How Many Glances? Modeling Multi-duration Saliency

Anonymous ICCV submission

Paper ID 5002

Abstract

What jumps out at you in a single glance of an image is different than what you might notice after closer inspection. Despite this, models of visual saliency have ignored the temporal aspect of visual attention and have produced prediction maps at fixed viewing durations. As a result, current applications leveraging saliency models are rigidly tailored for a fixed viewing duration, depending on the attention dataset they were trained on. To incorporate knowledge of viewing duration into saliency modeling, we first develop a "Codecharts UI" and use it to crowdsource human attention at various viewing durations. We collect the CodeCharts1K dataset, which contains viewing data at 0.5, 3, and 5 seconds on 1000 images from diverse computer vision datasets. Our analysis shows distinct differences in gaze locations at these time points and exposes recurring temporal patterns about which objects attract attention. Using insights from this analysis, we develop an LSTM-based model of saliency that simultaneously trains on data from multiple time points. Our Multi-Duration Saliency Excited Model (MD-SEM) achieves state-of-the-art performance on the LSUN 2017 Challenge with 57% fewer parameters than comparable architectures. Additionally, it provides new predictions at time points that current models can not predict. We explore applications where multi-duration saliency can be used to prioritize the visual content to keep, transmit, and render.

1. Introduction

How long an observer examines an image determines what they notice and what tasks they can complete. Despite this dependency of viewing behavior on time, most models of visual attention predict saliency at a fixed duration, e.g., by training on data collected with a viewing time of 3 or 5 seconds per image[1] [5, 18, 27, 25, 34].

In this paper, we introduce the first saliency model that outputs multiple saliency maps based on the viewing duration of an observer. We propose an efficient crowdsourcing methodology that allows us to collect at large scale human attention data at several viewing durations. We use it to assemble CodeCharts1K, a dataset of 1000 images with viewing patterns at three durations (0.5, 3, and 5 seconds) encompassing a variety of image genres such as actions, out-of-context objects, and memorable images. We find that humans are highly consistent with each other at a specific viewing duration but show significant differences across durations.

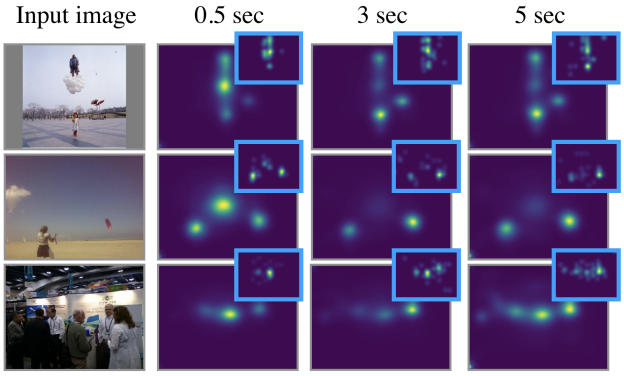

Figure 1. Predictions of our Multi-Duration Saliency Excited Model for images at viewing durations of 0.5, 3, and 5 seconds. Images here are from the Abnormal Objects [37], SALICON [27], and EyeCrowd [28] datasets (top to bottom row). Insets with blue borders contain human ground-truth gaze locations collected using our CodeCharts UI.

We then introduce an original multi-duration saliency model which, given an image as input, predicts saliency maps for three different durations. Our model achieves state-of-the art performance when evaluated at a single duration and outperforms other baseline models if they are trained to predict multiple durations. We show that the predicted saliency maps can be used as input to applications such as image cropping, compression and rendering, and captioning, to tailor them to different contexts based on time.

[1] See http://saliency.mit.edu/datasets.html

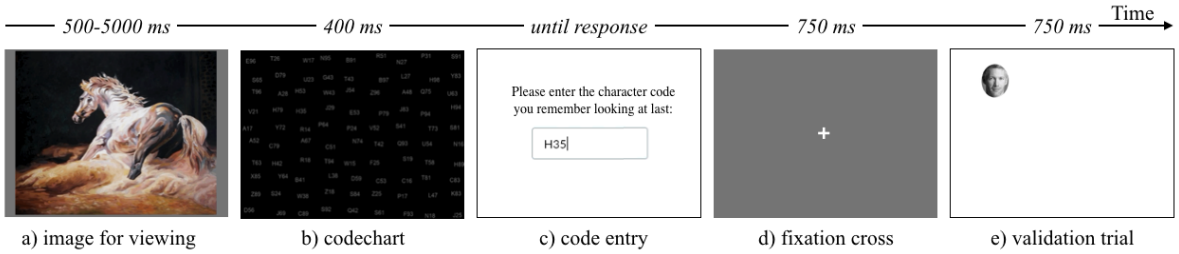

Figure 2. CodeCharts UI task flow: participants view an image for a variable amount of time (a), after which a codechart is briefly flashed on the screen (b). Participants are asked to report the character code they remember looking at last (c), as an indication of their gaze on the image. This process proceeds for a sequence of images, separated by fixation crosses (d) to re-center gaze. Validation trials (e) are interspersed among the experimental sequence to check if participants move their gaze as expected.

