Human-Computer Perception: Modeling Visual Perceptual Attributes

by

Anelise P. Newman

S.B., Massachusetts Institute of Technology (2019)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2020

Author .....
Department of Electrical Engineering and Computer Science
May 12, 2020

Certified by .....
Aude Oliva
Principal Research Scientist, CSAIL
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Human-Computer Perception: Modeling Visual Perceptual Attributes

by

Anelise P. Newman

Submitted to the Department of Electrical Engineering and Computer Science
on May 12, 2020, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Human perception provides clues as to which visual content is most crucial or engaging. Where people look indicates what they pay attention to and find relevant; what people remember is what the human brain deems to be worthy of preservation. Recently, Deep Neural Networks have made it possible to predict cognitive attributes like saliency and memorability from just an image or video, at the same time that advances in human-computer interaction and human cognition have made collecting human data more accessible than ever. In this work, we aim to reinforce the interplay between human perception and computational models. We develop new strategies for collecting perceptual data, build models that predict human responses to visual stimuli, and show how applications of these models can be used to prioritize content for human consumption. First, we develop a toolbox of web-based user interfaces for crowdsourcing attention data using only a laptop or mobile phone. Through experimentation and analysis, we show how to deploy these interfaces to collect attention data scalably and flexibly for a variety of use cases. Next, we use our toolbox to study a novel aspect of human attention, resulting in the first saliency model that is capable of producing multiple saliency heatmaps corresponding to different potential viewing durations. Finally, we turn our focus to memorability, by designing an online memory game to measure and predict how likely a person is to remember a video. Systems like these that are capable of modeling human perception can make intelligent decisions about what information to prioritize, create, enhance, and preserve.

Thesis Supervisor: Aude Oliva
Title: Principal Research Scientist, CSAIL

Acknowledgments

Thank you to Aude Oliva, for her enery, trust, support, and guidance, and for being an incredible advisor.

Thank you to Zoya Bylinskii, who first taught me how to do research, and who has been an amazing mentor and role model.

Thank you to my coauthors–Camilo Fosco, Barry McNamara, Yun Bin Zhang, Pat Sukhum, Matthew Tancik, Nam Wook Kim, Nanxuan Zhao, Allen Lee, and Vincent Casser–who made this research possible, and my labmates, who made it a joy to do.

And thank you to my friends and my family–my mom, Lisa, my dad, Kermit, and my brothers, Max and Colin–who have always been my greatest source of support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The human brain–especially the visual system–is a highly optimized data processing machine. It takes in a constant stream of visual data and quickly determines what is relevant to focus on in the present and what information may be useful in the future. For example, it must decide where to allocate attention by moving the eyes to look at the most pertinent location in a scene. It is continuously deciding which moments and details are important to preserve in memory and which can be safely forgotten. The brain automatically hones in on the most relevant parts of visual experience.

In this way, **human perception provides clues as to which visual content is most crucial or engaging.** Where people look indicates what they pay attention to and find relevant; what people remember is what the human brain deems to be worthy of preservation. These clues can be harnessed by computerized systems to produce predictions and applications that are tailored to a human audience.

Why might models of human visual perception be useful? For one, they can help produce content that is more engaging or effective for human viewers. For example, a model of human attention could give a designer suggestions about how to make a graphic design element more or less visually salient. A memorability model could guide educators to create catchy tutorials that their students are more likely to retain. Such models can also be used to prioritize and filter content for human consumption, such as by automatically cropping the most salient region of an image (see Figure 1-1), curating memorable moments from a camera roll, or rendering elements in a

Figure 1-1: Saliency-based cropping: an example application of human perceptual models. Compared to a simple center crop (middle column), crops based on where people are likely to look do a better job of focusing on the relevant actors in an image. These images are from the CAT2000 dataset [11] and saliency crops were generated based on predictions from our Multi-Duration Saliency Excited Model (Chapter 4).

virtual environment in the order in which they are likely to be viewed. Granting computers some of the data-processing power of the human brain can lead to outputs that are better tailored for people.

However, models like these require human data, for evaluation if not for training. In the past decade, computer vision has made huge strides by harnessing neural networks for a variety of tasks, including saliency and memorability prediction. These data-hungry models have increased demand for copious and on-demand human data, often collected through online crowdsourcing. Thus, designing experiments to measure human perception is also an active area of research.

In this work, we aim to reinforce the loop between human perception and computational models, depicted in Figure 1-2. We will focus on two main aspects of visual perception and cognition: attention and memory. For each, we will consider user interfaces and experimental strategies for collecting human data, models for predicting human responses, and potential applications.

In Chapter 2, we will introduce relevant background about measuring and model-

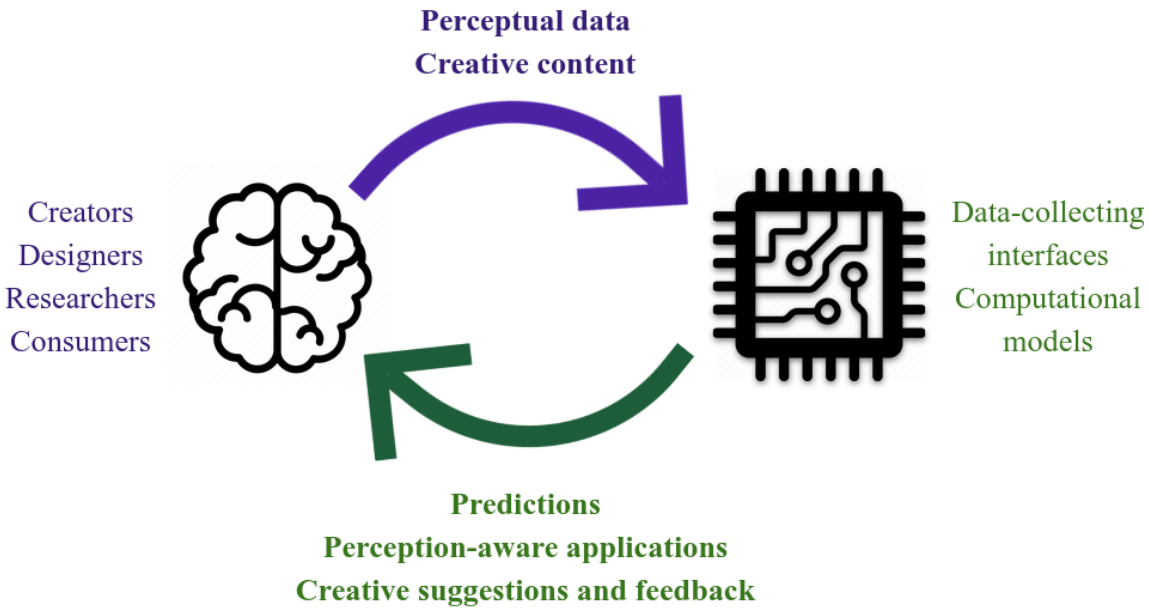

Figure 1-2: There is a rich interplay between human perception and perceptual models. Carefully-designed data-collecting interfaces collect perceptual data from human subjects. This data is used to train computational models to produce useful predictions. These predictions then feed another creative cycle of human-computer interaction, where creators such as designers or educators use perceptual predictions to produce more eye-catching, engaging, or memorable content.

ing attention and memorability. In Chapter 3, we will present a toolbox of web-based user interfaces for capturing human attention data using only a laptop or mobile phone, without eye tracking. In Chapter 4, we will use these tools to build a dataset and a predictive model for a new aspect of attention. In Chapter 5, we will turn our focus to memorability, in order to measure the memorability of video clips and predict how well someone will retain new events. Finally, in Chapter 6, we will discuss exciting future directions in harnessing human perception.

Chapter 2

Background and related work

In this chapter, we will discuss prior work on modeling saliency and memorability of visual stimuli. First, we will introduce the experimental paradigms and UI tools that have been used to collect data about these perceptual attributes. Then, we will discuss the state-of-the-art in computational models.

2.1 User interfaces for gathering perceptual data

2.1.1 Crowdsourcing human attention

Background on visual attention. Visual attention refers to the ability to selectively focus on part of a visual stimulus. Eye tracking is a commonly-used proxy to determine where on an image a person is attending. By aggregating the eye movements of multiple participants, we can measure which parts of an image are most likely to draw attention. This data is often represented as a *saliency heatmap* where higher heatmap values correspond to more-attended image regions (Figure 2-1). These heatmaps are created by blurring fixated (gazed) locations with a Gaussian.

However, specialized eye tracking equipment is expensive, and bringing participants into the lab to collect data can be slow, difficult to scale, and impractical. Online crowdsourcing, whereby crowdworkers complete short tasks through a computer interface on a platform such as Amazon's Mechanical Turk, provides an alternative

Figure 2-1: An example saliency heatmap from the CAT2000 eye tracking dataset [11]. **Left:** original image. **Center:** saliency heatmap. **Right:** heatmap overlaid on image.

means of collecting human data. Researchers have tried using built-in webcams to obtain coarse-grained eye data from crowdworkers [73, 86], but these methods are insufficiently robust, requiring controlled conditions. Efforts have thus turned to interaction techniques that approximate eye movements, falling into one of the following four categories.

Cursor-based interfaces. Prior work has investigated the correlation between mouse and gaze locations [45, 52, 93]. Cursor movements can complement eye movements, especially when a participant can use both to interact with visual content. A separate line of work considered using cursor-based interfaces to approximate eye tracking [7, 62, 98]. For instance, the moving-window methodology reveals only portions of an otherwise-obscured image depending on where a user positions the mouse cursor [60, 79, 90, 103]. An example of a moving-window interface is BubbleView, which was extensively explored in [69].

Self-report interfaces. Moving-window methodologies like BubbleView distort the underlying image. An alternative is to show viewers an undistorted image and ask them to report where they looked, often with the help of a visual aid like a labeled grid [26, 94]. Our CodeCharts interface (discussed in Chapter 3) is based on the work of Rudoy et al. [94].

Zoom-based interfaces. Zoom allows users to expand content that they find engaging and want to view in greater detail [5, 8]. Previous work investigated the zoomable viewport on a mobile phone as a measure of user engagement with an interface or list of search results [46, 47, 74, 75, 76, 78]. Huang et al. even proposed generating heatmaps based on the viewport [51]. Our ZoomMaps methodology (discussed in Chapter 3) expands on this work by using viewport data to produce an attention heatmap on an arbitrary image, treating the mobile phone as a restricted window through which users explore areas of interest.

Annotation interfaces. UI tools for collecting object segmentations in images were developed to produce training data for computer vision tasks such as object detection and recognition [95, 96]. However, they have also been used to identify graphic design elements that a viewer rates as important. ImportAnnots (discussed in Chapter 3) refers to the interface for capturing explicit "Importance Annotations", first introduced by O'Donovan et al. [83], and has been used to collect data for training computational models to predict importance of graphic designs [19, 83].

2.1.2 Measuring memorability

A landmark result in cognitive science is that memorability is an *intrinsic* property of an image: people are remarkably consistent in which images they remember and forget [4, 16, 44, 55, 56, 59, 67, 80]. Thus, it is possible to design an experiment to quantify the memorability of a visual stimulus. This is often done using the classical old-new recognition paradigm, in which subjects are shown a sequence of stimuli and asked to press a key when they recognize a repeated item. The fraction of people that recognize a repeated image, aggregated over many subjects, serves as its "memorability score" (Figure 2-2). This paradigm allows researchers to collect objective measurements of human memory at a large scale [14, 56, 67] and variable time scales.

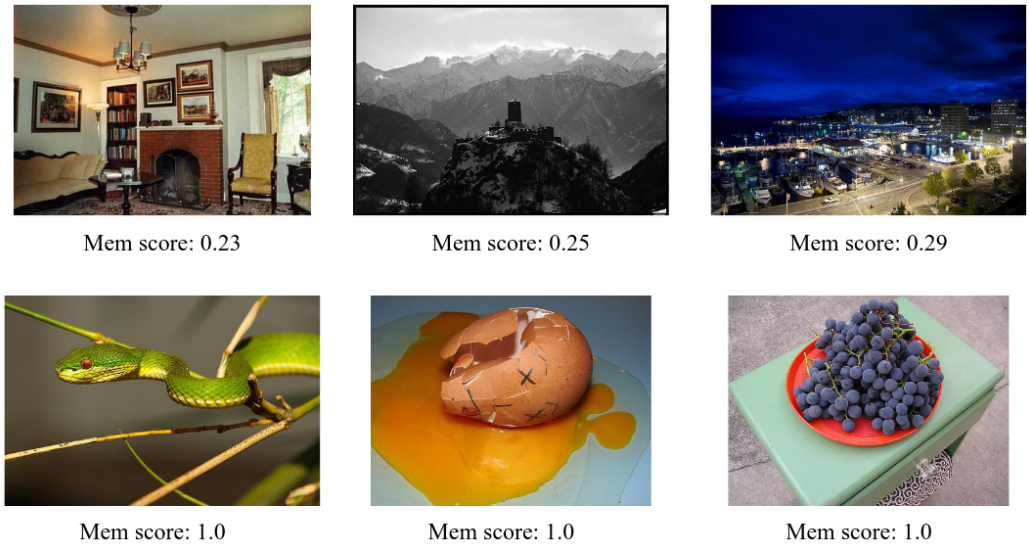

Figure 2-2: Some of the least (top row) and most (bottom row) memorable images from the LaMem test set [67], including their memorability scores.

2.2 Predicting visual saliency

The availability of large-scale saliency datasets has spurred interest in predictive models of visual saliency, which are increasing based on convolutional neural networks.

Saliency modeling. The large-scale attention datasets captured using the moving-window approaches SALICON [62] and BubbleView [69] enabled training neural network models of saliency (e.g., [19, 33, 53, 84, 91]). The top performers on the MIT Saliency Benchmark [17] were trained on SALICON data and have opened a wide performance gap to the previous, traditional models of saliency [20]. Driven by such improvements in efficiency and accuracy, saliency models have found wide use in applications like image cropping, retargeting, and view-finding for improved composition [10, 24, 38, 107].

Metrics for quantifying saliency. Saliency networks take in an image and output a saliency heatmap. To quantify the similarity of a predicted heatmap to a ground-truth heatmap, we use the Pearson's Correlation Coefficient (CC) and Normalized Scanpath Saliency (NSS) metrics. CC and NSS are the preferred metrics for evaluating saliency predictions and are highly correlated [18]. CC measures the pixel-wise correlation between two normalized heatmaps and ranges from -1 (inversely

correlated) to 1 (perfectly correlated). NSS measures the mean value of a normalized attention heatmap evaluated at ground-truth eye fixation locations and ranges from 0 (no heatmap density at fixated locations) to infinity (all heatmap density at fixated locations). These metrics were studied in detail in [18].

Scanpath modeling. A complementary approach to representing and modeling human attention is via scanpaths: the sequence of gaze locations that an observer makes on an image over time. Scanpath analysis and modeling is complicated by the fact that individual differences are huge at the level of single gaze locations [3, 77]. This hides the fact that different permutations of traversing image content may nevertheless correspond to a similar allocation of attention to the respective image regions.

2.3 Predicting memorability

The fact that memorability is intrinsic to a visual stimulus means that an item's memorability can be predicted from the stimulus itself. Thus, memorability has become an active field of research in computer vision.

Image memorability. Early works on computational memorability [54, 55, 56] pointed to content of images that were predictive of their memorability (i.e. people, animals and manipulable objects are memorable, but landscapes are often forgettable). Later work replicated the initial findings and extended prediction of memorability to many categories of photographs [2, 6, 35, 39, 68, 88, 112], faces [4, 66, 100], visualizations [12, 13] and videos [29, 31, 99]. The development of large-scale image datasets augmented with memorability scores [67] have allowed convolutional neural network models to predict image memorability at near human-level consistency [2, 6, 39, 67, 112] and even generate realistic memorable and forgettable photos [43, 100].

Video memorability. Compared to work on images, large-scale work on video memorability prediction has been limited. Cohendet et al. [30] introduced a video-based memorability dataset [29, 31] and made progress towards building a predictive

model of video memorability. Other works on video memorability have largely relied on smaller datasets collected using paradigms that are more challenging to scale. For example, Han et al. [48] collected memorability and fMRI data on 2400 video clips and showed that aligning audio-visual features with brain data improves prediction. Shekhar et al. [99] used a language-based recall task to collect memorability scores as a function of response time for 100 videos. They find that a combination of semantic, spatio-temporal, saliency, and color features can be used to predict memorability scores. Cohendet et al. [31] collected a long-term memorability dataset using clips from popular movies.

Relevance of semantic features. Past work has confirmed the usefulness of semantic features for predicting memorability [29, 101, 99]. Previous work at the intersection of Computer Vision and NLP has aimed to bridge the gap between images and text by generating natural language descriptions of images using encoder-decoder architectures (e.g. [65, 105, 110]) or by creating aligned embeddings for visual and textual content [36, 41, 70]; we draw on both these techniques in this work.

Metrics for memorability evaluation. A commonly-used metric for evaluating memorability is the Spearman rank correlation (RC) between the memorability ranking produced by the ground-truth memorability scores versus the predicted scores. This is a popular metric [29, 56, 67] because memorability rankings are generally robust across experimental designs. However, it does not capture the absolute accuracy of the predicted memorability values. Therefore, when evaluating the ability of our model to produce interpretable memorability value, we also report the R^2 value of our predictions.

Chapter 3

The TurkEyes Toolbox: crowdsourcing human attention

Figure 3-1: We consider approaches for crowdsourcing human visual attention data without the use of an eye tracker. The attention heatmaps generated by two of the TurkEyes interfaces, CodeCharts and ZoomMaps, mimic heatmaps obtained using eye tracking. While CodeCharts closely approximates eye movements, ZoomMaps gives a coarser approximation of attention with more emphasis on distant details. This chapter will cover how these methodologies capture stable aspects of human attention and the unique features that make them suitable for different applications.

Eye movements provide insight into what parts of an image a viewer finds most salient, interesting, or relevant to the task at hand. Unfortunately, eye tracking data,

a commonly-used proxy for attention, is cumbersome to collect. In this chapter, we explore an alternative method of capturing human attention: a comprehensive web-based toolbox for crowdsourcing visual attention data [1].

3.1 Problem and approach

Gaze provides a window into what aspects of an image, design, or visualization people find most engaging. Where someone looks on an image can predict whether they remember it or not [12, 16]. Attention-grabbing regions of a poster can be used to summarize the design for later retrieval [19]. The most salient parts of an image can guide automatic cropping and retargeting [37]. All of these applications rely on inferring where people are paying attention by capturing where they are looking.

However, attention data has historically been difficult to collect at scale, as it involves in-lab eye tracking using dedicated hardware. The time it takes to recruit and run each participant prevents quick data collection and iteration. Meanwhile, online crowdsourcing allows for rapidly collecting large amounts of human data. Although webcam-based eye tracking has been proposed as a crowdsourceable alternative [73, 86, 113], it has many requirements, such as specific lighting conditions and participant pose, that are difficult to enforce.

In this chapter, we seek to analyze and expand the state-of-the-art in interaction methodologies for capturing attention. We present TurkEyes[2], a toolbox of four interfaces for gathering attention data using just a laptop or mobile phone. None of the interfaces we consider explicitly measure eye movements. Rather, we make use of interaction methodologies from the literature that are correlated with visual attention (Figure 3-1). **ZoomMaps** is a novel *zoom-based* interface that captures viewing on a mobile phone. **CodeCharts** [94] is a *self-reporting* methodology that records points of interest at precise viewing durations. **ImportAnnots** [83] is an *annotation* tool for selecting important image regions, and *cursor-based* **BubbleView**

[1] For a full discussion of the TurkEyes toolbox, see [82]

[2] Data, code, and demos available at: `http://TurkEyes.mit.edu/`

[69] lets viewers click to deblur a small area. We do a deep-dive on these interfaces by conducting extensive experiments on Amazon's Mechanical Turk in order to determine the benefits and the ideal use cases of each interface.

