EyeSyn: Psychology-inspired Eye Movement Synthesis for Gaze-based Activity Recognition

Guohao Lan
TU Delft
g.lan@tudelft.nl

Tim Scargill
Duke University
timothyjames.scargill@duke.edu

Maria Gorlatova
Duke University
maria.gorlatova@duke.edu

ABSTRACT

Recent advances in eye tracking have given birth to a new genre of gaze-based context sensing applications, ranging from cognitive load estimation to emotion recognition. To achieve state-of-the-art recognition accuracy, a large-scale, labeled eye movement dataset is needed to train deep learning-based classifiers. However, due to the heterogeneity in human visual behavior, as well as the labor-intensive and privacy-compromising data collection process, datasets for gaze-based activity recognition are scarce and hard to collect. To alleviate the sparse gaze data problem, we present EyeSyn, a novel suite of *psychology-inspired generative models* that leverages only publicly available images and videos to synthesize a *realistic* and *arbitrarily large* eye movement dataset. Taking gaze-based museum activity recognition as a case study, our evaluation demonstrates that EyeSyn can not only replicate the distinct patterns in the actual gaze signals that are captured by an eye tracking device, but also simulate the signal diversity that results from different measurement setups and subject heterogeneity. Moreover, in the few-shot learning scenario, EyeSyn can be readily incorporated with either transfer learning or meta-learning to achieve 90% accuracy, without the need for a large-scale dataset for training.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing theory, concepts and paradigms**; • **Computing methodologies → Simulation types and techniques**.

KEYWORDS

Eye tracking, eye movement synthesis, activity recognition.

1 INTRODUCTION

Eye tracking is on the verge of becoming pervasive due to recent advances in mobile and embedded systems. A broad selection of commercial products, such as Microsoft HoloLens 2 [1], Magic Leap One [2], and VIVE Pro Eye [3], is already incorporating eye tracking to enable novel gaze-based interaction and human context sensing. Moreover, general-purpose RGB cameras, such as those embedded in smartphones [4], tablets [5], and webcams [6], can also be used to capture users' eye movements. The accessibility of eye tracking-enabled devices has given birth to a new genre of gaze-based sensing applications, including cognitive load estimation [7], sedentary activity recognition [8], reading comprehension analysis [9], and emotion recognition [10].

Recent gaze-based sensing systems leverage learning-based techniques, in particular deep neural networks (DNNs) [10–12], to achieve state-of-the-art recognition performance. However, the success of DNN-based methods depends on how well the training dataset covers the inference data in deployment scenarios. Ideally, one would like to collect a large-scale labeled eye movement dataset, e.g., hundreds of instances for each subject and visual stimulus [10], to derive robust DNN models that are generalized across different deployment conditions. However, this is impractical for three reasons. First, human visual behavior is highly heterogeneous across subjects, visual stimuli, hardware interfaces, and environments. For instance, eye movements involved in reading are diverse among subjects [13], layouts of the reading materials [9], and text presentation formats [14]. Thus, the countless possible combinations of the dependencies make the collection of a large-scale, labeled dataset impractical. Second, since eye movement patterns can reveal users' psychological and physiological contexts [15], a gaze dataset that is collected from dozens or hundreds of users over multiple activity sessions is vulnerable to potential privacy threats [16]. Lastly, the collection of eye movement data is a labor-intensive and time-consuming process, which typically involves the recruitment of human subjects to perform a set of pre-designed activities. It is even more challenging and problematic to perform large-scale data collection when human interactions are restricted, such as throughout the COVID-19 shelter-in-place orders.

These challenges make the collection of large-scale, labeled eye movement datasets impractical, which further limits the performance of existing gaze-based activity recognition systems. In fact, previous work has shown that the lack of sufficient training data can lead to a 60% accuracy deficiency [12]. While recent transfer learning [17] and meta-learning-based methods [18] can be adopted to mitigate the dependency of the DNN models on large-scale training datasets in the deployment stage, they still require a highly diverse base dataset to pre-train the models.

To move beyond the current limitations, we present EyeSyn, *a comprehensive set of psychology-inspired generative models* that can synthesize realistic eye movement data for four common categories of cognitive activity, including *text reading*, *verbal communication*, and *static and dynamic scene perception*. Specifically, EyeSyn leverages publicly available images and videos as the inputs, and considers them as the visual stimuli to generate the corresponding gaze signals that would be captured by an eye tracking device when the subject is performing a certain activity.

EyeSyn embraces three important features. First, distinct from the Generative Adversarial Network (GAN)-based data augmentation methods [19, 20], which require hundreds of data samples for training [21], EyeSyn is *training-free* and does not require any eye movement data for synthesis. Second, EyeSyn can readily use a wide range of image and video datasets to *generate an arbitrarily large and highly diverse eye movement dataset.* For instance, it can leverage a public painting image dataset [22], which contains 7,937 images of famous paintings, to synthesize the potential eye movements when subjects are viewing these paintings. It can also exploit a text image dataset [23], which consists of 600 images of scanned documents, to generate the corresponding eye movements when subjects are reading these texts. Third, in contrast to a conventional data collection process that is usually confined to specific setups, visual stimuli, or subjects, EyeSyn can *simulate different eye tracking setups*, including visual distance, rendering size of the visual stimuli, sampling frequency, and subject diversity. These features make EyeSyn an important first step towards the greater vision of automatic eye movement synthesis that can alleviate the sparse data problem in gaze-based activity recognition.