2. Related work

Recent efforts have engaged in collecting large-scale attention data using interfaces that can be run remotely, without an eye tracker [7, 27, 31, 32, 36]. Webcam-based approaches [32, 36] rely on good visibility and stability of the viewer in front of the web camera and raise privacy concerns. Such approaches, trained to predict gaze location from images of the eyes, are sufficient for coarse estimates of gaze location but have error rates that do not permit good localization. On the other hand, moving-window approaches like SALICON [27] and BubbleView [31] allow people to use their cursors to inspect small, bubble-shaped regions of blurred images. Data collected is pixel-accurate but has two downsides: (1) blurring images distorts the visual content and interferes with feature sizes in an image, and (2) moving the mouse cursor requires a different process than moving the eyes, which can impact which image regions are explored [31]. The CodeCharts UI is a response to these concerns. This approach does not require webcams nor does it distort the image, and it allows for fine-grained control of image presentation time. We exploit this latter property to capture multi-duration attention data.

The large-scale attention data captured using the SALICON [27] and BubbleView [31] methods has enabled training neural network models of saliency (e.g., [7, 18, 25, 34]). The top performers on the MIT Saliency Benchmark [5] have been trained on SALICON data, and have opened a wide performance gap to the previous, non neural network based models of saliency [8]. 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 [3, 11, 20, 40].

3. Dataset of multi-duration attention

We introduce an approach to capture human gaze data with a user interface that does not require explicit eye tracking but successfully captures the same attention patterns. Importantly, this interface allows easily manipulating the image presentation time for large-scale collection of multi-duration attention data.

3.1. CodeCharts UI

Task flow: We revisit the user interface of Rudoy et al. [39], which introduced the idea of using a character chart to validate self-reported gaze locations. In our task, participants view an image for 500-5000 milliseconds (ms), followed by a jittered grid of random, three-character codes ("codechart"). They then self-report the first three-character code they see when the image vanishes (Fig. 2a-c). Based on pilot experiments, the codechart duration is set as short as possible to maintain accurate performance. By construction, participants report the region of the image they were looking at last. We repeat the steps in Fig. 2a-c for dozens of images, shown in sequence in a single experiment. A fixation cross between trials helps to re-center gaze (Fig. 2d).

Validation: To ensure good quality data, we check that the three-character codes entered by participants are "valid", i.e., they occur in the corresponding codechart. We also insert validation trials throughout the image sequence: cropped faces [2] randomly placed on the image canvas. A code is marked "correct" if it overlaps with some portion of the face in the corresponding codechart. A pre-test phase is used to screen participants based on their performance entering valid codes for regular images and correct codes for face images. Our validation procedure has a significant impact on collected data quality (see Supplement).

3.2. Data collection

Pilot experiments: We ran an initial experiment with images from the CAT2000 dataset [4] to determine the optimal task flow (timing of task components, validation procedure, number of participants, etc.). We then sampled 50 images from the OSIE dataset [41] and collected the gaze locations of 50 participants per image for each of 6 image durations: 0.5, 1, 2, 3, 4, and 5 seconds. We found that the gaze maps obtained at 0.5, 3, and 5 seconds were the most different from each other (Fig. 3a), which motivated our de-

cision to collect further data at these 3 durations. The gaze locations collected with our CodeCharts UI at 3 seconds most closely matched the ground-truth OSIE data, which was originally also collected at 3 seconds, with a Pearson's Correlation Coefficient (CC score) of .71. This validates the ability of CodeCharts data to model natural human gaze.

CodeCharts1K: For our data collection, we sampled a variety of image types to better understand how attention evolves with time. We used 500 images from SALICON [27], 130 from LaMem [30], 120 from CAT2000 [4][2], 100 from EyeCrowd [28], 100 from a mix of Abnormal Objects [37] and Out-of-Context Objects [15], and 50 from the Stanford 40K Actions dataset [43][3]. Images were padded to have the same aspect ratio, and were resized in-browser to fit in an image window of 700×1000 pixels. The task sequence included 6 "tutorial" images to test participant 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 seconds).

4. Data analysis

4.1. Do people look in the same places?

To determine whether participants look in the same locations, we perform a split-half consistency analysis. We compare the gaze points of one half of the participants to the gaze points of the other half, by converting both sets of gaze points to heatmaps and computing a Pearson's Correlation Coefficient (CC score) between them. We perform this analysis per viewing duration. By performing this analysis also *across* viewing durations, we measure whether the gaze patterns are different for different viewing durations.

From the diagonal entries in Fig. 3, we see that the split-half consistency between participants is very high across all durations. This finding replicates across all the image sets we tested (more results in the Supplement). Importantly, however, **consistency does not degrade over time**: the differences between the diagonal entries are not statistically significant according to a two-sided T-test ($p > 0.05$ for all pairwise comparisons). In other words, **saliency is equally predictable across all the viewing durations**.

From the off-diagonal entries in Fig. 3, we see that the gaze locations of participants viewing an image at one duration are different from the gaze locations of participants viewing an image at a different duration. These differences are statistically significant for all pairs of durations ($p < 0.001$, Bonferonni-corrected), with a replication

[2]We used 100 images from the 'Action' [43] and 20 from the 'Low Resolution' [29] categories.

