Modeling Human Visual Attention and Distraction in Visual Search Tasks

Manoosh Samiei

Department of Electrical and Computer Engineering

McGill University, Montreal

July, 2021

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of

M.Sc. in Electrical Engineering

Abstract

Most studies in computational modeling of visual attention are focused on task-free observation of images. In a free-viewing scenario observers are mostly driven by stimuli-based saliency, caused by surprising/informative features of an image. However, free-viewing saliency considers very limited scenarios of humans' behavior in daily life. Most visual activities that humans perform daily are goal-oriented and demand a great amount of top-down attention control. Visual Search is a simple task that requires observers to look for a target object in a scene. Searching for a product on a supermarket shelf, searching for keys in a drawer, and looking for cheese in a fridge are examples of visual search in daily life. Compared to free-viewing saliency, visual search demands more top-down control of attention. In this thesis, we present two approaches to model humans' visual attention behavior during visual search. Our first approach predicts fixation density maps, the probability of eye fixation over pixels of the search image, using a two-stream encoder-decoder network. This method predicts which locations are more distracting when searching for a particular target. The second method aims to detect distractors and targets in search images at an object-level. Distractors are all the objects other than the target that the observers fixate during their search. We use Mask-RCNN to segment distractors and targets in the search images. COCOSearch18 is our main dataset for training our networks in both methods. This dataset is composed of fixation data of 10 observers while searching for 18 object categories in several images from the MS COCO dataset. We perform an analysis of this dataset to determine how different features of the target object such as size, eccentricity, and category can affect the search performance. Both of our methods achieve notable performance in modeling human visual search behavior. Our models reveal that the fixated items/locations during visual search are not random and could be predicted

by deep learning models. These fixated items can reveal some information regarding the visual attention policy of human brain during visual search.

Abrégé

La plupart des études sur la modélisation informatique de l'attention visuelle se concentrent sur l'observation d'images sans tâche. Dans un scénario de visionnage libre, les observateurs sont principalement motivés par la saillance basée sur les stimuli, causée par les caractéristiques surprenantes/informatives d'une image. Cependant, la saillance en visionnage libre considère des scénarios très limités de comportement humain dans la vie quotidienne. La plupart des activités visuelles que les humains effectuent quotidiennement sont axées sur des objectifs et nécessitent un contrôle de l'attention descendant. La recherche visuelle est une tâche simple qui oblige les observateurs à rechercher un objet cible dans une scène. La recherche d'un produit sur une étagère de supermarché, la recherche de clés dans un tiroir et la recherche de fromage dans un réfrigérateur sont des exemples de recherche visuelle dans la vie quotidienne. Par rapport à la saillance en visionnage libre, la recherche visuelle exige un contrôle de l'attention plus descendant. Dans cette thèse, nous présentons deux approches pour modéliser le comportement d'attention visuelle des humains lors de la recherche visuelle. Notre première approche prédit les cartes de densité de fixation, la probabilité de fixation de l'œil sur les pixels de l'image de recherche, en utilisant un réseau encodeur-décodeur à deux flux. Cette méthode prédit quels emplacements sont les plus gênants lors de la recherche d'une cible particulière. La deuxième méthode vise à détecter les distracteurs et les cibles dans les images de recherche au niveau de l'objet. Les distracteurs sont tous les objets autres que la cible que les observateurs fixent lors de leur recherche. Nous utilisons Mask-RCNN pour segmenter les distracteurs et les cibles dans les images de recherche. COCOSearch18 est notre principal ensemble de données pour former nos réseaux aux deux méthodes. Cet ensemble de données est composé de données de fixation de 10 observateurs lors de la recherche

de 18 catégories d'objets dans plusieurs images de l'ensemble de données MS COCO. Nous effectuons une analyse de cet ensemble de données pour déterminer comment différentes caractéristiques de l'objet cible telles que la taille, l'excentricité et la catégorie peuvent affecter les performances de recherche. Nos deux méthodes atteignent des performances remarquables dans la modélisation du comportement de recherche visuelle humaine. Nos modèles révèlent que les éléments/emplacements fixés lors de la recherche visuelle ne sont pas aléatoires et pourraient être prédits par des modèles d'apprentissage en profondeur. Ces objets fixés peuvent révéler des informations concernant la politique d'attention visuelle du cerveau humain lors de la recherche visuelle.

Acknowledgements

I would like to express my gratitude to my master's supervisor, Professor James Clark, for his insightful feed backs and inspirations throughout my research work. Your invaluable expertise guided me to find my research path. I appreciate your trust in my ideas, and the freedom you gave me in exploring my research interests.

I am grateful to Professor Michael Langer for his support and inspirations through Computational Perception course. I gained valuable knowledge from you.

I would like to thank my family, my parents, and my sibling, for their endless support and kindness through out my master's research.

I would like to acknowledge my friends and lab mates for helping me to stay motivated during the difficult times of pandemic.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Visual perception allows organisms to interact with their environment, satisfy their survival needs, and avoid potential dangers. The importance of vision can be validated by the fact that more than 50 percent of the cortex, the surface of the brain, is devoted to processing visual information [31]. Visual perception however has its limitations. Full processing of all the incoming visual information is intractable (estimated to be on the order of 10^7–10^8 bits per second at the optic nerve [51]). Selective attention provides organisms with the capacity to direct their gaze toward their object of interest to attend to only small parts of their visual environment that are more important to them, reducing the computational demand of vision. Moreover, humans and other animals have foveated visual systems which means that only a small central part of the retina has high receptor density allowing for the detailed perception of a scene. Hence making eye movements is essential for high-resolution perception of object of interest. Selective attention does not always involve eye movements. It is possible that observers attend to objects in the periphery of their visual field. Shift of focus in the absence of eye-movements is called 'covert attention' as opposed to 'overt attention' which involves eye movements [51]. However, covert attention is not completely puzzled. Some studies [43] suggest that microsaccades, or tiny eye movements that take place during periods of eye fixation, show the directions of covert attention shifts due to the subliminal activation of the oculomotor system. Overt and covert attention are interrelated. It is revealed by several studies that a shift in covert attention precedes an eye movement or shift in

overt attention [46]. Studying visual attention mechanisms involves recording eye movements of observers as they view an image or perform a task such as playing a game, cooking, cleaning, etc.

Visual attention has been studied under two major categories: bottom-up and top-down. Bottom-up attention is believed to be guided by low-level stimuli-based characteristics and involves conspicuity of the stimuli [24]. For instance, when we look at an image in which a red basket lies between a group of green baskets, we will fixate our eyes on the red basket because of its conspicuity. In top-down attention, the eyes are more in the control of high-level goal-oriented perception [24]. For example, when driving, regardless of how conspicuous each element of a scene is, a responsible driver looks at parts of the scene that are important to her driving decisions. Bottom-up and top-down attention are combined in the control of gaze. Considering the behavioral patterns of people in high-perceptual demand tasks such as driving, scientists claim that as the computational demands of a task increases, top-down attention dominates the low-level bottom-up attention.

To date, most research has been focused on understanding bottom-up attention by attempting to predict where people look when they are free-viewing a set of images. These studies are referred to as saliency prediction. Little research has been done on understanding the more important top-down element of attention that involves more complex goal-oriented tasks. One simple task that can be used to study top-down attention is visual search. In a visual search task, observers are asked to search for an object in an image and their eye movements are recorded.

Due to the modeling complexity associated with the combination of top-down and bottom-up attention in guiding eye movements, predicting human's eye fixations during visual search has not been well-studied. Furthermore, scarcity of public datasets for visual search tasks makes it difficult to employ state of the art deep learning approaches for visual search modeling. Fixations during visual search could be caused by spatial search, image-based conspicuity, personal preference, relevance/similarity to the target object (e.g., having similar color, shape, or position in a scene), etc. Additionally, the restrictions of the task i.e. looking for an object, limit the freedom associated with eye movements compared to a free-viewing scenario, and hence the fixated regions should be less scattered and more focused around/on the target objects.

The main purpose of our research is to model how observers are distracted to different locations/items in the scene while searching for an object. In our modeling, we call all the non-target objects/regions as 'distractors'. We hypothesize that fixated distractors contain valuable information regarding the visual attention policy of human brain.

In this thesis, we present two approaches. Our first method is pixel-based and generates fixation density maps for each search image. This method uses a two-stream encoder-decoder convolutional neural network. One stream of this network receives a sample target image from a target category, and the other receives the search image containing the target. Each stream contains a convolutional neural network that extracts features from the input image. The extracted features of the two streams are then convolved together and passed through a decoder to generate a target-specific fixation density map for the given search image.

Our second approach is object-based and predicts the distractor and target objects during visual search. This method uses a Mask-RCNN segmentation network pre-trained on Ms COCO and fine-tuned on COCOSearch18 dataset. Our second method also presents a way to measure the distraction score of each item.

The rest of the thesis is organized as follows. Chapter 2 discusses some background information and previous works in the field of visual attention and modeling. It first briefly describes two types of eye movements that are important for studying selective attention. Then it discusses important theories on visual search, selective attention, and distraction from the Psychology field. It also discusses some work around the foraging task, a more complex variant of visual search. The rest of the related work is devoted to the computational modeling of attention. The discussed models are studied under three major categories depending on the visual task being performed. First free-viewing saliency models are presented, second visual-search models, and finally interactive task models.

Chapter 3 presents a detailed description of our proposed solutions. This chapter first provides some analysis on the dataset. It provides insights into how different target features affect search performance, as well as individual differences in visual search among participants. After dataset analysis, we discuss our methods. For each method, we describe the data pre-processing step,

model architecture, training and technical details, performed experiments, and a discussion on the results, along with shortcomings, strengths, and applications of the method.

Finally, chapter 4 concludes our thesis with some final remarks, summary of our results, and directions for future research.

Chapter 2

Background and Related Work

2.1 Eye Movements

When looking at two dimensional scenes, saccadic eye movements and their associated fixations play the most important roles in our perception, and are of major research interest. Thus in this section, we briefly discuss each of these two components.

2.1.1 Saccade

A saccade is a rapid eye movement that moves the gaze from one location to another location. During a saccade, both eyes move in the same direction. Saccades bring the image of the target locations in the visual environment onto the fovea for a detailed perception. Saccades can be triggered voluntarily or involuntarily. Planning a saccade has a latency in the range of 100-1000 milliseconds depending on the task. [23] The average duration of a saccade is 20-40 milliseconds. Larger saccades produce longer durations; thus, the amplitude of a saccade and its duration are linearly correlated. [23] Moreover, when the eye is moving, the end point of a saccade cannot be changed. Saccades can occur in straight line or curved trajectories. When an observer makes a saccadic movement toward a target location, the saccade might not accurately land on that location, and either overshoot or undershoot it. Hence, subsequent small corrective saccades are made to place the location of interest on the fovea. [88]

Since the eyes jump rapidly during saccades the image received by the fovea is of low quality; and therefore it has been often believed that we do not 'see' during saccadic eye movements. This is known as saccadic suppression. [88] Therefore to perceive the visual environment we use a combination of fixations and saccades.

2.1.2 Fixation

A fixation is referred to the maintaining of the eyes on one location for a certain period of time between the saccades. Fixations allow for processing the details of the visual environment. The average duration of fixations typically varies in the range of 50-600 milliseconds, and can go beyond this range depending on the stimuli. [23] The fixation durations are important since they indicate which locations are being attended and how long they engage observers' attention. The recorded fixation duration is dependent on the sampling rate of the eye tracking device and the technique being used. [88]

The fixation duration includes both the time required for processing the foveal information as well as planning the location of the next fixation. Both the foveal and peripheral information contribute to planning the next saccade and fixation location. [88]

2.2 Visual Search

Visual search is a perceptual task that involves scanning the visual environment for a particular object or feature (the target) among other objects or features (the distractors) by means of visual attention. [102] Some examples of visual search in daily life are: searching for a product on a supermarket shelf, looking for a friend's number in a phone book, trying to find a person in a large crowd of people, or different games and puzzles such as 'Spot the Difference' and 'Where's Waldo'.

Early studies in visual search tried to discover the underlying mechanisms behind the two tasks of separable feature search ('disjunctive' or 'efficient' search) and conjunction search ('inefficient' or 'serial' search). An example of separable feature search is finding a green X among a set of

red Xs, where target and distractors are different in only one feature that is their color (figure 2.1). Observers are capable of finding the target shortly regardless of the number of distractors. This phenomenon is called the 'pop-out' effect. In pop-out, the target object stands out from surrounding distractors due to its unique features. [4] An example of conjunction search is searching for a green X among a group of green Os and red Xs (figure 2.1). Here we have two sets of distractors and each share one feature (color or shape) with the target. The experiments showed that observers attend to a series of locations before finding the target, and have a significantly longer response time compared to the separable feature search. In conjunction search, the response time increases as the number of distractors increases.

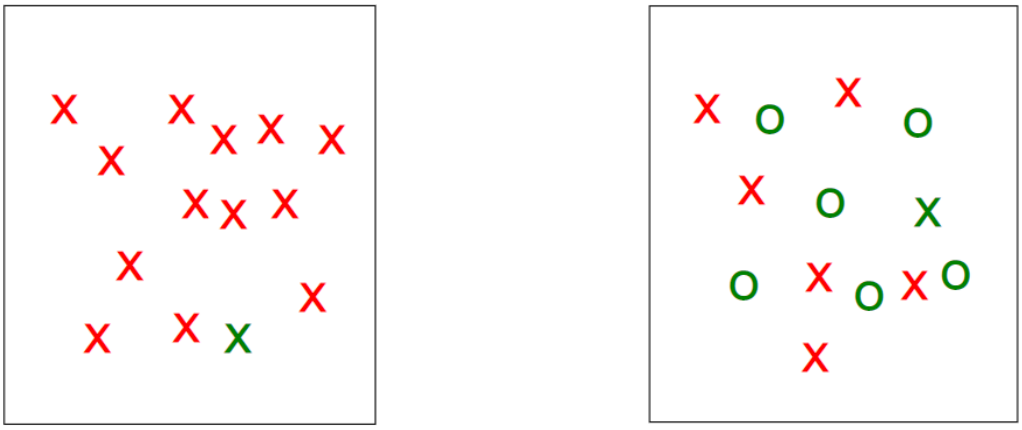
There are two popular theories: 'Feature Integration Theory' [102] and 'Guided Search' [106], that explains how visual attention operates during a conjunction search compared to a feature search.

2.2.1 Feature Integration Theory

In 1980, Treisman and Gelade proposed feature integration theory. According to this theory, "attention must be directed serially to each stimulus in a display whenever conjunctions of more than one separable feature are needed to characterize or distinguish the possible objects presented [102]." In other words, features are registered early, in a parallel manner; while objects are registered later in a serial way.

Treisman's theory introduces two different stages for visual attention process. The early preattentive stage encodes a scene to a number of separable feature dimensions (feature maps) such as orientation, color, spatial frequency, brightness, and direction of movement. If a feature is unique to a display, then it will be the only active location in its feature map. This permits pop out to occur, because the location of the unique, primitive feature is preattentively available. Unique combinations of features do not produce unique activity in a single feature map, and therefore cannot pop out. In a later attentive stage, which requires serial processing of the scene through focal attention, the separate features of each object are integrated to form a single object (saliency/master map). Therefore,"focal attention provides the glue which integrates the initially separable features into

unitary objects [102].” Conjunction search is mostly done through the slower attentive stage since it requires a set of feature dimensions to be integrated.



(a) Feature Search **(b)** Conjunction Search

Figure 2.1: Examples of feature and conjunction search.

2.2.2 Guided Search

Guided search 2.0 proposed by Wolfe (1994) [106], suggests that both bottom-up (stimuli-based) and top-down (goal-based) activation are combined in directing attention during the pre-attentive stage. Then based on this information, different locations of a scene are ranked in the order of their priority and attention is directed toward the elements with highest priority. If the fixated location does not contain the target, the next highest priority item is fixated and the process continues until the target is found. Guided search theory suggests that the priority map (activation map) that indicates the most likely location(s) to contain the target is constructed during the parallel stage based on the features (both bottom-up and top-down). In the next attentive stage, the peaks of the map are attended until the target is found.

During efficient search such as pop-out, the target generates the highest peak in the priority map and is found instantly, in a parallel way. During inefficient serial search such as when distractors share one or more features with the target, the generated peak by the target does not surpass the activation peak of the distractors. For instance when we are searching for a green X among a set

of green Os and red Xs, all green objects and all Xs produce high activation peaks in the priority map. Thus, these peaks need to be serially attended until the target is found. This can also explain why in conjunction search the response time increases as the number of distractors increases.

Both guided search and FIT have some strengths and weaknesses in explaining visual search. One of the major strengths of FIT is the scientific evidence from patients with Balint's syndrome. Treisman conducted an experiment with a Balint's syndrome sufferer who was not able to focus attention on items in the scene. When the person was showed simple stimuli such as a 'blue O' and a 'red T', he reported seeing a 'red O' and a 'blue T.' [35] This finding validated feature integration theory's prediction that a lack of focused attention would cause error in integrating features.

However, FIT has some weaknesses. For instance, some studies suggest that parallel and serial search processing are not separate [5]. FIT also does not account for faster responses in some conjunction task such as triple conjunctions of form, size and color, compared to double conjunctions with two features [107]. It also does not account for slower search when distractors are increased.

Guided search model was introduced later than FIT and probably tried to address some problems associated with FIT. The guided search model explains why certain conjunction searches may be fast or slow and why feature searches may also be slow. It accounts for distractors and solves the problem of the relevance of feature information from the parallel phase. Its key weakness is its inability to use real-life patients to demonstrate its findings like Treisman did. However in general, there is no solid evidence on which one of the two models is completely correct. [53]

2.3 Filter Theories of Selective Attention

Early studies in attention attempted to explain how much we process the unattended stimuli and how specific stimuli reach to our awareness. For this purpose several filter models have been introduced.

The first filter model named early selection model was proposed by Broadbent. Broadbent discovered that people were aware of the physical characteristics of an unattended auditory/visual stimuli such as the pitch of a voice or the color and size of a visual message. However, they were

unaware of the semantic meaning of the unattended stimuli. He came up with a model which filters the (irrelevant) information at an early stage and only the information that is passed through the filter (attended to) is processed for its meaning. The processed information then enters the short-term memory. [36] The selection of which stimuli should be passed through the filter is guided through attention. If an stimuli is attended for its relation to a task, top-down attention is employed; however if the features of a stimulus gained attention, bottom-up attention is employed. [38] Broadbent's model is shown in figure 2.2.

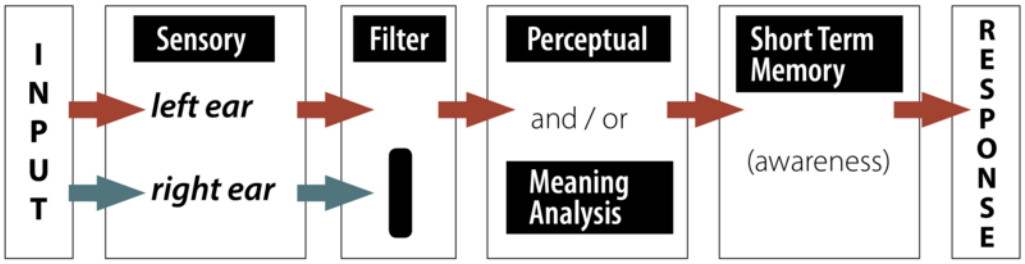


Figure 2.2: Broadbent's model. People are asked to only attend to the massage in their left ear, and ignore their right ear. Image credit: [36]

Broadbent's model cannot explain why listeners can hear and respond to their name when it is spoken in the unattended ear, even when they are fully engaged in a conversation and their name is only pronounced once. Since the unattended information is not processed semantically, they should not notice their name in the unattended ear. [36]

Treisman, a PhD student of Broadbent's, [101] carried out a number of listening experiments in which she presented two different stories to the two ears. She asked the attendees to only listen to the story in one ear. During the experiment, Treisman frequently switched the stories to the opposite ear. She discovered that people unconciously followed the story when it shifted from the attended ear to the unattended ear. Then they realized its the wrong ear and switched back to the correct ear. The fact that people tend to hear meaningful information even when they are not paying attention to it, suggest that they monitor the unattended information to some degree on the basis of its meaning. [36]

Therefore, claiming that unattended information is completely filtered at an early stage cannot be true. Treisman suggested that stimuli can be filtered either at physical level or at the meaning level. She stated that instead of a filter, there is an attenuator. The unattended information is not completely blocked at the physical level, it is just attenuated and perceived with less strength. Therefore, if the information received by the unattended ear is pertinent such as the subject's name, it will pass through the filter and enter the meaning analysis stage. Figure 2.3 shows information going in both ears. Selection of the left ear information strengthens that material, while the non-selected information in the right ear is weakened. However, if the preliminary analysis shows that the non-selected information is significant, then the Attenuation Control will instead strengthen the more meaningful information. [36]

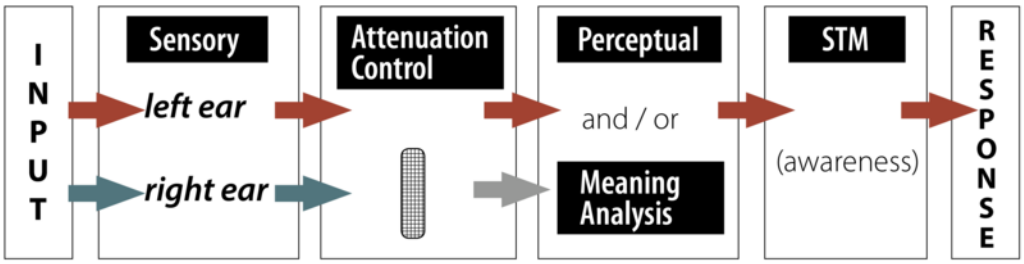


Figure 2.3: Treisman's Attenuation Model. Image credit: [36]

In 1963, Deutsch and Deutsch [27] introduced a late selection model, suggesting that filtering of the selected information happens after all information (relevant/irrelevant) is processed for meaning. However, only the attended information enters the short term memory and we become aware of it. This theory claims that all information is attended to, consciously or unconsciously. Then internal decisions of stimuli relevance must be made, before allowing it to gain conscious awareness. This model is consistent with ideas of subliminal perception; "in other words, we do not have to be aware of or attending a message for it to be fully processed for meaning [36]." Figure 2.4 shows Deutsch model of attention.

Gray and Wedderburn [42] conducted an experiment which validated the late selection model. They presented a mixture of numbers and words to each ear of the participants. For instance, the left ear received words 'Dear – 7 – Jane', and the right ear received '9 – Aunt – 6'. Then

they asked the participants to report what they heard. According to the early selection model, participants should have first reported all items presented to one ear, and then the items presented to the other ear. However, participants reported hearing, 'Dear Aunt Jane' and '9 – 7 – 6'. This study suggested that stimuli are not selected based on physical characteristics (e.g., location of sound) determined by the filter but according to their semantic meaning.

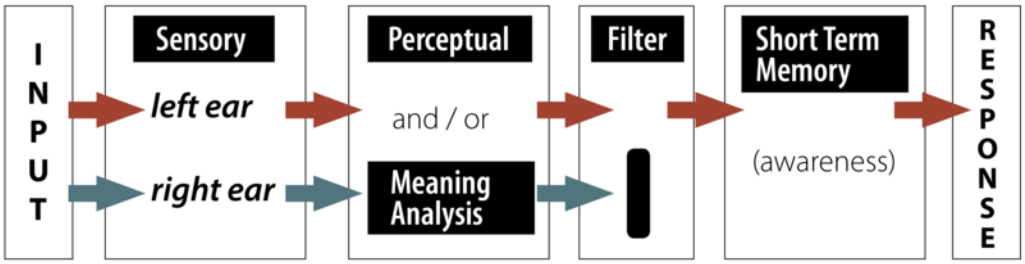


Figure 2.4: Late selection model by Deutsch and Deutsch. Image credit: [36]

2.4 Distraction

There are inconsistencies around the terminology of 'distractor' in visual search literature. Influential papers such as feature integration theory and guided search define a distractor as the item(s) that you are not looking for, and which distracts you from finding the target, and a target as the item that you need to find. In these papers, all the non-target items on the search display are referred to as distractors. We will use this definition for our distractor modeling in the Methodology section.

On the other hand, some papers make a distinction between non-target items and distractors. Based on the APA dictionary of psychology [1], a 'distractor' is a stimulus or an aspect of a stimulus that is irrelevant to the task or activity being performed. Based on this definition, non-target items on the search display in a visual search task are not distractors due to their relevance to the task of finding the target. However, an external stimuli such as a person playing music in the room where observers are performing visual search is considered irrelevant and distracting. When performing visual search in natural scenes; the non-target objects might provide contextual guidance for the target and even facilitate the search.

We argue that the inconsistency around distractor's definition arises from the definition of the task. Consider a scenario where a shopper is searching for pasta on a supermarket shelf and suddenly a special offer on a bag of chips absorbs his attention. He decides to buy a bag of chips besides pasta. If we define buying pasta as the target action and anything else as distraction, we can say that the customer is distracted by the offer. However, if we define shopping as the target task, then paying attention to the offer is not a distraction; while paying attention to a text massage popped up on the shopper's phone is a distraction. Hence, the definition of what is distracting is closely connected to the scope of the task. As we consider a broader scope for the task, the definition of distraction changes accordingly. In a visual search task, if we define finding the target item as the main task, finding any non-target item can be a distraction; while if we define searching among the search display for the target as the task, then only stimuli outside the search display are distracting.

2.4.1 Perceptual Load and Distraction

In 1973, Kahneman introduced 'The Capacity Model of Attention' which accounts for divided attention rather than selective attention. Based on this theory, attention is a mechanism that requires energy and mental effort; and for more complex mental tasks, greater mental effort is needed. "A capacity theory deals with three central questions:

1. What makes an activity more or less demanding? (momentary mental effort)
2. What factors control the total amount of capacity available at any time? (attentional capacity)
3. What are the rules of the allocation policy?

[57]"

Kahneman's model incorporates the bottom-up and top-down components of attention in its allocation policy. [34]

Later in 1995, Lavie [71] introduced the perceptual load theory of attention, which uses the same concepts of limited attentional capacity to explain selective attention, and solve the early versus late selection controversy. Her suggestion was that the perceptual load presented by the tasks that were competing for attention would determine the level of selection. This theory assumes

that for any task, the perceptual system uses all the capacity available at any given time. When the task load exceeds the capacity, we can predict early selection of relevant stimuli, where irrelevant/distracting stimuli are filtered out at an early stage. The extreme case of this early selection would be as the Broadbent filter, figure 2.2. On the other hand, when the load is low, the late selection of relevant stimuli occurs. The extreme late selection case would be similar to the Deutch and Deutch filter, figure 2.4. Perceptual Load theory is compatible with Treisman's Attenuation model that proposes a more flexible filtering which can be done either at the early physical level or late deeper processing (meaning) level. Perceptual load could be a component that controls the attenuater to allow for early or late selection of the information.

Lavie concludes that "irrelevant distractors are simply not processed when relevant processing consumes full perceptual capacity, and no specific mechanism is required for this early selectivity beyond the intrinsic capacity limitations of perception [71]."

It it worth noting that Lavie's definition of irrelevant distractors are those stimuli that are placed outside the search array and do not have any relevance to the search task. The non-target items that are in the search array are not considered distractors. She states in her later work in 1997 that "it is questionable whether non-targets in such visual search tasks are strictly irrelevant. Our main question is: What determines the efficiency of rejection for distractors that are entirely irrelevant (such as distractors placed well outside the current search area)? [72]"

In 1997, Lavie [72] performed another experiment that showed efficient visual search leads to inefficient distractor rejection. She asked a number of participants to look for target letter X or N in each central circle on a screen and press designated computer keys. In the easy search, the central nontargets were five Os. In the hard search, the central nontargets were heterogeneous letters that, like the target, were all angular (K, H, V, Z, W). A single peripheral distractor appeared 1.4° away from the nearest central letter on the left or right. There were three types of distractors: incompatible, neutral, and compatible. The incompatible distractor was a different target letter compared to the target letter on the central circle. The neutral distractor was a completely different letter compared to the two target letters, and the compatible distractor was exactly the same letter as the appeared target in the central circle. Participants were asked to ignore these irrelevant distractors.

Figure 2.5 shows several example displays in this experiment.

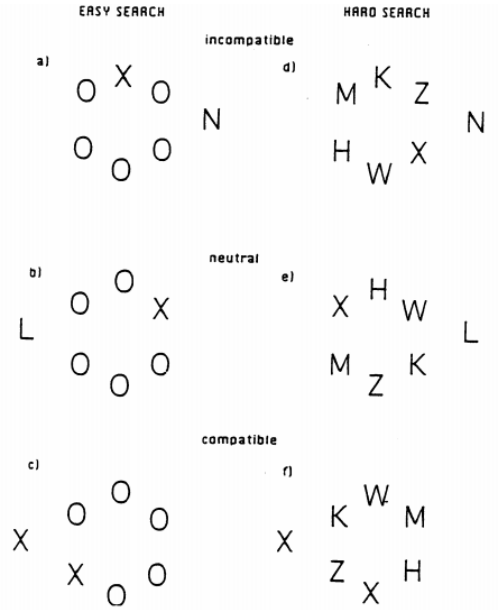


Figure 2.5: Example displays in Lavie's experiment. Image credit: [72]

Lavie deduced that when participants performed an efficient search in which a target was detectable by a distinct features from the non-targets (such as an angular character among a set of curved characters), a larger interference by the irrelevant peripheral distractors was observed, however during inefficient search where the target and non-targets were similar in their features, peripheral distractors interfered less with the task. [72]

An objection to this theory was proposed by Eltiti et al [30]. They suggested that the prominence of the distractor is the primary factor in causing distraction, rather than the perceptual load. In other words, interference may even occur in high-load displays if the distractor is more salient. There are other opposing theories to Lavie's load theory, but their discussion is out of the scope of this thesis.

Another popular study on interference is 'The Stroop Effect', named after John Ridley Stroop after publishing 'Studies of Interferences in Serial Verbal Reactions' [95]. In this paper, he investigated the interference between naming ink colors in color-words and the words themselves. He conducted three experiments, using three different stimuli as seen in figure 2.6. In the first experi-

ment, he asked participant to read the color-words written in black (stimulus 1 in 2.6), and to read color-words written in different colors independently of their color (stimulus 2 in 2.6). He deduced that "it took an average of 2.3 seconds longer to read color-words printed in colors different from that named by the word than to read the same names printed in black. [95]"

In the second experiment, he asked participants to name the color of the color-words independently of the written word (stimulus 2 in 2.6) and also to name the color of squares (stimulus 3). He realized that "the mean time for naming colors 74 percent increased in the presence of word stimuli compared to squares. [95]"

The third experiment tested participants at different stages of practice across the three different stimuli to account for the effects of association.

Stroop also noted that it took participants longer to name the colors of mismatched color-words in the second experiment compared to reading the written (mismatched) color-words in experiment 1. Naming the ink color of mismatched color-words has greater mental interference because we unintentionally determine the semantic meaning of a word, but must intentionally inhibit such a response to correctly name the color of the word.

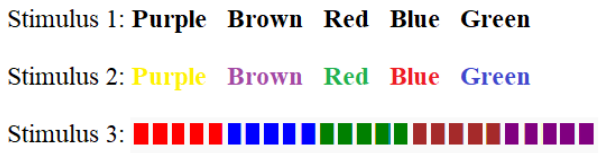


Figure 2.6: Stroop effect stimuli.

One determinant feature that affects distractors' interference in a visual task is their similarity to the target. Distractors that share some of the target's defining property are referred to as 'congruent', and those that are different from the target are 'incongruent'. Some studies indicate that congruent distractors cause a higher interference compared to incongruent distractors. For instance, in [13] congruent distractors slowed down saccadic reaction times more than incongruent distractors. Moreover, "the more participants had to rely on the target property to locate the target, the larger the congruency effect [13]."

A popular study by Navalpakkam and Itti [80] proposes a feature gain modulation theory that combines information from both the target and distracting clutter to maximize the relative salience of the target. They suggest that when the target feature is similar to the distractor feature, the optimal strategy is to boost a neuron tuned to the exaggerated target feature and not the exact target feature, as neurons tuned to the exact target feature respond to the distractor as well.

Heterogeneity (dissimilarity) of distractors is another effective component in visual search. Duncan and Humphreys suggest in their paper [28] that when non-targets (distractors) are homogeneous, search efficiency is not largely affected by the display size; while in case of searching through heterogeneous distractors, display size has a very large effect on the search performance, especially during target-absent trials. These results with heterogeneous distractors resemble those reported by feature integration theory for conjunction search. [28]

Some factors that are important in studying the strength of distractors are proposed by Horstmann et al. [47]. They suggest that search components such as skipping: effective ignoring of distractors (that do not show target features), dwelling time: the time spent on processing a distractor, and revisiting: multiple visits to the same stimulus in a search display, are not constant across varying levels of search efficiency and should be incorporated in visual search studies.

Their experiments showed that the dwelling was increased by target-distractor similarity, and in turn increased response time. However, they state that the estimations might not be accurate. First because in each fixation multiple stimuli might be attended and the dwelling time would be the sum of attentional dwell times on each of those stimuli. Second, the dwelling time might include the time to shift attention to another fixation target (covert shift of attention) [78].

Based on their results, revisiting was least affected by target-distractor similarity. Also, higher skipping rates caused more efficient search since skipping allows for a priori rejections of distractors as possible target candidates.

2.5 Foraging

Many of human search activities such as shopping, picking ripe apples, bag packing, looking for cooking ingredients in a fridge, etc. are not limited to search for a single item. Searching for multiple instances of a single target type among distractor items is called 'foraging'. The foraging task, as inferred from its name, has its roots in research on how animals feed. [60]

Kristjánsson et al. [61] performed some experiments on visual foraging patterns and realized that observers generally search for new items while collecting multiple targets. However, there is a limit to the number of items that observers plan for future collection. "Given enough time to view the display (1,000 ms or more), observers can typically plan collection of one to two items ahead. [61]"

A more complex version of foraging task involves searching for multiple instances of several target types, referred to as 'hybrid foraging' [108]. In hybrid foraging experiments, Wolfe et al. realized that observers tend to collect items of one target type instead of randomly searching among available targets. Reaction time (RT) data indicate searching again for the same item is more efficient than searching for any other target types, held in memory. Furthermore, observers' reaction time data showed that observers had already begun searching (their memory and visual scene) for additional targets while were collecting a target item. Thus, the hybrid foraging task is a useful way to study the interaction of visual and memory search. [108]

A recent publication [86] studies the behavior of human participants while performing foraging tasks in simulated 2D and 3D game environment. In particular, they test if Dawkins findings in [26] about chicks foraging behavior also applies to humans. Dawkins suggested in her experiments that chicks selected the same type of food for long 'runs', rather than randomly selecting from the two categories. "A 'run' refers to a sequence of selections from the same target category. When there are many short runs, this suggests that an animal is selecting at random. Fewer, longer runs, suggest that feeding is being guided in some way [86]."

The authors first tested humans foraging behavior in a 2D game environment consisted of disjunctive and conjunctive search. In the disjunctive(feature) search setting, observers had to

find red and green dots among blue and yellow dots, and the two target categories were different from the two distractor categories only in one single feature (color). The authors claim that in disjunctive search participants switched randomly between the two target types. Thus, their search were consisted of many short runs, suggesting random selection from the two target categories. On the other hand, during conjunction-based foraging where the attentional load was increased as targets differed from distractors in terms of two features, the participants had the opposite search behavior. In the conjunctive search, the targets could be red squares and green dots, while the distractors were green squares and red dots. In this case, the majority of participants foraged using far fewer runs, repeatedly selecting for the same target category, rather than switching between categories. This indicates that they had trouble using more than one conjunction template to guide their selection. Thus, authors justified the existence of a run-like behavior in humans similar to animals. [86]

The authors then created a complex interactive 3D game environment where food items were sparsely distributed, and from a given view point only a subset of items were visible. Hence, to find all the items the participants needed to navigate through the environment. Authors aimed to investigate whether the results of 2D foraging repeats in a 3D scenario. Their results indicate that in their 3D environment, unlike 2D foraging tasks, a feature/conjunction manipulation does not reduce the foraging runs.