EyeSyn is made possible by a comprehensive suite of novel models devised in this work. First, we introduce the ReadGaze model (Section 4.2) to simulate visual attention in text reading. Specifically, we design a text recognition-based optimal viewing position detection module to identify the potential viewing points in a given text stimulus. We also develop a skipping effect simulator to model the visual behavior of skip reading [24]. Second, we develop the VerbalGaze model (Section 4.3) which consists of a facial region tracking module and a Markov chain-based attention model to simulate the visual behaviors of fixating on and switching attention between different facial regions [25] in verbal communication. Lastly, we design the StaticScene and DynamicScene models (Section 4.4) to synthesize eye movements in static and dynamic scene perception. Specifically, we propose a saliency-based fixation estimation model to identify potential fixation locations in the visual scene, and propose a centrality-focused saliency selection module to model the effects of the central fixation bias [26] on fixation selection. Our major contributions are summarized as follows:

- We propose EyeSyn, a novel set of psychology-inspired generative models that synthesize eye movement signals in reading, verbal communication, and scene perception. Taking the actual gaze signals captured by an eye tracker as the ground-truth, we demonstrate that EyeSyn can not only replicate the distinct trends and geometric patterns in the gaze signal for each of the four activities, but can also simulate the heterogeneity among different subjects.
- We demonstrate that EyeSyn can leverage a wide range of publicly available images and videos to generate *an arbitrarily large and diverse* eye movement dataset. As shown in Section 5.1, using a small set of image and video stimuli we have prepared, EyeSyn synthesizes over 180 hours of gaze signals, which is 18 to 45 times larger than the existing gaze-based activity datasets [8, 12].
- Using gaze-based museum activity recognition as a case study, we demonstrate that a convolutional neural network (CNN)-based classifier, trained by the synthetic gaze signals generated by EyeSyn, can achieve 90% accuracy which is as high as state-of-the-art solutions, without the need for labor-intensive and privacy-compromising data collection.

The rest of the paper is organized as follows. We review related work in Section 2. We introduce the overall design, underlying cognitive mechanisms, and the case study in Section 3. We present the design details of the psychology-inspired generative models in Section 4. Section 5 introduces the system design and dataset. We evaluate our work in Section 6, and discuss the current limitations and future directions in Section 7. We conclude the paper in Section 8.

The research artifacts, including the implementation of the generative models and our own collected gaze dataset are publicly available at https://github.com/EyeSyn/EyeSynResource.

2 RELATED WORK

Gaze-based context sensing. Our work is related to recent efforts in gaze-based context sensing, including sedentary activity recognition [8, 12], reading behavior analysis [11], and emotion recognition [10, 27]. All these works require a large-scale gaze [8, 11, 12] or eye image dataset [10, 27] to train DNN-based classifiers for context recognition. Although recent transfer learning [17] and meta-learning-based methods [12, 18] can be adopted to mitigate the dependency of the DNN models on a large-scale training dataset in the deployment stage, they still require a highly diverse base dataset to pre-train the DNN models.

Gaze simulation. The problem of synthesizing realistic gaze signals has been studied in computer graphics and eye tracking literature [28]. For instance, Eyecatch [29] introduces a generative model that simulates the gaze of animated human characters performing fast visually guided tasks, e.g., tracking a thrown ball. Similarly, building on the statistics obtained from eye tracking data, EyeAlive [30] simulates the gaze of avatars in face-to-face conversational interactions. More recently, Duchowski et al. [31, 32] introduce a physiologically plausible model to synthesize realistic saccades and fixation perturbations on a grid of nine calibration points. Different from the existing efforts that rely solely on statistical models for gaze simulation, EyeSyn can leverage a wide range of images and videos to synthesize realistic gaze signals that would be captured by eye tracking devices.

Fixations estimation. Our work is also related to existing works on visual attention estimation [33], which predict a subject's fixation locations on images [34–37] and videos [38, 39]. Early works in this field either leverage low-level image features extracted from the image [34, 35], or combine image features with task-related contexts [36–38] to estimate a subject's visual attention. Recently, data-driven approaches have achieved more advanced performance in fixation estimation by taking advantage of deep learning models that are trained on large amounts of gaze data [40–42]. In this work, we build the scene perception model of EyeSyn (Section 4.4) on the image feature-based saliency detection model proposed by Itti et al. [34] to ensure training-free attention estimation, and advance it with a centrality-focused fixation selection algorithm to generate more realistic gaze signals. In addition, as shown in Sections 4.2 and 4.3, inspired by the research findings in cognitive science [24, 25], EyeSyn also introduces two novel models to estimate fixations in text reading and verbal communication.

3 OVERVIEW

3.1 Overall Design

An overview of EyeSyn is shown in Figure 1. It takes publicly available images and videos as the inputs to synthesize realistic eye movements for four common categories of cognitive activity, including: *text reading*, *verbal communication*, and *static and dynamic scene perception*. As shown, EyeSyn incorporates three psychology-inspired generative models to synthesize the corresponding visual behaviors that would be captured by an eye tracker when a sub-

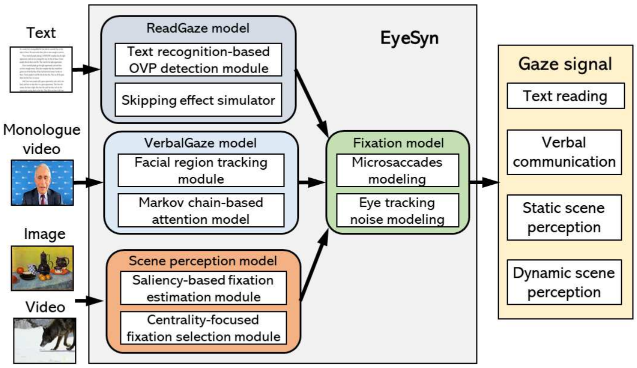

Text
Monologue video
Image
Video
EyeSyn
ReadGaze model
Text recognition-based OVP detection module
Skipping effect simulator
VerbalGaze model
Facial region tracking module
Markov chain-based attention model
Scene perception model
Saliency-based fixation estimation module
Centrality-focused fixation selection module
Fixation model
Microsaccades modeling
Eye tracking noise modeling
Gaze signal
Text reading
Verbal communication
Static scene perception
Dynamic scene perception

Figure 1: Overview of EyeSyn.

ject is performing the activity. Moreover, to generate realistic gaze signals, the fixation model is introduced to simulate gaze perturbations that result from both microsaccades and the measurement noise in eye tracking. EyeSyn opens up opportunities to generate realistic, large-scale eye movement datasets that can facilitate the training of gaze-based activity recognition applications [8, 9, 12], and eliminate the need for expensive and privacy-compromising data collection. Below, we introduce the underlying cognitive mechanism in eye movement control that motivates our design. For each of the four activities, we describe how the human visual system makes decisions about the fixation location and fixation duration by answering the questions of: *where and when will the eyes move?* and *why do the eyes move in such a way?*

3.2 Cognitive Mechanism and Motivation

3.2.1 Text reading. During reading, the human visual system makes decisions about the fixation location and fixation duration in two independent processes [24]. The fixation locations are largely determined by the low-level visual information, such as the length of the word and its distance to the prior fixation location [24]. It is generally argued that readers attempt to land their fixations on the center of the word, which is known as the *optimal viewing position* (OVP) [43]. The OVP is the location in a word at which the visual system needs the minimum amount of time to recognize the word. The fixation durations are determined by the characteristics of the word, in particular, the word length [24]. Moreover, words are sometimes skipped in reading, which is known as the *skipping effect*. In general, the probability of skipping a word decreases with the word length [24, 44, 45].