[3]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.

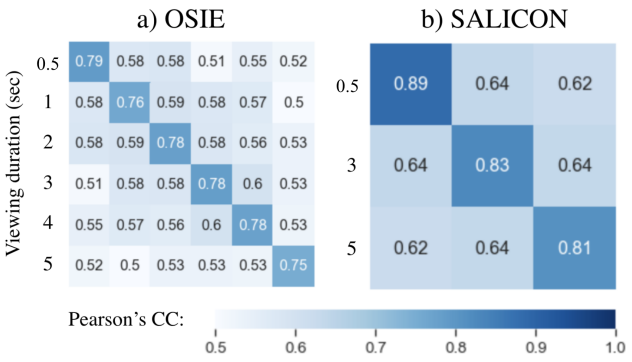

Figure 3. Split-half consistency of viewers across datasets and viewing durations (using Pearson's Correlation Coefficient). Participants are remarkably consistent in where they look. This consistency persists across different viewing durations, and replicates across datasets. Consistency is much higher between participants viewing images at the same duration, than across durations, indicating that the saliency patterns are different across durations.

across datasets. In other words, **different things are salient at different viewing durations.**

These analyses indicate that gaze data collected using our CodeCharts UI (i) contains a consistent signal at each of the viewing durations, (ii) this signal is equally strong at all the viewing durations, and (iii) the signal is significantly different between viewing durations. These findings set the stage for the computational model presented in Sec. 5.

4.2. What is salient at what time?

Things and stuff: We used COCO segmentation maps [9] of the SALICON images to compute gaze counts per image segment across time (full analysis in Supplement). From 0.5 sec to 3 sec, gaze frequently moves away from people and is pulled towards objects and furniture (e.g., paper, bottle, table). From 3 to 5 sec, there is an increase of 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 in an image.

Faces: It is known that gaze is attracted by faces [8, 10]. For a finer-grained analysis over time, we ran a face detection network [21] over all the images in CodeCharts1K. For the 303 images on which faces were detected, we computed face saliency by aggregating gaze counts per face region and then normalizing both by the number of gaze points per image and across all 3 durations. This ensures that for a given image across time, face saliency ranges between 0 and 1. Fig. 4 plots these results as one line per image, where thick lines are averages to visualize the dominant patterns. Across our CodeCharts1K dataset, we find a dominant "boomerang" pattern: people start out by looking at faces at 0.5 sec, their gaze shifts elsewhere at 3 sec, and

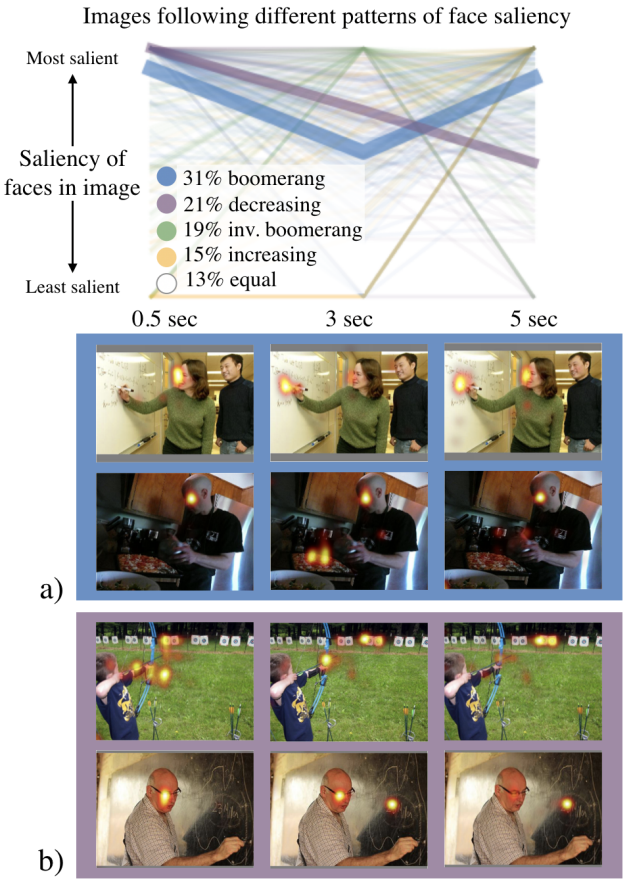

Figure 4. Dominant patterns of human gaze on faces across time: (a) face saliency decreases from 0.5 to 3 sec, and increases again from 3 to 5 sec ("boomerang"), and (b) face saliency decreases from 0.5 to 5 sec. Each line in the graph above represents how the saliency of faces within a single image varies over time, while the thicker lines represent an average over images. We normalize the saliency of faces across time on a per-image basis in this plot.

returns to faces at 5 sec. The second most prevalent pattern is a decrease in gaze on faces over time. Other patterns, like an increase in face saliency over time, were in the minority.

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. 4b). In other cases, saliency at 5 sec is a combination of saliency at the shorter durations: we hypothesize that in these cases, people's attention is initially grabbed by faces, before they explore an image and return to the most salient regions (Fig. 4a).

