predicted saliency map and ground truth. However, it is computationally inefficient, as it requires iterative optimization during computation. It is therefore too slow when used to evaluate, on large scale datasets, saliency prediction models.

Similarity or histogram intersection—SIM

The similarity metric, **SIM** (also referred to as histogram intersection), measures the similarity between two distributions, viewed as histograms. First introduced as a metric for color-based and content-based image matching [Swain and Ballard, 1991], it has gained popularity in the saliency community as a simple comparison between pairs of saliency maps. **SIM** is computed as the sum of the minimum values at each pixel, after normalizing the input maps.

Given a predicted saliency map P and the ground truth saliency map S:

$$\begin{aligned} \mathrm{SIM}(P,S) &= \sum_i \min(P_i, S_i) \\ \text{where } \sum_i P_i &= \sum_i S_i = 1 \end{aligned} \tag{2.5}$$

where i is the pixel's location.

SIM is computationally efficient, and easy to implement. However, it penalizes models with false negatives significantly more than false positives [Bylinskii et al., 2018].

Kullback-Leibler divergence—KL

Kullback-Leibler (**KL**) is a general information theoretic measure of the difference between two probability distributions. In the saliency literature, **KL** metric takes as input a predicted saliency map P and a ground truth saliency map S, and evaluates the loss of information when P is used to approximate S:

$$\mathrm{KL}(P,S) = \sum_i S_i \log(\varepsilon + \frac{S_i}{\varepsilon + P_i}) \tag{2.6}$$

where ε is a small regularisation constant to avoid numerical issues.

Similar to **EMD**, **KL** has elegant theoretic support to measure the difference between two distributions. However, if the prediction is close to zero where the ground truth has a non-zero value, the penalties can grow arbitrarily large.

Information gain—IG

Information Gain, **IG**, introduced in Kümmerer et al. [2015] as an information theoretic metric that measures saliency model performance beyond systematic bias (e.g., a center prior baseline). Given the binary fixation map F, model's predicted saliency map S, and a baseline map B, the information gain for the predicted saliency map is computed as:

$$\mathrm{IG}(P,F) = \frac{1}{N} \sum_i F_i[\log_2(\varepsilon + P_i) - \log_2(\varepsilon + B_i)] \tag{2.7}$$

where i is the index of the i^{th} fixated pixel, and N is the total number of fixated pixels in the fixation map F.

Similar to **KL**, **IG** can grow arbitrarily large when the prediction is close to zero.

A short summary for different saliency evaluation metrics

To sum up, different evaluation metrics focus on different perspectives of model's prediction. The usage of different evaluation metrics is dependent on the applied task [Bylinskii et al., 2018]. For the basic evaluation of saliency prediction models, we need to consider as more evaluation metrics as possible. For other tasks, **AUC**, **KL**, and **IG** are appropriate for detection applications, as they penalize target detection failures. However, where it is important to evaluate the relative importance of different image regions, such as for image-retargeting, compression, and progressive transmission, metrics like **NSS** or **SIM** are a better fit.

For better and fair comparison of saliency prediction models, a unified saliency evaluation metric is desired [Kummerer et al., 2018]. However, this thesis focus on other more important direction of saliency prediction, i.e., understanding the deep saliency models. In our study, we also adopt some saliency evaluation metrics, e.g., **NSS** and **AUC**, to measure the correlation between saliency maps or fixation maps.

2.4 An Alternative for Bottom-Up Attention

The ultimate goal of bottom-up attention is to predict all informative regions in an image or a video clip. A recent popular alternative for bottom-up attention model is object detection model (such as Faster-RCNN [Ren et al., 2015], as illustated in Figure 2.6).

Many works [Anderson et al., 2018; Huang et al., 2019] have used object detection models to first detect several regions in the input image or video and then use the detected regions as the input for the downstream tasks. Object detection models, as alternative for bottom-up attention, has achieved great progress in image captioning, visual questioning answering [Anderson et al., 2018], etc. In recent years, it actually has been the de facto bottom-up attention for many vision tasks. For example, after the pioneering work done by Anderson et al. [2018], almost all image captioning and visual question answering models adopt such a pipeline, in which an object detection model is used to detect $10-100$ informative regions in the input image and then feed those detected regions into their designed models with soft attention (or contextual attention that will discussed in the next section) for the decision making.

As object detection models explicitly detect semantic informative regions, it supplies more meaningful regions in the input and is more transparent. However, when the object detection models are used as bottom-up attention, they are usually trained with large scale

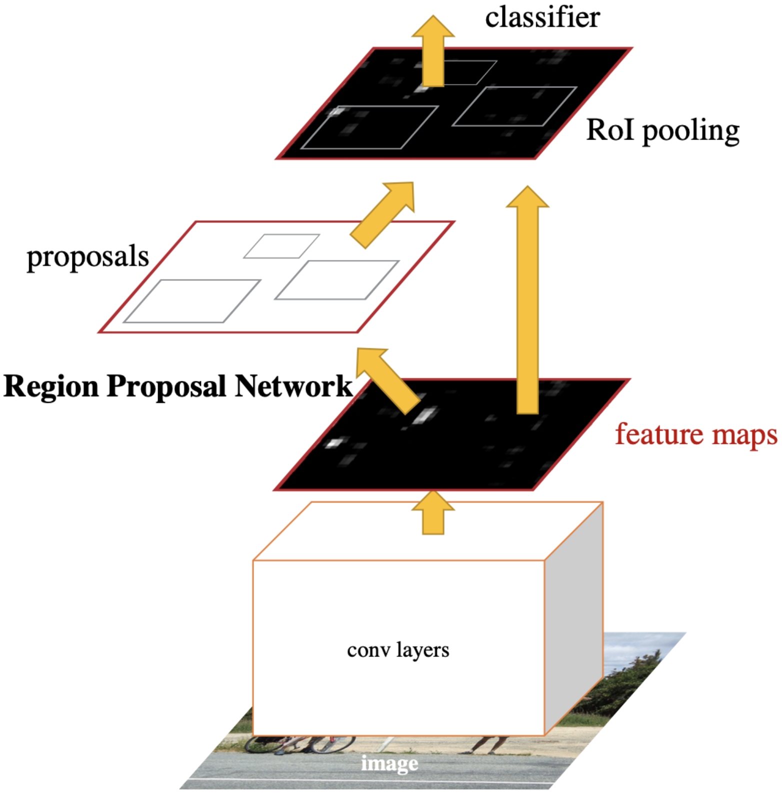


Fig. 2.6 Faster-RCNN pipeline [Ren et al., 2015].

dataset that has thousands of semantic categories[1], such as *Visual Genome* dataset [Krishna et al., 2017]) with more than 1600 categories. Current state-of-the-art object detection architectures cannot handle such case (They can work well when the number of categories is small, e.g., 80 classes in MS-COCO [Lin et al., 2014]) and have a low detection accuracy. The trained object detection model will give wrong detection that may mislead the decision making for the downstream tasks.

2.5 Top-Down Attention

Bottom-up attention, as a preprocessing step, can detect many informative regions in the input image. However, those detected regions are not equal for different downstream tasks. Top-down attention, which for a given task re-processes those informative regions predicted from bottom-up attention, has been widely used in a lot of computer vision tasks [Anderson et al., 2018; Xu et al., 2015].

[1] Note that those semantic categories are not limited to object, but also contain parts of object and also some other background information. For example, the wheel of a car, gloves on the hand, sky, sea, and mountain.

This thesis will focus on top-down attention in computer vision tasks. Based on the way it is used in different models, top-down attention can be categorised into two categories, namely, soft attention and hard attention. The following sections will detail them respectively.

2.5.1 Soft attention

Soft attention, namely, re-weighting each informative region in the images. With soft attention, important regions related to a task should be amplified while other regions would be suppressed. As most soft attention methods are differentiable, it has been widely used in visual recognition [Wang et al., 2017a], image captioning [Xu et al., 2015], and visual question answering [Anderson et al., 2018].

Spatial soft attention

In soft attention, based on how regions are re-weighted, it can be further divided into two groups. Spatial soft attention, which re-weight regions spatially, is frequently used for image understanding. For example, in image recognition, we have an input image I, which is

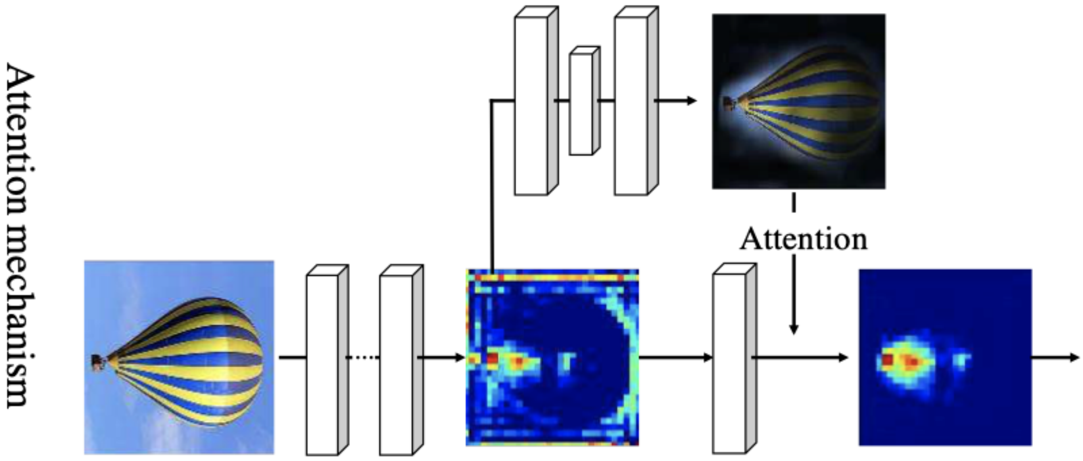


Fig. 2.7 Example of residual soft attention [Wang et al., 2017a].

processed by a convolutional neural network (*CNN*) for feature extraction. Then, we can represent image I with a set of feature vectors $\mathbf{f}$ (bottom-up features):

$$\mathbf{f} = \{f_0, f_1, \ldots, f_{N-1}\} \tag{2.8}$$

each feature vector (f_i, row vector) corresponds to the feature of a region or receptive field in original image. To recognize the object in the image, a model need to focus on the region of object in the image. As we have no direct annotation of where the object is in the image, spatial soft attention is thus designed to force the model to learn to automatically attend to the regions of interest. Concretely, according to the current feature vectors **f**, the attention module learns a weight vector (A) for the feature vectors, which can be formulated as :

$$\mathrm{A} = \mathrm{flatten}(\mathrm{concat}(f_0, f_1, \ldots, f_{N-1})) \cdot w^T \tag{2.9}$$

where w is the parameter (matrix) of a fully connected layer to be learned in the attention module.

The weight for each feature vector can also be computed in a non-parametric way as proposed in Jetley et al. [2018]:

$$\mathrm{A}_i = f_i G^T \tag{2.10}$$

where G can be the output of the first fully connected layer in the deep convolutional neural networks.

After that, the weight vector is normalised:

$$\alpha_i = \frac{\exp^{A_i}}{\sum_{j=0}^{N-1} \exp^{A_j}} \tag{2.11}$$

The normalised weight vector is then used to re-weight the corresponding feature of each region [Jetley et al., 2018]:

$$\widehat{f_i} = \alpha_i f_i \tag{2.12}$$

we can also do feature re-weighting in a residual manner [Wang et al., 2017a] (Figure 2.7):

$$\widehat{f_i} = f_i + \alpha_i f_i \tag{2.13}$$

The re-weighted feature vectors are used for final recognition.

Spatial soft attention is easy to implement and requires only few parameters. It has achieved great success in many computer vision tasks, e.g., visual recognition [Wang et al., 2017a]. However, it is still mysterious to us and lacks explanation. In this thesis, we strive to check the interpretability of soft attention, i.e., whether the learned top-down soft attention is similar to human attention under the same task? Chapter 4 will investigate this problem and provide us an in-depth understanding about the top-down attention.

Temporal soft attention

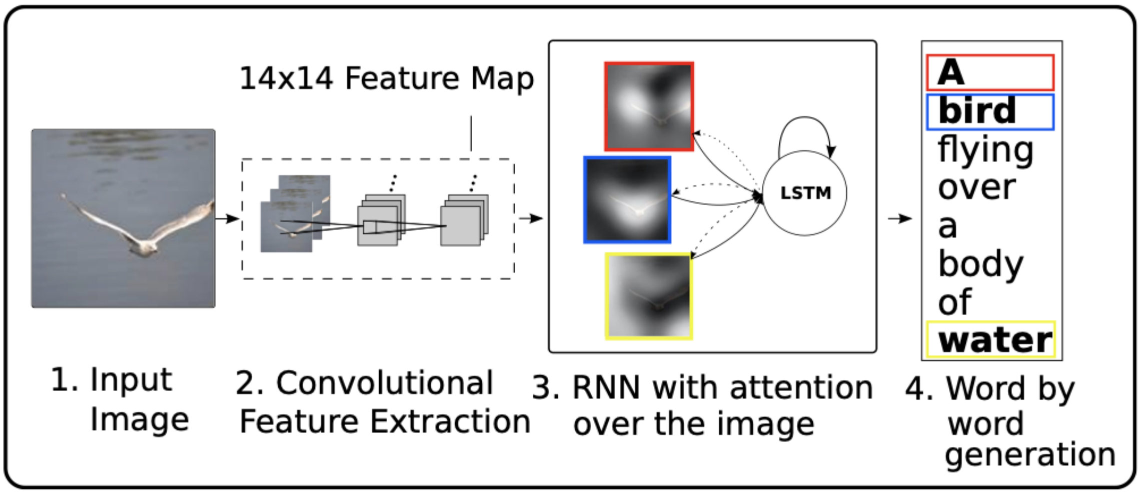

14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Fig. 2.8 Example of temporal soft attention [Xu et al., 2015].

Unlike image recognition, some other computer vision tasks need to take decisions at multiple steps. When applying soft attention in those tasks, a model should be able to attend to the relevant regions not only spatially but also temporally. We define it as temporal soft attention here. For example, in image captioning (Figure 2.8), a model need to generate a sequence of words. In such case, the temporal soft attention is usually combined with recurrent neural networks.

Another new element in temporal soft attention is the state vector s_t from the recurrent neural network. Again, given a set of feature vectors $\mathbf{f}$ of the input image, temporal soft attention usually combines visual input and state vector to generate the attention weight:

$$\begin{aligned} A_t &= (\overline{f} w_v^T + s_t w_s^T) W \\ \alpha_{ti} &= \frac{\exp^{A_{ti}}}{\sum_{j=0}^{N-1} \exp^{A_{tj}}} \end{aligned} \tag{2.14}$$

where, $\overline{f}$ is the mean feature of all feature vectors, w_v, w_s, and W are all parameters to be learned [Xu et al., 2015].

Temporal soft attention has achieved great success in many tasks that need sequential decision making, e.g., image captioning [Anderson et al., 2018; Xu et al., 2015]. Similar to spatial soft attention, it lacks interpretation. How it works and the difference between it and human attention under the same task is unknown to us. Chapter 4 will investigate on it.

2.5.2 Hard attention

In contrast to soft attention, which re-weights the feature of each region in the image. Hard attention directly learns to first "crop" some informative regions from the original image, and then only process the cropped region for the relevant task. Based on the formulation and the training methods of hard attention, it also can be categorised into two categories, namely, differentiable hard attention and non-differentiable hard attention.

Differentiable hard attention

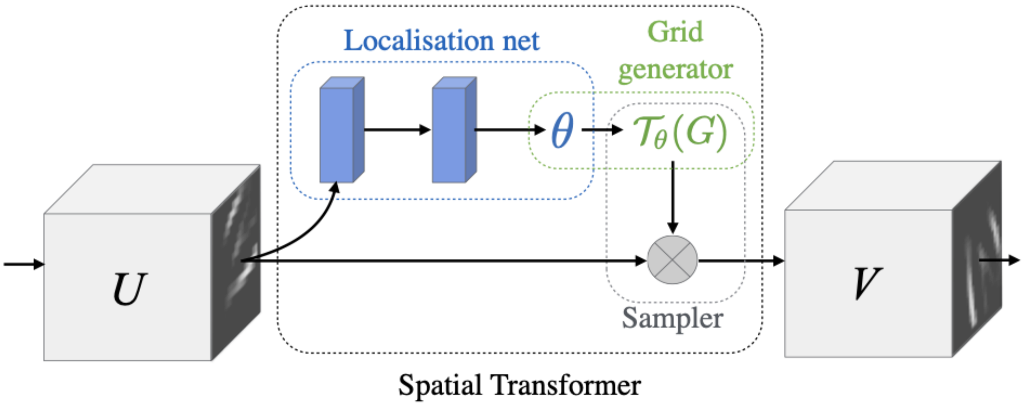


Fig. 2.9 Spatial transformer network [Jaderberg et al., 2015].

A typical differentiable hard attention is the spatial transformer network [Jaderberg et al., 2015] (Figure 2.9). Inside the spatial transformer network, the localisation net (N_{loc}) is the key module. Given an input image I, the spatial transformer network first uses a shallow feature extraction network N_f to extract the features of the whole input image F_I:

$$F_I = N_f(I) \tag{2.15}$$

The extracted feature is used to generate the localisation parameters (affine transformation matrix θ):

$$\theta = \begin{bmatrix} \theta_{i_{11}} & \theta_{i_{12}} & \theta_{13} \\ \theta_{i_{21}} & \theta_{i_{22}} & \theta_{23} \end{bmatrix} = N_{loc}(F_I) \tag{2.16}$$

Then a grid generator will generate a grid in the original image according to the affine transformation matrix θ. Afterwards, a sampler will "crop" the regions of interest in the original image using bi-linear sampling according to the grid. The cropped region is processed by another feature extraction network for the downstream tasks.

Spatial transformer network has achieved great success in digits recognition and fine-grained classification [Jaderberg et al., 2015]. However, it requires two networks to do feature extraction for the input image and the "croped" region, which is computationally inefficient. In chapter 4, we will use spatial transformer network to check if human attention guided bottom-up attention is better than automatically learned bottom-up attention in image captioning task.

Non-differentiable hard attention

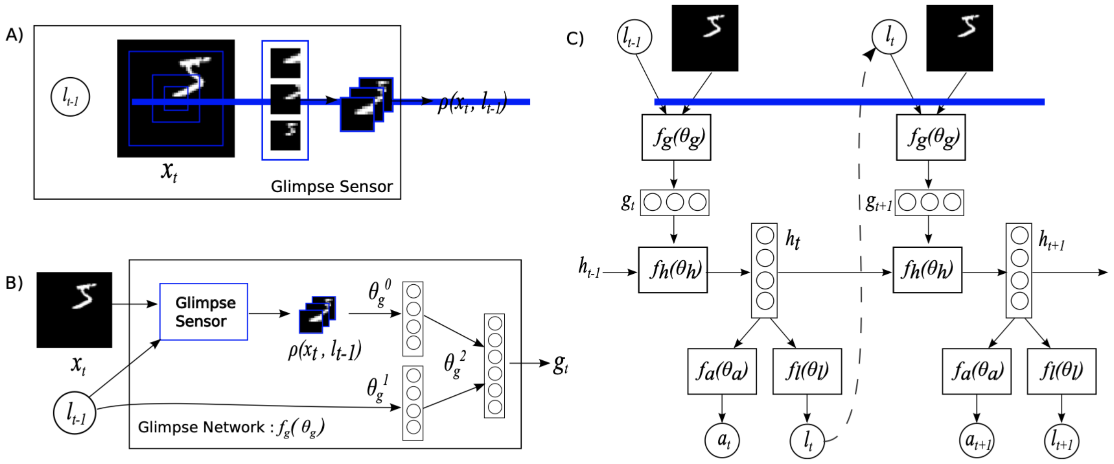


Fig. 2.10 Recurrent hard attention [Mnih et al., 2014].

Different from spatial transformer networks, which is able to learn to sample from the original image and is differentiable. Other hard attention models directly learn central positions of region of interests in image and then crop some regions at multi scale. Those methods are not differentiable and need to be trained with reinforced algorithm. Recurrent attention model (RAM) [Mnih et al., 2014] (Figure 2.10) is a typical non-differentiable hard attention model. RAM consists of a glimpse sensor (GS), glimpse network (GN) and and a recurrent neural network (RNN). Given an input image X, at each time step, the glimpse sensor crops a retina like representation (X_t, with 3 spatial scale regions) centered at the predicted location (l_t) from the recurrent neural network:

$$X_t = GS(X, l_t) \tag{2.17}$$

Then, glimpse network extracts the features for the cropped regions and concatenates them with the current embedded glimpse location feature:

$$g_t = \text{concat}(GN_s(X_t), GN_l(l_t)) \tag{2.18}$$

The concatenated features (g_t) are used to update the internal state (s_{t+1}) of the of the recurrent neural network, which is used to predict the next glimpse location (l_{t+1}):

$$\begin{aligned} s_{t+1} &= RNN(s_t, g_t) \\ l_{t+1} &= f_{\theta_l} s_{t+1} \end{aligned} \tag{2.19}$$

The final state of the recurrent neural network is used for the model's final decision, e.g, object recognition.

As the "cropping" step is not differentiable, the model is hard to train and can be only trained with reinforced algorithm or approximated gradients. It is thus less popular and less explored in the community. Furthermore, it usually requires sophisticated training scheme [Wang et al., 2020], which is not preferable.

2.6 Contextual Attention

Bottom-up and top-down attention discussed in previous sections aim at finding out "where to look" in the input. However, all regions in bottom-up and top-down attention are independent to each other. Contextual attention, which contextualizes each informative region through attention, is a complementary module to both bottom-up attention and top-down attention. It is like a relational graph, where all informative regions are nodes in the graph, but they are independent to each other. And contextual attention is the edge which connects the related nodes together.

Contextual attention is very useful in some high level computer vision tasks, e.g., visual question answering [Li et al., 2019b] and image captioning [Huang et al., 2019]. In those tasks, the model's decision is not only depends on a single region in the image, but also its relationships with other regions.

According to how we build the edge in the relational graph, contextual attention can be grouped into two categories, namely, graph based contextual attention and self-attention based contextual attention. The followiing sections will detail them respectively.

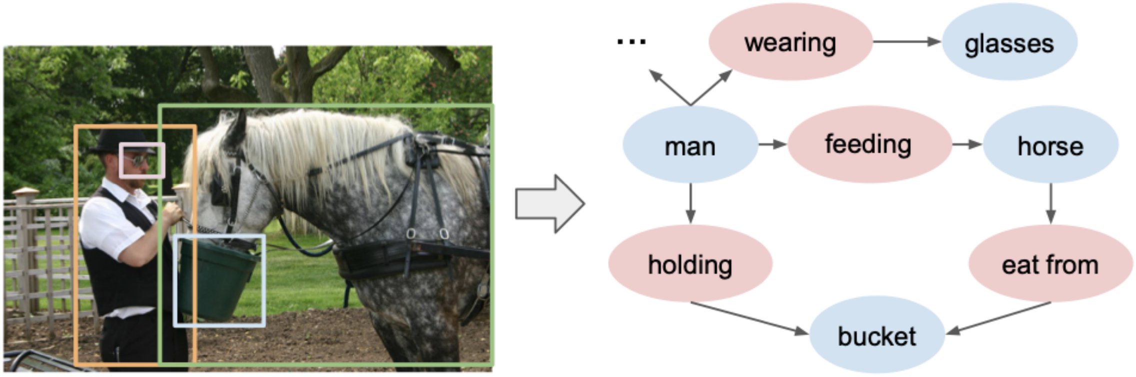


Fig. 2.11 A visual scene graph example [Xu et al., 2017].

2.6.1 Graph based contextual attention

Graph based contextual attention usually combines visual scene graph generation [Johnson et al., 2015] or visual relation detection [Xu et al., 2017] with the graph convolutional neural networks. It builds a partly connected graph, in which each node only connects to other related nodes based on some pre-defined semantic or spatial relationships. Given a region pair r_i and r_j from a region detection model, the visual relation detection model first detects whether they are related based on their appearance and relative positions (at the moment, the region pair are usually selected in a union box). If the two regions are related, the model then predicts their relationships (r_{ij}):

$$r_{ij} = R(r_i, r_j) \tag{2.20}$$

For example, in Figure 2.11, a man (r_i) is wearing (r_{ij}) a glasses (r_j), while the glasses has no semantic relation with the bucket. After traversing all region pairs, the visual scene graph can be built, in which each region is connected with other related regions. In the second stage, each region's feature is contextualized by graph neural network with their related regions and the corresponding relationships:

$$\widehat{r_i} = \frac{1}{|\mathcal{N}_i|} \sum_{k \in \mathcal{N}_i} g(r_i, r_k, r_{ik}) \tag{2.21}$$

where g is the graph convolutional neural network, $\mathcal{N}_i$ is the set of all related regions with region r_i, $|\mathcal{N}_i|$ is the total number of regions in set $\mathcal{N}_i$.

With graph convolution, it is clear that the contextualized feature for each region has already encoded its contextual information inside it. However, there are two limitations in graph based contextual attention. Firstly, the graph is built on some pre-trained visual scene graph model with limited pre-defined relationships. It may not be optimal for the downstream tasks. Secondly, it is computationally inefficient. It requires auxiliary model (scene graph models) to build the graph, and usually has two streams for semantic graph and spatial graph, respectively. In chapter 5, we will alleviate those limitations by the usage of self-attention.

2.6.2 Self-attention based contextual attention

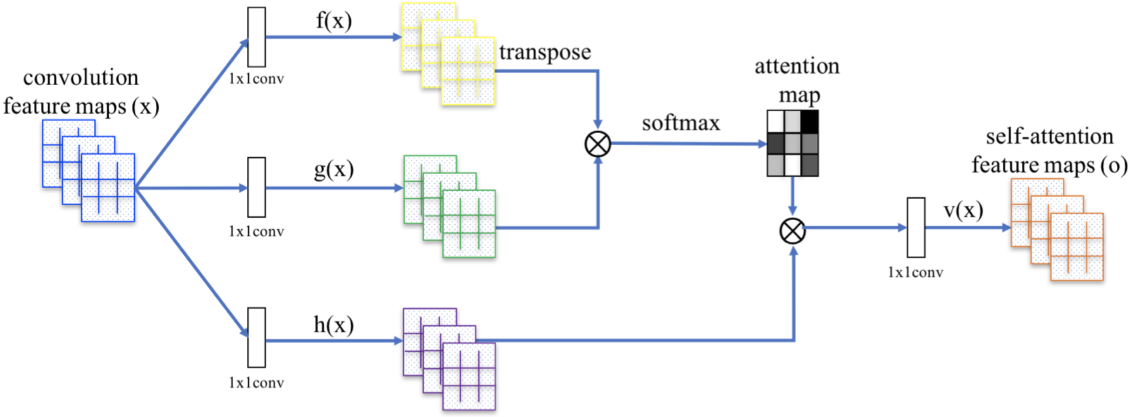


Fig. 2.12 Example for spatial contextual attention [Zhang et al., 2018].

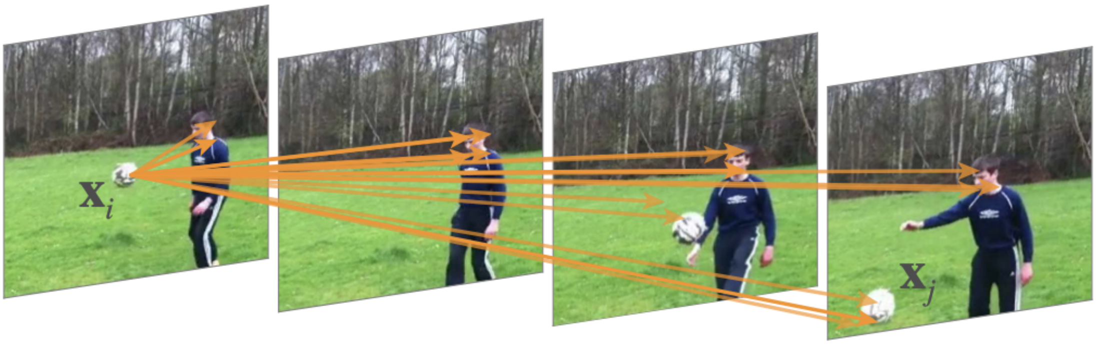


Fig. 2.13 Example for temporal contextual attention [Wang et al., 2018].

In contrast to graph based contextual attention, self-attention based contextual attention does not require pre-defined relationships and visual relationship detectors. The graph built with self-attention based contextual attention is a fully connected graph. And the edge's weight between two nodes is determined by their feature's dot product attention. Concretely, given a query region r_i and a key region r_j, the edge's weight connecting them is computed as:

$$r_{ij} = (r_i w_q^T)(w_k^T r_j) \tag{2.22}$$

where, w_q and w_k are linear transformations. The contextualization process is then done through:

$$\begin{aligned}\widehat{r_i} &= \sum_{k=0}^{N-1} \alpha_{ik} r_k w_v^T \\ \text{where } \alpha_{ik} &= \frac{\exp^{r_{ik}}}{\sum_{j=0}^{N-1} \exp^{r_{ij}}}\end{aligned} \tag{2.23}$$

where w_v is a linear transformation.

Self-attention based contextual attention has been widely used in many computer vision tasks. For example, the self-attention generative network (SGAN [Zhang et al., 2018], Figure 2.12) apply self-attention spatially to contextualize all receptive fields in the generated images for a better quality of the generated images. Non-local neural network [Wang et al., 2018] (Figure 2.13) applies self-attention temporally to connect the temporal correlated regions for the motion representation. Self-attention can also be applied in semantic segmentation [Fu et al., 2019].