Following this cognitive mechanism, we propose the **ReadGaze model** (Section 4.2) to simulate visual attention in text reading. As shown in Figure 1, ReadGaze consists of the text recognition-based OVP detection module to identify the potential fixation points in a given text stimulus, as well as the the skipping effect simulator to simulate the visual behavior of skip reading.

3.2.2 Verbal communication. Research in cognitive neuroscience has shown that participants in verbal communication direct most of their visual attention at their communication partner. Specifically, they tend to fixate on and scan different regions of the partner's face [25], even if the face occupies only a small portion of the visual field. Among different facial regions, the *eyes*, *nose*, and *mouth* are the three most salient fixation regions, as they provide many useful cues for both speech and cognitive perception [46]. The underlying motivation of this cognitive behavior is that listeners care about where the speaker is focusing, and thus eye gaze is used as the cue to track and follow the attention of the speaker [47]. Similarly, the movements of the mouth provide additional linguistic information and audiovisual speech cues for the listener [46]. Lastly, facial expressions in the nose region help in the recognition of emotions of the speaker [48].

We propose the **VerbalGaze** model (Section 4.3) to simulate eye movement in verbal communication. As shown in Figure 1, it leverages monologue videos that are widely available online as the inputs to simulate the interactions in verbal communication. Specifically, it models the eye movements of the people who are listening to the speaker in the video. In fact, monologue videos are widely used in cognitive science to study attention and eye movement patterns in social interactions [25, 46], and have been proven to have the same underlying cognitive mechanism as in-person verbal communication [49]. In our design, we propose the facial region tracking module and the Markov chain-based attention model to simulate the visual behaviors of fixating on and switching attention between different facial regions [25] in verbal communication.

3.2.3 Static and dynamic scene perception. When inspecting complex visual scenes, the human visual system does not process every part of the scene. Instead, it selects portions of the scene and directs attention to each one of them in a serial fashion [50]. Such selective visual attention can be explained by the *feature integration theory* [51], which suggests that the visual system integrates low-level features of the scene, such as color, orientation, and spatial frequency, into a topographic *saliency map* in the early stages of the process. Then, visual attention is directed serially to each of the salient regions that locally stands out from their surroundings [50]. The selection of fixation locations is also affected by the *central fixation bias* [26] which refers to the strong tendency in visual perception for subjects to look at the center of the viewing scene. Studies have shown that the center of the scene is an optimal location for extracting global visual information, and is a convenient starting point for the oculomotor system to explore the scene [52].

In this work, we design two generative models, **StaticScene** and **DyamicScene** (Section 4.4), to simulate eye movements in *static and dynamic scene perception* (subjects are viewing paintings or watching videos), respectively. As shown in Figure 1, we propose the saliency-based fixation estimation module to identify the potential fixation locations in the image, and propose a centrality-focused fixation selection module to model the effects of the central fixation bias [26] on fixation selection.

3.2.4 Fixation model. Lastly, EyeSyn also incorporates a set of statistical models that simulate the gaze perturbations in fixations (Section 4.1). Specifically, we model both the microsaccades, the subconscious microscopic eye movements produced by the human oculomotor system during the fixations, and the measurement noise in eye tracking to generate realistic fixation patterns.

3.3 Case Study

In this paper, we consider **gaze-based museum activity recognition for mobile augmented reality (AR)** as a case study. We show how the synthesized eye movement data from EyeSyn can improve the recognition accuracy of a DNN-based classifier without the need for a large-scale gaze dataset for training.

Different from traditional museum exhibitions, mobile AR allows augmenting physical exhibits with more vivid and informative content, which enhances visitors' engagement and experience. There are many practical deployments of AR-based museum exhibitions. For instance, the Skin and Bones [53] application, deployed at the Smithsonian National Museum of Natural History, provides visitors with a new way to see what extinct species looked like and how they moved.

To ensure accurate and timely virtual content delivery, it is essential to have a context-aware system that can continuously track and recognize the physical object the user is interacting with. Although one can leverage the camera on the AR device to recognize the object in the user's view directly [54], one practical aspect that has been largely overlooked is that *having the object in view does not always mean the user is interacting with it*. This is especially true in scenarios where head-mounted AR devices are used, for which one cannot simply rely on the location and orientation of the device as the indicators of potential user-object interaction. In fact, state-of-the-art head-mounted AR solutions have incorporated eye trackers to estimate the visual attention of the user [1, 2].

In this case study, we leverage the gaze signals captured by head-mounted AR devices to recognize four interactive activities that are performed by a visitor to a virtual museum:

- **Read**: reading text descriptions of an exhibit.
- **Communicate**: talking with someone in the museum or watching monologue videos of an artist.
- **Browse**: browsing paintings that are exhibited in the museum.
- **Watch**: watching a descriptive video about an exhibit.

To showcase how gaze-based activity recognition can be used to benefit an AR user's experience in this application, we develop a demo on the Magic Leap One AR headset [2]. A short video of the demo can be found at https://github.com/EyeSyn/EyeSynResource. Specifically, leveraging the gaze signals that are captured by the Magic Leap One, the context-aware system can recognize the interactive activity the user is performing. Then, based on the context, the system adjusts the digital content displayed in the user's view to enhance her engagement and learning experience.

