How much time do you have? Modeling multi-duration saliency

Camilo Fosco\textsuperscript{1}, Anelise Newman\textsuperscript{1}, Pat Sukhum\textsuperscript{2}, Yun Bin Zhang\textsuperscript{2}, Nanxuan Zhao\textsuperscript{3}, Aude Oliva\textsuperscript{1}, and Zoya Bylinskii\textsuperscript{4}

\textsuperscript{1}Massachusetts Institute of Technology, \textsuperscript{2}Harvard University, \textsuperscript{3}City University of Hong Kong, \textsuperscript{4}Adobe Research

\{camilolu, apnewman, oliva\}@mit.edu, \{psukhum, ybzhang\}@g.harvard.edu, nanxuanzhao@gmail.com, bylinski@adobe.com

Abstract

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 produce predictions at an arbitrary, fixed viewing duration, offering a limited view of the rich interactions between image content and gaze location. In this paper we propose to capture gaze as a series of snapshots, by generating population-level saliency heatmaps for multiple viewing durations. We collect the CodeCharts1K dataset, which contains multiple distinct heatmaps per image corresponding to 0.5, 3, and 5 seconds of free-viewing. We develop an LSTM-based model of saliency that simultaneously trains on data from multiple viewing durations. Our Multi-Duration Saliency Excited Model (MD-SEM) achieves competitive performance on the LSUN 2017 Challenge with 57\% fewer parameters than comparable architectures. It is the first model that produces heatmaps at multiple viewing durations, enabling applications where multi-duration saliency can be used to prioritize visual content to keep, transmit, and render.

1. Introduction

How long an observer has to examine an image determines what they notice and what tasks they can complete. Despite this dependency of viewing behavior on viewing time, most models of visual attention predict saliency at an arbitrary duration because they are trained on data accumulated over 3 or 5 seconds of viewing \cite{9, 22, 30, 33, 43}. On the other hand, scanpath models, which predict individual gaze trajectories over time, struggle to summarize the attention patterns of a population \cite{2, 3, 27, 42, 53}.

\textsuperscript{*}Equal contribution.

Figure 1: Predictions of our Multi-Duration Saliency Excited Model at three viewing durations. Images are from the Abnormal Objects \cite{46}, SALICON \cite{33}, and Eye-Crowd \cite{34} datasets (top to bottom). Insets with blue borders contain human ground truth collected using the CodeCharts UI.

In this paper, we introduce the concept of \textbf{multi-duration saliency}, which captures multiple attention snapshots corresponding to different viewing durations (Fig. 1). This offers richer insight into how gaze evolves over time than conventional saliency, while providing a more robust representation than scanpaths (Fig. 2). We leverage an efficient crowdsourcing methodology for collecting large scale human attention data at several viewing durations (Fig. 3). We use it to assemble CodeCharts1K, a dataset of 1000 images with viewing patterns at three durations: 0.5, 3, and 5 seconds. Our data shows that human gaze patterns are highly consistent at each viewing duration but can differ across durations, proving that saliency depends predictably on viewing duration.

\textsuperscript{1}Data, code, and models available at: http://multiduration-saliency.csail.mit.edu/
To capture this dependency, our multi-duration saliency model takes an image as input and predicts three distinct saliency maps for three different durations. This lightweight model achieves competitive performance when evaluated at a single duration, and outperforms baseline models trained to predict multiple durations. We show that the predicted saliency maps can be used as input to applications such as image cropping, compression/rendering, and captioning to tailor them to different contexts based on viewing time.

2. Related work

Crowdsourcing attention: Recent efforts at collecting large-scale attention data have used interfaces that can be run remotely, without an eye tracker [11,33,37,41,55]. Moving-window approaches like SALICON [33] and BubbleView [37] involve using the mouse cursor to inspect small, bubble-shaped regions of blurred images. While providing cheap, high-fidelity data, these interfaces have two downsides: (1) blurring images distorts the visual content and interferes with feature sizes at different resolutions, and (2) moving the cursor requires a different process than moving the eyes, which can impact which image regions are explored [37,52]. The CodeCharts UI [41,49] captures attention without distorting the underlying image or relying on mouse movements. It can account for over 80% of human consistency and outperforms other attention-capturing interfaces, including BubbleView, at approximating eye movements [41]. Because it allows for fine-grained control of image presentation time, we use it to capture multi-duration attention data.