Compared to graph based contextual attention, self-attention based contextual attention is more computationally efficient, as it does not require auxiliary models to detect the visual relationships and can be trained end-to-end. However, most self-attention based contextual attention adopts the inner architecture designed for natural language processing, regardless the difference between the semantic unit's structure between texts and images, i.e., a sentence is a 1-D signal while an image represents 3-D scene in 2-D spaces. In chapter 5, we will propose an self-attention based contextual attention module, with inner structure adapted to 2-D image space.

Chapter 3

Bottom-Up Attention—Understanding Deep Learning for Saliency Prediction

In computer vision, bottom-up attention is studied under the umbrella of saliency prediction. Since 2012, data-driven deep saliency models have surpassed classical saliency models, as demonstrated by the results on datasets such as the MIT300 [Bylinskii et al., 2012] and SALICON [Jiang et al., 2015]. Yet, there still remains a large gap between the performance of these models and the inter-human baseline. Some outstanding questions include what have these models learned, how and where they fail, and how they can be improved.

This chapter attempts to answer these questions by analyzing the representations learned by individual neurons located at the intermediate layers of deep saliency models. To this end, we follow the steps of existing deep saliency models, that is borrowing a pre-trained model of object recognition to encode the visual features and learning a decoder to infer the saliency. We consider two cases when the encoder is used as a fixed feature extractor and when it is fine-tuned, and compare the inner representations of the network. To study how the learned representations depend on the task, we fine-tune the same network using the same image set but for two different tasks: saliency prediction versus scene classification.

Our analyses reveal that: 1) some visual salient regions (e.g. head, text, symbol, and vehicle) are already encoded within various layers of the network pre-trained for object recognition, 2) using modern datasets, we find that fine-tuning pre-trained models for saliency prediction makes them favor some categories (e.g., head) over some others (e.g., text), 3) although deep models for saliency prediction outperform classical models on natural images, the converse is true for synthetic stimuli (e.g., pop-out search arrays), an evidence of significant difference between human and data-driven saliency models, and 4) we confirm that, after fine-tuning, the change in inner-representations is mostly due to the task and not the domain shift in the data.

3.1 Introduction

> "*What a model pre-trained for object recognition has encoded for visual saliency? What is changed after fine-tuning a pre-trained model for saliency prediction? What are the limitations of deep saliency models? What drives the inner representations' change during fine-tuning?*"

We have observed a surge in the development of data-driven models of saliency prediction under the umbrella of deep learning and large scale datasets. Such deep models have demonstrated significant performance improvements in comparison to classical models, which are based on hand-crafted features or psychological assumptions, outperforming them on most benchmarks. However, while there still remains a relatively large gap between deep models and the human visual system (see Table 3.1, deep models dominated the saliency prediction benchmark), the performance of deep models appears to have reached a ceiling. This raises the question of what is learned by deep models that drives their superior performance over classical models, and what are the remaining and missing ingredients to attain human-like performance. Internal representations of deep object recognition models have been visualized and analyzed extensively in recent years. Such efforts, however, are missing for saliency models and it is unclear how saliency models do what they do.

Model	**Backbone**	**Fine tuning**	**NSS**
Deep gaze II [Kümmerer et al., 2017]	VGG-19	×	2.34
SAM [Cornia et al., 2018]	ResNet-50/VGG-16	√	2.34/2.30
Deepfix [Kruthiventi et al., 2017]	VGG-16	√	2.26
SALICON [Huang et al., 2015]	VGG-16	√	2.12
PDP [Jetley et al., 2016]	VGG-16	√	2.05
Human IO	-	-	3.29

Table 3.1 Five state-of-the-art deep saliency models and their *NSS* scores on the MIT300 saliency benchmark [Bylinskii et al., 2012].

In this chapter, to better understand and push forward data-driven based saliency prediction models, we shed light on what is learned by deep saliency models by analyzing their internal representations:

• We train a simple yet effective deep saliency prediction model and annotate 3 datasets for analyzing the relationship between the deep model's inner representations and the visual saliency in the image.

• We propose a new dataset based on synthetic pop-out search arrays to compare deep and classical saliency models.

• We investigate what and how saliency information is encoded in a pre-trained deep model and look into the effect of fine-tuning on inner-representations of the deep saliency models.

• Finally, we study the effect of the task type on the inner representations of a deep model by comparing a model fine-tuned for saliency prediction with a model fine-tuned for scene recognition.

The rest of this chapter is organized as follows: section 3.2 will give the background of deep models for saliency prediction and the interpretation of deep models; section 3.3 will introduce the data we used for model training and inner representation analysis; section 3.4 will introduce how we train and analyse the deep saliency models; sections 3.5-3.7 will illustrates the analytical results for the deep saliency model followed by the conclusion in section 3.8.

3.2 Background

3.2.1 Deep saliency models

The SALICON challenge [Jiang et al., 2015] provided the first large scale dataset for saliency prediction. Although it is collected through mouse clicking, which is not accurate for model evaluation, it indeed has facilitated and pushed forward the development of deep saliency models [Tavakoli et al., 2017a]. Several such models learn a mapping from deep feature space to the saliency space, where a pre-trained object recognition network acts as the feature encoder which is then fine-tuned for the saliency task. For example, DeepNet [Pan et al., 2016] learns saliency using 8 convolutional layers, where only the first 3 layers are initialized from a pre-trained image classification model. It is believed that the features from shallow layers of a pre-trained CNN can only encode local non-semantic features (e.g. edge) [Zhou et al., 2015]. DeepNet thus cannot benefit from pre-trained semantic features encoded in the deeper layers, which are important for saliency prediction (see our analysis in the later sections). PDP [Jetley et al., 2016] treats the saliency map as a small scale probability map, and investigates different loss functions for saliency prediction. Their results suggest that Bhattacharyya distance is the best loss function for saliency prediction. This results, however, were only empirically confirmed and compared with other few weak loss functions (e.g. L2 loss) without any theoretical support. The SALICON [Huang et al., 2015] model uses multi-resolution inputs, and combines feature representations in the deep layers for saliency prediction. Although significant advancements have been made in SALICON, it is computationally inefficient with two resolution streams. Deepfix [Kruthiventi et al., 2017] combines deep architectures of VGG-16, GoogleNet [Szegedy et al., 2015], and Dilated convolutions [Yu and Koltun, 2015] in a network and adds a central bias, to achieve a higher performance than previous models. However, their model, without residual connection, suffers from gradient vanishing or explosion. SalGAN [Pan et al., 2017] uses an encoder-decoder architecture and proposes the binary cross entropy (BCE) loss function to perform pixel-wise (rather than image-wise) saliency estimation. After pre-training the encoder-decoder, it uses a Generative Adversarial Network (GAN) [Goodfellow et al., 2014] to boost performance. The idea to use GAN is somewhat novel in saliency prediction. Its benefit, however, is not significant and it requires more training time. DVA [Wang and Shen, 2018] uses multiple layer's representations, builds a decoder for each layer, and fuses them at the final stage for pixel-wise saliency prediction. Although multi level features are used in DVA, they still fail for synthetic images (see our analysis in the later sections), which indicates DVA does not make full use of features from different levels. SAM [Cornia et al., 2018] uses an attention module and a LSTM [Hochreiter and Schmidhuber, 1997] network to attend to different salient regions in the image. The use of LSTM requires multi-step refinement for the model's output, which significantly increase the model's inference time. DeepGaze II [Kümmerer et al., 2017] uses the features at different layers of a pre-trained deep model and combines them with the prior knowledge (i.e. center-bias). As their model is not trained end-to-end, the features used in their model are not optimal for saliency prediction. DSCLRCN [Liu and Han, 2018] uses multiple inputs by adding a contextual information stream, and concatenates the original representation and the contextual representation into a LSTM network for the final prediction. Again, with two stream inputs, DSLRCN is computationally inefficient.

As we can see from Table 3.1, state-of-the-art saliency prediction models have reached a ceiling. Instead of propose a new model for saliency prediction, this chapter tries to analyse what deep saliency models can/cannot learn, and what stop them to advance. Those analytical results can be a guideline for future research in saliency prediction.

3.2.2 Visualizing deep neural networks

The success of deep convolutional neural networks has raised the question of what representations are learned by neurons located in intermediate and deep layers. One approach towards understanding how CNNs work and learn is to visualize individual neurons' activations and receptive fields. Zeiler and Fergus [2014] proposed a deconvolution network in order to visualize the original patterns that activate the corresponding activation maps. A deconvolution network consists of three steps, namely, unpooling, transposed convolution, and ReLU operation. Some qualitative results in those method are attractive, it, however, cannot be scaled to systematic analysis. Yosinski et al. [2015] developed two tools for understanding deep convolutional neural networks. The first one is designed to visualize the activation maps at different layers for a given input image. The second tool aims to estimate the input pattern that a network is maximally attuned to for a given object class. In practice, the last layer of a deep neural network typically consists of one neuron per object class. They proposed to use gradient ascent (with regularization) to find the input image that maximizes the output of a specific neuron in regard to a specific object class. Hence, they derive the optimum input that appeals to the network for a specific class. Both tools can provide some surprising and valid patterns, but none of them can be interpreted quantitatively.

Both visualization methods discussed above are essentially qualitative. In contrast, Bau et al. [2017] proposed a quantitative method to give each activation map a *semantic meaning*. In their work, they proposed a dataset with 6 image categories and contains 63,305 images for network dissection, where each image is labeled with pixel-wise semantic meaning. At first, they forward all images in the dataset into a pre-trained deep model. For each activation map inside the model, different inputs have different patterns. Then, they compute the distribution of each unit activation map over the whole dataset, and determine a threshold for each unit based on its activation distribution. With the threshold for each unit, the activation map for each input image is quantized to a binary map. Finally, they compute the intersection over union (IOU) between the quantized activation map and the labeled ground truth to determine what objects or object parts a unit is detecting. This method can somehow be scaled for systematic analysis for the inner representation of deep neural networks. However, it is constrained by the annotated data and classes.

The aforementioned approaches provide useful insight into the internals of deep neural networks trained on ImageNet for the classification task. However, our understanding of the internal representations of deep saliency prediction models is somewhat limited. Bylinskii et al. [2016] tried to understand deep models for saliency prediction. But their study was mostly focused on where models fail, rather than how they compute saliency. To the best of our knowledge, the work in this chapter is the first to study the representations learned by deep saliency models[1].

3.3 Data and Annotation

We first introduce the data used in our experiments as well as our proposed annotated dataset.

[1] The codes and data can be found at https://github.com/SenHe/uavdvsm

SALICON: SALICON [Jiang et al., 2015] is the largest database for saliency prediction at the moment. It contains $10,000$ training images, $5,000$ validation images and $5,000$ testing images. The eye-fixations in this dataset were approximated by the mouse clicking. Each observer was given 5s to explore the image freely by moving the mouse cursor to anywhere they wanted to look in the displayed image. All pre-processed mouse samples for the same image were then aggregated and blurred with a Gaussian filter to generate a saliency map, same as the common practice to generate the fixation maps from eye-tracking data [Xu et al., 2014]. Here, we use it to fine-tune a pre-trained model for saliency prediction.

OSIE-SR: This acronym stands for the “Objects and Semantic Images and Eye-tracking Saliency Re-annotated" (annotations of salient regions). The original OSIE dataset [Xu et al., 2014] has 700 images with rich semantics. It contains eye movements of 15 subjects for each image recorded during free-viewing, and also has the annotated masks for objects in the image according to 12 attributes. Those eye movements are recorded by eye-trackers. For our analysis, we extract clusters of fixation locations, called *salient regions*. We then manually annotate each salient region as belonging to one of the 12 saliency categories, including: `person head`, `person part`, `animal head`, `animal part`, `object`, `text`, `symbol`, `vehicle`, `food`, `drink`, `plant`, and `other`. Similar categories have been exploited in previous researches [Bylinskii et al., 2016; Xu et al., 2014]. In the annotation process, we first add all eye-fixation points into the image. Then for each eye-fixation cluster, we draw a polygon mask and also assign a label to each mask. Isolated eye-fixation or clusters with only few fixation points were ignored (please see the annotation steps in Figure 3.1). Figure 3.2 provides an example where each annotated region has a label according to its salient category. The re-annotated data is used to measure the association between inner representations (activation maps) in the deep model (both pre-trained for object detection and fine-tuned for saliency prediction) and each salient category.

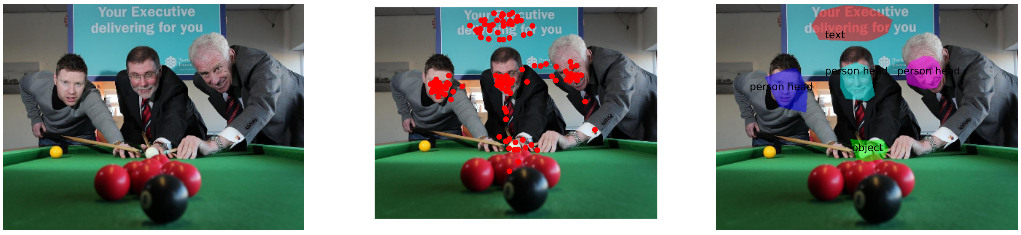


Fig. 3.1 Example of data annotation for OSIE-SR dataset. From left to right: original image, image with eye-fixation points, image with annotation masks

Synthetic Images: We selected 80 synthetic search arrays from Healey and Enns [2012], which are often used in pop-out experiments (Figure 3.3). This dataset contains various

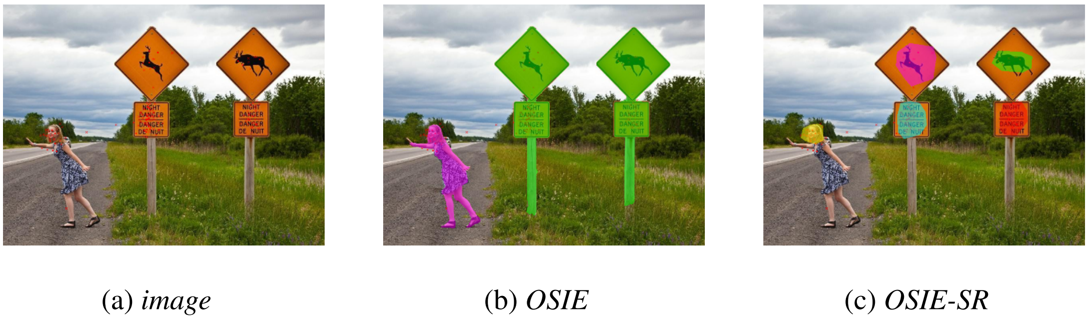


Fig. 3.2 (a) An example image from the OSIE dataset, (b) OSIE annotation, and (c) the re-annotated OSIE-SR labels.

pop-out patterns where a target stands out from the rest of the items in terms of color, orientation, density, curvature, etc. We provide mask annotation for the salient (i.e., pop-out) region in each image. This database is used to compare deep models and classical models on their ability to detect targets that pop-out in simple scenes. We also use it to study inner representations of deep models over synthetic patterns.

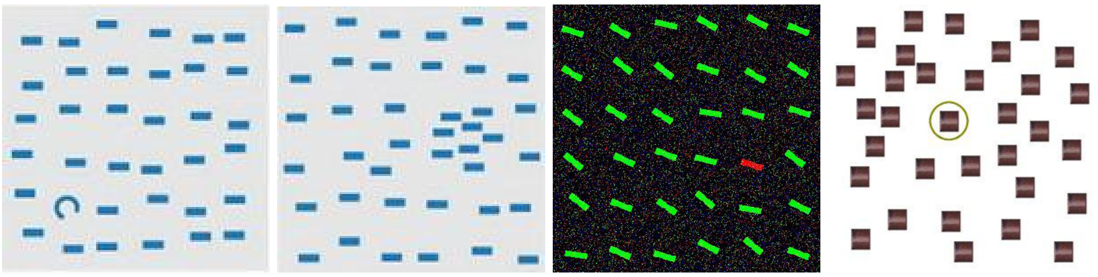


Fig. 3.3 Example synthetic pop out search arrays

SALICON-SAL-SCE: We select a subset of images from the SALICON dataset and annotate each image with a scene category label based on the categories of the Place-CNN dataset [Zhou et al., 2017]. We removed images belonging to the scene categories with fewer than 50 images. Eventually, we are left with 6,107 images with both fixation maps and scene category labels (26 categories in total, they are: `restaurant`, `river`, `airfield`, `pizzeria`, `park`, `hotel room`, `beach`, `bedroom`, `parking lot`, `dining room`, `baseball field`, `train station`, `bazaar(outdoor)`, `crosswalk`, `bus station`, `kitchen`, `conference room`, `office`, `toilet`, `street`, `televison room`, `farm land`, `living room`, `athletic field`, `bathroom`, `ski resort`). We use this database to compare the effect of saliency prediction and scene recognition on learned inner representations.

3.4 Methods

3.4.1 Saliency model

For our analysis, we develop a saliency model using the convolutional part of VGG-16 (conv1-1 to conv5-3) and add a simple 1×1 convolutional layer on top of the conv5-3 layer for saliency map regression. The model has a single resolution with input of size 224×224 and is optimized with $-NSS$ as the loss function. We consider 2 setups to analyze the representations.

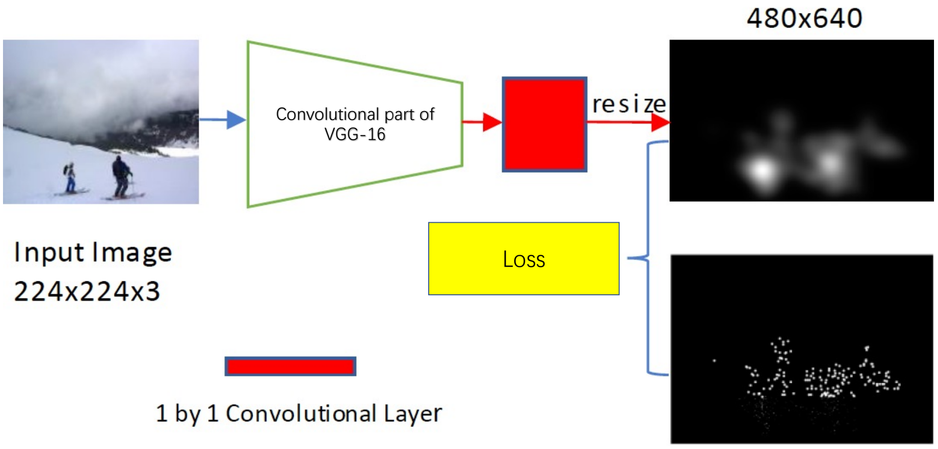


Fig. 3.4 Saliency prediction model's architecture.

• **Setup I:** We first look into the inner representations without fine-tuning the VGG part, i.e., we *only* learn the last 1×1 convolution layer, motivated by the performance of some of the existing models that achieve state-of-the-art performance without fine-tuning. In other words, we analyze what types of saliency information exist in the pre-trained VGG model and how it is distributed within different layers. In this setup, we analyze conv4-1 to conv5-3 layers, which correspond to the last two blocks in VGG-16. The activation maps from layers below conv4-1 are sensitive to edge-like patterns and do not correspond to annotated regions. We thus do not include them.

• **Setup II:** We then look into the fine-tuned model. In this setup, we learn the last 1×1 convolution layer and also fine-tune the VGG part of the model for different number of layers (each time from scratch) and examine the responses of neurons in the conv5-3 layer.

3.4.2 Modified NSS score

We propose using normalized scanpath score (NSS) within salient regions as a method to interpret the inner representations. We thus can look into the association between the activation maps and the salient regions of the image for analyzing the inner representations of the deep visual saliency models. To implement this, we first forward each image into the deep model and extract the activation maps from different layers in the model. Then, the association between each activation map (A) and each salient region (R) in the image is computed as:

$$f(A_{ij}, R_{lk}) = \mathrm{NSS}(A_{ij}, F_l \cdot M_{lk}) \tag{3.1}$$

where A_{ij} is the j_{th} activation map from the i_{th} layer for the input l_{th} image, and R_{lk} is the k_{th} salient region in the l_{th} image. F_l is the fixation on the l_{th} image, and M_{lk} is the annotated polygon mask in l_{th} image for region k. It is worth noting that all activation maps were normalized and reshaped to the size of the input image. More specifically, The normalization process of each acivation map is done by:

$$A_{ij} = \frac{A_{ij} - \mu(A_{ij})}{\sigma(A_{ij})} \tag{3.2}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ are mean and standard deviation operators, respectively.

3.4.3 Local saliency statistics

In the OSIE-SR dataset, each annotated region corresponds to one saliency category (has one label). To compute the statistics for saliency category c in each layer, we compute the mean value of the top 10 activation maps with high mean NSS values in Equation (3.1) for all the regions of category c. We also compute the number of activation maps whose mean NSS value is above a threshold (T) in Equation (3.1) for all regions for category c. For the computation of T, we use the mean of mean NSS for top 10 activation maps for all categories in all layers. If a neuron has a mean NSS higher than this value for a certain class, it indicates that this neuron is activated for that salient category.

3.5 Analysis of Learned Representations

How does fine-tuning the VGG neurons for saliency prediction affect the network inner representation? To answer this question, we train two saliency models, one keeping the

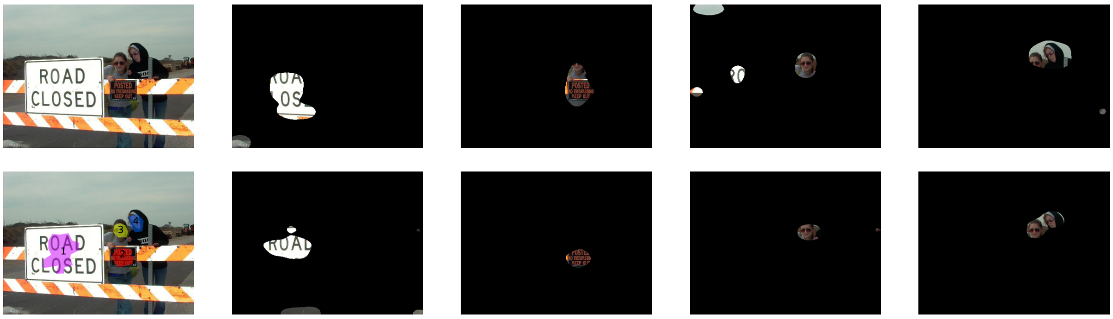


Fig. 3.5 An example of the pre-trained model inner representation. Top row: the original image, and the activation maps with highest activation at salient regions from layer conv5-1. Bottom row: the image with masked salient regions (1,2,3,4) and the activation maps that best respond to the salient regions of conv4-3.

CNN features fixed during the training and the other one fine-tuning the CNN features in conjunction with the 1 × 1 convolution layer for saliency prediction (corresponding to the two setups mentioned in section 3.4.1).

3.5.1 Saliency representation before fine-tuning

layer	*person head*	*person part*	*animal head*	*animal part*	*object*	*text*	*symbol*	*vehicle*	*food*	*plant*	*drink*	*other*
	mean NSS for top 10 activation maps											
conv5-3	1.21 ± 0.32	0.98 ± 0.23	2.04 ± 0.21	1.6 ± 0.15	0.69 ± 0.13	1.1 ± 0.38	1.07 ± 0.22	1.53 ± 0.34	1.25 ± 0.16	1.21 ± 0.64	1.48 ± 0.34	0.57 ± 0.13
conv5-2	2.57± 0.38	1.64 ± 0.12	2.89 ± 0.43	**1.91 ± 0.19**	1.07 ± 0.07	1.59 ± 0.17	1.91 ± 0.16	2.18 ± 0.19	1.59 ± 0.14	**2.05 ± 0.35**	**2.13 ±0.26**	0.92 ± 0.10
conv5-1	2.95± 0.41	1.58 ± 0.13	2.6 ± 0.18	1.67 ± 0.24	1.16 ± 0.18	1.91 ± 0.56	2.06 ± 0.31	2.48 ± 0.5§	**1.8 ± 0.23**	1.96 ± 0.34	1.85 ± 0.23	**1 ± 0.20**
conv4-3	**3.46 ± 0.74**	1.67 ± 0.13	**2.93 ± 0.58**	1.5 ± 0.14	**1.27 ± 0.12**	**2.4 ± 0.50**	**2.37 ± 0.49**	**2.71 ± 0.27**	1.54 ± 0.27	1.62± 0.32	1.88± 0.31	0.91± 0.14
conv4-2	2.85± 0.34	**1.78 ± 0.17**	2.63 ± 0.40	1.39 ± 0.22	1.19± 0.18	2.04± 0.34	2.05±0.23	2.33± 0.48	1.48± 0.24	1.59± 0.13	1.53± 0.19	0.74± 0.08
conv4-1	2.08 ± 0.33	1.56±0.17	1.89± 0.26	1.18±0.11	1.11± 0.09	2.17± 0.41	1.99± 0.32	1.93± 0.20	1.38±0.14	1.58±0.27	1.33± 0.17	0.72±0.10
	# activation maps above threshold ($T = 2.15$)											
conv5-3	0	0	2	0	0	0	0	1	0	1	0	0
conv5-2	13	0	**19**	**2**	0	0	1	4	0	2	**3**	0
conv5-1	12	0	12	0	0	2	4	7	**1**	**3**	1	0
conv4-3	**19**	0	14	0	0	**6**	**5**	**15**	0	1	2	0
conv4-2	14	0	8	0	0	2	3	4	0	0	0	0
conv4-1	3	0	1	0	0	4	3	2	0	1	0	0

Table 3.2 Inner representations in the pre-trained visual saliency model (training the 1 × 1 convolution layer) at different layers for all types of saliency categories (Please see text for details).

How the deep representations (trained on image classification) relate to the salient regions in the image? Table 3.2 depicts the statistics of the inner representations within different layers of the deep visual saliency model, where the convolution part has not been fine-tuned (setup I). From Table 3.2, we can see that many visual saliency categories, including: `person head`, `animal head`, `text`, `symbol`, `vehicle` and `drink`, have been encoded in the pre-trained CNN features. We observe not only high mean NSS scores, but also a large number of active response maps for each saliency category. We also observe that the visual saliency information is encoded within various layers, e.g. `person head`, `animal head`, and `text` are present in conv4-3. Figure. 3.5 visualizes some examples of the activation maps in the model without fine-tuning the VGG features. As depicted, there is a high association between salient regions and activation maps pre-trained for image classification.

3.5.2 Saliency representation after fine-tuning

# layers tuned	*person head*	*person part*	*animal head*	*animal part*	*object*	*text*	*symbol*	*vehicle*	*food*	*plant*	*drink*	*other*
	mean NSS for top 10 activation maps											
0	1.21±0.32	0.98±0.23	2.04±0.21	1.6±0.15	0.69±0.13	1.1±0.38	1.07±0.22	1.53±0.34	1.25±0.16	1.21±0.64	1.48±0.34	0.57 ±0.13
1	2.61±0.28	1.63±0.10	2.44±0.29	1.7±0.16	**1.31±0.12**	1.17±0.09	1.56 ±0.22	**2.21±0.27**	1.97±0.25	**1.89±0.22**	**1.99±0.27**	1.09±0.18
2	3.25±0.23	1.77±0.12	2.75±0.16	1.78 ±0.08	1.25±0.09	1.32±0.09	1.56±0.07	1.91±0.13	1.89±0.17	1.66±0.13	1.65±0.07	1.12 ±0.10
3	3.28±0.21	1.78±0.04	3.04±0.14	1.79±0.03	1.28±0.05	1.34±0.03	1.57±0.04	1.94±0.04	1.96±0.06	1.81±0.04	1.68±0.12	**1.2 ±0.05**
4	**3.38±0.16**	**1.83±0.03**	**3.08±0.15**	1.78±0.02	1.24±0.05	1.25±0.03	1.47±0.01	1.97±0.02	1.87±0.04	1.73±0.02	1.63±0.06	1.12±0.04
5	3.32 ±0.13	1.78±0.01	2.84 ±0.15	**1.82± 0.02**	1.28±0.02	**1.38±0.02**	**1.59±0.03**	2.1±0.06	1.95±0.03	1.75 ±0.03	1.77 ±0.06	1.17 ±0.02
6	3.08 ±0.10	1.83 ±0.03	2.57± 0.11	1.78±0.01	1.25±0.01	1.25±0.02	1.45± 0.03	2.03±0.05	**1.99±0.04**	1.8±0.03	1.66±0.04	1.11 ±0.02
all	2.85±0.17	1.75±0.05	2.36±0.13	1.65±0.05	1.14 ±0.06	1.2±0.05	1.41±0.03	1.72±0.06	1.84±0.07	1.71±0.04	1.33±0.05	1.07±0.03
	# activation maps above threshold ($T = 1.97$)											
0	1	0	4	0	0	0	0	1	0	1	1	0
1	11	0	12	**1**	0	0	**1**	8	5	**4**	**4**	0
2	30	0	26	0	0	0	0	4	5	0	0	0
3	39	0	49	0	0	0	0	5	5	0	0	0
4	48	0	56	0	0	0	0	8	0	0	0	0
5	**56**	0	**68**	0	0	0	0	**23**	3	0	0	0
6	45	0	49	0	0	0	0	12	**6**	0	0	0
all	21	0	17	0	0	0	0	0	0	0	0	0

Table 3.3 Inner representations in the last convolutional layer (conv5-3) before and after fine-tuning for all types of saliency categories. 0 in number of fine tuned layers indicate setup I, otherwise setup II.

How does the deep representations change after fine-tuning with saliency prediction? Table 3.3 reports the statistics of visual saliency in layer conv5-3, which is directly used for saliency prediction, after fine-tuning different number of layers in the model (0 indicates that only the final 1×1 convolutional layer above the pre-trained model for regression was trained on saliency data, without fine-tuning the pre-trained model; 1 means fine-tuning the layer conv5-3, 2 means fine-tuning the layers conv5-3 and conv5-2, and so on).