4 PSYCHOLOGY-INSPIRED GENERATIVE MODELS

Below, we present the detailed design of EyeSyn. We first introduce the fixation model, followed by three psychology-inspired models that synthesize eye movements in text reading, verbal communication, and scene perception. While these models are designed based on findings in psychology and cognitive science, to the best of our knowledge, we are the first to develop generative models to synthesize realistic eye movement signals for activity recognition.

4.1 Fixations Modeling

Gaze and fixation are the two most common eye movement behaviors. Gaze point refers to the instantaneous spatial location on the stimulus where the subject's visual attention lands, while fixation point refers to the spatial location where the subject tries to maintain her gaze. When the eyes are fixating on the fixation point, the gaze points captured by the eye tracker contain perturbations. To illustrate, we use the Pupil Labs eye tracker [55] to record the gaze points while a subject is fixating on a red calibration point displayed on a computer monitor. As shown in Figure 2, the recorded gaze points contain many perturbations and fluctuate around the calibration point. The two major sources of the perturbations are *the microsaccades* and *the noise in eye tracking*. Below, we introduce models that simulate the perturbations and generate realistic gaze signals in fixations.

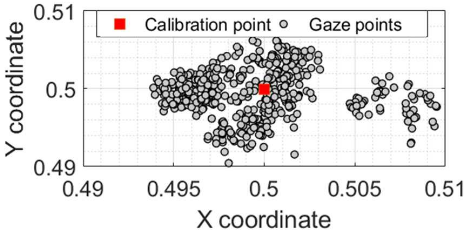


Figure 2: Example of gaze perturbations in a fixation.

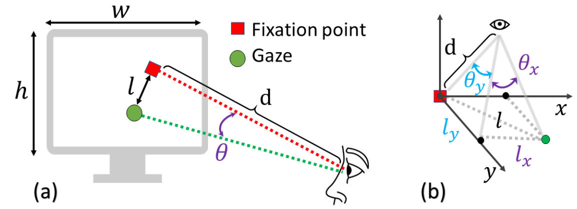
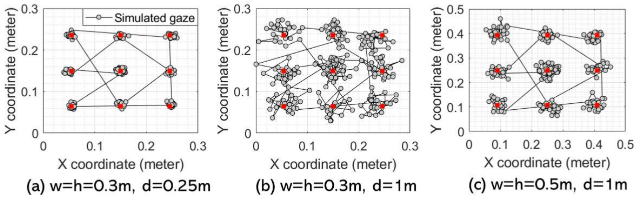


Figure 3: (a) Example of the gaze perturbation in terms of gaze angle (θ) and the gaze offset (l). (b) Decomposition of the overall gaze perturbation.

Metrics for modeling. To quantify the gaze perturbations, we introduce *gaze angle* and *gaze offset* as the metrics. As shown in Figure 3(a), the red dashed line is the direction from the eyes to the fixation point, while the green dashed line is the link between the eyes and the gaze measured by the eye tracker. The gaze angle θ measures the deviation in degrees between the two lines, while the gaze offset l captures the Euclidean distance between the measured gaze and the fixation point on the visual scene. The height and width (in the unit of meters) of the visual scene are denoted by h and w, respectively. The line-of-sight distance between the eyes and the fixation point is denoted by d. Below, we model the gaze perturbations in terms of the two metrics.

Modeling microsaccades. During fixations, the eyes make microscopic movements known as *microsaccades*, which are subconscious movements that are produced by the human oculomotor system to maximize visual acuity and visual perception during the fixation. Recent studies in neurophysiology have shown that the erratic fluctuations in fixations can be modeled by $1/f^{\alpha}$ noise [56], where f is the cyclic frequency of the signal and α is the inverse frequency power that ranges from 0 to 2. In this work, we simulate the microsaccades-induced gaze perturbation by applying a $1/f^{\alpha}$ filter on a stream of Gaussian white noise [57]. Specifically, we model the perturbations in gaze angle, θ_{micro}, by $\theta_{\text{micro}} = \mathrm{F}(s, \alpha)$, where s is the input white noise that follows the Gaussian distribution of $\mathcal{N}(0, 1/300)$ (in degrees) [31], and $\mathrm{F}(s, \alpha)$ is a $1/f^{\alpha}$ filter with the inverse frequency power of α. We set α to 0.7 for generating more realistic microsaccade patterns [31, 32].



(a) w=h=0.3m, d=0.25m (b) w=h=0.3m, d=1m (c) w=h=0.5m, d=1m

Figure 4: Example of simulated gaze points given different visual scene sizes (w and h) and distances d.

Modeling noise in eye tracking. The noise in eye tracking also contributes to the gaze perturbations. In practice, many factors can influence eye tracking quality [58], including: the environment (e.g., different lighting conditions), the eye physiology of the subjects, and the design of the eye tracker (e.g., resolution of the camera and the eye tracking algorithm). Following the literature [32], we model the gaze perturbations (in degrees) that result from eye tracking noise, θ_{track}, by a Gaussian distribution $\theta_{\text{track}} \sim \mathcal{N}(0, 1.07)$.

Gaze simulator. To sum up, as shown in Figure 3(b), taking the fixation point as the origin of the coordinate system, we can further decompose the overall perturbation in the X and Y directions, in which we use notations l_x, l_y, θ_x, and θ_y to denote the decomposed gaze offsets and gaze angles for the two directions, respectively. Then, we can obtain l_x and l_y by $l_x = 2d \cdot \sin(\theta_x/2)$ and $l_y = 2d \cdot \sin(\theta_y/2)$, respectively, where d is the line-of-sight distance between the eyes and the calibration point; θ_x and θ_y are the decomposed gaze angles in X and Y, respectively, which are modeled by $\theta_{\text{micro}} + \theta_{\text{track}}$.