Our contributions in this chapter are: 1) a comprehensive toolbox that gathers attention-gathering interfaces into a common code and analysis framework and 2) a user guide explaining how, when, and why to deploy each interface to gather attention data tailored to a particular use case.

3.2 Introducing the TurkEyes Toolbox

In this section, we explain each of the TurkEyes interfaces.

3.2.1 ZoomMaps 

ZoomMaps users use the pinch-zoom gesture on a mobile phone to explore an image (Figure 3-2). ZoomMaps captures these zoom patterns by keeping track of what fraction of the image is in the viewport at any given time. We show that these zoom patterns can be used to approximate visual attention.

The interface. We built an online image gallery webpage augmented with tracking capabilities using the PhotoSwipe JavaScript library[3]. We modify the library to capture any changes to the visible region of the image along with a timestamp. The interface allows pinching to zoom in or out on an image and swiping to switch images (Figure 3-2). The interaction data contains viewport coordinates on the image and a timestamp for every *event* triggered by the user (i.e., the image is re-scaled or re-positioned on the screen). During crowdsourced experiments, study participants are sent to a landing page that contains a QR code and a URL that they use to open the image gallery in their mobile browser. We require that participants spend a minimum amount of time on each image and zoom in on at least 20% of the images. Depending on the experiment, they may also be asked to answer questions about the images they explore.

[3] `https://github.com/dimsemenov/photoswipe`

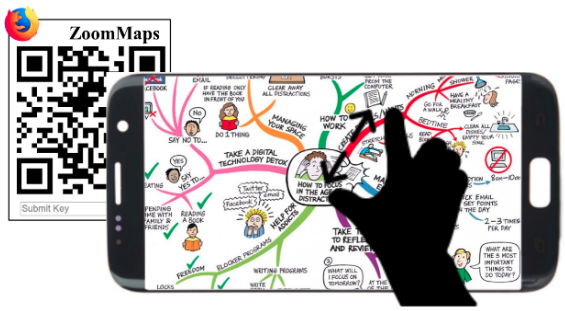

Figure 3-2: ZoomMaps UI. Participants use the pinch-zoom gesture on their phones to explore image content at higher resolutions.

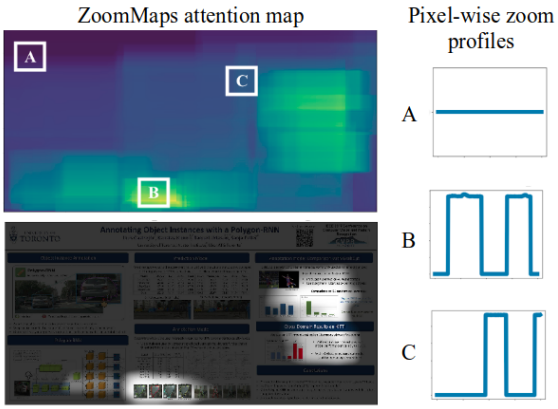

Figure 3-3: A ZoomMaps attention heatmap, visualized alone (top) and overlaid on the image used to create it (bottom). For three pixels, labeled A, B, and C, zoom over time is averaged to produce a value in the corresponding attention heatmap.

Generating attention heatmaps. Our mobile interface stores the bounding boxes of zoomed image regions along with timestamps of when they were in focus. We use this information to extract which parts of the image were viewed for how long. We then construct the attention heatmap as follows: for every pixel in the image, we compute its average *zoom level* over the entire viewing interval. We define the zoom level for an image region as the full image area divided by the area of the image region that has been magnified. We assign this zoom level to all the pixels contained in the image region. We then compute each pixel's average zoom level over the viewing duration to obtain a ZoomMaps attention heatmap (Figure 3-3). Higher values in the heatmap correspond to regions of the image that were inspected with closer zoom on average.

3.2.2 CodeCharts F37

CodeCharts collects individual gaze points by asking participants to self-report where they were looking on an image using a visual guide called a *codechart*.

The interface. A participant views an image on their screen for a preset duration,

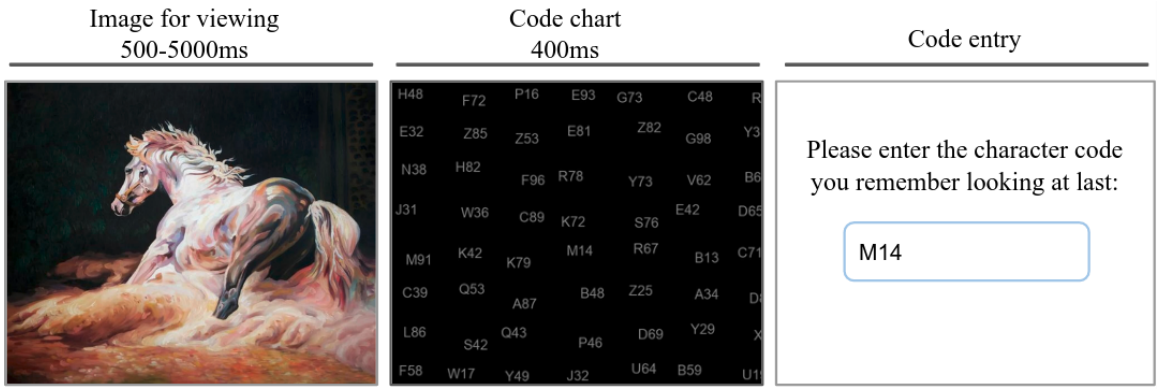

Figure 3-4: CodeCharts UI. Participants self-report a region of an image they gazed at using a grid of codes that appears after image presentation. Codes are shown larger than scale for ease of viewing.

typically a few seconds. When the image disappears it is replaced by a codechart, a jittered grid of three-character codes. The participant notes the code they see when the image vanishes, which approximates where on the image they were looking. When the codechart vanishes they self-report this alphanumeric code. This process, visualized in Figure 3-4, repeats for a sequence of images. We aim to display the codechart long enough for participants to read a triplet but not so long that their eyes can wander. We found that 400ms was the optimal exposure duration where the resulting data achieved maximum similarity to eye movements and participants maintained high accuracy at reporting a valid code.

Generating attention heatmaps. After collecting data from many participants (one gaze point per participant), we combine all the gaze points and blur them with a sigma of 50 to generate a heatmap. Our triplets are spaced approximately 100 pixels apart, so 50 is a good approximation of the radius of uncertainty in the interface.

3.2.3 ImportAnnots

O'Donovan et al. introduced the idea of having crowdworkers annotate important elements on graphic designs using binary masks and averaging them to construct importance heatmaps [83]. For our toolbox, we re-purpose the initial interface, add a validation procedure, and test the interface on different image types including natural

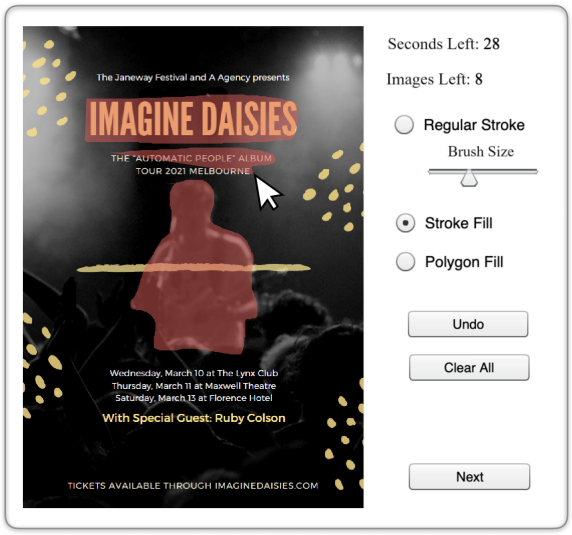

Figure 3-5: ImportAnnots UI. Participants paint over regions of a design that they consider important using binary masks. The mask is shown as a transparent red overlay.

scenes, infographics, and resumes.

The interface. Participants are presented with a series of images one at a time and are asked to annotate the most important regions (Figure 3-5). We do not define what should be considered "important". We used legacy code from O'Donovan et al. [83] for the annotation tool embedded in our task interface. It provides 3 annotation tools: *stroke fill* that allows tracing the contours of an object to provide a fine-grained segmentation, *polygon fill* that allows plotting points with connected lines for coarser annotation, and *regular stroke* for painting over a region. In our experiments, some of the images shown are validation images with a single main textual or graphical element; participants were required to correctly annotate the single salient element.

Generating attention heatmaps. Each participant generates one binary mask per image. The binary masks are averaged across participants to produce an overall attention heatmap. Despite high inter-observer variability and noisy annotations, averaged over a large number of participants (20-30), the mean importance maps give a plausible ranking of importance (see appendix of [83]).

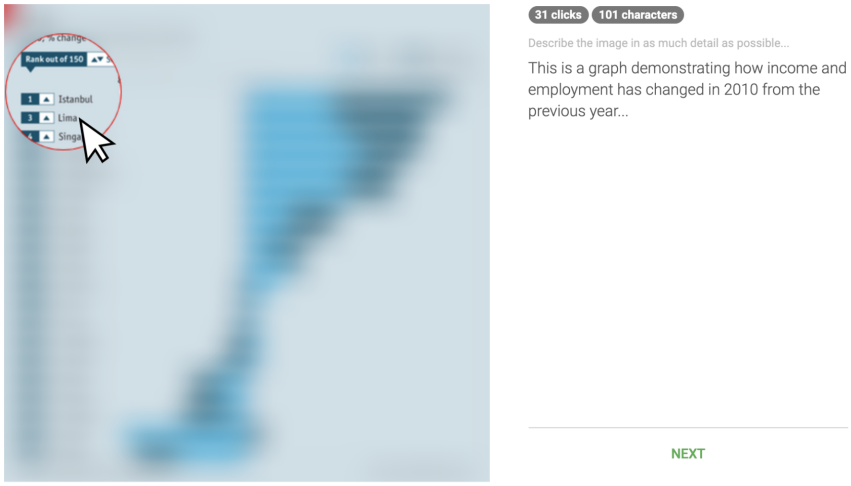

Figure 3-6: BubbleView UI. Participants click to deblur/expose small regions of an otherwise blurry image.

3.2.4 BubbleView

BubbleView is a cursor-based moving-window methodology. The original image is blurred to distort text regions and disable legibility, requiring participants to click around to de-blur small, circular "bubble" regions at full resolution (Figure 3-6). BubbleView was initially introduced in [69], which included a thorough comparison to eye movements on different image types and tasks. We reuse some of those analyses in this chapter.

The interface. Participants are asked to explore an image using their mouse cursor to click and deblur regions of an image. Clicking on a new location re-blurs the previously clicked location. Blurring the image loosely approximates peripheral vision. For free-viewing tasks (no instructions other than to freely explore the image), viewing time is fixed. As noted in Kim et al. [69], it takes 2-3 times longer for participants to explore an image with BubbleView than to view it naturally for the same number of gaze points in the same unit time. In other words, this interface slows down visual processing relative to natural viewing.

Generating attention heatmaps. Given a set of BubbleView mouse clicks on an image, an attention heatmap is computed by blurring the click locations with a

Gaussian with a particular sigma (a different one per image dataset [69]).

3.3 Toolbox evaluation

In this section, we analyze the attention data captured by the TurkEyes interfaces in order to evaluate them along various axes of interest. We used our interfaces to crowdsource attention data on a variety of stimuli (including natural and non-natural images) to determine what insights are discoverable by each tool. Although all the interfaces capture some common aspects of attention, they are best suited for different image types and tasks, and we provide guidelines as to applicable use cases for each.

3.3.1 Cost of data collection

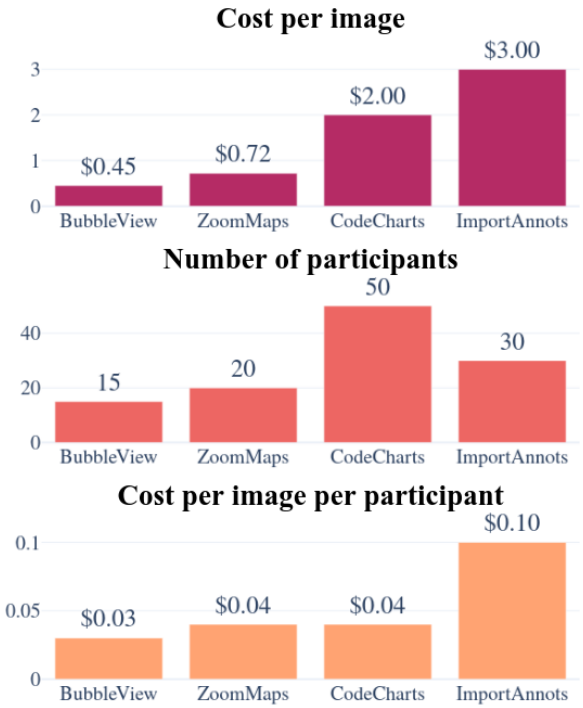

Figure 3-7: Cost comparison of the TurkEyes interfaces on natural images. We find that differences in price per image for each interface (top row) are driven more by number of participants required (middle row) than differences in the price of one person attending to an image (bottom row). ImportAnnots is a special case because it also requires the participant to segment objects, which drives up the time required per image and its price.

How much does it cost to obtain attention data for an image using each of these interfaces? The cost of obtaining an attention heatmap for an image is (number of participants required) $\times$ (cost per image per participant). The results of this calculation for our experiments on natural images are shown in the first row of Figure 3-7. BubbleView is the cheapest at \$0.45 per image, followed by ZoomMaps, CodeCharts, and ImportAnnots at \$3.00.

What drives this difference in the cost of attention? We observe that across interfaces, there is a remarkable consistency in cost per image per participant (bottom row of Figure 3-7). ImportAnnots stands out from this trend at a higher rate of roughly 10 cents per participant per image; this is because it requires segmentation in addition to paying attention. This suggests that the difference in price of an attention heatmap is driven by how many participants are required to obtain stable data. Thus, as a rule of thumb, we can expect interfaces that collect a lot of attention data per participant (such as BubbleView, which collects many clicks per image) to be cheaper than those that collect little data per participant (such as CodeCharts, which only collects a single gaze point).

3.3.2 Type of stimuli

Some types of visual stimuli are better suited for certain interfaces than for others; we consider some such cases below.

Image scale. ZoomMaps, ImportAnnots, and BubbleView allow panning/scrolling and are therefore compatible with images larger than the screen. By contrast, CodeCharts' brisk task progression requires that stimuli fit on the screen. This makes CodeCharts inappropriate for images with an extreme aspect ratio and limits the amount of detail that can be seen. Only ZoomMaps supports viewing images at varying resolutions. This makes it uniquely qualified to collect viewing data on images with multiscale content, such as infographics or data visualizations (Figure 3-8).

Natural vs. non-natural images. ImportAnnots is most appropriate for easily-segmentable, non-natural images, whereas the other interfaces can handle natural and non-natural images.

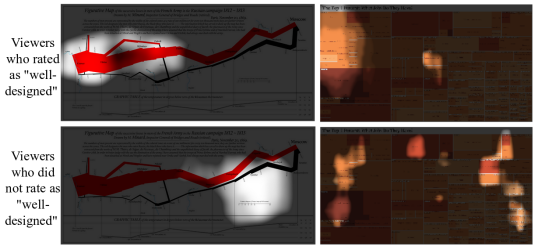

Figure 3-8: ZoomMaps on data visualizations. ZoomMaps is an ideal tool for evaluating complex images because viewers can study content at multiple scales via a natural interface. Here, we see ZoomMaps has potential as a visualization debugging tool; participants who rated an image as more "well-designed" had different viewing patterns from those who did not.

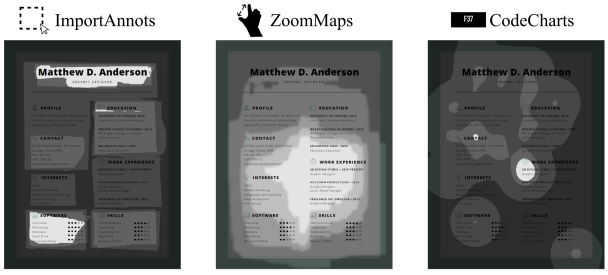

Figure 3-9: ImportAnnots, ZoomMaps, and CodeCharts on a resume. The interfaces expose different aspects of attention: CodeCharts shows what is immediately salient, ZoomMaps shows what people spent time exploring, and ImportAnnots shows what they think is most relevant after considering the document.

Dynamic content. Although we did not explore this possibility in our experiments, CodeCharts can collect gaze data on videos, as suggested in [94]. Instead of showing an image for a fixed duration, one can show a short video clip ending at a moment of interest to capture gaze locations at that frame. CodeCharts can also be used to collect attention data at different viewing durations, thus giving insight into how attention evolves with time.

Combining insights from multiple interfaces. These tools are not mutually exclusive. In fact, they can be used in combination to gain a more nuanced picture of attention (Figure 3-9).

3.3.3 Similarity to eye movements

How similar is this data to eye movements? To find out, we ran data collection using all four interfaces on a set of 35 images sampled from the CAT2000 dataset [11]. We computed the Normalized Scanpath Saliency (NSS) and Pearson's Correlation Coefficient (CC) scores for each of these attention heatmaps compared to ground-

	IOC	CodeCharts	BubbleView	ZoomMaps	ImportAnnots
CC	0.86	**0.76**	0.62	0.59	0.51
% of IOC	100%	**88%**	72%	69%	59%
NSS	2.42	**2.00**	1.58	1.37	1.22
% of IOC	100%	**83%**	65%	57%	50%

Table 3.1: Comparison of our toolbox to ground-truth eye movements. We report both Correlation Coefficient and Normalized Scanpath Saliency, where both metrics increase with higher similarity.

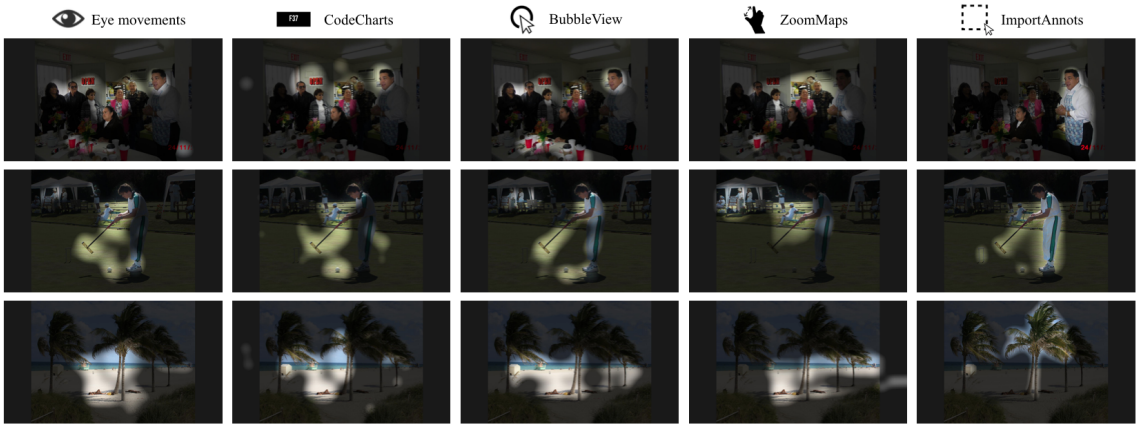

Figure 3-10: The TurkEyes interfaces compared to human eye movements on the CAT2000 dataset. CodeCharts best approximates human eye movements, including single fixations resulting from exploration. BubbleView also captures salient regions. ZoomMaps occasionally focuses on background elements instead of salient foreground objects, and ImportAnnots segments semantically important elements, often focusing on a single central object.

truth eye movements. As a human baseline, we computed Inter-Observer Consistency (IOC): for NSS, using attention heatmaps of N-1 participants to predict the remaining participant [18, 69]; for CC, comparing attention heatmaps of half the observers to the other half. The results are in Table 3.1. CodeCharts data is most similar to eye movements, accounting for over 80% of human consistency. It is followed by BubbleView, ZoomMaps, and ImportAnnots.