2.6 Computational Modeling of Attention

Selective visual attention is a powerful mechanism in introducing computational efficiency. Modeling this mechanism not only deepens our understanding of human's behavior but also gives us the ability to apply this mechanism to human-made vision-based systems such as self-driving cars and various robots. Using selective attention models we can predict which parts of a scene should be processed in higher resolution and which parts could be ignored.

Predicting the interesting parts of a visual scene that should be attended is referred to as 'saliency' prediction. There are three different ways that saliency is studied in the literature: first,

saliency based on image inherent features (purely bottom-up); second, saliency according to humans' eye fixations (bottom-up + top-down); and third, saliency based on observers' choice when they are asked to mark eye-catching objects or locations in images (bottom-up + top-down). Depending on the scene, there could be inconsistencies or agreements between saliency prediction under each definition. The first and second definitions are more widely used in saliency prediction. The first saliency models were mostly based on bottom-up image-based features following the first definition. However, recent saliency models are often based on eye tracking data following the second definition. For behavioral studies of visual attention, the second definition based on eye tracking data, which we also use in our modeling, is preferable.

Attention modeling studies can be divided into several criteria according to a comprehensive review by Borji and Itti [9].

One criterion indicates whether a model is space-based, feature-based, or object-based; meaning that it needs to work with raw spatial locations, odd features, or objects respectively, in a scene. [9] Most studies are focused on space-based attention; however, there exists evidence for feature-based and object-based attention. For instance, Rubin's vase illusion as seen in figure 2.7, is considered as an evidence for object-based attention. The visual effect presents the viewer with two shape interpretations, each of which is consistent with the image, but only one of which can be maintained at a given moment [91]. Thus, the observer's mental image switches back and forth between the two interpretations of the image.

Figure 2.7: Rubin's vase illusion. Image credit: [94]

However, object-driven attention cannot explain fixations that do not fall on objects. Moreover, the boundaries of objects might not be available during the pre-attentive stage of visual perception, which is believed to direct the eyes to different locations for more detailed processing.

Feature-based models can help observers in finding a particular target among various distractors. As each target object can be defined by a set of features, feature-based modeling can be merged into the category of object-based modeling as suggested by Borji and Itti [9].

In general, it is mostly believed that attention could be driven by a combination of space-based and object-based cues.

Another important criterion is whether modeling of attention is purely spatial or spatio-temporal. In purely spatial modeling, the effect of knowledge obtained over time is not taken into account. Spatial modeling thus involves static stimuli such as still images or search arrays. On the other hand, spatio-temporal modeling involves considering the motion of dynamic stimuli (e.g. videos) over time, or learning the sequences of attended objects or performed actions as the task progresses. For instance, modeling the scanpaths, sequential eye-movements of observers, while searching for a target in an image; or modeling observers' fixations while performing an interactive task such as cleaning, cooking, playing games, etc. all involve consideration of temporal information into account.

The type of task is another crucial factor in visual attention modeling. The studied visual tasks can be divided into three major categories: 1) free-viewing 2) visual search 3) interactive. [9]

Free-viewing is a low load task and requires observers to freely watch the stimuli for a specific duration of time (normally between 1-5 seconds, but can go up to 150-200 secs for some datasets) without engaging in any particular task. Modeling free viewing saliency gives us important information regarding human's visual behavior, preferences and biases. For instance, in paper [56] authors concluded that "people tend to look at text, other people, and specifically faces. If not people, they look at other living animals and specifically their faces. In the absence of specific objects or text, humans tend towards the center of the image or locations where low-level features are salient [56]"

Free-viewing saliency is important especially for behavioral studies but it considers a very limited scenario of our daily life. Most visual activities that humans perform on a daily basis are goal-oriented and demand a great amount of top-down attention control.

Visual Search is a simple task that requires observers to look for a target object in a scene. Compared to free-viewing saliency, visual search demands more top-down control of attention. Looking for a product on a supermarket shelf is a practical example of visual search tasks in daily life.

Interactive tasks constitute most of the daily activities of our life and requires great amount of goal-oriented attention. During interactive tasks, observers are involved in complex tasks through active engagement with the environment. Daily activities such as brushing your teeth, making coffee, preparing lunch, driving to work, working with a computer or mobile phone are considered interactive and are substantially controlled by top-down task-driven attention. Interactive tasks are typically composed of many sub-tasks that could be visual such as searching and object tracking, or associated with motor-control such as manipulating an object, or lifting.

In this section, we present a comprehensive overview of computational modeling of saliency. First, we review important free-viewing and visual search saliency models, and then we discuss some saliency models for interactive tasks.

2.6.1 Free-viewing Saliency Models

Koch and Ullman Model

The first attempt to computationally model selective visual attention was proposed by Koch and Ullman in 1984 [59] inspired by feature integration theory [102]. They suggest that visual attention first creates an early representation of visual environment by encoding environment to a set of topographical maps for various features such as orientation of edges, color, disparity or direction of movement. Then the information of each individual map is projected onto one global measure of conspicuity in another topographical map, called the saliency map. Finally, a cellular network called the Winner-Take-All network (WTA network; [32]) localizes the most active unit in the

saliency map while another network relays the properties of the selected location to the central representation. The WTA network then shifts automatically to the next most conspicuous location. The visual system sequentially processes a scene by selectively inspecting the information present in conspicuous locations. [59] To avoid revisiting the same location a long-lasting inhibition of the selected unit in the saliency map is implemented, this process is referred to as 'inhibition of return' abbreviated as 'IOR' in [52].

Itti and Koch Model

Later in 1998, Koch and his PhD student Itti, presented the full implementation of Koch and Ulman model, in their paper [52] (figure 2.8). As an early representation, their model computes feature maps for intensity, color, and orientation, in a center-surround manner at different scales. They perform center-surround computations to mimic biological visual receptive fields. The created feature maps are summed across scale, linearly combined and normalized to form a single "conspicuity map" for each dimension: intensity, color, and orientation. Then the three conspicuity maps are linearly combined to generate the saliency map. The peaks of the saliency map are detected by a WTA, and attended with an IOR mechanism.

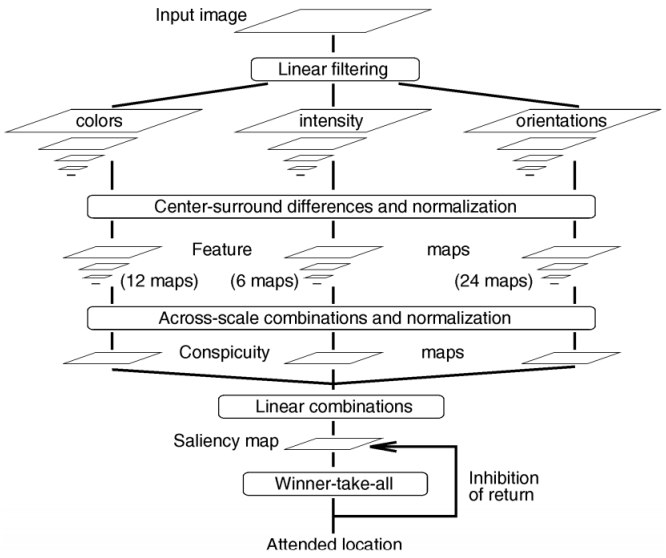


Figure 2.8: Architecture of Itti and Koch saliency model. Image credit [52].

Attention Based on Information Maximization Model (AIM)

Bruce and Tsotsos [15] introduced a Bayesian model of attention called AIM (Attention based on Information Maximization) based on the principle of maximizing information sampled from a scene. It has been believed that rare features are more informative, and can capture observers' attention. Applying this concept, AIM uses Shannon's self-information of sparse features, which is inversely proportional to the likelihood of observing those features, as a saliency measure. A higher self-information indicates a more salient location. Equation 2.1 shows Shannon's self-information.

$$I(X) = -\log(p(X)) \tag{2.1}$$

Bayesian Surprise

Itti and Baldi [50] proposed that surprising stimuli attract observers' attention. Surprise estimates the change in observers' prior and posterior beliefs about the world, after a particular observation. "Only data observations which substantially affect the observer's beliefs yield surprise, irrespective of how rare or informative in Shannon's sense these observations are [50]." Thus, surprise takes place when observed features at one location affect observer's prior beliefs obtained from other locations. Surprise is calculated using the KL divergence between the prior and posterior distributions, as shown in equation 2.2.

$$\begin{aligned} P(M|D) &= \frac{P(D|M)P(M)}{P(D)} \\ S(D,\mu) = KL(P(M|D),P(M)) &= \int_{\mu} P(M|D)\log\frac{P(M|D)}{P(M)}dM \end{aligned} \tag{2.2}$$

Where $P(M)_{M\in\mu}$ is the prior distribution of observer over the hypotheses model M and model space μ. $P(D|M)$ is the likelihood which quantifies how likely any data observation D is under the assumption that a particular model M is correct. $P(M|D)$ is the posterior distribution after data observation D.

Fixation-based Saliency Prediction

The models discussed so far can only predict visual attention caused by low-level image-driven features (purely bottom-up). Although in free-viewing condition observers are not involved with any particular task to employ high degrees of top-down attention, some higher level semantic features (that have survival benefits) guide their attention, while viewing a scene. Eye movements of observers are a reliable source to study observers' visual attention that could be caused by low-level image-based or high-level semantic-based features. With the advent of machine learning methods, data-driven approaches based on eye-tracking datasets became popular. These models attempt to predict human visual attention using overt eye-movements instead of low-level image-based statistics. Data-driven machine learning models learn some top-down image semantics that guides observers attention.

Before explaining the first saliency model that uses machine learning, we briefly describe support vector machines.

Support Vector Machines

A support vector machine (SVM) is a supervised (working with labeled training data) machine learning model that uses classification algorithms for binary classification problems. SVM model learn a hyperplane (in two dimensions would be a simple line) that best separates the training data between the two categories. This hyperplane is referred to as the 'decision boundary'. Samples falling on one side of this boundary are categorized as one group and the ones falling on the other side belong to the other group. An SVM chooses the hyperplane that maximizes the margins from both categories, i.e. whose distance to the nearest element of each group is the largest.

Judd Model

Judd et al. [56] proposed a new dataset composed of 1003 images along with recorded eye movements of 15 users free-viewing those images. They precomputed low-, mid-, and high-level features for every pixel of each image. Low-level features are composed of intensity, orientation and color contrast as introduced by Itti and Koch model as well as the values of the red, green and blue

channels; the probabilities of each of these channels; the probability of each color as computed from 3D color histograms of the image filtered with a median filter at 6 different scales; and finally the local energy of the steerable pyramid filters for sub-bands in four orientations and three scales. For extracting mid-level features a horizon line detector is trained from mid-level gist features that were introduced in [82]. To extract high-level features, authors use Viola Jones face detector [104] and the Felzenszwalb person detector [33]. Finally a feature indicating the distance to the center for each pixel is included to consider center prior, that is the tendency of observers to look at the center of images. For each image in the training and testing set they chose 10 positively labeled pixels randomly from the top 20% salient locations of the human ground truth saliency map and 10 negatively labeled pixels from the bottom 70% salient locations. They trained a linear support vector machine to model their data.

Judd et al. found that observers mostly look at faces of people or other animals, and texts in images. If there is no prominent object in the image, observers tend to look at the center of the image or locations where low-level features are salient.

Deep Learning

Feature engineering plays a crucial role in a modeling process. At first, all the required features for our problem should be extracted from the data. Next, the most important features that improve the performance of our machine learning model should be selected. Deep Learning enables us to automate the process of feature engineering, and train a network end-to-end such that all the relevant features for our problem are extracted.

The core concept of deep learning lies in structures called 'Artificial Neural Networks' or ANNs. An ANN is composed of a collection of simple inter-connected elements called artificial neurons, which loosely model the biological neural network. The neurons are typically organized into multiple layers. In feedforward networks (directed acyclic graph), neurons of one layer connect only to neurons of the immediately preceding and immediately following layers. The input layer in an ANN receives external data. The output layer of ANN produces the final result of network computations. Between the input and output layers there are zero or more hidden layers.

Layers in ANN can be connected by different connection patterns. In fully connected connection, every neuron in one layer connects to every neuron in the next layer. In pooling connections, a group of neurons in one layer connect to a single neuron in the next layer, which reduces the dimension of the layer's output and thus decreasing the number of the neurons in the next layer. Unlike feedforward connectivity, recurrent networks allow connections between neurons in the same or previous layers. Recurrent Neural Networks or RNNs can capture temporal dependencies in the input data which is required for dealing with sequential data. Convolutional neural networks, that we will discuss further in the following, can be created both feedforward and recurrent, depending on how different layers are connected to each other.

Convolutional Neural Networks (CNN)

Fully-connected neural networks have learnable weights and biases, and its neurons receive some input, perform a dot product and follow it up with a non-linear activation function such as ReLU (Rectified Linear Unit). However, the main drawback of fully connected layers while working with images is their huge number of tuning parameters (weights). For instance, for a 224 by 224 RGB image we have 224×224×3= 150528 weights for a single fully-connected neuron in a first hidden layer of a fully connected network, as we should convert the 2 dimension image to 1 dimension vector before feeding it to a fully-connected network. This conversion from 2d to 1d, also loses the spatial features of an image and the relationship between them. Spatial features refer to the arrangement of the pixels in an image. Convolutional neural networks can receive a 2d (3d for RGB images) image as their input. Convolutions without zero-padding and pooling (sub sampling) layers reduce the dimensions of input images. They can also capture the spatial features from an image, and help us to identify an object and its location in the scene.

The building blocks of CNNs are filters also known as kernels. Kernels are used to extract the relevant features from the input using the convolution operation. In classification we then use these features along with image labels to train a network in a supervised manner. In supervised learning, the networks attempts to minimize a cost function that is related to eliminating incorrect predictions. A commonly used cost is the mean-squared error, which computes the average squared

error between the network's output and the desired output. Gradient descent is an optimization algorithm used to minimize the cost function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient. In neural networks, gradient descent is used to update the weights in neural networks in a way to minimize the cost function.

With the availability of large scale labeled datasets and parallel graphics processing units (GPUs) which tremendously accelerate learning and inference, deep-learning-based saliency models, especially convolutional neural networks architectures dominated other classical approaches. One of the large free-viewing datasets that is frequently used in saliency modeling is 'SALICON' [54]. SALICON contains free-viewing data on 10,000 images from the Microsoft COCO (MS COCO) dataset with rich contextual information. They used a mouse tracking technique instead of eye tracking to record viewing behaviors of observers. The experiment is deployed on the Amazon Mechanical Turk to enable large-scale data collection. The aggregation of the mouse trajectories from different viewers indicates the probability distribution of visual attention.

Ensemble of Deep Networks (EDN)

Ensemble of Deep Networks(EDN) [103] uses convolutional layers to extract more complex biologically-inspired features from the input images. To constrain the computational complexity, they consider one- to three-layer (L1–L3) convolutional feature extractors. They augment their feature set by combining multiple representations to take advantage of the structural diversity of the individual models in the blend. To find the best parameters of feature extraction models and hyperparameters of the classifier (such as the strength of regularization) they use a method [7] that involves defining a search space and a loss function to be minimized. They create a training set of salient (the positive class) and non-salient (negative class) pixels following the approach of Judd et al. [56]: for each image in the MIT1003 training (600 images) and testing set(403 images) they sampled 10 positively labeled pixels randomly from the top 20% salient locations of the human ground truth saliency map and 10 negatively labeled pixels from the bottom 70% salient locations. At these sampled locations, features are extracted from the image and normalized over the entire training set to zero mean and unit variance. The labeled feature vectors are fed into an L2-regularized,

linear, L2-loss SVM, which is trained to predict the probability of fixation for each location in an image. L2-loss, or L2-norm loss function computes the least squares error between the ground truth and predictions. Regularization is a an important technique in machine learning to prevent overfitting. This technique adds a regularization term in order to prevent the coefficients to fit so perfectly to the training data that lose their generalizability to the unseen test data. The difference between the L1 and L2 regularization is that L2 is the sum of the square of the weights, while L1 is the sum of the weights.

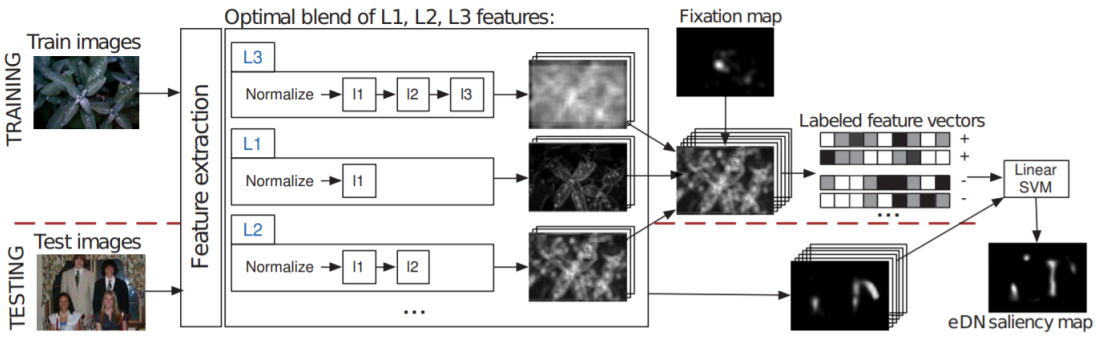


Figure 2.9: Schematic diagram of EDN pipeline. Multilayer feature extractors are found by guided hyperparameter search (not shown) and combined into an optimal blend. Resulting feature vectors are labeled with empirical gaze data and fed into a linear SVM. Image and caption credit: [103].

Transfer Learning and Deep Gaze I

Another approach that was applied to compensate for small-sized saliency datasets is to leverage the large amount of data samples from other domains like object recognition. Kummerer et al [68] showed that human fixations are largely clustered around objects. This tendency has led some models to incorporate more high level features in their modeling. For instance, Judd et al. [56] included object-level features by including detectors for faces, people, and cars along with low-level features. Kummerer et al. [68] used features directly from AlexNet convolutional neural network [62] trained for object recognition on ImageNet dataset [92], and showed promising results for free-viewing saliency prediction, with a model called Deep Gaze I. Depending on the evaluation

metric, Deep Gaze I either marginally outperformed previous methods, or obtained same accuracy as EDN.

Deep Gaze II

Later in 2016, Kummerer et al. [69] introduced another model called deep gaze II. Their model extracts the features from an input image using a VGG19 convolutional neural network pretrained on Imagenet dataset. A given input image is subsampled by a factor 2 and passed through the normalized VGG19 network. Then, the feature maps of five high-level convolutional layers: conv5_1, relu5_1, relu5_2, conv5_3, and relu5_4 are up-sampled by a factor of 8 (such that spatial resolution is sufficient for precise prediction), and concatenated into one 3-dimensional tensor with 2560 (5×512) channels. This tensor is used as input features for a second network called readout network. Readout network is made of four layers of 1×1 convolutions with 32, 16, 2, and 1 filters, followed by ReLu nonlinearities. The readout network can only compute a point-wise non-linearity in VGG features across channels. It cannot learn spatial relationships in features as all the convolution layers are 1×1. A figure of the network can be seen in 2.10.

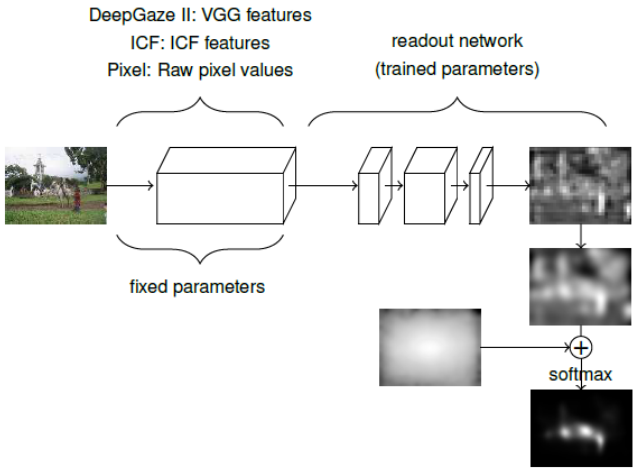


Figure 2.10: The architecture of the proposed models. Image credit: [67].

For regularizing the predictions, the output of the readout network is convolved with a Gaussian. The authors remove the effect of center bias in their modeling by adding the (negative) log

density of center prior (of their dataset) to their predictions. Equation 2.3 shows how center prior density, indicated by $p_{baseline}$, is removed from the predictions. Finally, authors use a softmax to convert their predictions to a probability distribution.

$$S'(x,y) = S(x,y) + log(p_{baseline}(x,y)) \tag{2.3}$$

Their training process is composed of two stages: pre-training, and fine-tuning. In pretraining phase, readout network is initialized randomly and trained on SALICON dataset and validated on MIT1003 dataset to check when to stop the training. In the second stage, the model is fine-tuned on MIT1003 and cross-validated over images. Note that the weights of the VGG feature extracting layers are kept fixed through out the two stages of the training process. DeepGaze II is trained probabilistically to minimize the negative log likelihood in the locations of groundtruth fixations in images. The mathematical expression for their loss function is as follows:

$$Loss = -\frac{1}{N}\sum_{i}^{n} \log p(x_i, y_i | I_i) \tag{2.4}$$

Where i is the index of the fixation in the data, indicating that image I_i is fixated at location (x_i, y_i).

The authors evaluated performance of their model using information gain, as shown in equation 2.5. Information gain explains what "the model knows about the data beyond a given baseline model [65], for which an image-independent center bias is used. [67]" In equation 2.5, $\hat{p}(x_i, y_i | I_i)$ is the density of the model at location (x, y).

$$IG(\hat{p}||p_{baseline}) = \frac{1}{N}\sum_{i}^{N} \log \hat{p}(x_i, y_i | I_i) - \log p_{baseline}(x_i, y_i) \tag{2.5}$$

The authors further introduce another metric termed information gain explained (IGE) that evaluates the absolute performance of the model. IGE compares model's performance to a gold standard model that uses a Gaussian kernel density estimate to predict one observer's fixations for a given image from the fixations of all other observers.

Multi-scale Information Network (MSI-Net)

MSI-Net (multi-scale information network) [63] is another successful saliency model. The architecture of the model has an encoder–decoder structure and includes an 'Atrous Spatial Pyramid Pooling' (ASPP) module with multiple convolutional layers at different dilation rates to capture multi-scale features in parallel. In an encoder-decoder architecture, the left half of the network maps raw image pixels to a rich representation of a collection of feature vectors. The right half of the network takes these features, produces an output and maps the output back into the image pixels format, as seen in figure 2.11. The authors use a pretrained VGG16 architecture [93] on ImageNet dataset for object recognition task as an image encoder to extract increasingly complex features along its hierarchy. Same as Deep Gaze II, they use the extracted feature maps from multiple layers of VGG16.

Similar to DeepGaze II, they first trained their model on SALICON before fine-tuning the learned weights towards fixation predictions on other saliency datasets such as MIT1003.

The model output is normalized such that they can be treated as a probability distribution, i.e. all values are non-negative with unit sum. The authors use Kullback–Leibler (KL) divergence, which computes the difference between the groundtruth and predicted fixation density maps, as their loss function to minimize during training via Adam optimization algorithm. This loss function is defined in equation 2.6.

$$D_{KL}(P||Q) = \sum_i Q_i \ln\left(\varepsilon + \frac{Q_i}{\varepsilon + P_i}\right) \tag{2.6}$$

Here, Q represents the target distribution, P the predicted/estimated distribution, i each pixel index, and ε a regularization constant. The network is trained for 10 epochs and the best-performing checkpoint is used for inference.

Our first method in Methodology section uses a very similar architecture to MSI-Net. Thus, we will elaborate more on the details of this model architecture in that section.

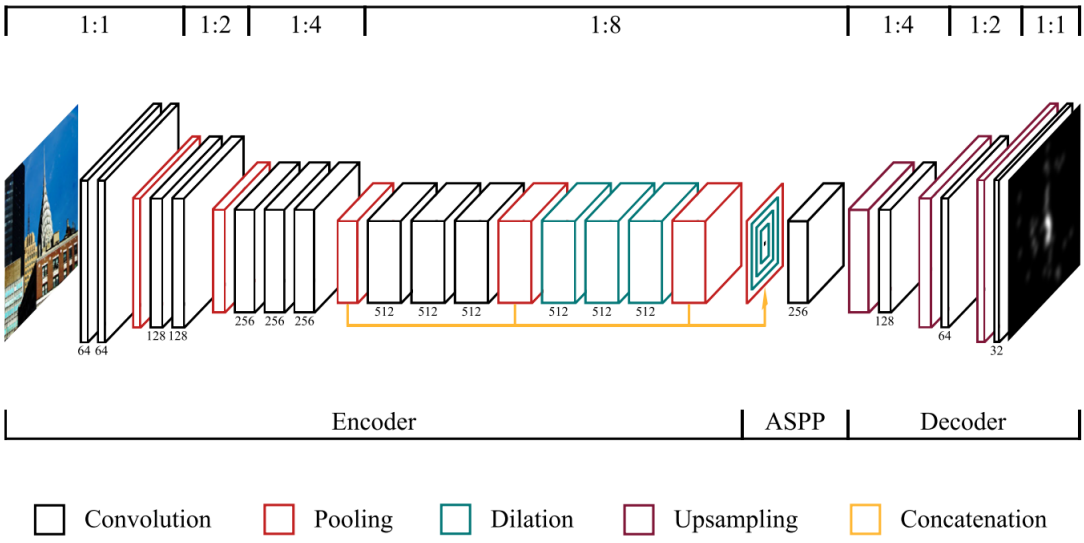


Figure 2.11: The encoder-decoder architecture of MSI-Net. Image credit: [63]

MIT/Tuebingen Saliency Benchmark

MIT/Tuebingen Saliency Benchmark [64], the successor of the MIT Saliency Benchmark, contains a leader board ranking all free-viewing saliency models based on 7 saliency evaluation metrics: Information Gain (IG), two variants of area under ROC curve (AUC-Judd and shuffled AUC), Normalized Scanpath Saliency (NSS), Pearson's Correlation Coefficient (CC), Kullback-Leibler (KL) Divergence, and Similarity (SIM). The benchmark evaluates saliency models on the MIT300 and CAT2000 datasets. In this benchmark, MSI-Net and DeepGaze II have very close performance with ranks 5 and 6 respectively.

Saliency Evaluation Metrics

In the following, we briefly discuss each of the evaluation metrics that are used in saliency literature.

Information Gain

Information Gain [65] is the difference in average log-likelihood between the model's predictions and an image-independent center-bias prior distribution, measured in bits per fixation.

AUC-Judd

AUC-Judd (AUC), is a version of the Area Under ROC curve measure. The saliency map is treated as a binary classifier to separate positive from negative samples at various thresholds. The true positive (tp) rate is the proportion of saliency map values above threshold at fixation locations. The false positive (fp) rate is the proportion of saliency map values above threshold at all pixels.

shuffled AUC

The shuffled AUC (sAUC) [96] is another version of the Area Under ROC curve measure. The saliency map is treated as a binary classifier to separate positive from negative samples at various thresholds. The true positive (tp) rate is the proportion of saliency map values above threshold at fixation locations. The false positive (fp) rate is the proportion of saliency map values above threshold at pixel locations that are fixated in other images.

AUC-Borji

Another variant of AUC by Borji et al. [12], called AUC-Borji, uses a uniform random sample of image pixels as negatives and defines the saliency map values above threshold at these pixels as false positives. The false positive calculation in AUC-Borji is a discrete approximation of the calculation in AUC-Judd. Because of a few approximations in the AUC-Borji implementation that can lead to sub-optimal behavior it is not included in MIT/Tuebingen Saliency Benchmark. [16]

Normalized Scanpath Saliency

The normalized scanpath saliency (NSS) [84] measures the mean saliency value at fixated locations of the normalized (zero mean, unit variance) saliency map.

Correlation Coefficient

The Pearson's correlation coefficient (CC) is the linear correlation coefficient between a model saliency map and an empirical saliency map gained from convolving the fixation locations with a Gaussian kernel.

Kullback-Leibler Divergence

The Kullback-Leibler divergence normalizes the model's saliency map and an empirical saliency map to be densities by dividing by the sum and then computes the Kullback-Leibler divergence between these two distributions. It is a non-symmetric measure of the information lost when the saliency map is used to estimate the empirical saliency map.

Similarity Measure

Similarity measure (SIM) is also called histogram intersection and measures the similarity between two different saliency maps when viewed as distributions. It is computed by first normalizing the model's saliency map and an empirical saliency map to be densities by dividing by the sum and then adding the pixel-wise minimums of both distributions.

Earth Mover's Distance

Earth Mover's distance (EMD) [75] measures the spatial distance between two probability distributions over a region. This metric computes the minimum cost of morphing one distribution into the other. Multiplying the amount of density moved by the distance moved gives the total cost. [16]

KLD, CC, SIM, and EMD are location-based metrics which require ground truth maps as binary fixation matrices. On the other hand, IG , NSS, AUC quantify saliency approximations after convolving gaze locations with a Gaussian kernel and representing the target output as a probability distribution. [63] [16]

2.6.2 Visual Search Saliency Models

Computational modeling of visual search literature is not as prolific as the free-viewing scenario and therefore has great potential for future research. In this section, we present some of the existing visual search models.

Rosenholtz's Model

Rosenholtz [89] [90] designed a model of visual search, in which the saliency of a target item is defined as the distance between distractors' features and target features. If target features (e.g. uniform color space) are defined as vector T, and the mean and covariance of the distractors' features are indicated by μ and Σ respectively; then the model computes the saliency, Δ as:

$$\Delta^2 = (T-\mu)'\Sigma^{-1}(T-\mu) \tag{2.7}$$

Where $(T-\mu)'$ is the transpose of $(T-\mu)$. This saliency model predicts that the higher the saliency, Δ, the easier the search task. This model measure saliency using an estimate of the rarity of features for each image location x.

Integrated Model of Top-down and Bottom-up

Navalpakkam and Itti [79] integrated bottom-up and top-down cues to optimize target detection in visual search tasks. They computed the optimal top-down weights to maximize the target's salience relative to the distractors. In particular, they attempted to maximize the signal-to-noise ratio (SNR) of the target versus distractors.

Contextual Guidance Model

Bayesian modeling in visual search saliency is used to include both target features and contextual information of the scene in a probabilistic way. Torralba and Oliva et al. [100] proposed a Bayesian framework for visual search tasks using equation 2.8.

$$p(O=1,X|L,G)=\frac{1}{p(L|G)}p(L|O=1,X,G)p(X|O=1,G)p(O=1|G). \tag{2.8}$$

Where L(X) contains a set of local measurements, and G is a set of global features that provides context representation (scene gist). Local features characterize a localized region of the image, and global features characterize the entire image. Global features are particularly important during search for a target in its natural scene contexts. O is a binary variable where O = 1 denotes target present and O = 0 denotes target absent in the image, and X defines the location of the target in the image when the target is present (O = 1). $p(L|O=1,X,G)$ represents the top-down knowledge of the target appearance and how it contributes to the search, and $p(O=1|G)$ provides the probability of target's presence in the scene. The authors decided to exclude these two terms by ignoring any contribution from the appearance of the target and also assuming $p(O=1|G)$ to be constant. The summarized formula can be seen in 2.9.

$$S(X)=\frac{1}{p(L|G)}p(X|O=1,G). \tag{2.9}$$

In 2.9 formula, the function S(X) is a contextually modulated saliency map that is constrained by the task (searching for the target). The term $1/p(L|G)$ provides a measure of how unlikely (rare) it is to find a set of local measurements within the context of an image. This term contains the bottom-up saliency in an image, i.e. the probability of having rare features in different location of an image. This term does not depend on the target and is purely bottom-up. The term, $p(X|O=1,G)$, provides context-based priors on the location of the target. It relies on past experience to learn the relationship between target locations and global scene features. [100]

The authors approximate conditional distribution $1/p(L|G)$ with $1/p(L)$, using the statistics of the current image. Also to avoid having saliency consistently dominated by one of the two terms, they apply an exponent γ to the local saliency term, as seen in 2.10. γ is tuned using a validation set.

$$S(X)=p(L)^{-\gamma}p(X|O=1,G). \tag{2.10}$$

Saliency Using Natural Statistics (SUN)

Zhang et al. proposed another Bayesian model of visual search called 'SUN' (saliency using natural statistics) [58]. This model contains a bottom-up saliency component (similar to [100]) and a top-down component that guides attention to the areas of the scene likely to be a target based on appearance alone. The equation used in their Bayesian framework is shown in 2.11.

$$S(X) = \frac{1}{p(L)} p(L|O=1)p(O=1|X) \tag{2.11}$$

For a better comparison, here we use the notation of contextual guidance model by Torralba et al., where $\frac{1}{p(L)}$ is bottom-up saliency (independent of target), $p(L|O=1)$ is the likelihood of having a set of features given the presence of the target i.e. feature values consistent with our knowledge of the target's appearance, and $p(O=1|X)$ is the location prior i.e. the possiblity of presence of target at a given location in the scene. These last two terms $p(L|O=1)p(O=1|X)$ together form the top-down saliency that is dependent on the target.

By taking the log of saliency formula, we can interpret the framework in an information theoretic way.

$$\log(S) = -\log(p(L)) + \log(p(L|O=1)) + \log(p(O=1|X)) \tag{2.12}$$

In information theory, $-\log(p(L))$ is referred to as the self-information of the random variable L. As the probability of features decreases, self-information increases. In other words, rare (surprising) features have a higher self-information; and thus a higher bottom-up saliency.

If the location prior is uniform, the third term would be a constant and can be dropped from the formula. The combination of the first two terms leads to the point-wise mutual information between the local features and the presence of a target:

$$\log(S) = -\log(p(L)) + \log(p(L|O=1)) = \log \frac{p(L, O=1)}{p(L)p(O=1)} \tag{2.13}$$

In summary, SUN computes the most likely locations in the image that contain the target. This goal is achieved by maximizing the point-wise mutual information between features and the target class [58].

Ehinger's People Search Model

Ehinger et al. [29] used the Bayesian framework of Torralba et al. as seen in equation 2.8 to explain the eye movements in looking for people in a database of about 912 natural scenes. They computed the saliency maps predicted by each component of the framework, namely bottom-up saliency, gist, and target features; and then combined them by multiplying the weighted saliency maps. They evaluated their models using 14 observers' recorded eye movements as they searched for pedestrians. They first computed inter-observer agreement by using the fixations generated by all-except-one observers, and then used this to predict the fixations of the excluded observer. This process was iterated for all observers for each image. Their model was able to achieve 94% of human agreement. Among all components, scene context had the most contribution to the model's performance.

PAAV Model

PAAV [81], pre-attentive and attentive vision module, is another computational model that integrates bottom-up and top-down processing in visual search through a mathematical framework. The pre-attentive stage is driven by the stimuli features such as color, orientation, size, and location in image; these features are detectable if they surpass visual acuity thresholds. The attentive stage considers the similarity of stimuli with the target features. Finally, the weighted sum of pre-attentive and attentive saliency activations directs the visual attention.

Deep-learning-based Approaches

In the rest we will present several more recent deep-learning-based approaches for modeling attention in visual search tasks.