After modeling the gaze offset, we use a sequence of m fixation points, $\mathrm{P} = \{\mathrm{p}_1, \ldots, \mathrm{p}_m\}$, as the input to simulate gaze points in fixations. Each point $\mathrm{p}_i = (x_i, y_i)$ represents a *potential fixation point* on a normalized 2D plane, where x_i and y_i are the X and Y coordinates, respectively. Moreover, given a $w \times h$ visual scene, we can transfer p_i from the normalized plane to the coordinate of the visual scene by: $\mathrm{p}_i' = (x_i \times w, y_i \times h)$, $\forall \mathrm{p}_i = (x_i, y_i) \in \mathrm{P}$. This transformation allows us to take the size of the visual scene into account when simulating the gaze points.

Then, we use $\mathrm{G_i} = \{g_{i,1}, \ldots, g_{i,n}\}$ to denote a sequence of n gaze points that will be captured by the eye tracker when the subject is fixating on p_i'. The length of the sequence, n, is equal to $t_i \times f_s$, in which t_i is the fixation duration on p_i', and f_s is the sampling frequency of the eye tracking device. The kth gaze point, $\mathrm{G_i}(k) = g_{i,k}$, is obtained by adding a gaze offset to p_i':

$$\mathrm{G_i}(k) = \mathrm{p}_i' + \mathrm{L_i}(k), \tag{1}$$

where $\mathrm{L_i}$ is a sequence of gaze offsets generated based on Equations (1)-(3), and $\mathrm{L_i}(k) = \left(l_x(i), l_y(i)\right)$ is the kth gaze offset in the sequence. As an example, Figure 4 shows the simulated gaze points when taking a grid of nine fixation points as the inputs. Different visual scene sizes $w \times h$ and distances d are used in the simulation. We observe that a longer visual distance d or a smaller visual scene leads to higher perturbations in the simulated gaze signal, which matches the observations with practical eye trackers [59].

4.2 Eye Movement in Reading

Below, we introduce the details of the **ReadingGaze** model, which incorporates both OVP theory and the skipping effect to simulate the eye movements in text reading.

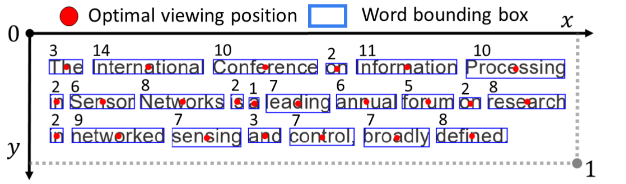


Figure 5: Example of detecting the optimal viewing positions on the input text image.

Table 1: Probability of fixation and mean fixation duration (in ms) on the target word as a function of the word length (in number of letters) [44, 45].

Word length	Fixation probability	Fixation duration
1	0.077	209
2	0.205	215
3	0.318	210
4	0.480	205
5	0.800	229
6	0.825	244
7	0.875	258
8	0.915	260
⩾9	0.940	276

4.2.1 *Text recognition-based OVP detection.*

We introduce a text recognition-based OVP detection module to identify the potential fixation points in a given text stimulus. Specifically, we leverage the Google Tesseract optical character recognition engine [60] to detect the locations and lengths of the words in an input text image. We use Tesseract because of its high efficiency and its support of more than 100 languages [61]. As shown in Figure 5, the words in the input text image are detected and highlighted by blue bounding boxes. The centers of the detected words are regarded as the OVPs. The associated word lengths are shown above the bounding boxes. Note that we are not interested in recognizing the exact text. Rather, we leverage the coordinates of the detected bounding box to calculate the OVP. Moreover, we obtain the length of the word (in number of letters) and use it to simulate the skipping effect.

4.2.2 *Skipping effect and fixation simulation.*

We leverage the eye movement statistics reported in Rayner et al. [44, 45] as the inputs to simulate the skipping effect and the fixation decision in text reading. Specifically, Table 1 shows the probability of fixation and the mean fixation duration on the target word as a function of the word length (in number of letters). Note that the fixation durations in Table 1 do not consider refixation (i.e., the behavior of fixating on a given word more than once), because given the OVP as the landing position for fixation, the probability of refixating is lower than 6%, regardless of the word length [24].

4.2.3 *The ReadingGaze model.*

Putting everything together, our model takes the text image as the input and detects a sequence of OVPs with the associated word lengths (as shown in Figure 5). Then, leveraging the statistics given in Table 1, it simulates the skipping effect on each of the detected OVPs based on its word length, and assigns fixation durations to the selected OVPs (i.e., OVPs that will be fixated on). The outputs of the ReadingGaze model are a sequence of m fixation points $\mathrm{P} = \{\mathrm{p}_1, \ldots, \mathrm{p}_m\}$ and the associated fixation durations $\mathrm{T} = \{\mathrm{t}_1, \ldots, \mathrm{t}_m\}$, where each point $\mathrm{p}_i = (x_i, y_i)$ is an OVP at which the subject will fixate on while reading, and t_i

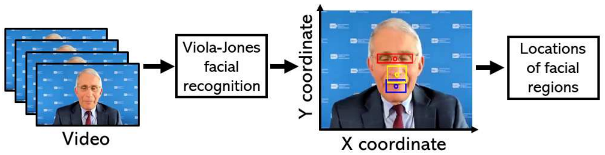


Figure 6: The pipeline of facial regions tracking.

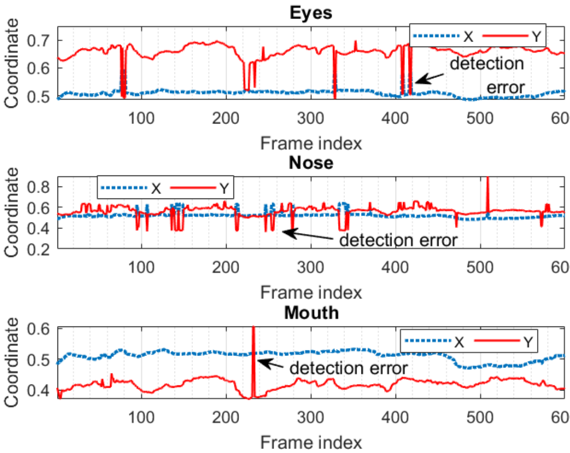


Figure 7: The tracked coordinates of the three facial regions in a 20-second video. The outliers in the time series are due to the detection errors of the Viola-Jones algorithm.

is the associated fixation duration. Lastly, we take P and T as the input of the gaze simulator in Equation 1.