Figure 3-10 shows some representative examples of the results on CAT2000 images. Human gaze (whether collected using an eye tracker or using the CodeCharts UI) falls on certain object regions only (e.g., faces, hands, points of contact, etc.), as does BubbleView. By contrast, ImportAnnots tends to highlight a few objects per

scene, ascribing uniform importance over entire objects. ZoomMaps occasionally over-focuses on distant background objects at the expense of salient foreground objects, as in the middle row of Figure 3-10.

3.3.4 Saliency vs. importance

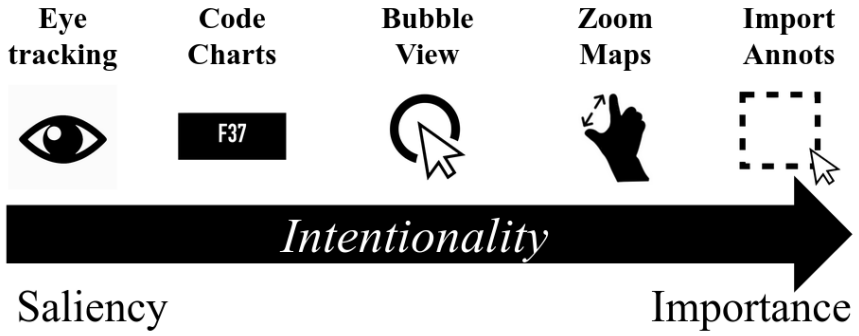

Figure 3-11: Our interfaces can be organized on an "intentionality" scale based on the degree to which they measure saliency (more spontaneous) or importance (more intentional).

Attention comes in different flavors. Saliency is a bottom-up measure of what parts of an image are most attention-grabbing [57]. It is most commonly measured by aggregating eye movements across participants. Importance is a top-down measure of which elements in an image are most relevant [83]. The former is more spontaneous and happens automatically during viewing, while the latter requires the viewer to consider and evaluate the image before making a determination.

We hypothesize that an interface's place on the saliency-importance continuum is a function of its "intentionality": the amount of cognitive processing required to use a particular interface's interaction methodology while viewing an image. The more involved a given interaction methodology, the more it elicits top-down importance; the less interaction required, the more closely it measures saliency. Interaction has the effect of slowing down the viewing process, allowing the user to explore the image before attention data is recorded.

Figure 3-11 places the attention-capturing interfaces on an intentionality scale, where intentionality increases and similarity to eye movements decreases to the right.

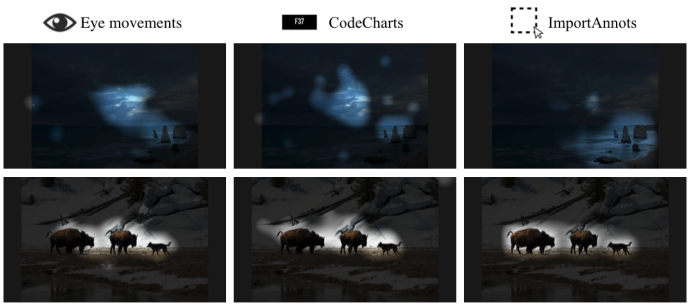

Figure 3-12: Natural images where CodeCharts and ImportAnnots agree and differ. **Top:** CodeCharts shows center bias and image explorations while ImportAnnots finds objects of interest. **Bottom:** The interfaces agree because the animals are salient and easily-segmented.

Eye tracking requires no explicit user interaction and thus is the most direct measure of saliency. CodeCharts is the second-best measure of saliency because it does not distort the image or require user interaction while viewing the image. BubbleView still captures image locations that draw people's attention, but is more intentional because it slows down viewing time, distorts the image, and requires users to click to expose areas of interest. ZoomMaps requires the user to decide to interact with the image by pinching and zooming, but uses a familiar and almost second-nature mechanism. Finally, ImportAnnots measures importance instead of saliency: participants are given ample time to view the image and are asked to select regions (not single gaze points) that best represent the content of the image after considering the entire design.

To understand the difference between saliency and importance, we compare data from interfaces on either end of the intentionality spectrum: CodeCharts and ImportAnnots. CodeCharts reflects common patterns in eye tracking data like center bias (the tendency of humans to gaze at the center of an image [102]), exploration (gaze points scattered throughout an image), and emphasis on faces, while ImportAnnots produces large regions of uniform importance that coincide with discrete objects (Figure 3-12 top). ImportAnnots and CodeCharts are most similar when the image contains a handful of objects that are both salient and segmentable (Figure 3-12 bottom).

On graphic designs (Figure 3-13), CodeCharts heatmaps have a strong center bias

Figure 3-13: Graphic designs where CodeCharts and ImportAnnots agree and differ. **Top:** The title is important but not salient. **Bottom:** The cat is salient and important, but saliency is concentrated at a point whereas importance segments the entire photo.

(probably because people do not have time to examine the details of the design), whereas ImportAnnots indicates that people find text to be important. Quantitatively, CodeCharts and ImportAnnots heatmaps are weakly correlated (CC of 0.413 for natural images, 0.491 for graphic designs). When using each to rank graphic design elements, the two interfaces achieve a Spearman's rank correlation of 0.509. An important object is not necessarily a salient one and vice versa.

3.4 Which interface should I use?

Table 3.2 contains a summary of the advantages of each of the four interfaces. ZoomMaps can collect attention data on detailed, multi-scale content via an intuitive interface, but it provides a coarse-grained approximation of attention and sometimes places outsized emphasis on smaller items. CodeCharts most accurately replaces eye tracking, is the only interface where stimuli exposure time is carefully controlled by the experimenter, and does not require image distortion, but it does require many participants and is relatively expensive. ImportAnnots provides high-fidelity element segmentations and emphasizes importance over saliency. BubbleView is versatile,

Interface	Use Case	Advantages	Drawbacks
Zoom Maps 	Capturing exploration of large images at multiple scales	Works on images with multi-scale content, natural form of interaction	Coarse approximation of attention
Code Charts 	Approximating eye-tracking, esp. for precise viewing durations	Doesn't distort stimuli, experimenter controls timing, fun	Little data per participant, images must fit on screen
Import Annots 	Comparing importance of graphic design elements	Produces clean segmentations, captures importance	Not ideal for natural images, measures importance over attention
Bubble View 	Approximating eye-tracking, esp. during description tasks	Versatile, cheap	Distorts stimuli and viewing experience

Table 3.2: Summary of use cases and trade-offs for the TurkEyes interfaces.

cheap, and a reasonable approximation of eye data, but it distorts the underlying image and slows down the viewing process. The best interface depends on the use case, stimuli, and type of data desired.

By providing a set of scalable and versatile attention-capturing interfaces, TurkEyes makes attention data an accessible tool for researchers and creators who want to better understand how humans respond to visual content. This lays the groundwork for future work in exploring different image types, viewing tasks, and applications of attention. For example, as we will show in Chapter 4, computational models trained on cheap, scalably crowdsourced attention data can help machines understand images the way humans do.

Chapter 4

Modeling multi-duration saliency

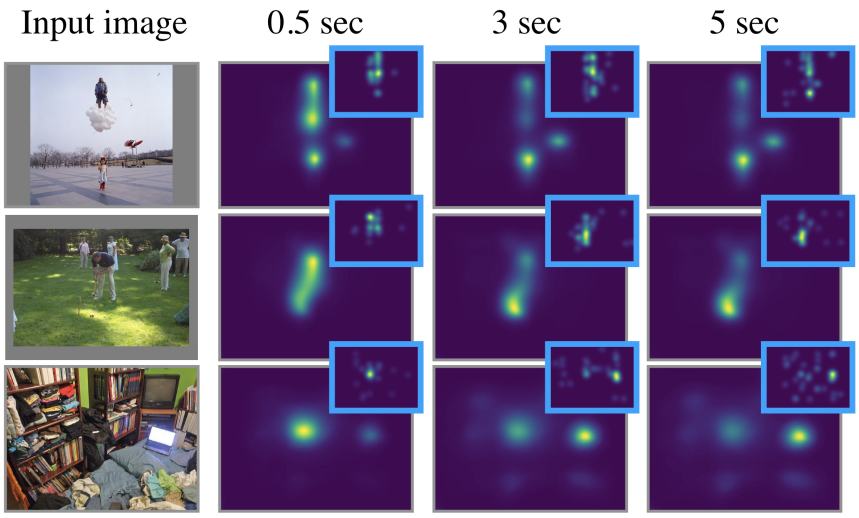

Figure 4-1: Predictions of our Multi-Duration Saliency Excited Model at three viewing durations. Images are from the Abnormal Objects [87], SALICON [62], and EyeCrowd [63] datasets (top to bottom). Insets with blue borders contain human ground truth collected using the CodeCharts UI.

What jumps out in a single glance of an image is different than what you might notice after closer inspection. Yet conventional models of visual saliency, like those discussed in Section 2.2, produce predictions at an arbitrary, fixed viewing duration, offering a limited view of the rich interactions between image content and gaze location. In this chapter, we propose to capture gaze as a series of snapshots, by generating population-level saliency heatmaps for multiple viewing durations (Section 4.1). In Section 4.2, we use the CodeCharts interface from Chapter 3 to collect

the CodeCharts1K dataset, which contains multiple distinct heatmaps per image corresponding to 0.5, 3, and 5 seconds of free-viewing. In Section 4.3, we develop an LSTM-based model of saliency that simultaneously predicts saliency heatmaps for multiple viewing durations (Figure 4-1). Our Multi-Duration Saliency Excited Model (MD-SEM) achieves competitive performance on the LSUN 2017 Challenge with 57% fewer parameters than comparable architectures. In Section 4.4, we show how generating heatmaps at multiple viewing durations can enable applications where multi-duration saliency can be used to prioritize visual content to keep, transmit, and render[1].

4.1 Problem statement

In this chapter, we introduce the concept of **multi-duration saliency**, which captures multiple attention snapshots corresponding to different viewing durations[2]. This offers richer insight into how gaze evolves over time than conventional saliency, while providing a more robust representation than scanpaths.

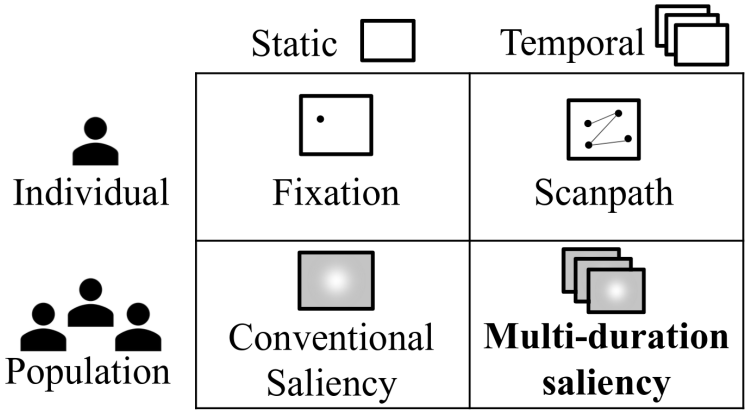

Figure 4-2: Multi-duration saliency compared to other gaze prediction tasks: combining the stability and generalizability of a population-level metric with rich temporal data.

Multi-duration saliency maintains the robustness of population-level saliency modeling and the temporal resolution of scanpaths. We introduce multi-duration saliency

[1]For a full discussion of multi-duration saliency, see [40]

[2]Data, code, and models available at: `http://multiduration-saliency.csail.mit.edu/`

as a way to snapshot attention at a few distinct time points. Unlike conventional saliency, multi-duration saliency accounts for the effect of viewing duration on gaze patterns and provides insight into how attention evolves over time. However, in contrast to scanpath prediction, multi-duration saliency is a population-level metric that produces stable, interpretable, and generalizable attention heatmaps. This framing addresses questions like: what content do people prioritize, and what is initially attention grabbing versus noticeable only after seconds of viewing?

4.2 The CodeCharts1k dataset

Using the CodeCharts UI (presented in Section 3.2.2), we collect the first multi-duration saliency dataset, containing 1000 images with saliency heatmaps corresponding to 0.5, 3, and 5 second of viewing.

4.2.1 Data collection

CodeCharts is an appropriate attention-capturing tool for multi-duration attention because it is a good approximation for eye tracking (see Table. 3.1) and allows for fine-grained control of image presentation time.

Experiment procedure. Our task sequence includes 6 practice images to prescreen for attentiveness, 50 dataset images, and 5 validation trials spaced throughout the sequence. Validation trials consist of a cropped human face on a plain background, where participants are expected to enter a code that overaps with the face. To ensure data quality, we filter out participants who enter nonexistent codes, fail over 25% of validation images, or look at the same spot repeatedly. We collect 50 gaze points per image per viewing duration, which produces on average 44 valid gaze points after filtering. We blur all gaze points (with a Gaussian sigma of 50 pixels) to produce an attention heatmap. A pilot experiment on the OSIE dataset [109] showed that attention heatmaps at 0.5, 3, and 5 seconds were most distinct from each other, so we collected CodeCharts1k data at these durations. We used Amazon's Mechanical Turk and paid participants at an hourly rate of $10. Data collection cost $4.90 per

image for 150 unique gaze points.

Images. Our collected dataset contains a variety of image types to provide a broad picture of differences in attention over time. We used 500 images from SALICON [62], 130 from LaMem [67], 120 from CAT2000 [11][3], 100 from EyeCrowd [63], 100 from a mix of Abnormal Objects [87] and Out-of-Context Objects [27], and 50 from the Stanford 40K Actions dataset [111][4].

4.2.2 Is saliency predictable at multiple durations?

To measure whether gaze patterns across participants are consistent for a given viewing duration, we perform a split-half consistency analysis on the CodeCharts1K data. We divide participants into two groups, generate a heatmap from each group's gaze points, and compute Pearson's Correlation Coefficient (CC) between the heatmaps. We repeat this computation over 10 splits of participant data and average the scores. To measure whether the gaze patterns vary systematically across durations, we select participants from different viewing duration conditions.

Saliency is predictable across viewing durations: The split-half consistency between participants is high across all durations (CC=.76 at 0.5 sec, CC=.68 at 3 sec, CC=.67 at 5 sec). While the highest consistency occurs at the briefest duration [15, 104], consistency remains high across the longer viewing durations.

Different things are salient at different durations: When there are differences in what is salient at different durations, CC scores between participants viewing an image at the same duration are higher than CC scores between participants viewing an image at different durations. Gaze patterns are different between .5 and 3 sec for 51% of images from CodeCharts1K; 55% of images show differences between .5 and 5 sec, and 27% of images show differences between 3 and 5 sec.

These analyses indicate that gaze data collected using the CodeCharts UI contains a consistent signal at each of the viewing durations and the signal differs between

[3]Using 100 "Action" [111] and 20 "Low Resolution" [64] images.

[4]We used action classes that explicitly contained an interaction of a person and an object, by selecting 10 images each of: shooting an arrow, throwing a frisby, walking the dog, writing on a board, writing on a book.

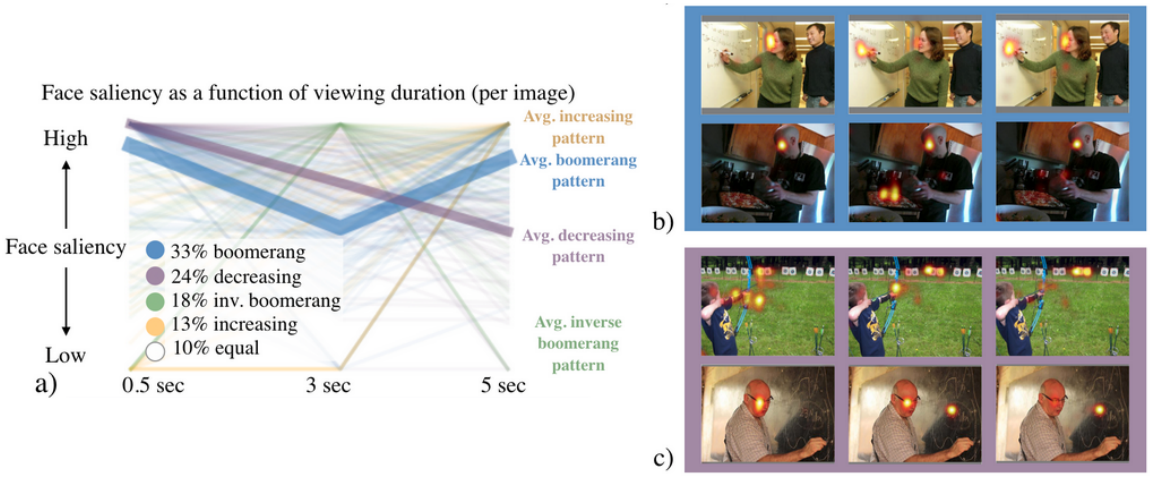

Figure 4-3: Dominant patterns of human gaze on faces across time. (a) Individual lines plot how the saliency of faces within an image varies across viewing durations. Thicker lines (labeled) are averages over the dominant patterns. We include the percent of images that follow each pattern. (b) Examples where face saliency decreases from 0.5 to 3 sec, increasing again from 3 to 5 sec ("boomerang"). (c) Examples where face saliency decreases from 0.5 to 5 sec.

viewing durations. This suggests that saliency is predictable at different viewing durations, setting the stage for the computational model in Sec. 4.3.

4.2.3 What is salient when?

Things and stuff: We used COCO segmentation maps [21] of the SALICON images to compute gaze counts per object class over time. From 0.5 to 3 seconds, gaze frequently moves away from people and towards objects and furniture (e.g., paper, bottle, table). From 3 to 5 seconds, there is an increase in attention on "stuff" (like grass, carpet, and road) that may contain other objects. At these longer durations people gaze more at small and distant objects.

Faces: We know that gaze is attracted by faces [20, 23]. For a finer-grained analysis, we ran a face detection network [42] over images in CodeCharts1K. Across the 266 images where faces were detected, we computed a measure of face saliency at different durations. At each duration, we counted all the gaze points that land on a face region and normalized by the number of gaze points per image across all

3 durations, so face saliency ranges between 0 and 1. Figure 4-3a plots face saliency as a function of viewing duration for each image. Across CodeCharts1K, we find a dominant "boomerang" pattern (found in 33% of images with faces): people notice faces at 0.5 sec, their gaze shifts elsewhere at 3 sec, and returns to faces at 5 sec. The second most prevalent pattern is a decrease in gaze on faces over time (24%). Other patterns, like an increase in face saliency over time, were in the minority. These observations are consistent with the phenomenon known as inhibition of return (IOR) [58, 89], or the relative suppression of visual cues that were recently attended to. Samuel and Kat [97] found that IOR lasts for approximately 3 seconds, which might explain why attention tends to shift away from faces between 0.5 and 3 sec but often returns to faces at 5 sec.

Qualitatively, human gaze frequently moves from the actor (at 0.5 sec) to the action (at 3 and 5 sec). Sometimes this shift in attention is gradual: saliency at 3 sec is a combination of saliency at 0.5 and 5 sec (Figure 4-3c). In other cases, saliency at 5 sec is more similar to that at 0.5 sec; in these cases it seems that people explore an image before returning to the most interesting regions (Figure 4-3b).