5. Multi-duration saliency model

In order to predict these changes in human attention over time, we introduce the Multi-Duration Saliency Excited Model (MD-SEM), a new architecture adapted to multi-duration saliency. Our model is capable of producing saliency maps at T different viewing durations (where T=3 in our implementation). The architecture is based on two concepts: (1) a powerful encoder-decoder architecture built on an ImageNet-pretrained Xception network [16]; (2) a Temporal Excitation Module, a novel block that applies a time-based re-weighting to saliency feature maps with a minimal increase in parameters. Our model achieves first place on the LSUN 2017 SALICON saliency challenge, and is the current state-of-the-art on CodeCharts1K, our multi-duration saliency dataset.

5.1. Architecture motivation

The current state-of-the-art saliency models are built on convolutional neural networks, exploiting their ability to understand semantic components of images to produce accurate saliency maps. Current architectures tend to be bulky, with large numbers of parameters and specialized modules. One of the current top performers, EML-NET [26], contains both a DenseNet and NASNet. Another top performer, the Saliency Attentive Model (SAM) [18], uses an Attentive Convolutional LSTM for refinement as well as a Learned Prior module with several 5x5 convolutions, bringing the total model size to more than 70 million parameters. We present a model of reduced size and complexity that nevertheless achieves an increase in accuracy.

We simplify saliency prediction by distilling the required components to the bare minimum: (1) a strong encoder, (2) a processing module on the compressed representation, and (3) a regularized decoder. Our multi-duration saliency network obeys these three constraints: we use a parameter-efficient encoder, a concise temporal re-weighting module (which explicitly performs computations on vectors with reduced dimensionality), and a regularized decoder composed of 3 sets of dilated convolutions with Dropout.

5.2. Convolutional encoder-decoder

Convolutional encoder-decoder architectures have been extensively used, being particularly effective for image-to-image tasks like segmentation [1, 12, 33], saliency prediction [26, 34] and importance prediction [7]. Encoding the image input through a convolutional encoder allows for rich feature extraction and reduces the dimensionality of the input to values that facilitate feature manipulation. Decoding the generated representation through convolutional layers can be thought of as learned upsampling. Given the necessity of deep semantic understanding, we use a state-of-the-art network in image recognition as our encoder: the Xception network [16], pretrained 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 [13], a related task. It proposes a complete decoupling of cross-channel and spatial

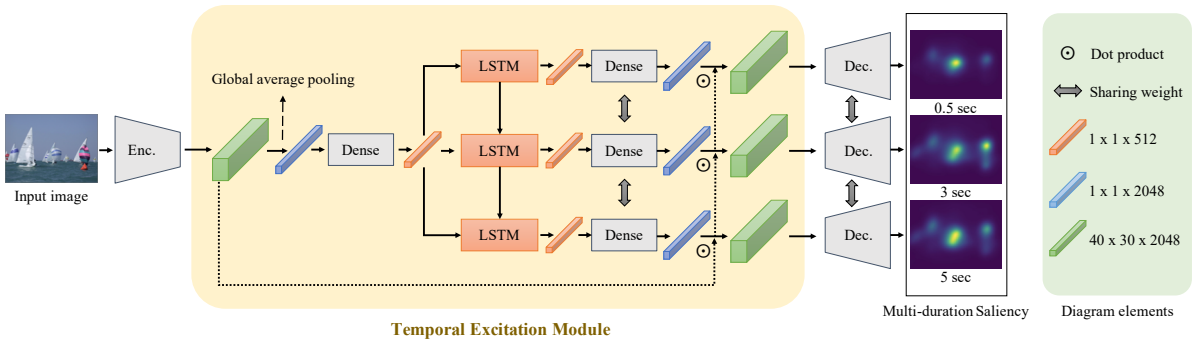

Figure 5. Multi-Duration Saliency Excited Model (MD-SEM) architecture.

correlations of feature maps. Its convolutional layers perform depthwise-separable convolutions, operating on each channel separately before merging the transformed channels through a pointwise (1x1) convolution.

5.3. Temporal Excitation module

In order to predict saliency across durations, we introduce a module capable of recursively manipulating the feature representation generated by the encoder to adapt it to each timestep. Our module utilizes a Long Short Term Memory (LSTM) network to generate scaling vectors that re-weight the feature maps differently for each timestep. Feature map re-weighting is a concept that has been explored before in the form of Squeeze and Excitation Modules [24], but has not been exploited as a temporal modification tool. A previous top-performing saliency model, SAM [18], also contains an LSTM, but only for internal map refinement, rather than for capturing the temporal aspect of saliency. The main advantage is simplicity: whereas SAM utilizes a parameter-heavy Attentive Convolutional LSTM that operates on full feature maps, we approach the problem with a simple LSTM that works on a squeezed low-dimensional vector obtained from pooling those maps. In Sec. 6.3, we show that our architecture performs better than SAM on predicting multi-duration saliency.