Saliency modeling: The large-scale attention data captured using SALICON [33] and BubbleView [37] enabled training neural network models of saliency (e.g., [11,22,30,43]). The top performers on the MIT Saliency Benchmark [9] were trained on SALICON data and have opened a wide performance gap to the previous, traditional models of saliency [12]. 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 [6,15,24,54].

Scanpath modeling: Ground truth saliency maps are computed by accumulating gaze locations of multiple observers over a fixed viewing duration, which averages out patterns in gaze location over time. 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 [3,39]. 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.

How does multi-duration saliency relate to conventional saliency and scanpaths? We propose an intermediate representation of attention that maintains the robustness of population-level saliency modeling and the temporal resolution of scanpaths (Fig. 2). We introduce multi-duration saliency 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?

3. Collecting multi-duration saliency

In this section, we introduce a scalable approach to measuring multi-duration saliency that uses a web-based interface as a proxy for eye tracking.

CodeCharts UI: In the CodeCharts methodology [41,49], participants view an image for 500-5000 milliseconds followed by a jittered grid of three-character codes (“codechart”). They then self-report the first three-character code they see when the image vanishes (Fig. 3a-c). By construction, participants report the region of the image they were looking at last. The steps in Fig. 3a-d are repeated for dozens of images. The task also contains validation trials consisting of randomly placed cropped faces [5], where we expect the participant to enter a code that overlaps 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 (more details in the supplement). 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.

Pilot experiments: We ran an initial experiment with 50 images from the OSIE dataset [55] to analyze differences in gaze patterns across viewing durations. We collected gaze...
locations for 6 durations: 0.5, 1, 2, 3, 4, and 5 seconds. Attention heatmaps at 0.5, 3, and 5 seconds of viewing were the most distinct from each other (Fig. 4), so we collect further data at these 3 durations. Attention heatmaps at 3 seconds most closely matched the ground truth OSIE data, originally also collected at 3 seconds (Pearson’s CC = .62, averaged over 50 OSIE images). This further validates the ability of CodeCharts data to model natural human gaze.

**CodeCharts1K:** 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 [33], 130 from LaMem [35], 120 from CAT2000 [7], 100 from EyeCrowd [34], 100 from a mix of Abnormal Objects [46] and Out-of-Context Objects [19], and 50 from the Stanford 40K Actions dataset [56]. Images were padded to the same aspect ratio and resized in-browser to fit in a 700 × 1000 pixel window. The task sequence included 6 practice images to pre-screen for attentiveness, 50 dataset images, and 5 validation trials of faces spaced throughout the sequence. 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 (50 participants each at 0.5, 3, and 5 second viewing durations).

**What does CodeCharts measure?** We conducted an analysis to understand what aspect of gaze CodeCharts data captures. We had two hypotheses: CodeCharts approximates either (1) people’s last fixation before the end of the viewing duration, or (2) all fixations within the viewing duration. For 100 images from the CAT2000 Action category, we used the sequential ground truth eye fixations and assumed that they were equally distributed within the 5-second viewing interval. We then generated two sets of ground truth heatmaps by sampling an equal number of fixations according to our two hypotheses. The similarity between ground truth and CodeCharts heatmaps best supported hypothesis 2 (Pearson’s CC of .57 versus .54), which gives us reason to believe that CodeCharts measures what was most salient within the entire viewing interval.

### 4. Data analysis

#### 4.1. Is multi-duration saliency predictable?

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 viewing durations.
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 [8, 51], consistency remains high across the longer viewing durations. These findings hold across the image sets tested (more results in the supplement).

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 viewing durations. This suggests that saliency is predictable at different viewing durations, setting the stage for the computational model in Sec. 5.