4.3 Eye Movement in Verbal Communication

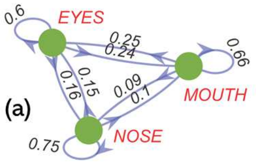
Below, we introduce the detailed design of VerbalGaze, which consists of a facial regions tracking module and a Markov chain-based attention model.

4.3.1 Facial region tracking. Taking the monologue video as input, we leverage the resource-efficient Viola-Jones algorithm [62] to detect the eyes, nose, and mouth of the speaker in the video frames. The centers of the detected facial regions are considered as the potential fixation locations. The processing pipeline of the facial regions tracking is shown in Figure 6. The detected eyes, nose, and mouth are bounded by red, yellow, and blue boxes, respectively, with their centers marked by circles. We denote the time series of the tracked coordinates of eyes, nose, and mouth, by C_{eyes}, C_{nose}, and C_{mouth}, respectively.

Figure 7 is an example of tracking the facial regions for a 20-second video with a 30fps frame rate (thus, 600 frames). The time series of the tracked positions are normalized. We can see outliers in the tracked positions, which result from the detection errors of the Viola-Jones algorithm, and appear mostly when the eyes or the mouth of the speaker are closed. We apply a scaled median absolute deviation-based outlier detector on a sliding window of 60 points to detect and remove these errors.

4.3.2 Markov chain-based attention model. We design a three-state Markov chain to simulate the visual behaviors of *fixating on* and *switching attention between different facial regions* in verbal communication. As shown in Figure 8(a), we model the behaviors of fixating on the eyes, nose, and mouth regions as three states of a discrete-time Markov chain with state space $\mathcal{X} = \{EYES, NOSE, MOUTH\}$. We model the attention shift from one facial region to another by a Markovian transition. For instance, the attention shift from eyes to mouth is modeled by the transition from *EYES* to *MOUTH*. Lastly, each transition is assigned a transition probability. In this work, the transition probabilities are calculated based on the eye movement statistics reported by Jiang et al. [25]. Note that we can easily adjust the transition probabilities to fit the eye movement behaviors in different scenarios. For instance, we can increase the probability of fixating on eyes to simulate verbal communication in a face-to-face scenario, in which listeners tend to look more at the speaker's eyes due to more frequent eye contact [49]. Then, to simulate the attention shifts among the three facial regions, we perform a random walk on the Markov chain to generate a sequence of states $x_{1:n} \triangleq (x_1, \ldots, x_n)$, where $x_t : \Omega \to \mathcal{X}$ and $x_t \in x_{1:n}$ represents the state at step t. An example of the simulated state sequence is shown in Figure 9(a), where the initial state x_1 is *EYES*.

(a)



(b)

States	μ	σ
EYES	2.27	1.07
NOSE	0.65	0.22
MOUTH	2.07	1.52

Figure 8: (a) Diagram of the three-state Markov chain; the three states *EYES, NOSE,* and *MOUTH,* represent the eye movement behavior of fixating on the eyes, nose, and mouth regions, respectively; the transitions model the *attention shift* between facial regions. (b) the Gaussian distributions of the ISI on the three states.

4.3.3 Adding the 'sense of time'. We use the *inter-state interval* (ISI) to represent the duration of time (in seconds) that the attention will stay in each state $x_t \in x_{1:n}$. Moreover, since the three facial regions function differently in the cognitive process of verbal communication, they lead to different fixation durations [25, 46]. Thus, as shown in Figure 8(b), we use three Gaussian distributions to model the ISI of the three states. The mean, μ, and standard deviation, σ, of the distributions are adopted from the statistics reported in [25].

For a video with m frames, we generate the attention sequence, $a_{1:m} \triangleq (a_1, \ldots, a_m)$, to simulate the subject's attention on each of the video frames. Formally, attention shift is simulated to occur at frame index $\tau \in \tau_{1:n} \triangleq (\tau_1, \ldots, \tau_n)$ of the video, where $\tau_1 = f_v \times ISI_1$, and $\tau_i = \tau_{i-1} + f_v \times ISI_i$, $\forall x_i \in x_{1:n}$. Notation ISI_i denotes the inter-state interval for attention state x_i, and is sampled from the corresponding Gaussian distribution defined in Figure 8(b); f_v is the frame rate (in fps) of the video. Then, $a_{1:m}$ is generated by assigning each of the image frames with the corresponding attention state value $|x_i|$:

$$a_{(\tau_{i-1}+1):\tau_i} = |x_i|, \ \forall \ \tau_i \in \tau_{1:n}, \tag{2}$$

where $|x_i| \in \{EYES, NOSE, MOUTH\}$. As an example, Figure 9(b) shows the attention sequence $a_{1:m}$ for a 100-second video.

4.3.4 The VerbalGaze model. We combine the simulated attention sequence, $a_{1:m}$, with the location time series, i.e., C_{eyes}, C_{nose}, and C_{mouth}, obtained from the facial region tracking module, to generate a sequence of m fixation points $\mathrm{P} = \{\mathrm{p}_1, \ldots, \mathrm{p}_m\}$. Each fixation point $\mathrm{p}_i = (x_i, y_i)$ represents the location of the corresponding

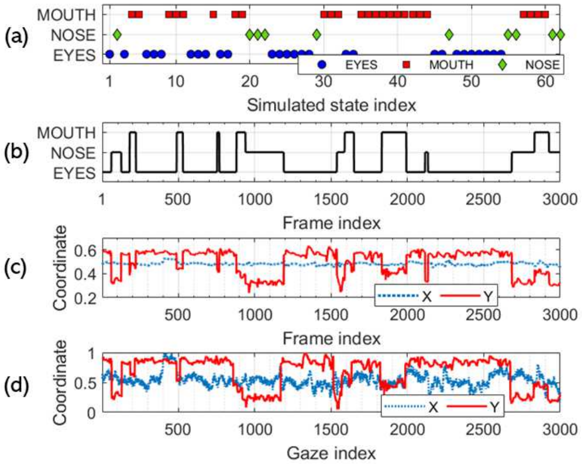


Figure 9: (a) Simulated discrete state sequence $x_{1:n}$ with $n = 62$ and $x_1 = EYES$; (b) the corresponding attention sequence $a_{1:m}$ on 3000 video frames (with $f_v = 30$); (c) simulated fixation sequence P; (d) simulated gaze time series.