The architecture of the Temporal Excitation Module is included in Fig. 5. First, the feature maps generated by the encoder are pooled through global average pooling, generating a 1x1xC vector. The vector is then passed through a dense (fully connected) layer, which reduces the dimensionality of the vector and aids in generalization. As explained in [24], applying this type of operation is key to modeling channel-wise relationships. The output of the dense layer is replicated T (in our case, 3) times and fed as a sequence to the LSTM. This is the key of the Temporal Excitation module: the LSTM outputs T vectors, which contain information specific to each timestep and will scale each feature map differently. A ReLU non-linearity is applied before passing the vectors through a dimensionality-increasing dense layer. Finally, a sigmoid non-linearity ensures that the scaling weights s_c remain in a sensible range, following insights by [24]. The block outputs a set of T feature maps, which are rescaled according to each one of the weight vectors generated. Formally, the block outputs T sets of feature maps $F^{(t)}$, where each feature map $f_c^{(t)}$ is computed as:

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

where I_c are the c input feature maps, $f_c^{(t)}$ are the T sets of c feature maps, and $s_c^{(t)}$ are the scaling weights generated by the previously defined operations.

The Temporal Excitation module introduces few new parameters to the model. With the choice of hyper-parameters described in Sec. 5.4, we obtain an architecture with 30 million parameters, 57% smaller than SAM [18]. This allows us to further reduce model complexity and overfitting.

5.4. Implementation details

Architecture: Our network's blocks are implemented as follows. The Xception decoder is instantiated with its original definition, removing the last fully connected layer, and obtaining a decoder that outputs an $H \times W \times 2048$ tensor of feature maps. The Temporal Excitation Module is instantiated with a 512-wide 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. Finally, the decoder is composed of 3 sets of convolutional blocks. The first block contains two convolutional layers with 256 filters each, followed by a 2x2 upsampling layer and a dropout operator with probability 0.3. The second and third blocks are identical, but use 128 and 64 filters each on their convolutions. Finally, a 1x1 convolution with 1 filter is used to get the final set of feature maps to a single-channel saliency heatmap. We set T=3 to match the data available in the CodeCharts1K dataset. Note that the same decoder is applied to each one of the T outputs of the Temporal Excitation Module, thus concentrating time information exclu-

sively in that module, and reducing model complexity. A description of the architecture is in the Supplement.

Core loss: Following previous works, the network's core loss is defined as a weighted combination of Kullback Leibler divergence (KL), Linear Correlation Coefficient (CC) and Normalized Scanpath Saliency (NSS) (see [6] for details on the formulas). As discussed in [6], NSS is more robust than other metrics at measuring the quality of saliency predictions. Thus, we defined the weights of the network with a higher emphasis on NSS. After evaluation, we defined the weights to be 10 for KL, -5 for CC and -10 for NSS during SALICON-MD training (see subsection 6.1 for details on the dataset), but changed the NSS weight to -1 for CodeCharts training to account for the reduction in the number of fixations per image.

CC Match: To ensure that our network is correctly modeling the differences across durations, we introduce a novel training loss called Correlation Coefficient Match (CCM). This loss forces the network to output saliency maps that are similar to the ground-truth saliency maps at each duration yet different across durations. Given a set of T 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. Let $y^{(j)}$ be the heatmap corresponding to duration j where CC is defined as:

$$CC(y_1, y_2) = \frac{\sigma(y_1, y_2)}{\sigma(y_1) \cdot \sigma(y_2)}, \tag{1}$$

where $\sigma(y_1, y_2)$ is the covariance of y_1 and y_2. Our CCM loss is:

$$L_{CCM}(y_g, y_p) = \frac{1}{T-1} \sum_{j=0}^{T-1} \Big| CC\Big(y_g^{(j)}, y_g^{(j+1)}\Big) - CC\Big(y_p^{(j)}, y_p^{(j+1)}\Big) \Big| \tag{2}$$

where $y_g^{(t)}$ and $y_p^{(t)}$ are the ground-truth and predicted saliency maps for time t. Our experiments showed that a weight of 3 is appropriate for this loss. We show in Table 1 that this novel loss significantly boosts our performance when evaluating multi-duration predictions.

6. Evaluation

6.1. Datasets

For training, we rely on the SALICON-MD (Multi-Duration) and CodeCharts1K datasets. SALICON-MD is derived from the original SALICON dataset by bucketing a participant's fixations based on when they occurred. We assume that fixations occur at a constant interval across the viewing duration (from 0 to 5 sec) and split them into 6 non-overlapping buckets. This time-separated data serves as an approximate but large pretraining dataset. For final training and evaluation, we use ground-truth multi-duration data from CodeCharts1K, defined in Sec. 3.2.

Model	*NSS* ↑	*CC* ↑	*KL* ↓	*SIM* ↑	*CCM* ↑
SAM-MD w/o CCM	2.700	0.744	0.434	0.616	0.041
SAM-MD w/ CCM	2.765	**0.778**	0.401	**0.641**	0.061
MD-SEM w/o CCM	2.778	0.754	0.565	0.598	0.073
MD-SEM w/ CCM	**2.875**	0.773	**0.392**	0.633	**0.091**

Table 1. MD-SEM results on CodeCharts1K with and without CCM loss. We report performance on NSS, CC, KL, SIM and our custom CC Match loss. NSS and CC are effective saliency metrics (they are symmetric to false positives and false negatives [6]). A lower value is better for KL.