4.2. What is salient when?

Things and stuff: We used COCO segmentation maps [13] 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 [12, 14]. For a finer-grained analysis, we ran a face detection network [26] 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. Fig. 5a 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) [31, 47], or the relative suppression of visual cues that were recently attended to. Samuel and Kat [50] 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 (Fig. 5c). 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 (Fig. 5b).

5. 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 (Fig. 6). MD-SEM is the first model that outputs multiple saliency maps corresponding to different viewing durations. The core of our model is a new 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.

5.1. Architecture motivation

Current saliency architectures tend to be bulky, with large numbers of parameters and specialized modules. For instance, the Saliency Attentive Model (SAM) [22] uses an Attentive Convolutional LSTM and a Learned Prior module with several 5x5 convolutions, bringing the total model size to more than 70M parameters. CASNet [25], which applies channel reweighting over 1024 channels, has 142M parameters. We present an accurate model of reduced size and complexity by distilling the required components to a minimum: (1) a strong Xception-based encoder [20], (2) a temporal processing module that operates on a compressed representation, and (3) a simple regularized decoder.

5.2. Convolutional encoder-decoder

Convolutional encoder-decoder architectures are particularly effective for image-to-image tasks like segmentation [4][16][30] and saliency prediction [11][32][33]. Encoding the image allows for rich feature extraction and reduces the dimensionality of the input. We use a state-of-the-art backbone as our encoder: the Xception network [20] pre-trained on ImageNet. The Xception network is lightweight and accurate (0.790 top-1 accuracy on ImageNet with only 22M parameters) and has shown success in semantic segmentation [17]. 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.

5.3. 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 [29], but has not been exploited as a temporal modification tool.

The architecture of the Temporal Excitation Module (TEM) is shown in Fig. 6. 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 ($C$). 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 $f_e(t)$ is computed as:

$$f_e(t) = I_e \cdot s_e(t),$$
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 [22], 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 [22]. In Sec. 6.3, we show that our architecture outperforms SAM.

### 5.4. 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 \( \sigma(y_1) \) and \( \sigma(y_2) \) 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:

\[
L_{CCM}(y_g, y_p) = \frac{1}{T-1} \sum_{t=0}^{T-1} \left( CC(y^{(t)}, y^{(t+1)}_g) - CC(y^{(t)}_p, y^{(t+1)}_p) \right)
\]

where \( y^{(t)}_g \) and \( y^{(t)}_p \) 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 (see the supplement for detailed numbers).

### 5.5. Implementation details

**Architecture:** We remove the last fully connected layer from the Xception decoder in order to obtain a feature map of size \( H \times W \times 2048 \). TEM contains a 512-unit fully connected layer, followed by an LSTM with 512 cells, a ReLU non-linearity, and a sigmoid-activated fully connected layer with 2048 parameters to transform the scaling vector back to its input size. The decoder is composed of 3 sets of convolutional blocks with Dropout. Finally, a 1x1 convolution with 1 filter is used to reduce the final set of feature maps to a single-channel saliency heatmap. Note that the same decoder is applied to each of the \( T \) outputs of TEM, thus concentrating time information exclusively in that module and reducing model complexity.

**Loss:** The network’s loss is a weighted combination of our novel CCM loss, Kullback Leibler divergence (KL), Pearson’s Correlation Coefficient (CC) and Normalized Scanpath Saliency (NSS) (see [10] for formulations). Since NSS is more robust than other metrics at measuring the quality of saliency predictions, we place a higher weight on NSS. We set the weights to 3 for CCM, 10 for KL, -5 for CC and -10 for NSS during SALICON-MD training (Sec. 6.1), but changed the NSS weight to -1 for CodeCharts1K training to account for the reduction in the number of fixations per image.

### 6. Evaluation

#### 6.1. Datasets

For training, we use the SALICON-MD (Multi-Duration) and CodeCharts1K datasets. We created SALICON-MD from the original SALICON dataset [33] by bucketing a participant’s attention locations based on when they occurred. 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 timed-bucketed data serves as an approximate but large pretraining dataset. For final training and evaluation, we use ground truth multi-duration data from CodeCharts1K (introduced in Sec. 3).