facial region the subject will fixate on:

$$\mathrm{p}_i = \begin{cases} \mathrm{C}_{eyes}(i) & \text{if } a_i = EYES, \\ \mathrm{C}_{nose}(i) & \text{if } a_i = NOSE, \\ \mathrm{C}_{mouth}(i) & \text{if } a_i = MOUTH, \end{cases} \quad \forall \mathrm{p}_i \in \mathrm{P}. \tag{3}$$

As $a_{1:m}$ simulates the visual attention for all the video frames, the associated set of fixation durations $\mathrm{T} = \{\mathrm{t}_1, \ldots, \mathrm{t}_m\}$ is obtained by: $\mathrm{t_i} = 1/f_s$, $\forall \mathrm{t_i} \in \mathrm{T}$. An example of P is shown in Figure 9(c), which is generated by taking the tracked facial region locations (shown in Figure 7) and the attention sequence (shown in Figure 9(b)) as the inputs. Finally, P and T are fed into the gaze simulator (in Equation 1) to synthesize the gaze signal shown in Figure 9(d).

4.4 Eye Movement in Scene Perception

Below, we introduce two generative models, **StaticScene** and **DynamicScene**, to synthesize eye movements in static and dynamic scene perception, respectively. Specifically, we design the image feature-based saliency detection model to identify the potential fixation locations in the scene, and develop a centrality-focused saliency selection algorithm to simulate the effects of the central fixation bias on the selection of fixation location.

4.4.1 Saliency-based fixation estimation. We leverage the widely used bottom-up saliency model proposed by Itti et al. [34, 35] to identify the saliency of an input image. In brief, the saliency estimation model first extracts low-level vision features to construct the intensity, color, and orientation feature maps, respectively. Then, the three feature maps are normalized and combined into the final saliency map [35]. Taking the saliency map $\mathcal{S}$ as the input, we simulate the *serial and selective visual attention behavior* in scene perception. Specifically, for each of the salient regions in $\mathcal{S}$, we first identify the location of its local maxima, which indicates the point to which attention will most likely be directed. Then, we generate a set of m fixation points $\mathrm{P} = \{\mathrm{p}_1, \ldots, \mathrm{p}_m\}$, in which m is the number of salient regions in $\mathcal{S}$, and each fixation point $\mathrm{p}_i = (x_i, y_i) \in \mathrm{P}$ corresponds to the location of one local maxima. As shown in Figure 10(a), six salient regions and their local maxima are identified in $\mathcal{S}$, which correspond to six potential fixation locations. Finally, we simulate the serial attention behavior by connecting the identified fixation locations in order of their local maxima. As shown, a fixation sequence $\mathrm{P} = \{\mathrm{p}_1, \mathrm{p}_2, \mathrm{p}_3, \mathrm{p}_4, \mathrm{p}_5, \mathrm{p}_6\}$ is generated, in which p_1 and p_6 correspond to the fixation points that have the highest and the lowest local maxima in $\mathcal{S}$, respectively.

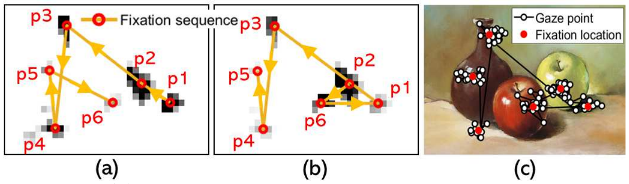


Figure 10: (a) Original saliency map $\mathcal{S}$ with the associated fixation sequence $\mathrm{P} = \{\mathrm{p}_1, \mathrm{p}_2, \mathrm{p}_3, \mathrm{p}_4, \mathrm{p}_5, \mathrm{p}_6\}$ overlaid on it. (b) The weighted saliency map $\bar{\mathcal{S}}$ with the new fixation sequence $\bar{\mathrm{P}} = \{\mathrm{p}_2, \mathrm{p}_6, \mathrm{p}_1, \mathrm{p}_3, \mathrm{p}_4, \mathrm{p}_5\}$ overlaid on it. (c) The simulated gaze points overlaid on the input image.

4.4.2 Centrality-focused fixation selection. To simulate the *central fixation bias effect*, we further weight each of the fixation points in P by its distance to the image center. Specifically, we use notation $\mathcal{S}(\mathrm{p}_i)$ to denote the saliency value of p_i in $\mathcal{S}$. The weighted saliency value $\bar{\mathcal{S}}(\mathrm{p}_i)$ is obtained by:

$$\bar{\mathcal{S}}(\mathrm{p}_i) = \mathcal{S}(\mathrm{p}_i) \cdot e^{-\|\mathrm{p_i} - \mathrm{A}\|}, \forall \mathrm{p}_i \in \mathrm{P}, \tag{4}$$

where A denotes the center of the saliency map, and $\|\mathrm{p_i} - \mathrm{A}\|$ is the Euclidean distance between $\mathrm{p_i}$ and A. This distance metric gives more weight to fixation points that are closer to the image center. Then, by sorting the weighted fixation points, we generate a new fixation sequence $\bar{\mathrm{P}}$. An example is shown in Figure 10, in which the original saliency map $\mathcal{S}$ is compared with the weighted saliency map $\bar{\mathcal{S}}$. The fixation point p_6, which is closer to the image center, has a higher saliency value after the weighting, and is selected as the second attention location in the weighted fixation sequence $\bar{\mathrm{P}}$. Below, we introduce two generative models we developed to synthesize gazes in *static and dynamic scene perception.*