Invariant Visual Search Network (IVSN)

Zhang et al. [113] propose a biologically inspired computational model, named 'Invariant Visual Search Network' (IVSN), that locates targets through sequential fixations without exhaustive search. Also, the model is zero-shot meaning that it can generalize to novel objects without any training. The model consists of two separate streams of feed-forward convolutional neural network (VGG16) pre-trained for object recognition task on ImageNet dataset. One stream extracts features from a sample object from the target category (if the target object is a horse, then a different horse is presented to this stream of network) and another stream extracts features from the entire search image. The features of a sample target image are convolved with the feature map of a search image, and an attention map is generated. This process is an imitation of pre-frontal cortex top-down modulation that contains the task-dependent information about a target. Fixation locations are generated in the descending order of maximum locations in the attention map using a winner-take-all mechanism (WTA). If the target is not found at the current fixation, inhibition of return is applied (IOR), the fixation location is set to 0 in the attention map and the next maximum location is selected. This process is repeated until the target is found. [113]

The authors performed search on three different types of datasets: object arrays, natural images, and waldo images. They first acquired human subjects' fixations when searching for a target in each of the images. The subjects were first presented with a sample image of the target category, referred to as the 'target cue' (target cue is not the same as the target object in the search image i.e if the target object is a cat, a different cat is shown as the target cue). Then the search image was presented to the subjects, and their fixations were recorded. These fixations were then used for a comparison with IVSN model generated fixations. The authors claim their model has similarities to human's fixation behavior despite not being explicitly trained on human fixation data. The average number of fixations by IVSN for search in object arrays was similar to human subjects, and for natural and waldo images was more than human subjects. One key difference between the model and human behavior is that the fixations generated by IVSN are not in close proximity and can move between far parts of the image (i.e. going from bottom left corner to the top right corner); while, humans tend to move their eyes more smoothly.

Visual Search on Websites and Graphical Layouts

Yuan et al. [111] proposed a model that predicts visual search time of observers while searching for 5 types of targets: image, text, link, button and input field on web pages, using a combination of deep learning and non deep learning hand-crafted features. The deep learning part of the model is composed of two branches, one branch extracts the features of the entire webpage using 3 convolutional layers where each layer has a kernel size of 3×3 and output channels of 4 and the other branch extracts the features of the target object using 6 convolutional layers with 3×3 kernels. Then the cosine similarity between the target embedding vector and webpage embedding is calculated. The other stream of their model extracts heuristic-based structured features including 1) the (x, y) coordinates of the top left corner of the target bounding box 2) the width, the height, and the area of the target 3) the number of candidates or distractors and 4) target type; there are five possible target types (image, text, link, button and input field). They used real-valued embedding vectors to encode different target types and these embeddings are learned from the data. The other features are encoded as numerical variables. All structured features, including the original value of the numerical variables and the embedding of the categorical variable are combined and fed to a 2-layer fully connected network with 100 and 50 hidden units and tanh activation function. The output of this embedding is then concatenated with the output of cosine similarity calculated between the webpage and target embeddings, and fed to a final fully-connected layer with a single output unit, with no activation function. Their model predicts the search time as a regression task, and the level of difficulty of the task as a classification task. They trained their model on a dataset collected via crowdsourcing. A large set of random Webpages were used to collect a dataset of search time labels from English-speaking crowd workers recruited via Amazon Mechanical Turk. They found that the y-coordinate of the top left corner of the target is positively correlated with the search time, implying that a dominant top-down vertical search strategy employed by the human workers. Moreover, both the width and the height of the target negatively correlated with the search time, i.e., the bigger the target is, the shorter the search time. They also claim that the more items on a page, the longer it should take a user to find a target. Furthermore, some target types

are found within a shorter time. For instance, image targets are the easiest to find and text targets are the hardest.

Numerous other studies such as [55], [97], [98], [37], and [99] present computational models of visual search on graphical layouts and web pages. [85] designed a model for grid tasks on touchscreen mobile devices by combining traditional analytical methods and data-driven machine learning approaches. [6], [20] and [22] provide models for menu search and performance.

Also, there are models that predict visual saliency of web pages under tasks other than visual search. For instance, Zheng et al. [114] propose a model to predict webpage saliency under five common tasks: signing-up (email), information browsing (product promotion), form filling (file sharing, job searching), shopping and community joining (social networking). Their model has two branches, one branch predicts bottom-up saliency, and the other branch predicts task-driven (top-down) saliency. The final prediction is produced by combining the output of the two branches.

All of these models can be used for a more robust design of webpages, digital screens, and menus to make them more user-friendly. This way users will spend the least possible search time looking for their target location, application, product, etc.

Scan-path Prediction Using CNN and RNN

Several researchers have attempted to model the scan-path (i.e. sequence of saccades and fixations) of observers while performing visual search tasks. Zelinsky et al. [112] introduced a new dataset for categorical visual search containing images of clock and microwave objects. They modeled the scan path of observers while they were looking for each of the two categories in images. They presented two models, one CNN-based and the other RNN-based. In their first method, they used cumulative foveated images, in which the information accumulates over fixations by progressively de-blurring a blurred foveated image based on high-resolution information obtained at each new fixation. This method is inspired by organism's foveated retina, i.e. having high visual acuity only at the 1 degree region surrounding the central vision, and progressively more blurred vision with distance away from the fixated location. Then they trained a convolutional neural network to input a cumulative foveated image based on a given fixation in a scanpath and output the location

of the next fixation. The CNN is based on ResNet-50, a popular CNN architecture, (pretrained on ImageNet) with the last average-pooling layer and fully-connected layer removed. In the last layers, they added two 1×1 convolutional layers and a soft-max layer that maps the feature maps from ResNet-50 to a location probability map. Finally, a WTA network is used to choose the location in output image with the highest probability as the next predicted fixation.

Their second method involves training three types of recurrent neural networks (simple RNN, LSTM, and GRU), which is suitable for sequential data modeling. They first extract features from images using ResNet-50, pre-trained on ImageNet, and then use this feature map to predict the next fixation given the previous fixation location. The first fixation is assumed to be at the center of the image. They train the RNNs using cross-entropy loss between the groundtruth and predicted fixations. The authors also incorporate inhibition of return (IOR) mechanisms using masking techniques to avoid revisiting the same location multiple times. The results showed that the scanpath similarity was highest among participants (people scanpaths agreement with each other) compared to model's predictions. Also among the two models, RNN-based model outperformed the CNN model. One drawback of their methods is that a separate model should be trained for each target category, i.e. one model for microwave search and another for clock search. The authors addressed this problem in their later work. [109]

Scan-path Prediction Using Inverse Reinforcement Learning

In 2020, Zelinsky et al. [109] proposed a model to predict human scan-paths during visual search using inverse reinforcement learning. They attempt to learn the reward function and policy used by humans during visual search using adversarial training. In this formulation, reward and policy are considered as part of the discriminator and generator, respectively. The discriminator assigns high reward to an expert's behavior and low reward to a non-expert's behavior, where behavior is represented as state-action pairs. The generator/policy is optimized using a reinforcement learning algorithm, called GAIL (generative adversarial imitation learning) to get higher rewards by behaving more like the expert. The diagram of this pipeline is shown in figure 2.12.

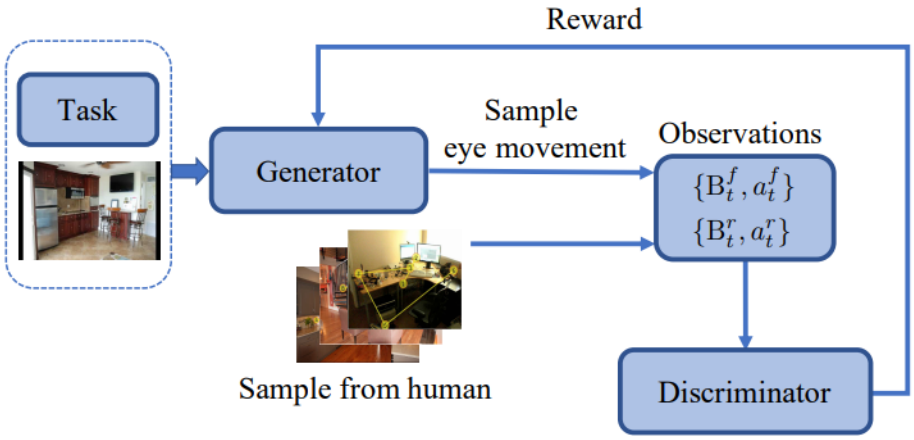


Figure 2.12: "Inverse reinforcement learning framework. Generator creates fake state-action pairs, by sampling fixations from images and tasks. The discriminator is trained to differentiates real human state-action pairs from the generated ones and provides reward to train the generator [109]." Image and caption credit: [109]

For the state representation they used a method called 'dynamic contextual belief maps of object locations'. Similar to their previous work [112], they have implemented the effect of fovea (i.e. higher resolution in the fovea and progressive blurriness depending on the distance away from fovea) in their state representation, which they termed retina-transformed image. However, they did not apply a progressive blurring on the input image. Instead, for each fixation they placed a high-resolution local patch of the image around the fixated location and the blurred representation of the rest of the image as the peripheral input. Moreover, the authors create panoptic segmentation of the scene, generating masks for 80 object and 54 background classes. These segmentations are computed to account for the hypothesis that humans parse a scene into objects and backgrounds to build a belief map of where a target should be. For instance, authors show that in a search for TV, observers mostly fixated at walls as TVs are mostly hanging on walls. Authors generate these contextual belief maps for 134 categories as the primary component of the state representation. Initially, the state is based on the contextual beliefs on the low-resolution image corresponding to a peripheral visual input. Then for each fixation by the searcher, they update the state by replacing the portion of the low-resolution belief maps with the corresponding high-resolution portion obtained

at the new fixation location. Moreover, as humans search scan-paths differ based on the search target, the authors augment the state by concatenating it with a one-hot task vector.

To train and evaluate their model, authors have also created a large-scale dataset named COCO-Search18 including search fixations of 10 people viewing 6202 images while searching for each of 18 target-object categories. We will elaborate more on this dataset in the methodology chapter, as we also use it to train and evaluate our implemented models.

The authors compare their model with 5 other models on scan-path evaluation metrics. They conclude that the IRL algorithm outperforms the other methods on all metrics. They also show that reward maps recovered by the IRL model depend greatly on the category of the search target.

Visual Question Answering

Chen et al. created a human eye-tracking dataset for Visual Question Answering (VQA) tasks [19]. During the experiments, participants were asked several questions from an image such as the color of a particular object or the spatial relationship between two objects, and their eye movements were recorded. Visual question answering tasks include visual search as part of the process, but they also demand visual reasoning which makes them more complicated compared to visual search.

Predicting Task From Eye Movements

There are several studies such as [44] [10] [8] that model the reverse operation. They attempt to infer the task of observers from their eye tracking data. Haji-Abolhassani et Clark [44] used Hidden Markov Models to infer 4 types of tasks from observers' eye tracking data. The 4 tasks were: memorizing the picture, determining the decade in which the picture was taken, determining how well the people in the picture know each other, and determining the wealth of the people in the picture.

Research on inferring task from eye tracking data validates the dependency between observers' scan-path patterns and their visual task. In [44] authors suggest that observers fixated at faces when they were asked about the decade that the picture was taken, while they fixated at inanimate objects

for estimating the wealth of the people. All of these distinct task-driven features in eye movements can be used to study how human's visual attention works.

2.6.3 Interactive Tasks Saliency Models

In most real-world scenarios, we are involved in complex tasks that demands active interaction with the environment, combining perceived visual information with decision making and motor control. Moreover, most stimuli that we interact with on a daily basis are 3-dimensional objects. It is therefore crucial to study visual attention mechanisms in 3D environments.

Task-specific Attention During Game Playing

Borji et al. [11] proposed a Bayesian framework to model task-driven visual attention by combining different sources of information. They integrated global context of a scene, previous fixated locations, and previous motor actions, to predict the next fixation. To evaluate their model, they recorded eye movements of participants while playing five 2D and 3D games, and compared their model's predictions with the recorded eye movements.

Gaze In Ego-centric Video

Another important factor in task-driven attention prediction is the visual perspective. Humans see the world through ego-centric (first-person) vision, meaning that they sense the environment from a first-person point of view. Li et al. [73] introduce a model to predict gaze in egocentric view from a head-mounted camera. This camera records a video of head motion and hand location of participants while they perform a daily activity. These information are used to estimate the gaze location in the captured video. They use a graphical model that combines single frame egocentric cues with temporal dynamics of gaze to predict gaze position. A more recent work by Huang et al. [48] predicts gaze in ego-centric videos during daily actions, using temporal and spatial CNNs, and long short term memory (LSTM).

DR(eye)VE

One of the daily tasks in which attention plays an important role is driving. Alletto et al. [3] created a dataset of driver's eye movements called 'DR(eye)VE'. This dataset is captured in different weather conditions (sunny, rainy, cloudy), three different times of the day (morning, evening, night), three different landscapes (highway, countryside, downtown), and for 8 different drivers. They also published a paper named 'Predicting the Driver's Focus of Attention:the DR(eye)VE Project' [83], in which they predict the attention map of the driver in the last video frame of a sequence composed of 16 frames. In their method, the attention maps for each frame are composed of the aggregated fixations over 25 frames (12 frames before and 12 frames after each frame). They also compute optical flow and semantic segmentation of each input video frame and feed all three domains (original video (RGB), optical flow, and segmentation) to three different branches of their proposed model. Their model has 3D convolutional layers to capture (short) temporal dependencies in the gaze data. Also each branch has two streams, coarse and refine, to prevent the network from learning a central bias.

They also attempt to investigate if a driver is distracted/inattentive using their gaze pattern. For this purpose, they annotate their dataset with 4 subcategories: 1) errors 2) subjective events 3) acting 4) inattentive. We describe each of these subcategories in the following.

1) Errors: When the fixations are too scattered, authors label them as erroneous. Errors might happen either due to failures in the measuring tool (e.g. in extreme lighting conditions) or in the data processing phase (SIFT matching).

2) Subjective Events: This category describes the situations in which the fixation pattern is related to the experience of the driver, e.g. "a road sign on the side might be an interesting element to focus for someone that has never been on that road before but might be safely ignored by someone who drives that road every day. [83]"

3) Acting: Situations where driver deviates his attention from the central pattern (road vanishing point) due to an intention linked to task-specific actions (e.g. turning, changing lanes, overtaking).

4) Inattentive: Finally the last category denotes the inattentiveness of the driver which occurs when the driver fixates on objects that are not related to the driving task (e.g. looking at an advertisement).

They also studied the effect of perceptual load that happens in high-speed vs low-speed driving. Interestingly, they realized that the fixations are less sparse and more focused on the vanishing point of the road during high-speed driving such as in highways compared to low-speed driving streets. This happens because in high speed driving the temporal density of information (i.e. the amount of visual information that the driver needs to perceive per unit of time) increases and as the perceptual capacity is limited, the useful visual field of the driver shrinks to allow for more relevant information processing (and suppression of distractors). In this case, the drivers control other important elements of the scene (such as a car coming from the opposite side of the road) through peripheral vision.

2.6.4 Summary

In this chapter, we gave an overview of popular theories in psychology about visual attention and distraction. We then elaborated on the computational models that predict salience during free-viewing, visual search, and interactive tasks. In the next chapter, we introduce two methods to address visual search salience modeling.

Our first introduced method is related to the two discussed works [63] and [113]. In particular, we extend MSI-Net [63] free-viewing model to visual search scenario by adding an additional stream for the target object, similar to Invariant Visual Search Network [113]. Inspired by several psycho-physical observations such as Rubin's vase illusion, we propose a second method that predicts object-based salience and distraction using Mask-RCNN segmentation network. In both of our methods, we use COCOSearch-18 dataset [21]. This dataset was also used in research paper [109] to predict scan-paths during visual search.

Chapter 3

Methodology and Experiments

In this chapter, we introduce two methods for modeling human's gaze behavior in visual search tasks, using the COCO-Search18 fixation dataset [21]. Our main goal is to predict attention to both the target and the distracting objects during visual search. The first method generates fixation density maps, containing the fixation probability of each pixel in the search image, using a similar encoder-decoder architecture to MSI-Net [63]. This method is able to locate the targets and some distracting locations with good accuracy. The second method generates the instance segmentation of distractors and targets in an image, using Mask R-CNN architecture [45]. This method also gives us a distraction score for each segmented distractor. In the implementation of both methods, we use Tensorflow framework [2].

At the beginning of this chapter, we first present some analysis on the COCO-Search18 fixation dataset to gain insight into how different parameters can affect search performance. We investigate how reaction time, total number of fixations, fixation duration on targets, and fixation duration on non-targets change for different target categories, different observers, various levels of target eccentricity, and different sizes of target. We also look into the correlation between the error rate of observers, i.e. when the fail to detect the correct target, and four search performance parameters, i.e. reaction time, the total number of fixations, fixation duration on targets, and fixation duration on non-targets. These analyses are performed solely for information acquisition, and do not have a significant effect in our modeling.

3.1 Dataset Analysis

COCO-Search18 is a fixation dataset containing the fixation locations of 10 observers searching for each of the 18 object categories in 6202 images. Among these 6202 images, 3101 images contain the target object (used for target-present trials) and 3101 do not contain the target (used for target-absent trials). In our modeling we only use 3101 target-present images and their corresponding fixation data. Observers' eye positions were sampled every millisecond using an eye tracker. The obtained 70,000,000 raw gaze positions were clustered into 268,760 search fixations after excluding the incorrect trials. Each trial was initiated with a fixation dot appearing at the center of the screen. Participants started a trial by pressing a button on a game-pad controller while looking at the fixation dot. An image of a scene was displayed and the participant's task was to quickly decide if the target is present in the image or not, by responding 'yes' or 'no' using the right or left triggers of a game-pad controller. The 18 categories of target objects are composed of bottle, bowl, car, chair, clock, cup, fork, keyboard, knife, laptop, microwave, mouse, oven, potted plant, sink, stop sign, toilet, and TV; all of which appear in their natural context. For instance, cars appear on the streets or other outdoor scenes, laptops appear on desks/tables at the offices, and microwaves appear in the kitchen environment. All images in the dataset were resized to 1680×1050 with zero padding and aspect ratio kept. The fixation data are presented in json files. There are two splits available for the dataset. Split1 files have the same train/valid split as was used in [109] and we also use in our modelings. Split2 files have all training images from COCO2014 training set and all validation images from COCO2014 validation set. The json files also contain the fixations' time duration and participants' reaction time. The content of json files for each conducted visual search is organized as shown in table 3.1.

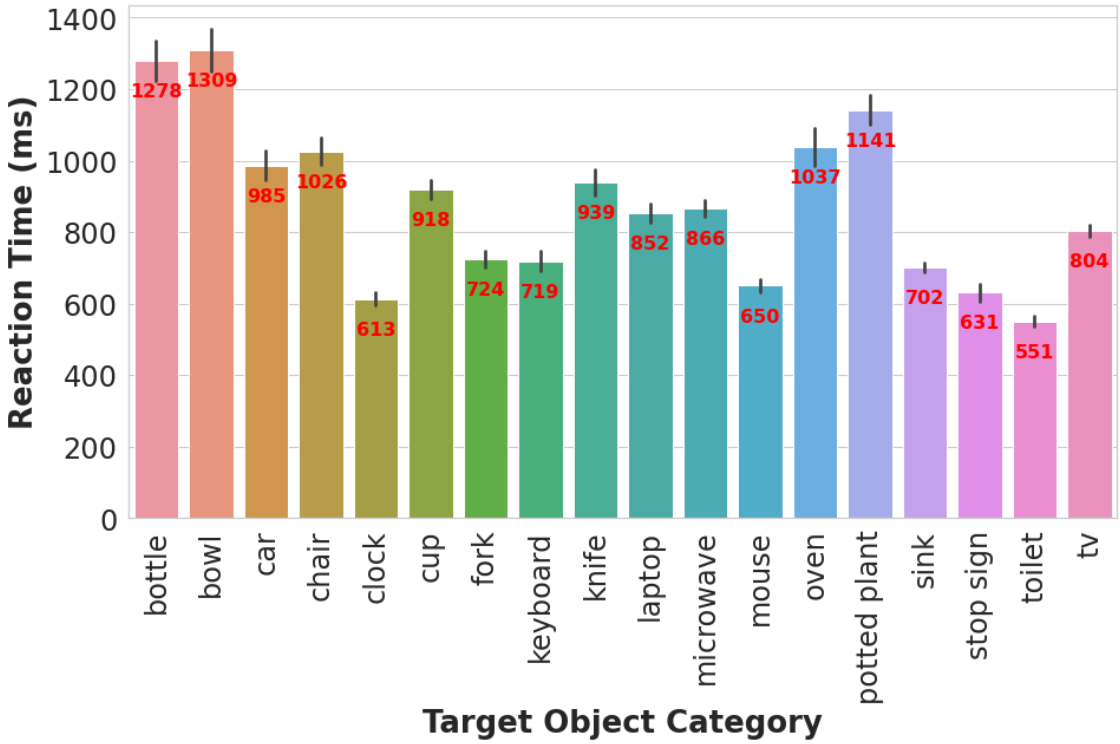
‘name’: image name,
‘subject’: subject number,
‘task’: target object category,
‘condition’: target present or target absent images,
‘bbox’: bounding box coordinates of the target [x,y,w,h],
‘X’: X coordinates of fixations from the first fixation to the last in this trial (including the initial center fixation),
‘Y’: Y coordinates of fixations from the first fixation to the last in this trial (including the initial center fixation),
‘T’: fixation durations from the first fixation to the last in this trial (including the initial center fixation),
‘length’: number of total fixations in this trial,
‘correct’: correct response = 1, incorrect response = 0,
‘RT’: search reaction time,
‘split’: training set = ’train’, validation set = ’valid’

Table 3.1: COCOSearch18 fixation data format.

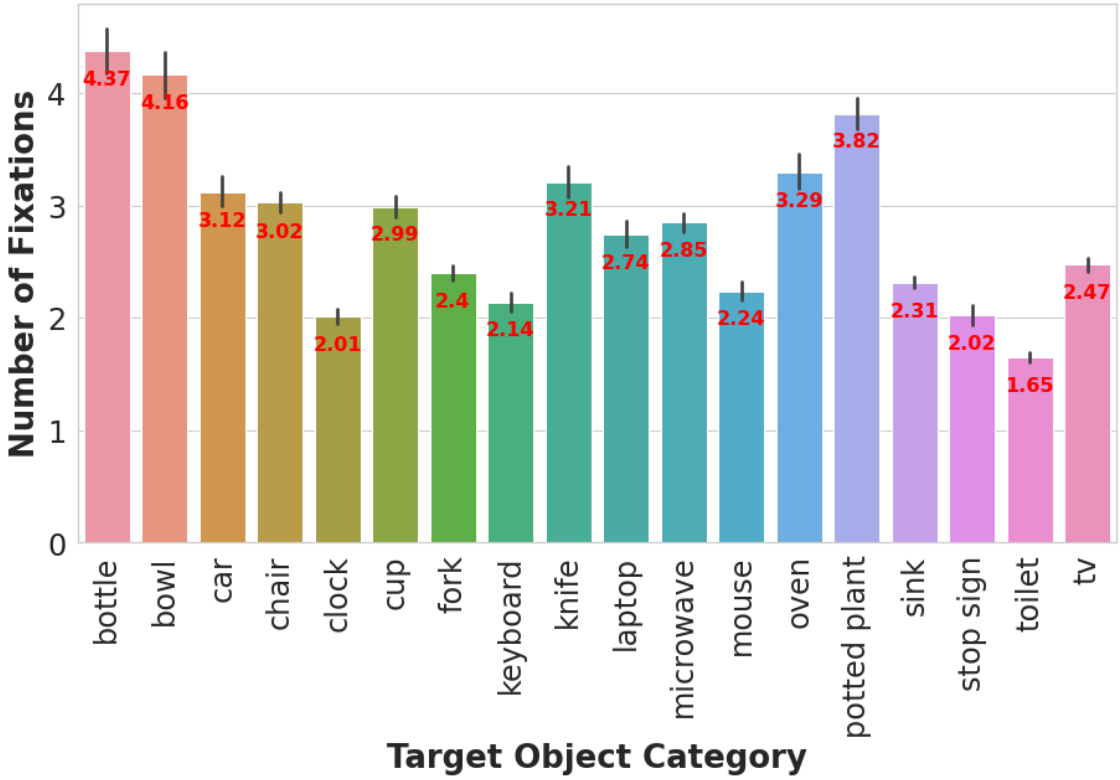
The COCO-Search18 dataset is accessible at `https://saliency.tuebingen.ai/datasets/COCO-Search18/`.

In order to understand the differences in search performance between object categories, we plot the average reaction time, average number of fixations (excluding the initial center fixation, including target fixation), average fixation duration time on non-target (distractors) objects (excluding the initial center fixation), and average fixation duration time on target objects (excluding the initial center fixation), for each object category.

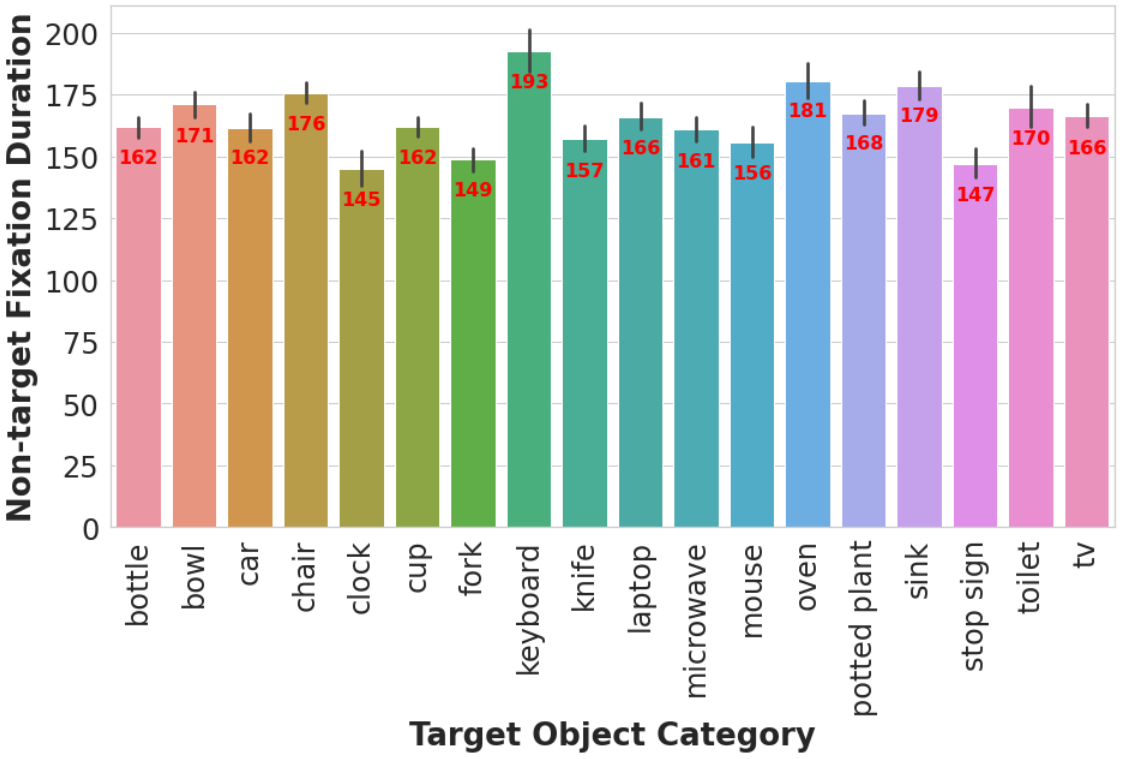
The error bars in plots provide some indication of the uncertainty around that estimate, based on a normal distribution 95% confidence interval. In other words, if our data were randomly and independently sampled from a Gaussian distribution, we can be 95% sure that the confidence


Reaction Time (ms)
1400
1200
1000
800
600
400
200
0
1278
1309
985
1026
613
918
724
719
939
852
866
650
1037
1141
702
631
551
804
bottle
bowl
car
chair
clock
cup
fork
keyboard
knife
laptop
microwave
mouse
oven
potted plant
sink
stop sign
toilet
tv
Target Object Category

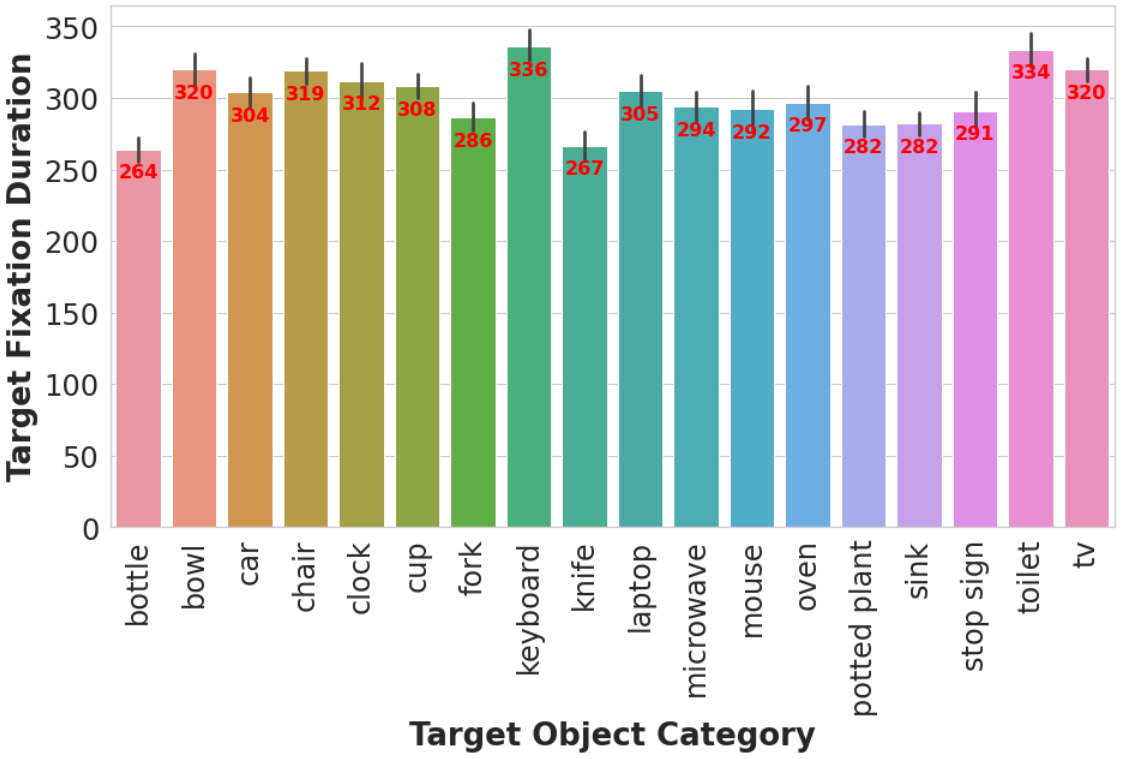
(a)


Number of Fixations
4
3
2
1
0
4.37
4.16
3.12
3.02
2.01
2.99
2.4
2.14
3.21
2.74
2.85
2.24
3.29
3.82
2.31
2.02
1.65
2.47
bottle
bowl
car
chair
clock
cup
fork
keyboard
knife
laptop
microwave
mouse
oven
potted plant
sink
stop sign
toilet
tv
Target Object Category

(b)



(c)



(d)

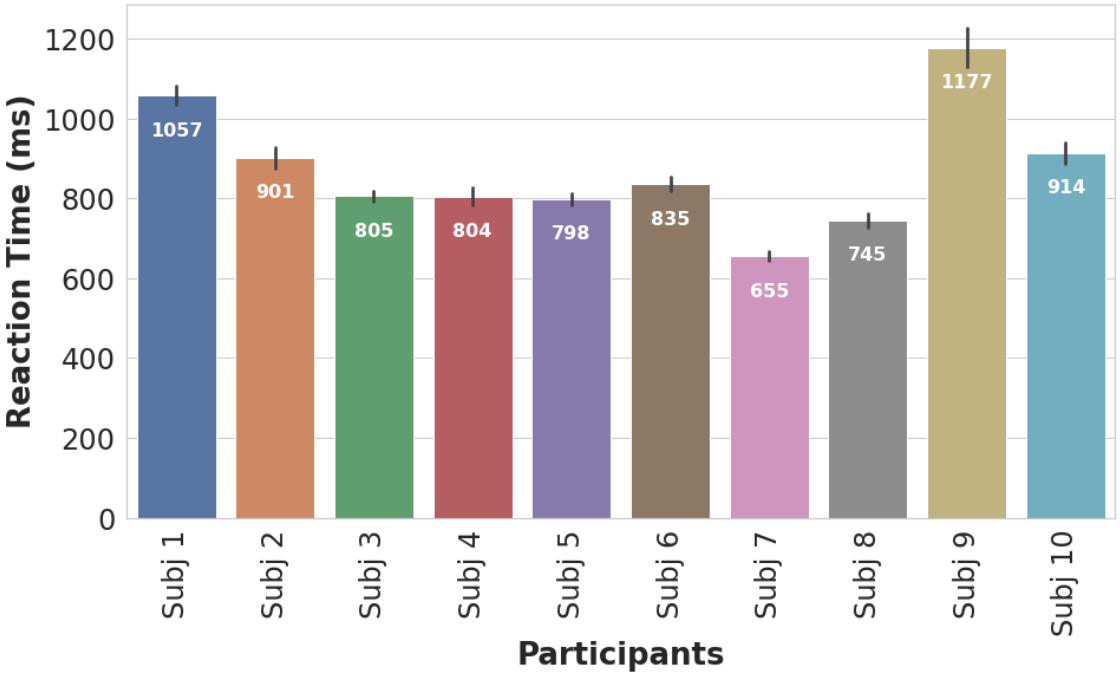
Figure 3.1: Search Performance over target object categories. (a) Average reaction time. (b) Average number of fixations. (c) Average non-target fixation duration. (d) Average target fixation duration.

interval (CI) contains the true population standard deviation (SD) [41]. The first two plots in figure 3.1 are also drawn by the authors of COCOSearch-18 dataset.

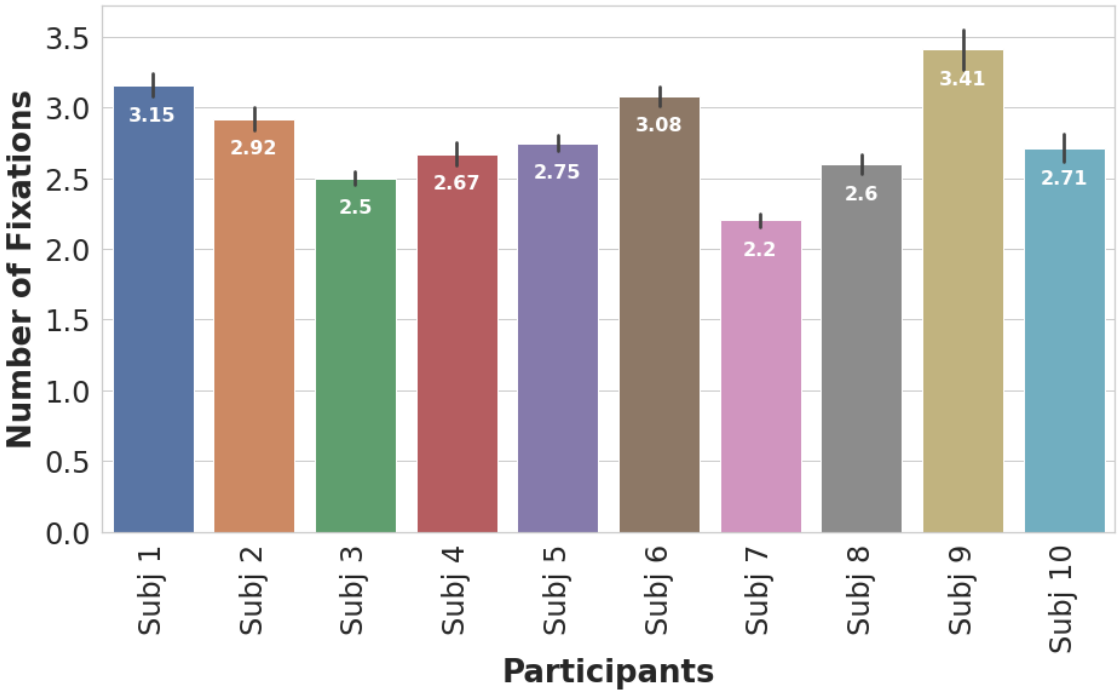
It can be seen in figure 3.1 that the hardest categories to find are bottle, bowl, and potted plant with the longest average search times and highest average number of fixations. Toilet, clock, and stop sign are the easiest targets, with the shortest average search times and lowest average number of fixations. The average fixation duration for both target and non-target fixations (figure 3.1) are relatively constant across categories with a maximum variation of 50 milliseconds for non-target fixations and 75 milliseconds for target fixations. The Keyboard category has a slightly higher fixation duration compared to other objects, even though it is among easier targets with a relatively low number of fixations and short search time. It is believed that duration time is composed of two factors, the cognitive processing time of fixated object(s) and the covert shift of attention along with the time required to initiate the next eye movement by the oculomotor system. [78] Moffitt explains in his paper [78] that the fixation duration time is adjusted to process the information of one or a cluster of items, while the saccade extent depends on the distance between the items (cluster of items). He claims that although in cluttered displays with more visual information we expect a higher fixation duration, a more flexible strategy could be applied by visual system. In high-density displays, the saccade extent could be adjusted to allow for fixation duration to remain relatively constant. Our results also show a relatively constant fixation duration for different target categories, in images with varied levels of clutter.