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, which forces the Temporal Excitation Module to make use of the ImageNet-pretrained feature maps from the encoder instead of distorting them. We find that 10 epochs of training on SALICON-MD and 5 on CodeCharts1K is sufficient.

For SALICON-MD, we use the pre-established test, train, and validation sets. For the code charts data, we train on 70% of the images, validate on 5%, and test on 25%. Our reported test results for CodeCharts1K are the average scores from cross-validating on four splits of the data.

6.3. Comparison to state-of-the-art

On CodeCharts1K, our final model achieves an NSS of 2.875 and CC of 0.773 averaged across the three durations (Table 2). This is compared to an average ground-truth human score of 0.843 on the same dataset (Fig. 3). In addition to accurately capturing gaze location at a single timestep, our high CCM score (see Table 1) indicates that our network effectively captures the variability of human gaze across durations. Our model reflects many of the qualitative patterns of human gaze movement that stand out in the code charts data, such as the boomerang effect and the tendency of humans to focus on the object of an action (see Fig. 6).

Our model is first-of-its-kind in its ability to predict multi-duration saliency. Thus, we benchmark against the

Model	*NSS* ↑	*CC* ↑	*KL* ↓	*SIM* ↑	*Params* ↓
SAM x3	2.020	0.803	0.330	**0.708**	210.3M
SAM-MD	2.057	0.792	0.338	0.695	70.1M
MD-SEM	**2.061**	**0.811**	**0.324**	**0.708**	**30.9M**

Table 2. Comparison to state-of-the-art models on SALICON-MD. Our model marginally outperforms SAM-MD (SAM adapted by ourselves for multi-duration prediction) with 57% less parameters. We also report improvements on SAM trained on each timestep separately (SAM x3).

Model	*NSS* ↑	*CC* ↑	*KL* ↓	*SIM* ↑
SAM-res [18]	1.990	**0.899**	0.610	**0.793**
EML-Net [26]	2.050	0.886	**0.520**	0.780
SalNet [35]	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 3. Comparison to state-of-the-art on SALICON test set (LSUN 2017 Challenge).

currently-available option for generating saliency at multiple durations: training T separate copies of a standard saliency network, each on one of T splits of multi-duration saliency data. Furthermore, in order to prove that our architecture is uniquely effective at the multi-duration problem, we also benchmark against a modified version of SAM. In SAM-MD, the LSTM is modified to produce a map at each timestep, and during training these maps are compared to the ground-truth at the corresponding duration. The results of these benchmarks are in 2. Not only is MD-SEM better at approximating human gaze and differentiating across durations, but it also uses significantly fewer parameters than the other baselines.

Finally, we show that our model performs at state-of-the-art for traditional single-duration saliency by achieving first place on the LSUN 2017 challenge (Table 3).

7. Applications

As saliency neural network models approach human-level prediction, saliency has been used for many applications including cropping, retargeting, and image captioning. Our multi-duration saliency model can bring additional context by accounting for the expected time that a viewer may have to explore an image.

Cropping: Automatic cropping of images is useful for image thumbnails, view-finding for improved composition, and retargeting for different use cases [19]. Multi-duration saliency allows us to additionally take into account the expected time a viewer will spend on an image, and retarget the image accordingly (e.g., an image that is part of a passing advertisement should be simpler than if it is the main image on a page). In Fig. 7 we used our multi-duration saliency maps to crop windows [14] within an image that capture 90% of the most salient image regions *at that viewing duration*. Our automatically-generated thumbnails tend to contain close-ups of content for shorter durations.