From Table 3.3, we can see that after fine-tuning, the activation maps became more selective to visual saliency, as the mean NSS values improve for all saliency categories. The improvements are, however, uneven as can be seen in Fig. 3.6. Some categories improve more than other categories. For example, *person head* improves the most from 1.21 to 3.38 when fine-tuning 4 layers (see Figure 3.7), while for *text*, the improvement is relatively small (at most from 1.1 to 1.38 when fine-tuning 5 layers). Similar effects can be seen in Figure 3.8, where activation maps which are most selective to texture regions in the image after fine-tuning still respond to *head* regions in the image, and with high activation values.

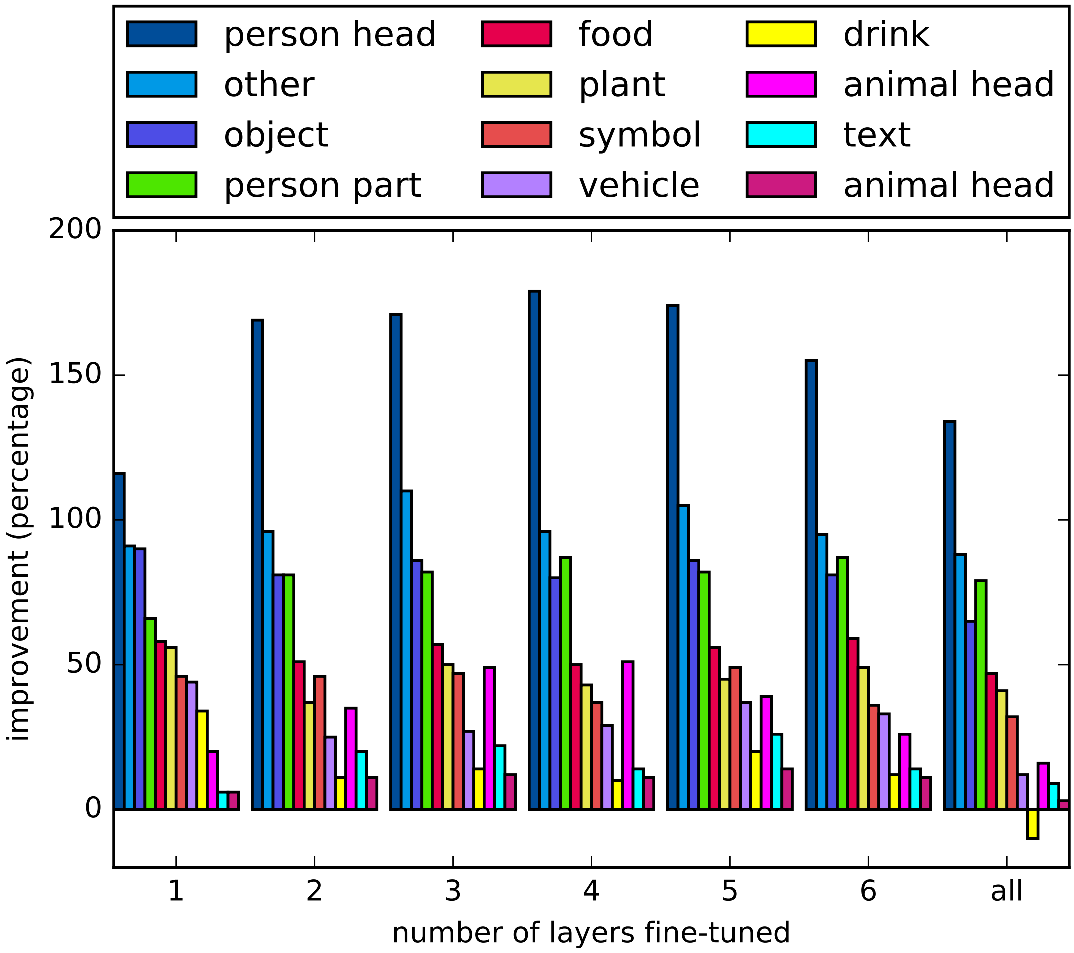


Fig. 3.6 Top 10 mean NSS improvements for each category when fine-tuning different numbers of layers

An ANOVA [Cuevas et al., 2004] test and multiple comparisons indicate `person head`, `animal head`, and `other` are significantly different than all other categories and each other.

An interesting observation is that by fine-tuning more layers, i.e., more than 3 layers, the mean NSS scores and the number of activation maps start to decrease for some saliency categories—this is esepcially true when fine-tuning *all* layers. The model also does not gain more saliency prediction improvement by fine-tuning more layers (see section 3.5.3). We speculate that one reason behind this observation might be the quality and quantity of the current saliency data, which is biased towards specific salient objects and regions.

How the representation change in other layers (the layers that are not directly used for saliency prediction)? Figure 3.9 and 3.10 provide the statistics for `person head` and `text` in layers conv4-1 to conv5-2 after fine-tuning different number of layers. For `person head`,

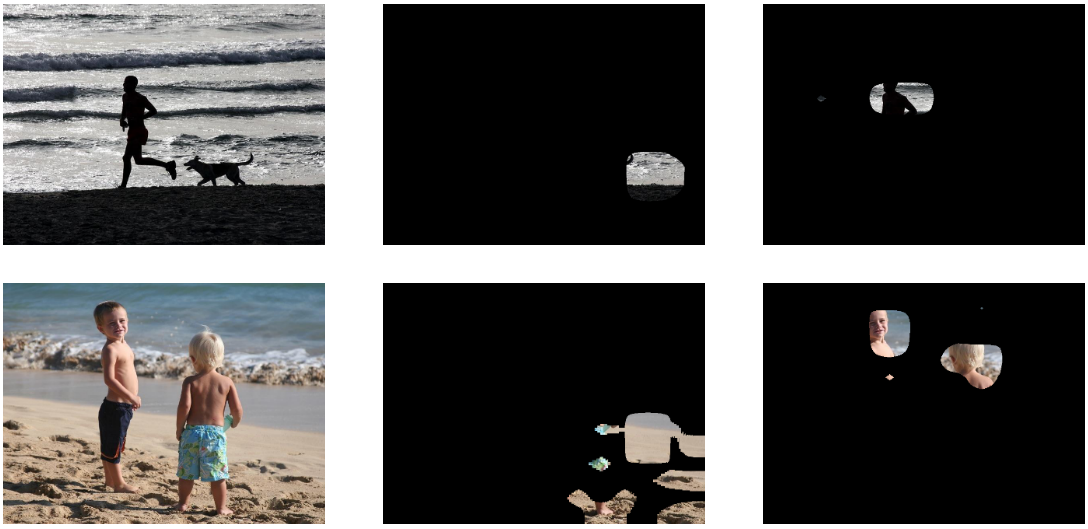


Fig. 3.7 The 288th activation map in layer conv5-3 before and after fine-tuning for an input image. Notice that it becomes attuned to *person head* after fine-tuning.

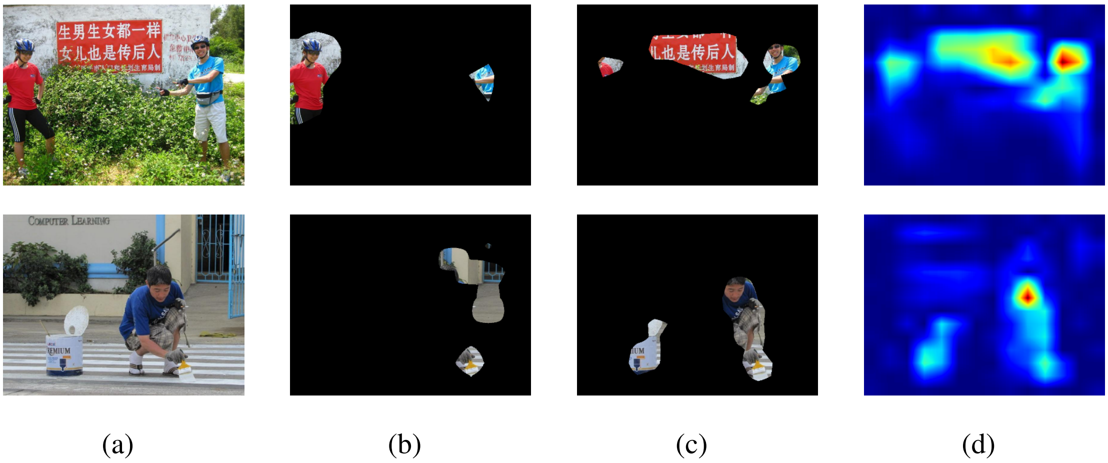


Fig. 3.8 The 512th activation map in layer conv5-3 before and after fine-tuning (tuning 3 layers). (a) the input image, image overlapped with the activation map before fine-tuning (b), after fine-tuning (c), and the activation map after fine-tuning (d). After fine-tuning, despite the fact that this activation map has the highest mean NSS score for all regions annotated as *text* in the dataset, it still favors *heads*.

the maximum mean value is improved (before fine-tuning, it is in layer conv4-3, with mean value of 3.46), but for `text`, it is decreased.

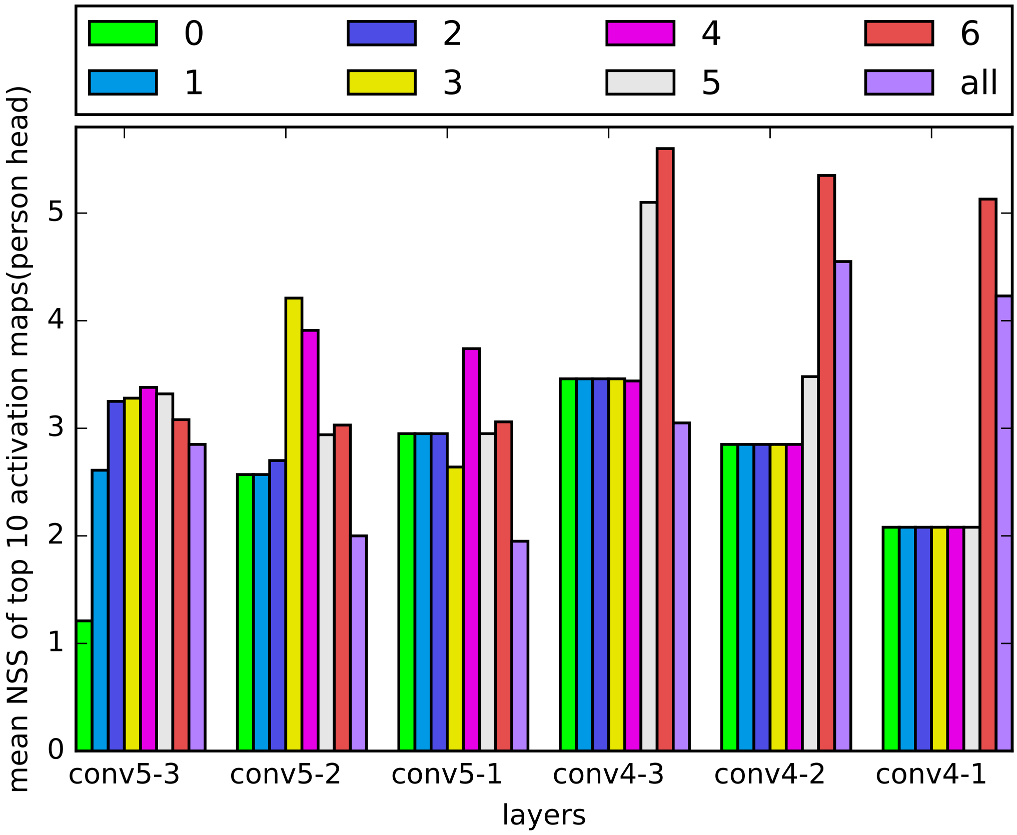


Fig. 3.9 The change of representation in different layers after fine-tuning different number of layers for `person head`.

3.5.3 Model's performance after fine-tuning

We compare our trained saliency model (trained with setup II), which was used for analyzing the inner representations, with state-of-the-art saliency prediction models, including, Deep gaze II [Kümmerer et al., 2017], SAM [Cornia et al., 2018], SalGAN [Pan et al., 2017]. It is worth noting that Deep gaze II and SAM were both trained on SALICON and fine-tuned on MIT1003 dataset, containing real fixation data, and they also add centre-bias in their model.

We report the results when fine-tuning different layers of the CNN trunk. While the architecture of our model is simple, the results in Table 3.4 shows that its performance is comparable to the state-of-the-art. Importantly, as depicted in Table 3.4, we can see that fine-tuning more layers does not always yield improvements in performance; fine-tuning only three layers performs best.

Figure 3.11 provides some example predictions that shows that almost all the saliency models predict similar saliency maps and the difference lies on small nuance details.

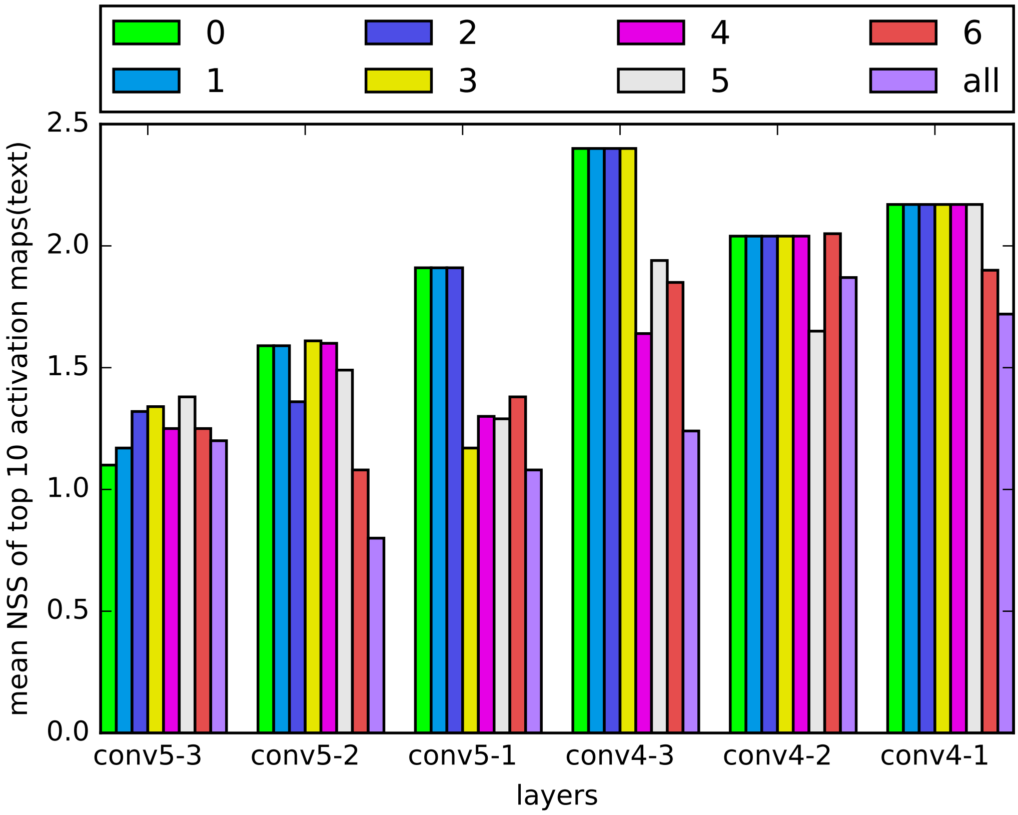
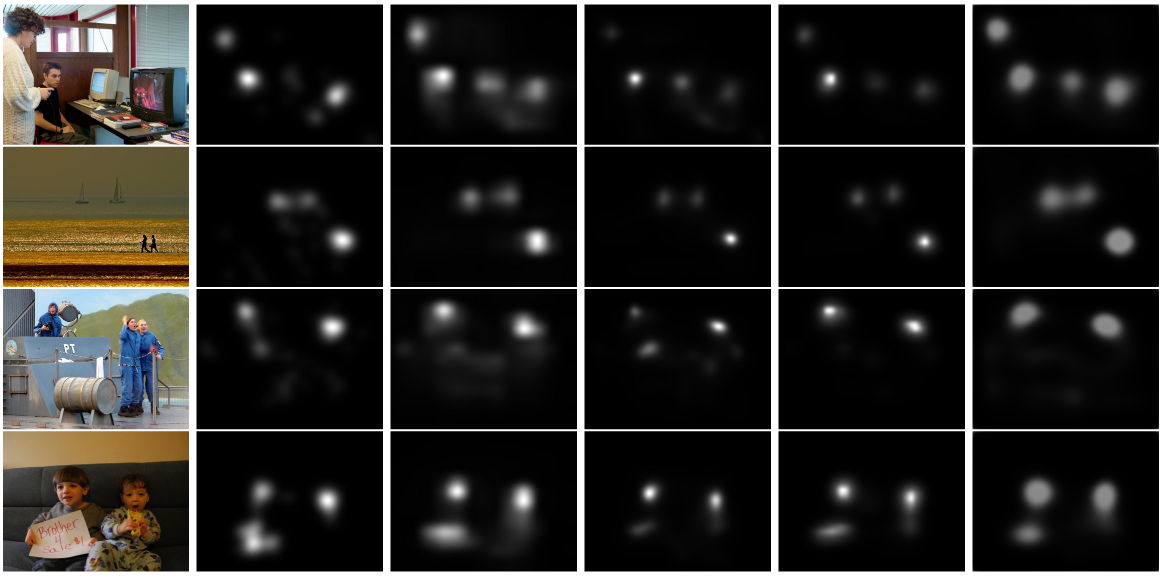


Fig. 3.10 The change of representation in different layers after fine-tuning different number of layers for `text`.

	our models number of layers fine-tuned								other models		
	0	*1*	*2*	*3*	*4*	*5*	*6*	*all*	*Deep gaze II*	*SAM*	*SalGAN*
NSS ↑	1.56	2.49	2.60	**2.67**	2.63	2.64	2.61	2.51	**3.20**	3.10	2.47
AUC-Judd ↑	0.83	0.89	0.90	**0.90**	0.90	0.90	0.90	0.89	**0.91**	0.90	0.89
KL ↓	1.10	0.61	0.58	**0.53**	0.56	0.55	0.57	0.60	0.95	1.37	**0.77**

Table 3.4 Model's performance on OSIE dataset for our model, when fine-tuning different number of layers: 0 means fine-tuning 0 layers, just train the 1 by 1 convolutional layer, model 1 means train the 1 by 1 convolutional layer, and fine-tuning layer conv5-3 in the VGG-16, model 2 means train the 1 by 1 convolutional layer, and fine-tuning layer conv5-3 and conv5-2 in the VGG-16, and so on



(a) image (b) Ground truth (c) Our model (d) DeepGaze II (e) SAM (f) SalGAN

Fig. 3.11 Some qualitative comparisons between our model (our results showed here are from the model with three fine-tuned layers) and other state-of-the-art deep saliency prediction models on OSIE dataset.

3.5.4 Inner representations in different backbones

	person head	*person part*	*animal head*	*animal part*	*object*	*text*	*symbol*	*vehicle*	*food*	*plant*	*drink*	*other*
	mean NSS for top 10 activation maps											
before	0.78 ± 0.04	0.83 ± 0.07	0.90± 0.07	1.03 ± 0.12	0.85 ± 0.07	0.77 ± 0.04	1.01±0.13	0.99±0.07	0.80± 0.08	1.07± 0.08	1.12 ± 0.07	0.88 ± 0.05
after	2.44 ± 0.02	2.51±0.01	2.34±0.01	2.40±0.02	2.38±0.01	2.52±0.01	2.27±0.01	2.49±0.02	2.29±0.01	2.38±0.01	2.57±0.03	2.48± 0.01
	# activation maps above threshold ($T = 2.42$)											
before	0	0	0	0	0	0	0	0	0	0	0	0
after	252	252	252	250	252	252	245	251	243	249	254	252

Table 3.5 Inner representations, ResNet-18 as backbone, before and after finetuning for saliency prediction.

In the previous section, we showed the inner representations before and after fine-tuning in the VGG backbone. What are the differences in inner representations when using other backbones? For example, ResNet-18 [He et al., 2016] backbone with residual connection. To study this, we looked into ResNet-18, which has skip connections. We follow setup II to see how the representations change after fine-tuning. We replace the original VGG-16 backbone with the ResNet-18 backbone and compare the differences between them. The quantitative results are summarized in Table 3.5. In ResNet-18, more activation maps start to fire on salient regions after fine-tuning from image recognition to saliency prediction. ANOVA test and multiple comparisons indicate both models favour some categories significantly than

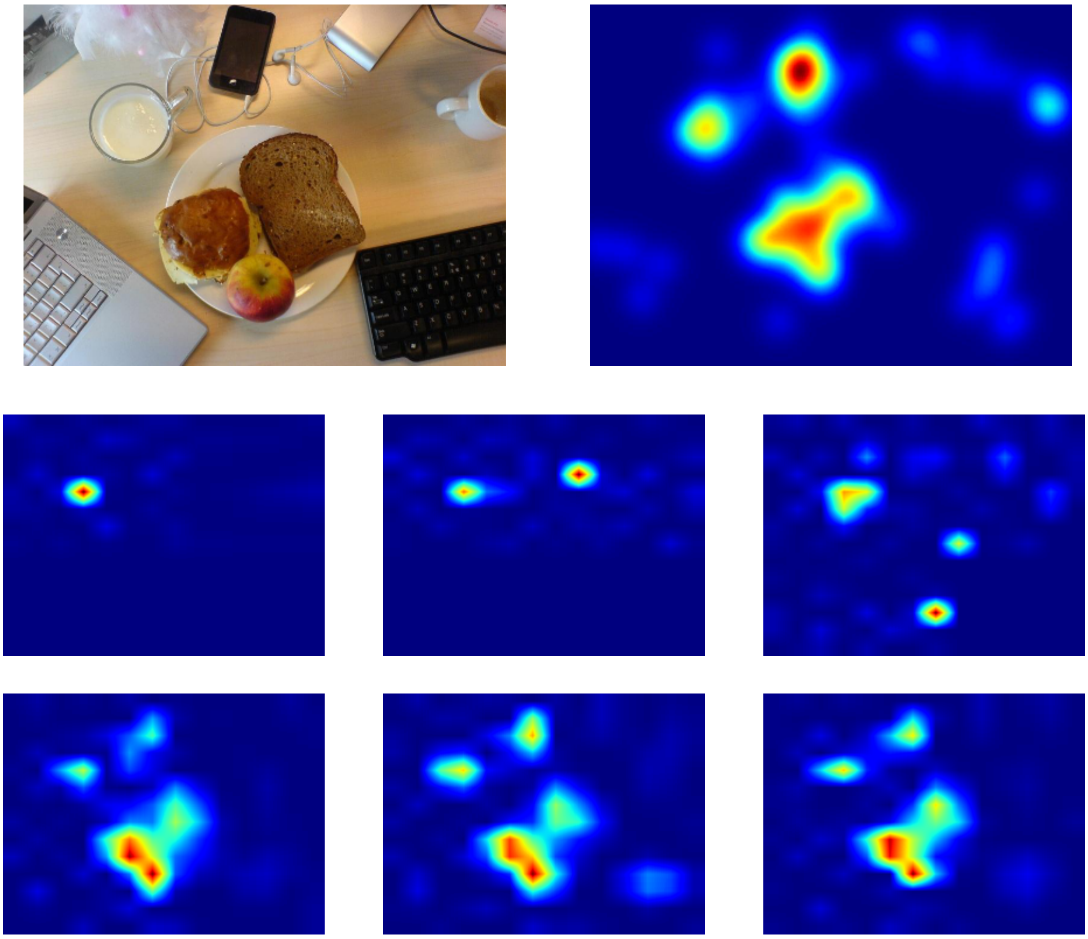


Fig. 3.12 Top row: example image and ground truth saliency map; Middle row: three activation maps from different neurons in the last convolutional layer of VGG-16 after fine-tuning for the example image; Bottom row: three activation maps from different neurons in the last convolutional layer of ResNet-18 after fine-tuning for the example image.

other categories. In VGG-16, `person head`, `animal head`, and `other` are significantly different than all other categories and each other, while for ResNet-18, `drink`, `person head`, `animal head` are significantly different than all others and each other. VGG-16 prefers `person head`, while ResNet-18 prefers `drink`.

Another interesting result is that more activation maps activated on the salient regions in the images in ResNet-18, as showed in Table 3.5. Different activation maps in ResNet-18 attend to the same and most salient regions in the image. While in VGG-16, different activation maps attend to different and few salient regions in the image. A qualitative example is illustrated in Figure 3.12.

3.5.5 The Relationship between Intermediate and Final Representations

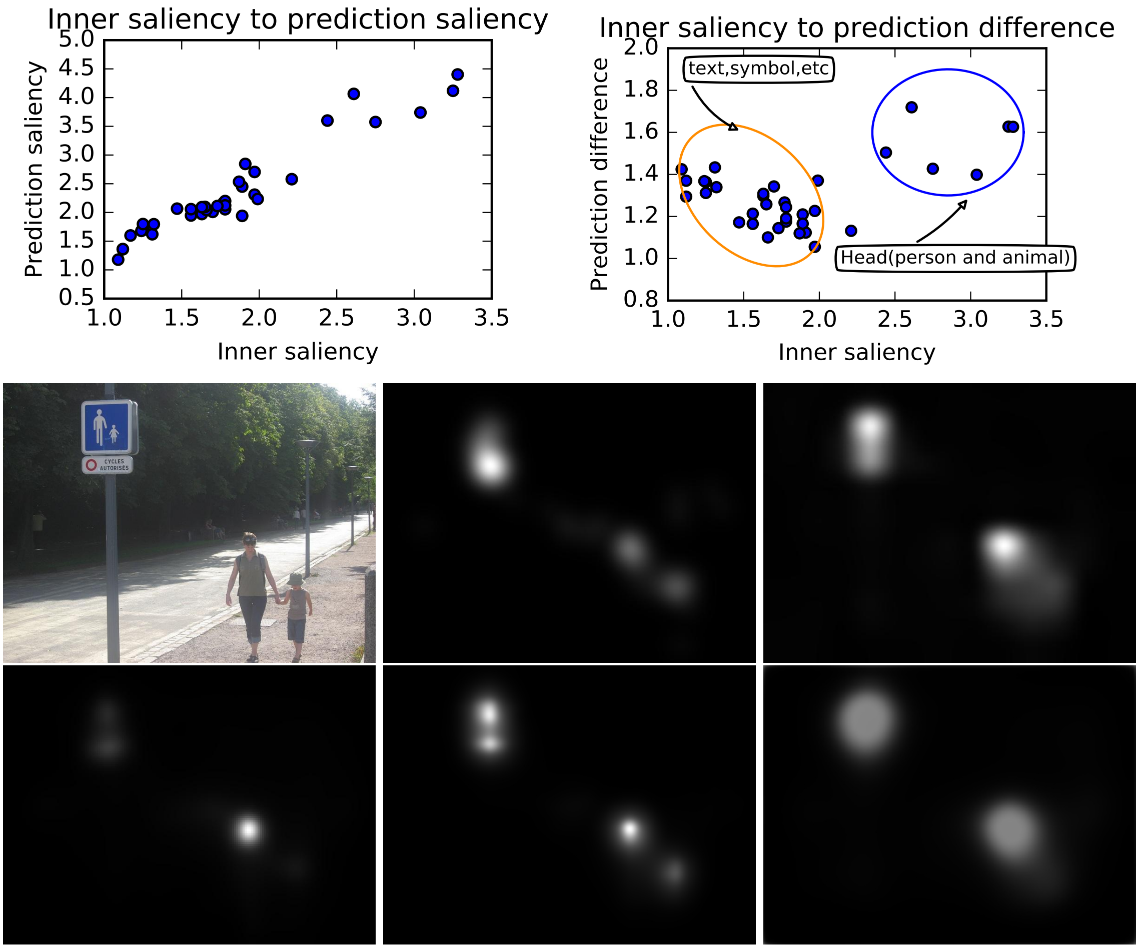


Fig. 3.13 Top row: the relationship between the inner representation saliency (inner saliency, x-axis, adopted from table 3.3 for different salient categories) and output saliency (prediction saliency, y-axis in the left figure, defined in equation 3.3), and the output prediction difference (prediction difference, y-axis in the right figure, defined in equation 3.4). Middle row: an example image, ground truth saliency map, and model prediction. Bottom row: the model prediction from Deep Gaze II, SAM and SalGAN for the example image.

What is the relation between a model's inner representation and its output? In other words, are the categories that are more salient in the inner representations also more active in the output saliency map? To what extend the salient categories within the inner representations agree with the ground truth salient categories? To answer these questions, we define the inner saliency for each category as the mean NSS value of the top 10 activation maps for it, which is the same as the values in Table 3.3. Following steps of Tavakoli et al. [2017a] for fine-grained contextual analysis, we assign the output saliency of a model, defined as OS_c for category c, as the output's mean NSS score of all salient regions in the OSIE-SR dataset that belong to category c,

$$OS_c = \frac{1}{A} \sum_{i=1}^{N} \sum_{j=1}^{M} \mathrm{NSS}(p_i, m_{i,c_j} \cdot f_i) \tag{3.3}$$

where, A is the total number of salient regions for category c. N is the total number of images. M is total number of regions for category c in i_{th} image. p_i is the model prediction for i_{th} image. m_{i,c_j} is the annotated mask for the j_{th} region in i_{th} image for category c (if category c exists in the image) and f_i is the fixation location on the i_{th} image.

To measure the relation between salient categories and representations, we define the concept of output salient category difference, denoted as OD_c. It measures the mean difference of the NSS score between a model prediction and the ground truth with respect to the salient categories,

$$OD_c = \frac{1}{A} \sum_{i=1}^{A} \sum_{j=1}^{M} |f_{c_j}(p_i) - f_{c_j}(G_i)| \tag{3.4}$$

where, $f_{c_j}(pred_i) = \mathrm{NSS}(p_i, m_{i,c_j} \cdot f_i)$, and G_i is the ground truth saliency map.