To investigate the individual differences in search performance, we plot the same variables, average reaction time, average number of fixations, average non-target fixation duration, and average target fixation duration, along with the number of incorrect responses for each of the 10 participants in figure 3.2. The first two plots in figure 3.2 are also drawn by the authors of COCOSearch-18 dataset.

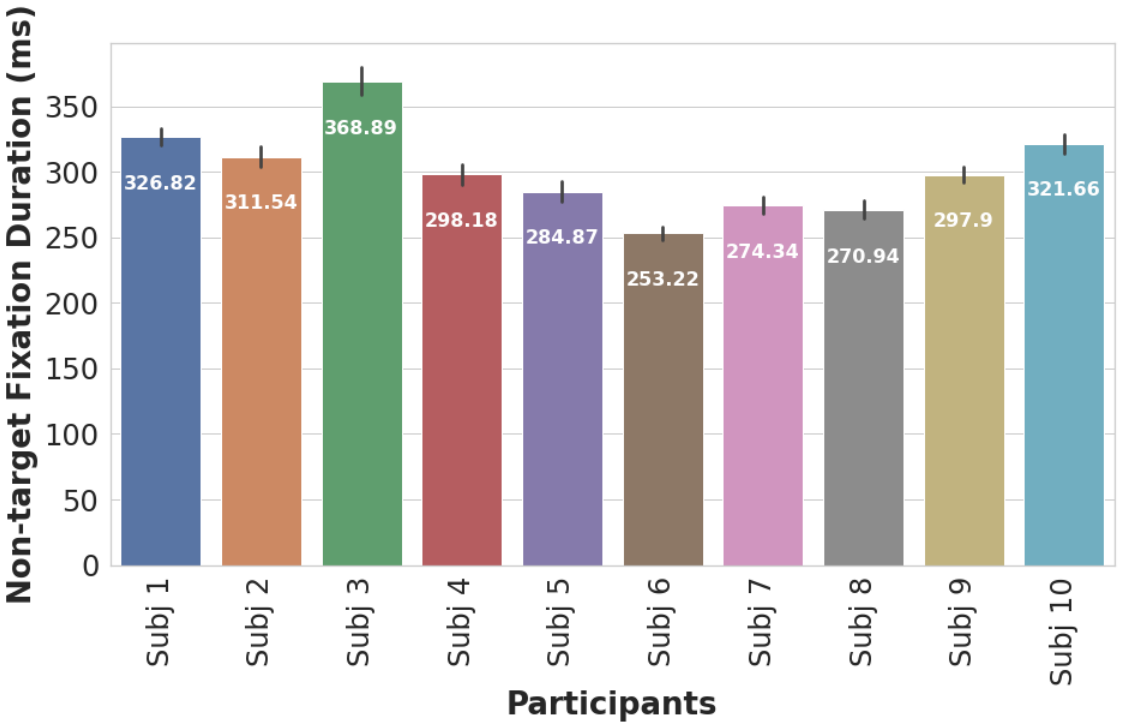
Among all participants, participant 7 has the lowest response time and number of fixations, while participant 9 has the highest response time and fixation number. Although the search time and number of fixations are important factors in determining search efficiency, the correctness of responses are also crucial. Participants 10 and 1 have the lowest number of errors among others,


Reaction Time (ms)
1200
1000
800
600
400
200
0
1057
901
805
804
798
835
655
745
1177
914
Subj 1
Subj 2
Subj 3
Subj 4
Subj 5
Subj 6
Subj 7
Subj 8
Subj 9
Subj 10
Participants

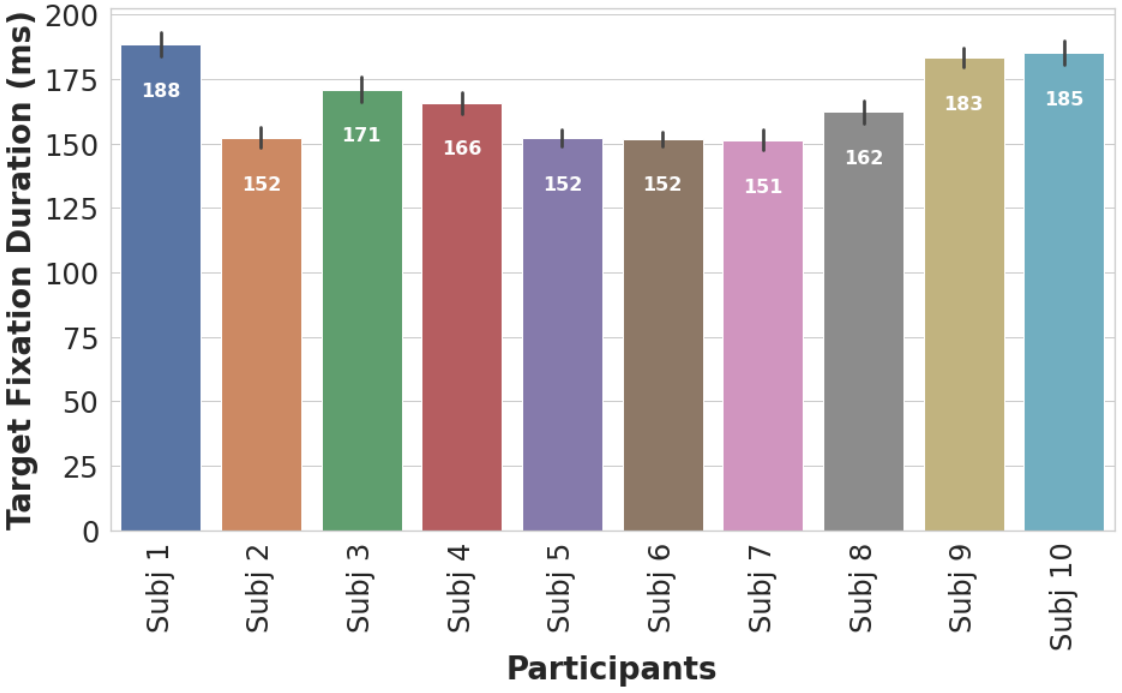
(a)


Number of Fixations
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
3.15
2.92
2.5
2.67
2.75
3.08
2.2
2.6
3.41
2.71
Subj 1
Subj 2
Subj 3
Subj 4
Subj 5
Subj 6
Subj 7
Subj 8
Subj 9
Subj 10
Participants

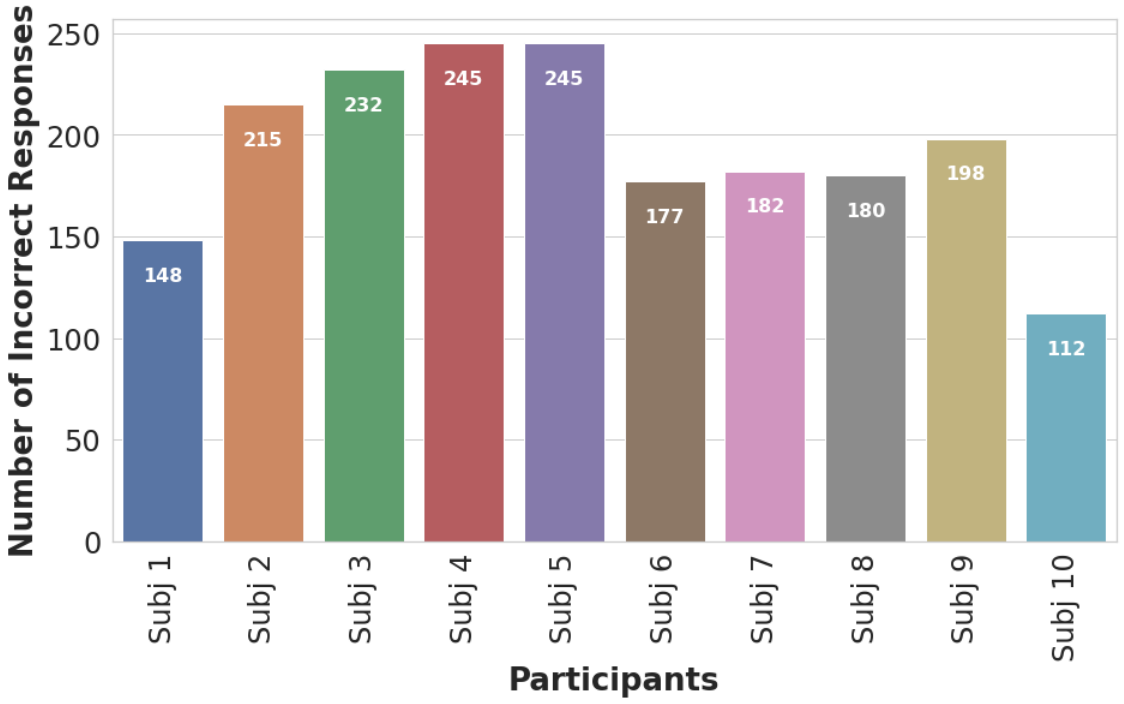
(b)


Non-target Fixation Duration (ms)
0
50
100
150
200
250
300
350
326.82
311.54
368.89
298.18
284.87
253.22
274.34
270.94
297.9
321.66
Subj 1
Subj 2
Subj 3
Subj 4
Subj 5
Subj 6
Subj 7
Subj 8
Subj 9
Subj 10
Participants

(c)


Target Fixation Duration (ms)
0
25
50
75
100
125
150
175
200
188
152
171
166
152
152
151
162
183
185
Subj 1
Subj 2
Subj 3
Subj 4
Subj 5
Subj 6
Subj 7
Subj 8
Subj 9
Subj 10
Participants

(d)



(e)

Figure 3.2: Search performance for each participant. (a) Average reaction time. (b) Average number of fixations. (c) Average non-target fixation duration. (d) Average target fixation duration. (e) Number of incorrect responses.

even-though their search time is relatively high. These subjects tend to search the display more exhaustively, minimizing their risk of error. Participants 3, 4, and 5 have the highest number of errors, while their search time is among the lowest.

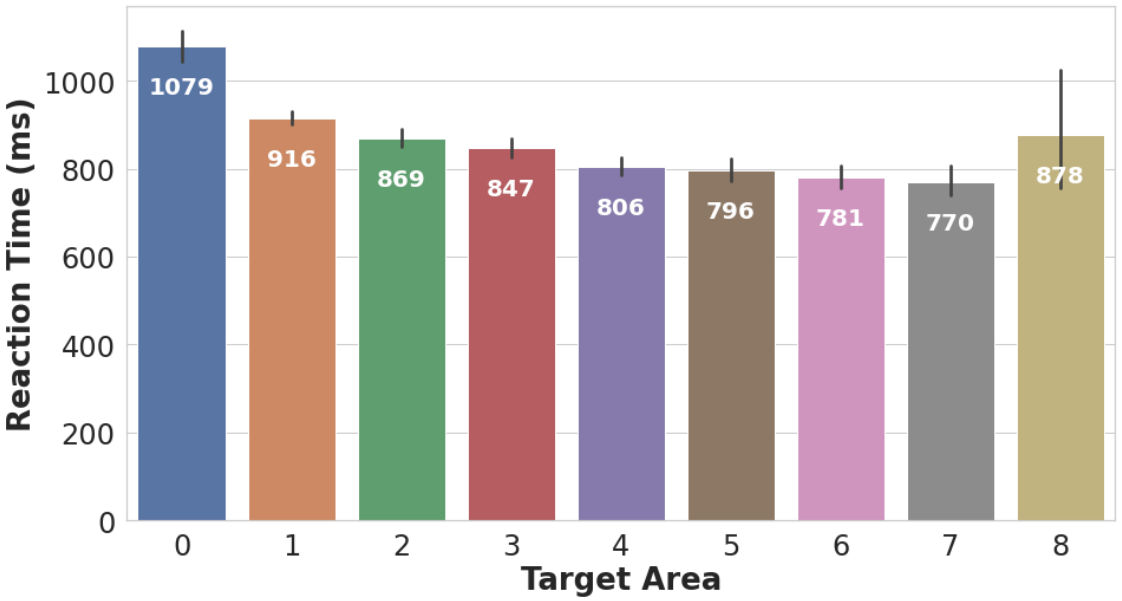
The average fixation duration on the target objects is equal among participants with a small variation of 30 milliseconds. However, the fixation duration on distracting objects is more varied, with a maximum difference of 115 milliseconds. Subject 3 and 6 have respectively the longest and the shortest distractor fixation duration. Interestingly, these two subjects have an almost equal reaction time. Subject 3 tends to make fewer number of fixations but longer, while subject 6 tends to make more fixations but shorter. Thus their average overall reaction time is almost equal, with only 30 milliseconds higher average for subject 6.

The characteristics of a target object have significant impact on search performance. One of these important characteristics is the size or area of the target object. We assume that the

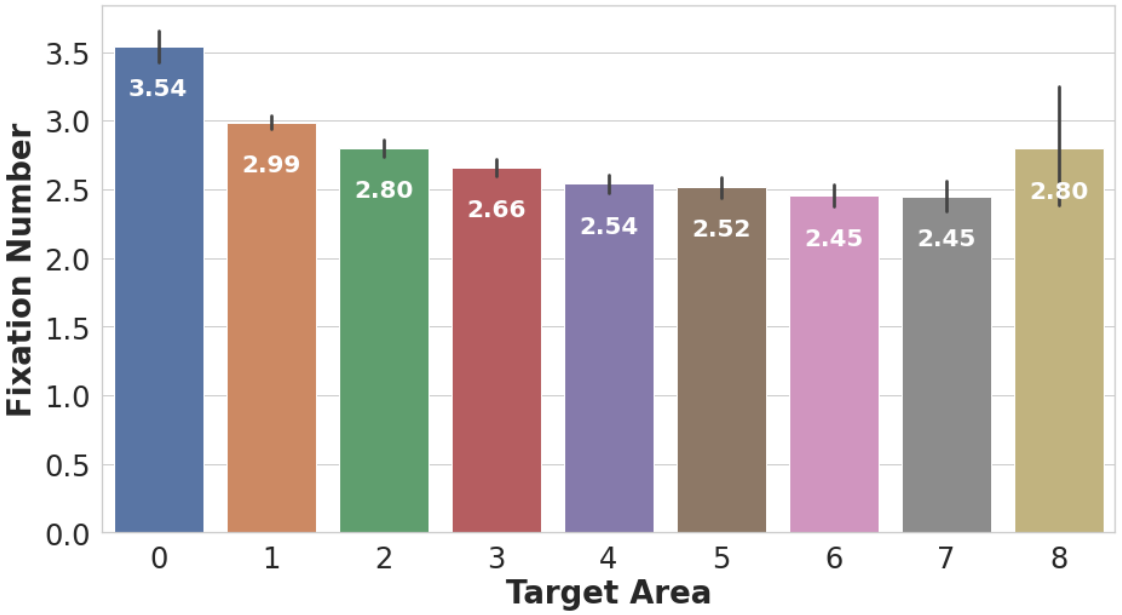
bigger the target, the higher the visibility. So the bigger targets should be found faster and with fewer fixations. We investigate this assumption with multiple figures, by plotting the average reaction time, average number of fixations (excluding the initial center fixation, including the target fixations), average non-target fixation duration time (excluding the initial center fixation), average target fixation duration time (excluding the initial center fixation), and average number of fixations on the target object versus the target area, in figure 3.3. Target area is calculated by multiplying the width and height of the bounding box surrounding the target object. The calculated areas are then separated into bins, where every bin has a range of 2000 pixels. Thus, bin 0 covers 0-2000 pixel area, bin 1 covers 2000-4000 area, ..., and bin 8 covers 16000-18000 area. There is no target object with area bigger than 18000 pixels in images.

In figure 3.3, we observe that the response time is higher for smaller targets and lower for bigger targets, excluding the last bin of the histogram which contains the biggest target size. The results of the last bin are less accurate as suggested by the error bars. This is because we have fewer cases where the target object area is between 16000-18000 pixels. We can thus rely on the rest of the bins to estimate the relationship between the variables. The number of fixations as well as distractor fixation duration time also decrease as the target area increases (probably due to pop-out effect, i.e. the target object stands out from the non-targets). The target duration time also decreases for bigger targets. This might be due to less focus needed to register bigger targets' information. The average number of fixations made on the target is relatively constant, with slightly more average fixations on bigger targets. This is intuitive; as the bigger the target, the higher the probability of fixating at it, since it occupies more space in the image.

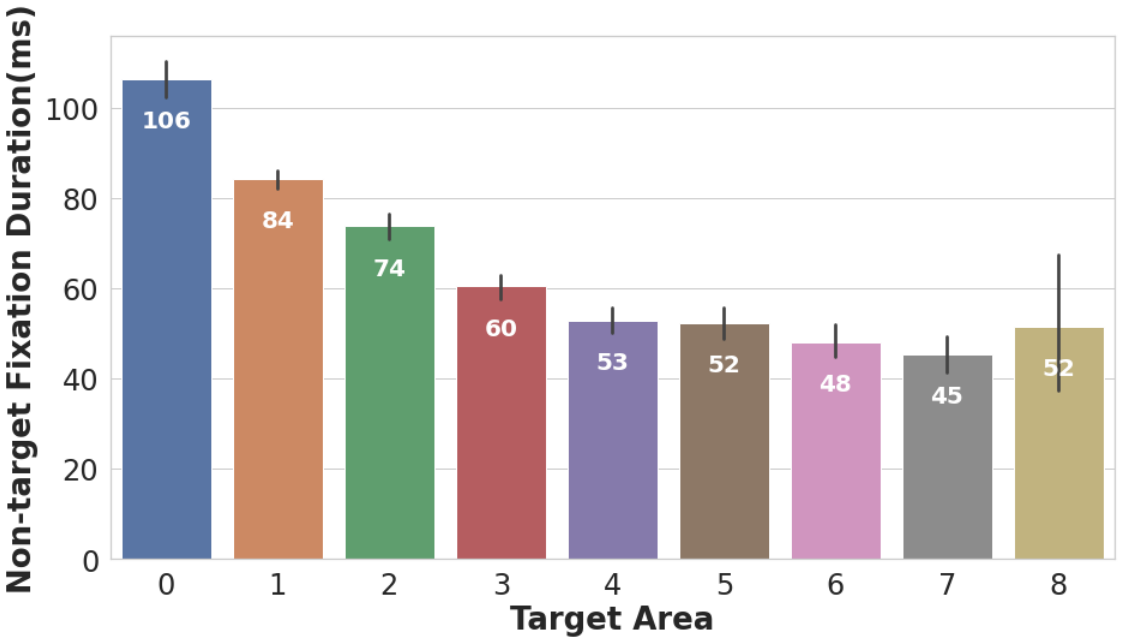
Another crucial factor in determining the search performance is the eccentricity of the target objects, i.e. the distance of the target from the center of the image. It has been believed that humans tend to look toward the central parts of images. Also, during trials the participants initially placed their gaze on a dot in the center of images. We therefore assume that if a target is at the center of the image, participants should be able to find it more rapidly. We test our hypothesis by plotting the previous variables but this time versus the targets eccentricity. We compute the euclidean distance between the center pixel of the target's bounding box and the center pixel of the image. We then


Reaction Time (ms)
1000
800
600
400
200
0
1079
916
869
847
806
796
781
770
878
0
1
2
3
4
5
6
7
8
Target Area

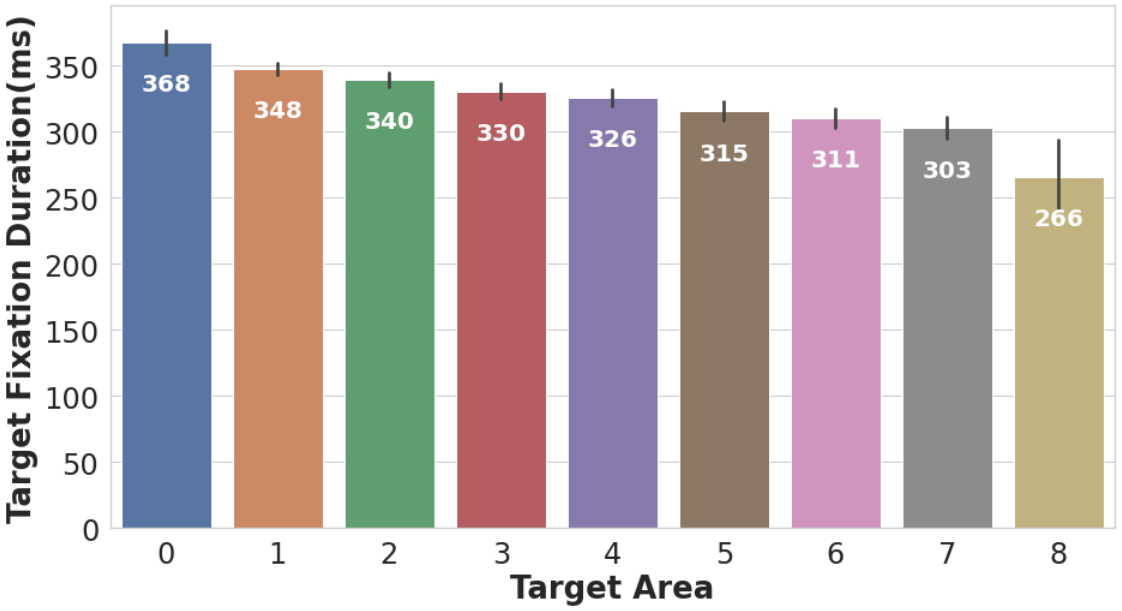
(a)


Fixation Number
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
3.54
2.99
2.80
2.66
2.54
2.52
2.45
2.45
2.80
0
1
2
3
4
5
6
7
8
Target Area

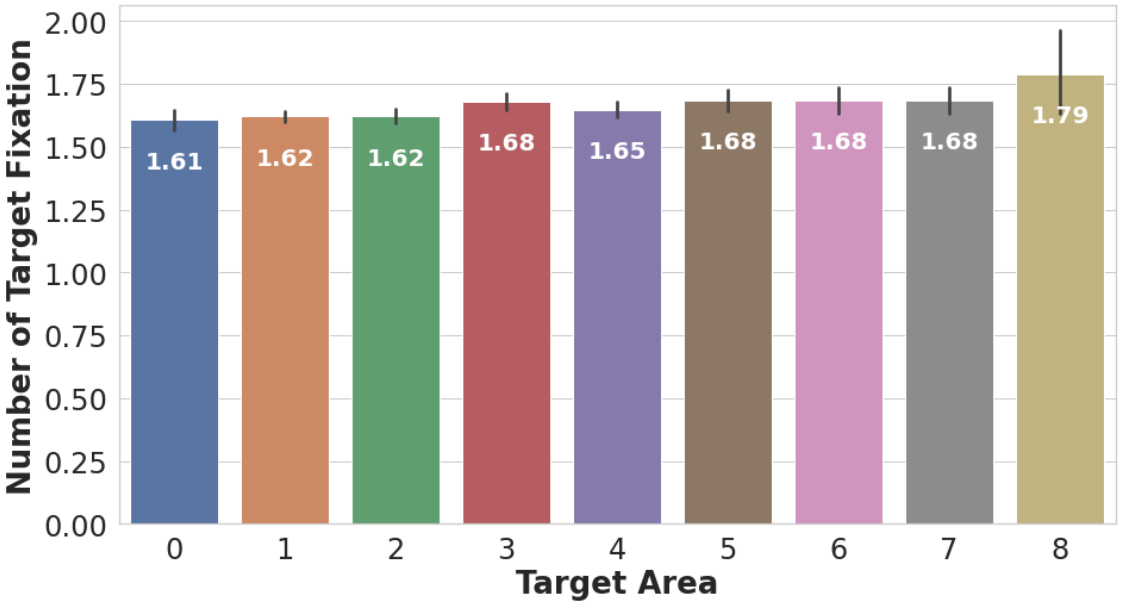
(b)


Non-target Fixation Duration(ms)
0
20
40
60
80
100
106
84
74
60
53
52
48
45
52
0
1
2
3
4
5
6
7
8
Target Area

(c)


Target Fixation Duration(ms)
0
50
100
150
200
250
300
350
368
348
340
330
326
315
311
303
266
0
1
2
3
4
5
6
7
8
Target Area

(d)



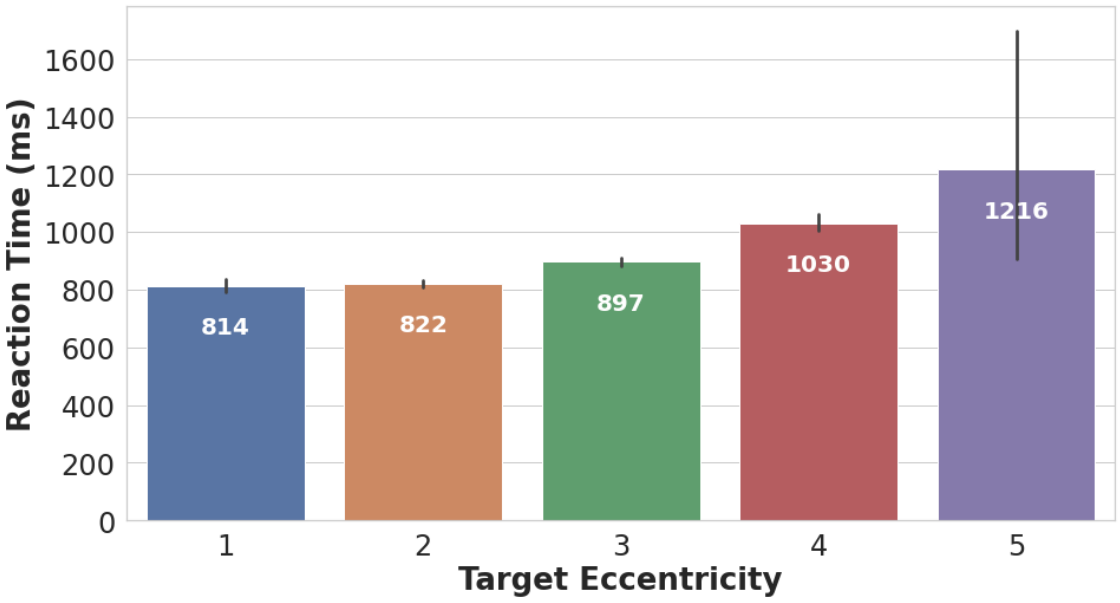
(e)

Figure 3.3: Search performance versus target object size (area). (a) Average reaction time. (b) Average fixation number. (c) Average non-target fixation duration. (d) Average target fixation duration. (e) Average number of target fixation.

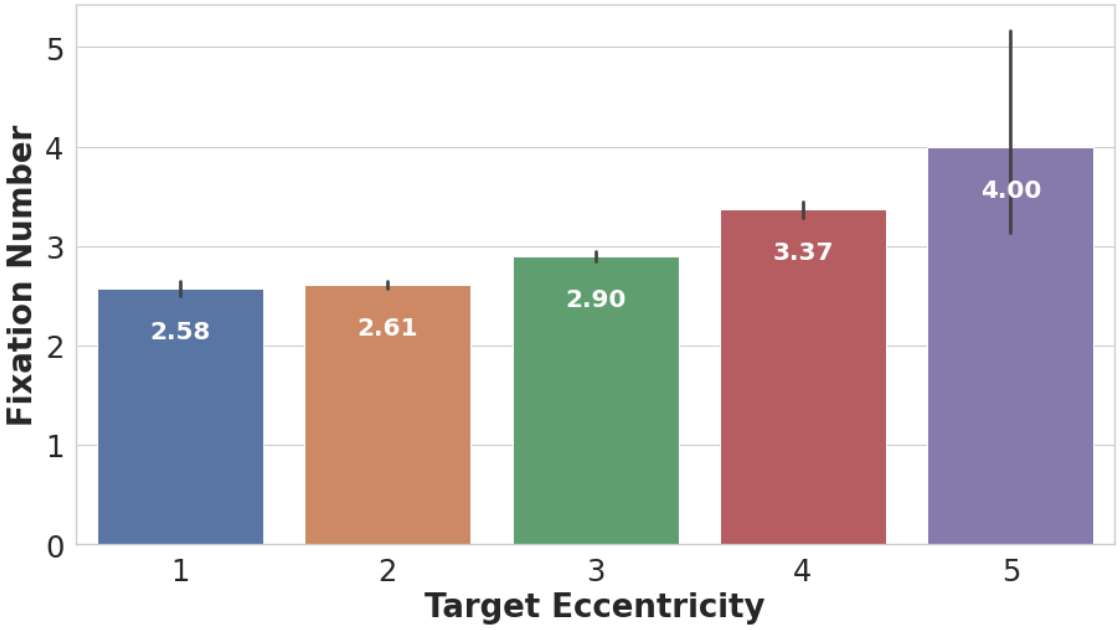
divide the distances into 6 bins, each containing 50 pixel euclidean distance. Bin 0 corresponds to 0-50 pixel distance, bin 1 contains 50-100 pixel distance, ..., and bin 5 contains 250-300 pixel distance. The results can be seen in figure 3.4

The results show that the average reaction time, average number of fixations and average distractor fixation duration increase as the target's distance from the center increases. This validates our previous assumption that targets further away from the center are harder to find with more fixations and longer response time. The observers also fixate at more non-target objects while looking for the distant targets and therefore have higher distractors fixation duration. The number of fixations made on the target and the duration of these fixations are almost constant across various levels of targets' eccentricity.

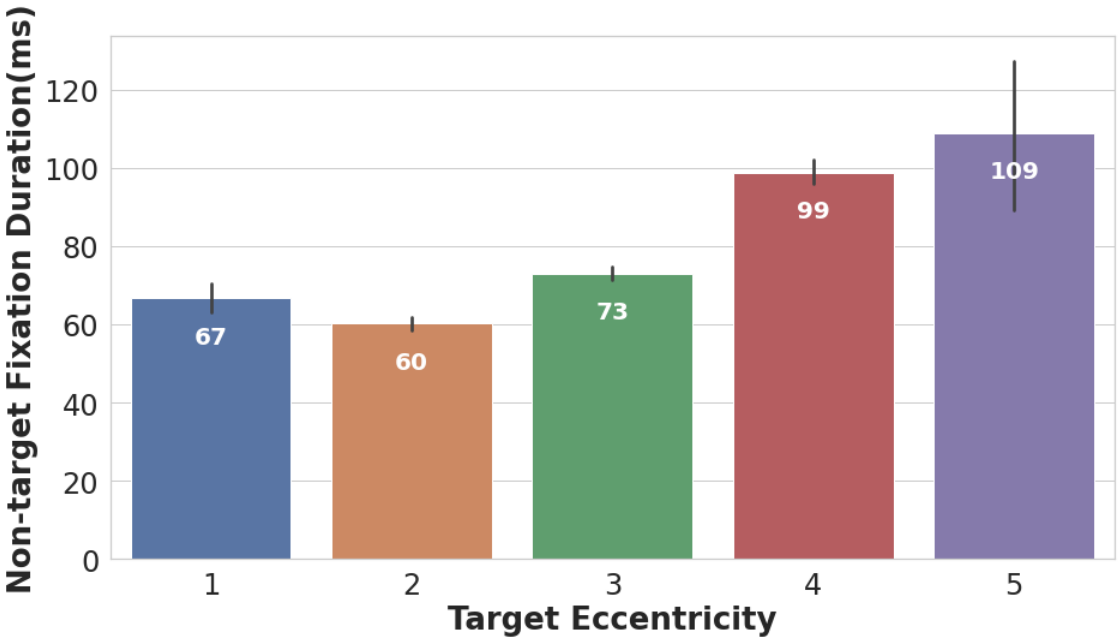
Finally, we investigate the correlation between the error rate of participants with average reaction time, average number of fixations, average non-target fixation duration, and average target


Reaction Time (ms)
1600
1400
1200
1000
800
600
400
200
0
814
822
897
1030
1216
1
2
3
4
5
Target Eccentricity

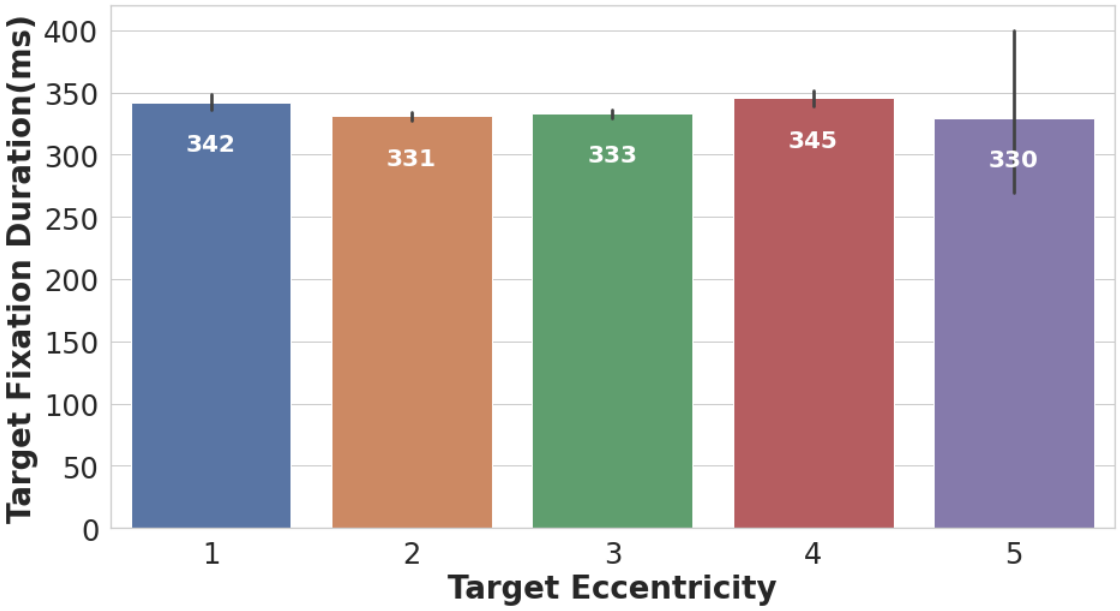
(a)


Fixation Number
5
4
3
2
1
0
2.58
2.61
2.90
3.37
4.00
1
2
3
4
5
Target Eccentricity

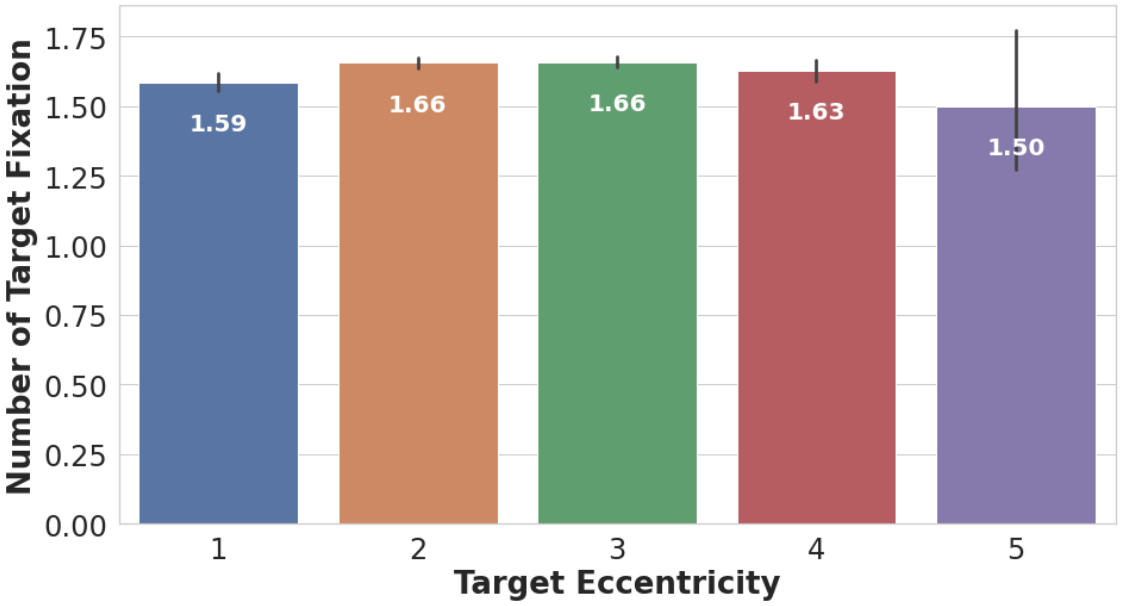
(b)