4.3 Modeling multi-duration saliency

To efficiently and accurately predict multiple saliency maps for a single image, we introduce the Multi-Duration Saliency Excited Model (MD-SEM), a novel architecture designed for multi-duration saliency. MD-SEM is the first model that outputs multiple saliency maps corresponding to different viewing durations. The core of our model is a Temporal Excitation Module (TEM) that applies a time-based re-weighting to saliency feature maps with a minimal increase in parameters. We also design a new loss, the Correlation Coefficient Match (CCM) loss, that encourages the network to capture temporal patterns.

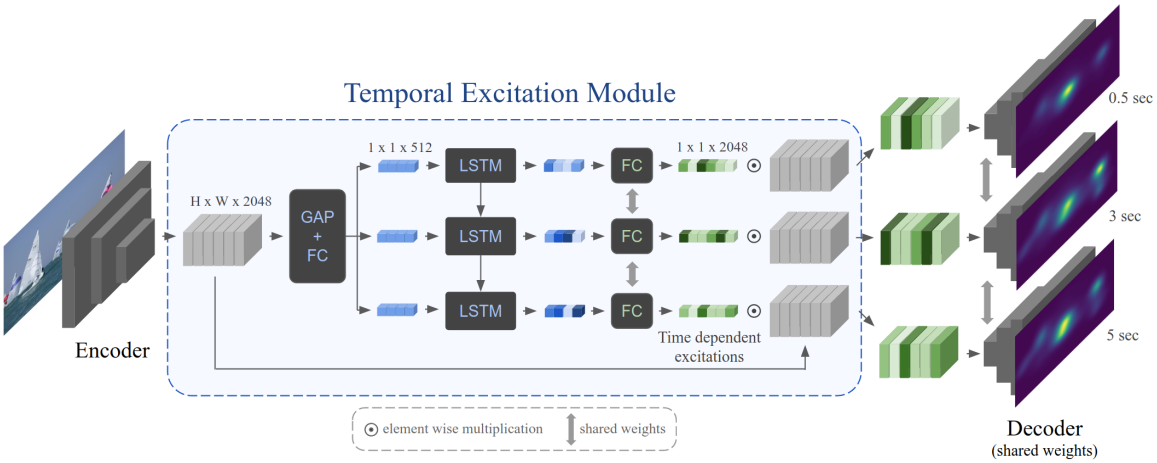

Figure 4-4: Architecture of the Multi-Duration Saliency Excited Model (MD-SEM). The encoder outputs compressed feature maps that are fed to the Temporal Excitation Module (TEM). In order to predict saliency across durations, TEM uses LSTM cells to generate scaling vectors that re-weight the feature maps differently for each duration. The modified feature maps are then decoded into saliency heatmaps. Reused features and shared weights keep the architecture lightweight.

4.3.1 Encoder-decoder architecture

The architecture of MD-SEM is shown in Figure 4-4. We design an accurate model of reduced size and complexity by distilling the required components to a minimum: (1) a strong encoder [28], (2) a temporal processing module that operates on a compressed representation, and (3) a simple regularized decoder. We use a state-of-the-art Xception network [28] pretrained on ImageNet as the backbone. For the decoder, our experiments showed that a simple module composed of 3 sets of convolution, up-sampling and dropout layers are sufficient for this task. This choice of module reduces model complexity and implicitly regularizes the network.

4.3.2 Temporal Excitation Module

To predict multi-duration saliency, we introduce a module that recursively manipulates the feature representation generated by the encoder to adapt it for each duration. Our module uses a Long Short Term Memory (LSTM) network to generate scaling vectors that re-weight the feature maps differently for each of T timesteps (where

$T = 3$ in our implementation). Feature map re-weighting has been explored in the form of Squeeze and Excitation Modules [50], but has not been exploited as a temporal modification tool.

The architecture of the Temporal Excitation Module (TEM) is shown in Figure 4-4. First, the feature maps generated by the encoder are pooled through global average pooling and passed through a fully connected layer, which reduces the dimensionality of the feature vector and aids in generalization. The output of the dense layer is replicated T times and fed as a sequence to the LSTM. The LSTM then outputs T vectors, which contain information specific to each timestep and will be used to rescale each feature map differently. These vectors are passed through a fully-connected layer that increases their dimensionality to match the channel dimension of the feature maps (C), yielding scaling vectors $s^{(t)}$ of length C. A sigmoid non-linearity ensures that the scaling weights remain within a sensible range. Finally, the block outputs a set of T feature maps, which are obtained by rescaling the original feature maps according to each of the T vectors s. Formally, the module outputs T sets of C feature maps, where each feature map $f_c^{(t)}$ is computed as:

$$f_c^{(t)} = I_c \cdot s_c^{(t)},$$

where I_c is the c-th input feature map and $s_c^{(t)}$ is the scaling weight for duration t and channel c.

Efficiency: TEM is designed to be lightweight. TEM's LSTM operates over a squeezed, low-dimensional vector obtained from pooling input feature maps. By contrast, SAM [33], a top-performing saliency model, uses an LSTM for internal map refinement that operates on full 3D feature maps. Our approach results in an architecture with 30 million parameters, 57% smaller than SAM [33]. In Sec. 4.3.4, we show that our architecture outperforms SAM.

4.3.3 Correlation Coefficient Match Loss

To ensure that our network correctly captures differences across viewing durations, we introduce a novel training loss called Correlation Coefficient Match (CCM) loss. This loss forces the network to output saliency maps that reproduce the correlations between ground truth saliency maps at adjacent durations. If ground truth maps at durations t and $t+1$ are dissimilar, we encourage the network to produce equally dissimilar maps at these durations. Given a set of T viewing durations for which we want to predict saliency maps, we calculate the CCM loss by computing Pearson's Correlation Coefficient (CC) on pairs of saliency maps at adjacent durations, then computing the difference between the ground truth and predicted scores. CC is defined as: $CC(y_1, y_2) = \frac{\sigma(y_1,y_2)}{\sigma(y_1)\cdot\sigma(y_2)}$, where $\sigma(y_1, y_2)$ is the covariance of y_1 and y_2. If we let $y^{(t)}$ be the heatmap corresponding to duration t, our CCM loss is:

$$L_{CCM}(y_g, y_p) = \frac{1}{T-1}\sum_{t=0}^{T-1}\left|CC\left(y_g^{(t)}, y_g^{(t+1)}\right) - CC\left(y_p^{(t)}, y_p^{(t+1)}\right)\right|$$

where $y_g^{(t)}$ and $y_p^{(t)}$ are the ground truth and predicted saliency maps for duration t, respectively.

This novel loss boosts performance on multi-duration saliency prediction, increasing the NSS score of MD-SEM by nearly 5% on CodeCharts1K (Table 4.1).

Model	*NSS* ↑	*CC* ↑	*KL* ↓	*SIM* ↑	*CCM* ↓
SAM-MD w/o CCM	2.700	0.744	0.434	0.616	0.231
SAM-MD w/ CCM	2.739	0.753	0.458	0.609	0.198
MD-SEM w/o CCM	2.778	0.754	0.565	0.598	0.228
MD-SEM w/ CCM	**2.915**	**0.765**	**0.430**	**0.620**	**0.195**

Table 4.1: MD-SEM results on CodeCharts1K with and without CCM loss. We report performance on NSS, CC, KL, SIM and our custom CCM loss. These results correspond to the average over all durations.

4.3.4 Model evaluation

Datasets. For training, we use the SALICON-MD (Multi-Duration) and CodeCharts1K datasets. We created SALICON-MD from the original SALICON dataset [62] by temporally bucketing each participant's data. Since no timestamps were provided, we assumed an even distribution across the viewing duration (from 0 to 5 seconds) and split the attention locations into 6 buckets. This time-bucketed data serves as an approximate but large pretraining dataset. For final training and evaluation, we use ground truth multi-duration data from CodeCharts1K (Sec. 4.2).

Multi-duration evaluation. Our model is first-of-its-kind in its ability to predict saliency at multiple durations. To demonstrate the superiority of our model over existing single-duration models, we compare to a baseline that represents the best alternative for obtaining multiple distinct saliency heatmaps: training multiple copies of a state-of-the-art architecture on the ground truth for 3 different durations. We call this approach *SAM×3*. Next, to demonstrate the advantages of our particular architecture, we benchmark against *SAM-MD*, a modified, multi-duration version of SAM where the LSTM is modified to produce a different saliency map at each timestep. Each output map corresponds to a different viewing duration and the network trains on all three durations simultaneously. The results of these comparisons are shown in Table 4.2. MD-SEM is better at approximating human gaze and differentiating across durations, while using many fewer parameters than the other models.

		All durations		
Model	*Params* ↓	*CC* ↑	*NSS* ↑	*KL* ↓
SAM ×3	210.3M	0.734	2.708	0.483
SAM-MD	70.1M	0.753	2.739	0.458
MD-SEM	**30.9M**	**0.765**	**2.915**	**0.430**

Table 4.2: Comparison of multi-duration saliency models evaluated on CodeCharts1K. Baselines are SAM×3 (three copies of SAM, each trained exclusively on data for one duration) and SAM-MD (a custom modification of SAM whose LSTM outputs multiple maps). MD-SEM (ours) excels while using substantially fewer parameters. In fact, MD-SEM outperforms the other models across all three durations in CodeCharts1k; see Appendix A for performance broken out by duration.

Model	*NSS* ↑	*CC* ↑	*KL* ↓	*SIM* ↑
SAM-res [33]	1.990	**0.899**	0.610	**0.793**
EML-Net [61]	2.050	0.886	**0.520**	0.780
SalNet [85]	1.859	0.622	-	-
CEDNS	2.045	0.862	1.026	0.753
MD-SEM (Ours)	**2.058**	0.868	0.568	0.774

Table 4.3: Comparison to state-of-the-art on SALICON test set (LSUN 2017 Challenge).

Single-duration evaluation. We also evaluated our architecture on the conventional single-duration saliency task and obtained performance competitive with state-of-the-art saliency models (Table 4.3). MD-SEM currently achieves a second-place NSS score on the LSUN 2017 challenge [1].

Qualitative evaluation. Qualitatively, our model accurately reproduces many of the dominant human gaze patterns from the CodeCharts1K dataset, such as the tendency of humans to focus on the object of an action at longer viewing durations, and for attention to shift from the center of the image to smaller details and secondary objects (Figure 4-5).

4.4 Applications of multi-duration saliency

Saliency models have proven useful for many image processing applications, including smart cropping, retargeting, and image captioning. Our multi-duration saliency model can contribute additional context by accounting for the expected time that a viewer may have to explore an image. Below, we indicate some future directions for how multi-duration saliency can benefit traditional saliency-based applications. (For more examples of application results, see Appendix B).

Cropping. Automatic image cropping is useful for thumbnailing, view-finding for improved composition, and retargeting for different use cases [37]. Multi-duration saliency allows us to additionally take into account the expected time a viewer will spend on an image (e.g., an image that is part of a passing advertisement should

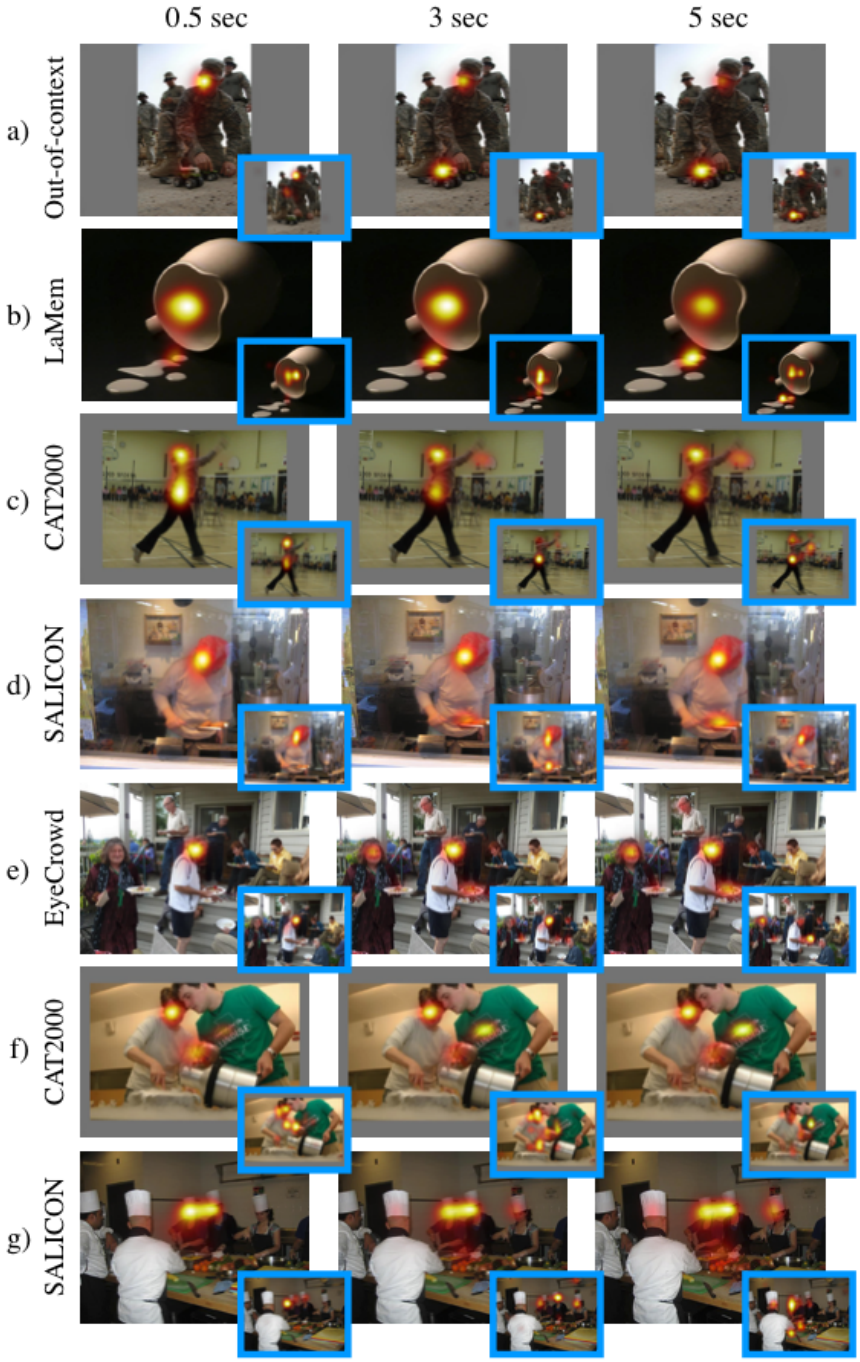

Figure 4-5: MD-SEM predictions on various datasets. Insets contain human ground truth from CodeCharts1K. Our model approximates human attention by shifting saliency from faces to objects of action across time (a,c,d) and shifting the center of focus from the center of the image to secondary image regions at longer viewing durations (b,e). Difficult cases for our model include cluttered scenes with many objects, people, or complex actions (f,g). More results in Appendix B.

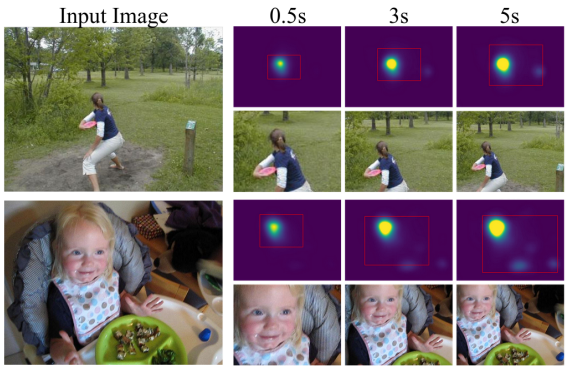

Figure 4-6: **Cropping.** Images automatically cropped based on cumulative viewing duration by selecting the window with 90% of the most salient image regions as predicted by our model. Image crops for shorter viewing durations contain close-ups of key elements.

Figure 4-7: **Compression.** We visualize instance detections that are predicted to attract gaze at different viewing durations (accumulated over time). Content that is salient at short durations could be rendered before content that becomes salient later.

probably contain fewer salient elements than if it is the main image on a page). In Figure 4-6 we use our multi-duration saliency maps to crop windows that capture 90% of the heatmap density that occurs at or below a particular viewing duration [25]. Our automatically-generated thumbnails contain close-ups of the most important objects at shorter viewing durations.

Compression and rendering. Multi-duration saliency heatmaps can indicate the order in which items in a scene should be rendered to provide a seamless user experience. In Figure 4-7 we visualize which elements would be prioritized at different viewing durations. To generate these visualizations, we used Mask R-CNN for instance segmentation [49]. We accumulated saliency heatmap density for each instance across time to determine which instances to prioritize. Instances with a mean saliency score in the 90th percentile were kept and the rest of the image was blurred and darkened for visualization purposes.

Captioning. Image captions can facilitate search and improve accessibility. Some recent work attempts to use a saliency map to guide attention for captioning [32]. In Figure 4-8, we used our saliency predictions to focus an image captioning model [92] on regions that stand out at different viewing durations. This can produce varied

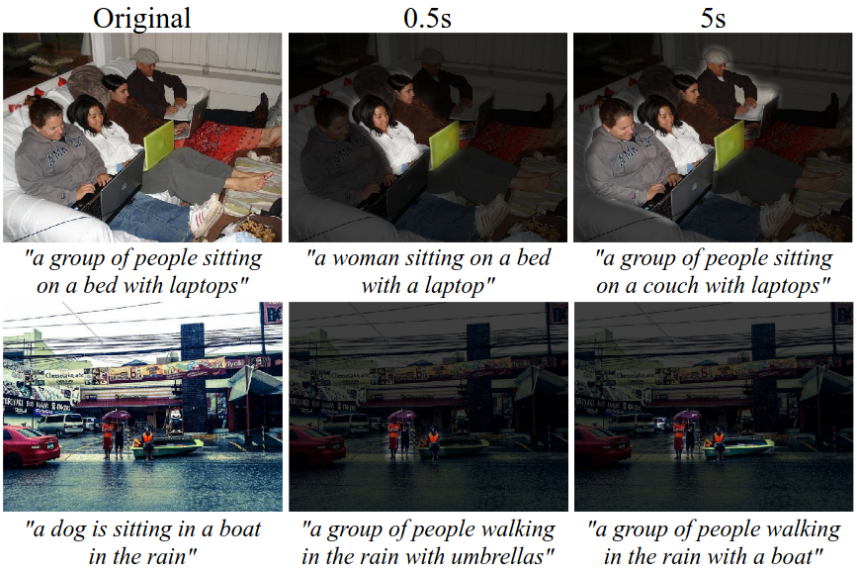

Figure 4-8: **Captioning.** Captions generated by passing saliency-enhanced images to an image captioning model [92], using saliency at different viewing durations to prioritize image content.

captions that focus on different aspects of the image.

Impact. In this chapter, guided by the insight that where you focus on an image depends on how much time you have to explore it, we tackled the problem of predicting multi-duration saliency: saliency as a function of viewing duration. We propose a scalable, crowdsourceable experiment for gathering ground truth multi-duration saliency data and use it to collect the CodeCharts1K dataset. Our LSTM-based saliency model is a top performer at predicting conventional saliency while also providing predictions at multiple durations. Finally, we showed how multi-duration saliency could be used to add temporal context to applications which require prioritizing visual content.

Chapter 5

Modeling video memorability

In Chapter 3, we discussed designing crowdsourceable interfaces to collect cognitive data. In Chapter 4, we talked about collecting a dataset using these interfaces and building a neural network model to predict human responses. In this chapter, we tie together these themes in order to collect, analyze, and predict the memorability of videos.