The results are shown in Figure. 3.13. As depicted in the top left of this figure, the model's saliency output is correlated to the saliency of inner representations. In other words, we can see that if a category is more salient in the model's inner representation, it is also more salient in the model's output (Spearman's correlation coefficient: $r_s = 0.96$). The salient categories in inner representations and output, however, migh be different. The output salience category difference is lower for less salient inner categories and higher for more salient inner ones. For example, as depicted in top right panel of Figure. 3.13, the *head* category has more salient inner representations and higher output salient category difference (ANOVA test with measurements and clusters as factors showing significant difference $p = 9e^{-9} < 0.05$). In other words, the model has learned to fire on faces, irrespective of whether they are salient in the context of the given image. It is worth noting that this is not a special case for our trained model, as we can see from the bottom row in Figure 3.13, other state-of-the-art models also give more attention to the person head region than the traffic sign.

3.6 Model Performance and Representations over Synthetic Search Arrays

The previous section analysed the deep saliency models for natural images. How do deep saliency models perform on synthetic images? We compare the performance of deep and classical saliency models on a set of synthetic images. These images are designed to simulate the feature pop-out, and have been extensively used to study human attention [Treisman and Gelade, 1980], but they have not been considered for evaluating deep saliency models. There exists no eye-fixation on such images, but it is easy to locate the target/salient item in the array (in fact we manually labeled the images). We assess the models' performance using Normalized Mean value under the annotated Mask (NMM) for each image:

$$NMM_i = \text{mean}_{m_i}(\frac{p_i - \mu(p_i)}{\sigma(p_i)}) \tag{3.5}$$

where m_i and p_i are the annotated mask and model prediction for the i_{th} image, respectively.

Deep models			Classical models	
DG	SAM	DVA	GBV	BMS
1.66	1.25	1.19	2.57	3.65

Table 3.6 Performance comparison between deep models and classical models on synthetic images in terms of NMM.

conv5-3	*conv5-2*	*conv5-1*	*conv4-3*	*conv4-2*	*conv4-1*	*conv3-3*	*conv3-2*	*conv3-1*	*conv2-2*	*conv2-1*	*conv1-2*	*conv1-1*
0.38 ± 0.18	0.94 ±0.22	1.47 ± 0.51	1.96 ± 0.55	1.94 ± 0.40	**2.12 ± 0.34**	1.54 ± 0.48	1.57 ±0.50	1.57 ± 0.27	1.27±0.31	1.50± 0.55	1.24± 0.50	1.46±0.48

Table 3.7 The mean NMM for top 10 activation maps in each layer (from conv1-1 to conv5-3) in the pre-trained model for synthetic images.

# layers fine-tuned							
0	1	2	3	4	5	6	all
0.38 ± 0.18	0.8 ± 0.17	0.98± 0.27	0.92±0.08	0.65± 0.08	0.86± 0.05	0.83±0.09	0.73±0.07

Table 3.8 The mean NMM score for top 10 activation maps in the last convolutional layer for synthetic images, when fine-tuning different number of layers.

Table 3.6 shows the performance of deep and classic models on the synthetic images. Surprisingly, although deep models achieve state-of-the-art performance on the MIT300 benchmark, they completely fail on synthetic images and are outclassed by classical models.

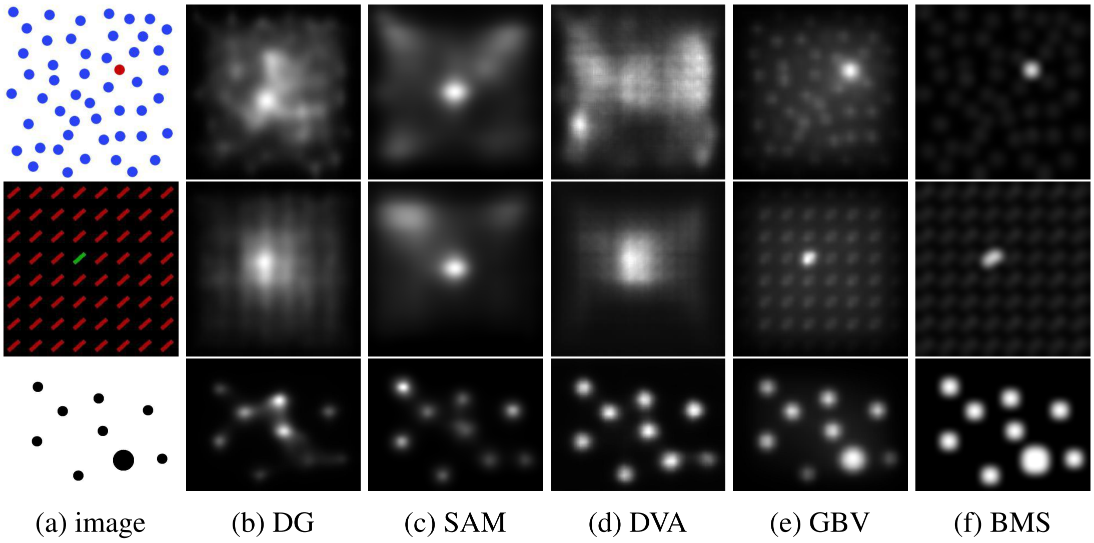


Fig. 3.14 Some qualitative examples for deep and classical models on synthetic images, from left to right: image, predictions from DeepGaze II, SAM, DVA, GBVS, and BMS.

Moreover, despite the DVA [Wang and Shen, 2018] has connections between shallow layer and output layer in their model, their performance is still not competitive on synthetic images. Figure 3.14 illustrates saliency predictions on example stimuli by the considered methods.

To investigate the reason behind this failure, we looked into the effect of fine-tuning on the inner representations and neuron responses to synthetic patterns. The results in Table 3.7 show that deep models indeed capture the salient patterns within the *middle layers* of the architecture (e.g conv4-1 layer). Some examples (including curvature, orientation, etc) are shown in Figure 3.15. Nevertheless, as indicated in Table 3.8, no matter how many layers are fine-tuned, the output of the deep saliency model never highlights such salient patterns. One possible reason might be that the current large databases, e.g., SALICON, are biased towards natural scenes containing daily objects (text, faces, animals, cars, etc) and do not include any image similar to synthetic patterns. The pop out patterns, thus, are ignored in higher representations as it is not relevant in natural saliency .

3.7 The Influence of Task on the Learned Representations

What is the driving cause for the observed change in representations after fine-tuning in previous sections? Is it due to the network being fine-tuned to a new task (saliency prediction) or the network being fine-tuned to a different set of data (images from a saliency prediction dataset)? To figure out, we compare two tasks of saliency prediction and scene recognition on SALICON-SAL-SCE, which provides saliency information and scene type labels. Note

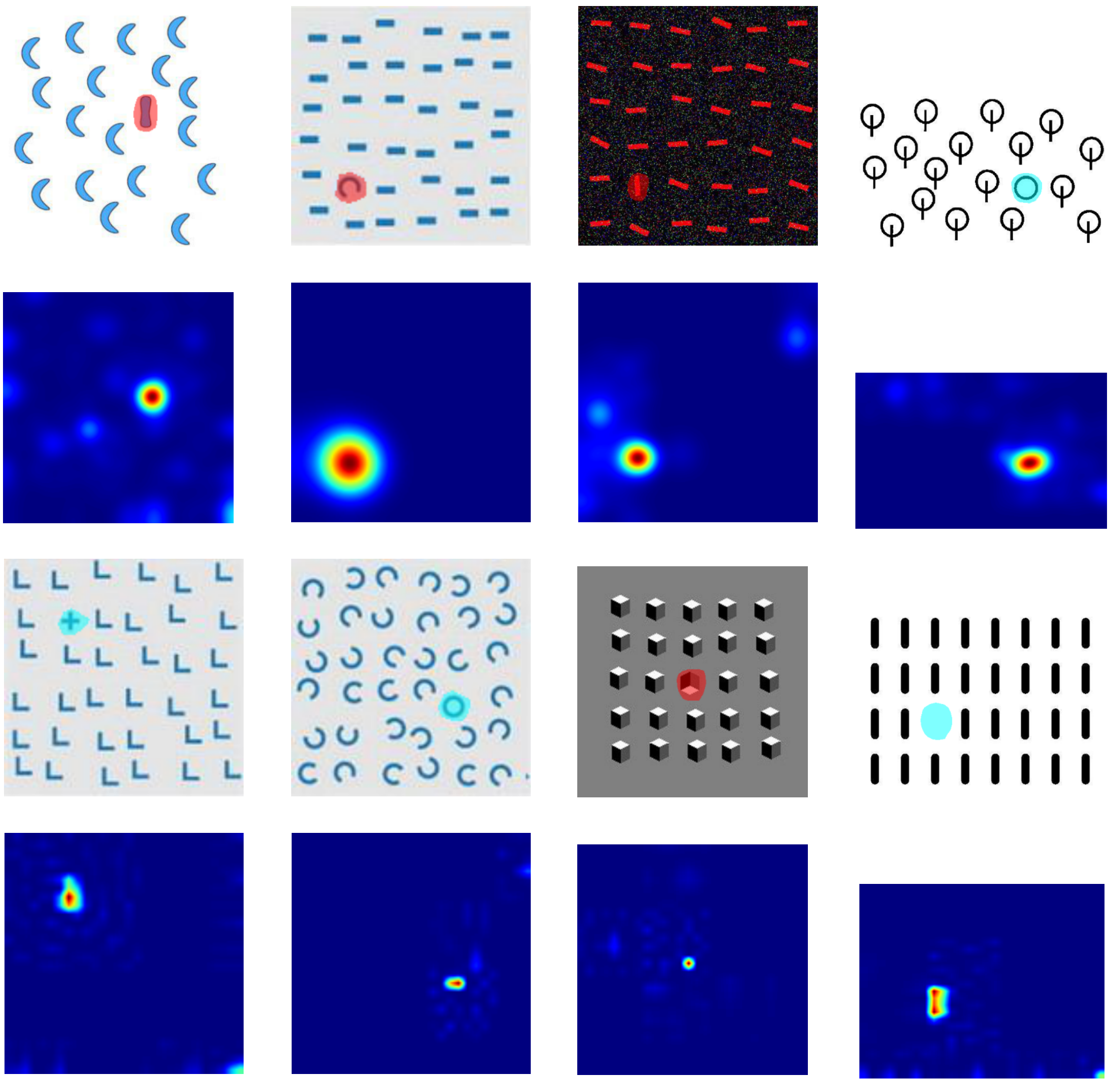


Fig. 3.15 In every two rows, the top row are synthetic search arrays with their annotated masks and the bottom row are the activation maps from layer conv4-1 that best correlated with the masked regions in the synthetic images.

that in both cases, the images used for fine-tuning are the same. Therefore if the observed shift in representations is only due to the data, the inner representations should be similar in both tasks; conversely, if the task is what is driving the change, the representations should differ.

We check three CNN trunks, including, 1) a pre-traind CNN based on VGG network for object recognition (pt), 2) a CNN fine-tuned for saliency prediction (sp), and 3) a CNN fine-tuned for scene recognition (sr). The saliency prediction is the same as explained above. The scene recognition network consists of the VGG convolutional part and 3 fully connected

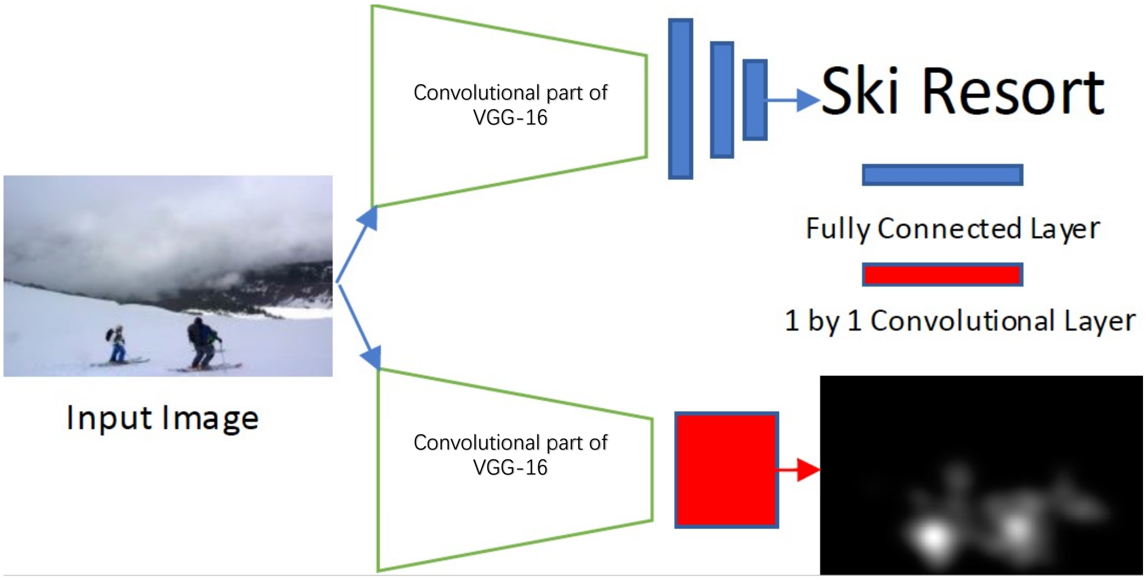

Convolutional part of VGG-16
Ski Resort
Fully Connected Layer
1 by 1 Convolutional Layer
Input Image
Convolutional part of VGG-16

Fig. 3.16 Our architecture to study task-dependency of representations (saliency prediction vs. scene classification). We use the same data to fine-tune the pre-trained model for different tasks.

layers (see Figure 3.16). Due to the data limitation, we only fine-tuned 1 layer of the pre-trained model (layer conv5-3) for both tasks. Weight balance[2] was used when fine-tuning the scene recognition model to compensate imbalanced categories.

	person head	*person part*	*animal head*	*animal part*	*object*	*text*	*symbol*	*vehicle*	*food*	*plant*	*drink*	*other*
	mean NSS for top 10 activation maps											
pt	1.21±0.32	0.98±0.23	2.04±0.21	1.6±0.15	0.69±0.13	1.1±0.38	1.07 ±0.22	1.53±0.34	1.25±0.16	1.21±0.64	1.48±0.34	0.57±0.13
sp	2.51±0.18	1.6 ±0.17	2.32±0.18	1.7±0.12	1.3±0.14	1.44±0.13	1.74±0.16	2.29±0.31	1.86±0.14	1.58±0.14	2.01 ±0.22	1.13±0.09
sr	0.47±0.17	0.57±0.14	1.54±0.25	1.24±0.28	0.57±0.12	0.83±0.21	0.61±0.14	0.88 ±0.25	0.91±0.13	0.97±0.41	1.28±0.36	0.41±0.13
	# activation maps above threshold ($T = 1.79$)											
pt	1	0	11	1	0	1	0	2	0	1	3	0
sp	27	1	17	3	0	0	4	13	6	2	8	0
sr	0	0	1	1	0	0	0	0	0	1	2	0

Table 3.9 Inner representation for different tasks (saliency prediction and scene recognition) before and after fine-tuning (pt: pre-traind CNN based on VGG network scene recognition, sp: CNN fine-tuned for saliency prediction, sr: CNN fine-tuned for scene recognition).

Table 3.9 shows how the inner representation changes with respect to saliency categories, once the CNN is fine-tuned for different tasks using the same data. The results show that fine-tuning for saliency prediction drives the inner representations to became more selective

[2] We use the inverse of the percentage of each class in the dataset to weight the cross entropy loss of each class during training.

to salient categories, while fine-tuning for scene recognition leads to less selectivity to salient categories and inhibition of some other salient regions. Examples of the activation map change for each task is provided in Figure 3.17.

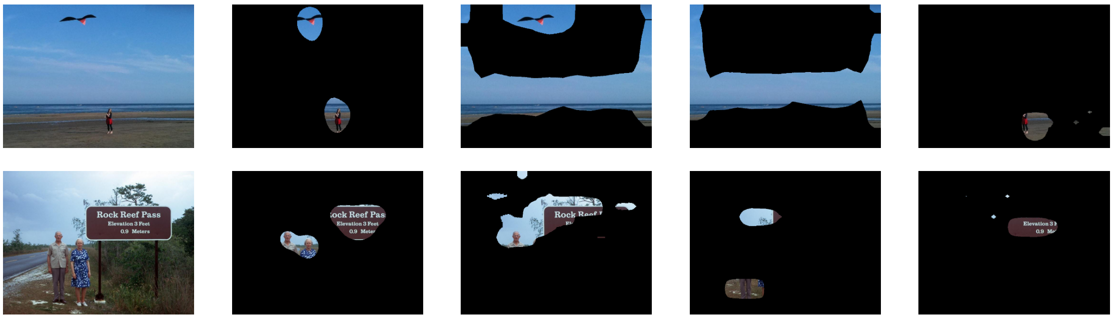


Fig. 3.17 From left to right: Original image, image overlapped with the ground-truth fixation map, overlapped with the activation map by the pre-trained model, overlapped with the activation map after fine-tuning for scene recognition, overlapped with the activation map after fine-tuning for saliency prediction.

correct prediction	wrong prediction
0.12	0.11

Table 3.10 The NSS scores of mean activation maps for correct and wrong prediction in scene recognition task.

To what degree the inner representations in the scene recognition network is consistent with human attention? Does the scene recognition model attend to the locations a human may find salient? We investigate this by computing the NSS score between the attention of model (the mean of 512 activation maps in layer conv5-3) and the human fixation on the image. The results are summarized in Table 3.10, showing that the NSS score is small irrespective of whether the model's prediction is correct or not. In other words, the model's attention in scene recognition is different from human attention in free-viewing.

To summarize, the above results indicate that the inner representations, during fine-tuning the same CNN for the two different tasks of saliency prediction and scene recognition on the same data, mostly change because of the task and not the data.

3.8 Discussion and Conclusion

In this chapter, we analyzed the internals of deep saliency models. To this end, we annotated 3 datasets and conducted several experiments to unveil the secrets of deep saliency models. Our analysis on this data revealed that a deep neural network pre-trained for image recognition already encodes some visual saliency in the image. Fine-tuning this pre-trained model for saliency prediction produces a model with uneven response to saliency categories, e.g neurons sensitive to textual input start attending more to human head. We showed that although deep models do capture synthetic pop-out stimuli within their inner layers, they fail to predict such salient patterns in their output, contrary to classical models of saliency prediction. We also confirmed that the observed change in the inner representations after fine-tuning is mainly due to fine-tuning for the *task* and not the *data*. In our study, fine-tuning the model for saliency prediction resulted in more selective responses to salient regions, though uneven. On the other hand, fine-tuning the model for scene recognition had inhibitory effect and the inner representations were losing their selectivity to some of the existing salient patterns.

To conclude, pushing the development of better data-driven deep visual saliency models further may require delicate attention to the diversity of salient categories within images. In other words, we may need not only a large scale dataset, but also a dataset with rich saliency categories to ensure generalization.

Chapter 4

Top-Down Attention—Human Attention under Image Captioning

Chapter 3 studied bottom-up attention, which can rapidly capture the gist of the scene by finding out a set of task-agnostic salient regions. Top-down attention, which is complementary to the former one, seeks to find the most relevant information related to a task. For example, to recognize a person, people usually focus on the face of the person regardless his or her eye-catching shoes. In this chapter, we study the top-down attention in computer vision. We present a novel dataset[1] consists of eye movements and verbal descriptions recorded synchronously over images. Using this data, we study the differences in human attention during free-viewing and image captioning tasks. We look into the relationship between human attention and language constructs during perception and sentence articulation. We also analyze attention deployment mechanisms in the top-down soft attention approach that is argued to mimic human attention in captioning tasks, and investigate whether visual saliency can help image captioning. Our study in this chapter reveals that (1) human attention behaviour differs in free-viewing and image description tasks. Humans tend to fixate on a greater variety of regions under the latter task, (2) there is a strong relationship between described objects and attended objects (97% of the described objects are being attended), (3) a convolutional neural network as feature encoder accounts for human-attended regions during image captioning to a great extent (around 78%), (4) soft-attention mechanism differs from human attention, both spatially and temporally. And there is low correlation between caption scores and attention consistency scores. These indicate a large gap between humans and machines in regards to top-down attention, and (5) by integrating the soft attention model with image saliency, we can significantly improve the model's performance on Flickr30k and MSCOCO benchmarks.

[1] The dataset can be found at: https://github.com/SenHe/Human-Attention-in-Image-Captioning

4.1 Introduction

"What is the difference of humans' attention in free viewing and image captioning task? How humans use their attention to do image captioning task? What is the difference between the machine learned soft-attention and humans' attention for image captioning? How to use humans' attention to help machine in image captioning?"

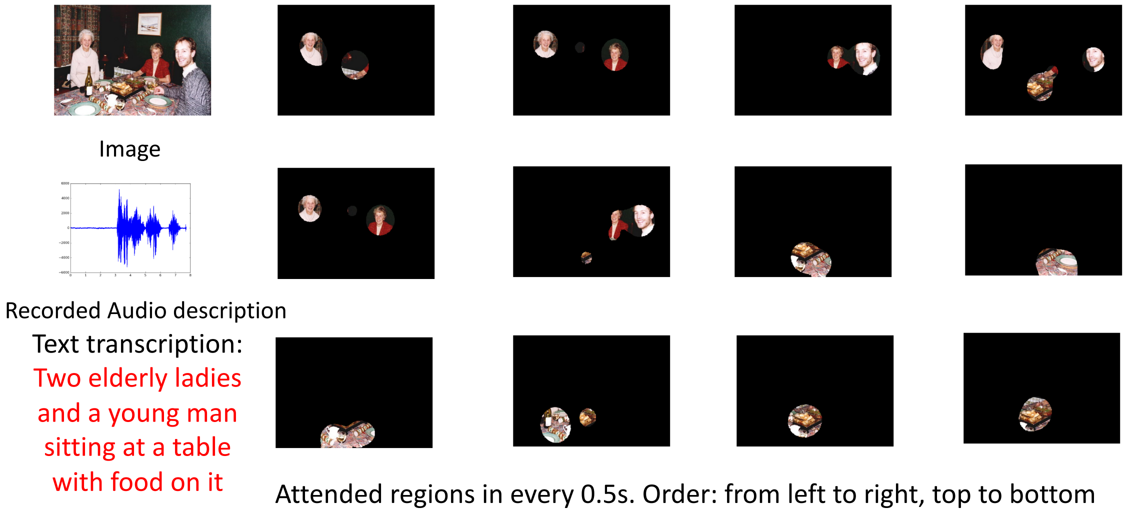


Image
Recorded Audio description
Text transcription:
Two elderly ladies and a young man sitting at a table with food on it
Attended regions in every 0.5s. Order: from left to right, top to bottom

Fig. 4.1 An example of the data collected in our dataset, including the image shown to the subject, the subject's audio description for this image, the textual transcription of this description, and the sequence of attended regions while the subject watched and described the image. Each mask is generated by thresholding a saliency map, which is generated by convolving the binary fixation map in that 0.5s with a Gaussian kernel.

> "Two elderly ladies and a young man sitting at a table with food on it."

This sentence is an example of how someone would describe the image in Figure 4.1. Describing images in a few words and extracting the gist of the scene while ignoring unnecessary details is an easy task for humans, that can in some cases be achieved from only a very brief glance [Thorpe et al., 1996; VanRullen and Thorpe, 2001]. In a stark contrast, providing a formal algorithm for the same task is an intricate challenge that has been beyond the reach of computer vision for decades. Recently, with the availability of powerful deep neural network architectures and large scale datasets, new data-driven approaches have been proposed for automatic captioning of images and have demonstrated intriguing performance [Chen et al., 2017; Lu et al., 2017; Vinyals et al., 2015; Xu et al., 2015]. Although there is no proof that such models can fully capture the complexity of visual scenes, they appear to be able to produce credible captions for a variety of images. This raises the question of whether such artificial systems are using similar strategies employed by the human visual system to generate captions.

One clue onto how humans perform the captioning task is through the study of visual attention via eye-tracking. Attention mechanisms have been studied from different perspectives under the umbrella terms of visual attention (bottom-up and top-down mechanisms of attention), saliency prediction (predicting fixations), as well as eye movement analysis. A large number of studies in computer vision and robotics have tried to replicate these capabilities for different applications such as object detection, image thumbnailing, and human-robot interaction [Borji, 2018; Borji and Itti, 2013; Borji et al., 2012]. There has been a recent trend in adopting attention mechanisms for automatic image captioning [Chen et al., 2017; Lu et al., 2017; Xu et al., 2015]. Such research papers often show appealing visualizations of feature importance over visual regions accompanied with the corresponding phrase "mimicking human attention". One may ask, "Is this really the same as human attention?" and "How much such mechanisms agree with human attention during describing content?".

In this chapter, we strive to answer the aforementioned questions. We establish a basis by studying how humans attend to scene items under the captioning task.

• we introduce a dataset with synchronously recorded eye-fixations and scene descriptions (in verbal form), which provides the largest number of instances at the moment.

• we compare human attention during scene free-viewing with human attention during describing images

• we analyze the relationship between eye-fixations and descriptions during image captioning.

• we compare human attention and machine attention in image captioning.

• we integrate image saliency with soft attention to boost image captioning performance.

The rest of this chapter is organised as follows: section 4.2 will introduce the background of top-down attention in image captioning; section 4.3 will introduce the data collected for the research in this chapter; section 4.4 will analyse the human attention under image captioning task; followed by the conclusion in section 4.5.

4.2 Background

4.2.1 Bottom-up attention and saliency prediction

Predicting where humans look in an image or a video is a long standing problem in computer vision, a comprehensive review of which is beyond the scope of this chapter (See Borji [2019] and last chapter). However, current saliency models are trying to replicate humans' bottom-up attention mechanism during free-viewing of natural scenes. By contrast, this chapter will focus on top-down attention, namely, where people would look in the image under the image captioning task.

4.2.2 Neural image captioning

Image captioning can be seen as a machine translation problem, i.e., translating an image into an English sentence. A breakthrough in this task has been achieved with the help of large scale databases (e.g, Flickr30k [Young et al., 2014] and MSCOCO [Lin et al., 2014]) that contain a large number of images and captions (i.e, source image and target instances). The neural captioning models often consist of a deep Convolutional Neural Network (CNN) and a Long Short Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] language model, where the CNN part generates the feature representation for the image, and the LSTM cell acts as a language model, which decodes the features from the CNN part to the text [Vinyals et al., 2015]. In this chapter, we mainly focus on models that incorporate attention mechanisms (top-down attention). Xu et al. [2015] introduces a soft-attention mechanism to the approach in Vinyals et al. [2015]. That is, during the generation of a new word, based on the previously generated word and the hidden state of the language model, their model learns to put spatial weight on the visual features. Instead of re-weighting features only spatially, Chen et al. [2017] exploits spatial and channel wise weighting. Lu et al. [2017] utilizes memory to prevent the model in attending mainly to the visual content and enforce it to utilize textual context as well. This is referred to as *adaptive attention*. Chen and Zhao [2018] applies the visual saliency (free-viewing saliency) to boost the captioning model. They use weights learned from saliency prediction to initialize their captioning model, but the relative improvement is marginal compared to training their model from scratch.

All aforementioned attention models have pushed the developments of automatic image captioning. Crucially, they've shown impressive visualisation results about the attention heat map, which are said to "attend to the correct region for each generated text". However, those visualisation results can only be qualitatively interpreted without any references. In this chapter, we try to analyze those attention from humans' perspective, i.e., whether they're doing the same thing as humans. To that end, we collected a dataset which recorded human attention under the image captioning task. With the collected dataset, we compare the difference of models' attention learned from data and humans' attention.

4.2.3 Human attention and image descriptions

Dataset	# images	# subjects	# Instances
DIDEC [Miltenburg et al., 2018]	305	45	4604
SNAG [Vaidyanathan et al., 2018]	100	30	3000
sbugaze [Yun et al., 2013]	1000	3	3000
Ours	4000	16	14000

Table 4.1 Comparing our data with other similar datasets.

In the computer vision community, some previous works have investigated the relationship between human attention and image captioning [Itti and Arbib, 2006; Tanenhaus et al., 1995]. Yun et al. [2013] studied the relationship between gaze and descriptions, where the human gaze was recorded under the free-viewing condition. In their work, subjects were shown an image for 3 seconds, and another group of participants seperately described the image content (We will refer to their collected data as *sbugaze*). However, their analysis is based on data collected under the free-viewing scenario. Furthermore, the sentence and gaze used in their analysis is not come from the same person. Their analysis thus cannot be used to compare with the top-down attention models. Tavakoli et al. [2017c] pushed further to investigate the relation between machine-generated and human-generated descriptions. They looked into the contribution of boosting visual features spatially using saliency models as a replicate to bottom-up attention. Again, their analytical results are still based on the free-viewing data, which is not directly linked with the image captioning task. Das et al. [2017] studied the relationship between human attention and machine attention in visual question answering. Although they draw the conclusion that human attention is totally different from machine attention in visual question answering. Whether the conclusion is the same for image captioning is unknown to us, which is to be investigated in this chapter.

Contrary to previous studies, this chapter focus on the human attention under the image captioning task and investigate the attention behaviour by humans and machines.