Non-target Fixation Duration(ms)
0
20
40
60
80
100
120
67
60
73
99
109
1
2
3
4
5
Target Eccentricity

(c)


Target Fixation Duration(ms)
0
50
100
150
200
250
300
350
400
342
331
333
345
330
1
2
3
4
5
Target Eccentricity

(d)

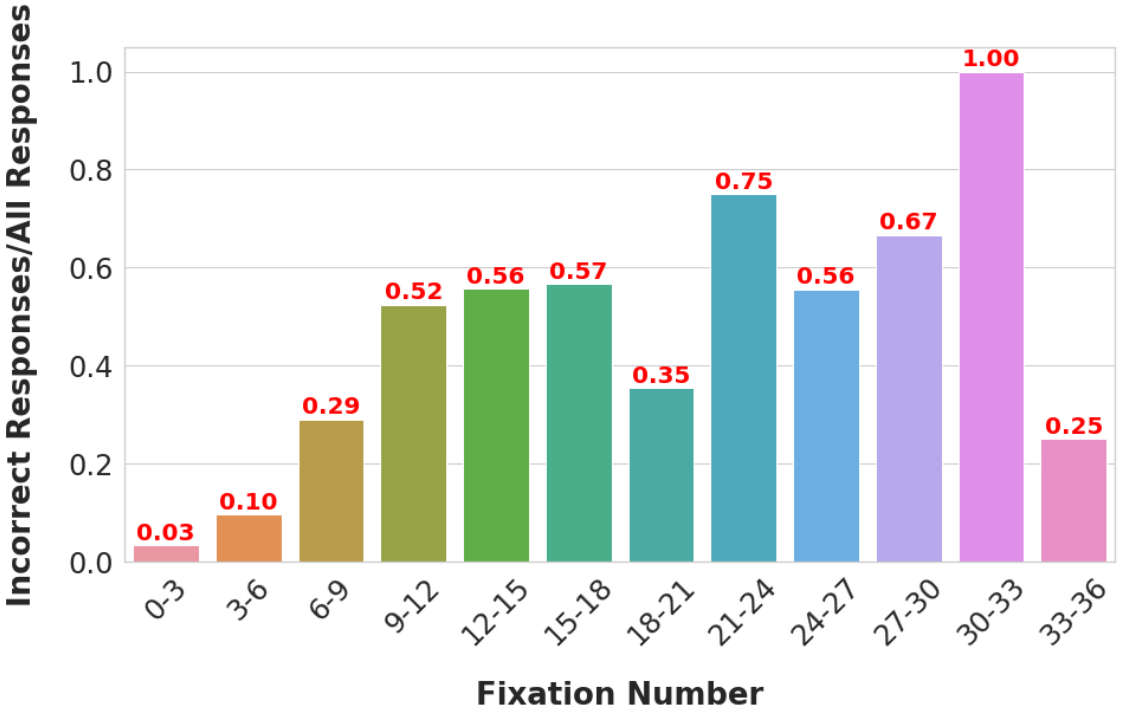


(e)

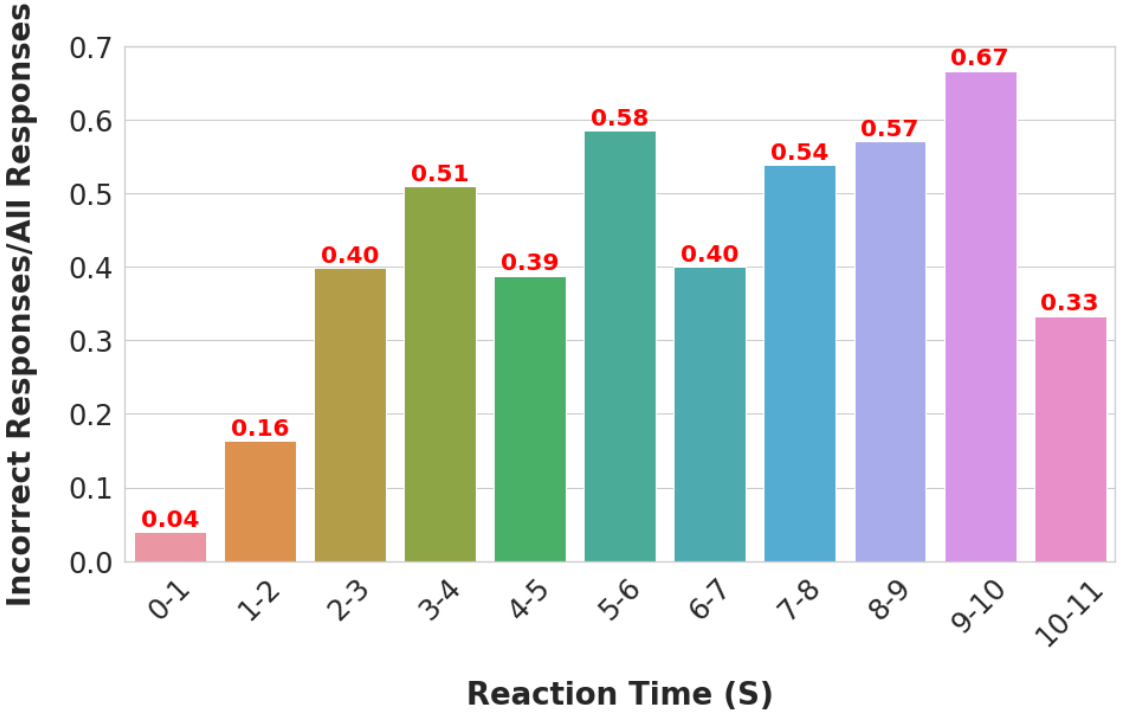
Figure 3.4: Search performance versus target eccentricity (target's euclidean distance from the center of an image). (a) Average reaction time. (b) Average fixation number. (c) Average non-target fixation duration. (d) Average target fixation duration. (e) Average number of target fixation.

fixation duration, in figure 3.5. The error rate is defined as the number of incorrect responses (by all participants) divided by the total number of responses for each bin of the histograms.

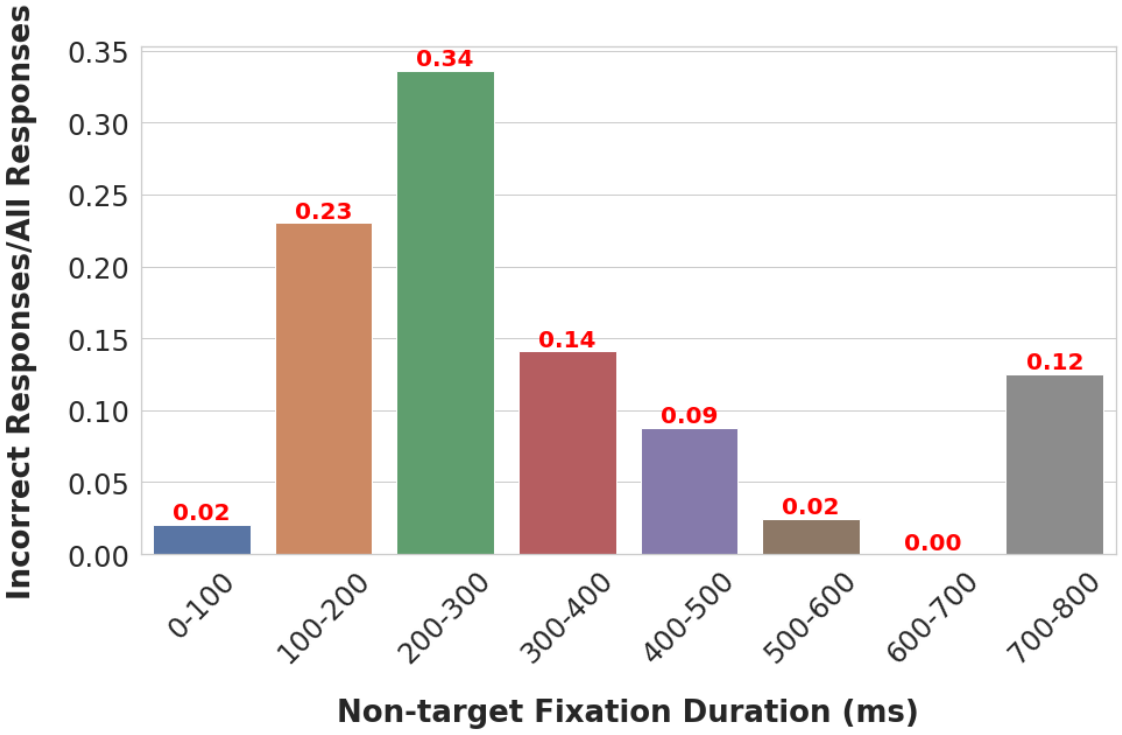
From figure 3.5, we can derive that longer reaction times correspond to higher error rates. As in more difficult search images that the target is less detectable, participants spend longer time searching the images, and some eventually fail to find the target. This explanation does not contradict our earlier discussion on individual differences in reaction time. We previously stated that participants who have longer average reaction time had less number of incorrect responses, as a result of their more careful search. However, trials with higher reaction times contain both participants who perform exhaustive search and those who perform non-exhaustive search but are involved in a more difficult search that makes their reaction times longer. We previously discussed that the exhaustive searchers make less errors; therefore, we can attribute the higher error rates to the second case; where fast searchers are involved in a difficult search. The same pattern exists for the number of fixations. A higher number of fixation typically corresponds to a more difficult search, and con-


Incorrect Responses/All Responses
1.0
0.8
0.6
0.4
0.2
0.0
0.03
0.10
0.29
0.52
0.56
0.57
0.35
0.75
0.56
0.67
1.00
0.25
0-3
3-6
6-9
9-12
12-15
15-18
18-21
21-24
24-27
27-30
30-33
33-36
Fixation Number

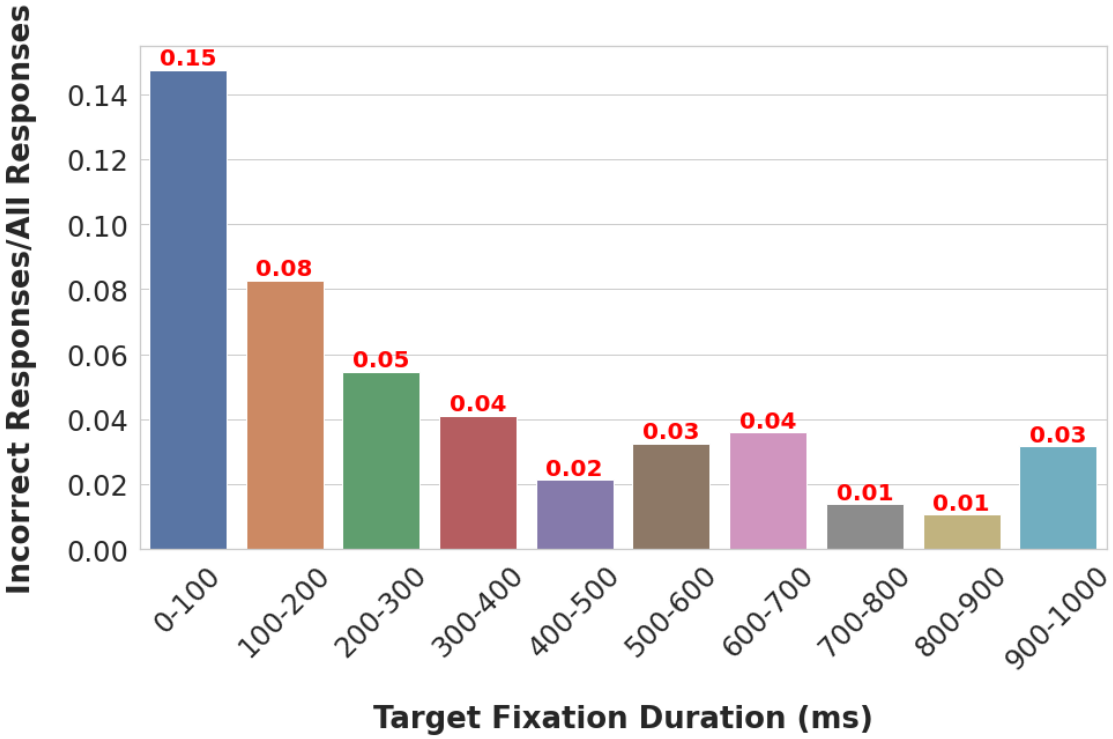
(a)


Incorrect Responses/All Responses
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.04
0.16
0.40
0.51
0.39
0.58
0.40
0.54
0.57
0.67
0.33
0-1
1-2
2-3
3-4
4-5
5-6
6-7
7-8
8-9
9-10
10-11
Reaction Time (S)

(b)



(c)



(d)

Figure 3.5: Ratio of the incorrect responses to the total number of responses versus: (a) Reaction time. (b) Fixation number. (c) Non-target fixation duration. (d) Target fixation duration.

sequently higher error rates. Furthermore, fixating at non-target objects for a duration of 200-300 milliseconds correspond to the highest erroneous responses. Also, shorter fixations at target objects correspond to more incorrect answers. Thus we can infer that when participants recognize the target, they fixate at it longer than when they look at the target but do not recognize it.

3.2 Method 1: Predicting Salience During Search

In this section, we propose a unified model that can predict the locations in an image that are likely to be considered salient by a human observer while searching for a specific target category. Our network thus aims to generate the fixation density maps of observers. In some research papers fixation density maps are referred to as saliency maps. However, we avoid this terminology as suggested by Kummerer et al. [66]; they propose that a saliency map should be defined as a metric-specific prediction derived from the model fixation density, in order to generate the highest performance of the model for each saliency metric (such as IG, NSS, CC, etc). On the other hand, a model's FDM (fixation density map) contains the the probability $p(x, y|I)$ of observing a fixation at a given pixel in a given image.

3.2.1 Data Pre-processing

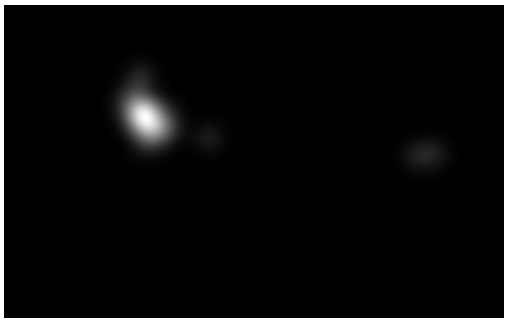
To create the ground-truth fixation density maps from the fixation data, we use Gaussian blurring kernels on the fixation points. The original images of COCO-Search18 dataset are 1050 by 1680 pixels. For more efficient computation we resize the images to 320 by 512 pixels. To transform the fixation locations to the new image dimension, we first normalize the fixation coordinates between 0 and 1 and then multiply them by the new dimensions (320 , 512). Then, for each fixation location, except the initial center fixation and those that are out of the search screen frame, we apply a Gaussian kernel (with width and height of the search image) centered at that point with standard deviation of 11 (chosen manually). We sum these Gaussians on the image plane, and then normalize the map between 0 and 1 (dividing by the maximum value in the map). This creates the fixation density maps. It is still not a probability distribution as the pixel values do not sum to

one. Dividing the pixel values by their sum over an image will convert density map to a probability distribution. However, for the purpose of visualization, we do not divide the density maps by their pixels' summation. Instead, we multiply the fixation density maps by 255 to expand the values between 0 and 255. To generate heat-maps, we apply jet color-map with a threshold of 10 to the generated FDMs. We also create another map that takes into account the fixations' duration time. We multiply Gaussian kernels by the fixation duration before summing them together. We call these maps 'gaze-duration maps'. Multiplying fixations by their duration will affect the heatmaps such that the longer fixations will standout, and shorter fixations will smooth out. Figure 3.6 shows an example of a generated FDM, gaze heat-map, gaze-duration density map and gaze-duration heatmap for a search image. As can be seen in 3.6, gaze-duration maps are more focused on the target object with fewer distractor fixations and therefore are easier to model. However, since we aim to model fixated distractors along with the target and also for agreement with saliency literature, we use solely spatial fixation density maps in our main modeling.

After removing the trials with incorrect responses from our dataset, there will remain 2150 training and 324 validation task-images in COCOSearch18 dataset. Some of the search images are used more than once for different search targets. Hence we need to define task-image pairs to distinguish between the search targets in similar search images. From the 2150 training task-image pairs, we then separate 324 pairs for the test dataset, such that there are 18 task-images from 18 different categories. Hence, our train-validation-test split contains 1826-324-324 task-image pairs. Training data is used to fit a model that learns to map the input search images to a fixation density map (as in MSI-Net [63]). Validation data is used to provide unbiased evaluation of a model fitted on the training set through tuning the hyper-parameters. During epochs of training, the model starts to fit to the validation data, and this evaluation becomes more biased. Test data is then used to provide a complete unbiased estimate of the final model's performance. A histogram of the number of task-image instances for train, validation, and test splits over target categories can be seen in figure 3.7.

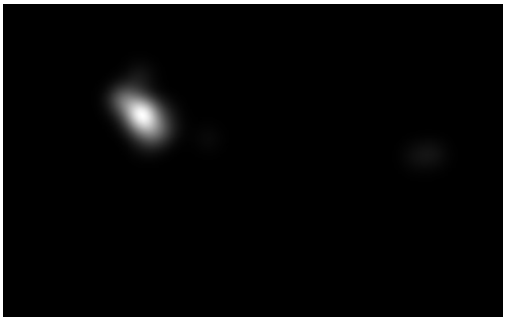
We augment the training data by horizontally flipping the search images along with their ground-truth fixation density maps. This will result in 3652 training instances. The motivation be-

(a) Gaze heat-map overlaid on the search image.



(b) Ground truth fixation density map.

(c) Gaze-duration heat-map overlaid on the search image.



(d) Fixation-duration density map.

Figure 3.6: The fixation maps are generated by the sum of the blurred fixation points of all participants on each search image. All FDMs are 320 × 512 pixels. Gaze-duration maps are generated through scaling Gaussian blurring kernels by fixations' duration. We can see that a short fixation on a sandwich in the gaze heatmap is smoothed out in the gaze-duration map.

hind choosing horizontal flip is originated from paper [17]. They applied several common transformations such as cropping, flipping (mirroring), rotating, changing brightness, and shearing to the search images and their fixation maps. Then they obtained the ground-truth fixation maps for those transformed images, by recording observers' eye movements. They compared the ground-truth and the transformed fixations maps to investigate which transformation causes the least change in the fixation maps. Among these transformations, horizontal flip had the least effect.

3.2.2 Model Architecture

The architecture of our network is a convolutional encoder-decoder, same as MSI-Net [63], that maps an input image to a fixation density map. The encoder extracts relevant features from input

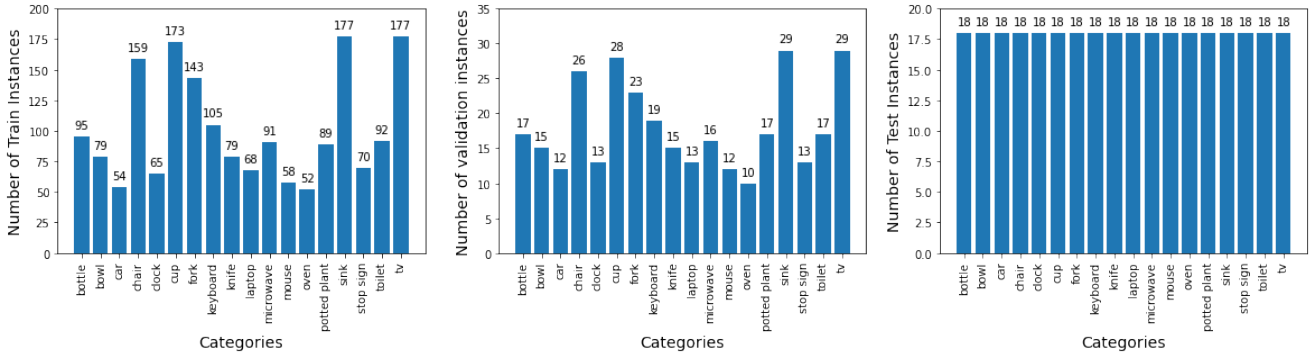


Figure 3.7: Distribution of task-image instances across target categories for train, validation, and test splits.

images. Inspired by [113], our encoder is composed of two streams: the first stream extracts features from the search image and the second stream extracts features from a sample image of the target category. The parameters of the networks are shared between the two streams. The sample target fed into the target stream is not the same as the target object that exists in the search image. We use 18 fixed sample images from the 18 target categories to be received by the target stream. These sample target objects can be seen in figure 3.8.

Figure 3.8: Sample objects for 18 target categories. For instance, when we aim to generate fixation density map of searching for a TV, we feed the TV sample image, presented in this figure, to the target stream of our network. All sample targets are resized to 64×64 dimension before entering the target stream.

Each encoder stream is composed of two sections. The first part contains a VGG16 convolutional neural network [93] pretrained on ImageNet dataset [92]. Unlike [113] that uses pretrained VGG16 for feature extraction in a forward pass manner without training, we train our two-stream network end-to-end. VGG16 is composed of 5 blocks of convolutional layers followed by pooling layers, and three final fully-connected layers. For the purpose of feature extraction, we remove the last fully-connected layers. Each of the first two blocks are composed of two convolutional layers and each of the remaining three blocks are composed of three convolutional layers. All layers have 3×3 kernels. The convolutional layers of the five blocks have 64, 128, 256, 512, and 512 filters from input to output respectively. All layers use zero-padding to keep the width and height of their input tensors unchanged. Hence, only pooling layers cause reduction in dimensions. As in MSI-Net we remove the strides of the last two pooling layers in VGG16 to keep the width and height of the features as $1/8$ of the input image. Also, the last three convolutional layers have a dilation rate of 2 to compensate for the higher resolution. Dilated convolutions [110] have specific spacing between the values of their kernels. For example, a 3×3 kernel with a dilation rate of 2 has the same field of view as a 1-dilated 5×5 kernel, while having fewer parameters than a 5×5 kernel (9 vs. 25). If we delete the even columns and rows in a 1-dilated 5×5 kernel we will produce a 2-dilated 3x3 kernel, as seen in figure 3.9.

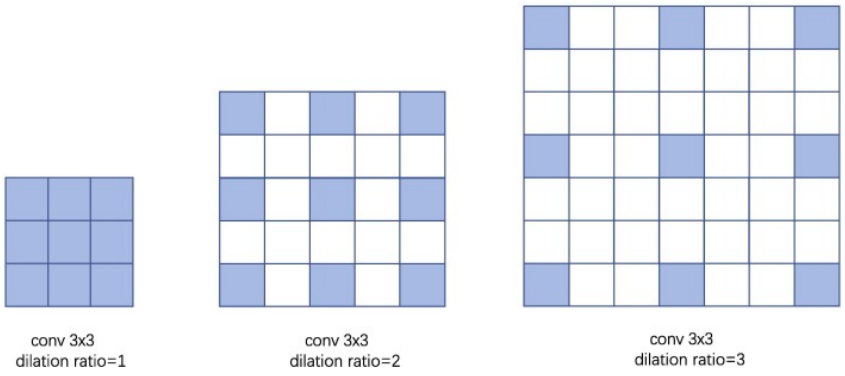


Figure 3.9: Dilated convolutions. Figure credit: [70]

To obtain a multi-scale feature representation, we concatenate the output of the three last pooling layers, all of which have an output dimension of $1/8$ of the input image. The pooling layer in block three has 256 output channels and pooling layers of blocks four and five each has 512 chan-

nels. Hence the concatenated features tensor has 1280 output channels. This tensor is then sent to another feature extraction block called 'Atrous Spatial Pyramid Pooling' abbreviated as 'ASPP'. ASPP, introduced by [18], is a semantic segmentation module that is composed of convolutional layers with different dilation rates ('Atrous' convolution is another term for dilated convolution). This causes the layers to use filters with different effective fields of view, that capture objects and contextual information at different scales. The ASPP architecture used in MSI-Net and our network is composed of six convolutional layers. Four convolutional layers receive the extracted feature tensor (with 1280 channels) directly as their input. One of them has 1×1 kernels and only perform point-wise non-linearity without learning spatial relationships. The other three have 3×3 kernels with 4, 8, or 12 dilation rate, which allows extracting multi-scale dependencies from the feature tensor. The fifth convolution layer receives the mean of the features across spatial dimensions, which is believed to contain global contextual information of the image. This convolution then applies point-wise non-linearity with 1×1 kernels to the mean tensor. The output of this convolution is then upsampled to match the spatial dimension of the original feature tensor. Finally, the output of all these five convolutions are concatenated to form a single tensor. As the output of each convolution has 256 channels, the concatenated tensor yields 1280 channels. The concatenated tensor is then passed through another convolutional layer with point-wise 1×1 kernels and ReLU non-linearity. The output of this layer yields the final multi-scale feature tensor with 256 channels. The architecture of ASPP can be seen in image 3.10.

Both target and stimulus streams use the same encoder and ASPP architecture to extract features from the sample target image and search image. The parameters of these two streams are shared. Then the extracted features from the target and search image are convolved together. The convolution is performed using a convolutional layer with the target feature tensor treated as the filter and the search image feature tensor as the input to the layer, without any non-linear activation function. The output of this convolution has 256 channels, which is then passed through the decoder part to retrieve the spatial dimension of the input image. Decoder is composed of three bi-linear up-sampling blocks, followed by 3×3 convolution layers to avoid checkerboard artifacts. There are four convolution layers in the decoder with 128, 64, 32, and 1 filters. The first three

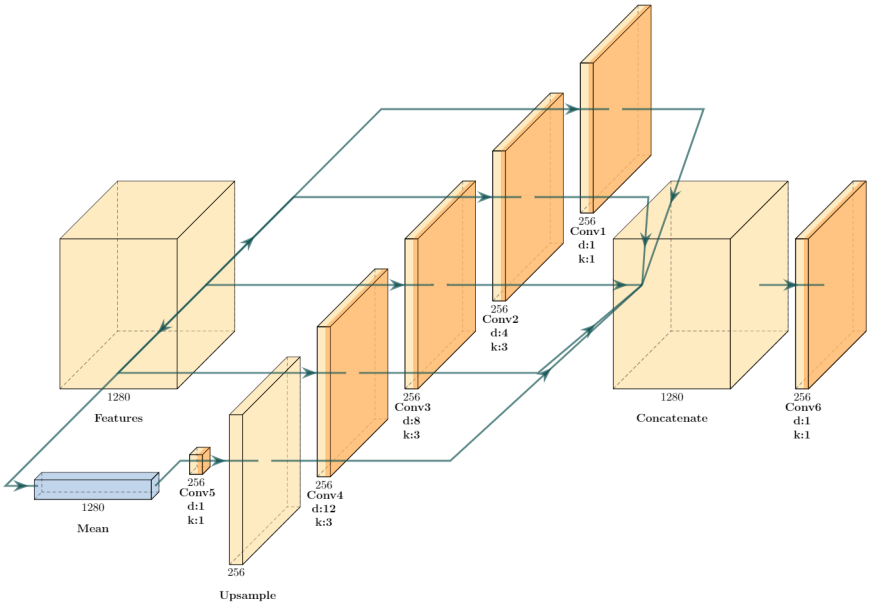


Figure 3.10: ASPP Architecture.

convolutions that come immediately after up-sampling layers are followed by ReLU non-linearity; while, the last convolution layer that generates the output FDM is not modified by a ReLU. The output values are then normalized between 0 and 1, such that all values sum to one. This modification converts the generated map to a probabilistic density map. A visualization of our network is presented in figure 3.11.

3.2.3 Training

We initialize the VGG16 network with its pre-trained weights on ImageNet dataset for object recognition task, and then fine-tune (by adjusting all layers' weights) it on COCOSearch18 for our problem. It has been shown that high-level features extracted from object recognition networks can improve the performance of free-viewing saliency prediction. We therefore apply the same technique for visual search saliency. The weights of VGG networks used in both streams are shared. This weight sharing reduces the network's complexity significantly, while also improving the accuracy.

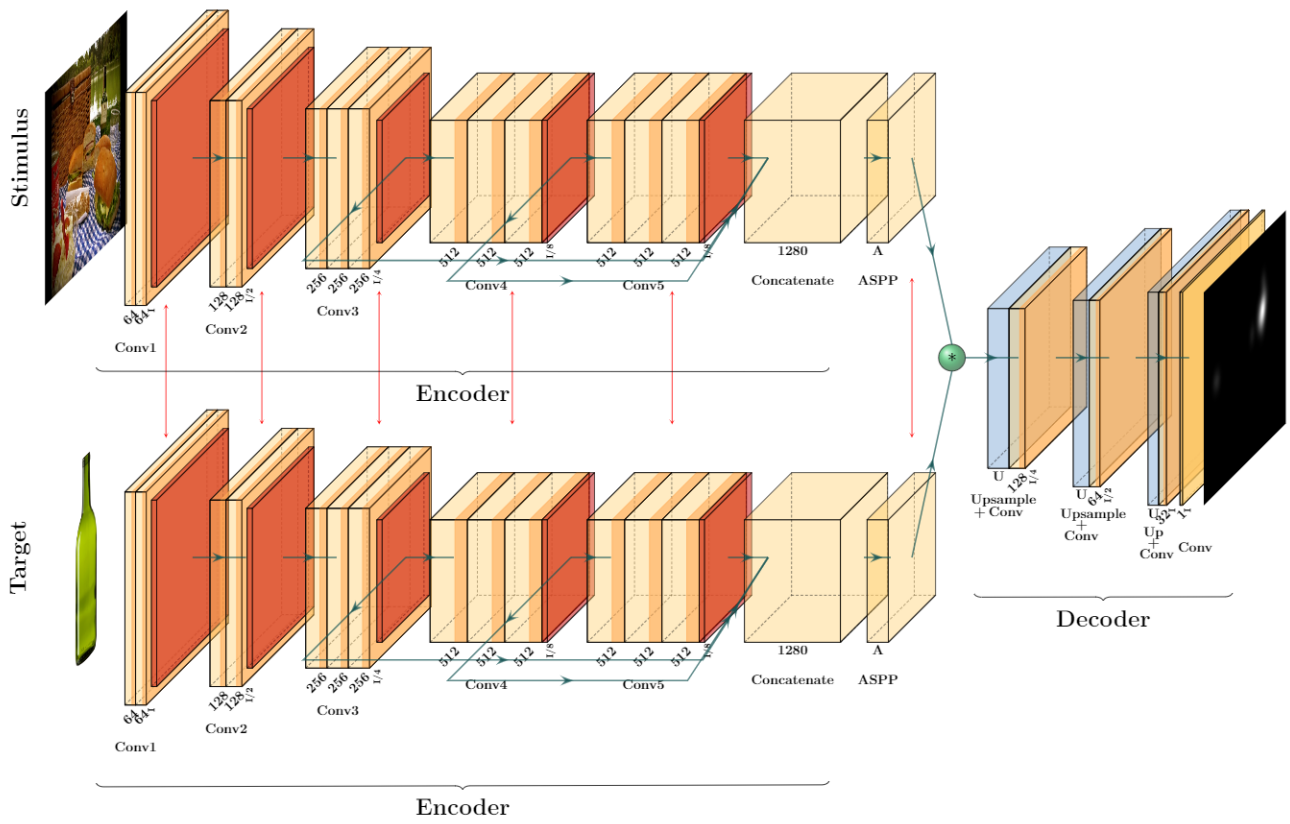


Figure 3.11: Network Architecture. The red arrows indicate that the weights of the convolution layers in encoder and ASPP modules are shared between the two streams.

We train the network on COCOSearch18 dataset by minimizing KL-divergence loss 3.1 between the predicted and ground-truth fixation density maps. We use Adam optimizer with learning rate of 10^{-5}, and train the network for 10 epochs, with batch size of 1. The input search images are all resized to 320×512, and the target images are resized to 64×64. The output fixation density map is also 320×512 same as the input search image.

$$D_{KL}(P||Q) = \sum_i Q_i \ln\left(\varepsilon + \frac{Q_i}{\varepsilon + P_i}\right) \tag{3.1}$$

3.2.4 Experiments

We examine our trained network on 6 saliency metrics: AUC, sAUC, NSS, KLD, CC, and SIM (previously discussed in chapter 2 in section "Saliency Evaluation Metrics") using pysaliency li-

brary [76]. The performance of the model on unseen test data is listed in table 3.2. The overall performance of the model is obtained on the whole test dataset with 324 images, containing 18 sample images for each of 18 target categories.

We further compare the performance of our unified model with single category-specific networks trained for each target category. Our current unified model has two streams that allows it to receive information about the target category, and output target-specific FDM predictions based on the target. An alternative approach is to train separate one-stream networks for each of 18 target categories. As these network are inherently target-specific, they do not need a separate stream for a sample target. For a better comparison, we report the results of our two-stream network on test images of separate categories (18 test images for each category) along with the results of single category-specific trained models in table 3.2. Although in some cases one-stream models trained on one category achieve better performance and are faster to train (each training epoch for 1-stream models take around 2 minutes while each epoch of two-stream model takes 11-12 minutes.), a unified two-stream model that works for all categories demands far less memory and is more biologically plausible.

The bold numbers in 3.2 show the highest performance of the two-stream model among different target categories. Our two-stream model achieves the highest AUC for the microwave and oven categories, highest NSS, CC, and SIM for the stop sign category, and lowest KL divergence for microwave category. The sAUC is highest for the overall test set with a mixture of all categories.

Since there is no previous work that models visual search fixation location on COCOSearch18 dataset, we use the MIT/Tubingen saliency benchmark, which is for free-viewing saliency, to gain a sense of how good our model performs on saliency metrics. Our overall model has excellent AUC,sAUC, NSS, and CC scores, corresponding to the top 5 models of the MIT/Tubingen saliency benchmark. But KLD and SIM scores are not as satisfactory, and correspond to the top 16 models of the MIT/Tubingen benchmark. However these comparisons are not very reliable as we are using a free-viewing benchmark to verify the performance of a visual search saliency model.

To gain insight into the qualitative performance of our model, we visualized the network's predicted density maps and compared them to the ground-truth density maps for the test dataset.

Some of these visualizations are shown in figures 3.12 and 3.13. The qualitative results show that the network accomplishes to detect the targets in most of the images. It also learns where to look for each target category based on its experience in the training data. For instance, when looking for oven or toilet, the locations closer to the ground are considered salient; however, for bottle, bowl, laptop, fork, knife, mouse, and cup, the top of surfaces such as tables are mostly salient. While searching for TV and clock the higher items in images are often considered salient. Moreover, the items sharing some features with the objects of target category are considered salient both by the network and human observers. As an example, when observers are looking for a bottle, objects such as glasses and cups that are also relatively cylindrical, are fixated. In general, the network's predictions and ground-truth FDMs have great similarities with many shared distraction patterns.

We investigate the effect of data augmentation, presence of ASPP, presence of decoder, initializing VGG with random weights, and not sharing weights between the target and search image streams.

We realized that with data augmentation (horizontal flip) the training KL loss decreased by 0.05. However, adding more augmentations such as vertical flipping led to decreased performance. The reason might be due to the effect of vertical flipping transformation on the fixation pattern. The fixation map of a vertically flipped image might no longer be in accordance with the vertically flipped fixation map of the original image. This would create erroneous data, which causes the model's performance to drop.