Deciding which events from past experience to forget and which must be remembered is a key capability of an intelligent system. Towards this goal, we develop a predictive model of human visual event memory and how those memories decay over time. Through the Memento Game, an online memory game (Section 5.2), we collect *Memento10k*, a new, dynamic video memorability dataset containing human annotations at different viewing delays (Section 5.3). Based on our findings we propose a new mathematical formulation of memorability decay (Section 5.4), resulting in a model that is able to produce the first quantitative estimation of how a video decays in memory over time (Section 5.5). In contrast with previous work, our model can predict the probability that a video will be remembered at an arbitrary delay. Importantly, our approach is multimodal, combining visual and semantic information to fully represent the meaning of events. Our experiments on two video memorability benchmarks, including Memento10k, show that our model significantly improves upon the best prior approach (by 15% on average).

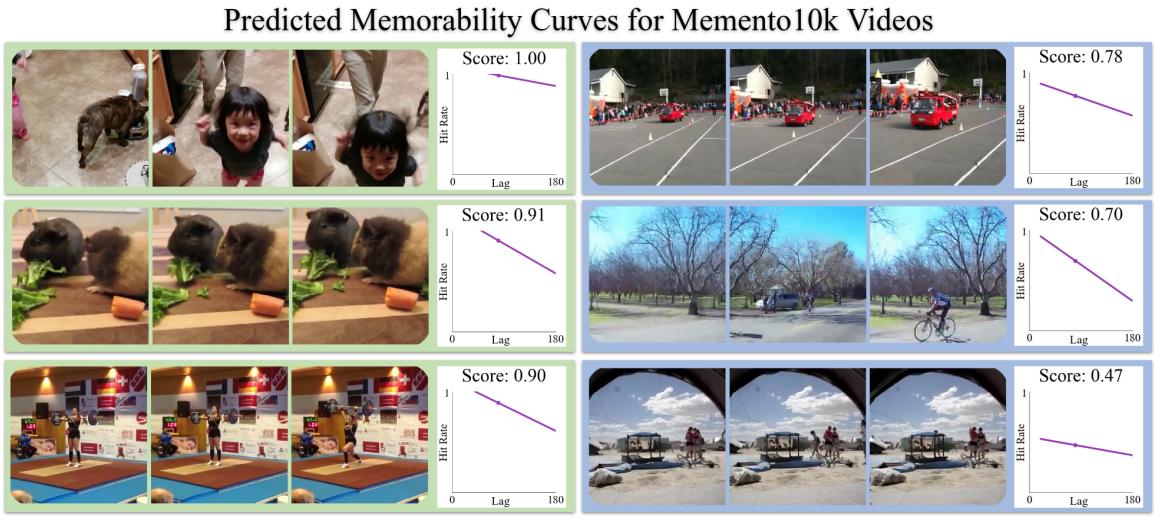

Figure 5-1: **How do visual and semantic features impact memory decay over time?** We introduce a multimodal model, SemanticMemNet, that leverages visual and textual information to predict the memorability decay curve of a short video. We show predictions on videos from Memento10k, our new video memorability dataset. SemanticMemNet is the first model to predict a decay curve that represents how quickly a video falls off in memory over time.

5.1 Problem statement

Memorability of dynamic events is challenging to predict because it depends on many factors. First, different visual representations are forgotten at different rates: while some events persist in memory even over long periods, others are forgotten within minutes [4, 44, 55, 72]. This means that the probability that someone will remember a certain event varies dramatically as a function of time, introducing challenges in terms of how memorability is represented and measured. Second, memorability depends on both visual and semantic factors. In human cognition, language and vision often act in concert for remembering an event. Events described with richer and more distinctive concepts are remembered for longer than events attached to shallower descriptions, and certain semantic categories of objects or places are more memorable than others [55, 71].

In this chapter, we introduce a new dataset and a model for video memorability prediction that address these challenges[1]. Memento10k, the most dynamic video

[1]Data, model, and demo at: `http://memento.csail.mit.edu/`

memorability dataset to date, contains both human annotations at different viewing delays and written captions, making it ideal for studying the effects of delay and semantics on memorability. Based on this data, we train SemanticMemNet, a multi-modal model for predicting the decay in memory of a short video clip (Figure 5-1). SemanticMemNet is the first model that predicts the entire memorability decay curve, which allows us to estimate the probability that a person will recall a given video after a certain delay. We also enhance our model's features to include information about video semantics by jointly predicting verbal captions.

5.2 Memento: The Memory Game

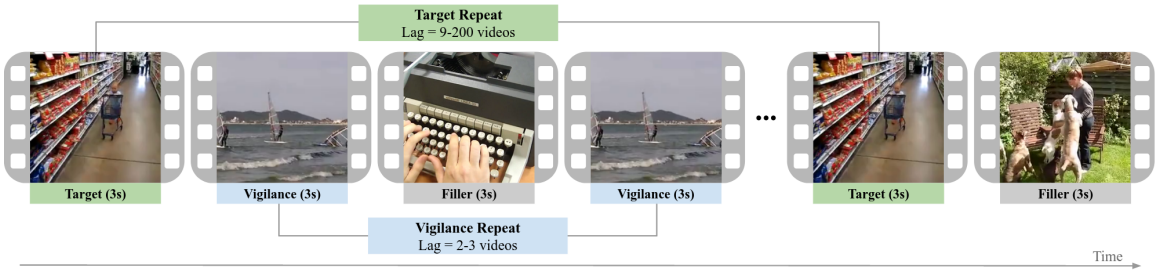

Figure 5-2: Task flow diagram of *Memento: The Video Memory Game*. Participants see a continuous stream of videos and press the space bar when they see a repeat.

We designed a memory game to collect memorability data for videos, inspired by the old-new recognition paradigms used in previous large scale experiments [56, 67]. In *Memento: The Video Memory Game*, crowdworkers from Amazon's Mechanical Turk watched a continuous stream of three-second video clips and were asked to press the space bar when they saw a repeated video. Unlike [56, 67], where each image is separated by a blank screen, our videos are shown back-to-back like a "movie trailer" to keep the pace engaging and game-like. When participants press the space bar, they receive feedback in the form of a red flash for an incorrect response or a green flash for a correct response. The task flow diagram of our experiment is in Figure 5-2. When a worker correctly identifies a repeat, that is known as a "hit" and the stream skips ahead to the next video; there is no feedback for missed repeats.

Each level of the memory game contains on average 204 videos (with repeats) and lasts around 9 minutes. The number of intervening videos between the first and second occurrence of a repeated video is known as the "lag". The game consists of "vigilance" repeats that occur at short lags of 2-3 videos and are used to filter out inattentive workers and "target" repeats at lags of 9-200 videos that provide memorability data. In order to ensure high-quality annotations, we invalidate a level of the game if a participant's vigilance accuracy is below 80%, if their false positive rate is above 50%, or if the participant fails a quality check early in the level. These checks discarded around 15% of levels started.

Target repeats, which comprise around 20% of the video presentations in each level, form the core of our dataset. A target video's "hit rate" is the fraction of people who correctly identified it as a repeat. This wide range in target repeat lags allows us to measure how a video's hit rate changes as a function of lag. After an initial burn-in period, target repeats occur with an approximately uniform probability (25%) at each position. This uniformity is important so that participants cannot infer when the next repeat will come. All target repeats occur within one level of the game and videos are not reused across levels.

5.3 Memento10k: a multimodal memorability dataset

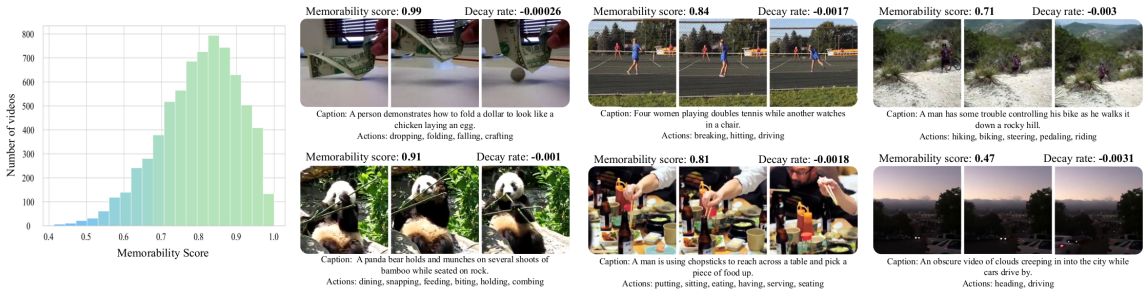

Figure 5-3: **The Memento10k dataset** contains the memorability scores, alpha scores (decay rates), action labels, and five unique captions for each of the 10,000 videos. **Left:** The distribution of memorability scores over the entire dataset. **Right:** Example videos arranged from high memorability to low memorability along with their memorability score, decay rate, actions, and an example caption.

5.3.1 Dataset contents

Dynamic, In-The-Wild Videos. The Memento10k Dataset (Figure 5-3) is composed of natural videos scraped from the Internet and cropped to 3-second segments.[2] We first asked crowdworkers whether each clip was a "homemade video" and discarded videos that did not meet this criterion in order to limit our clips to non-artificial scenes with an everyday context. After removing videos that contained undesirable properties (i.e. watermarks), the dataset resulted in 10,000 "clean" videos, which we break into train (7000), validation (1500), and test (1500) sets.

The Memento10k dataset represents a significant step towards understanding memorability of real-world events. It is the most dynamic memorability dataset to date with videos containing a variety of motion patterns encompassing camera motion and moving objects. The mean magnitude of optical flow in Memento10k is nearly double that of VideoMem [29] (15.476 vs. 7.296), whose clips tend to be fairly static. Memento10k's diverse, natural content enables the study of memorability in a dynamic everyday context, and its annotations spread over lags of 30 seconds to 10 minutes allows for a robust estimation of a video's decay rate.

Semantic Annotations. We augment our dataset with captions, providing a source of rich textual data that we can use to relate memorability to semantic concepts (examples in Figure 5-3). We asked crowdworkers to describe the events in the video clip in full sentences and we manually vetted the captions for quality and corrected spelling mistakes. Each video has 5 unique captions from different crowdworkers.

5.3.2 Human results

We measured human consistency for the Memento10k dataset following [29, 56]: we randomly split our participant pool into two groups and calculate the Spearman's rank correlation between the memorability rankings produced by each group, where the rankings are generated by sorting videos by raw hit rate. The average rank correlation (ρ) over 25 random splits is 0.73 (compared to 0.68 for images in [67], and

[2]The Memento videos have partial overlap with the Moments in Time [81] dataset.

Figure 5-4: **Examples of high and low memorability videos.** Video clips involving people, faces, hands, man-made spaces, and moving objects are in general more memorable, while clips containing distant/outdoor landscapes or dark, cluttered, or static content tend to be less memorable.

0.616 for videos in [29]). This high consistency is partially owing to the high number of annotations (90+ per video) in Memento10k. However, the high consistency between human observers confirms that videos have strong intrinsic visual, dynamic or semantic features that a model can learn from to predict memorability of new videos.

Figure 5-4 illustrates some qualitative results of our experiment. We see similar patterns as in image memorability in terms of what visual content makes an event memorable: memorable videos tend to contain saturated colors, people and faces, manipulable objects, and man-made spaces, while less memorable videos are dark, cluttered, or inanimate. Additionally, videos with interesting motion patterns can be highly memorable whereas static videos often have low memorability.

5.4 A mathematical model of memorability decay

Most memories decay over time. In psychology this is known as the forgetting curve, which estimates how the memory of an item naturally degrades over time. Because Memento10k's memorability annotations occur at lags of anywhere from 9 videos (less than 30 seconds) to 200 videos (around 9 minutes), we have the opportunity to calculate the strength of a given video clip's memory at different lags.

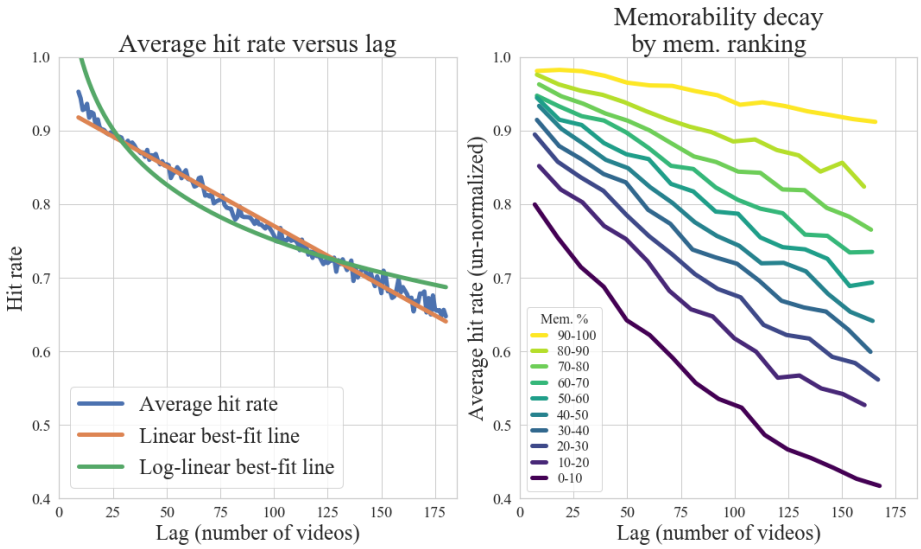

Figure 5-5: Our data suggests a **memory model where each video decays linearly** in memory according to an individual decay rate $\alpha^{(v)}$. **Left:** A linear trend is a better approximation for our raw data ($r = -0.993$) than a log-linear trend ($r = -0.963$). **Right:** We confirm our assumption that α varies by video by grouping videos into deciles based on their normalized memorability score and plotting group average hit rate as a function of lag. Videos with lower memorability clearly have a faster rate of decay.

A naive method for calculating a memorability score is to simply take the video's target hit rate, or the fraction of times that the repeated video was correctly detected. However, since we expect a video's hit rate to go down with time, annotations at different lags are not directly comparable. Instead, we derive an equation for how each video's hit rate declines as a function of lag.

First, for the lags tested in our study, we observe that hit rate decays linearly as a function of lag. This is notable because previous work on image memorability has found that images follow a log-linear decay curve [44, 55, 106]. Figure 5-5 (left) shows that a linear trend best fits our raw annotations.

Second, in contrast to prior work, we find that different videos decay in memory at different rates. Instead of assuming that all stimuli decay at one universal decay rate, α, as in [67], we assume that each video decays at its own rate, $\alpha^{(v)}$. Following the procedure laid out in [67], we find a memorability score and decay rate for each video that approximates our annotations. We define the memorability of video v as

$m_T^{(v)} = \alpha^{(v)}T + c^{(v)}$, where T is the lag (the interval in videos between the first and second presentation) and $c^{(v)}$ is the base memorability of the video. If we know $m_T^{(v)}$ and $\alpha^{(v)}$, we can then calculate the video's memorability at a different lag t with $m_t^{(v)} = m_T^{(v)} + \alpha^{(v)}(t - T)$ (5.1).

To obtain values for $m_T^{(v)}$ and $\alpha^{(v)}$, we minimize the L2 norm between the raw binary annotations from our experiment $x_j^{(v)}$, $j \in \{0, ..., n^{(v)}\}$ and the predicted memorability score at the corresponding lag, $m_t^{(v)}$. The error equation is:

$$E(\alpha^{(v)}, m_T^{(v)}) = \sum_{j=1}^{n^{(v)}} x_j^{(v)} - {m_t^{(v)}}_2^2 = \sum_{j=1}^{n^{(v)}} x_j^{(v)} - \left[m_T^{(v)} + \alpha^{(v)}(t_j^{(v)} - T)\right]_2^2 \tag{5.2}$$

We find update equations for $m_T^{(v)}$ and $\alpha^{(v)}$ by taking the derivative with respect to each and setting it to zero:

$$\alpha^{(v)} \leftarrow \frac{\frac{1}{n^{(v)}} \sum_{j=1}^{n^{(v)}} (t_j^{(v)} - T) \left[x_j^{(v)} - m_T^{(v)}\right]}{\frac{1}{n^{(v)}} \sum_{j=1}^{n^{(v)}} \left[t_j^{(v)} - T\right]^2} \qquad m_T^{(v)} \leftarrow \frac{1}{n^{(v)}} \sum_{j=1}^{n^{(v)}} \left[x_j^{(v)} - \alpha^{(v)}(t_j^{(v)} - T)\right] \tag{5.3}$$

We initialize $\alpha^{(v)}$ to -5e^{-4} and $m_T^{(v)}$ to each video's mean hit rate. We set our base lag T to 80 and optimize for 15 iterations to produce $\alpha^{(v)}$ and $m_{80}^{(v)}$ for each video. We thus define a video's "memorability score", for the purposes of memorability ranking, as its hit rate approximated at a lag of 80; however, we can use equation 5.1 to calculate its hit rate at an arbitrary lag within the range that we studied.

Next, we validate our hypothesis that videos decay in memory at different rates. We bucket the Memento10k videos into 10 groups based on their normalized memorability scores and plot the raw data (average hit rate as a function of lag) for each group. Figure 5-5 (right) confirms that different videos decay at different rates.

	Approach	RC - Memento10k (test set)	RC - VideoMem (validation set)
	Human consistency	0.730	0.616
(Sec. 5.5.1)	Flow stream only	0.580	0.425
	Frames stream only	0.601	0.527
	Video stream only	0.618	0.492
	Flow + Frames + Video	0.659	0.555
(Sec. 5.5.2)	Video stream + captions	0.624	0.512
	Video stream + triplet loss	0.614	-
	SemanticMemNet (ours)	**0.664**	**0.556**

Table 5.1: **SemanticMemNet ablation study.** We experiment with different ways of incorporating visual and semantic features into memorability prediction. We measure performance by calculating the Spearman's rank correlation (RC) of the predicted memorability rankings with ground truth rankings on both Memento10k and VideoMem.

5.5 Modeling memorability

In this section, we explore different architecture choices for modeling video memorability that take into consideration both visual and semantic features.

5.5.1 Modeling visual features

Baseline: Static Frames. We evaluate the extent to which static visual features contribute to video memorability by training a network to predict a video's memorability from a single frame. We first train an ImageNet-pretrained DenseNet-121 to predict image memorability by training on the LaMem dataset [67], then finetune on the video datasets. At test time, the video memorability score is calculated by averaging predictions over every 4th frame.

3D architectures: video and optical flow. Training on the RGB videos allows the network to access information on both motion and visual features, while training on optical flow lets us isolate the effects of motion. We train I3D architectures [22] on raw video and optical flow (computed using OpenCV's TV-L1 implementation). Our models were pretrained on the ImageNet and Kinetics datasets. We test the different

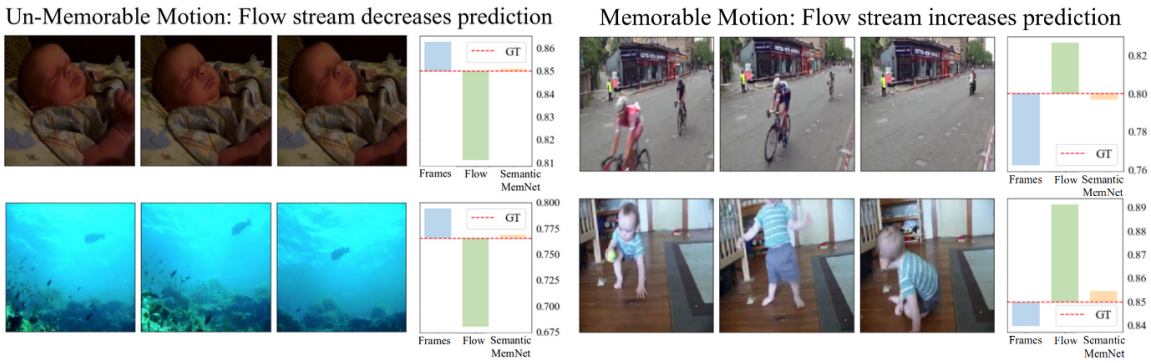

Figure 5-6: **Our model leverages visual and motion information to produce accurate memorability scores.** Here, we compare the contributions of the frames stream (static information only) and the optical flow stream (motion information only) to separate the contributions of visual features and temporal dynamics. **Top:** Flow decreases the frames prediction. The frames stream detects memorable features like a human face or saturated colors, but the flow stream detects a static video and predicts lower memorability. **Bottom:** Flow increases the frames prediction. The flow stream picks up on dynamic patterns like fast bikers or a baby falling and increases the memorability prediction.

visual feature architectures on both Memento10k and VideoMem videos.