4.2.4 Eye-Tracking in NLP

In the natural language processing community, eye-tracking and image descriptions have been used to study the cause of ambiguity between languages, e.g., English vs Dutch [Miltenburg

Dataset	CIDEr mean/variance	METEOR mean/variance
sbugaze [Yun et al., 2013]	0.938/0.038	0.368/ 0.012
Ours	0.937/0.060	0.366/ 0.015

Table 4.2 Assessing quality of the collected captions against 50 ground-truth captions of the Pascal-50S.

et al., 2018]. Vaidyanathan et al. [2018] investigated the relation between linguistic labels and important regions in the image by utilizing eye-tracking data and image descriptions. However, all eye-tracking datasets used in NLP are still in small scale. In contrast to existing datasets in the natural language processing community, our dataset features a higher number of instances and images in total, making it more suitable for vision related tasks (see Table 4.1 for a comparison). In contrast to prior works, we also pursue a different goal which is: *understanding how well current computational attention mechanisms in image captioning models align with human attention behaviour during image description task.*

4.3 Data Collection

Stimuli: Our collected data is organised in two corpora, one for analysis and another one for model training, denoted by *capgaze1* and *capgaze2*, respectively. The *capgaze1* corpus is used for analysis in this chapter and the *capgaze2* corpus is used for modelling the visual saliency under the image captioning task. For *capgaze1*, 1,000 images were selected from the Pascal-50S dataset [Vedantam et al., 2014], which provide 50 captions per image by humans and annotated semantic masks with 222 semantic categories (the same images as in *sbugaze*). For *capgaze2*, 3,000 images were randomly chosen from the MSCOCO [Lin et al., 2014]. Yun et al. [2013] recorded eye movements of subjects during free-viewing images in *sbugaze*. Thus, we can use *capgaze1* to compare humans attention between free-viewing and captioning.

Apparatus: Precise recording of subjects' fixations in the image captioning task requires specialized accurate eye-tracking equipment, making crowd-sourcing impractical for this purpose. We used a *Tobii X2-30* [Tobii] eye-tracker to record eye movements under the image captioning task in a controlled laboratory condition. The eye-tracker was positioned at the bottom of the laptop screen with a resolution of 1920 × 1080. The subject's distance from the screen was about 40cm. Subject was asked to simultaneously look at the image and describe it in one sentence in verbal form. The eye-tracker and an embedded voice recorder in the computer recorded the subject's eye movements and descriptions synchronously for each image.

In *capgaze1* corpus, five subjects (postgraduate students, native English speakers, 3 males and 2 females) participated in the data collection. All five subjects finished the data collection over all 1,000 images in this corpus. *capgaze2* corpus, eleven subjects (postgraduate students, 3 females and 8 males) participated in the data collection. Each image in this corpus has the recorded data from three different subjects. The image presentation order was randomized across subjects. For each subject, we divided the data collection into 20 images per session. Before each session, the eye-tracker was re-calibrated. At the start of a session, the subject was asked to fixate on a central red cross, which appeared for 2s. The image was then displayed on the screen and the subject viewed and described the image. After describing the image, the subject presses a designated button to move to the next image in the session. An example of the collected data is illustrated in Figure 4.1. During the experiments, subjects often looked at the image silently for a short while to scan the scene, and then started describing the content spontaneously for several seconds.

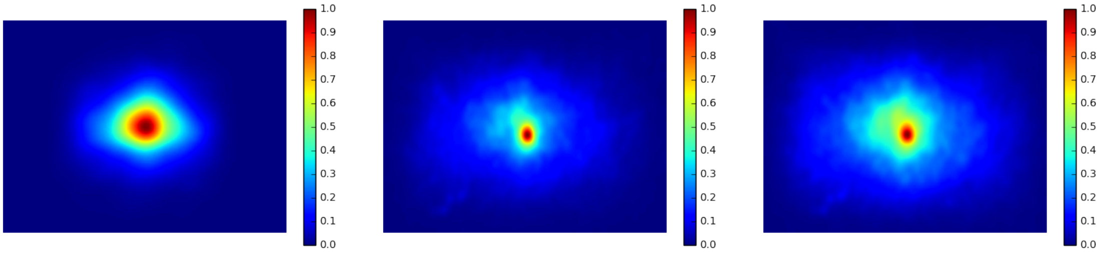
Post-processing: After data collection, we manually transcribed the oral descriptions in *capgaze1* corpus into text for all images and subjects. The transcriptions were double-checked and cross checked with the images. We used off-the-shelf part of speech (POS) tagging software [Manning et al., 2014] to extract the nouns in the transcribed sentences. We then formed a mapping from the extracted nouns to the semantic categories present in the image. For example, *boys* and *girls* are both mapped into the *person* category.

To check the quality of captions in our collected data, we compute the CIDEr [Vedantam et al., 2015] and METEOR [Denkowski and Lavie, 2014] scores of the collected captions based on the ground truth in *Pascal-50S* dataset (50 sentences for each image). This is to ensure that eye tracking and simultaneous voice recording have not affected the quality of captions adversely. We compared our scores with the scores of *sbugaze* [Yun et al., 2013] captions, that were collected in text form, with asynchronous eye tracking and description collection. Table 4.2 summarizes the results, showing that eye-tracking does not appear to distract the subjects as their descriptions rated comparably to the ones in *sbugaze*.

4.4 Analysis

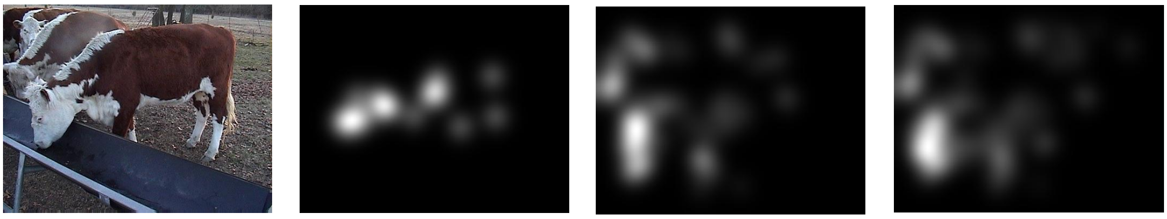
In this section, we provide a detailed analysis of (i) attention during free-viewing and captioning tasks, (ii) the relationship between fixations and generated captions, and (iii) attentional mechanisms in captioning models.

4.4.1 Attention in free-viewing vs. attention in image captioning



(a) *free* (b) *cap3s* (c) *cap*

Fig. 4.2 Average fixation map across the whole dataset for (a) the free-viewing condition (*sbugaze*), (b) first 3 seconds of image captioning condition (*cap3s*), and (c) the whole duration of captioning condition (*cap*).



Several cows eating from a trough; Cows eating from a trough in a field; Cows feeding out of a pen; Brown and white cows in a field eating from a trough; Cows eating from a trough

Fig. 4.3 An example of the difference between fixations in free-viewing and image captioning tasks. From left to right: original image, free-viewing fixations, first 3s fixations, and all fixations in the captioning task. The captions generated by 5 subjects are shown at the bottom.

	Reference task		
	free	cap3s	cap
free	0.81	0.78	0.75
cap3s	0.84	0.84	0.81
cap	0.84	0.85	0.83

Table 4.3 Cross task IOC in terms of AUC-Judd

How does attention differ between free-viewing versus describing images? We first analyze the differences between the two tasks by visualizing the amount of attentional center-bias and the degree of cross task inter-observer congruency (IOC). The *sbugaze* dataset contains gaze for a maximum of duration of 3s, in free-viewing condition. In our experiments, subjects need in average $6.79s$ to look and describe each image. To ensure the difference in gaze locations is not solely due to viewing duration, we divide the visual attention in the image captioning task into two cases: i) fixations during the first $3s$ (*cap3s*), and ii) fixations during the full viewing period (*cap*).

The difference in visual attention between free-viewing and image captioning is shown in Figure 4.2 and 4.3. We find that visual attention in free-viewing is more focused towards the central part of the image (i.e, high centre-bias), while attention under the image captioning task has higher dispersion over the whole duration of the task.

Table 4.3 reports the cross task *Inter Observer Congruency* (IOC). To compute cross task IOC, we leave fixations of one subject in one task out and compute its congruency with the fixations of other subjects in another task using the AUC-Judd [Bylinskii et al., 2018] evaluation score. The results shows that humans' attention in captioning task is different from free-viewing. And higher AUC score under the image captioning task indicates us that few people is required per image to collect the eye fixations, which would significantly reduce the manual cost in the task-based humans' eye-fixation data collection.

What is the difference between the saliency model trained with free-viewing saliency (bottom-up attention) and top-down saliency (top-down attention)? We hypothesize that model trained on bottom-up saliency should have a better performance when evaluated on free-viewing dataset, and vice versa. To confirm our hypothesis, we adopt the deep saliency prediction architecture in chapter 3, and trained two models. One model is trained on our *capgaze2* corpus (2500 images for training and 500 images for validation), and another model is trained on Salicon [Jiang et al., 2015] free-viewing saliency database. We test the two models on the our *capgaze1* corpus (task saliency) and MIT1003 benchmark [Judd et al., 2009] (free-viewing saliency). The models performance is reported in Table 4.4. We can see that the model trained on our dataset has better performance to predict the image captioning saliency, whereas the model trained on free-viewing database is better to predict free-viewing saliency. This result also suggests that, when saliency is applied in image captioning task [Chen and Zhao, 2018; Tavakoli et al., 2017b], models trained on free-viewing dataset are not optimal to predict the salient regions for image captioning. Our newly collected data provides a better solution for model training.

Furthermore, as shown in the qualitative examples in Figure 4.4 (first and second row), when training the saliency prediction model using different dataset, the model tend to predict similar salient regions in the image. The main difference is that the importance of each region

training dataset	evaluation on *capgaze1*			evaluation on MIT1003		
	S-AUC	NSS	SIM	S-AUC	NSS	SIM
Salicon	0.65	1.63	0.58	0.73	2.28	0.46
capgaze2	0.694	1.92	0.6	0.69	1.67	0.4

Table 4.4 Saliency prediction performance's comparison using different training dataset. Salicon and MIT1003 are free-viewing datasets. *capgaze1* and *capgaze2* are eye-fixation datasets under the image captioning task.

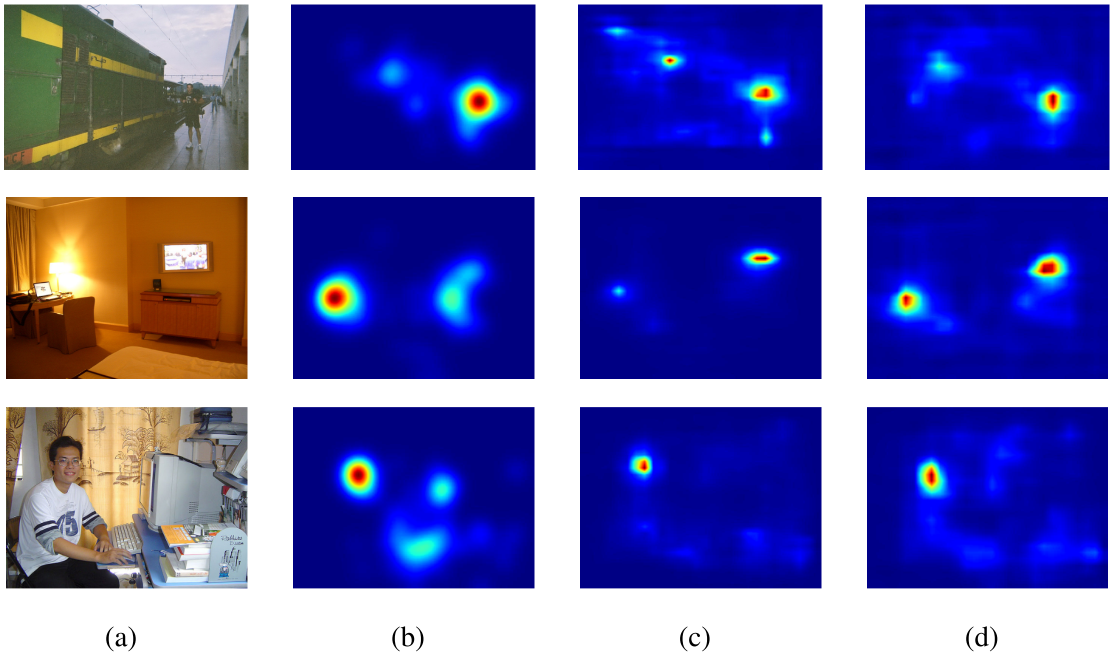


Fig. 4.4 Qualitative examples of predicting the saliency map under image captioning task. From left to right: (a) image, (b) ground truth, (c) saliency map predicted by model trained on free viewing database (Salicon), (d) saliency map predicted by model trained on our *capgaze2* corpus

is different when the model is trained using different dataset. Model trained on our *capgaze2* corpus is more accurate to highlight those salient regions in image captioning.

4.4.2 Analyzing the relationship between fixations and scene descriptions

How does task-based attention relates to image description? To answer, we analyze the distribution of fixations on objects in the scene and the relation between attention allocation and noun descriptions in the sentences. Given described objects ($\mathscr{D}(O)$), non-described objects ($\neg\mathscr{D}(O)$), described background (e.g, mountain, sky, wall) denoted as $\mathscr{D}(B)$, non-described background ($\neg\mathscr{D}(B)$), and fixated objects ($F(O)$), we compare the distribution of the fixation data on objects and image background in free-viewing, first $3s$ captioning and full captioning tasks. We compute the attention allocation for regions of interest as:

$$\text{attention allocation} = \frac{\text{\# fixations on a region}}{\text{\# total fixations on the image}} \tag{4.1}$$

	$\mathscr{D}(O)$	$\neg\mathscr{D}(O)$	$\mathscr{D}(B)$	$\neg\mathscr{D}(B)$
free	0.66 ±0.08	0.09 ±0.03	0.14 ±0.04	0.11 ±0.04
cap(3s)	0.68 ±0.05	0.09 ±0.02	0.14 ±0.03	0.09 ±0.03
cap	0.63 ±0.04	0.10 ±0.02	0.16 ±0.02	0.11 ±0.03

Table 4.5 Mean attention allocation on different regions (object vs background). Statistics computed from our *capgaze1* dataset from 5 participants.

Are described objects more likely to be fixated? Table 4.5 shows the results in terms of overall attention allocation. In all viewing conditions most of the fixations correspond to objects that are described in the caption. This is in line with previous findings in Tavakoli et al. [2017c]: described objects receive more fixations than background (either described or not) and non-described objects. When comparing the fixations in the free-viewing and captioning conditions, we see that in the first $3s$ of captioning (the common viewing duration for free-viewing), slightly more attention is allocated to the described objects. Analyzing the captioning task for the full duration, we observe a *decrease* in the attention allocation on described objects and an *increase* in the attention to the described background. This indicates that subjects are more likely to attend to the items which are going to be described in the first few seconds, before shifting their attention towards context-defining elements in the scene.

Are the objects that appear at the start, rather than the end of the description, more likely to be fixated? Table 4.6 shows the magnitude of attention allocation to objects with respect to their order of appearance in the descriptions (noun order). We see that nouns that are described first receive a larger fraction of fixations than the subsequent nouns. The slightly lower number in the captioning condition is associated with the change in viewing strategy observed after the first $3s$, as discussed previously.

How much time do subjects spend viewing described objects? Synchronous eye tracking and description articulation enables us to investigate the duration of fixations T_F on scene elements, specifically on described and non-described objects. As shown in Table 4.7, described objects attract longer fixations than non-described objects. This indicates that once an important object grabs the attention, more time is allocated to scrutinize it.

	Noun Order				
	1	2	3	4	5
cap	0.486	0.201	0.147	0.097	0.053
free	0.502	0.204	0.158	0.107	-

Table 4.6 Attention allocation on described objects based on the order in which they appear in the description

	$\mathscr{D}(O)$	$\neg\mathscr{D}(O)$
$T_F(O)$	1.68 s	0.52s

Table 4.7 The mean fixation duration (T_F) on described objects vs. non-described objects

	$p(\mathscr{D}(O)\|F(O),O)$	$p(F(O)\|\mathscr{D}(O),O)$
free	0.56	0.87
cap (3s)	0.48	0.95
cap	0.44	0.96

Table 4.8 The probability of an object being described when fixated ($p(\mathscr{D}(O)|F(O),O)$) vs. fixated when described ($p(\mathscr{D}(O)|F(O),O)$). Under the free viewing condition, one group people freely view the image for eye fixation collection, and another group people generate captioning for the each image.

How likely is an object to be described if it is fixated? We compute the probability, $p(\mathscr{D}(O)|F(O),O)$), and compare it with the probability that an object is fixated when it is described (when it is present in the image), $p(F(O)|\mathscr{D}(O),O)$. In other words, are we more likely to fixate on what we describe, or to describe what we fixate? To this end, we use the annotated semantic mask of each object in *capgaze1* and our collected and transcribed text description to find which object in the image was/wasn't fixated and described. More specifically, if an object mask has eye-fixation on it, it is considered as fixated. If the transcribed text has the word of the semantic mask's category in the image, we consider the corresponding object under that mask has been described[2]. Results are summarized in Table 4.8. They confirm the expectation that described objects are very likely to be fixated, whereas many fixated objects are not described. Interestingly, under the image captioning task, more fixated objects are not described, whereas described objects are more likely to be fixated.

How often do subjects describe something not annotated in the image (i.e, not present in the image at all)? Also, which nouns are described more often and which nouns are less likely to be mentioned? The data is visualized in Figure 4.5, 4.6,and 4.7. Most occurrences

[2] Usually, each category only has one mask in an image in the dataset.

$p(\neg\mathscr{D}(O)\vert F(O),\mathscr{D}(S))$	$p(\neg\mathscr{D}(O)\vert F(O),\neg\mathscr{D}(S))$
0.48	0.21

Table 4.9 Probability of fixation on non-described objects when the scene category is described and not described

of described but un-annotated nouns are *scene categories* and *places* nouns, that are not annotated as scene elements (because annotations are local and pixel-based). One glaring exception to this, where an object not present in the scene is described, is the special case of 'camera' (example in Figure 4.8). The reference to 'camera' is often associated with captions that refer to the *photographer* taking the picture. Since the word camera in this case denotes a property of the scene rather than the material object (here actual camera), we can loosely construe such cases as a scene category.

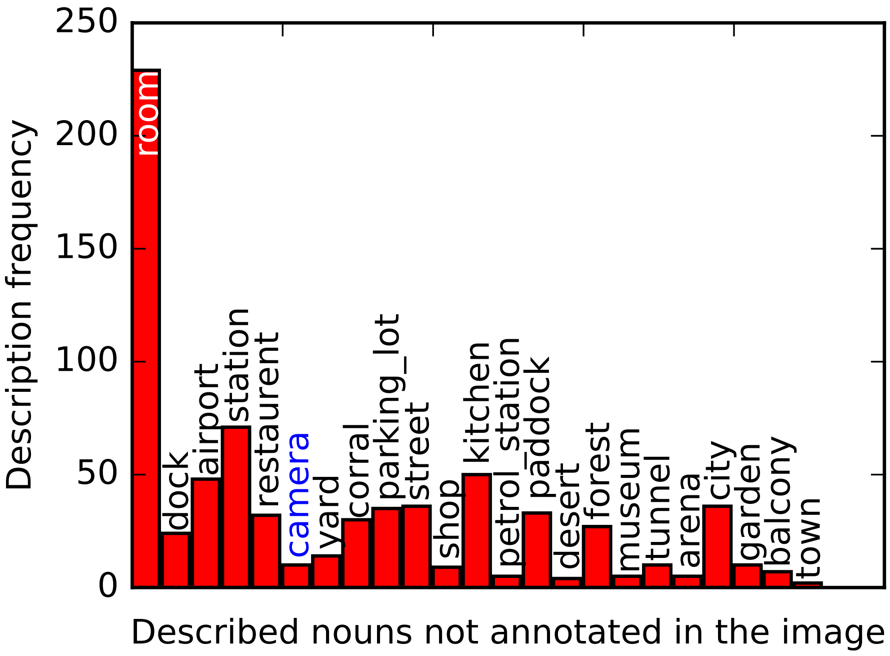


Fig. 4.5 The nouns described in the caption but not annotated in the image.

How does scene category affect the description of objects? We measure the probability of non-described objects if they are fixated when the scene category is mentioned, i.e, $p(\neg\mathscr{D}(O)|F(O),\mathscr{D}(S))$, and when the scene is not mentioned, $p(\neg\mathscr{D}(O)|F(O),\neg\mathscr{D}(S))$. The results, summarized in Table 4.9, indicates that non-described but fixated objects tend to occur when subjects describe the scene context. This indicates that fixated yet non-described objects likely contribute to the perception of the scene context.

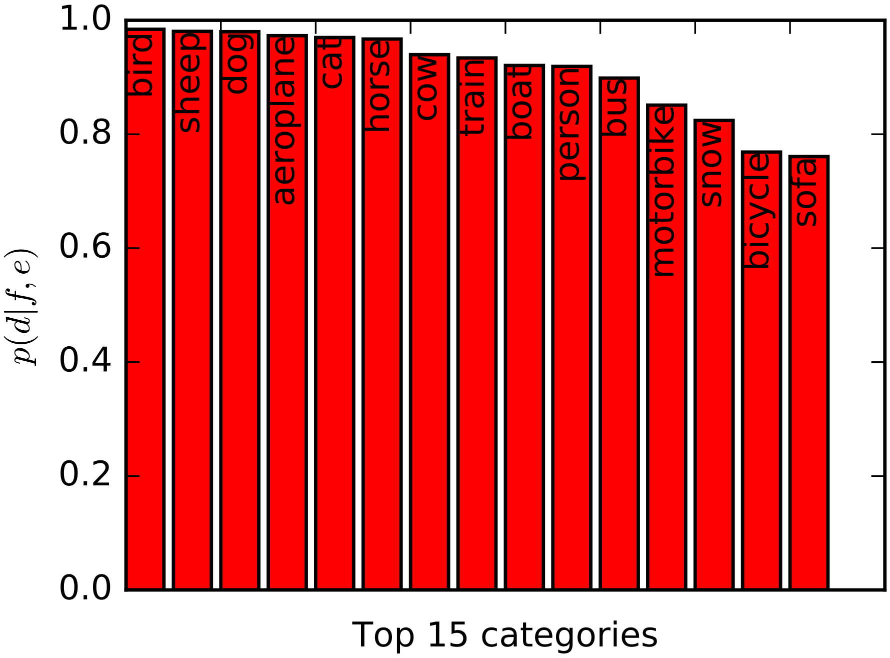


Fig. 4.6 The fixated objects (top 15) that have a very high likelihood to be described.

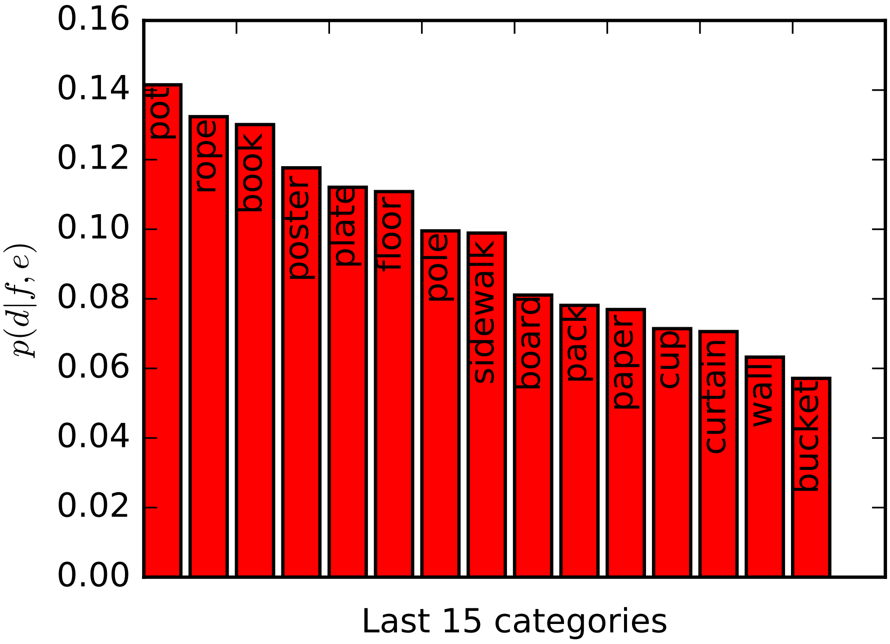
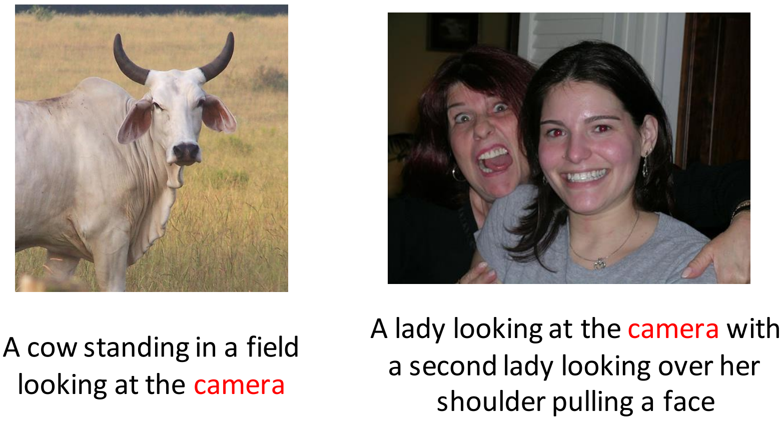


Fig. 4.7 The fixated objects that have a very low likelihood to be described.



A cow standing in a field looking at the camera

A lady looking at the camera with a second lady looking over her shoulder pulling a face

Fig. 4.8 Two examples when ***camera*** was described by subjects but did not exist in the image.

4.4.3 Comparing human and machine attention

How similar are humans' and machines' attention in image captioning? This section describes two analyses performed to answer this question.

Attention in the visual encoder

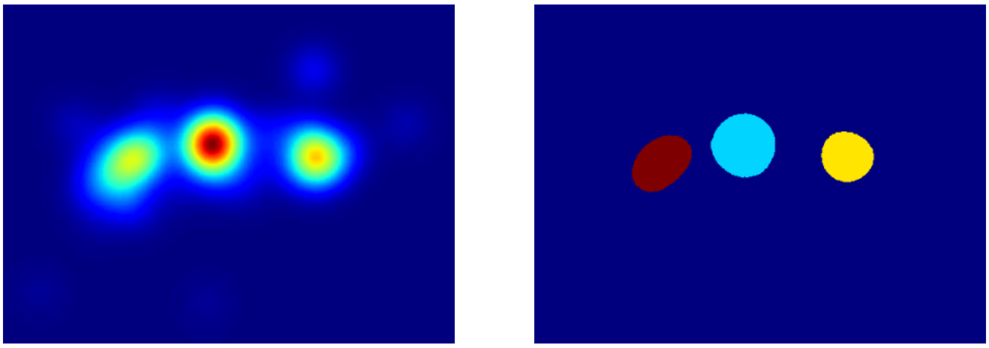


Fig. 4.9 An example of obtaining attended regions from human fixation map. Left: saliency map; Right: all attended region extracted from saliency map (different colors represent different regions).

An overlooked aspect in previous research is the amount of saliency that may have been encoded implicitly within the visual encoder of a deep neural network. Consider the situation where a standard convolutional neural network (CNN) architecture, often used for encoding visual features, is used to provide the features to a language model for captioning. We ask (1) *to what extend does this CNN capture salient regions of the visual input?* and (2) *how well*

	percentage	mean value
free	72.5%	5.43
cap3s	78.1%	5.62
cap	77.9%	5.61

Table 4.10 Attention agreement between human and the visual encoder (pre-trained CNN).

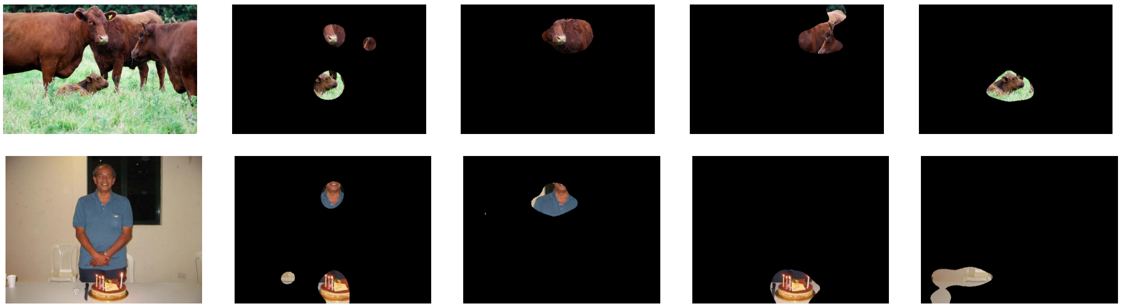


Fig. 4.10 Example of human attention under captioning task and VGG-16's attention. From left to right: image, human attended regions, and VGG-16 attended regions that best correlated with each human attended region.

do the salient regions of the CNN correspond to human attended locations in the captioning task?

To answer these questions, as per standard, we first transform the collected fixation data into saliency maps by convolving them with a Gaussian filter (σ corresponding to one degree of visual angle in our experiments). Then, we threshold the saliency map by its top 5% value and extract the attended regions (examples in Figure 4.9). We then check how well the activation maps in the CNN, here layer conv5-3 of the VGG-16 [Simonyan and Zisserman, 2014] (including 512 activation maps) correspond to those attended regions. To this end, for each region, we identify if there is an activation map that has a NSS score [Bylinskii et al., 2018] higher than a threshold[3] within that region. If there exists one, then the corresponding region is also attended by the CNN. We report how many regions in images attended by humans are also attended by machine, as well as the mean highest NSS score of all the attended regions in all images (each connected region has a highest NSS score from 512 activation maps). We use fixation maps from free-viewing attention (free), first 3s fixations under the captioning task (cap3s), and the fixations of the whole duration of the image captioning task (cap). Results are shown in Table 4.10. It can be seen that there exists a large agreement between internal activation maps of the encoder CNN and the human attended regions (over 70%). Interestingly, despite not fine-tuning the CNN for captioning,

[3] as per method in last chapter for threshold selection, we compute top 10 mean NSS for each region, and then compute T as the mean of top 10 mean NSS from all regions. T=4 here.

this agreement is higher for the task-based eye movement data than that for free-viewing fixations (See examples in Figure 4.10).