To explore the effect of ASPP architecture, we removed it and directly fed the concatenated VGG16 features to a 1×1 convolution with 256 filters. Then we passed the output to the decoder. This modification increased training KL loss by approximately 0.05, which validates the effectiveness of ASPP architecture in feature extraction.

As a second architecture change, we removed the decoder from the network, and up-sampled the output of the convolved streams by 8 in one shot. Then we normalized the output to create a probabilistic density map. Removing the decoder leads to checkerboard artifacts in the predicted density maps, and the maps seems to be more sparse.

Category	Model	Saliency Metrics					
		AUC	**sAUC**	**NSS**	**KLD**	**CC**	**SIM**
All	**2-stream**	0.89	**0.81**	3.09	1.77	0.73	0.54
Bottle	**2-stream**	0.88	0.79	3.22	2.17	0.77	0.57
	1-stream	0.87	0.77	3.20	1.93	0.76	0.56
Bowl	**2-stream**	0.86	0.74	2.49	2.32	0.71	0.50
	1-stream	0.87	0.77	2.52	1.00	0.71	0.47
Car	**2-stream**	0.87	0.79	2.42	2.11	0.57	0.45
	1-stream	0.84	0.77	2.58	1.34	0.63	0.45
Chair	**2-stream**	0.88	0.75	2.59	2.09	0.71	0.53
	1-stream	0.89	0.77	2.41	1.19	0.60	0.44
Clock	**2-stream**	0.90	0.80	3.57	1.68	0.72	0.51
	1-stream	0.89	0.80	4.42	1.78	0.85	0.62
Cup	**2-stream**	0.89	0.79	2.61	1.07	0.72	0.55
	1-stream	0.88	0.80	2.70	1.02	0.71	0.51
Fork	**2-stream**	0.90	0.78	3.44	1.04	0.81	0.57
	1-stream	0.88	0.81	3.58	1.76	0.82	0.57
Keyboard	**2-stream**	0.91	0.71	2.78	1.22	0.66	0.47
	1-stream	0.90	0.73	2.61	1.19	0.75	0.54
Knife	**2-stream**	0.84	0.74	2.18	2.25	0.57	0.45
	1-stream	0.82	0.75	2.02	1.52	0.54	0.40
Laptop	**2-stream**	0.87	0.78	2.67	2.00	0.63	0.46
	1-stream	0.87	0.77	2.84	1.16	0.70	0.50
Microwave	**2-stream**	**0.92**	0.80	3.22	**0.95**	0.76	0.57
	1-stream	0.89	0.81	3.63	1.67	0.80	0.60
Mouse	**2-stream**	0.86	0.76	3.66	2.71	0.81	0.56
	1-stream	0.86	0.77	2.87	1.41	0.70	0.47
Oven	**2-stream**	**0.92**	0.77	3.13	1.06	0.73	0.57
	1-stream	0.92	0.78	3.32	0.88	0.77	0.56
Potted Plant	**2-stream**	0.88	0.80	3.00	1.79	0.69	0.51
	1-stream	0.89	0.84	3.57	1.03	0.79	0.55
Sink	**2-stream**	0.89	0.77	2.87	1.31	0.74	0.56
	1-stream	0.93	0.80	3.42	0.93	0.82	0.61
Stop Sign	**2-stream**	0.89	0.77	**4.17**	1.35	**0.83**	**0.61**
	1-stream	0.85	0.80	4.43	2.97	0.827	0.65
Toilet	**2-stream**	0.90	0.77	3.16	1.55	0.79	0.56
	1-stream	0.89	0.77	3.18	0.86	0.79	0.57
TV	**2-stream**	0.90	0.77	3.62	1.72	0.82	0.58
	1-stream	0.93	0.80	4.21	0.86	0.85	0.65

Table 3.2: The performance of the model on COCOSearch18 test dataset. The values are averaged over 4 independent runs. The results of the proposed two-stream model and one-stream category-specific models are reported for comparison.

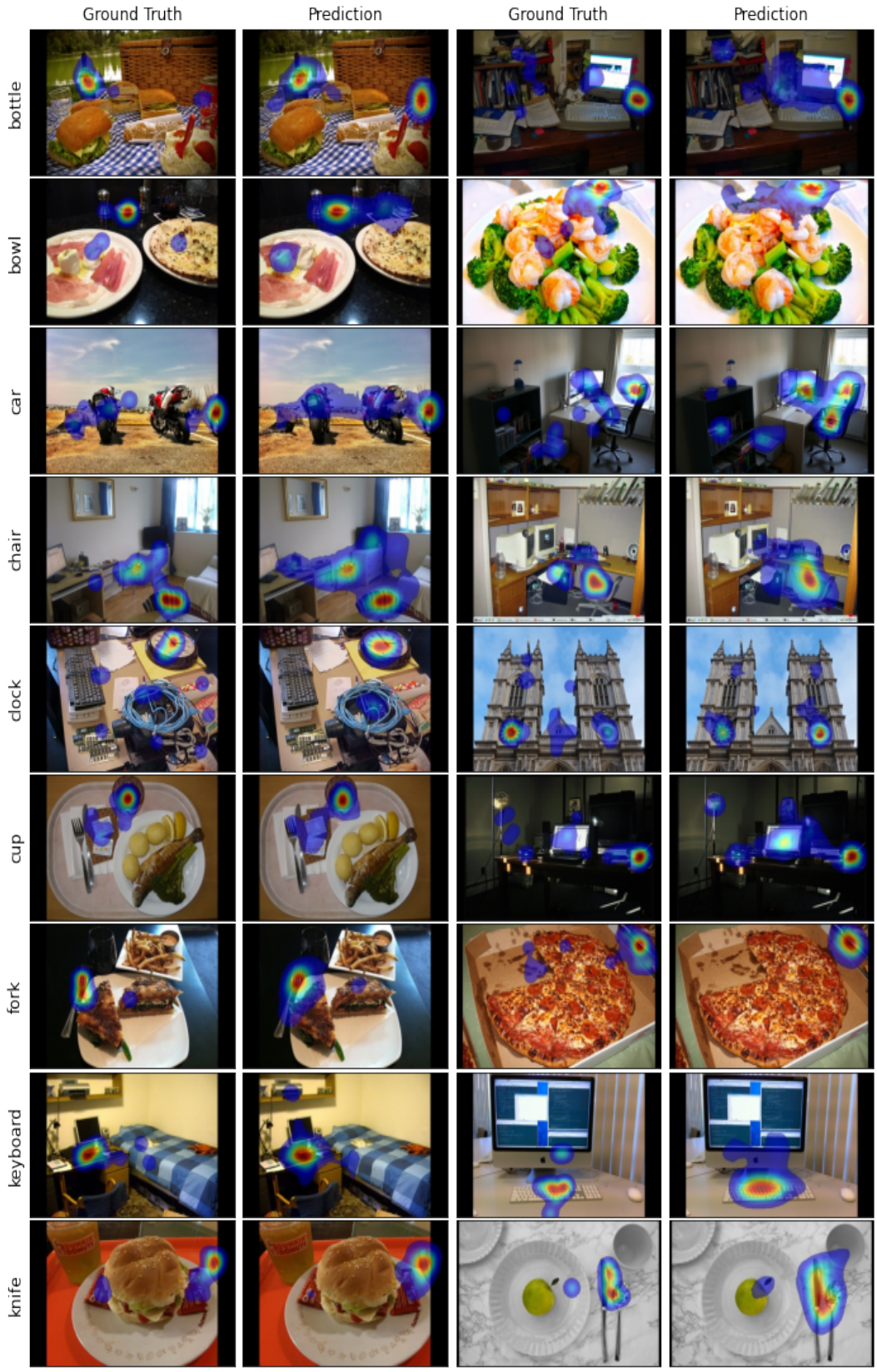


Figure 3.12: A visualization of network's predictions and ground-truth fixation density maps on unseen test data for 9 categories. The target category is specified on the left of each row.

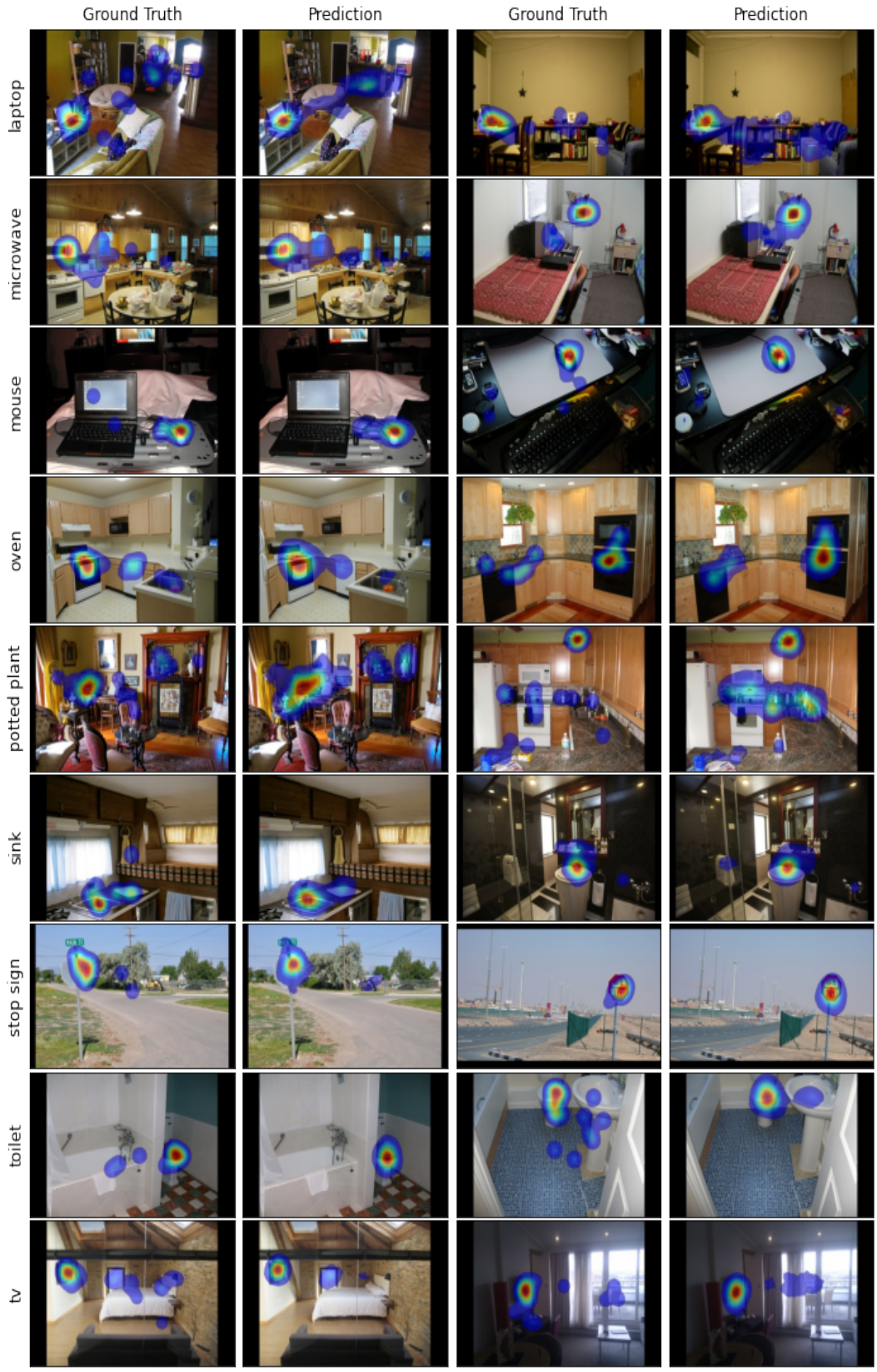


Figure 3.13: A visualization of network's predictions and ground-truth fixation density maps on unseen test data for the remaining 9 categories.

Initializing VGG16 network weights randomly instead of using pretrained weights on ImageNet, dramatically decreases the performance. This modification increased the training KL loss by at least 1.

We tested the scenario where the weights of the encoders and ASPPs are not shared between the target and search image streams. This modification slows down the training process as the number of training parameters are almost doubled. The training loss also increases significantly.

The other important components of our modeling are the sample targets that are fed to the target stream to represent the target category. We realized that the network performance is dependent on the chosen sample targets. For instance, when we changed the sample knife target image from a utility knife to a dinner knife, the accuracy improved as most knives in search images are dinner knives. The more similar these sample target images are to the real targets in search images, the more accurate the model can predict the target fixations. However, feeding the original targets that are present in the search image to the target stream decreases the accuracy. This might be due to the change in the aspect ratio and resolution of the original targets when resized to a fixed size (64×64) to enter the target stream of the network. Also, in COCOSearch18 collection trials, participants did not have access to the original target object, and were only informed about the name of the category. Thus, their search was only guided by their mental image of the target features. Our target stream with a sample target object attempts to apply the same strategy. Also, to remove the model's dependence on the target image, we tested the model's performance while randomly choosing the target image from a set of five example images of a given category. However, we observed a small decrease in model's performance compared to the fixed-target case. We believe that a careful choice of sample target image is key in our modeling, and more experiments are needed to explore the features of a good sample target image.

We further investigate whether pre-training the one-stream category-specific models on SALICON free-viewing dataset can enhance their performance on COCOSearch18. Unlike the free-viewing scenario where pre-training on SALICON dataset leads to performance improvement, in case of visual search saliency, pretraining on SALICON degrades the performance. The main reason comes from the significant difference between a visual search FDM and free-viewing FDM.

In the free-viewing condition, the gaze locations are more spread out over the image, while in visual search gaze is more focused on the target object. Also SALICON has 10000 training samples, almost three times larger than our augmented dataset. Hence, retraining the network on COCOSearch18 after it is pre-trained on SALICON does not change its behavior significantly, and the predicted FDMs are too scattered.

We also investigate the performance of the two-stream model on predicting gaze-duration density maps, which takes into account the fixation duration for each gaze location. The network achieves a higher accuracy in modeling gaze-duration maps compared to gaze maps as the fixations are more focused on the target object.

Same as MSI-Net, our model is relatively light-weight. As the weights are shared between the two streams of our network, our model has 24,934,209 parameters similar to the one-stream MSI-Net. MSI-Net is among the lightest free-viewing saliency models, and here we exploit this feature by extending MSI-Net to a visual search saliency predictor.

3.2.5 Discussion

To the best of our knowledge, the model we proposed here is the first attempt to predict visual search fixation density maps for COCOSearch18. Although the creators of COCOSearch18 have partially worked on fixation density maps in [109] and [21], their main concern is predicting sequential scan-paths. COCOSearch18 is the first dataset that contains a relatively large number of target categories in comparison with the previous eye fixation datasets of visual search tasks. This dataset has the potential for conducting further behavioral research on human's visual attention.

One failure of our proposed model is its confusion in identifying similar target categories such as fork and knife. In most images where a fork and knife are both present, the network mistakes one target with the other. Also our network sometimes generates high fixation probabilities for non-target objects that share close similarity (such as similar color or shape) with the target. Hence the FDM peaks will be mistakenly placed on these non-target objects. This behavior has been also seen in the ground-truth data of human observers. Many of the fixated non-target objects by human

observers have resemblances to the target. In most cases, humans are able to find the target after fixating on the similar non-target objects; while the network sometimes fail to do that.

A potential application of our method could be in visual marketing, i.e. using visual information to guide customers' attention. By extending our method to a 3D environment, we can measure shoppers' distraction by different products and advertisements in a supermarket, and use this data to direct their attention toward healthier products. For instance, our data analyses show that an increased target eccentricity decreases the distraction and makes the target search easier. Same principal also applies in marketing strategies. In [105], Wedel suggests that "a larger number of facings, and top and/or central placement of products on the shelves increases attention to and consideration of the brand [105]." Hence to encourage healthy products, one could place them in the upper and central parts of the shelves.

Another factor that facilitates the visual search is the bigger size of the target, which causes more visibility. Similarly, Wedel suggests that in print advertising and banners, the size of the ad has a positive effect on attention. Hence by choosing the neighbor products of a healthy product such that the healthy one is more differentiating (e.g. slighty bigger) and visible, we might increase the shoppers' chance of buying healthy products.

Another important component in encouraging healthy products is by directing customers' attention to the health warnings and nutrition labels on the products' package. For instance, Wedel discusses that information near the center of the label, which is typically dense, is less visible compared to the bottom or top of the label. Also, by using colors and pictures, health warnings could be designed more noticeable. [105]

Wedel also proposes that customers only use one or two features at a time to find their target brand. Thus, advertisements should create strong memory associations between a brand and a few unique features (e.g. color and shape) [105]. By advertising healthy products in an effective way we can further encourage their purchase.

3.3 Method 2: Predicting Segmentation of Targets and Distractors

In our second method, we aim to segment the distracting and target objects during visual search. Our method involves training a separate Mask-RCNN instance segmentation network for each target category. We also attempt to predict a distraction score for the distracting objects, which is defined as the probability of an object belonging to the distractor class. As in object-level saliency, we treat all pixels belonging to the same object as having similar distraction (saliency) score. This method is in accordance with object-based visual attention.

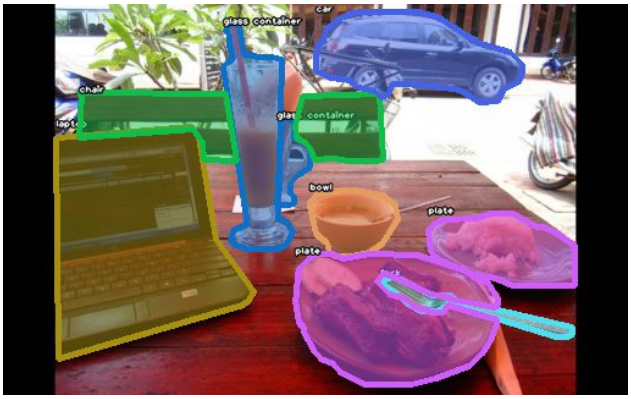
3.3.1 Data Segmentation

COCOSearch-18 dataset does not provide the segmentation of the objects that are present in the search images. Although all images in COCOSearch are taken from COCO dataset which provides partial segmentation of objects, most of these segmentations contain neither the target object nor the fixated distractors. Hence, we use COCO Annotator [14], an online annotation platform, to manually segment the fixated distractors and targets in COCOSearch-18. We annotated all fixated and several non-fixated objects (for statistical purposes) by all participants in each search image. In the subsequent steps of data processing, we investigate if the fixation locations lie on each of the segmented objects. We then use the fixated non-targets (distractors) along with the target objects as input to the Mask-RCNN network. So far, we have annotated 3 categories: bottle, bowl, and car, out of 18 categories. We aim to segment all the remaining categories and release our annotations publicly. We show some examples of our segmentations in figure 3.14.

The output of COCO Annotator is a json file similar to the format of COCO dataset. The json file is a collection of “images”, “annotations”, and “categories”, formatted as below:

(a)

(b)



(c)

Figure 3.14: Sample segmented images for 3 categories (a) bottle, (b) bowl, and (c) car. The segmentations are drawn using polygon tool in COCO Annotator [14].

```
{
“images”: [...],
“annotations”: [...],
“categories”: [...]
}
```

Table 3.3: The output json file of COCO Annotator.

The “images” section is composed of a list of annotated images in our dataset along with other information such as dimensions, object categories, number of annotations, etc. The annotation data such as segmentation or bounding boxes are not present in this part. The most important component of this section is the “id” of each image, which needs to be unique. This “id” is then used in the annotation section to determine the reference image of each annotation. The format of images section is shown in box 3.4.

“images”:[{

“id”:333,

“dataset_id”:2,

“category_ids”:[1,3,61,72,93],

“path”:“/datasets/coco_/bottle_000000001455.jpg”,

“width”:512,

“height”:320,

“file_name”:“bottle_000000001455.jpg”,

“annotated”:true,

“annotating”:[],

“num_annotations”:5,

“metadata”:,

“milliseconds”:1157036,

“events”:[“_cls”:“SessionEvent”,“user”:“Manoosh.samiei”,“milliseconds”:3462,“tools_used”:[],

...],

“regenerate_thumbnail”:false,

“is_modified”:false },

...]

Table 3.4: The “images” section of COCO Annotator output file.

The “annotations” section contains a list of each individual object annotation from all images in the dataset. Each annotation has an id, which is unique. The “image id” corresponds to a specific image in the dataset, which was specified as “id” in “images” section. The category id corresponds to a single category specified in the categories section. The “segmentation” contains a list of polygon vertices (x, y pixel positions) around the object that we segmented. The area bounded by the annotation of the object is measured in pixels. The rectangular bounding box has the format: [top left x position, top left y position, width, height]. Iscrowd specifies whether the

segmentation is for a single object or for a group of objects. As in most cases we only segment one object with a polygon, iscrowd is always false in our data. isbbox determines whether a bounding box has been specified. As we have only drawn polygons around the objects, isbbox argument is always false in our dataset. The specified bbox coordinates are therefore an approximation derived from the polygon vertices. The format of annotations section is shown in box 3.5.

“annotations”:[{
“id”:16,
“image_id”:333,
“category_id”:1,
“dataset_id”:2,
“segmentation”:[[274.7,36.6,279,30.5,279,6.3,287.5,3.2,292.3,3.2,300.2,5.7,305,9.3,305,30.5,
306.9,32.3, 308.7,49.9,303.8,57.2,304.4,80.2,289.3,82.1,276.5,80.8,277.2,60.2,274.7,56.6]],
“area”:2172,
“bbox”:[275,3,34,79],
“iscrowd”:false,
“isbbox”:false,
“creator”:”Manoosh.samiei”,
“width”:512,
“height”:320,
“color”:”#fa20db”,
“metadata”:{ } ,
“milliseconds”:19993,
“events”:[{ “_cls”:“SessionEvent”,“created_at”:“$date”:1619383275885,
“user”:“Manoosh.samiei”, “milliseconds”:19993,“tools_used”:[“Polygon”]}]},
...]

Table 3.5: The “annotations” section of COCO Annotator output file.

Finally, the “categories” section contains a list of 18 object categories (e.g. car) in COCOSearch18. Each of these categories can belong to a supercategory (e.g. vehicle); however, we have not specified any super categories. Each category has a unique id. For instance the id of ‘bottle’, ‘car’, and ‘bowl’ categories are 1, 5, and 4, respectively.

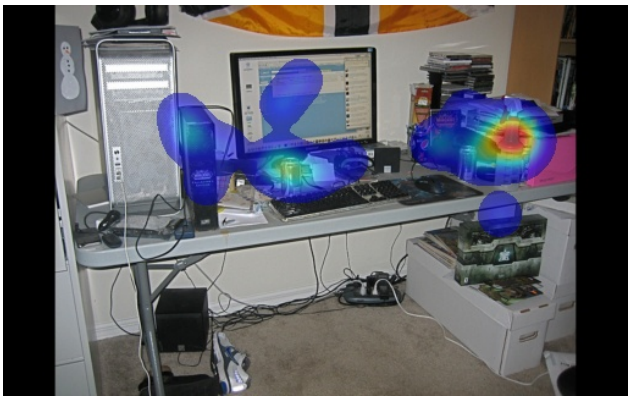
“categories”:[{
“id”:1,
“name”:“bottle”,
“supercategory”:“ ”,
“color”:“#008db2”,
“metadata”: { },
“creator”:“Manoosh.samiei”,
“keypoint_colors”:[]},
...]

Table 3.6: The “categories” section of COCO Annotator output file.

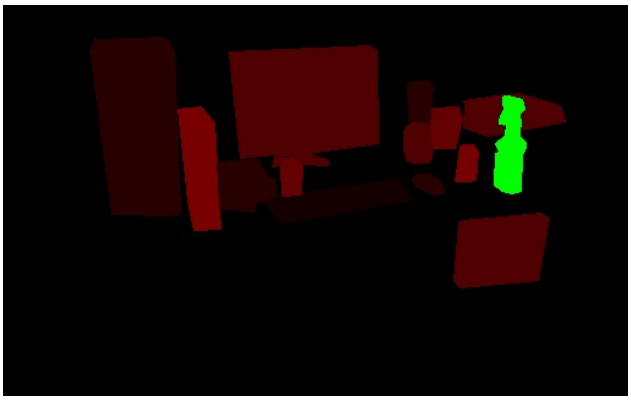
Note that for each search image we segmented the target, all distractors, and few random non-fixated objects (non-distractors). After we gathered these annotations for each target category, we process them to assign them a distraction level. For each segmented non-target object, we count the number of participants that made fixation on it. As we have 10 participants in total, the distraction level vary between 0 and 10. A score of 0 means no distraction (no fixation), and 10 means very high distraction. Figure 3.15 shows a segmented image mapped to RGB color space based on the distraction levels. Figure 3.16 shows the histogram of distraction levels for segmented fixated distractors in two targets categories ‘bottle’ and ‘car’.

As part of data processing pipeline, we save the non-target segmentations along with their distraction label and reference image name in a text file. For target objects’ segmentations, we use a label of ‘11’, and save them in the same text files as non-targets. We also split the search images

(a)

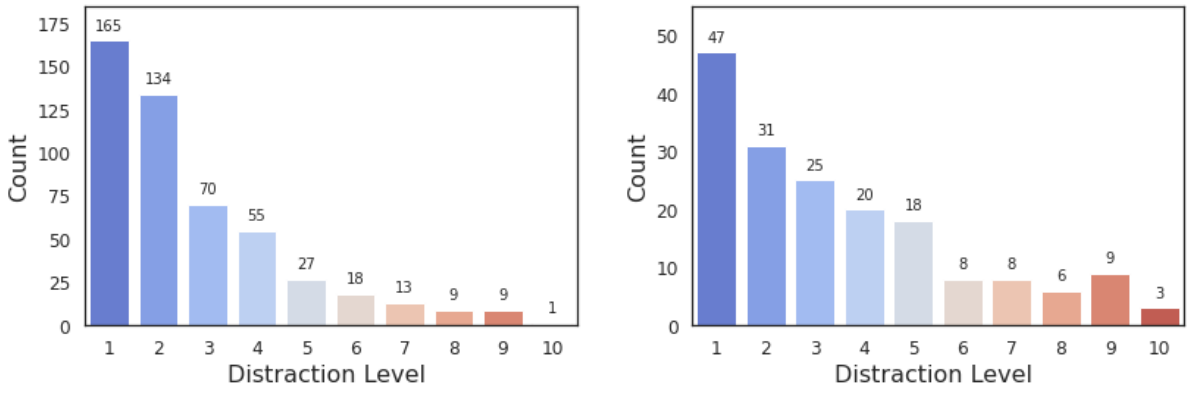


(b)



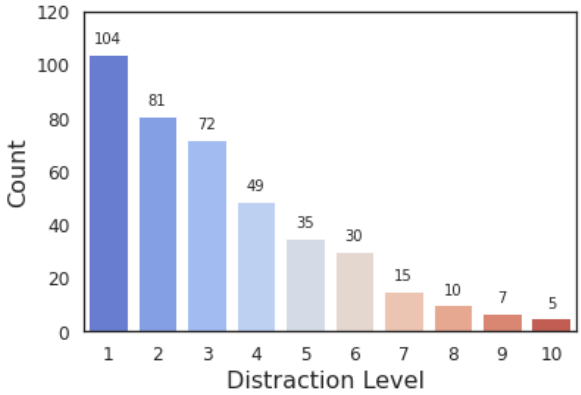
(c)

Figure 3.15: (a) Search image stimuli; (b) ground-truth fixation density map; (c) RGB-mapped object segmentation. Green indicates the target (bottle), and red objects are distractors. More intense red colors correspond to higher distraction levels.



(a) Bottle Category

(b) Car Category



(c) Bowl Category

Figure 3.16: The distribution of distraction levels in bottle, bowl, and car data. Distraction levels of 1 and 2 are the most prevalent, indicating that most distractors are fixated by one or two observers. A distraction level of more than 7 while searching for bottle and bowl, and a distraction level of more than 5 while searching for car is relatively rare.

into train, validation, and test parts. A sample trajectory from our generated text file is shown in box 3.7.

{“file_name”: “bottle_000000001455.jpg”,
“segmentation”: [[[274.7, 36.6], [279.0, 30.5], [279.0, 6.3], [287.5, 3.2], [292.3, 3.2],
[300.2, 5.7], [305.0, 9.3], [305.0, 30.5], [306.9, 32.3], [308.7, 49.9], [303.8, 57.2], [304.4, 80.2],
[289.3, 82.1], [276.5, 80.8], [277.2, 60.2], [274.7, 56.6]]],
“label”: 11}

Table 3.7: The label of this segmentation is 11, meaning that the segmented object is a target.

Then using our generated text file we create the ground truth instance segmentation of targets and fixated distractors (ignoring 0 labels), such that each object forms a separate channel in a one-hot encoded format as shown in figure 3.17. The first channel contains the target object and the other channels contain the distractors (The number of distractors vary for each image, but they are limited to 20 instances).

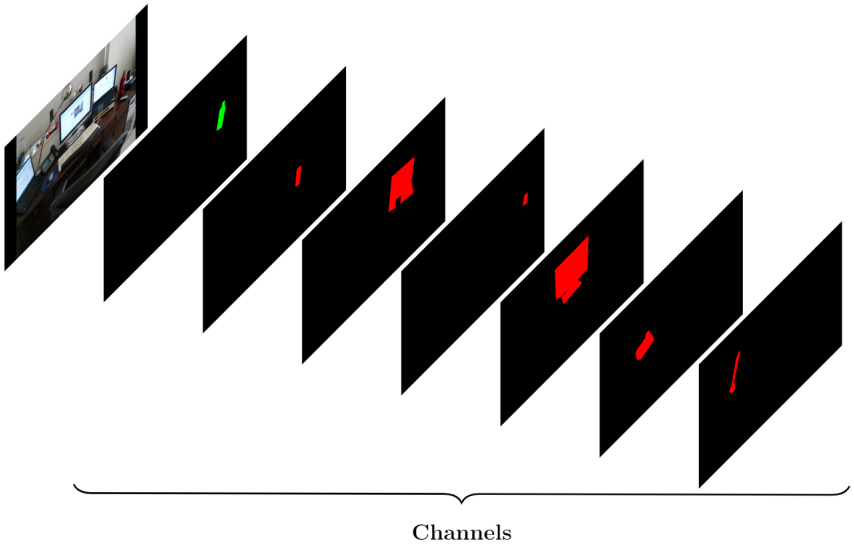


Figure 3.17: The RGB_mapped segmentation masks used as input to Mask-RCNN pipeline.

3.3.2 Mask-RCNN Architecture

We use Mask-RCNN [45] instance segmentation network in our modeling. Mask-RCNN belongs to the RCNN (Regions with CNN features) group of networks. The first RCNNs, namely RCNN [40], fast-RCNN [39], and faster RCNN [87], were solely for object detection purpose. Mask-RCNN on the other hand provides pixel-level segmentation, by adding a new branch to faster-RCNN along with some modifications. We first describe faster-RCNN architecture and then extend that to Mask-RCNN.

As the first step, faster-RCNN extracts multi-scale image features using FPN (Feature Pyramid Networks) [74] architecture. FPN consists of a bottom-up pathway, and a top-down architecture with lateral connections. The bottom-up pathway consists of popular CNN architectures such as ResNet or VGG, which extracts features from input images. Top-down pathway generates feature pyramid map, which has the same size as the bottom-up pathway. To create the pyramid, as we go down the top-down path, we upsample the previous layer by 2 using nearest neighbors upsampling. We apply a 1×1 convolution to the corresponding feature maps in the bottom-up pathway, and add it to the upsampled previous layer element-wise. A 3×3 convolution is then applied to all merged layers, which reduces the aliasing effect when merged with the upsampled layer. FPN outperforms single CNNs due to creating high-level semantic feature maps at various scales [49]. This process is shown in figure 3.18.

As the next step, the network proposes several boxes in the image and checks if any of them corresponds to an object. For this purpose, faster-RCNN has a fully convolutional network, known as region proposal network (RPN), on top of the extracted CNN features. This network passes a sliding window over the CNN feature map; at each window outputting k potential bounding boxes and the scores of how good each bounding box is. The k proposed bounding boxes are compared to k reference boxes, which are called anchors. Each anchor is centered at the sliding window in question, and has a scale and aspect ratio. At each sliding position, there are typically 9 anchors with 3 different scales and 3 aspect ratios. To generate features for each of these proposed bounding boxes, the pre-computed CNN features of the whole image are reused through a method

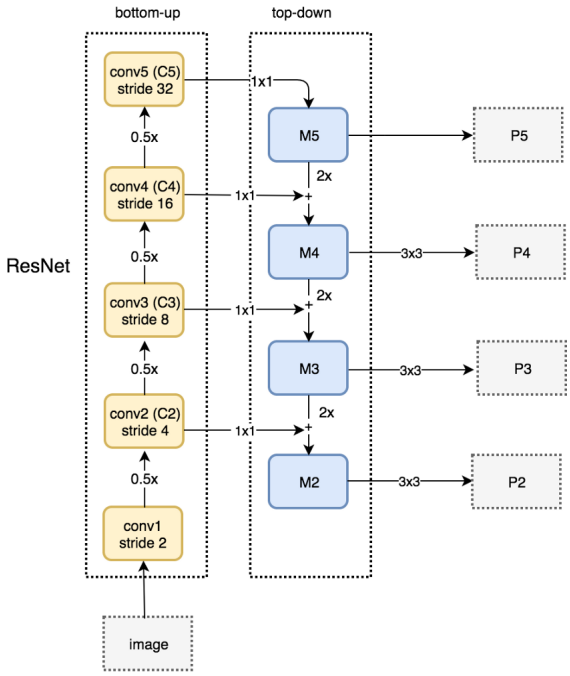


Figure 3.18: FPN Architecture. Image credit: [49]

called ROI pooling. Then for each bounding box, we pass its features into faster RCNN to generate a classification and run a linear regression to fit a tighter bounding box to the object.

To generate pixel-wise segmentation along with object classification and bounding box detection, authors introduced Mask-RCNN. In mask-RCNN, a fully convolutional neural network is added to the top of cnn features to generate a mask segmentation output. This is in parallel to the classification and bounding box regressor network of the faster RCNN. Also, authors realized that the regions of the feature map selected by ROI pooling were slightly misaligned from the regions of the original image. Since the image segmentation required pixel-level specificity unlike bounding boxes these misalignments led to inaccuracies. The authors solved this problem by introducing a method called ROI Align, which forces the cell boundaries of the target feature map to realign with the boundary of the input feature maps. The overall architecture of MASK-RCNN network is shown in figure 3.19.

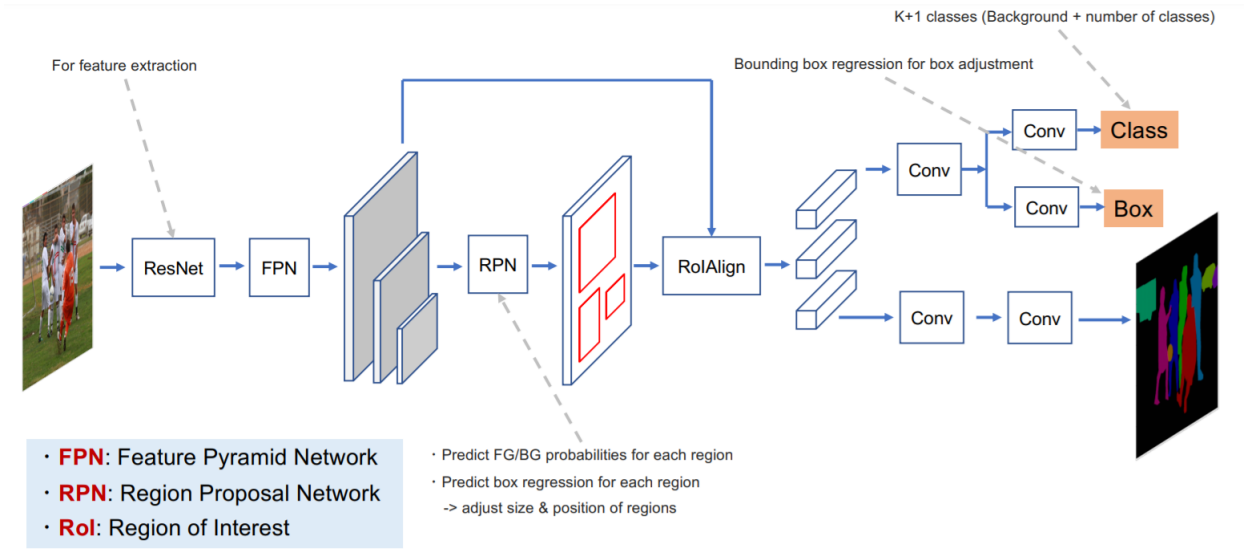


Figure 3.19: Mask-RCNN Architecture. Image credit: [25]

3.3.3 Technical Details and Training

We define 2 object classes 'target' and 'distractor' along with the default background class, which is defined implicitly in Mask-RCNN pipeline. The input segmentation masks shown in 3.17 are then reconfigured in MASK-RCNN pipeline based on their segmentation class, as shown in 3.20.