Our results are in the top section of Table 5.1. Out of the three input representations (frames, flow, and video), optical flow achieves the poorest performance, probably because of the lack of access to explicit visual features like color and shape. Static frames perform remarkably well on their own, even outperforming a 3D-video representation on VideoMem (as VideoMem is a fairly static dataset, this result is reasonable). The relatively high level of motion in the Memento10k dataset could explain why the video representation improves performance over static frames on Memento10k and not VideoMem. For both datasets, combining the three streams maximizes performance, which is consistent with previous work [22] and reinforces that both visual appearance and motion are relevant to predicting video memorability. In fact, our three-stream approach leverages motion information from the optical flow stream to refine its predictions, as shown in Figure 5-6.

5.5.2 Modeling semantic features

It is well-known that semantics are an important contributor to memorability [29, 31, 71, 99]. To increase our model's ability to extract semantic information, we jointly train it on memorability prediction and on a captioning task, which ensures that the underlying representation learned for both problems contains relevant event semantics. To test this approach, we enhance the video stream from the previous section with an additional module that aims to solve one of two tasks: generating captions or learning a joint text-video embedding. We choose to augment the video stream as opposed to the flow or frames streams because it contains complete visual and motion information required to reconstruct a video caption.

Caption Generation. Our first approach is to predict captions directly. This has the benefit of forcing the model to encode a rich semantic representation of the image, taking into account multiple objects, actions, and colors. However, it also involves learning an English language model, which is tangential to the task of predicting memorability. We feed the output features of the video I3D base into an LSTM that learns to predict the ground-truth captions. For Memento10k, we tokenize our 5 ground-truth captions and create a vocabulary of 3,870 words that each appear at least 5 times in our training set. We use the pre-trained FastText word embeddings [9] to map our tokens into 300-dimensional feature vectors that we feed to the recurrent module. For the VideoMem dataset, a single brief description is provided with each video; we process these descriptions the same way. We train with teacher forcing and at test-time feed the output of the LSTM back into itself.

Mapping Videos into a Semantic Embedding. Our second approach is to learn to map videos into a sentence-level semantic embedding space using a triplet loss. We pre-compute sentence embeddings for the captions using the popular transformer-based network BERT [34].[3] At training time, we stack a fully-connected layer on top of the visual encoder's output features and use a triplet loss with squared distance to ensure that the embedded representation is closer to the matching caption than a randomly selected one from our dataset. This approach has the benefit that our

[3]Computed using [108]

network does not have to learn a language model, but it may not pick up on fine-grained semantic actors in the video.

Captioning Results. The results of our captioning experiments are in the second section of Table 5.1. Caption generation outperforms the semantic embedding approach on Memento10k. Learning to generate captions provides a boost over only the video stream for both datasets. Figure 5-8 contains examples of captions generated by our model.

5.5.3 Modeling memorability decay

Up until this point, we have evaluated our memorability predictions by converting them to rankings and comparing them to the ground truth. However, the Memento10k data and our parameterization of the memorability decay curve unlocks a richer representation of memorability, where $m_t^{(v)}$ is the true probability that an arbitrary person remembers video v at lag t. Thus, we also investigate techniques for predicting the ground-truth values of the memorability decay curve. Again, we consider two alternative architectures.

Mem-α Model. "Mem-α" models produce two outputs by regressing to a video's memory score and decay coefficient. To train these models, we define a loss that consists of uniformly sampling 100 values along the true and predicted memorability curves and calculating the Mean Absolute Error on the resulting pairs. Equation 5.1 can then be used to predict the raw hit rate at a different lag.

Recurrent Decay Model. This model directly outputs multiple probability values corresponding to different points on the decay curve. It works by injecting the feature vector produced by the video encoder into the hidden state of an 8-cell LSTM, where the cells represent evenly spaced lags from $t = 40$ through 180. At each time step, the LSTM modifies the encoded video representation, which is then fed into a multi-layer perceptron to generate the hit rate at that lag. The ground truth values used during training are calculated from $\alpha^{(v)}$ and $m_{80}^{(v)}$ using Equation 5.1.

Decay Results. We evaluate the models in two ways. First, we calculate rank correlation with ground truth, based on memorability score (defined as $m_{80}^{(v)}$). We

Approach	RC	R^2		
		T=40	T=80	T=160
Mem-α	**0.618**	0.18	**0.34**	0.30
Recurrent head	0.615	**0.31**	0.33	**0.32**

Table 5.2: **Multi-lag memorability prediction**: Rank correlation (RC) and raw predictions (R^2) at 3 different lags (T, representing the number of intervening videos)

also compare their raw predictions for different values of t, for which we report R^2. The results are in Table 5.2.

We find that the Mem-α has better performance than the recurrent decay model at predicting hit rates at $T = 80$; since we define a video's memorability score as its hit rate at lag 80, this means that the Mem-α version achieves a better rank correlation vis-a-vis human rankings. However, the recurrent decay model outperforms the Mem-α model at predicting raw hit rates at other lags. It makes sense that the performance of the Mem-α model falls off at lags further away from $T = 80$, since any error in the prediction of alpha (the slope of the decay curve) is amplified as we extrapolate away from the reference lag. The recurrent decay model avoids this downside, but it has many more parameters and underperforms at the reference lag. These two models present a trade-off between simplicity and ranking accuracy (mem-α) and numerical accuracy along the entire decay curve (recurrent decay). Because of its relative simplicity and effectiveness at $T = 80$, we use a Mem-α architecture for our final predictions.

5.5.4 Modeling results

SemanticMemNet (Figure 5-7) combines our findings from the three previous sections. We use a three-stream encoder that operates on three different representations of the input video: 1) the raw frames, 2) the entire video as a 3D unit, and 3) the 3D optical flow. In addition, we jointly train the video stream to output memorability scores and captions for the video. Each of our streams predicts both the memorability and the decay rate of the video, which allows us to predict the probability that an observer will recall the video at an arbitrary lag within the range we studied.

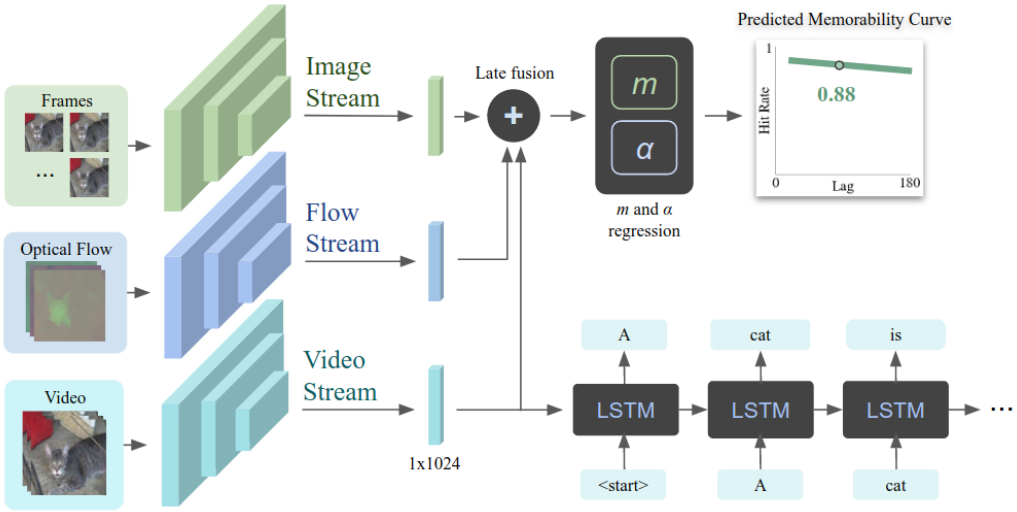

Figure 5-7: **The architecture of SemanticMemNet**. An I3D is jointly trained to predict memorability and semantic captions for an input video. Its memorability predictions are combined with a frames-based and optical flow stream to produce m_{80} and α, the parameters of the memorability decay curve.

Approach	**RC - Memento10k (test set)**	**RC - VideoMem (validation set)**
Human consistency	0.730	0.616
MemNet Baseline [67]	0.480	0.425
Feature extraction + regression (as in [99])*	0.636	0.427
Cohendet et al. (ResNet3D) [29]	0.511	0.508
Cohendet et al. (Semantic)[29]	0.554	0.503
SemanticMemNet	**0.664**	**0.556**

Table 5.3: **Comparison to state-of-the-art** on Memento10k and VideoMem. Our approach, SemanticMemNet, approaches human consistency and outperforms previous approaches. Note: we evaluate on the VideoMem validation set as the test set has not been made public. *Uses ground-truth captions at test-time.

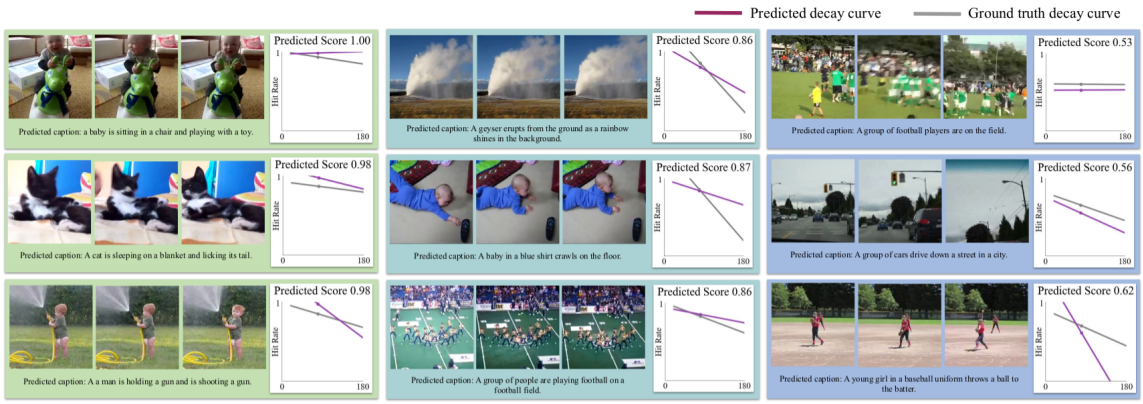

Figure 5-8: **Memorability and captions predictions from SemanticMemNet.** We show examples from the Memento10k test set that the model scores as high (left), medium (middle), and low (right) memorability. For each example, we plot the predicted memorability decay curve based on SemanticMemNet's values for m_T and α in purple, as well as the ground truth in gray. More results in Appendix C.

To evaluate the effectiveness and generalizability of our model, we compare against prior work in memorability prediction. MemNet [67] is a strong baseline for image memorability; we apply it to our videos by averaging its predictions over 7 frames uniformly sampled from the video. "Feature extraction and regression" is based on the approach from Shekhar et al. [99], where semantic, spatio-temporal, saliency, and color features are first extracted from the video and then a regression is learned to predict the memorability score. The final two baselines are the best-performing models from Cohendet et al. [29]. The results of our evaluations are in Table 5.3. Our model achieves state-of-the-art performance on both Memento10k and VideoMem. Example predictions generated by our model are in Fig. 5-8.

5.6 Applications and future work

While our model approaches human performance on predicting rankings, memorability is not a solved problem. Of course, there is still work to be done in terms of improving the understanding and accuracy of our model. Figure 5-9 analyses instances where the model fails because of competing visual attributes or complex semantics that humans can grasp but the model does not.

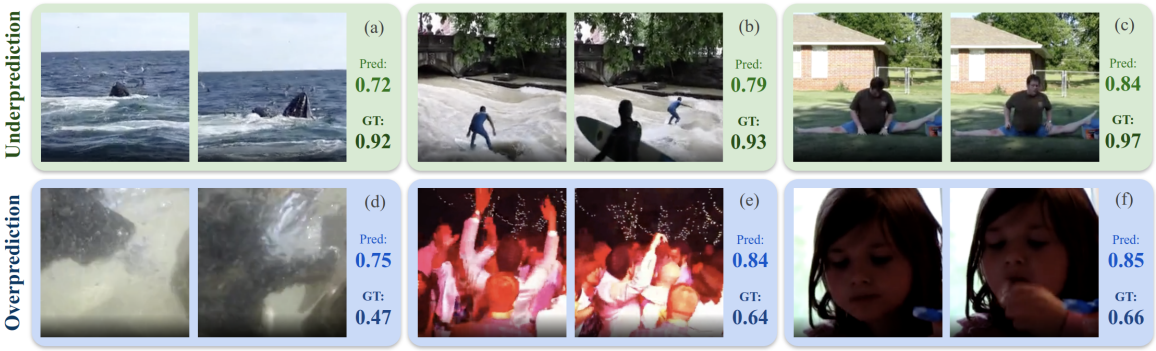

Figure 5-9: **Under and overpredictions of SemanticMemNet.** Our network often underestimates the memorability of visually bland scenes with a single distinctive element, like an ocean punctuated by a whale sighting **(a)** or a backyard with a man in an unusual pose **(c)**. It can fail on out-of-context events, like someone surfing on a flooded concrete river **(b)**. By contrast, it overestimates the memorability of choppy, dynamic scenes without clear semantic content **(d)** and of scenes that contain memorable elements, such as humans and faces, but that are overly cluttered **(e)**, dark **(f)**, or otherwise shaky or poor-quality.

In addition, future work may extend our understanding of memorability to longer video sequences. Our approach of modeling how memory changes over time makes progress towards continuous memorability prediction for long videos (i.e. first-person live streams, YouTube videos) where memorability models should handle past events *and* their decay rates, to assess memorability from different points in the past. As a proof-of-concept, we created a demo application[4] that applies the Memento model to video clips longer than 3 seconds using a simple sliding window approach (Figure 5-10). The resulting "memorability graphs", which plot memorability as a function of time, could be used to select the most memorable parts of a video to create a "memory-jogging" video thumbnail, or could be applied to a live video stream to decide on the fly which video segments to preserve or discard. There is exciting potential for models and applications that predict how longer chains of events will be preserved in memory.

[4] `http://demo.memento.csail.mit.edu/`

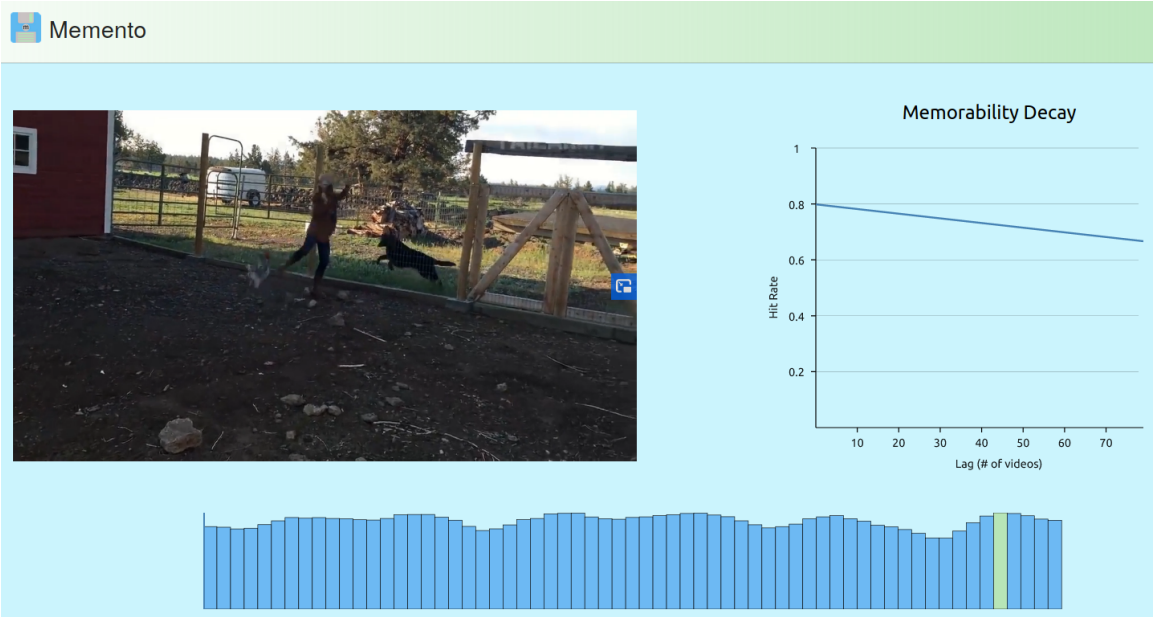

Figure 5-10: **Memento demo on long video segments.** This is a screenshot showing the memorability graph and average decay rate for a longer video clip of about one minute. The memorability graph (bottom of the screen) indicates which parts of the video are more or less memorable. In this screenshot, the model indicates that a segment where a woman is being attacked by a chicken is one of the most memorable moments in this clip.

Chapter 6

Conclusion

6.1 Contributions

6.1.1 UI tools for measuring human perception

One goal of this work is to develop techniques for studying human perception that scale to the type of data required by modern machine learning systems.

In Chapter 3, we sought to make attention capture more flexible and accessible. We introduced TurkEyes, a crowdsourceable UI toolbox that relies on user interaction, not eye tracking, to capture attention data. By studying and standardizing different interaction techniques from the literature, we are able to provide guidance for how to choose the best interface for a particular use case.

In Chapter 4, we used one of the TurkEyes techniques to explore a novel aspect of visual attention: how saliency changes as a function of viewing duration. We harnessed the CodeCharts interface to gather gaze points at precise viewing durations, resulting in CodeCharts1k, the first multi-duration saliency dataset. In our analysis of the dataset, we found that saliency is predictable across viewing durations, and discovered some systematic patterns in how attention changes across durations, such as the "boomerang" pattern on faces.

In Chapter 5, we expanded the old-new recognition paradigm used to study image memorability to videos. We created *The Memento Game*, a fast-paced and engaging

online memory game that we used to collect the Memento10k dataset. Memento10k is the biggest video memorability dataset to date and provides dynamic, in-the-wild videos, semantic annotations, and many annotations per video to robustly estimate memory decay over time. Analysis of our data suggests a model of memory decay where stimuli decay linearly with time (within the range that we studied) and where a video's decay rate must be estimated on a per-stimulus basis.

6.1.2 Models and applications for perceptual attributes

The second goal of this work is to create computer models that predict human perceptual responses directly from pixels, paving the way for applications that can automatically curate visual content.

In Chapter 4, we create a Multi-Duration Saliency Excited Model that accurately predicts saliency heatmaps corresponding to multiple viewing durations. This novel architecture models differences across viewing durations with fewer parameters than competing models. We show that multi-duration saliency predictions can be used to enhance applications like cropping, rendering, and captioning, tailoring results to the amount of time a viewer is expected to allocate to different image regions.

In Chapter 5, we create SemanticMemNet, a multimodal memorability model that achieves state-of-the-art performance on video memorability prediction. SemanticMemNet explicitly models semantic information that is known to be important for memorability and uses our insights about memory decay to predict a video's memorability score at an arbitrary delay. SemanticMemNet, which works on short three-second sequences, is the first step towards predicting the memorability of longer events.