Attention in image captioning models

Model	Ground truth	
	free-viewing	image captioning
SalGAN [Pan et al., 2017]	1.929/0.72	1.618/0.677
Soft-attention [Xu et al., 2015]	1.149/0.622	1.128/0.622

Table 4.11 Spatial attention consistency evaluation for bottom-up saliency model (SalGAN), and top-down attention captioning model (Soft-attention). Evaluated by NSS/s-AUC.

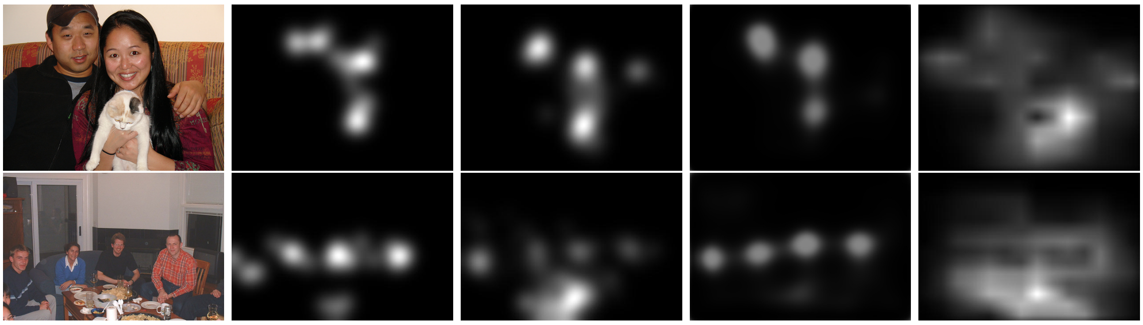


Fig. 4.11 Example of spatial attention difference. From left to right: original image, attention in free-viewing, attention in image captioning, saliency map predicted by SalGAN, and saliency map from top-down image captioning model.

How well does the top-down attention mechanism in the automatic image captioning model agree with human attention when describing images? We study the spatial and temporal consistency of the soft-attention mechanism in Xu et al. [2015] with human attention in image captioning, where the model is trained on image captioning dataset to mimic the human top-down attention mechanism.

Spatial consistency: We assess the consistency between the spatial dimension of human attention and machine. For machine, we use the model proposed in Xu et al. [2015], which is trained with soft-attention on image captioning dataset [Lin et al., 2014]. The spatial attention for this model is computed as the mean saliency map over all the generated words. We compute the NSS and s-AUC [Bylinskii et al., 2018] over this saliency map using human fixations. We also compare with bottom-up saliency models by computing the NSS and s-AUC over the saliency maps of SalGAN [Pan et al., 2017], a leading saliency model without centre-bias trained on large-scale saliency prediction dataset [Jiang et al., 2015].

Table 4.11 summarizes the consistency of the saliency maps generated by a standard bottom-up saliency model (trained on free-viewing data) [Pan et al., 2017] and a top-down soft attention image captioning system [Xu et al., 2015], with ground truth saliency maps captured either in the free-viewing or the captioning condition (full duration). Interestingly, the bottom-up saliency obtains higher scores on both free-viewing and task-based ground-truth data. In other words, a bottom-up saliency prediction model is a better predictor of human attention than the top-down soft-attention model, *even for the captioning task*. Figure 4.11 illustrates some example maps.

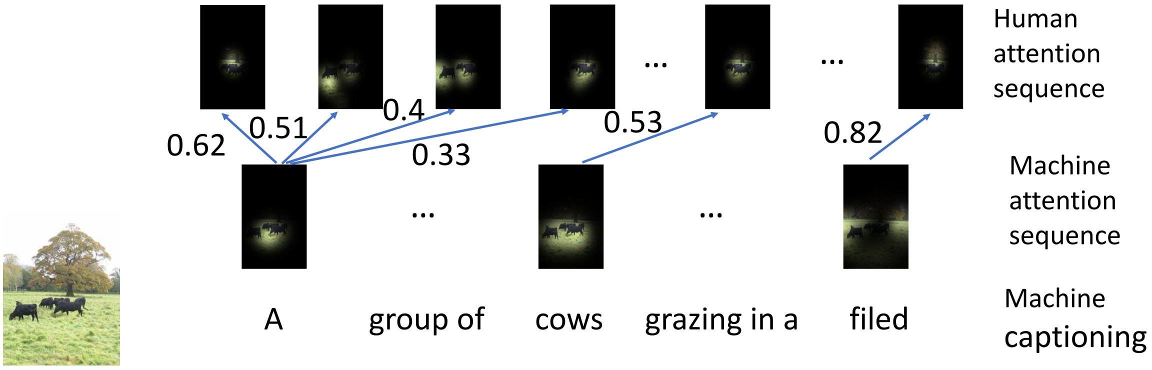


Fig. 4.12 Example of Dynamic Time Warping between human attention and machine attention. Top row is the human attention sequence on the image when describing the image, the bottom row is the attention sequence from soft-attention model when generating the caption for the image. The number besides each blue arrow is the distance for each warping step.

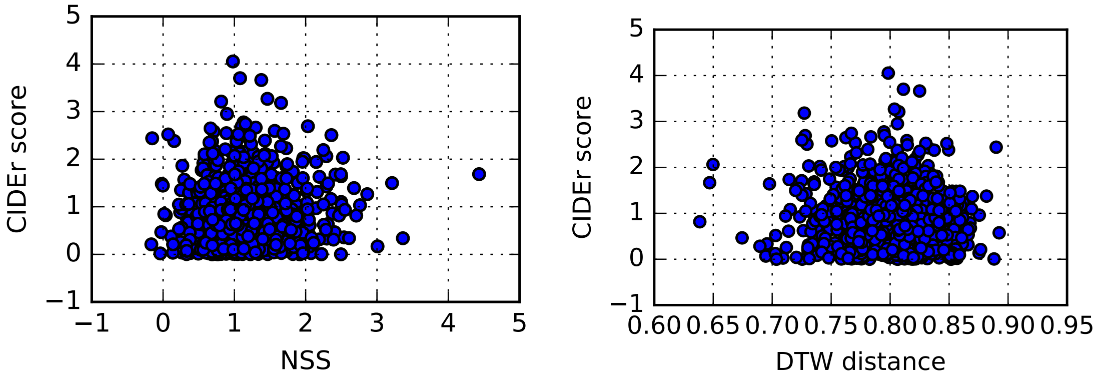


Fig. 4.13 Correlation between machine-human attention congruency (spatial and temporal) and machine performance on image captioning (CIDEr score).

Temporal consistency: What is the temporal difference between human and machine attention in image captioning? Here, for the human fixation data, we split the sequence of fixations by intervals of $0.5s$ using the recorded sample time stamps. The fixations of each interval are then transformed into separate saliency maps, resulting in a sequence of saliency maps. For machine attention, we use the sequence of generated saliency maps during the scene description. We then employ Dynamic Time Warping (DTW) [Müller, 2007] to align the sequences and compute the difference between them. Figure 4.12 shows this process for an example sequence. We report the distance between each frame pair as $1-\text{SIM}(h_i, m_j)$, where h_i is the i^{th} frame in the human attention sequence, m_j is the j^{th} frame in the machine attention sequence, and SIM is the similarity score [Bylinskii et al., 2018] between the two attention maps. The final distance between two sequences is the total distance divided by the path length in DTW. Our analysis shows a mean difference of 0.8, which is large and demonstrates that the two attention patterns differ to a large extent over time.

Correlation between machine captioning performance and machine-human attention congruency: Is the consistency between the machine and human subjects' attention patterns a predictor of the quality of the descriptions generated by the machine? To answer this question, we compute the *Spearman correlation coefficient* between the machine performance on each image instance in terms of caption quality (CIDEr score) and the consistency of machine attention (spatial and temporal) with human (NSS score for spatial consistency, DTW distance for temporal consistency). Results are visualized in Figure 4.13, indicating a very low coefficient, 0.01 and -0.05 for spatial and temporal attention, respectively. In other words, there seems to be no relation between the similarity of the machine's attention to humans' and the quality of the generated descriptions.

4.5 Can Saliency Help Image Captioning?

Based on the analytical result in Table 4.8, 96% of described objects are fixated (87% in free-viewing), which means the image saliency map provides a prior knowledge of where to attend in image for captioning. In contrast, the soft-attention models for image captioning first treat all regions *equally*, before *re-weighting* each region when generating each word.

Here, we check if image saliency can help image captioning by proposing a generic architecture, which combines the visual saliency and soft-attention mechanism for image captioning as depicted in Figure 4.14. Our architecture has three parts: a *saliency prediction module* (SPM), a *perception module* (PM), and a *language model* (LM). In the SPM part, we train a saliency prediction model f_{sal} on the *capgaze2* corpus to predict a saliency map (S)

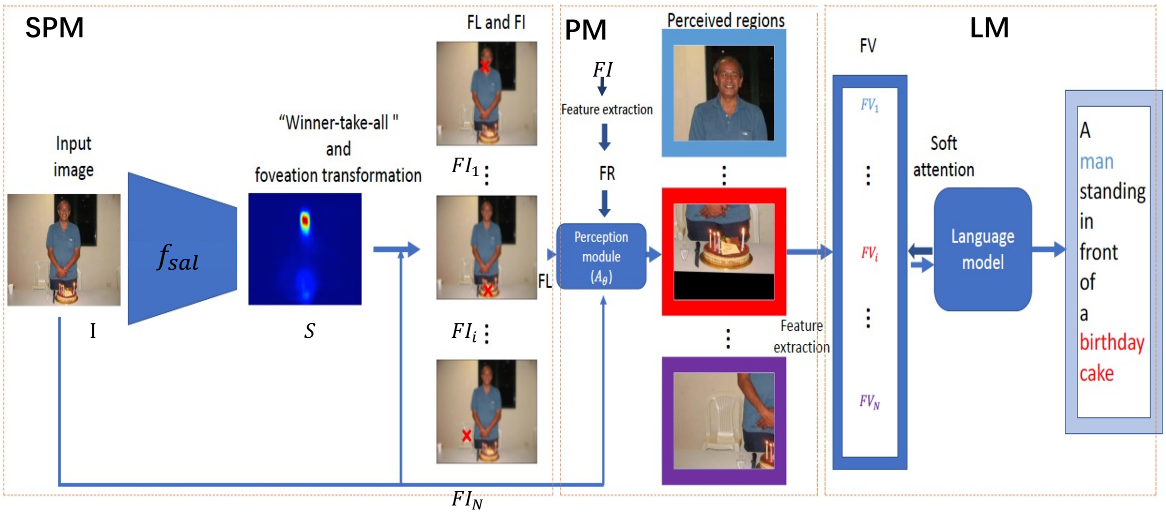


Fig. 4.14 Architecture of saliency based image captioning. Red crosses indicate fixated locations FL. The saliency prediction module (SPM) predicts the saliency map of the input image, which is used to extract the fixated locations (FL) and the foveated images (FI). Then the perception module precepts a region from each foveated image. These perceived regions are then processed and fed into the language model (LM) for final sentence generation.

for each image I:

$$S = f_{sal}(I) \tag{4.2}$$

From this saliency map, we use a "winner-take-all" approach [Koch and Ullman, 1987] to extract a set of fixated locations (FL) for each image. Specifically, we extract the fixated location and update the saliency map iteratively:

$$\begin{aligned} (x_t, y_t) &= \arg\max\ S_t \\ S_{t+1} &= S_t(1 - g(x_t, y_t)) \end{aligned} \tag{4.3}$$

where $g(x_t, y_t)$ is a Gaussian blob centered at location (x_t, y_t)[4].

We end the location extraction process if the number of total extracted locations exceed a threshold T_N or the maximum value in the updated saliency map is less than a threshold T_V.

Those extracted locations are denoted as:

$$FL = \{(x_1, y_1), \cdots, (x_i, y_i), \cdots, (x_N, y_N)\} \tag{4.4}$$

[4] σ is chosen as per standard in [Wang et al., 2017b].

For each fixated location in the image, we apply a foveation transformation [Wang et al., 2017b]:

$$FI_i = (1 - g(x_i, y_i))(I * g_f) + g(x_i, y_i)I \tag{4.5}$$

where g_f is a Gaussian blur filter, producing a set of foveated images (FI) for these fixated locations:

$$FI = \{FI_1, \cdots, FI_i, \cdots, FI_N\} \tag{4.6}$$

We further process each foveated image with a pre-trained CNN (resnet-18), yielding a K dimensional vector for each foveated image. Finally, for each image, we have a set of foveated representations (FR):

$$FR = \{FR_1, \cdots, FR_i, \cdots, FR_N\} \tag{4.7}$$

The bridge between our SPM and LM is a learned perception module (PM), parameterized by a function f, in which we use a Localised Spatial Transformer Network [Jaderberg et al., 2015] (LSTN). Concretely, for each fixated image location (x_i, y_i), the PM generates an affine transformation (A_i), based on the corresponding FR_i, to perceive a region centred at the fixated location:

$$\begin{aligned} A_i &= \left[f(FR_i) \;\middle|\; (x_i, y_i)^\top \right] \\ &= \left[\begin{array}{cc|c} \theta_{i_{11}} & \theta_{i_{12}} & x_i \\ \theta_{i_{21}} & \theta_{i_{22}} & y_i \end{array} \right] \end{aligned} \tag{4.8}$$

Each perceived region is then processed by a feature extraction network, and represented by a vector of dimension K. Finally, for each image, it has a set of feature vectors (FV):

$$FV = \{FV_1, \cdots, FV_i, \cdots, FV_N\} \tag{4.9}$$

The language model is a LSTM with a soft attention module (parameterized by a learned function att). The soft attention module (att) receives the FV as input. Based on the hidden state of LSTM ($\mathbf{h}_t$) at time step t and each feature vector in FV, LM generates a weight (w) for each feature vector and then takes the weighted sum of those feature vectors (WSFV) in FV to update the LSTM state and to generate the next word:

$$w_{(t+1),i} = att(FV_i, \mathbf{h}_t) \tag{4.10}$$

$$WSFV_{t+1} = \frac{1}{N}\sum_{i=1}^{N} w_{(t+1),i} \cdot FV_i \tag{4.11}$$

$$\mathbf{h_{t+1}} = LSTM(\mathbf{h}_t, WSFV_{t+1}) \tag{4.12}$$

The only difference between our model and the original soft attention model in Xu et al. [2015] is that our attention module only emphasizes salient regions *guided by the SPM and perceived by PM*, whereas the original soft attention model emphasizes *all* regions in the image when generating each word.

Our architecture is trained in two stages. In the first stage, we train the SPM, and extract the FL and FR. Then, the PM and LM are trained jointly by minimizing the cross entropy loss of the caption generation. The pre-trained feature extraction for FR is resnet-101 [He et al., 2016], which transforms each perceived region into a 2,048-dimensional feature vector. The learning rate is set to 10^{-3}, and is decreased by a factor of 0.8 every 3 epochs. Early stopping is used if the BLEU-4 [Papineni et al., 2002] score does not increase in five consecutive epochs. Our model is trained and tested on Flickr30k and MSCOCO benchmarks using the Karpathy's split [Karpathy and Fei-Fei, 2015].

Four metrics are used for evaluation, including BLEU-4 (B4), ROUGEL (RG) [Lin, 2004], METEOR (MT), and CIDEr (CD)[5]. We also consider the use of free-viewing saliency in the image captioning (i.e, saliency prediction model trained on Salicon [Jiang et al., 2015] database).

The performance of our architecture is shown in Table 4.12 and 4.13. Our baseline model is the soft attention model in Xu et al. [2015] (for fair comparison, we re-implement this model with ResNet-101 as backbone). Our model significantly improves the performance of the soft-attention model by integrating a bottom-up saliency approach to the soft attention model. The model using *task saliency* (saliency prediction model trained on our *capgaze2* corpus) performs better than the one trained using free-viewing saliency—although the difference is not large. Our model is a general architecture, which could easily be integrated with other CNN backbones or the *adaptive attention* mechanism in Lu et al. [2017]. We also believe that the architecture can be applied to other tasks where visual saliency is important.

Model	B4 ↑	MT ↑	RG ↑	CD ↑
baseline (soft attention)	0.224	0.197	0.456	0.451
ours-free	0.238	0.201	0.462	0.476
ours	**0.246**	**0.208**	**0.468**	**0.486**

Table 4.12 Performance on Flickr30k testing dataset (ours-free means saliency prediction model trained on free-viewing saliency database)

[5] The first three metrics range from 0 to 1. CIDEr ranges from 0 to 10.

Model	B4 ↑	MT ↑	RG ↑	CD ↑
baseline (soft attention)	0.331	0.258	0.54	1.04
ours-free	0.342	0.263	0.551	1.06
ours	**0.348**	**0.265**	**0.552**	**1.08**

Table 4.13 Performance on MSCOCO testing dataset

4.6 Discussions and Conclusion

In this chapter, to study human attention in image captioning task and to examine machine learned attention in the same task, we introduced a novel and relatively large dataset consisting of synchronized multi-modal attention and caption annotations.

With our collected dataset, we revisited the consistency between human attention and captioning models on this data, and showed that human eye-movements differ between image captioning and free-viewing conditions. We also reconfirmed the strong relationship between described objects and attended ones, similar to the findings that have been observed in free-viewing experiments. This indicates that human attention is probably a good prior for the for first step of image captioning, i.e., finding out salient regions in image to be described. Interestingly, we demonstrated that the top-down soft-attention mechanism used by automatic captioning systems captures neither spatial locations nor the temporal properties of humans' attention during captioning. Also, the similarity between humans' and machine's attention has no bearing on the quality of the machine generated captions. This suggests that current soft attention method needs more exploration, either its interpretability or its algorithm design. Finally, with our findings, we combined image saliency in image captioning with soft attention based language models. It demonstrates significant performance improvement to the purely top-down soft attention approach.

Overall, the proposed dataset and analysis offer new perspectives for the study of top-down attention mechanisms in captioning pipelines, providing critical hitherto missing information that we believe will assist further advancements in developing and evaluating image captioning models.

Chapter 5

Contextual Attention—Image Captioning through Image Transformer

The previous two chapters studied where humans and machines look either under free viewing condition or given the specific task of image captioning. However, those attended regions are processed independently: their relationship, i.e., their contextual information, have not been explored. It has been well recognized in the vision community for years that contextual information, or relation between objects, helps downstream tasks, e.g., recognition and detection [Chen and Gupta, 2017; Divvala et al., 2009; Galleguillos and Belongie, 2010].

In this chapter, we will study role of relationships between regions. More specifically, we study how to contextualize a region with other related and/or correlated regions through attention mechanism under the image captioning task.

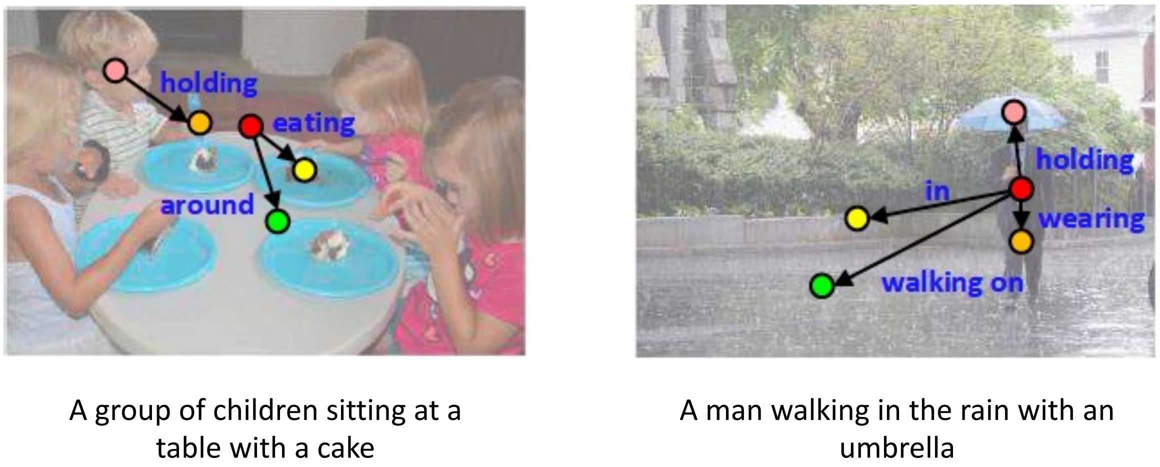
Automatic captioning of images is a task that combines the challenges of image analysis and text generation. Similar to the research in this thesis, the development of image captioning has also gone through the same process, i.e., from "where to look?" to "how to contextualize a region?". Inspired by the successes in text analysis and translation, previous works have proposed the *transformer* architecture [Vaswani et al., 2017] for image captioning. However, the structures between the *semantic units* in images (usually the detected regions from object detection models) and sentences (each single word) are different. Directly apply the transformer's architecture used in NLP to CV tasks is sub-optimal. Limited works have been done to adapt the transformer's internal architecture to images.

In this chapter, we introduce ***image transformer***, which consists of a modified encoding transformer and an implicit decoding transformer, motivated by the relative spatial relationship between image regions. Our design widens the original transformer layer's inner architecture to adapt to the structure of images. With only region's feature as input, our model achieves superior performance on MSCOCO benchmarks.

5.1 Introduction

> *"Based on the structure's difference between one dimension sentences and two dimension images, can we develop more advanced self-attention based contextual attention architecture for computer vision?"*

To describe an image with a sentence, we need to figure out the most salient objects in the image and the relationships between salient objects. Chapter 4 studied the most salient objects in the image through investigating where human would look in the image given the image captioning task and used human attention to help the computational models to find the most salient objects in the image to be described. However, a coherent sentence that captioning an image does not only contains subjects (noun or noun phrase) which describe the most salient objects or regions in the image, but also predicates, e.g., verb, which describe the relationship between salient objects or regions. As illustrated in Figure 5.1, model proposed by Anderson et al. [2018] with explicit salient region detection can clearly describe the salient objects in the image, e.g., children and cake. However, due to lack of context modelling for each salient region in the image, it fails to describe the relationship between salient objects, e.g., children are eating cake. Previous study in Chapter 4 focus on the subject in the sentence but lacks explicitly modeling of the predicate in the sentence. In this Chapter, we will study the ignored part in the previous Chapter by proposing a new model for image captioning.



A group of children sitting at a table with a cake

A man walking in the rain with an umbrella

Fig. 5.1 Images and generated captions from Anderson et al. [2018] which has no context modelling.

As illustrated in Figure 5.1, the main relationships that contribute to the predicate in sentence are either semantic (eating, holding, etc) or spatial (around, in, etc) relationships between salient objects. Without directly modelling these relationships between salient objects, it is difficult for a model to caption them only with the features of independent salient regions.

Scene graph [Xu et al., 2017] was first introduced to address the aforementioned problems, where an auxiliary model, e.g., visual relation detection model, is used to build the visual scene graph[1] in the image. This is followed by a graph convolutional neural network [Kipf and Welling, 2016] that contextualizes each region in the image. However, those approaches [Guo et al., 2019; Yang et al., 2019; Yao et al., 2018, 2019] usually require auxiliary models (e.g., visual relationship detection and/or attribute detection models) to build the visual scene graph in the image in the first place. In contrast, in the natural language processing field, the transformer architecture [Vaswani et al., 2017] was developed to contextualize embedded words in sentences, and can be trained end to end without auxiliary models that explicitly detecting such relations. Recent image captioning models [Herdade et al., 2019; Huang et al., 2019; Li et al., 2019a] adopted the transformer architectures to contextualize informative regions in the image through dot-product attention, and achieved state-of-the-art performance.

However, the transformer architecture was designed for machine translation of text. In a sentence, a word is either to the left or to the right of another word, with different distances. In contrast, images are two-dimensional (indeed, represent three-dimensional scenes), so that a region may not only be on the left or right of another region, it may also contain or be contained in another region (see Figure 5.2). The relative spatial relationship between the semantic units in images (salient regions) has a larger degree of freedom than that (words) in sentences. Furthermore, in the decoding stage of machine translation, a word is usually translated into another word in other languages (one to one decoding), whereas for an image region, we may describe its context, its attributes and/or its relationships with other regions (one to more decoding).

One limitation of previous transformer-based image captioning models [Herdade et al., 2019; Huang et al., 2019; Li et al., 2019a] is that they adopt the transformer's internal architecture designed for the machine translation, where each transformer layer contains a single (multi-head) dot-product attention module. In this paper, we introduce the ***image transformer*** for image captioning, where each transformer layer implements multiple sub-transformers, to encode spatial relationships between image regions and decode the diverse information in image regions.

[1] In the scene graph, nodes represent salient regions in the image. Edges represent relationships between salient regions.

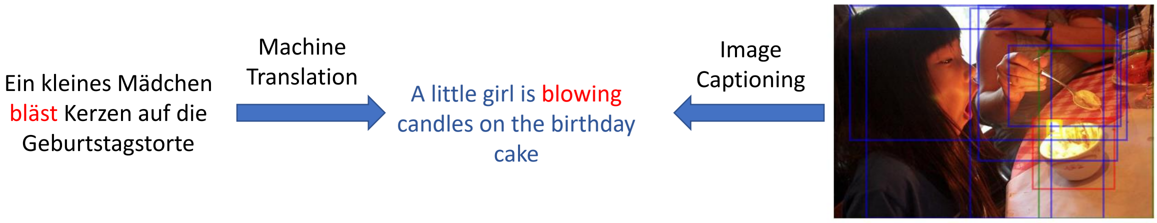

Ein kleines Mädchen bläst Kerzen auf die Geburtstagstorte
Machine Translation
A little girl is blowing candles on the birthday cake
Image Captioning

Fig. 5.2 Machine translation vs image captioning.

The difference between our method and previous transformer based models [Herdade et al., 2019; Huang et al., 2019; Li et al., 2019a] is that our method focuses on the *inner architectures* of the transformer layer, in which we widen the transformer module. Yao et al. [2019] used a hierarchical concept in the encoding part of their model, but our model ***focus on the spatial relationships for each query region***. Furthermore, our model does not require auxiliary models (i.e., for visual relation detection and instance segmentation) to build the visual scene graph. Our encoding method can be viewed as the combination of a visual semantic graph and a spatial graph which use a transformer layer that explicitly combine them without auxiliary relationship and attribute detectors.

The rest of the chapter is organized as follows: Sec. 5.2 reviews the related attention-based image captioning models; Sec. 5.3 introduces the standard transformer model and our proposed image transformer; Followed by the experiment results and analysis in Sec. 5.4; Finally, we will conclude this chapter in Sec. 5.5.

5.2 Background

We characterize current attention-based image captioning models into single-stage attention models, two-stages attention models, visual scene graph based models, and transformer-based models. We will review them one by one in this section.

5.2.1 Single-stage attention based image captioning

The attention mechanism we studied in chapter 4 is the single-stage attention. Single-stage attention-based image captioning models are the models where attention is applied at the decoding stage, i.e., the decoder attends to the most informative region [Luo et al., 2016] in the image when generating a corresponding word. We refer the reader to last chapter for more detailed methodology used in single-stage attention models for image captioning.

The single-stage attention models are computationally efficient, as their models only have two parts, i.e., a CNN for image feature processing and a language model with attention for sentence generation. However, as their models directly work on the regular image grid, they cannot accurately detect the salient regions that in most cases do not locate in the regular image grid for image captioning.

5.2.2 Two-stages attention based image captioning

Two stage attention models consist of *bottom-up* attention and *top-down* attention whereby bottom-up attention first uses object detection models to detect multiple informative regions in the image, then top-down attention attends to the most relevant regions when generating a word.

Instead of relying on the coarse receptive fields as informative regions in the image, as single-stage attention models do, Anderson et al. [2018] train the detection models on the *Visual Genome* dataset [Krishna et al., 2017]. The trained detection models can detect $10-100$ informative regions in the image. They then use a two-layers LSTM network as decoder, where the first layer generates a state vector based on the embedded word vector and the mean feature of the detected regions and the second layer uses the state vector from the previous layer to generate a weight for each detected region. The weighted sum of detected regions' feature is used as a context vector to predict next word. Lu et al. [2018] developed a similar network, but with a detection model trained on *MSCOCO* [Lin et al., 2014], which is a smaller dataset than *Visual Genome*, and therefore less informative regions are detected.

Our previous saliency based architecture for image captioning in section 4.5 can be also treated as two-stages attention based image captioning. However, the object detection based method is superior to our previous saliency based region perception due to the following aspects:

- Object detection based two-stages attention has direct supervision for detecting the accurate locations of informative regions in the image through the *visual genome* dataset, while the saliency based region perception can only predict coarse salient regions (sometimes, they're just part of objects, e.g., face of a person) on the regular grid of images.
- No feature fine tuning from other large scale dataset for the perceived regions' feature in saliency based region perception method.
- Object detection based attention gives more informative regions in the image (10-100 regions) compared to saliency based region perception (1-13 regions).

The two-stages attention based image captioning models perform better than single-stage attention based models. However, each detected region is independent to others. The contextual information of each region, a key part for image captioning, is missing in the two-stage attention based models.

5.2.3 Visual scene graph based image captioning

Visual scene graph based image captioning models extend two-stage attention models by using a graph convolutional neural network to contextualize all detected informative regions before feeding them into the decoder.