Figure 3.20: The segmentation masks for each image are categorized into 'target' and 'distractor' classes in Mask-RCNN pipeline.

As the backbone feature extractor we use ResNet101 with strides 4, 8, 16, 32, and 64. We use a batch size of 1 for our training and validation. The network is able to detect maximum 100 object instances. The input images are 320 × 512, and are transformed to square shape of 512 × 512 dimension. The learning rate, momentum, and weight decay are set to 0.001, 0.9, and 0.0001 respectively. There are 5 different losses that are jointly trained in an end-to-end fashion. These losses are: RPN classification loss, RPN bounding box regression loss, Mask-RCNN classification loss, Mask-RCNN bounding box regression loss, and Mask-RCNN instance segmentation loss. All of these weights have equal weights(importance) during training, and are calculated for both training and validation sets in each epoch. During training the model tries to minimize the summation of all these losses, as seen in equation 3.2.

$$\begin{aligned} Loss = {} & rpn\_class\_loss + rpn\_bbox\_loss \\ & + mrcnn\_class\_loss + mrcnn\_bbox\_loss \\ & + mrcnn\_mask\_loss \end{aligned} \tag{3.2}$$

We initialize MASK-RCNN with pre-trained weights on MS COCO dataset. Then we freeze all the layers, except the head branches. The head branches are composed of the RPN, classifier and mask heads of the network. We train these head layers for 20 epochs. During training we save the weights of the model with the least validation loss.

3.3.4 Experiments

Table 3.8 shows the mean and standard deviation of minimum validation losses obtained for each target-specific model averaged over 5 independent runs.

We also calculate the accuracy of target and distractor detection by comparing the predictions with the ground-truth data. We count the number of correctly classified distractors and targets, and then divide these numbers by the total number of targets and distractors in each image, to obtain an accuracy for each image. Then to obtain an accuracy for each run, we compute the average

accuracy over all images. Finally, we take the average of accuracy over 3 independent runs. We repeat these steps for each target category, and report the results in table 3.9.

Model		Added Loss	RPN Class Loss	RPN Bbox Loss	Mrcnn Class Loss	Mrcnn Bbox Loss	Mrcnn Mask Loss
Car	μ	0.82	0.01	0.38	0.02	0.20	0.22
	σ	0.032	0.003	0.044	0.002	0.020	0.015
Bottle	μ	1.11	0.02	0.51	0.05	0.27	0.26
	σ	0.041	0.002	0.021	0.006	0.011	0.024
Bowl	μ	0.85	0.02	0.35	0.04	0.21	0.24
	σ	0.025	0.005	0.023	0.006	0.012	0.024

Table 3.8: The mean and standard deviation of the least validation losses obtained for models of each target category, over 5 independent runs.

Model		Target Detection Accuracy	Distractor Detection Accuracy
Car	μ	0.81	0.69
	σ	0.037	0.089
Bottle	μ	0.75	0.552
	σ	0.028	0.101
Bowl	μ	0.72	0.83
	σ	0.214	0.098

Table 3.9: The mean and standard deviation of target and distractor detection accuracy computed over 3 independent runs, for each target-specific model.

Among the three categories, 'Car' has the highest target detection accuracy, and 'Bowl' has the highest distractor detection accuracy. The mean accuracy of 'Bowl' category has a relatively high standard deviation, indicating that the accuracy of different runs can vary depending on the difficulty of test images.

Figures 3.21, 3.22, 3.24, 3.25, 3.27 and 3.28, show sample predictions of MaskRCNN for 'target' and 'distractor' categories along with their ground-truth segmentations. Figures 3.23, 3.26, and 3.29 show some unsuccessful predictions of the network, where it fails to detect the target due to occlusion, blurriness, or a very small size. The network also outputs a confidence score/probability for each classified object. We define each distractor's probability as an indication

of how distracting that object is or the 'distraction score' of that object. In the generation of figures 3.21 to 3.26, we have defined various thresholds on the distraction scores to only output the top 3 or 4 predicted distractors in each image. Without defining these thresholds, the network shows all of the predicted distractors even those with low distraction scores, that do not even contain a real object. For most images a distraction threshold of 90-95% removes most false predictions.

Our network segments and classifies most foreground objects that do not belong to the target category as distractors. However, we believe that the distraction scores are related to the object class of the distractor, similarity to the target, proximity to the target, proximity to the center of the image, and distractor's size. Hence defining the threshold on this distraction scores could be a way to give us the most distracting objects.

We discussed earlier that we labeled each segmented object with its distraction level, i.e. number of observers who fixated at that non-target object. To test if a MASK-RCNN can distinguish between distractors with different distraction levels, We trained a Mask-RCNN with 3 instance categories, corresponding to 'target', 'low-distractor', and 'high-distractor'. Low-distractors are non-target objects fixated by 1 or 2 people, while high-distractors are those fixated by 3 or more observers. Figure 3.30 shows some example results of the network in this configuration. It can be seen that the network does not always succeed to correctly determine the highly distracting objects and the low distracting ones.

We also tested fine-tuning all Mask-RCNN layers on COCOSearch18, instead of only the head layers. This modification led to less accuracy in detecting the 'target' category in some test images. We hypothesise that the features that are extracted for object-class instance segmentation on COCO dataset are useful for target-distractor segmentation on COCOSearch18. Thus freezing those layers and only training the heads can enhance target classification.

We tried augmenting the training dataset with horizontal flip. However, it did not have a significant impact on the performance.

As the last experiment, we compared the segmentation results of our fine-tuned Mask-RCNN with an untrained version, that is only pre-trained on COCO dataset. Figure 3.31 shows that our training changes the predicted segmentations by Mask-RCNN, in favor of target-distractor

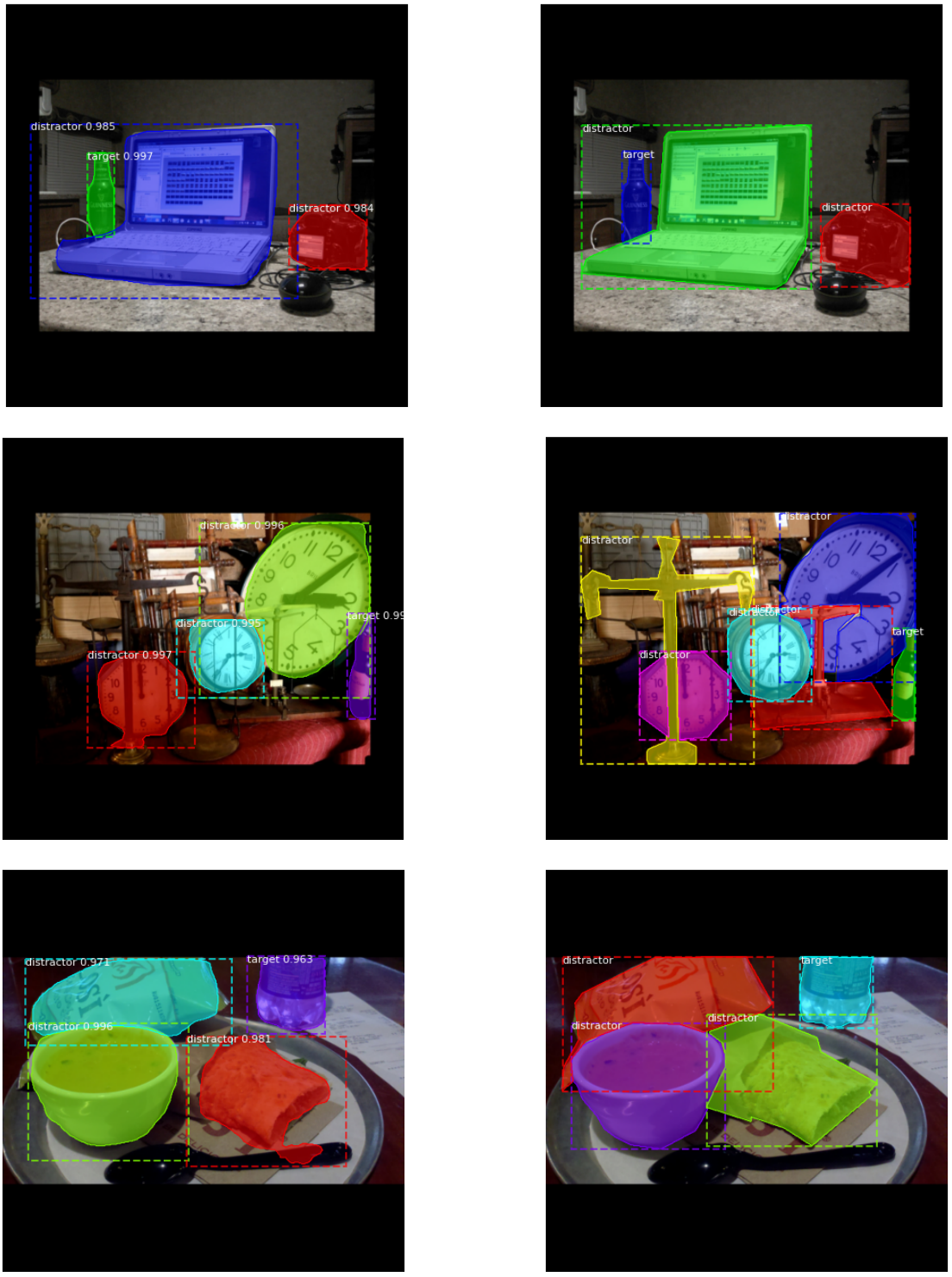


Figure 3.21: Sample results of network predictions on unseen test data for 'Bottle' target category. The left column contains the predictions. The right column contains the ground-truth segmentations.

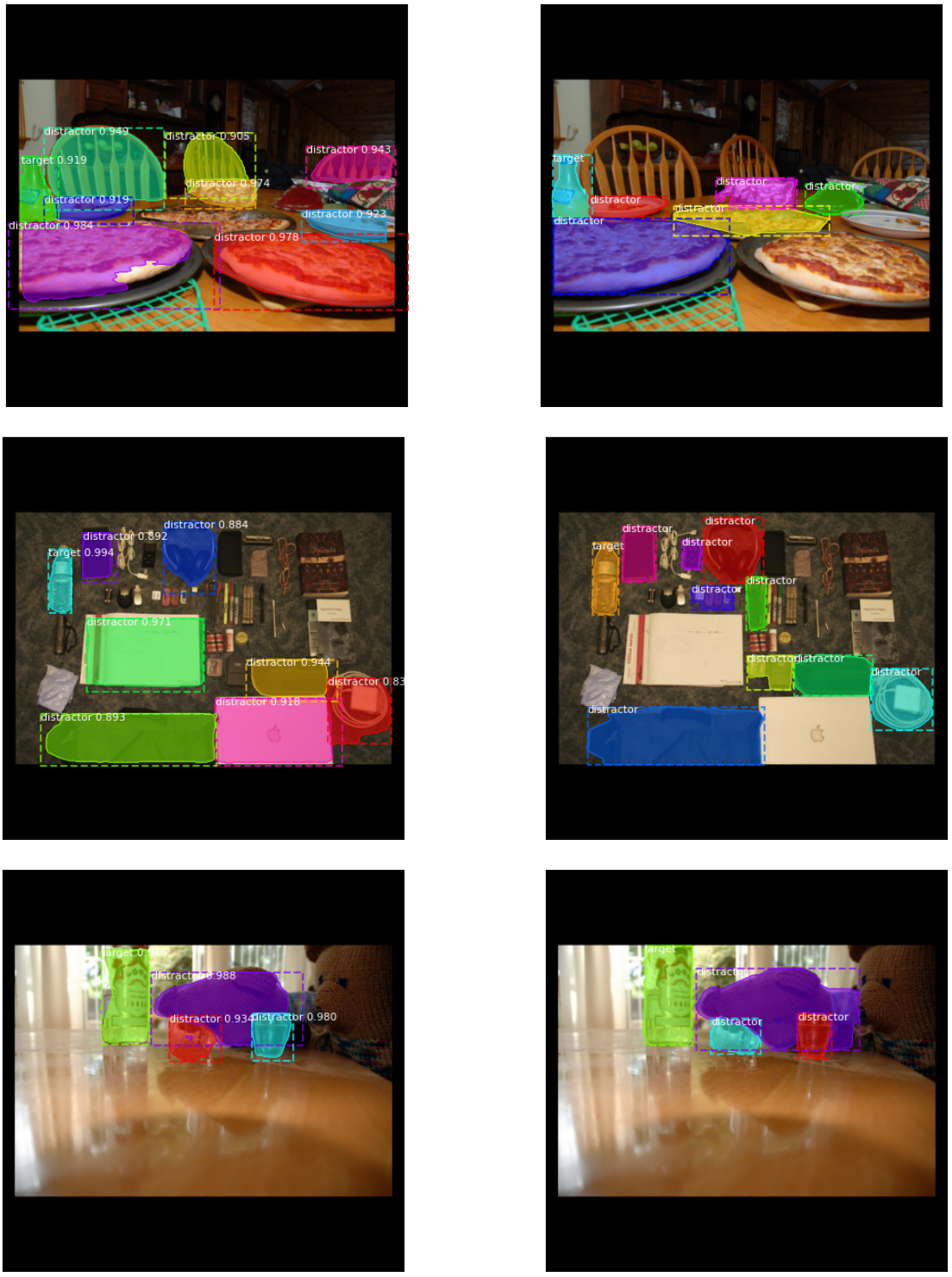


Figure 3.22: Sample results of network predictions on unseen test data for 'Bottle' target category. The left column contains the predictions. The right column contains the ground-truth segmentations.

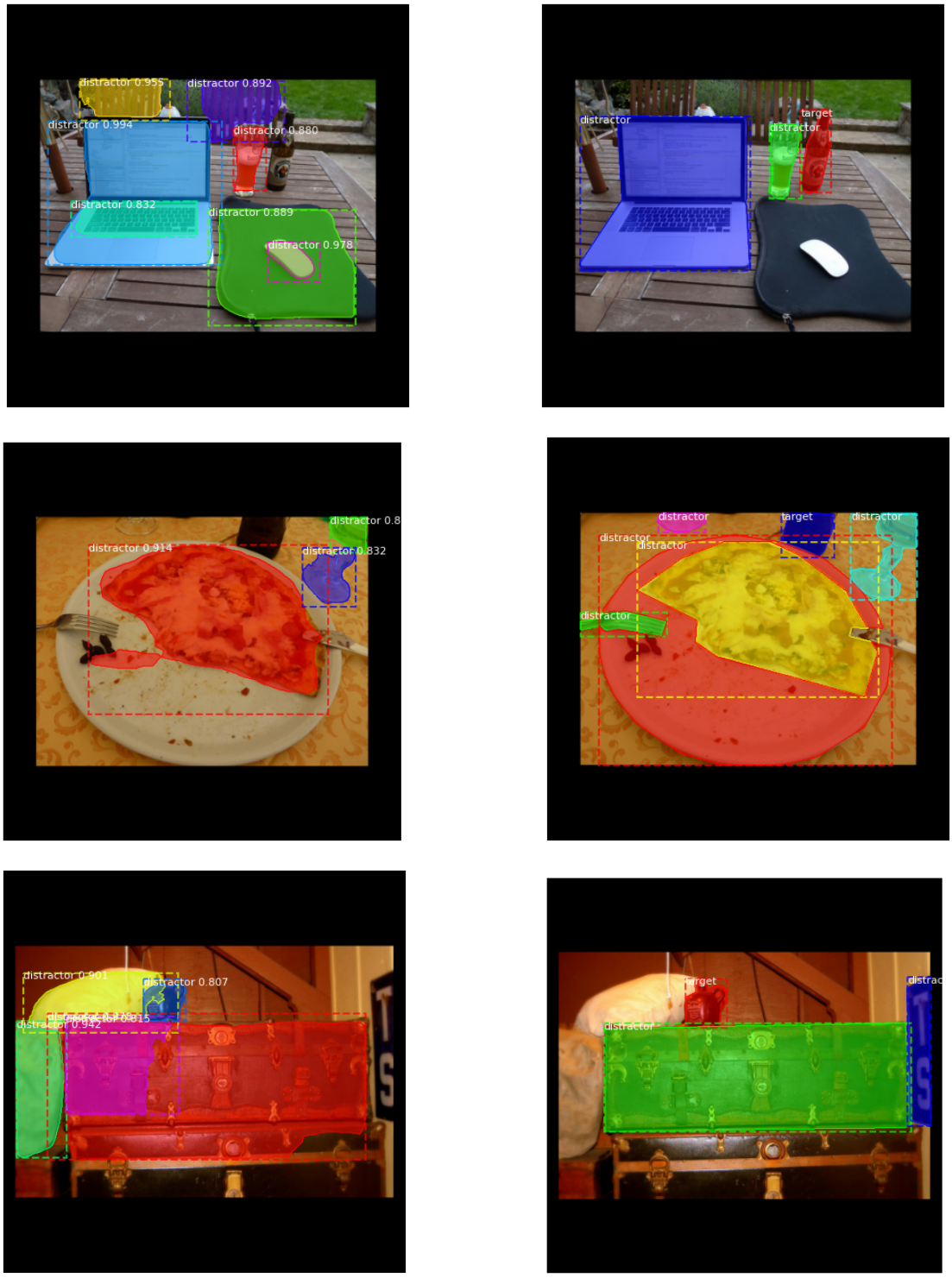


Figure 3.23: Sample failures of network predictions, where the the network fails to detect the target, in unseen test data for 'Bottle' target category. The left column contains the predictions. The right column contains the ground-truth segmentations.

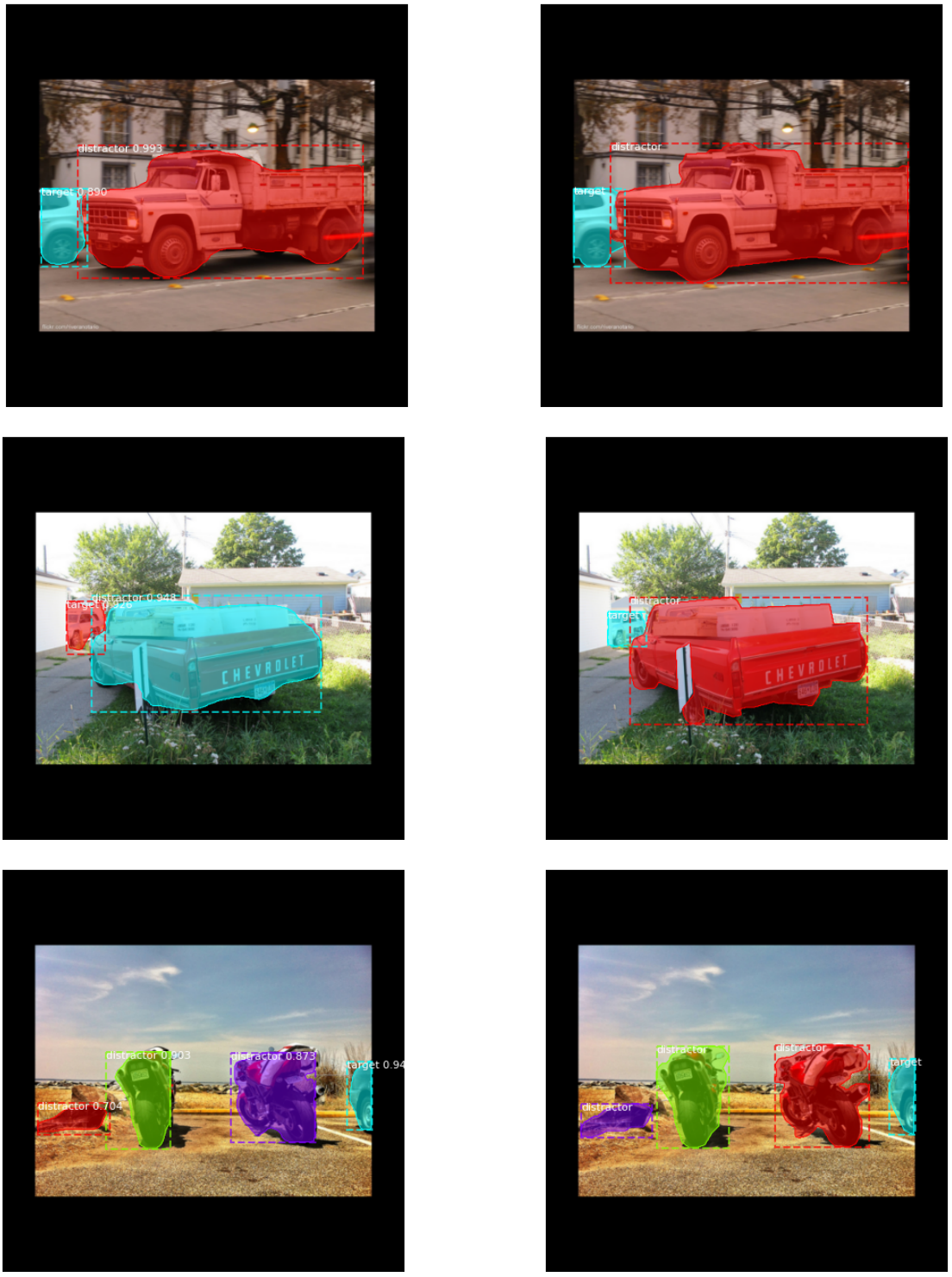


Figure 3.24: Sample results of network predictions on unseen test data for 'Car' target category. The left column contains the predictions. The right column contains the ground-truth segmentations.

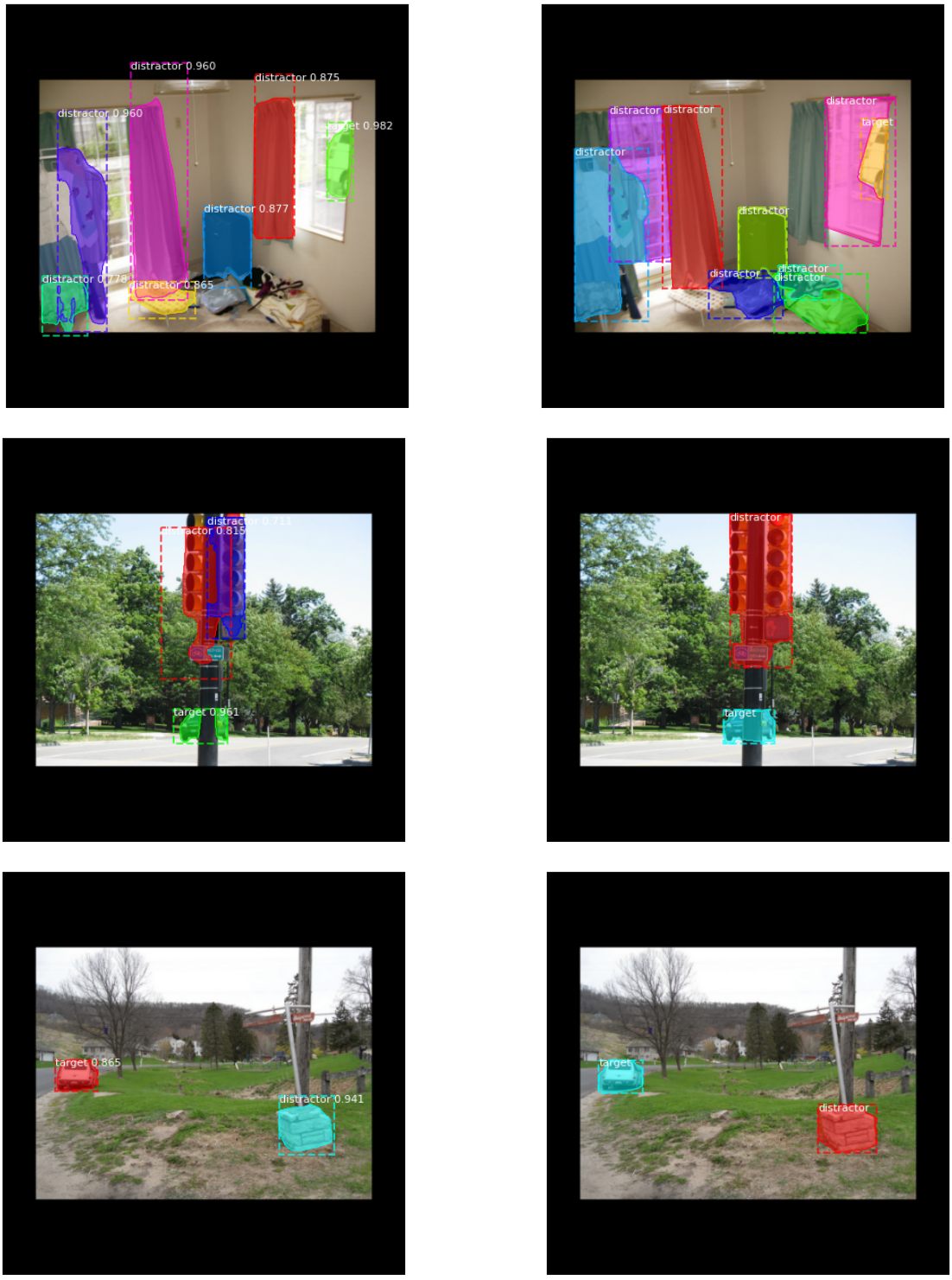


Figure 3.25: Sample results of network predictions on unseen test data for 'Car' target category. The left column contains the predictions. The right column contains the ground-truth segmentations.

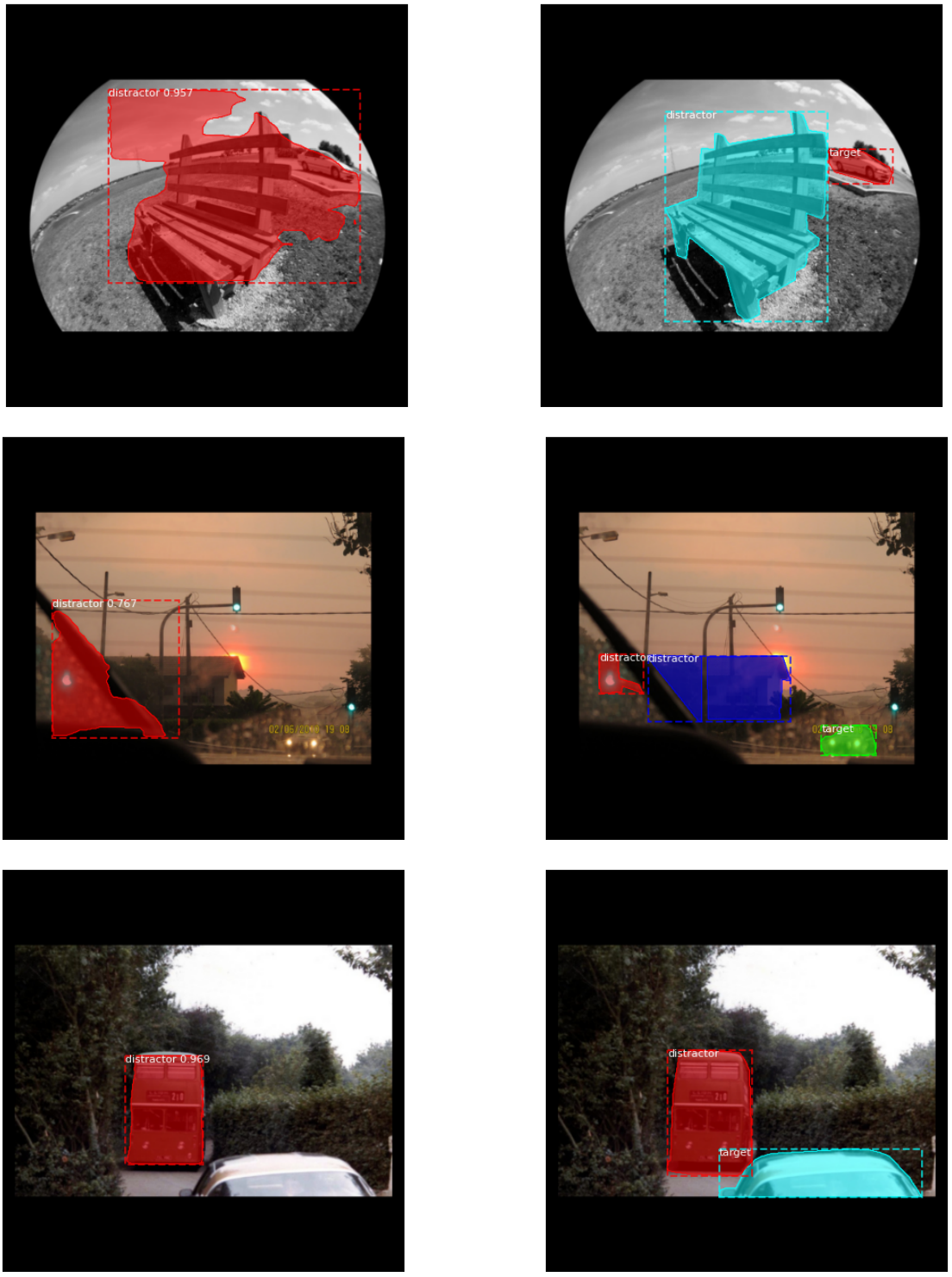


Figure 3.26: Sample failures of network predictions, where the the network fails to detect the target, in unseen test data for 'Car' target category. The left column contains the predictions. The right column contains the ground-truth segmentations.

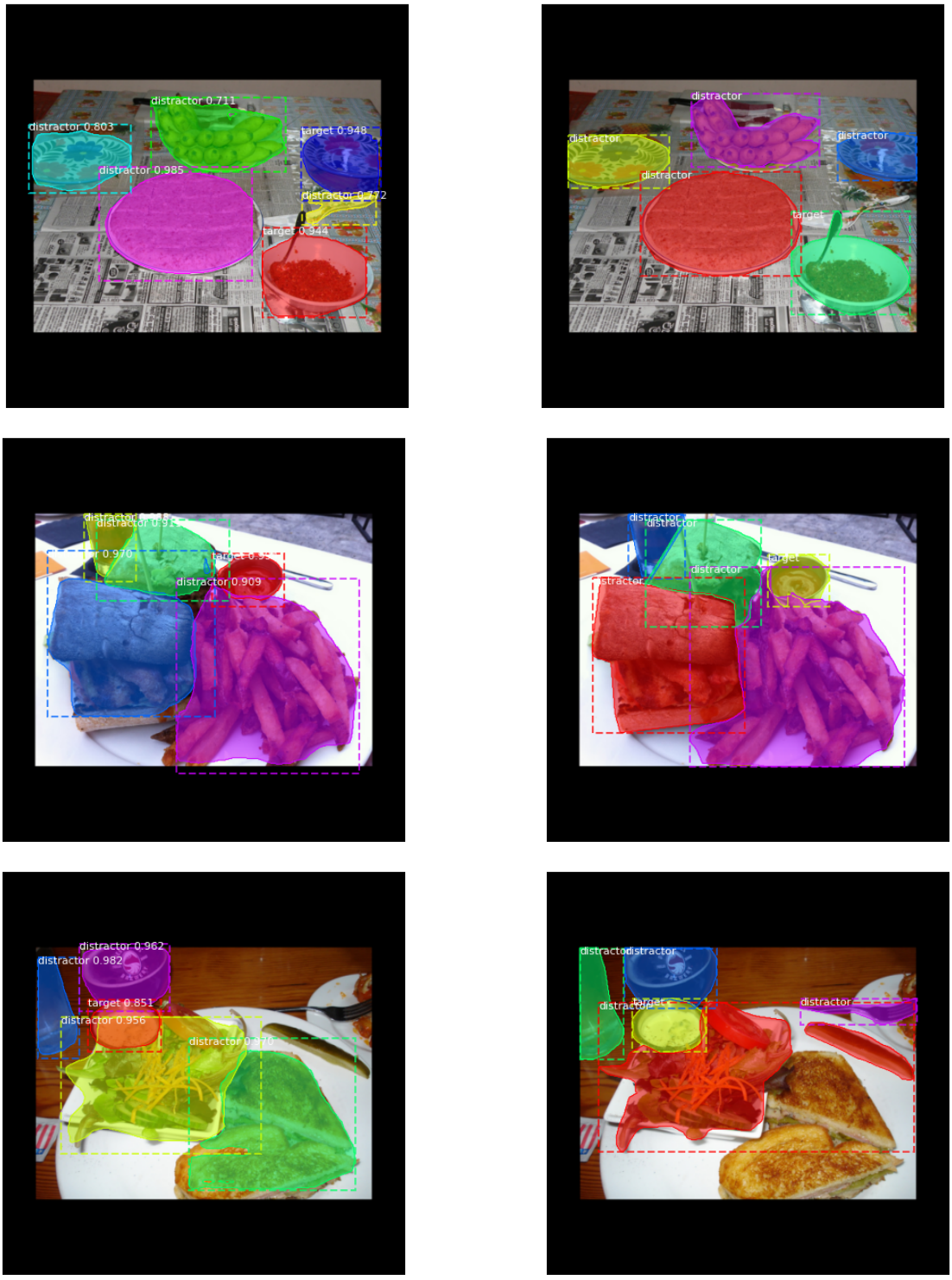


Figure 3.27: Sample results of network predictions on unseen test data for 'Bowl' target category. The left column contains the predictions. The right column contains the ground-truth segmentations.

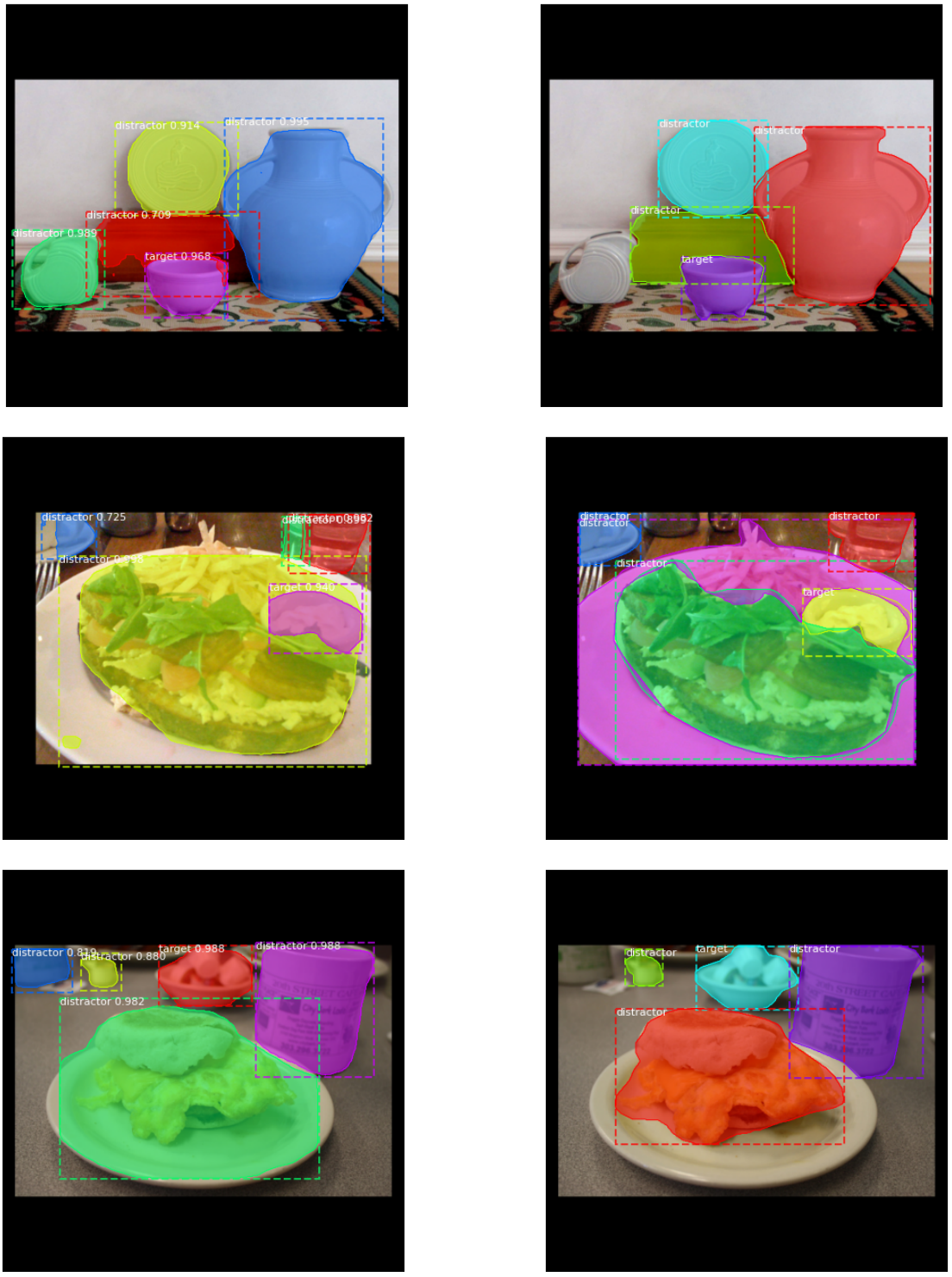


Figure 3.28: Sample results of network predictions on unseen test data for 'Bowl' target category. The left column contains the predictions. The right column contains the ground-truth segmentations.