6.2 Looking forward

A significant consequence of this work is making human perception data and predictions more accessible to researchers and creators. We hope our data, tools, and models will be used to explore different image types, tasks, and applications. For

example, crowdsourced attention or models trained on it could be used to identify areas of interest in satellite pictures or medical images. Attention can be used as a debugging tool for designers, to answer questions like: Are the correct parts of a design attention-grabbing? Is the most important element also the most salient? Are people spending time attending to details or just skimming the big picture? Cheap, scalably crowdsourced attention data can be used to train computational saliency models on stimuli that have been under-studied, like non-natural images or even videos. Furthermore, video memorability models open the door to many exciting applications in computer vision. They can be used to provide recommendations to designers and educators to select and post-process clips that will be durable in memory. They can improve summarization by selecting segments most likely to be recalled. They can be used to guide generation of more memorable content. They can act as a measure of the utility of different video segments in space-constrained systems; for instance, a camera in a self-driving car or a pair of virtual assistant glasses could discard data once it has fallen below a certain memorability threshold.

Predicting human perception will lead to systems that make intelligent decisions about what information to prioritize, create, enhance, and preserve. That will in turn make these systems more useful for the humans who perceive them.

Appendix A

Quantitative evaluation of MD-SEM on CodeCharts1k

Table A.1 compares MD-SEM to SAM $\times 3$ and SAM-MD across all three durations in CodeCharts1k.

		500ms			3000ms			5000ms			All durations		
Model	*Params* ↓	*CC* ↑	*NSS* ↑	*KL* ↓	*CC* ↑	*NSS* ↑	*KL* ↓	*CC* ↑	*NSS* ↑	*KL* ↓	*CC* ↑	*NSS* ↑	*KL* ↓
SAM ×3	210.3M	0.804	3.236	0.366	0.693	2.409	0.545	0.706	2.480	0.537	0.734	2.708	0.483
SAM-MD	70.1M	0.805	3.181	0.370	0.738	2.541	0.469	0.715	2.495	0.535	0.753	2.739	0.458
MD-SEM	**30.9M**	**0.816**	**3.374**	**0.351**	**0.745**	**2.694**	**0.452**	**0.734**	**2.677**	**0.487**	**0.765**	**2.915**	**0.430**

Table A.1: Comparison of multi-duration saliency models evaluated on CodeCharts1K. Baselines are SAM×3 (three copies of SAM, each trained exclusively on data for one duration) and SAM-MD (a custom modification of SAM whose LSTM outputs multiple maps). MD-SEM (ours) excels across all three viewing durations, while using substantially fewer parameters.

Appendix B

Additional multi-duration saliency predictions

Model predictions. Fig. B-1 shows representative predictions of MD-SEM on various datasets.

Applications. Fig. B-2 contains examples of how multi-duration saliency heatmaps can be used to crop parts of an image that are salient at different times. Fig. B-3 shows how multi-duration saliency can be used to select which elements in an image should be rendered first (or at higher resolution). Fig. B-4 contains additional examples of how multi-duration saliency can be used to generate captions that pick up on additional objects or focus on salient parts of an image.

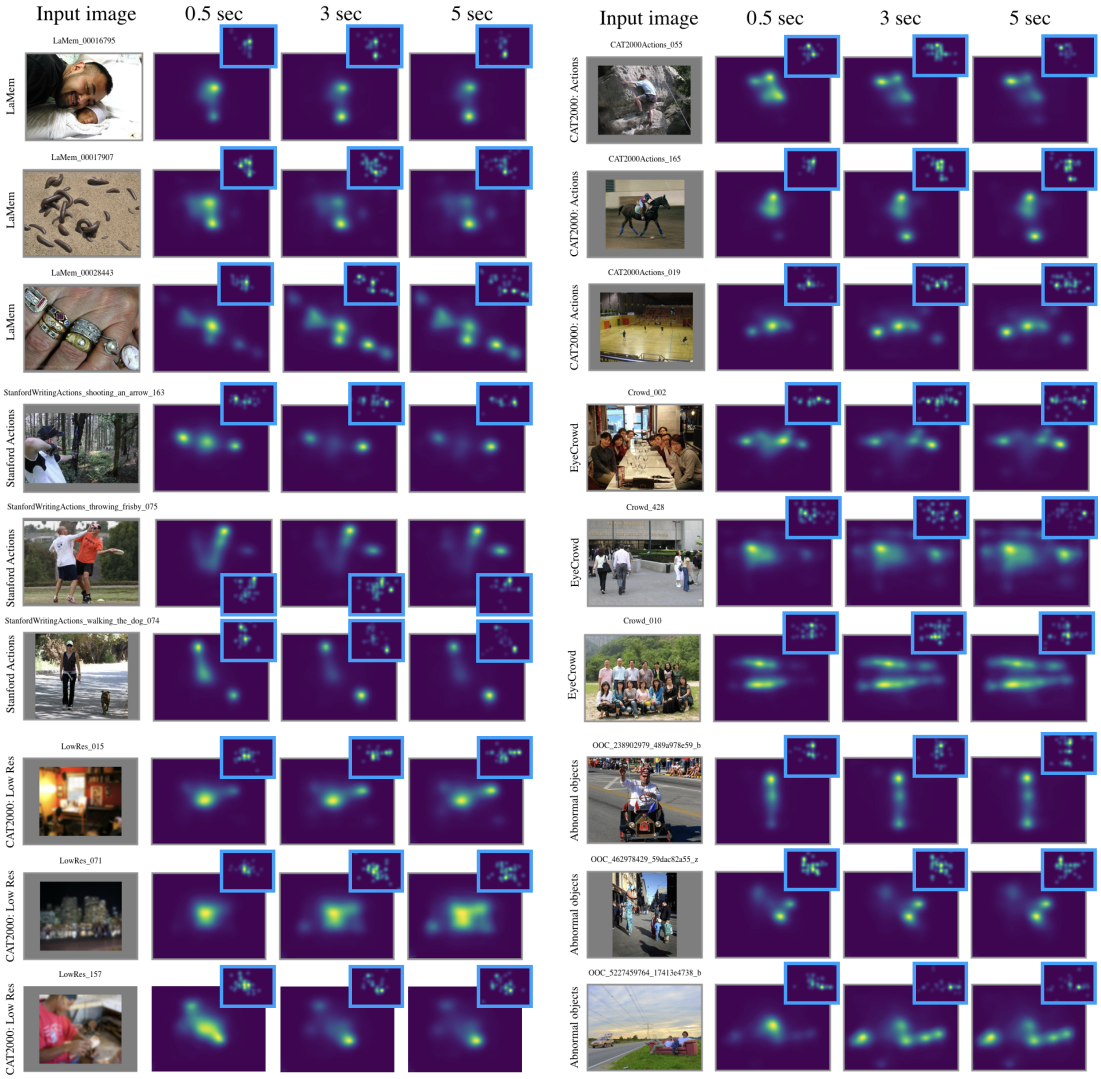

Figure B-1: Saliency predictions of MD-SEM on various datasets. Insets with blue borders contain human ground-truth gaze locations collected using our CodeCharts UI. In all cases, we see that our model has learned to make distinctively different predictions for the different viewing durations. The model learns to start either more centrally or by focusing on the main actor in the scene in the first 0.5 sec. With longer viewing durations the model's predictions move towards other salient image elements that are smaller or more distant from the center. We can see a failure mode of our model on the large crowd of people in the right column, as our model struggles to determine who to focus on. We see another two difficult cases in the last two rows of the same column, where the model needs semantic knowledge to correctly distribute attention to objects that are out of place.

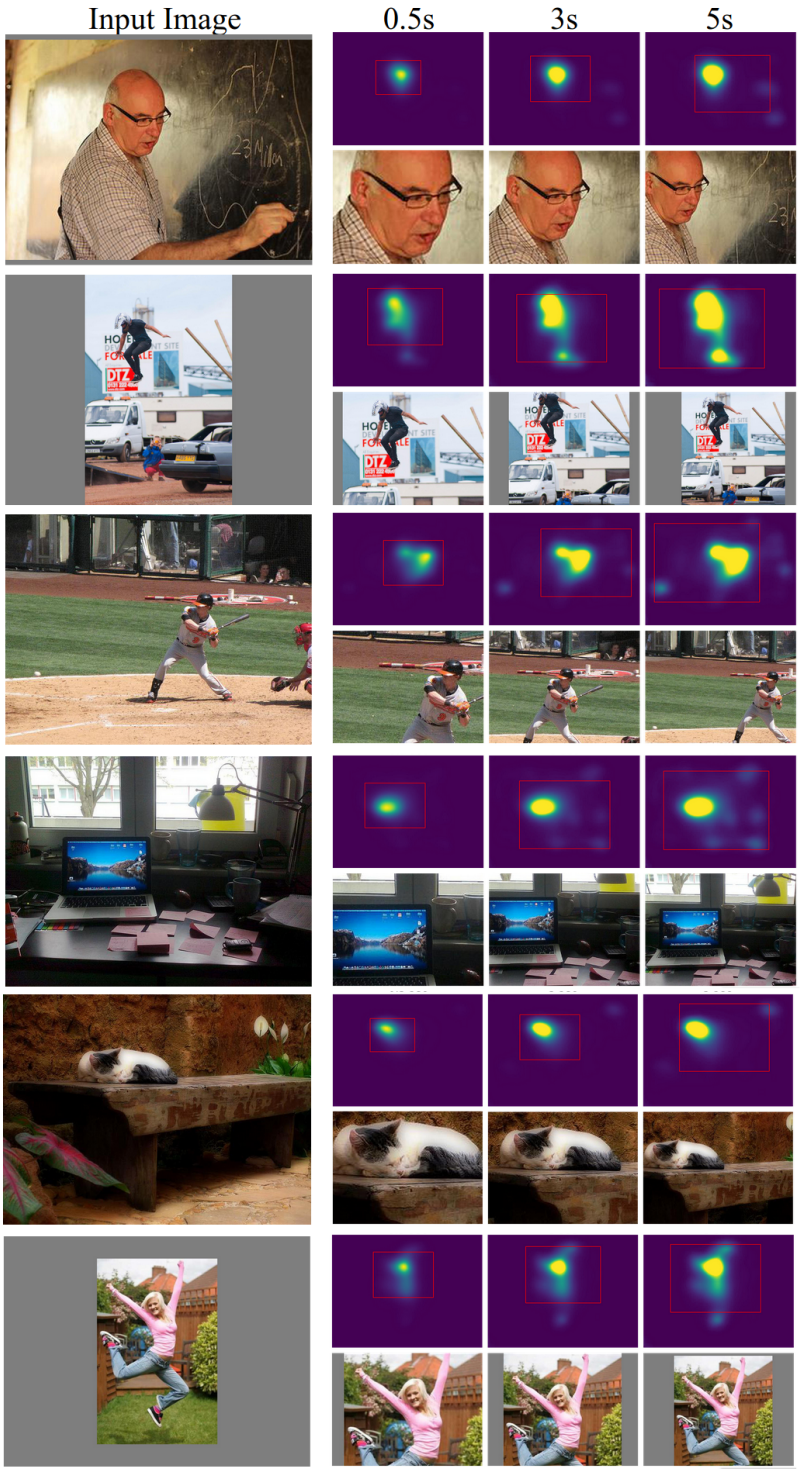

Figure B-2: Example crops generated based on saliency maps for different viewing durations. The original images appear on the left. On the right we show the predicted saliency heatmap, along with the 90% bounding box, for each duration (top row) and the resulting cropped image (bottom row). Crops for 0.5 seconds tend to focus on a single highly salient object or point, while crops at longer durations expand to include other parts of the image such as the background or the object of the action.

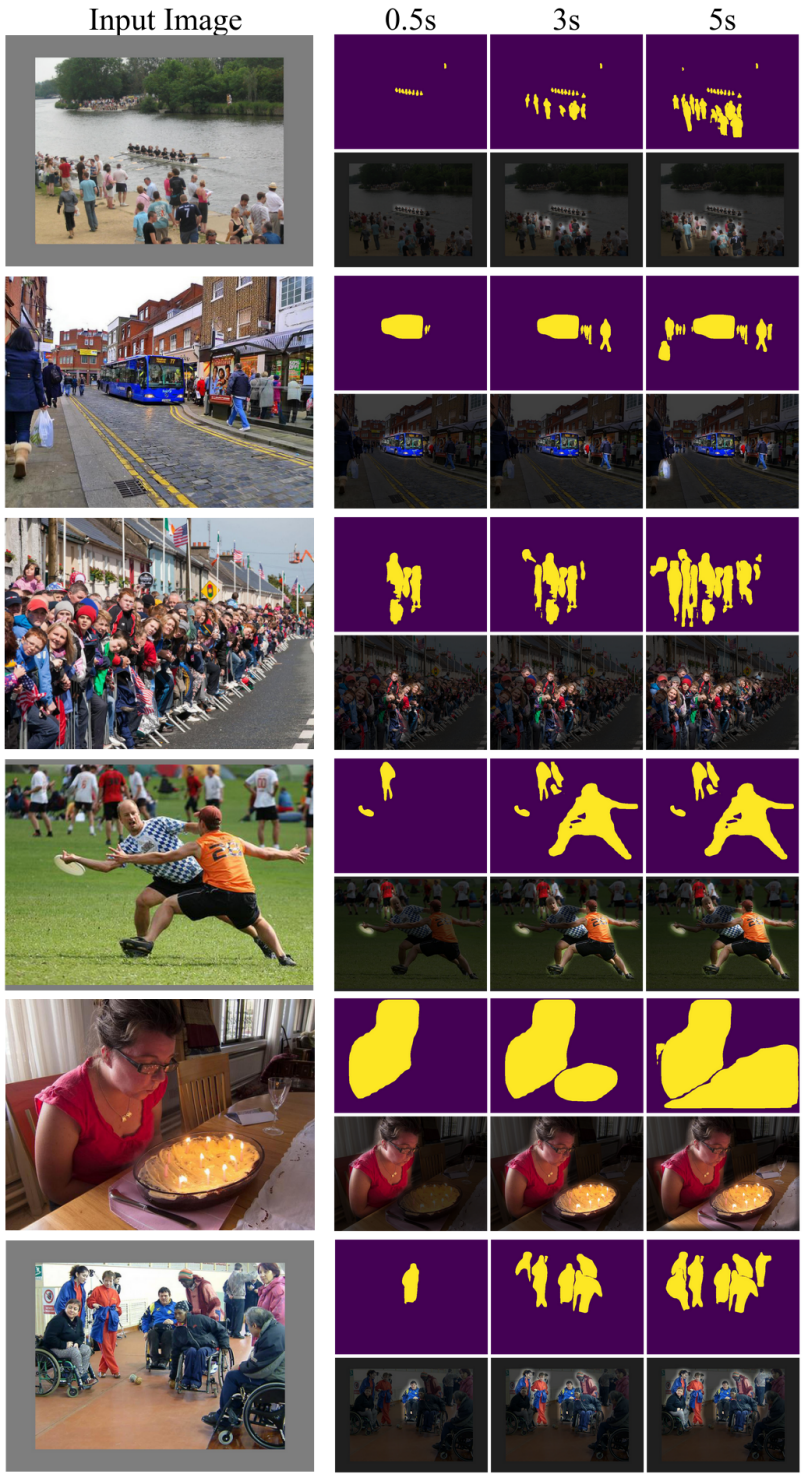

Figure B-3: Examples of how multi-duration saliency can be applied to compression and rendering. The original images appear on the left. On the right we show the segmentation maps of instances with saliency scores in the 90th percentile based on the cumulative saliency map for that duration (top row) and a visualization of those salient objects (bottom row). Objects that are highly salient at 0.5 or 3 seconds could be rendered before objects that become salient later.

Figure B-4: Examples of how multi-duration saliency can be applied to captioning. The captions corresponding to saliency-enhanced images for different durations can sometimes produce different captions by refocusing attention on relevant areas in a scene.

Appendix C

Additional memorability predictions

We show multiple example predictions of our SemanticMemNet. Figure C-1 shows some good predictions across the memorability spectrum, while Figure C-2 shows some additional failure cases.

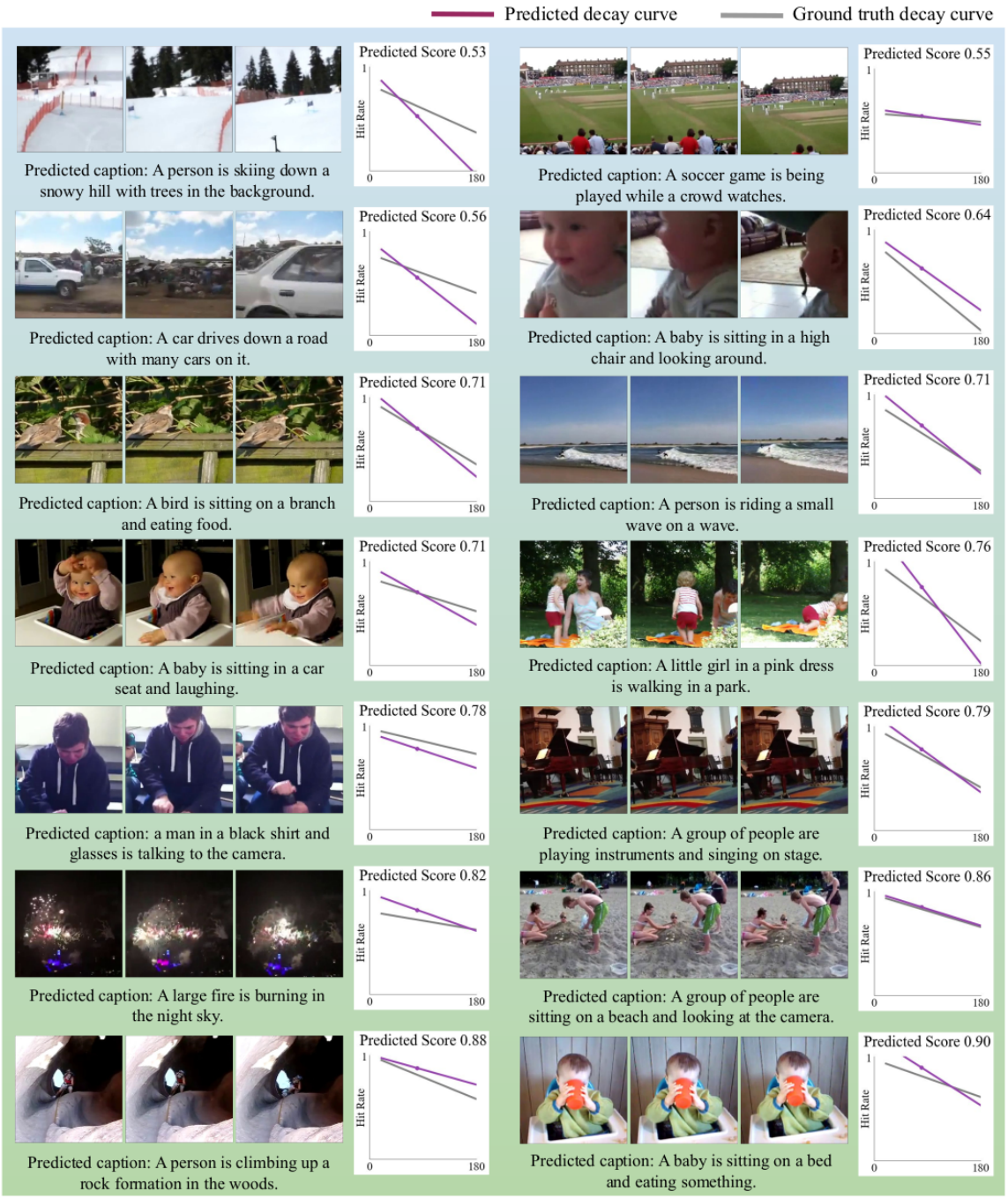

Figure C-1: More memorability and captions predictions from SemanticMemNet on the Memento10k test set.