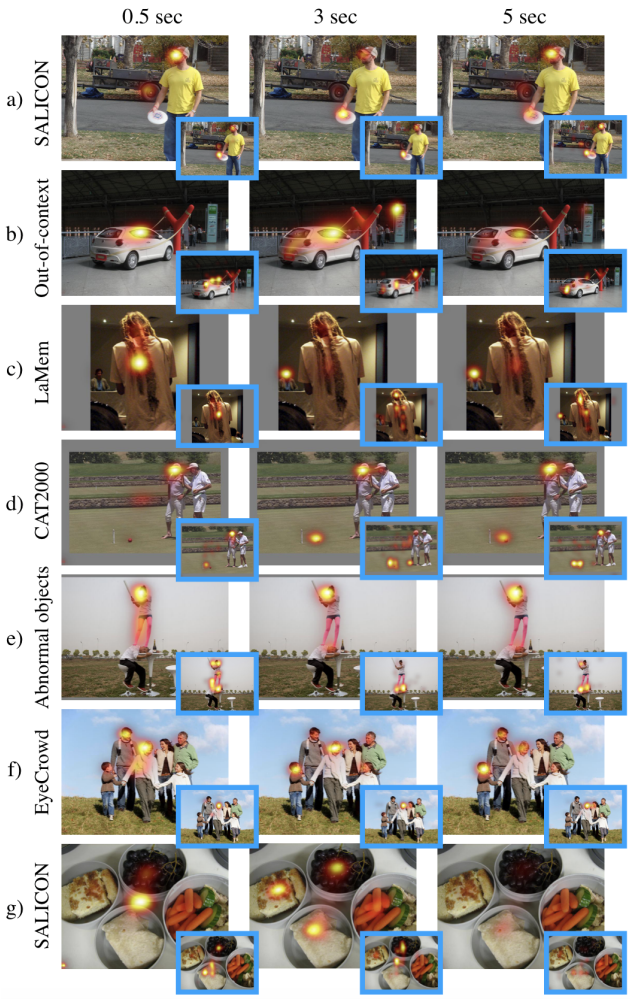

Figure 6. Saliency predictions of MD-SEM on various datasets. Insets with blue borders contain human ground-truth gaze locations collected using our CodeCharts UI. a) The "boomerang" pattern of saliency: starts out at face (0.5 sec), moves to object (3 sec), and back to face (5 sec). b) The boomerang pattern applied to objects: gaze starts at central object, moves to new salient location, then back to initial object. c) More distant objects (especially faces) become the objects of focus at later durations. d) Saliency distributes over time from faces to multiple scene elements. Difficult cases for our model: e) Our model predicts faces will be salient for all durations, but people focus on the abnormal region of the scene (woman standing on man's head). f) Model knows to change focus of saliency from one face to another between durations, but chooses the wrong face to focus on relative to ground-truth. g) Model distributes saliency over too many objects in a scene when it's not clear which one should be the focus.

Compression and rendering: Just as saliency has been

Figure 7. **Cropping.** Images cropped based on viewing time by selecting the window with 90% of the most salient image regions, as predicted by our multi-duration saliency model. Automatically-computed image crops for shorter viewing durations tend to contain close-ups of image content, focusing on fewer elements.

used as a mechanism to prioritize the visual content to preserve and render, multi-duration saliency can add a temporal aspect to these applications. For instance, if an image is expected to be viewed for shorter periods of time, fewer visual elements need to be rendered (or preserved, in the case of compression). In Fig. 8 we provide a visualization of which visual content would be prioritized at different viewing durations for such applications. To generate these visualizations, we use Mask R-CNN for instance segmentation on the image [22]. The detected instances are then mapped to the saliency predictions from our model. Rather than using the saliency directly at a given time point, we accumulate the saliency across time. Instances that have a mean saliency score in the 90-th percentile are kept, using saliency at 0.5 sec as a reference, and the rest of the image is blurred and darkened, for visualization purposes.

Captioning: Captions can be used to facilitate search and improve accessibility. Despite captioning being tied to a sequential viewing of the image, current approaches do not leverage this information to produce better captions. Some approaches have used an attention mechanism to attend to different parts of the image [42], yet machine attention does a worse job of approximating human attention than saliency models when performing captioning [23]. While some recent work attempts to produce an attention map at each timestep for captioning [17], the saliency model used does not explicitly model the temporal aspect of human attention. Our multi-duration saliency maps offer a closer approximation to how humans view images and provide an opportunity to focus . Here, we used our saliency

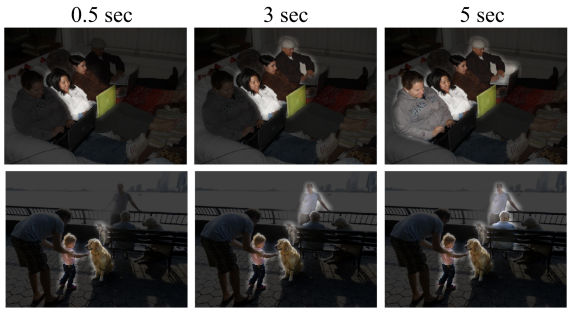

Figure 8. **Compression.** The more you look, the more you see. Visualized are image regions that are predicted to attract gaze at different viewing durations (accumulated over time). A better understanding of saliency across time can facilitate saliency-driven applications like image compression, transmission, and rendering to take viewing duration into account. Will the end user have less than 1 second to look at an image? Saliency can be used to prioritize the visual content to retain, transmit, and render.

Figure 9. **Captioning.** Captions generated by passing saliency-enhanced images to an image captioning model [38], using saliency at different durations to prioritize image content.

predictions to focus an image captioning model [38] on image regions that should stand out at different viewing durations. Removing the non-salient visual clutter indeed produces promising initial results (Fig. 9).