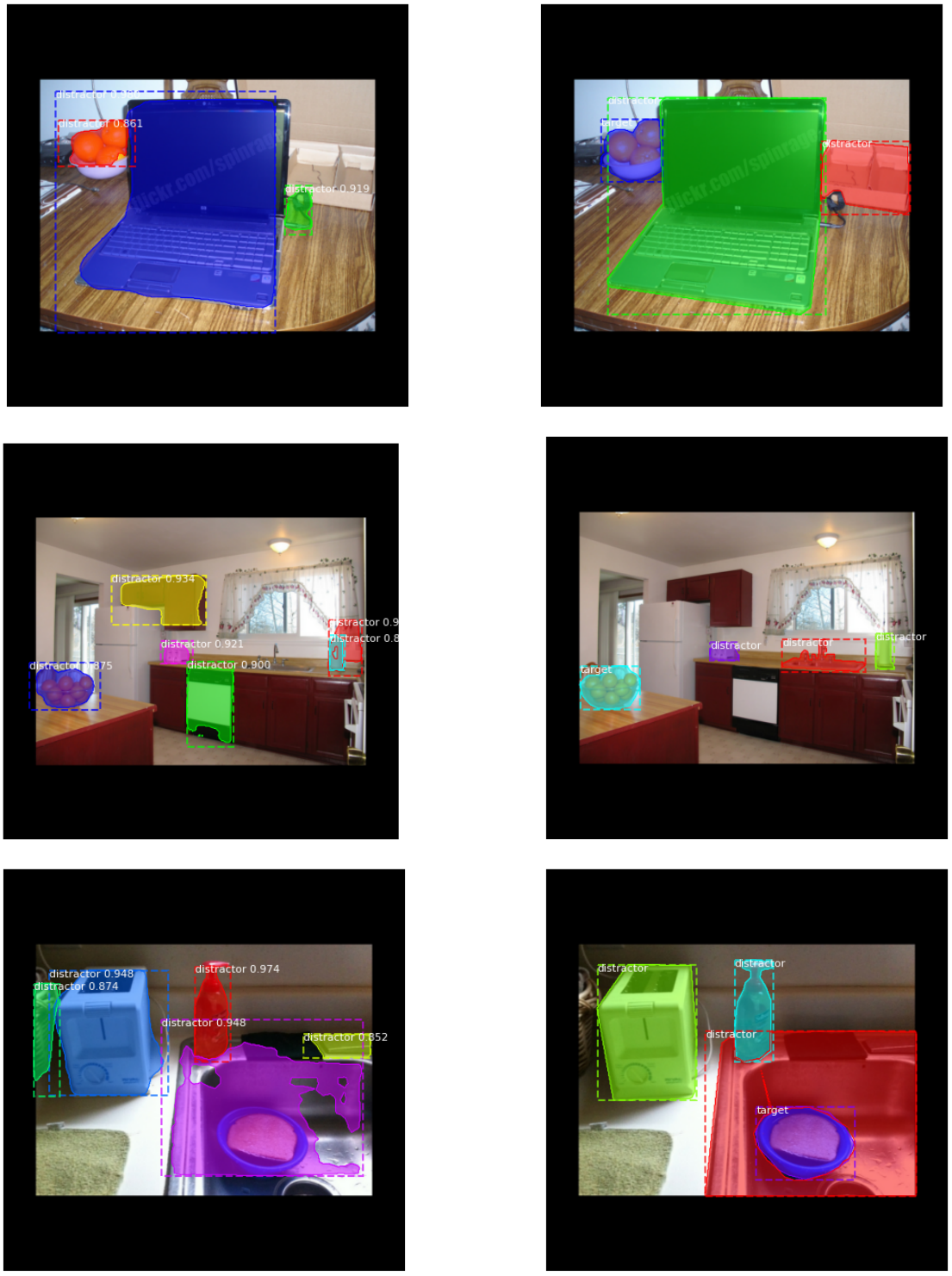


Figure 3.29: Sample failures of network predictions, where the the network fails to detect the target, in unseen test data for 'Bowl' target category. The left column contains the predictions. The right column contains the ground-truth segmentations.

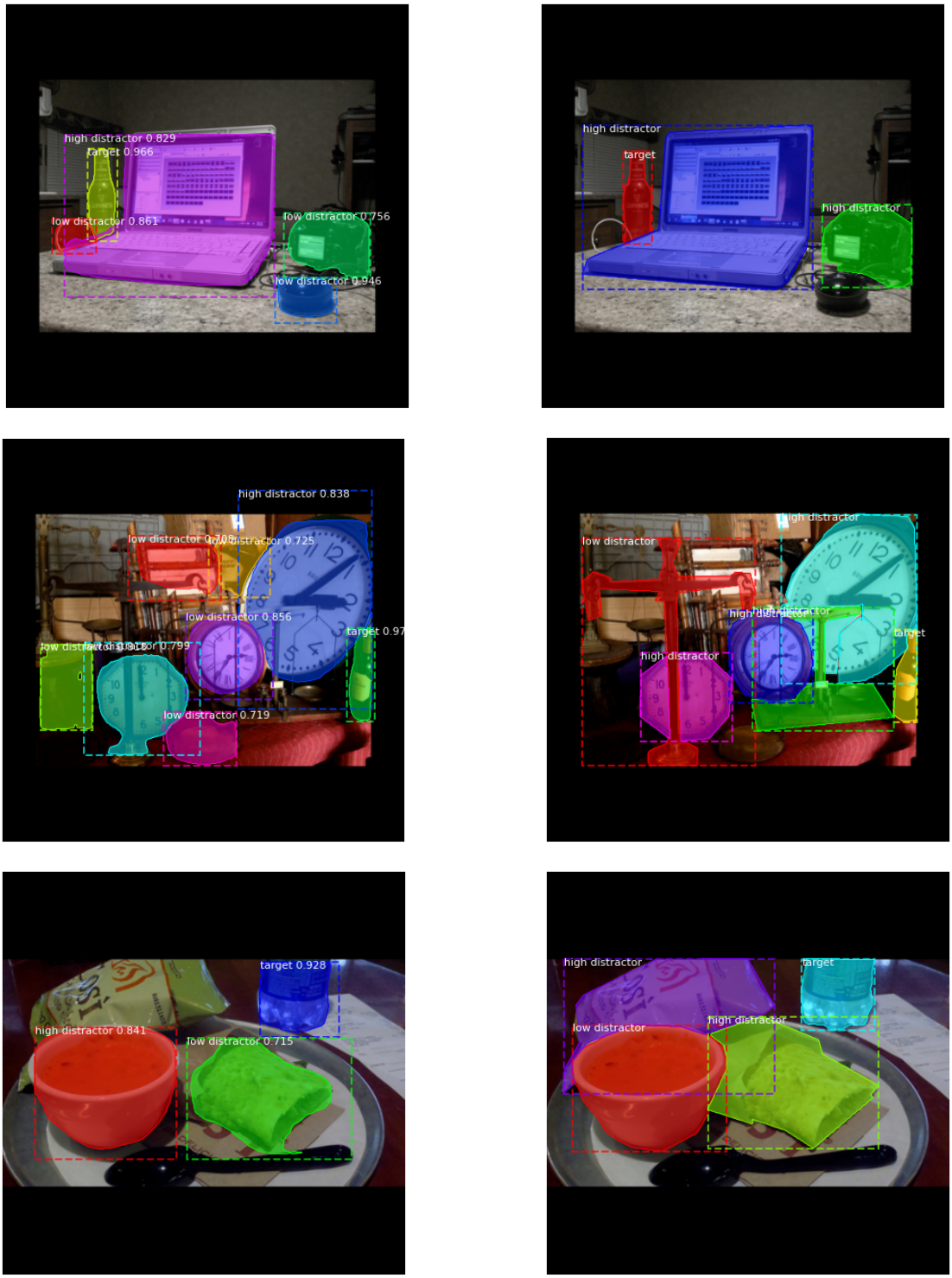


Figure 3.30: Sample results of Mask-RCNN with 3 categories: target, low-distractor, and high-distractor. Left column are predictions and right column are ground-truths.

Figure 3.31: The left column shows the predictions of a Mask-RCNN pretrained on COCO dataset. The right column shows the results of a Mask-RCNN pretrained on COCO and fine-tuned on COCOSearch18 for target-distractor detection.

detection. Also for some images, it removes the false-object segmentations, as seen in the top left image in figure 3.31, where a backpack is incorrectly segmented on top of the motorcycle.

3.3.5 Discussion

The method we discussed here is a good way of segmenting target and distracting objects in images. The benefit of our method is its quick training and inference.

One drawback of our method is that it needs to train a separate Mask-RCNN for each target category. One possible future direction is to unify these category-specific models into one model.

Another shortcoming of our method is that we only consider an object as fixated if the fixation location exactly lies on its segmentation. However, it might be the case that multiple objects are fixated by each fixation. This would need us to consider a radius (similar to the high-resolution fovea image which typically covers a region of 1.5° of the visual field) around each fixation location and consider all objects whose segmentation partially lies under this radius as fixated.

Furthermore, object-based visual attention has the assumption that all parts of an object are equally salient, or in our case equally distracting. This assumption often does not hold true. For instance, in one of the search images there is a big monitor displaying images that attract most observers attention. While these images are only placed on the monitor's screen, we consider the whole monitor as highly distracting.

The limited number of participants (10) in our dataset could be another shortcoming of our approach. We hypothesise that with more participants we can better distinguish the level of distraction for different objects, as the individual preferences might average out.

3.4 General Discussion

In general, visual search is similar to the object detection problem in computer vision. In fact in both of our methods, we employ pre-trained networks for object-detection task to predict visual search behavior of humans. State-of-the-art object detectors such as Mask-RCNN (and faster-RCNN) have object proposal networks that suggest the most probable regions that could contain an object. Similarly, when searching for an object, humans fixate at locations that are likely to contain the target object. The contextual information/gist (which is assumed to be computed pre-attentively), and prior knowledge about where an object should be located affect these fixated locations.

A difference between a visual search model and RCNN object detectors, besides using eye tracking data, is that it considers searching for one target object at a time (similar to how COCOSearch-18 dataset was collected); while, an RCNN aims at detecting multiple object types and possibly more than one instance for each type.

In terms of internal architecture, a CNN-based object detector and visual attention theories share some similarities. For instance, CNN uses kernels (filters) each corresponding to a specific feature. Each filter attempts to find the locations in the image where the activation of that feature is maximized. Then during network training, the best filters (features) for detecting a specific target object are chosen. Feature integration theory also suggests that in conjunction search, humans

scan the scene in a serial way, merging different features of each fixated location until finding the target. Feature integration theory proposes that human brain possesses feature-specific neurons that activate in locations where those features are present. However, unlike a CNN where features are calculated across the whole image pixels; humans only perceive features around the fixated location in an image. This is due to the fact that humans' eye receive high-resolution image in a small region within the fovea; hence, eye movements are needed to perceive different parts of the scene with high resolution. The other reason is the computational limitation of human brain, which makes it impossible to perceive the whole scene with high-resolution.

In this thesis, we introduced two deep-learning based approaches to model human visual attention and distraction behavior during visual search tasks. Although our models act very similar to object detectors in computer vision, the main purpose of our work is not only detecting the target but also the distractors, to explore how human observers search an image for a particular target.

Chapter 4

Conclusion

In general, both of our methods provide evidence that distracting regions or objects in images during visual search are predictable. Our first model's predicted fixation maps show that the model learns where to look for each target category. For instance, when looking for oven or toilet, the locations closer to the ground were considered salient; whereas for TV or clock the higher items in images were often considered salient. Moreover, the similarity of the items to the target led to higher distraction. The items sharing some features with the objects of target category were sometimes considered salient both by the network and human observers.

Our second method involves training separate Mask-RCNN networks for each target category to segment and classify the target and distracting objects. This network outputs a confidence score for each classified object, which we used as a distraction score for the classified distractors. By defining the right threshold for each image, we were able to detect the most distracting objects existing in that image. We interpreted that these distraction scores are related to the size of the distractor, object class of the distractor, similarity to the target, proximity to the target, and proximity to the center of the image.

Our methods could be extended to 3D environments and be applied to supermarkets for analyzing customers behavior. Learning how customers are distracted to different products can help us to guide customers attention toward more healthy products. For application to a supermarket, one might extend our approach to predicting visual behavior during visual foraging tasks, where

multiple target categories are being searched at the same time. This would require collecting a visual foraging dataset similar to COCOSearch-18 (which currently does not exist).

One limitation of modeling task-oriented visual attention is the scarcity of fixation-labeled datasets. This problem might be due to the difficulty associated with collecting eye tracking data, especially with an added visual task. COCOSearch18 is the first dataset that provides fixation labels for searching a relatively large number of target categories (18). However, COCOSearch18 is viewed by only 10 observers. We believe that for a more reliable modeling of distraction, a greater number of observers are needed, so that the individual preferences are less prominent in the modeling. Currently, our model is trained and tested to predict fixation density maps of these 10 observers, who might not be a good representative of a larger population.

Another possible direction for future improvement could be to merge our first and second method, such that the predictions of the fixations density maps are transformed into object-based segmentation. There exists multiple methods that propose object segmentation based on fixation points such as [77], which we did not explore due to time constraint.

All in all, both of our proposed methods are good starting points in modeling and understanding human's visual search behavior. Further research is needed to validate our results, and develop stronger predictors for humans' distraction during visual search.

Bibliography

[1] Apa dictionary of psychology. https://psychologydictionary.org/distractor/, 2020.

[2] ABADI, M., AGARWAL, A., BARHAM, P., BREVDO, E., CHEN, Z., CITRO, C., CORRADO, G. S., DAVIS, A., DEAN, J., DEVIN, M., GHEMAWAT, S., GOODFELLOW, I., HARP, A., IRVING, G., ISARD, M., JIA, Y., JOZEFOWICZ, R., KAISER, L., KUDLUR, M., LEVENBERG, J., MANÉ, D., MONGA, R., MOORE, S., MURRAY, D., OLAH, C., SCHUSTER, M., SHLENS, J., STEINER, B., SUTSKEVER, I., TALWAR, K., TUCKER, P., VANHOUCKE, V., VASUDEVAN, V., VIÉGAS, F., VINYALS, O., WARDEN, P., WATTENBERG, M., WICKE, M., YU, Y., AND ZHENG, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[3] ALLETTO, S., PALAZZI, A., SOLERA, F., CALDERARA, S., AND CUCCHIARA, R. Dr(eye)ve: A dataset for attention-based tasks with applications to autonomous and assisted driving. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2016), pp. 54–60.

[4] ASHCRAFT, M. H., AND RADVANSKY, G. A. *Cognition*, 6th ed. Boston: Pearson Education, 2016.

[5] AWH, E., BELOPOLSKY, A., AND THEEUWES, J. Top-down versus bottom-up attentional control: A failed theoretical dichotomy. *Trends in cognitive sciences 16* (07 2012), 437–43.

[6] BAILLY, G., OULASVIRTA, A., BRUMBY, D. P., AND HOWES, A. Model of visual search and selection time in linear menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2014), CHI '14, Association for Computing Machinery, p. 3865–3874.

[7] BERGSTRA, J., YAMINS, D., AND COX, D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *ICML* (2013).

[8] BOISVERT, J. F., AND BRUCE, N. D. Predicting task from eye movements: On the importance of spatial distribution, dynamics, and image features. *Neurocomputing 207* (2016), 653–668.

[9] BORJI, A., AND ITTI, L. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 1 (2013), 185–207.

[10] BORJI, A., AND ITTI, L. Defending Yarbus: Eye movements reveal observers' task. *Journal of Vision 14*, 3 (03 2014), 29–29.

[11] BORJI, A., SIHITE, D. N., AND ITTI, L. Probabilistic learning of task-specific visual attention. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012), pp. 470–477.

[12] BORJI, A., SIHITE, D. N., AND ITTI, L. Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Transactions on Image Processing 22*, 1 (2013), 55–69.

[13] BORN, S., AND KERZEL, D. Congruency effects in the remote distractor paradigm: Evidence for top-down modulation. *Journal of Vision 9*, 9 (08 2009), 3–3.

[14] BROOKS, J. COCO Annotator. `https://github.com/jsbroks/coco-annotator/`, 2019.

[15] BRUCE, N. D. B., AND TSOTSOS, J. K. Saliency based on information maximization. In *Proceedings of the 18th International Conference on Neural Information Processing Systems* (Cambridge, MA, USA, 2005), NIPS'05, MIT Press, p. 155–162.

[16] BYLINSKII, Z., JUDD, T., OLIVA, A., TORRALBA, A., AND DURAND, F. What do different evaluation metrics tell us about saliency models? *IEEE Transactions on Pattern Analysis and Machine Intelligence 41*, 3 (2019), 740–757.

[17] CHE, Z., BORJI, A., ZHAI, G., MIN, X., GUO, G., AND LE CALLET, P. How is gaze influenced by image transformations? dataset and model. *IEEE Transactions on Image Processing 29* (2020), 2287–2300.

[18] CHEN, L.-C., PAPANDREOU, G., KOKKINOS, I., MURPHY, K., AND YUILLE, A. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence PP* (06 2016).

[19] CHEN, S., JIANG, M., YANG, J., AND ZHAO, Q. Air: Attention with reasoning capability. *arXiv* (2020).

[20] CHEN, X., BAILLY, G., BRUMBY, D. P., OULASVIRTA, A., AND HOWES, A. The emergence of interactive behavior: A model of rational menu search. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2015), CHI '15, Association for Computing Machinery, p. 4217–4226.

[21] CHEN, Y., YANG, Z., AHN, S., SAMARAS, D., HOAI, M., AND ZELINSKY, G. Cocosearch18 fixation dataset for predicting goal-directed attention control. *Scientific Reports 11* (04 2021).

[22] COCKBURN, A., GUTWIN, C., AND GREENBERG, S. A predictive model of menu performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2007), CHI '07, Association for Computing Machinery, p. 627–636.

[23] COMPANY, T. Types of eye movements. https://www.tobiipro.com/learn-and-support/learn/eye-tracking-essentials/types-of-eye-movements/.

[24] CONNOR, C., EGETH, H., AND YANTIS, S. Visual attention: Bottom-up versus top-down. *Current Biology 14*, 19 (Oct. 2004), R850–R852.

[25] DAT, N. P. T. Mask-rcnn for instance segmentation. https://speakerdeck.com/nptdat/mask-rcnn-for-instance-segmentation, 2019.

[26] DAWKINS, M. Shifts of 'attention' in chicks during feeding. *Animal Behaviour 19*, 3 (1971), 575–582.

[27] DEUTSCH, J. A., AND DEUTSCH, D. Attention: Some theoretical considerations. *Psychological Review 70*, 1 (1963), 80–90.

[28] DUNCAN, J., AND HUMPHREYS, G. Visual search and stimulus similarity. *Psychological review 96 3* (1989), 433–58.

[29] EHINGER, K. A., HIDALGO-SOTELO, B., TORRALBA, A., AND OLIVA, A. Modelling search for people in 900 scenes: A combined source model of eye guidance. *Visual Cognition 17* (2009), 945 – 978.

[30] ELTITI, S., WALLACE, D., AND FOX, E. Selective target processing: Perceptual load or distractor salience? *Perception and psychophysics 67* (08 2005), 876–85.

[31] ESSEN, D. C. V., FELLEMAN, D. J., DEYOE, E. A., OLAVARRIA, J., AND KNIERIM, J. Modular and hierarchical organization of extrastriate visual cortex in the macaque monkey. In *Cold Spring Harbor Symposia on Quantitative Biology, Vol. LV* (Cold Spring Harbor, NY, 1990), Cold Spring Harbor Laboratory, pp. 679–696.

[32] FELDMAN, J., AND BALLARD, D. Connectionist models and their properties. *Cognitive Science 6*, 3 (1982), 205–254.

[33] FELZENSZWALB, P., MCALLESTER, D., AND RAMANAN, D. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (2008), pp. 1–8.

[34] FRIEDENBERG, J., AND SILVERMAN, G. *Cognitive science : an introduction to the study of mind*. Sage, Thousand Oaks, 2006.

[35] FRIEDMAN-HILL, S., ROBERTSON, L., AND TREISMAN, A. Parietal contributions to visual feature binding: Evidence from a patient with bilateral lesions. *Science (New York, N.Y.) 269* (09 1995), 853–5.

[36] FRIEDRICH, F. . *Attention in NOBA textbook series: Psychology*. DEF Publishers.

[37] FU, W.-T., AND PIROLLI, P. Snif-act: A cognitive model of user navigation on the world wide web. *Human–Computer Interaction 22*, 4 (2007), 355–412.

[38] GAZZANIGA, M., IVRY, R., AND MANGUN, G. *Cognitive Neuroscience: The Biology of Mind*. W.W. Norton, 01 2002.

[39] GIRSHICK, R. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 1440–1448.

[40] GIRSHICK, R., DONAHUE, J., DARRELL, T., AND MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (11 2013).

[41] GRAPHPAD. Confidence interval of a standard deviation. `https://www.graphpad.com/guides/prism/latest/statistics/stat_confidence_interval_of_a_stand.htm`.

[42] GRAY, J. A., AND WEDDERBURN, A. A. I. Shorter articles and notes grouping strategies with simultaneous stimuli. *Quarterly Journal of Experimental Psychology 12*, 3 (1960), 180–184.

[43] HAFED, Z. M., AND CLARK, J. J. Microsaccades as an overt measure of covert attention shifts. *Vision Research 42*, 22 (2002), 2533–2545.

[44] HAJI-ABOLHASSANI, A., AND CLARK, J. J. An inverse yarbus process: Predicting observers' task from eye movement patterns. *Vision Research 103* (2014), 127–142.

[45] HE, K., GKIOXARI, G., DOLLÁR, P., AND GIRSHICK, R. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence 42*, 2 (2020), 386–397.

[46] HOFFMAN, J. E., AND SUBRAMANIAM, B. The role of visual attention in saccadic eye movements. *Perception and Psychophysics* (1995), 787795.

[47] HORSTMANN, G., BECKER, S., AND GRUBERT, A. Dwelling on simple stimuli in visual search. *Attention, Perception, and Psychophysics 82* (2019), 607–625.

[48] HUANG, Y., CAI, M., LI, Z., AND SATO, Y. Predicting gaze in egocentric video by learning task-dependent attention transition. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 754–769.

[49] HUI, J. Understanding feature pyramid networks for object detection (fpn). `https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c`, 2018.

[50] ITTI, L., AND BALDI, P. Bayesian surprise attracts human attention. *Vision Research 49*, 10 (2009), 1295–1306. Visual Attention: Psychophysics, electrophysiology and neuroimaging.

[51] ITTI, L., AND KOCH, C. Computational modeling of visual attention. *Nature reviews. Neuroscience 2* (04 2001), 194–203.

[52] ITTI, L., KOCH, C., AND NIEBUR, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 11 (1998), 1254–1259.

[53] IVYPANDA. Comparison of guided search model and feature integration model of visual search. https://ivypanda.com/essays/comparison-of-guided-search-model-and-feature-integration-model-of-visual-search/, (2019, December 29).

[54] JIANG, M., HUANG, S., DUAN, J., AND ZHAO, Q. Salicon: Saliency in context. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 1072–1080.

[55] JOKINEN, J. P., WANG, Z., SARCAR, S., OULASVIRTA, A., AND REN, X. Adaptive feature guidance: Modelling visual search with graphical layouts. *International Journal of Human-Computer Studies 136* (2020), 102376.

[56] JUDD, T., EHINGER, K., DURAND, F., AND TORRALBA, A. Learning to predict where humans look. In *IEEE International Conference on Computer Vision (ICCV)* (2009).

[57] KAHNEMAN, D. *Attention and Effort*. Prentice-Hall, 1973.

[58] KANAN, C., TONG, M. H., ZHANG, L., AND COTTRELL, G. W. Sun: Top-down saliency using natural statistics. *Visual Cognition 17*, 6-7 (2009), 979–1003.

[59] KOCH, C., AND ULLMAN, S. Selecting one among the many: A simple network implementing shifts in selective visual attention. *Artificial Intelligence Lab Memo No. 770* (01 1984).

[60] KRISTJÁNSSON, Á., ÓLAFSDÓTTIR, I. M., AND KRISTJÁNSSON, T. *Visual Foraging Tasks Provide New Insights into the Orienting of Visual Attention: Methodological Considerations*. Springer US, New York, NY, 2020, pp. 3–21.

[61] KRISTJÁNSSON, T., THORNTON, I., CHETVERIKOV, A., AND KRISTJÁNSSON, A. Dynamics of visual attention revealed in foraging tasks. *Cognition 194* (08 2019), 104032.

[62] KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*

(2012), F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25, Curran Associates, Inc.

[63] KRONER, A., SENDEN, M., DRIESSENS, K., AND GOEBEL, R. Contextual encoder–decoder network for visual saliency prediction. *Neural Networks 129* (2020), 261–270.

[64] KÜMMERER, M., BYLINSKII, Z., JUDD, T., BORJI, A., ITTI, L., DURAND, F., OLIVA, A., AND TORRALBA, A. Mit/tübingen saliency benchmark. https://saliency.tuebingen.ai/.

[65] KÜMMERER, M., WALLIS, T. S. A., AND BETHGE, M. Information-theoretic model comparison unifies saliency metrics. *Proceedings of the National Academy of Sciences 112*, 52 (2015), 16054–16059.

[66] KUMMERER, M., WALLIS, T. S. A., AND BETHGE, M. Saliency benchmarking made easy: Separating models, maps and metrics. In *Proceedings of the European Conference on Computer Vision (ECCV)* (September 2018).

[67] KUMMERER, M., WALLIS, T. S. A., GATYS, L. A., AND BETHGE, M. Understanding low- and high-level contributions to fixation prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Oct 2017).

[68] KÜMMERER, M., THEIS, L., AND BETHGE, M. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. *arXiv* (2015).

[69] KÜMMERER, M., WALLIS, T. S. A., AND BETHGE, M. Deepgaze ii: Reading fixations from deep features trained on object recognition. *arXiv* (2016).

[70] LAN, M., ZHANG, Y., ZHANG, L., AND DU, B. Global context based automatic road segmentation via dilated convolutional neural network. *Information Sciences 535* (2020), 156–171.

[71] LAVIE, N. Perceptual load as a necessary condition for selective attention. *Journal of experimental psychology. Human perception and performance 21* (07 1995), 451–68.

[72] LAVIE, N., AND COX, S. On the efficiency of visual selective attention: Efficient visual search leads to inefficient distractor rejection. *Psychological Science 8*, 5 (1997), 395–398.

[73] LI, Y., FATHI, A., AND REHG, J. M. Learning to predict gaze in egocentric video. In *2013 IEEE International Conference on Computer Vision* (2013), pp. 3216–3223.

[74] LIN, T.-Y., DOLLÁR, P., GIRSHICK, R., HE, K., HARIHARAN, B., AND BELONGIE, S. Feature pyramid networks for object detection. *arXiv* (2017).

[75] LIN, Y., TANG, Y. Y., FANG, B., SHANG, Z., HUANG, Y., AND WANG, S. A visual-attention model using earth mover's distance-based saliency measurement and nonlinear feature combination. *IEEE Trans. Pattern Anal. Mach. Intell. 35*, 2 (Feb. 2013), 314–328.

[76] MATTHIAS KUMMERER. pysaliency. `https://github.com/matthias-k/pysaliency`, 2013.

[77] MISHRA, A., ALOIMONOS, Y., AND FAH, C. L. Active segmentation with fixation. In *2009 IEEE 12th International Conference on Computer Vision* (2009), pp. 468–475.

[78] MOFFITT, K. Evaluation of the fixation duration in visual search. *Perception and psychophysics 27* (05 1980), 370–2.

[79] NAVALPAKKAM, V., AND ITTI, L. An integrated model of top-down and bottom-up attention for optimizing detection speed. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2* (02 2006), 2049 – 2056.

[80] NAVALPAKKAM, V., AND ITTI, L. Search goal tunes visual features optimally. *Neuron 53*, 4 (2007), 605–617.

[81] NYAMSUREN, E., AND TAATGEN, N. A. Pre-attentive and attentive vision module. *Cognitive Systems Research 24* (2013), 62–71. Cognitive Systems Research:Special Issue on ICCM2012.

[82] OLIVA, A., AND TORRALBA, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision 42* (05 2001), 145–175.

[83] PALAZZI, A., ABATI, D., CALDERARA, S., SOLERA, F., AND CUCCHIARA, R. Predicting the driver's focus of attention: The dr(eye)ve project. *IEEE Transactions on Pattern Analysis and Machine Intelligence PP* (05 2017).

[84] PETERS, R. J., IYER, A., ITTI, L., AND KOCH, C. Components of bottom-up gaze allocation in natural images. *Vision Research 45*, 18 (2005), 2397–2416.

[85] PFEUFFER, K., AND LI, Y. Analysis and modeling of grid performance on touchscreen mobile devices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2018), CHI '18, Association for Computing Machinery, p. 1–12.

[86] PRPIC, V., KNIESTEDT, I., CAMILLERI, E., MAUREIRA, M. G., KRISTJÁNSSON, Á., AND THORNTON, I. A serious game to explore human foraging in a 3d environment. *PLoS ONE 14* (2019).

[87] REN, S., HE, K., GIRSHICK, R., AND SUN, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence 39* (06 2015).

[88] RESEARCHERS, T. B. M. Types of eye movements. `https://thebrain.mcgill.ca/flash/capsules/pdf_articles/type_eye_movement.pdf`.

[89] ROSENHOLTZ, R. A simple saliency model predicts a number of motion popout phenomena. *Vision Research 39*, 19 (1999), 3157–3163.

[90] ROSENHOLTZ, R., NAGY, A. L., AND BELL, N. R. The effect of background color on asymmetries in color search. *Journal of Vision 4*, 3 (03 2004), 9–9.

[91] RUBIN, E. *Synsoplevede figurer: Studier i psykologisk analyse.* København og Kristiania: Gyldendal, Nordisk forlag, 1915.

[92] RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., HUANG, Z., KARPATHY, A., KHOSLA, A., BERNSTEIN, M., BERG, A. C., AND FEI-FEI, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV) 115*, 3 (2015), 211–252.

[93] SIMONYAN, K., AND ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[94] SMITHSON, J. Rubin vase image. `https://en.wikipedia.org/w/index.php?title=File:Rubin2.jpg`, 2007.

[95] STROOP, J. R. Studies of interference in serial verbal reactions. *Journal of Experimental Psychology 18*, 6 (1935), 643–662.

[96] TATLER, B., BADDELEY, R., AND GILCHRIST, I. Visual correlates of fixation selection: Effects of scale and time. *Vision research 45* (04 2005), 643–59.

[97] TEHRANCHI, F., AND RITTER, F. Modeling visual search in interactive graphic interfaces: Adding visual pattern matching algorithms to act-r. In *Proceedings of ICCM 2018 - 16th International Conference on Cognitive Modeling* (2018), I. Juvina, J. Houpt, and C. Myers, Eds., Proceedings of ICCM 2018 - 16th International Conference on Cognitive Modeling, University of Wisconsin, pp. 162–167.

[98] TEO, L.-H., JOHN, B., AND BLACKMON, M. Cogtool-explorer: A model of goal-directed user exploration that considers information layout. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2012), CHI '12, Association for Computing Machinery, p. 2479–2488.

[99] TODI, K., JOKINEN, J., LUYTEN, K., AND OULASVIRTA, A. Individualising graphical layouts with predictive visual search models. *ACM Trans. Interact. Intell. Syst. 10*, 1 (Aug. 2019).

[100] TORRALBA, A., OLIVA, A., CASTELHANO, M., AND HENDERSON, J. Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search. *Psychological review 113* (11 2006), 766–86.

[101] TREISMAN, A. Monitoring and storage of irrelevant messages in selective attention. *Journal of Verbal Learning and Verbal Behavior 3*, 6 (1964), 449–459.

[102] TREISMAN, A., AND GELADE, G. A feature-integration theory of attention. *Cognitive Psychology 12*, 1 (1980), 97–136.

[103] VIG, E., DORR, M., AND COX, D. Large-scale optimization of hierarchical features for saliency prediction in natural images. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 2798–2805.

[104] VIOLA, P., AND JONES, M. Robust real-time object detection. In *International Journal of Computer Vision - IJCV* (01 2001), vol. 57.

[105] WEDEL, M., AND PIETERS, R. Eye tracking for visual marketing. *International Journal of Cooperative Information Systems* (2008).

[106] WOLFE, J. Guided search 2.0 a revised model of visual search. *Psychonomic Bulletin and Review 1* (06 1994), 202–238.

[107] WOLFE, J., CAVE, K., AND FRANZEL, S. Guided search: An alternative to the feature integration model for visual search. *Journal of experimental psychology. Human perception and performance 15* (09 1989), 419–33.

[108] WOLFE, J. M., AIZENMAN, A. M., BOETTCHER, S. E., AND CAIN, M. S. Hybrid foraging search: Searching for multiple instances of multiple types of target. *Vision Research 119* (2016), 50–59.

[109] YANG, Z., HUANG, L., CHEN, Y., WEI, Z., AHN, S., ZELINSKY, G., SAMARAS, D., AND HOAI, M. Predicting goal-directed human attention using inverse reinforcement learning. pp. 190–199.

[110] YU, F., AND KOLTUN, V. Multi-scale context aggregation by dilated convolutions. *arXiv* (2016).

[111] YUAN, A., AND LI, Y. Modeling human visual search performance on realistic webpages using analytical and deep learning methods. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2020), CHI '20, Association for Computing Machinery, p. 1–12.

[112] ZELINSKY, G., YANG, Z., HUANG, L., CHEN, Y., AHN, S., WEI, Z., ADELI, H., SAMARAS, D., AND HOAI, M. Benchmarking gaze prediction for categorical visual search. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2019), pp. 828–836.

[113] ZHANG, M., FENG, J., MA, K., LIM, J., ZHAO, Q., AND KREIMAN, G. Finding any waldo with zero-shot invariant and efficient visual search. *Nature Communications 9* (09 2018).

[114] ZHENG, Q., JIAO, J., CAO, Y., AND LAU, R. W. H. Task-driven webpage saliency. In *Computer Vision – ECCV 2018* (Cham, 2018), V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Springer International Publishing, pp. 300–316.