#### 6.2. Training details

Our training scheme takes advantage of both datasets to create a model that is generalizable and accurate at every duration. In order to learn from as much data as possible, we pretrain on SALICON-MD. Pretraining on temporal data that exhibits differences across durations is important so that our model learns at the outset to discriminate between timesteps. We then fine-tune on ground truth CodeCharts1K. For both datasets, we set the batch size to 8 and the initial learning rate to 1e-4, which is reduced by a factor of ten every three epochs. At the beginning of training we freeze the weights of the encoder for one epoch. We found that 10 epochs of training on SALICON-MD and 5 on CodeCharts1K was sufficient. For SALICON-MD, we used the provided test, train, and validation splits. For CodeCharts1K, we trained on 70% of the images, validated on 5%, and tested on 25%.

#### 6.3. Comparison to state-of-the-art

**Multi-duration baselines:** 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
Table 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.

<table>
<thead>
<tr>
<th>Model</th>
<th>Params ↓</th>
<th>500ms</th>
<th></th>
<th>3000ms</th>
<th></th>
<th>5000ms</th>
<th></th>
<th>All durations</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CC ↑</td>
<td>NSS ↑</td>
<td>KL ↓</td>
<td>CC ↑</td>
<td>NSS ↑</td>
<td>KL ↓</td>
<td>CC ↑</td>
<td>NSS ↑</td>
</tr>
<tr>
<td>SAM ×3</td>
<td>210.3M</td>
<td>0.804</td>
<td>3.236</td>
<td>0.366</td>
<td>0.693</td>
<td>2.409</td>
<td>0.545</td>
<td>0.706</td>
<td>2.480</td>
</tr>
<tr>
<td>SAM-MD</td>
<td>70.1M</td>
<td>0.805</td>
<td>3.181</td>
<td>0.370</td>
<td>0.738</td>
<td>2.541</td>
<td>0.469</td>
<td>0.715</td>
<td>2.495</td>
</tr>
<tr>
<td>MD-SEM (Ours)</td>
<td>30.9M</td>
<td>0.816</td>
<td>3.374</td>
<td>0.351</td>
<td>0.745</td>
<td>2.694</td>
<td>0.452</td>
<td>0.734</td>
<td>2.677</td>
</tr>
</tbody>
</table>

Table 2: Comparison to state-of-the-art on SALICON test set (LSUN 2017 Challenge).

<table>
<thead>
<tr>
<th>Model</th>
<th>NSS ↑</th>
<th>CC ↑</th>
<th>KL ↓</th>
<th>SIM ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAM-res [22]</td>
<td>1.990</td>
<td>0.899</td>
<td>0.610</td>
<td>0.793</td>
</tr>
<tr>
<td>EML-Net [32]</td>
<td>2.050</td>
<td>0.886</td>
<td>0.520</td>
<td>0.780</td>
</tr>
<tr>
<td>SalNet [44]</td>
<td>1.859</td>
<td>0.622</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CEDNS</td>
<td>2.045</td>
<td>0.862</td>
<td>1.026</td>
<td>0.753</td>
</tr>
<tr>
<td>MD-SEM (Ours)</td>
<td>2.058</td>
<td>0.868</td>
<td>0.568</td>
<td>0.774</td>
</tr>
</tbody>
</table>

Figure 7: **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 8: **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.

Figure 9: **Captioning.** Captions generated by passing saliency-enhanced images to an image captioning model [48], using saliency at different viewing durations to prioritize image content.

Captioning: Image captions can facilitate search and improve accessibility. Some recent work attempts to use a saliency map to guide attention for captioning [21]; however, the saliency model does not explicitly model the temporal aspect of human attention. In Fig. 9, we used our saliency predictions to focus an image captioning model [48] on regions that should stand out at different viewing durations. Removing the non-salient visual clutter can benefit caption quality.