8. Conclusion

Guided by results on how time affects people's focus of attention, we introduce an LSTM-based model of saliency which, when trained on crowdsourced gaze data from different viewing times, achieves state-of-the-art performance, while providing novel predictions at time points that other saliency approaches cannot model. Importantly, we show that the time an observer, or a system, spent on an image determine which tasks they can complete. We demonstrate the direct impact of our multi-duration saliency approach to applications which require prioritizing the visual content for image cropping, compression and rendering and captioning.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 4

[2] W. A. Bainbridge, P. Isola, and A. Oliva. The intrinsic memorability of face photographs. *Journal of Experimental Psychology: General*, 142(4):1323, 2013. 2

[3] A. Borji. Saliency prediction in the deep learning era: An empirical investigation. *arXiv preprint arXiv:1810.03716*, 2018. 2

[4] A. Borji and L. Itti. CAT2000: A large scale fixation dataset for boosting saliency research. *CVPR'15 Workshop on the Future of Datasets*, 2015. 2, 3

[5] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba. Mit saliency benchmark. 1, 2

[6] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand. What do different evaluation metrics tell us about saliency models? *IEEE transactions on pattern analysis and machine intelligence*, 41(3):740–757, 2019. 6

[7] Z. Bylinskii, N. W. Kim, P. O'Donovan, S. Alsheikh, S. Madan, H. Pfister, F. Durand, B. Russell, and A. Hertzmann. Learning visual importance for graphic designs and data visualizations. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 57–69. ACM, 2017. 2, 4

[8] Z. Bylinskii, A. Recasens, A. Borji, A. Oliva, A. Torralba, and F. Durand. Where should saliency models look next? In *European Conference on Computer Vision*, pages 809–824. Springer, 2016. 2, 3

[9] H. Caesar, J. Uijlings, and V. Ferrari. Coco-stuff: Thing and stuff classes in context. In *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018. 3

[10] M. Cerf, E. P. Frady, and C. Koch. Faces and text attract gaze independent of the task: Experimental data and computer model. *Journal of vision*, 9(12):10–10, 2009. 3

[11] J. Chen, G. Bai, S. Liang, and Z. Li. Automatic image cropping: A computational complexity study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 507–515, 2016. 2

[12] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018. 4

[13] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018. 4

[14] Y.-L. Chen, T.-W. Huang, K.-H. Chang, Y.-C. Tsai, H.-T. Chen, and B.-Y. Chen. Quantitative analysis of automatic image cropping algorithms:a dataset and comparative study. In *IEEE WACV 2017*, 2017. 7

[15] M. J. Choi, A. Torralba, and A. S. Willsky. Context models and out-of-context objects. *Pattern Recognition Letters*, 33(7):853–862, 2012. 3

[16] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 4

[17] M. Cornia, L. Baraldi, G. Serra, and R. 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. 8

[18] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. Predicting human eye fixations via an lstm-based saliency attentive model. *IEEE Transactions on Image Processing*, 27(10):5142–5154, 2018. 1, 2, 4, 5, 7

[19] R. 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. 7

[20] S. A. Esmaeili, B. Singh, and L. 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. 2

[21] A. Geitgey. Face recognition. http://github.com/ageitgey/face_recognition. 3

[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 8

[23] S. He, H. R. Tavakoli, A. Borji, and N. Pugeault. A synchronized multi-modal attention-caption dataset and analysis. *arXiv preprint arXiv:1903.02499*, 2019. 8

[24] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 5

[25] X. Huang, C. Shen, X. Boix, and Q. 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. 1, 2

[26] S. Jia. Eml-net: An expandable multi-layer network for saliency prediction. *arXiv preprint arXiv:1805.01047*, 2018. 4, 7

[27] M. Jiang, S. Huang, J. Duan, and Q. Zhao. Salicon: Saliency in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1072–1080, 2015. 1, 2, 3

[28] M. Jiang, J. Xu, and Q. Zhao. Saliency in crowd. In *European Conference on Computer Vision*, pages 17–32. Springer, 2014. 1, 3

[29] T. Judd, F. Durand, and A. Torralba. Fixations on low-resolution images. *Journal of Vision*, 11(4):14–14, 2011. 3

[30] A. Khosla, A. S. Raju, A. Torralba, and A. Oliva. Understanding and predicting image memorability at a large scale. In *International Conference on Computer Vision (ICCV)*, 2015. 3

[31] N. W. Kim, Z. Bylinskii, M. A. Borkin, K. Z. Gajos, A. Oliva, F. Durand, and H. Pfister. Bubbleview: an interface for crowdsourcing image importance maps and tracking

visual attention. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24(5):36, 2017. 2

[32] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184, 2016. 2

[33] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 4

[34] J. Pan, C. Canton, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. a. Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. In *arXiv*, January 2017. 1, 2, 4

[35] J. Pan, E. Sayrol, X. Giro-i Nieto, K. McGuinness, and N. 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. 7

[36] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays. Webgazer: Scalable webcam eye tracking using user interactions. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence-IJCAI 2016*, 2016. 2

[37] S. Park, W. Kim, and K. M. Lee. Abnormal object detection by canonical scene-based contextual model. In *European Conference on Computer Vision (ECCV)*, 2012. 1, 3

[38] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017. 8

[39] D. Rudoy, D. B. Goldman, E. Shechtman, and L. Zelnik-Manor. Crowdsourcing gaze data collection. *Proceedings of ACM Collective Intelligence Conference*, 2012. 2

[40] Z. Wei, J. Zhang, X. Shen, Z. Lin, R. Mech, M. Hoai, and D. 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. 2

[41] J. Xu, M. Jiang, S. Wang, M. S. Kankanhalli, and Q. Zhao. Predicting human gaze beyond pixels. *Journal of vision*, 14(1):28–28, 2014. 2

[42] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015. 8

[43] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. 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. 3