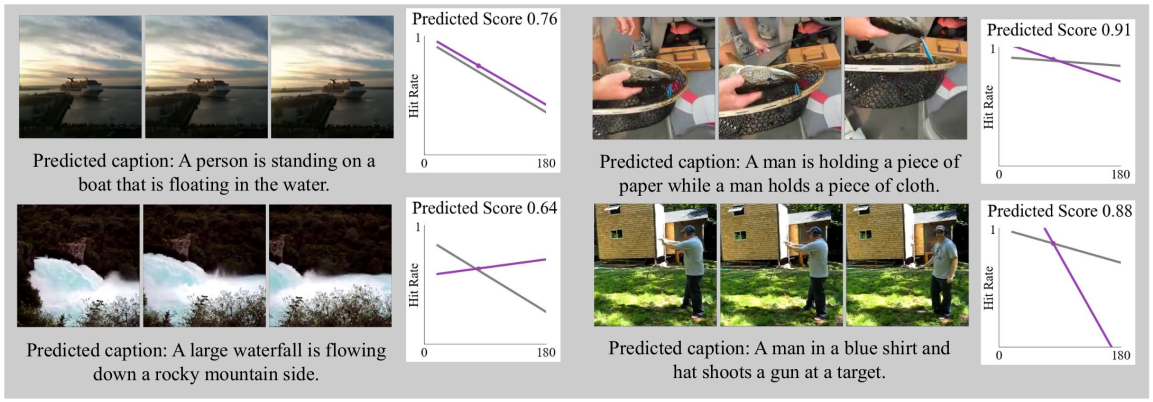

Figure C-2: Failure cases for SemanticMemNet. Top row: The model outputs an accurate memorability curve, even though the caption has a significant error (there are no people visible in the video). Bottom row: in spite of correctly captioning the videos, the predicted memorability curve is off. Significantly, the model often predicts the correct memorability *score* but an incorrect α. This is likely owing to our emphasis on generating an accurate memorability ranking.

Bibliography

[1] SALICON Saliency Prediction Challenge (LSUN 2017). `https://competitions.codalab.org/competitions/17136#results`.

[2] Erdem Akagunduz, Adrian G. Bors, and Karla K Evans. Defining Image Memorability using the Visual Memory Schema. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*.

[3] Marc Assens Reina, Xavier Giró-i Nieto, Kevin McGuinness, and Noel E. O'Connor. Saltinet: Scan-path prediction on 360 degree images using saliency volumes. In *ICCV Workshop on Egocentric Perception, Interaction and Computing*, Oct 2017.

[4] Wilma Bainbridge, Phillip Isola, and Oliva Aude. The Intrinsic Memorability of Face Photographs. *Journal of Experimental Psychology: General*, 142:1323–1334, 2013.

[5] Lyn Bartram, Albert Ho, John Dill, and Frank Henigman. The continuous zoom: A constrained fisheye technique for viewing and navigating large information spaces. pages 207–215, 01 1995.

[6] Yoann Baveye, Romain Cohendet, Matthieu Perreira Da Silva, and Patrick Le Callet. Deep Learning for Image Memorability Prediction: The Emotional Bias. In *Proceedings of the 24th ACM International Conference on Multimedia*, pages 491–495. ACM, 2016.

[7] Roman Bednarik and Markku Tukiainen. Effects of display blurring on the behavior of novices and experts during program debugging. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 1204–1207, New York, NY, USA, 2005. ACM.

[8] Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. Toolglass and magic lenses: The see-through interface. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 73–80, New York, NY, USA, 1993. ACM.

[9] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

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

[11] Ali Borji and Laurent Itti. CAT2000: A large scale fixation dataset for boosting saliency research. *CVPR'15 Workshop on the Future of Datasets*, 2015.

[12] A.M. Borkin, Z. Bylinskii, N.W. Kim, C.M. Bainbridge, C.S. Yeh, D. Borkin, H. Pfister, and A. Oliva. Beyond Memorability: Visualization Recognition and Recall. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):519–528, 2016.

[13] Michelle A Borkin, Azalea A Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. What makes a visualization memorable? *TVCG*, 19(12):2306–2315, 2013.

[14] Timothy F Brady, Talia Konkle, George A Alvarez, and Aude Oliva. Visual Long-Term Memory Has a Massive Storage Capacity for Object Details. *Proceedings of the National Academy of Sciences*, 105(38):14325–14329, 2008.

[15] Guy Thomas Buswell. How people look at pictures: a study of the psychology and perception in art. 1935.

[16] Zoya Bylinskii, Phillip Isola, Constance Bainbridge, Antonio Torralba, and Aude Oliva. Intrinsic and Extrinsic Effects on Image Memorability. *Vision Research*, 116:165–178, 2015.

[17] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. Mit saliency benchmark, 2014.

[18] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *CoRR*, abs/1604.03605, 2016.

[19] Zoya Bylinskii, Nam Wook Kim, Peter O'Donovan, Sami Alsheikh, Spandan Madan, Hanspeter Pfister, Fredo Durand, Bryan Russell, and Aaron Hertzmann. Learning visual importance for graphic designs and data visualizations. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software & Technology*, UIST '17. ACM, 2017.

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

[21] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018.

[22] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4724–4733, 2017.

[23] Moran Cerf, E Paxon Frady, and Christof Koch. Faces and text attract gaze independent of the task: Experimental data and computer model. *Journal of vision*, 9(12):10–10, 2009.

[24] Jiansheng Chen, Gaocheng Bai, Shaoheng Liang, and Zhengqin Li. Automatic image cropping: A computational complexity study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 507–515, 2016.

[25] Yi-Ling Chen, Tzu-Wei Huang, Kai-Han Chang, Yu-Chen Tsai, Hwann-Tzong Chen, and Bing-Yu Chen. Quantitative analysis of automatic image cropping algorithms:a dataset and comparative study. In *IEEE WACV 2017*, 2017.

[26] Shiwei Cheng, Zhiqiang Sun, Xiaojuan Ma, Jodi L. Forlizzi, Scott E. Hudson, and Anind Dey. Social eye tracking: Gaze recall with online crowds. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '15, pages 454–463, New York, NY, USA, 2015. ACM.

[27] Myung Jin Choi, Antonio Torralba, and Alan S Willsky. Context models and out-of-context objects. *Pattern Recognition Letters*, 33(7):853–862, 2012.

[28] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[29] Romain Cohendet, Claire Demarty, Ngoc Q. K. Duong, and Engilberge Martin. VideoMem: Constructing, Analyzing, Predicting Short-Term and Long-Term Video Memorability. In *Proceedings of the IEEE international conference on computer vision*, pages 2531–2540, 2019.

[30] Romain Cohendet, Claire-Hélène Demarty, Ngoc Q. K. Duong, Mats Sjöberg, Bogdan Ionescu, Thanh-Toan Do, and France Rennes. MediaEval 2018: Predicting Media Memorability Task. *CoRR*, abs/1807.01052, 2018.

[31] Romain Cohendet, Karthik Yadati, Ngoc QK Duong, and Claire-Hélène Demarty. Annotating, Understanding, and Predicting Long-Term Video Memorability. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 178–186. ACM, 2018.

[32] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Paying more attention to saliency: Image captioning with saliency and context attention. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(2):48, 2018.

[33] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Predicting human eye fixations via an lstm-based saliency attentive model. *IEEE Transactions on Image Processing*, 27(10):5142–5154, 2018.

[34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[35] Rachit Dubey, Joshua Peterson, Aditya Khosla, Ming-Hsuan Yang, and Bernard Ghanem. What Makes an Object Memorable? In *Proceedings of the ieee international conference on computer vision*, pages 1089–1097, 2015.

[36] Martin Engilberge, Louis Chevallier, Patrick Pérez, and Matthieu Cord. Finding beans in burgers: Deep semantic-visual embedding with localization. *CoRR*, abs/1804.01720, 2018.

[37] Rachel England. Twitter uses smart cropping to make image previews more interesting. `https//engadget.com/2018/01/25/twitter-uses-smart-cropping-to-make-image-previews-more-interest`, 2018.

[38] Seyed A Esmaeili, Bharat Singh, and Larry S Davis. Fast-at: Fast automatic thumbnail generation using deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4622–4630, 2017.

[39] Jiri Fajtl, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Amnet: Memorability Estimation with Attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6363–6372, 2018.

[40] Camilo Fosco*, Anelise Newman*, Pat Sukhum, Yun Bin Zhang, Nanxuan Zhou, Aude Oliva, and Zoya Bylinskii. How much time do you have? modeling multi-duration saliency. In *2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[41] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2121–2129. Curran Associates, Inc., 2013.

[42] Adam Geitgey. Face recognition. `http://github.com/ageitgey/face_recognition`.

[43] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. GANalyze: Towards Visual Definition of Cognitive Image Properties. In *IEEE International Conference on Computer Vision, ICCV 2019, Seoul, Korea*, pages 5744–5753, 2019.

[44] Lore Goetschalckx, Pieter Moors, and Johan Wagemans. Image memorability across longer time intervals. *Memory*, 26:581–588, 5 2017.

[45] Qi Guo and Eugene Agichtein. Towards predicting web searcher gaze position from mouse movements. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 3601–3606, New York, NY, USA, 2010. ACM.

[46] Qi Guo, Haojian Jin, Dmitry Lagun, Shuai Yuan, and Eugene Agichtein. Mining touch interaction data on mobile devices to predict web search result relevance. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 153–162, New York, NY, USA, 2013. ACM.

[47] Qi Guo and Yang Song. Large-scale analysis of viewing behavior: Towards measuring satisfaction with mobile proactive systems. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 579–588, New York, NY, USA, 2016. ACM.

[48] Junwei Han, Changyuan Chen, Ling Shao, Hu Xintao, Han Jungong, and Liu Tianming. Learning computational models of video memorability from fMRI brain imaging. *Cybernetics, IEEE Transactions on*, 45(8):1692–1703, 2015.

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

[50] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[51] Jeff Huang and Abdigani Diriye. Web user interaction mining from touch-enabled mobile devices. 2012.

[52] Jeff Huang, Ryen White, and Georg Buscher. User see, user point: Gaze and cursor alignment in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1341–1350, New York, NY, USA, 2012. ACM.

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

[54] Phillip Isola, Devi Parikh, Antonio Torralba, and Aude Oliva. Understanding the intrinsic memorability of images. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011.*, pages 2429–2437, 2011.

[55] Phillip Isola, Jianxiong Xiao, Devi Parikh, Antonio Torralba, and Aude Oliva. What Makes a Photograph Memorable? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(7):1469–1482, 2014.

[56] Phillip Isola, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. What Makes an Image Memorable? In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 145–152, 2011.

[57] Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40(10):1489 – 1506, 2000.

[58] Laurent Itti and Christof Koch. Computational modeling of visual attention. *Nature reviews. Neuroscience*, 2:194–203, 04 2001.

[59] A. Jaegle, V. Mehrpour, Y. Mohsenzadeh, T. Meyer, A. Oliva, and N. Rust. Population response magnitude variation in inferotemporal cortex predicts image memorability. *eLife*, 8:e47596, 2019.

[60] Anthony R. Jansen, Alan F. Blackwell, and Kim Marriott. A tool for tracking visual attention: The restricted focus viewer. *Behavior Research Methods, Instruments, & Computers*, 35(1):57–69, 2003.

[61] Sen Jia. Eml-net: An expandable multi-layer network for saliency prediction. *arXiv preprint arXiv:1805.01047*, 2018.

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

[63] Ming Jiang, Juan Xu, and Qi Zhao. Saliency in crowd. In *European Conference on Computer Vision*, pages 17–32. Springer, 2014.

[64] Tilke Judd, Fredo Durand, and Antonio Torralba. Fixations on low-resolution images. *Journal of Vision*, 11(4):14–14, 2011.

[65] Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. *CoRR*, abs/1412.2306, 2014.

[66] Aditya Khosla, Wilma Bainbridge, Antonio Torralba, and Aude Oliva. Modifying the Memorability of Face Photographs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3200–3207, 2013.

[67] Aditya Khosla, Akhil S Raju, Antonio Torralba, and Aude Oliva. Understanding and Predicting Image Memorability at a Large Scale. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2390–2398, 2015.

[68] Aditya Khosla, Jianxong Xiao, Antonio Torralba, and Aude Oliva. Memorability of Image Regions. In *Advances in Neural Information Processing Systems*, pages 305–313, 2012.

[69] Nam Wook Kim, Zoya Bylinskii, Michelle A Borkin, Krzysztof Z Gajos, Aude Oliva, Fredo Durand, and Hanspeter Pfister. Bubbleview: an interface for crowdsourcing image importance maps and tracking visual attention. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24(5):36, 2017.

[70] Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539, 2014.

[71] Talia Konkle, Timothy Brady, George Alvarez, and Aude Oliva. Conceptual distinctiveness supports detailed visual long-term memory for real-world objects. *Journal of Experimental Psychology: General*, 139:558–578, 3 2010.

[72] Talia Konkle, Timothy Brady, George Alvarez, and Aude Oliva. Scene Memory Is More Detailed Than You Think: The Role of Categories in Visual Long-Term Memory. *Psychological Science*, 21:1551–6, 10 2010.

[73] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2176–2184, June 2016.

[74] Dmitry Lagun, Chih-Hung Hsieh, Dale Webster, and Vidhya Navalpakkam. Towards better measurement of attention and satisfaction in mobile search. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 113–122, New York, NY, USA, 2014. ACM.

[75] Dmitry Lagun and Mounia Lalmas. Understanding user attention and engagement in online news reading. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, pages 113–122, New York, NY, USA, 2016. ACM.

[76] F. Lamberti, Gianluca Paravati, Valentina Gatteschi, and Alberto Cannavò. Supporting web analytics by aggregating user interaction data from heterogeneous devices using viewport-dom-based heat maps. *IEEE Transactions on Industrial Informatics*, PP:1–1, 01 2017.

[77] Olivier Le Meur and Thierry Baccino. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior research methods*, 45(1):251–266, 2013.

[78] Yixuan Li, Pingmei Xu, Dmitry Lagun, and Vidhya Navalpakkam. Towards measuring and inferring user interest from gaze. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 525–533, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[79] George W. McConkie and Keith Rayner. The span of the effective stimulus during a fixation in reading. *Perception & Psychophysics*, 17(6):578–586, 1975.

[80] Y. Mohsenzadeh, C. Mullin, A. Oliva, and D. Pantazis. The perceptual neural trace of memorable unseen scenes. *Scientific Reports*, 8:6033, 2019.

[81] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Yan Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[82] Anelise Newman, Barry McNamara, Camilo Fosco, Yun Bin Zhang, Pat Sukhum, Matthew Tancik, Nam Wook Kim, and Zoya Bylinskii. TurkEyes: A web-based toolbox for crowdsourcing attention data. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020.

[83] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. Learning layouts for single-page graphic designs. *IEEE Transactions on Visualization and Computer Graphics*, 20(8):1200–1213, Aug 2014.

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

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

[86] Alexandra Papoutsaki, Patsorn Sangkloy, James Laskey, Nediyana Daskalova, Jeff Huang, and James Hays. Webgazer: Scalable webcam eye tracking using user interactions. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3839–3845. AAAI, 2016.

[87] Sangdon Park, Wonsik Kim, and Kyoung Mu Lee. Abnormal object detection by canonical scene-based contextual model. In *European Conference on Computer Vision (ECCV)*, 2012.

[88] Shay Perera, Ayellet Tal, and Lihi Zelnik-Manor. Is Image Memorability Prediction Solved? In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[89] Michael Posner and Yoav Cohen. Components of visual orienting. *Attention and performance X: Control of language processes*, 32:531–, 01 1984.

[90] Keith Rayner. The gaze-contingent moving window in reading: Development and review. *Visual Cognition*, 22(3-4):242–258, 2014.

[91] Navyasri Reddy, Samyak Jain, Pradeep Yarlagadda, and Vineet Gandhi. Tidying deep saliency prediction architectures. *ArXiv*, abs/2003.04942, 2020.

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

[93] Kerry Rodden, Xin Fu, Anne Aula, and Ian Spiro. Eye-mouse coordination patterns on web search results pages. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pages 2997–3002, New York, NY, USA, 2008. ACM.

[94] Dmitry Rudoy, Dan B Goldman, Eli Shechtman, and Lihi Zelnik-Manor. Crowdsourcing gaze data collection. *Proceedings of ACM Collective Intelligence Conference*, 2012.

[95] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3):157–173, 2008.

[96] Amaia Salvador, Axel Carlier, Xavier Giro-i Nieto, Oge Marques, and Vincent Charvillat. Crowdsourced object segmentation with a game. In *Proceedings of the 2Nd ACM International Workshop on Crowdsourcing for Multimedia*, CrowdMM '13, pages 15–20, New York, NY, USA, 2013. ACM.

[97] Arthur Samuel and Donna Kat. Inhibition of return: A graphical meta-analysis of its time course and an empirical test of its temporal and spatial properties. *Psychonomic bulletin review*, 10:897–906, 01 2004.

[98] Michael Schulte-Mecklenbeck, Ryan O. Murphy, and Florian Hutzler. Flashlight - recording information acquisition online. *Comput. Hum. Behav.*, 27(5):1771–1782, September 2011.

[99] Sumit Shekhar, Dhruv Singal, Harvineet Singh, Manav Kedia, and Akhil Shetty. Show and Recall: Learning What Makes Videos Memorable. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2730–2739, 2017.

[100] Oleksii Sidorov. Changing the Image Memorability: From Basic Photo Editing to GANs. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[101] Hammad Squalli-Houssaini, Ngoc Duong, Marquant Gwenaëlle, and Claire-Hélène Demarty. Deep learning for predicting image memorability. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.

[102] Benjamin W. Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of Vision*, 7(14):4–4, 11 2007.

[103] Benjamin W. Tatler, Roland J. Baddeley, and Iain D. Gilchrist. Visual correlates of fixation selection: effects of scale and time. *Vision Research*, 45(5):643 – 659, 2005.

[104] Benjamin W. Tatler, Mary M. Hayhoe, Michael F. Land, and Dana H. Ballard. Eye guidance in natural vision: Reinterpreting salience. *Journal of Vision*, 11(5):5, 2011.

[105] Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence – video to text. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[106] Melissa Le-Hoa Võ, Zoya Bylinskii, and Aude Oliva. Image Memorability In The Eye Of The Beholder: Tracking The Decay Of Visual Scene Representations. *bioRxiv*, page 141044, 2017.

[107] Zijun Wei, Jianming Zhang, Xiaohui Shen, Zhe Lin, Radomír Mech, Minh Hoai, and Dimitris Samaras. Good view hunting: learning photo composition from dense view pairs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5437–5446, 2018.

[108] Han Xiao. bert-as-service. `https://github.com/hanxiao/bert-as-service`, 2018.

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

[110] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044, 2015.

[111] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *2011 International Conference on Computer Vision*, pages 1331–1338. IEEE, 2011.

[112] Soodabeh Zarezadeh, Mehdi Rezaeian, and Mohammad Taghi Sadeghi. Image Memorability Prediction Using Deep Features. In *2017 Iranian Conference on Electrical Engineering (ICEE)*, pages 2176–2181. IEEE, 2017.

[113] Xucong Zhang, Michael Xuelin Huang, Yusuke Sugano, and Andreas Bulling. Training person-specific gaze estimators from user interactions with multiple

devices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 624. ACM, 2018.