Yao et al. [2018] developed a model which consists of a semantic scene graph and a spatial scene graph. In the semantic scene graph, each region is connected with other semantically related regions. And the relationship between two connected regions is usually determined by a visual relationship detector among a union box. In the spatial scene graph, the relationship between two regions is defined by their relative positions. Then the feature of each node in the scene graph is refined with their related nodes through graph neural networks [Kipf and Welling, 2016]. Yang et al. [2019] used an auto-encoder, where they first encode the graph structure in the sentence based on the SPICE [Anderson et al., 2016] evaluation metric to learn a dictionary, then the semantic scene graph is encoded using the learnt dictionary. The previous two works treat the semantic relationships as edges in the scene graph, while Guo et al. [2019] treats them as nodes in the scene graph. Also, their decoder focuses on different aspects of a region. Yao et al. [2019] further introduces the tree hierarchy and instance level feature into the scene graph.

Introducing the graph neural network to contextualize informative regions yields a sizeable performance improvement for image captioning models, compared to two-stages attention models. However, it requires auxiliary models to detect and build the scene graph at first. A small detection error yields a different graph and thus contextualize each region with wrong information. Also those models usually have two parallel streams, one responsible for the semantic scene graph and another for spatial scene graph, which are computationally inefficient[2].

5.2.4 Transformer based image captioning

Transformer based image captioning models use dot-product attention mechanism [Vaswani et al., 2017] to contextualize informative regions.

Since the introduction of original transformer model [Vaswani et al., 2017], more advanced architectures were proposed for machine translation based on the structure or the natural characteristic of sentences, for example, the injection of explicit phrase information

[2] This will double the computation in the region contextualization process.

[Hao et al., 2019; Wang et al., 2019b] and tree structure transformer [Wang et al., 2019c]. In image captioning, AoANet [Huang et al., 2019] uses the original internal transformer layer architecture, with the addition of a channel attention based *gated linear layer* [Dauphin et al., 2017] on top of the multi-head attention. The object relation network [Herdade et al., 2019] injects the relative spatial attention into the dot-product attention, such that the attention weight between a query and a key is not only determined by their appearance feature but also their relative position. Another interesting result described by Herdade et al. [2019] is that the simple position encoding, in which a region's feature is augmented by simply concatenating its appearance feature with its spatial location feature (as proposed in the original transformer) did not improve image captioning performance. The entangled transformer model [Li et al., 2019a] features a dual parallel transformer to encode and refine visual and semantic information in the image, which is fused through a gated bilateral controller.

Compared to scene graph based image captioning models, transformer based models do not require auxiliary models to detect and build the scene graph at first, which is more computationally efficient. However current transformer based models for image captioning still use the inner architecture of the original transformer, designed for text, where each transformer layer has a single multi-head dot-product attention refining module. This structure does not allow to model the full complexity of relations between image regions. Therefore, we propose to change the inner architecture of the transformer layer to adapt it to image data. We widen the transformer layer, such that each transformer layer has multiple refining modules for different aspects of regions both in the encoding and decoding stages.

5.3 Image Transformer

In this section, we first review the original transformer layer [Vaswani et al., 2017], and then elaborate the encoding and decoding part for the proposed ***image transformer*** architecture.

5.3.1 Problem definition

In image captioning, given an image I, the goal is to train a model f_θ, parameterized by θ, which can generate a sentence $y_{1:T}$, with length T, to describe its input image I. Our proposed framework is illustrated in Figure 5.3. We will detail each part in section 5.3.3 and 5.3.4.

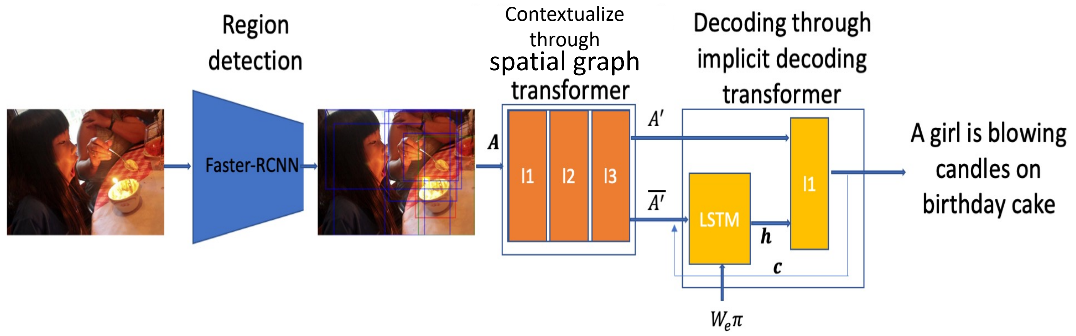

Region detection
Faster-RCNN
A
Contextualize through spatial graph transformer
l1
l2
l3
A'
$\overline{A'}$
Decoding through implicit decoding transformer
LSTM
l1
h
c
$W_e\pi$
A girl is blowing candles on birthday cake

Fig. 5.3 The overall architecture of our model. Faster-RCNN [Ren et al., 2015] first detects several informative regions (A) in the image. Those regions are then contextualized by a 3 layer (l1, l2, l3 in orange boxes) spatial graph transformer. The decoder (with one LSTM layer and one implicit decoding transformer layer (l1 in yellow box)) takes as input those contextualized regions to generate the sentence.

5.3.2 Transformer layer

A transformer consists of a stack of multi-head dot-product attention based transformer layers.

Transformer takes as input $A \in \mathbb{R}^{N\times D}$, which consists of N entries of D dimensions. In natural language processing, the input A can be the embedded features of words in a sentence, and in computer vision or image captioning, the input A is a set of features describing all regions detected by Faster-RCNN [Ren et al., 2015] in an image. The core operation of transformer is to contextualize each entry through multi-head dot-product attention with other entries. As per standard in Vaswani et al. [2017], each layer of a transformer first transforms its input A[3] into queries ($Q = AW_Q$, $W_Q \in \mathbb{R}^{D\times D_k}$), keys ($K = AW_K$, $W_K \in \mathbb{R}^{D\times D_k}$) and values ($V = AW_V$, $W_A \in \mathbb{R}^{D\times D_v}$) through 3 different linear transformations, i.e., W_Q, W_K and W_V. Then the scaled dot-product attention is applied:

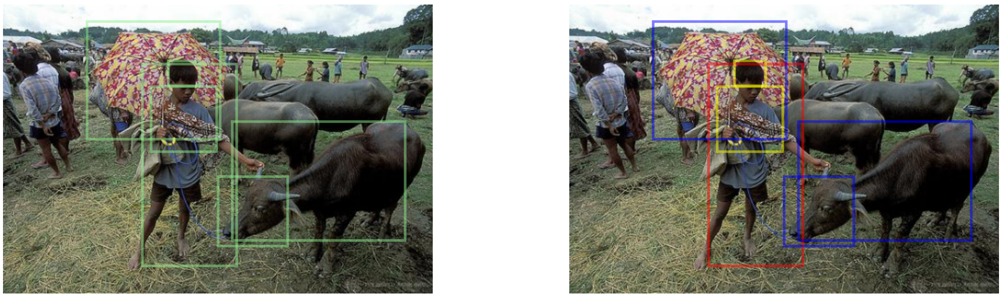
$$\text{Attention}(Q,K,V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{D_k}}\right)V, \tag{5.1}$$

where D_k is the dimension of the key vector and D_v the dimension of the value vector ($D = D_k = D_v$ in the implementation). To improve the performance of the attention layer, multi-head attention[4] is applied:

$$\begin{aligned} &\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\dots,\text{head}_h)W_O, \\ &\text{head}_i = \text{Attention}(AW_{Q_i},AW_{K_i},AW_{V_i}). \end{aligned} \tag{5.2}$$

[3] For simplicity, here we use first layer, which takes as input A , as example.

[4] Multi-head attention takes input as a split (in channel dimension) of original input. For example, AW_{Q_i} is the i_{th} split of Q.



(a) (b)

Fig. 5.4 (a) Image with detected regions; (b) An example of query region in the image (man in the red bounding box), and its neighbor regions (regions in blue bounding boxes, bull, umbrella, etc), child regions (regions in the yellow bounding boxes, hair,cloth). No parent regions for this query region, as it is not contained in any other detected region in the image.

Each head is composed of a split of the linear transformations (W_Q, W_K and W_V) and takes as input a slice of features (Q, K and V) respectively. The output from the multi-head attention is then added with the input, which is then normalized and yields an intermediate feature A_m:

$$A_m = \mathrm{Norm}(A + \mathrm{MultiHead}(Q, K, V)), \tag{5.3}$$

where $\mathrm{Norm}(\cdot)$ denote layer normalisation.

The transformer implements residual connections in each module, such that the final output of a transformer layer $A^{'}$ is:

$$A^{'} = \mathrm{Norm}(A_m + \phi(A_m W_f)), \tag{5.4}$$

where ϕ is a feed-forward network with non-linearity.

Each refining layer takes the output of its previous layer as input (the first layer takes the original input). The decoding part is also a stack of transformer refining layers, which take the output of encoding part as well as the embedded features of previous predicted word.

5.3.3 Spatial graph encoding transformer layer

In contrast to the original transformer, which only considers spatial relationships between query and key pairs as *neighborhood* and is thus limited to encode the 2-D spatial relationships in the image, we propose to use a spatial graph transformer in the encoding part, where we consider three common categories of spatial relationship for each query region in a graph structure: *parent*, *neighbor*, and *child* as shown in Figure 5.4. Thus we widen each transformer layer by adding three sub-transformer layers in parallel in each layer. Each sub-transformer responsible for a category of relationship, all sharing the same query. In the encoding stage, we define the relative spatial relationship between two regions based on their spatial overlap. We first compute the graph adjacent matrices $\Omega_p \in \mathbb{R}^{N\times N}$ (parent node adjacent matrix), $\Omega_n \in \mathbb{R}^{\in N\times N}$ (neighbor node adjacent matrix), and $\Omega_c \in \mathbb{R}^{\in N\times N}$ (child node adjacent matrix) for all regions in the image:

$$\begin{aligned}\Omega_p[l,m] &= \begin{cases} 1, \text{ if } \frac{\mathrm{Area}(l\cap m)}{\mathrm{Area}(l)} \geqslant \varepsilon \text{ and } \frac{\mathrm{Area}(l\cap m)}{\mathrm{Area}(l)} > \frac{\mathrm{Area}(l\cap m)}{\mathrm{Area}(m)} \\ 0, \text{ otherwise.}\end{cases} \\ \Omega_c[l,m] &= \Omega_p[m,l] \\ &\text{with} \sum_{i\in\{p,n,c\}} \Omega_i[l,m] = 1\end{aligned} \tag{5.5}$$

where $\varepsilon = 0.9$ in our experiment. The spatial graph adjacent matrices are used as the spatial hard attention embedded into each sub-transformer to combine the output of each sub-transformer in the encoder. More specifically, the original encoding transformer defined in Eqs. (5.1) and (5.2) are reformulated as:

$$\mathrm{Attention}(Q,K_i,V_i) = \Omega_i \circ \mathrm{Softmax}\left(\frac{QK_i^T}{\sqrt{d}}\right)V_i, \tag{5.6}$$

where, $\circ$ is the Hadamard product, and $i \in \{p,n,c\}$. The the original intermediate feature A_m is then reformulated as:

$$A_m = \mathrm{Norm}\left(A + \sum_{i\in\{p,n,c\}} \mathrm{MultiHead}(Q,K_i,V_i)\right). \tag{5.7}$$

As we widen the transformer, we thus halve the number of stacks in the encoder to achieve similar complexity as the original one (3 stacks, while the original transformer features 6 stacks). Note that the original transformer architecture is a special case of the proposed architecture, when no region in the image either contains or is contained by another.

5.3.4 Implicit decoding transformer layer

In natural language processing, transformer (with one sub-transformer in each layer) is usually used to translate a word into another word in other languages (one to one decoding). However, for an image region, we may describe its context, its attributes and/or its relationships with

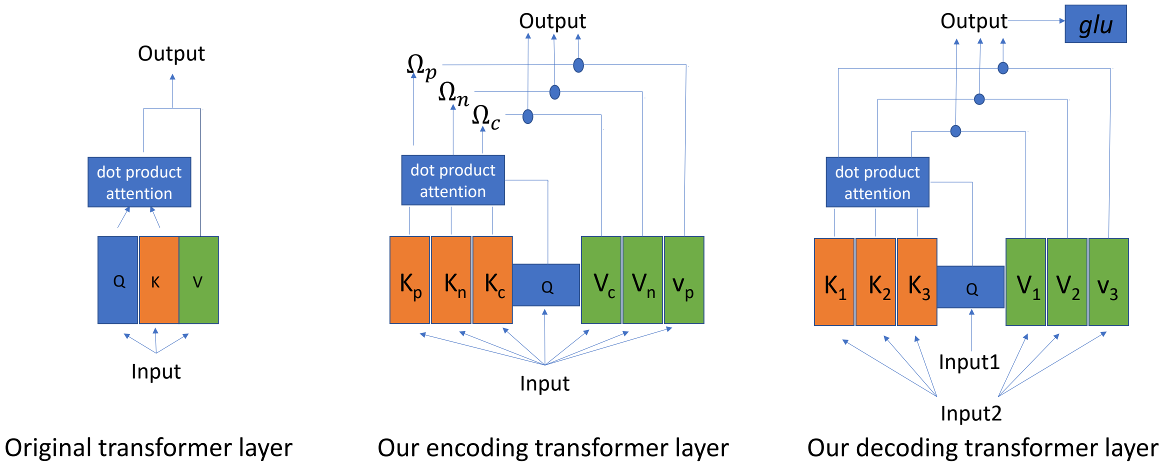


Fig. 5.5 The difference between the original transformer layer and the proposed encoding and decoding transformer layers.

other regions. Original transformer layer with only one sub-transformer is limited to decode the diverse information in each region.

Therefore, we propose a decoder consists of a LSTM [Hochreiter and Schmidhuber, 1997] layer and an implicit transformer decoding layer, which we proposed to decode the diverse information in a region in the image.

At first, the LSTM layer receives input x_t as the concatenation of summation of mean of the output ($\overline{A} = \frac{1}{N}\sum_{i=1}^{N} A_i^{'}$) from the encoding transformer and a context vector (c_{t-1}) at last time step and the embedded feature vector $W_e\pi_t$ of current word in the ground truth sentence:

$$\begin{aligned} x_t &= [W_e\pi_t, \overline{A} + c_{t-1}] \\ h_t, m_t &= \mathrm{LSTM}(x_t, h_{t-1}, m_{t-1}) \end{aligned} \tag{5.8}$$

Where, W_e is the word embedding matrix, π_t is the t^{th} word in the ground truth. The output state h_t is then transformed linearly and treated as the query for the input of the implicit decoding transformer layer. The difference between the original transformer layer and our implicit decoding transformer layer is that we also widen the decoding transformer layer by adding several sub-transformers in parallel in one layer, such that each sub-transformer can implicitly decode different aspects of a region. It is formalised as follows:

$$A_{t,i}^{D} = \mathrm{MultiHead}(W_{DQ}h_t, W_{DKi}A^{'}, W_{DVi}A^{'}) \tag{5.9}$$

where $A_{t,i}^{D}$ is the output from the i_{th} decoding sub-transformer. W_{DQ} is the linear transformation for query, shared by all sub-transformers. W_{DKi} and W_{DVi} are linear transformations for key and value in the $i_t h$ sub-transformer, respectively. Then, the mean of the sub-transformers' output is passed through a gated linear layer (GLU) [Dauphin et al., 2017] to extract the new context vector (c_t) at the current step by channel:

$$c_t = \text{GLU}\left(h_t, \frac{1}{M}\sum_{i=1}^{M} A_{t,i}^{D}\right) \tag{5.10}$$

The context vector is then used in the final prediction layer to predict the probability of word at time step t:

$$p(y_t|y_{1:t-1}) = \text{Softmax}(w_p c_t + b_p) \tag{5.11}$$

where w_p and b_p are weight and bias for the prediction layer.

The overall architecture of our model is illustrated in Figure 5.3, and the difference between the original transformer layer and our proposed encoding and decoding transformer layer is illustrated in Figure 5.5.

5.3.5 Training objectives

Given a target ground truth as a sequence of words $y_{1:T}^{*}$, for training the model parameters θ, we follow the previous method, such that we first train the model with cross-entropy loss:

$$L_{XE}(\theta) = -\sum_{t=1}^{T} \log(p_\theta(y_t^*|y_{1:t-1}^*)) \tag{5.12}$$

where $p_\theta(y_t^*|y_{1:t-1}^*)$ is given by equation 5.11. Then, we use self-critical reinforced training [Rennie et al., 2017] by optimizing the CIDEr score [Vedantam et al., 2015]:

$$L_R(\theta) = -E_{(y_{1:T}^s \sim p_\theta)}[r(y_{1:T}^s)] \tag{5.13}$$

where:

$$y_{1:T}^s = (y_1^s, \cdots, y_T^s) \tag{5.14}$$

and y_t^s is the sampled word at time step t from p_θ, $r(y_{1:T}^s)$ is the CIDEr score for the sampled sentence. The gradient is approximated by:

$$\bigtriangledown_\theta \approx -(r(y_{1:T}^s) - (\hat{y}_{1:T}))\bigtriangledown_\theta \log p_\theta(y_{1:T}^s) \tag{5.15}$$

where $r(y_{1:T}^s)$ is a single Monte carlo sample from p_θ and $\hat{y}_{1:T}$ is a baseline or reference.

5.4 Experiment

5.4.1 Datasets and evaluation metrics

Our model is trained on the MSCOCO image captioning dataset [Chen et al., 2015]. We follow Karpathy's splits [Karpathy and Fei-Fei, 2015], with 11,3287 images in the training set, 5,000 images in the validation set and 5,000 images in the test set. Each image has 5 captions as ground truth. We discard the words which occur less than 4 times, and the final vocabulary size is 10,369. We test our model on both Karpathy's offline test set (5,000 images) and MSCOCO online testing datasets (40,775 images). As per standard in previous works [Huang et al., 2019; Rennie et al., 2017], we use BLEU [Papineni et al., 2002], METEOR [Banerjee and Lavie, 2005], ROUGE-L [Lin, 2004], CIDEr [Vedantam et al., 2015], and SPICE [Anderson et al., 2016] as evaluation metrics.

5.4.2 Implementation details

Following previous work, we first train Faster R-CNN on Visual Genome [Krishna et al., 2017] use ResNet-101 [He et al., 2016] as backbone, pretrained on ImageNet [Deng et al., 2009]. For each image, we can detect $10-100$ informative regions, the boundaries of each are first normalised and then used to compute the graph adjacent matrices. We then train our proposed model for image captioning using the computed graph adjacent matrices and extracted features for each image region. We first train our model with *cross-entropy* loss for 25 epochs, the initial learning rate is set to 2×10^{-3}, and we decay the learning rate by 0.8 every 3 epochs. Our model is optimized through Adam [Kingma and Ba, 2014] with a batch size of 10. We then further optimize our model by reinforced learning for another 35 epochs. The size of the decoder's LSTM layer is set to 1024, and beam search of size 3 is used in the inference stage.

5.4.3 Experiment results

We compare our model's performance with published image captioning models. The compared models include the top performing single-stage attention model, Att2all [Rennie et al., 2017]; two-stages attention based models, n-babytalk [Lu et al., 2018] and up-down [Anderson et al., 2018]; visual scene graph based models, GCN-LSTM [Yao et al., 2018], AUTO-ENC [Yang et al., 2019], ALV [Guo et al., 2019], GCN-LSTM-HIP [Yao et al., 2019]; and transformer based models Entangle-T [Li et al., 2019a], AoA [Huang et al., 2019], VORN [Herdade et al., 2019]. The comparison on the MSCOCO Karpathy offline test set is

model	Bleu1	Bleu4	METEOR	ROUGE-L	CIDEr	SPICE
single-stage model						
Att2all [Rennie et al., 2017]	-	34.2	26.7	55.7	114.0	-
two-stages model						
n-babytalk [Lu et al., 2018]	75.5	34.7	27.1	-	107.2	20.1
up-down [Anderson et al., 2018]	79.8	36.3	27.7	56.9	120.1	21.4
scene graph based model						
GCN-LSTM* [Yao et al., 2018]	80.9	38.3	28.6	58.5	128.7	22.1
AUTO-ENC [Yang et al., 2019]	80.8	38.4	28.4	58.6	127.8	22.1
ALV* [Guo et al., 2019]	-	38.4	28.5	58.4	128.6	22.0
GCN-LSTM-HIP* [Yao et al., 2019]	-	39.1	28.9	**59.2**	130.6	22.3
transformer based model						
Entangle-T* [Li et al., 2019a]	**81.5**	**39.9**	28.9	59.0	127.6	22.6
AoA [Huang et al., 2019]	80.2	38.9	**29.2**	58.8	129.8	22.4
VORN [Herdade et al., 2019]	80.5	38.6	28.7	58.4	128.3	22.6
Ours	80.8	39.5	29.1	59.0	**130.8**	**22.8**

Table 5.1 Comparison on MSCOCO Karpathy offline test split. * means fusion of two models.

model	B1		B4		M		R		C	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
scene graph based model										
GCN-LSTM* [Yao et al., 2018]	80.8	**95.9**	38.7	69.7	28.5	37.6	58.5	73.4	125.3	126.5
AUTO-ENC* [Yang et al., 2019]	-	-	38.5	69.7	28.2	37.2	58.6	73.6	123.8	126.5
ALV* [Guo et al., 2019]	79.9	94.7	37.4	68.3	28.2	37.1	57.9	72.8	123.1	125.5
GCN-LSTM-HIP* [Yao et al., 2019]	**81.6**	**95.9**	39.3	71.0	28.8	38.1	59.0	74.1	**127.9**	**130.2**
transformer based model										
Entangle-T* [Li et al., 2019a]	81.2	95.0	38.9	70.2	28.6	38.0	58.6	73.9	122.1	124.4
AoA [Huang et al., 2019]	81.0	95.0	39.4	71.2	**29.1**	**38.5**	58.9	**74.5**	126.9	129.6
Ours	81.2	95.4	**39.6**	**71.5**	**29.1**	38.4	**59.2**	**74.5**	127.4	129.6

Table 5.2 Leaderboard of recent published models on the MSCOCO online testing server. * means fusion of two models.

illustrated in Table 5.1. Our model achieves new state-of-the-art on the CIDEr[5] and SPICE score, while other evaluation scores are comparable to the previous top performing models. Note that because most visual scene graph based models used both semantic and spatial scene graph, and require the auxiliary models to build the scene graph at first, our model is more computationally efficient. VORN [Herdade et al., 2019] also integrated spatial attention in their model, and our model performs better than them among all kinds of evaluation metrics,

[5] This is most important evaluation metric in the community, as the objective of the reinforced training is to maximize the CIDEr score.

which shows the superiority of our proposed spatial graph. The MSCOCO online testing results are listed in Tab. 5.2, our model outperforms previous transformer based model on several evaluation metrics.

5.4.4 Ablation study and analysis

model	Bleu1	Bleu4	METEOR	ROUGE-L	CIDEr	SPICE
baseline (AoA [Huang et al., 2019])	77.0	36.5	28.1	57.1	116.6	21.3
positions to embed our spatial graph encoding transformer layer						
baseline+layer1	77.8	36.8	28.3	57.3	118.1	21.3
baseline+layer2	77.2	36.8	28.3	57.3	118.2	21.3
baseline+layer3	77.0	37.0	28.2	57.1	117.3	21.2
baseline+layer1,2,3	77.5	37.0	28.3	57.2	118.2	21.4
effect of spatial relationships in the encoder						
baseline+layer1,2,3 w/o SP	77.5	36.8	28.2	57.1	117.8	21.4
number of sub-transformers in the implicit decoding transformer layer						
baseline+layer1,2,3 (M=2)	77.5	37.6	28.4	57.4	118.8	21.3
baseline+layer1,2,3 (M=3)	78.0	37.4	28.4	57.6	119.1	21.6
baseline+layer1,2,3 (M=4)	77.5	37.8	28.4	57.5	118.6	21.4

Table 5.3 Ablation study, results reported without RL training. baseline+layer1 means only the first layer of encoding transformer uses our proposed spatial graph transformer layer, other layers use the original one. M is the number of sub-transformers in the decoding transformer layer. SP means three spatial relationships in our proposed spatial graph.

In the ablation study, we use AoA [Huang et al., 2019] as a strong baseline[6] (with a single multi-head dot-product attention module per layer), which adds the gated linear layer [Dauphin et al., 2017] on top of the multi-head attention. In the encoder part, we study the effect of our proposed spatial graph in the encoder, where we ablate the three spatial relationships by simply taking the mean output of three sub-transformers in each layer by reformulating Eqs. 5.6 and 5.7 as:

$$\text{Attention}(Q, K_i, V_i) = \text{Softmax}\left(\frac{QK_i^T}{\sqrt{d}}\right) V_i \tag{5.16}$$

$$A_m = \text{Norm}\left(A + \frac{1}{3} \sum_{i \in \{p,n,c\}} \text{MultiHead}(Q, K_i, V_i)\right) \tag{5.17}$$

We also study where to use our proposed spatial graph encoding transformer layer in the encoding part: in the first layer, second layer, third layer or three of them? In the decoding

[6] Our experiments are based on the code released at: https://github.com/husthuaan/AoANet

part, we study the effect of the number of sub-transformers (M in Eq. 5.10) in the implicit decoding transformer layer.

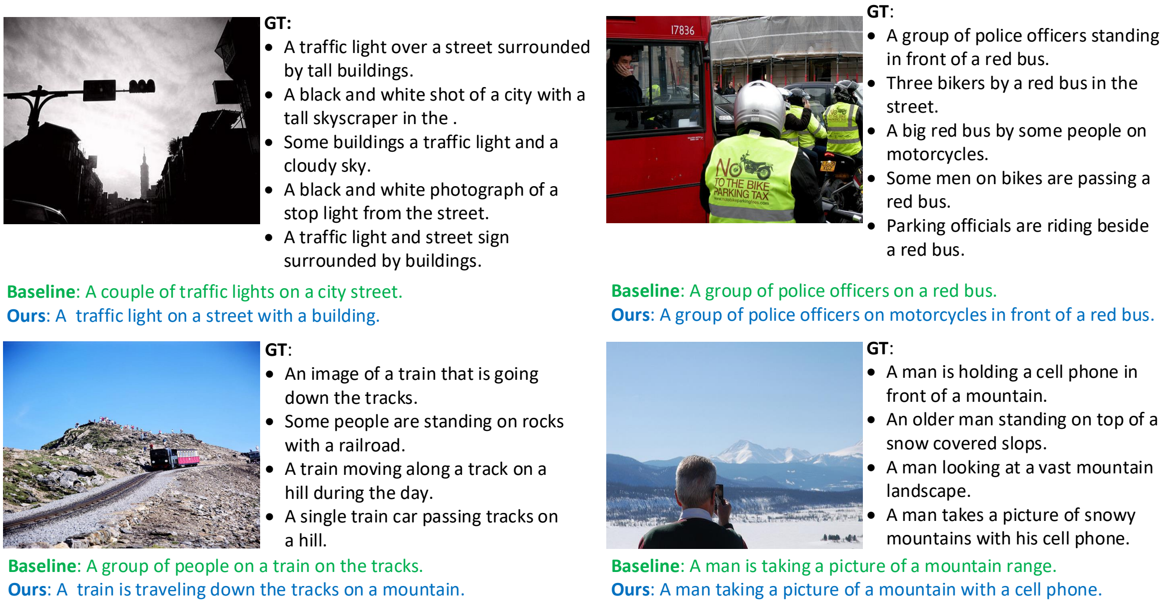
As we can see from Table 5.3, by widening the encoding transformer layer, there is a significant improvement on the model's performance. While not every layers in the encoding transformer are equal, when we use our proposed transformer layer at the top layer of the encoding part, the improvement was reduced. This may be because spatial relationships at the top layer of the transformer are not as informative. We thus use our spatial graph transformer layer at all layers in the encoding part. When we remove the spatial relationships in our proposed wider transformer layer, there is also some performance reduction, which shows the importance of the spatial relationships in our design. After widening the decoding transformer, the improvement was further increased (the CIDEr score increased from 118.2 to 119.1 after widening the decoding transformer layer with 3 sub-transformers. The magnitude of improvement is significant as per standard in previous works [Huang et al., 2019; Yao et al., 2018].). While not more wider gives better result, with 4 sub-transformers in the decoding transformer layer, there is some performance decrease. Therefore, the final design of our decoding transformer layer has 3 sub-transformers in parallel. The qualitative examples of our model are illustrated in Figure 5.6. As we can see, the baseline model without spatial relationships wrongly described the police officers on a red bus (top right), and people on a train (bottom left). And some qualitative examples of the decoder's attention, that is, when generating each word, the region with the highest dot-product attention weight with the query from LSTM layer in equation 5.9, are illustrated in Figure 5.7, our model's decoder can accurately attend to most relevant region when generating the corresponding word.

Number of stacks in the encoding part: How many layers we should to stack in the encoding part? Here, we give more experiment results with different number of the proposed spatial graph transformer layers.

Table 5.4 shows the model's performance for different number of stacks of our spatial graph transformer layer in the encoding part. We can see that 3 stacks perform best among all the experiments, which is our original design in the experiment.

Num of stacks	Bleu1	Bleu4	METEOR	ROUGE-L	CIDEr	SPICE
2	80.5	39.3	29.0	58.9	129.6	22.7
3	**80.8**	**39.5**	**29.1**	**59.0**	**130.6**	**22.8**
4	80.5	39.0	29.0	58.7	128.9	22.6

Table 5.4 The performance for different number of stacks of the proposed spatial graph transformer layer in the encoding part.