8. Conclusion

Guided by the insight that where you focus on an image depends on how much time you have to explore it, we tackle the problem of predicting multi-duration saliency: saliency as a function of viewing duration. We propose a scalable, crowdsourceable technique 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. We provided initial hints of how multi-duration saliency could be used in applications which require prioritizing visual content, but unfortunately, your viewing time is up.

Acknowledgements: We thank our funding sources: the Vannevar Bush Faculty Fellowship program by the ONR (N00014-16-1-3116) and the SystemsThatLearn@CSAIL / Ignite Grant (to A.O). Work supported in part by cloud credits from the MIT Quest for Intelligence.
References


[30] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. Salicon: Reducing the semantic gap in saliency prediction by


Supplementary Material
How much time do you have? Modeling multi-duration saliency

Camilo Fosco*1, Anelise Newman*1, Pat Sukhum2, Yun Bin Zhang2, Nanxuan Zhao3, Aude Oliva1, and Zoya Bylinskii4

1Massachusetts Institute of Technology, 2Harvard University, 3City University of Hong Kong, 4Adobe Research

{camilolu, apnewman, oliva}@mit.edu, {psukhum, ybzhang}@g.harvard.edu, nanxuanzhao@gmail.com, bylinski@adobe.com

The supplement contains additional details about the CodeCharts1K dataset, the architecture of MD-SEM, and the effectiveness of CCM loss, as well as additional results.

1. CodeCharts Validation Procedure

Our validation procedure for CodeCharts1K was designed to eliminate data from inattentive or noncompliant participants. We follow the validation procedure in [2]. We include validation images consisting of a cropped human face on a plain background, and participants are expected to enter a code that overlaps the validation cue (the face image). We discard data from participants who miss more than 25% of validation images. We also discard data from participants who look at the same spot on an image (within a radius of 100 pixels) for at least five images in a row, to eliminate people who consistently fixate at the same spot. For CodeCharts1K, we collected 50 gaze points per image per duration; after data filtering, we had on average 44 gaze points (we discarded approximately 12% of the data). These validation procedures were implemented for the CodeCharts1K dataset based on observations during our pilot study on the OSIE data, which is why we see lower inter-observer consistency (IOC) on OSIE.

2. Data analysis

2.1. Do people look in the same places?

To judge whether people look at the same image regions at different durations, we ran a split-half consistency analysis using different dataset subsets from CodeCharts1K. For each image and viewing duration, we generate 10 splits of participant data by resampling gaze points for a given duration. We use these gaze points to generate a heatmap and use it to compute the Pearson’s Correlation Coefficient (CC) score between heatmaps at different durations. The final numbers in Fig. 1 are produced by averaging over im-

![Figure 1. Split-half consistency of viewers, computed using Pearson’s Correlation Coefficient (CC), within and across viewing durations on different subsets of CodeCharts1K (a-g) and on the full dataset (h).](image-url)
Figure 2. People’s gaze falls on different things and “stuff” in images over time. From 0.5 sec to 3 sec, gaze frequently moves away from people and is pulled towards objects and furniture. From 3 to 5 sec, gaze on contextual “stuff” like grass, carpet, pavement, and road increases. COCO segmentations [1] were used for this analysis. An object in an image was assigned a score of +1, -1, or 0 depending on whether it increased, decreased, or didn’t change in saliency from one duration to the next. These scores were added up over all the images where the objects occurred to produce the attention scores plotted on the y-axis. Example images containing patterns of gaze change over time for the people, road, and tv object categories are included above.

2.2. What is salient at what time?

We used COCO segmentation maps [1] available for the SALICON images in our dataset to compute gaze counts per image segment across time (Fig. 2). We compute an “attention score” per object by giving the object a score of +1 every time it increases in saliency from one viewing duration to the next, a score of -1 if it decreases, and 0 otherwise. We sum these scores across the images in our dataset. The benefit of this score is avoiding image-specific saliency scale differences. We find that from 0.5 sec to 3 sec, gaze frequently moves away from people and is pulled towards objects and furniture (e.g., TVs, tables, bottles, chairs, books, etc.). From 3 to 5 sec, there is an increase of attention on contextual “stuff” like roads, walls, windows that may contain other objects. At these longer durations people notice smaller and more distant objects in an image.