GT:
- A traffic light over a street surrounded by tall buildings.
- A black and white shot of a city with a tall skyscraper in the .
- Some buildings a traffic light and a cloudy sky.
- A black and white photograph of a stop light from the street.
- A traffic light and street sign surrounded by buildings.

Baseline: A couple of traffic lights on a city street.
Ours: A traffic light on a street with a building.

GT:
- A group of police officers standing in front of a red bus.
- Three bikers by a red bus in the street.
- A big red bus by some people on motorcycles.
- Some men on bikes are passing a red bus.
- Parking officials are riding beside a red bus.

Baseline: A group of police officers on a red bus.
Ours: A group of police officers on motorcycles in front of a red bus.

GT:
- An image of a train that is going down the tracks.
- Some people are standing on rocks with a railroad.
- A train moving along a track on a hill during the day.
- A single train car passing tracks on a hill.

Baseline: A group of people on a train on the tracks.
Ours: A train is traveling down the tracks on a mountain.

GT:
- A man is holding a cell phone in front of a mountain.
- An older man standing on top of a snow covered slops.
- A man looking at a vast mountain landscape.
- A man takes a picture of snowy mountains with his cell phone.

Baseline: A man is taking a picture of a mountain range.
Ours: A man taking a picture of a mountain with a cell phone.

Fig. 5.6 Qualitative examples from our method on the MSCOCO image captioning dataset [Chen et al., 2015], compared against the ground truth annotation and a strong baseline method (AoA [Huang et al., 2019]).

Wider decoding transformer layer or more heads: We used 3 sub-transformers in the implicit decoding transformer, and each sub-transformer has 8 heads inside it, thus 24 heads in total in the implicit decoding transformer layer. In this part, we check if we can achieve the same performance by simply adding 24 heads in a single transformer layer. Table 5.5 shows the experiment results, there is a clear gap between a single transformer layer with 24 heads and our implicit decoding transformer layer with 3 sub-transformers.

model	Bleu1	Bleu4	METEOR	ROUGE-L	CIDEr	SPICE
ours	80.8	39.5	29.1	59.0	130.6	22.8
24 heads	80.1	38.5	28.8	58.6	127.9	22.4

Table 5.5 Performance comparison between our proposed implicit decoding transformer layer (single transformer with 3 sub-transformer layer in parallel, each sub-transformer has 8 heads) and single transformer layer with 24 heads.

Encoding graph visualisation: The transformer layer can be seen as an fully connected graph, which contextualizes the informative regions through dot-product attention. Here we visualise how our proposed spatial graph transformer layer learns to connect the informative regions through attention in Figure 5.8. In the top example, the original transformer layer strongly relates the train with the people on the mountain, which yields wrong description,

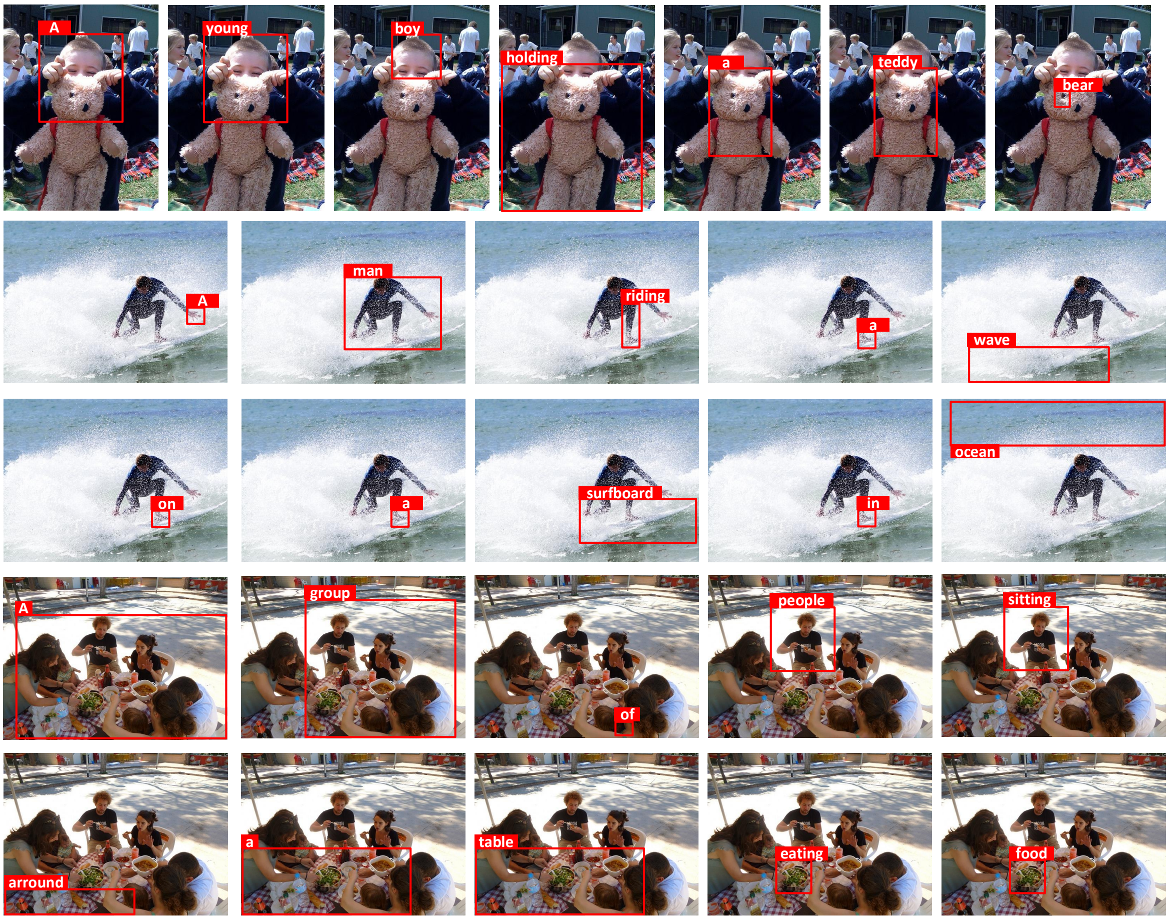


Fig. 5.7 Qualitative examples for the decoder's attention. The region in the red bounding box is the region which has the highest dot-product attention weight with the query from the the LSTM layer.

while our proposed transformer layer relates the train with the tracks and mountain. In the bottom example, the original transformer relates the bear with its reflection in water and treats them as 'two bears', while our transformer can distinguish the bear from its reflection and relate it to the snow area.

Decoding feature space visualisation: We also visualised the output of our decoding transformer layer (Figure 5.9). Compared to the original decoding transformer layer that only has one sub-transformer inside it, the output of our proposed implicit decoding transformer layer covers a larger area in the reduced feature space than the original one, which means that our decoding transformer layer decoding more information in the image regions. In the original feature space (1,024 dimensions) from the output of decoding transformer layer, we compute the trace of the feature maps' co-variance matrix from 1,000 examples. The

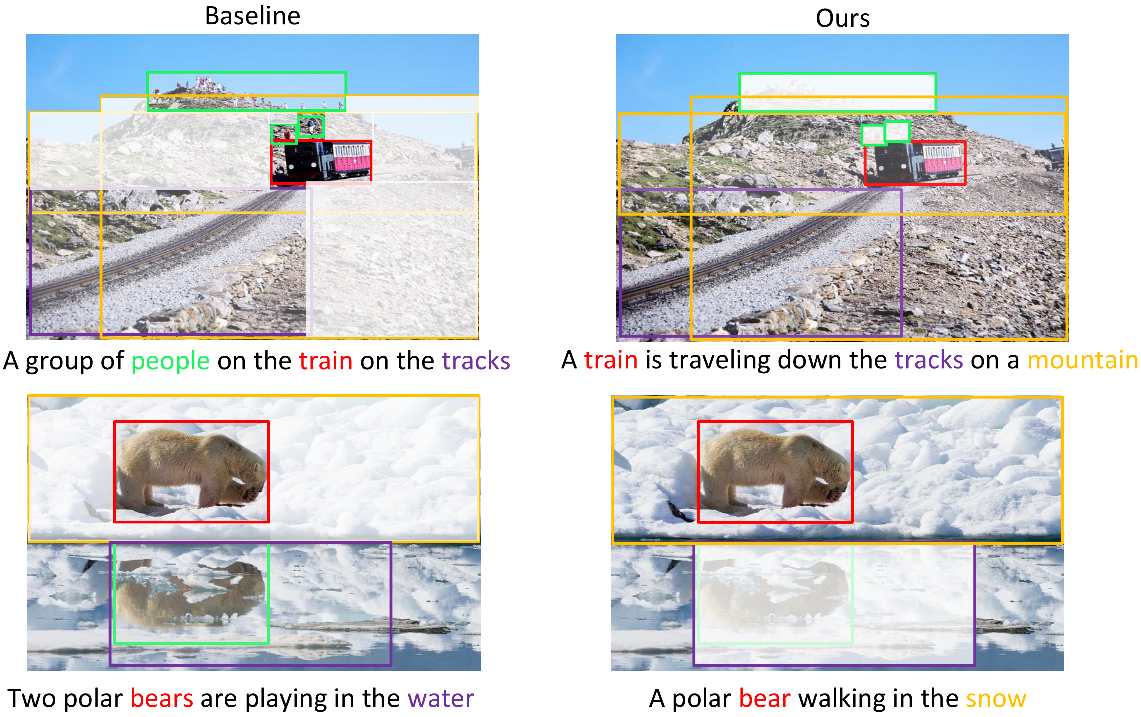
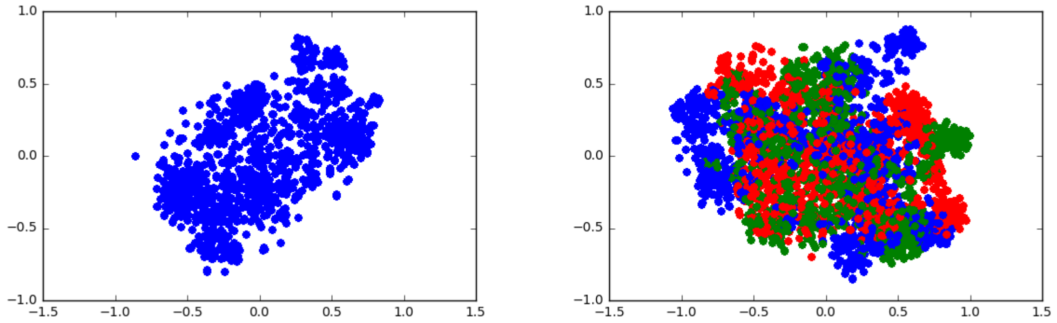

Baseline
Ours
A group of people on the train on the tracks
A train is traveling down the tracks on a mountain
Two polar bears are playing in the water
A polar bear walking in the snow

Fig. 5.8 A visualization of how the query region relates to key regions through attention in transformer layer. The region in the red bounding box is the query region and other regions are key regions. The transparency of each key region shows its dot-product attention weight with the query region. Higher transparency means larger dot-product attention weight, vice versa.

trace for original transformer layer is 30.40 compared to 454.57 for our wider decoding transformer layer, which indicates that our design enables the decoder's output to cover a larger area in the feature space. However, it looks like individual sub-transformers in the decoding transformer layer still do not learn to disentangle different factors in the feature space (as there is no distinct cluster from the output of each sub-transformer), we speculate this is because we have no direct supervision to their output, which may not able to learn the disentangled feature automatically [Locatello et al., 2019].



(a) original

(b) ours

Fig. 5.9 t-SNE [Maaten and Hinton, 2008] visualisation of the output from decoding transformer layer (1,000 examples), different color represent the output from different sub-transformers in the decoder in our model.

5.5 Discussion and Conclusion

In this chapter, we introduced the ***image transformer*** architecture for image captioning task. Different from previous transformer based image captioning models, which simply borrow the internal transformer architecture that can only encode the "left or right" spatial relation for text. The core idea behind the proposed architecture is to widen the original transformer layer, designed for machine translation, to adapt it to the structure of images that has more complex spatial relationships. In the encoder, we widen the transformer layer by exploiting the three common spatial relationships between image regions, i.e., neighbor (next to other regions), child (be contained in other regions) or parent (contain other regions). Each sub-transformer encodes one spatial relationship and each pair of region only has one spatial relationship. In the decoder, the wider transformer layer that can decode more information (e.g., category, attribute, relationship) in the image regions.

Extensive experiments were done to show the superiority of the proposed model. The qualitative and quantitative results were illustrated in the experiments to validate the proposed encoding and decoding transformer layer. Compared to the previous top models in image captioning, our model achieves a new state-of-the-art CIDEr and SPICE score, while in the other evaluation metrics, our model is either comparable or outperforms the previous best models, with a better computational efficiency compared to the graph-based method (a single stream which encodes both semantic and spatial graph with half number of stacks (3 stacks) than original transformer (6 stacks)).

The proposed model is the first one that tries to adapt the architecture of transformer into image domain. It proves that encoding diverse information (either 2D spatial relationships with other regions or diverse semantic information for each region) in transformer layer is a crucial design for images, which gives some insights for future research when applying transformer in images.

Chapter 6

Conclusion and Future Perspectives

In conclusion, this thesis studied bottom-up attention, top-down attention and contextual attention in computer vision in the deep learning era.

Our analysis for deep saliency prediction models revealed that a deep neural network pre-trained for image recognition already encodes some visual saliency in the image. Fine-tuning this pre-trained model for saliency prediction produces a model with uneven response to saliency categories, e.g., neurons sensitive to textual input start attending more to human head. We showed that although deep models do capture synthetic pop-out stimuli within their inner layers, they fail to predict such salient patterns in their output, contrary to classical models of saliency prediction.

We introduced a novel, relatively large dataset consisting of synchronized multi-modal attention and caption annotations. We revisited the consistency between human attention and image captioning models on our collected data, and demonstrated that the top-down soft-attention mechanism used by automatic captioning systems captures neither spatial locations nor the temporal properties of human attention during captioning. Also, the similarity between human and machine attention has no bearing on the quality of the machine generated captions. We also reconfirmed the strong relationship between described objects and attended ones, similar to the findings that have been observed in free-viewing experiments. We proposed an image captioning model, which combines image saliency with soft attention. It demonstrated significant performance improvement to the purely top-down soft attention approach.

We proposed image transformer architecture to contextualize the regions in the image based on their spatial relationships. The proposed architecture widens the original transformer layer, designed for machine translation, to adapt it to the structure of images. In the encoder, we widen the transformer layer by exploiting the spatial relationships between image regions. In the decoder, we widen transformer layer such that it can decode more information in each region in the image. Extensive experiments were done and showed the superiority of the proposed model.

How far are we from having an effective attention model for a computer vision system? We are still far from an effective attention model for a computer vision system. Over the last few years, the attention based deep computer vision systems have evolved from the previous soft and hard attention based models to the object detection or region detection based system. Significant performance improvement [Anderson et al., 2018] has been achieved with this evolvement. Early soft attention follows the human visual attention mechanism in the way that they are sequential process, while the performance of these soft attention based models are inferior to object or region detection based attention models. Although superior performance compared to soft attention models, the object or region detection model itself has some limitations. Its detection performance is highly dependent on the dataset it trained on and the architectures we used. For the current object or region detection based attention models, the detected objects or regions are still not accurate enough. In the meanwhile, from the perspective of human attention system, in which the attentive process is a sequential process with memory, object or region detection is in parallel without memory.

In the future, from the computational perspective, although imperfect, we still believe hard attention or region detection will dominate the attention based computer vision system, which pre-extracts several informative regions in the static image or video clips for a lot of computer vision tasks. Below, based on our previous experience in the research on attention, we discuss some potential research directions for attention in computer vision.

Applications of saliency prediction model in computer vision Current deep learning based saliency prediction has progressed a lot, although they have some limitations as we studied in chapter 3. However, only predicting saliency map of an image on its own is meaningless in computer vision. A gray scale saliency map gives us nothing for image understanding. Future research on saliency prediction should shed more light on the application of image's saliency map in computer vision. For example, as the saliency map of an image indicates the most salient regions in image, how can image's saliency map help some downstream task for image understanding (recognition, object detection, image captioning and visual question answering, etc) ?

Efficiency problem in the differentiable hard attention for natural images Current differentiable hard attention are spatial transformer based method, no matter the original spatial transformer network [Jaderberg et al., 2015; Recasens et al., 2018] or the saliency

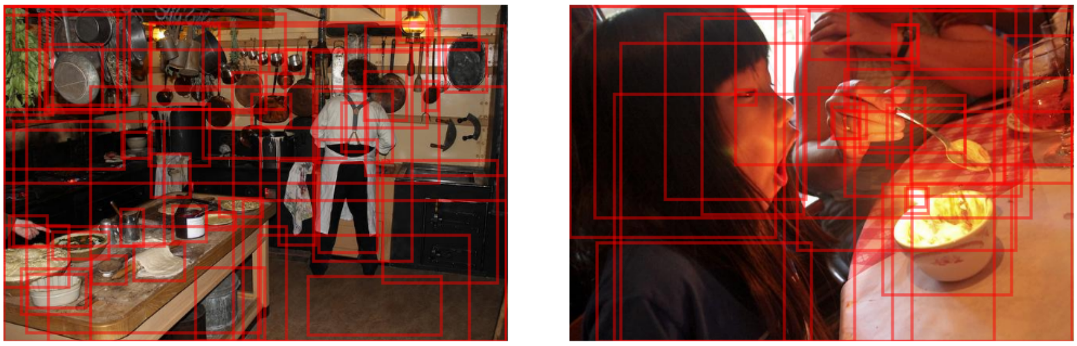


Fig. 6.1 Some region detection examples frequently used in image captioning and visual question answering at the moment.

region perception approach we proposed in chapter 4. There exists a severe problem which hinder this differentiable hard attention approach to the natural images. It usually requires a "region cropping" network to crop an image's region and another feature extraction network to extract the features of the "cropped" regions. This design doubles the requirements for both computation and memory of a model. A better design of differentiable hard attention is required for both model's efficiency and training. For example, ROI pooling/align proposed by He et al. [2017]; Ren et al. [2015] can be used to replace the feature extraction network. In this way, however, hard attention would turn from region cropping in the image space to feature map cropping in the deep feature space. It therefore loses its original meaning.

Accurate and general region detection. Current bottom-up attention has been dominated by popular detection models, e.g., faster-rcnn [Ren et al., 2015], which are trained on large scale dataset with rich annotated semantic categories, such as *Visual Genome* dataset. It has achieved great success in image captioning and visual question answering. However, a problem behind this is the low detection accuracy due to the large amount of region categories in the annotated dataset and large amount of annotated regions in each image. Examples are illustrated in Figure 6.1. As reported by Anderson et al. [2018], the region detection accuracy is only 10.2% for such cases. When those detected regions are provided to the downstream tasks, two problems are raised. First, a lot of regions are noise regions, which will increase the learning difficulty for the downstream tasks; Second, when feed such a large number of detected regions to the downstream tasks, the computation of a model would be increased. Therefore, a more accurate and generic[1] detection algorithm is required for such kind of

[1] generic means more semantic categories which are not restricted to common objects, e.g., 80 object classes in MSCOCO [Lin et al., 2014].

region detection model trained on dataset with rich semantic annotations that has thousands of categories.

References

Subutai Ahmad. Visit: A neural model of covert visual attention. In *Advances in neural information processing systems*, pages 420–427, 1992.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017.

Ali Borji. Saliency prediction in the deep learning era: An empirical investigation. *arXiv preprint arXiv:1810.03716*, 2018.

Ali Borji. Saliency prediction in the deep learning era: Successes and limitations. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):185–207, 2013.

Ali Borji, Dicky N Sihite, and Laurent Itti. Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Transactions on Image Processing*, 22(1):55–69, 2012.

Ali Borji, Hamed R Tavakoli, Dicky N Sihite, and Laurent Itti. Analysis of scores, datasets, and models in visual saliency prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 921–928, 2013.

Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A benchmark. *IEEE transactions on image processing*, 24(12):5706–5722, 2015.

Neil Bruce and John Tsotsos. Saliency based on information maximization. In *Advances in neural information processing systems*, pages 155–162, 2005.

Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. Mit saliency benchmark, 2012. http://saliency.mit.edu/results_mit300.html, Last accessed on 2020-03-19.

Zoya Bylinskii, Adrià Recasens, Ali Borji, Aude Oliva, Antonio Torralba, and Frédo Durand. Where should saliency models look next? In *European Conference on Computer Vision*, pages 809–824. Springer, 2016.

Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6298–6306. IEEE, 2017.

Shi Chen and Qi Zhao. Boosted attention: Leveraging human attention for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 68–84, 2018.

Xinlei Chen and Abhinav Gupta. Spatial memory for context reasoning in object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4086–4096, 2017.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Predicting Human Eye Fixations via an LSTM-based Saliency Attentive Model. *IEEE Transactions on Image Processing*, 27(10):5142–5154, 2018.

Antonio Cuevas, Manuel Febrero, and Ricardo Fraiman. An anova test for functional data. *Computational statistics & data analysis*, 47(1):111–122, 2004.

Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. Human attention in visual question answering: Do humans and deep networks look at the same regions? *Computer Vision and Image Understanding*, 163:90–100, 2017.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 933–941. JMLR, 2017.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380, 2014.

Santosh K Divvala, Derek Hoiem, James H Hays, Alexei A Efros, and Martial Hebert. An empirical study of context in object detection. In *2009 IEEE Conference on computer vision and Pattern Recognition*, pages 1271–1278. IEEE, 2009.

Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

Carolina Galleguillos and Serge Belongie. Context based object categorization: A critical survey. *Computer vision and image understanding*, 114(6):712–722, 2010.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Longteng Guo, Jing Liu, Jinhui Tang, Jiangwei Li, Wei Luo, and Hanqing Lu. Aligning linguistic words and visual semantic units for image captioning. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 765–773, 2019.

Jie Hao, Xing Wang, Shuming Shi, Jinfeng Zhang, and Zhaopeng Tu. Multi-granularity self-attention for neural machine translation. *arXiv preprint arXiv:1909.02222*, 2019.

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

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

Christopher Healey and James Enns. Attention and visual memory in visualization and computer graphics. *IEEE transactions on visualization and computer graphics*, 18(7): 1170–1188, 2012.

Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. In *Advances in Neural Information Processing Systems*, pages 11135–11145, 2019.

HNB-13b. Hnb-13b eye tracker. http://ilab.usc.edu/itrack/, Last accessed on 2020-03-24.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *2007 IEEE Conference on computer vision and pattern recognition*, pages 1–8. Ieee, 2007.

Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4634–4643, 2019.

Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 262–270, 2015.

Laurent Itti and M. A. Arbib. Attention and the minimal subscene. In M. A. Arbib, editor, *Action to Language via the Mirror Neuron System*, pages 289–346. Cambridge University Press, Cambridge, U.K., 2006.

Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001.

Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20 (11):1254–1259, 1998.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

Saumya Jetley, Naila Murray, and Eleonora Vig. End-to-end saliency mapping via probability distribution prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5753–5761, 2016.

Saumya Jetley, Nicholas A Lord, Namhoon Lee, and Philip HS Torr. Learn to pay attention. *arXiv preprint arXiv:1804.02391*, 2018.

Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. Salicon: Saliency in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1072–1080, 2015.

Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *2009 IEEE 12th international conference on computer vision*, pages 2106–2113. IEEE, 2009.

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer, 1987.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

Srinivas SS Kruthiventi, Kumar Ayush, and R Venkatesh Babu. Deepfix: A fully convolutional neural network for predicting human eye fixations. *IEEE Transactions on Image Processing*, 26(9):4446–4456, 2017.

Matthias Kümmerer, Thomas SA Wallis, and Matthias Bethge. Information-theoretic model comparison unifies saliency metrics. *Proceedings of the National Academy of Sciences*, 112(52):16054–16059, 2015.

Matthias Kummerer, Thomas S. A. Wallis, Leon A. Gatys, and Matthias Bethge. Understanding low- and high-level contributions to fixation prediction. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Matthias Kümmerer, Thomas SA Wallis, Leon A Gatys, and Matthias Bethge. Understanding low-and high-level contributions to fixation prediction. In *2017 IEEE International Conference on Computer Vision*, pages 4799–4808, 2017.

Matthias Kummerer, Thomas SA Wallis, and Matthias Bethge. Saliency benchmarking made easy: Separating models, maps and metrics. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 770–787, 2018.

Olivier Le Meur, Patrick Le Callet, and Dominique Barba. Predicting visual fixations on video based on low-level visual features. *Vision research*, 47(19):2483–2498, 2007.

Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. Entangled transformer for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8928–8937, 2019a.

Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10313–10322, 2019b.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

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

Nian Liu and Junwei Han. A deep spatial contextual long-term recurrent convolutional network for saliency detection. *IEEE Transactions on Image Processing*, 27(7):3264–3274, 2018.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning-Volume 97*, pages 4114–4124. JMLR, 2019.

Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 6, page 2, 2017.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7219–7228, 2018.

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in neural information processing systems*, pages 4898–4906, 2016.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

Emiel Miltenburg, Ákos Kádár, Ruud Koolen, and Emiel Krahmer. Didec: The dutch image description and eye-tracking corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3658–3669, 2018.

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

Nabil Ouerhani and Heinz Hugli. Computing visual attention from scene depth. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 1, pages 375–378. IEEE, 2000.

Junting Pan, Elisa Sayrol, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E O'Connor. Shallow and deep convolutional networks for saliency prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 598–606, 2016.

Junting Pan, Cristian Canton Ferrer, Kevin McGuinness, Noel E O'Connor, Jordi Torres, Elisa Sayrol, and Xavier Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. *arXiv preprint arXiv:1701.01081*, 2017.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Robert J Peters, Asha Iyer, Laurent Itti, and Christof Koch. Components of bottom-up gaze allocation in natural images. *Vision research*, 45(18):2397–2416, 2005.

Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1): 79–87, 1999.

Adria Recasens, Petr Kellnhofer, Simon Stent, Wojciech Matusik, and Antonio Torralba. Learning to zoom: a saliency-based sampling layer for neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 51–66, 2018.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

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

Michael J Swain and Dana H Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

Michael K Tanenhaus, Michael J Spivey-Knowlton, Kathleen M Eberhard, and Julie C Sedivy. Integration of visual and linguistic information in spoken language comprehension. *Science*, 268(5217):1632–1634, 1995.

Benjamin W Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of vision*, 7(14):4–4, 2007.

Hamed R Tavakoli, Fawad Ahmed, Ali Borji, and Jorma Laaksonen. Saliency revisited: Analysis of mouse movements versus fixations. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 1774–1782, 2017a.

Hamed R. Tavakoli, Rakshith Shetty, Ali Borji, and Jorma Laaksonen. Paying attention to descriptions generated by image captioning models. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017b.

Hamed R Tavakoli, Rakshith Shetty, Ali Borji, and Jorma Laaksonen. Paying attention to descriptions generated by image captioning models. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2506–2515. IEEE, 2017c.

Simon Thorpe, Denis Fize, and Catherine Marlot. Speed of processing in the human visual system. *nature*, 381(6582):520–522, 1996.

Tobii. Tobii eye tracker. https://www.tobii.com, Last accessed on 2020-03-24.

Antonio Torralba. Contextual modulation of target saliency. In *Advances in neural information processing systems*, pages 1303–1310, 2001.

Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136, 1980.

John K Tsotsos. On the relative complexity of active vs. passive visual search. *International journal of computer vision*, 7(2):127–141, 1992.

John K Tsotsos. *A computational perspective on visual attention.* MIT Press, 2011.

Preethi Vaidyanathan, Emily T Prud'hommeaux, Jeff B Pelz, and Cecilia O Alm. Snag: Spoken narratives and gaze dataset. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 132–137, 2018.

Rufin VanRullen and Simon J Thorpe. Is it a bird? is it a plane? ultra-rapid visual categorisation of natural and artifactual objects. *Perception*, 30(6):655–668, 2001.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Collecting image description datasets using crowdsourcing. *arXiv preprint arXiv:1411.3041*, 2014.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2017a.

Wenguan Wang and Jianbing Shen. Deep visual attention prediction. *IEEE Trans. Image Process*, 27(5):2368–2378, 2018.

Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, and Haibin Ling. Salient object detection in the deep learning era: An in-depth survey. *arXiv preprint arXiv:1904.09146*, 2019a.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. Self-attention with structural position representations. *arXiv preprint arXiv:1909.00383*, 2019b.

Yau-Shian Wang, Hung-Yi Lee, and Yun-Nung Chen. Tree transformer: Integrating tree structures into self-attention. *arXiv preprint arXiv:1909.06639*, 2019c.

Yixiu Wang, Bin Wang, Xiaofeng Wu, and Liming Zhang. Scanpath estimation based on foveated image saliency. *Cognitive processing*, 18(1):87–95, 2017b.

Yulin Wang, Kangchen Lv, Rui Huang, Shiji Song, Le Yang, and Gao Huang. Glance and focus: a dynamic approach to reducing spatial redundancy in image classification. *arXiv preprint arXiv:2010.05300*, 2020.

Sean Welleck, Jialin Mao, Kyunghyun Cho, and Zheng Zhang. Saliency-based sequential image attention with multiset prediction. In *Advances in neural information processing systems*, 2017.

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

Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419, 2017.

Juan Xu, Ming Jiang, Shuo Wang, Mohan S Kankanhalli, and Qi Zhao. Predicting human gaze beyond pixels. *Journal of vision*, 14(1):28–28, 2014.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10685–10694, 2019.

Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 684–699, 2018.

Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Hierarchy parsing for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2621–2629, 2019.

Alfred L Yarbus. *Eye movements and vision*. Springer, 2013.

Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

Kiwon Yun, Yifan Peng, Dimitris Samaras, Gregory J Zelinsky, and Tamara L Berg. Studying relationships between human gaze, description, and computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 739–746, 2013.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015.

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