Table 1. The architecture of temporal excitation module.

<table>
<thead>
<tr>
<th>Input</th>
<th>Dense 2048</th>
<th>LSTM 512</th>
<th>Dense 512</th>
<th>Sigmoid 2048</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAP</td>
<td>2048</td>
<td>512</td>
<td>512</td>
<td>2048</td>
</tr>
</tbody>
</table>

3. Model architecture

Tables 1 and 2 lay out the architecture of the custom modules of MD-SEM. Table 1 covers the Temporal Excitation Module and Table 2 covers the decoder.

4. Effectiveness of CCM Loss

The Correlation Coefficient Match (CCM) loss explicitly encourages our network to model temporal differences in saliency data. Table 3 shows how adding CCM to our loss function improves the performance of MD-SEM and SAM-MD on CodeCharts1K.

5. Additional evaluations and predictions

Fig. 3 compares predictions from MD-SEM to other models (SAM-MD and SAMx3). Fig. 4 shows representative predictions of MD-SEM on various datasets.
<table>
<thead>
<tr>
<th>Kernel</th>
<th>Stride</th>
<th>Dialation</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv. 3 x 3</td>
<td>1 x 1</td>
<td>2 x 2</td>
<td>256</td>
</tr>
<tr>
<td>Conv. 3 x 3</td>
<td>1 x 1</td>
<td>2 x 2</td>
<td>256</td>
</tr>
<tr>
<td>Upsample</td>
<td>-</td>
<td>2 x 2</td>
<td>- 256</td>
</tr>
<tr>
<td>Dropout (0.3)</td>
<td>-</td>
<td>-</td>
<td>- 256</td>
</tr>
<tr>
<td>Conv. 3 x 3</td>
<td>1 x 1</td>
<td>2 x 2</td>
<td>128</td>
</tr>
<tr>
<td>Conv. 3 x 3</td>
<td>1 x 1</td>
<td>2 x 2</td>
<td>128</td>
</tr>
<tr>
<td>Upsample</td>
<td>-</td>
<td>2 x 2</td>
<td>- 128</td>
</tr>
<tr>
<td>Dropout (0.3)</td>
<td>-</td>
<td>-</td>
<td>- 128</td>
</tr>
<tr>
<td>Conv. 1 x 1</td>
<td>1 x 1</td>
<td>1 x 1</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2. The architecture of the decoder.

<table>
<thead>
<tr>
<th>Model</th>
<th>NSS ↑</th>
<th>CC ↑</th>
<th>KL ↓</th>
<th>SIM ↑</th>
<th>CCM ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAM-MD w/o CCM</td>
<td>2.700</td>
<td>0.744</td>
<td>0.434</td>
<td>0.616</td>
<td>0.231</td>
</tr>
<tr>
<td>SAM-MD w/ CCM</td>
<td>2.739</td>
<td>0.753</td>
<td>0.458</td>
<td>0.609</td>
<td>0.198</td>
</tr>
<tr>
<td>MD-SEM w/o CCM</td>
<td>2.778</td>
<td>0.754</td>
<td>0.565</td>
<td>0.598</td>
<td>0.228</td>
</tr>
<tr>
<td>MD-SEM w/ CCM</td>
<td>2.915</td>
<td>0.765</td>
<td>0.430</td>
<td>0.620</td>
<td>0.195</td>
</tr>
</tbody>
</table>

Table 3. 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.

Table 4. MD-SEM results on the different sub-datasets that compose CodeCharts1K.

6. Applications

Fig. 5 contains examples of how multi-duration saliency heatmaps can be used to crop parts of an image that are salient at different times. Fig. 6 shows how multi-duration saliency can be used to select which elements in an image should be rendered first (or at higher resolution). Fig. 7 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.

References

Figure 4. 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 5. 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 6. 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 7. 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.