4.4.3 Static scene perception. A static scene refers to the scenarios in which the salient regions of the scene do not change over time (e.g., paintings). In this case, the input for the eye movements simulation is simply the image of the static visual scene. We introduce the **StaticScene** model which leverages the aforementioned image saliency-based and centrality-focused fixation estimation algorithm to generate a sequence of fixation points, $\bar{\mathrm{P}} = \{\mathrm{p}_1, \ldots, \mathrm{p}_n\}$, to simulate visual attention when a subject is viewing the static scene. We further model the fixation durations, $\mathrm{T} = \{\mathrm{t}_1, \ldots, \mathrm{t}_n\}$, in static scene perception by a Gamma distribution $\mathrm{T} \sim \Gamma(\alpha = 2.55, \beta = 71.25)$. The values of the shape parameter α and the rate parameter β are estimated based on 16,300 fixation duration instances extracted from the DesktopActivity [12] and the SedentaryActivity [8] eye tracking datasets. Specifically, we leverage the dispersion-based fixation detection algorithm [63] to detect fixations from the raw gaze signal, and fit a Gamma distribution on the calculated fixation durations. Finally, for gaze signal simulation, we use P and T as the inputs of the gaze simulator in Equation 1. As an example, Figure 10(c) shows the gaze points synthesized by the StaticGaze model when a subject is viewing the painting.

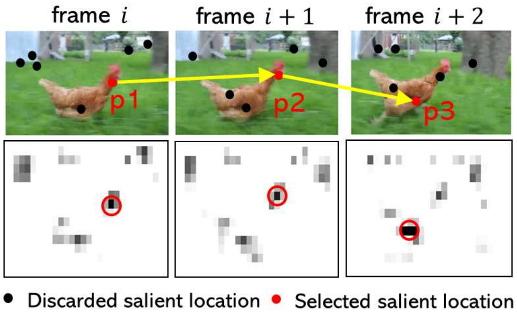


Figure 11: Example of generating the fixation sequence in a dynamic scene: figures in the first row are three continuous video frames; figures in the second row are the corresponding weighted saliency maps.

4.4.4 Dynamic scene perception. In dynamic scene perception (e.g., a subject watching videos or performing visual search in free space), the salient objects of the visual scene change over time. We introduce the **DynamicScene** model which takes a stream of video frames as the input for gaze simulation. According to the literature, the mean fixation duration in scene perception and visual search is around 180-330ms [64]. Thus, when the frame rate of the input video is higher than 5.4fps, i.e., with a frame duration shorter than 180ms, there will be only one fixation point in each video frame. In the current design, we assume the frame rate of the input video is higher than 5.4fps, and thus, instead of considering the local maxima of all the salient regions as fixation points, for each of the video frames we only select the location with the highest saliency value as the fixation point. As shown in Figure 11, a fixation sequence $\mathrm{P} = \{\mathrm{p}_1, \mathrm{p}_2, \mathrm{p}_3\}$ is generated by selecting the salient region with the highest saliency in each of the three continuous frames. The fixation durations $\mathrm{T} = \{\mathrm{t}_1, \ldots, \mathrm{t}_n\}$ in dynamic scene perception are determined by the frame rate f_v of the video: $\mathrm{t}_i = 1/f_v$, $\forall \mathrm{t_i} \in \mathrm{T}$. P and T are used as the inputs of Equation 1 to synthesize eye movement signals.

5 SYSTEM DESIGN AND DATASET

5.1 Synthetic Eye Movement Dataset

We implement EyeSyn in MATLAB, and use it to construct a massive synthetic eye movement dataset, denoted as **SynGaze**. The details of SynGaze are summarized in Table 2. Specifically, we use the following image and video data as the inputs to simulate gaze signals for the four activities:

- **Read**: we extract 100 text images from each of the three digital books, "Rich Dad Poor Dad", "Discrete Calculus", and "Adler's Physiology of the Eye". The three books differ in both text layout and font size. The extracted text images are used as the inputs to the *ReadingGaze* model.
- **Communicate**: we extract 100 monologue video clips from the online interview series of the "ACM Turing Award Laureate Interview" as the inputs to the *VerbalGaze* model. Each video clip lasts 5 to 7 minutes with a frame rate of 30fps.
- **Browse**: we leverage a public dataset with 7,937 images of famous paintings [22] as the input to the *StaticScene* model.
- **Watch**: we extract 50 short videos from the "National Geographic Animals 101" online documentary video series as the input to the *DynamicScene* model. Each video lasts 2 to 6 minutes.

Table 2: Summary of the synthetic eye movement dataset.

Activity	Simulation inputs	Simulated data length
Read	300 text images from three books	9.9 hours
Communicate	100 video clips of monologue interview	30.9 hours
Browse	7,937 images of paintings	132.3 hours
Watch	50 video clips of documentary videos	11.7 hours

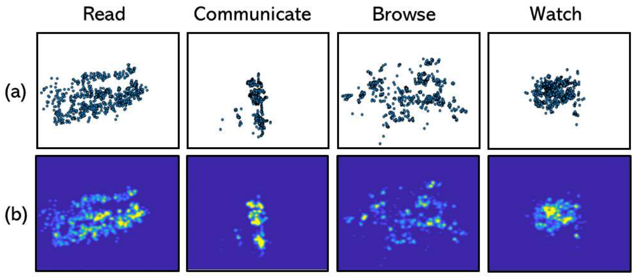


Figure 12: (a) The scatter plots of the aggregated gaze signals; and (b) the gaze heatmap generated from the gaze signal.

When modeling the microsaccades and the eye tracking noise in fixations (Section 4.1), we consider different settings of the scale parameters to simulate various rendering sizes of the visual stimuli ($w = h = 0.5$ and $w = h = 1m$), and viewing distances ($d = 0.5m$, $d = 1m$, and $d = 2m$). The sampling frequency for the simulation is set to 30Hz.

Extension feasibility. Note that SynGaze can easily be extended by using a variety of simulation settings, and by taking different datasets as the inputs. For instance, EyeSyn can be readily applied to the visual saliency dataset [39] which contains 431 video clips of six different genres, the iMet Collection dataset [65] which contains over 200K images of artwork, and the text image dataset [23] which consists of 600 images of scanned documents, to synthesize realistic gaze signals on new sets of visual stimuli.

5.2 Gaze-based Activity Recognition

Gaze heatmap. We propose the gaze heatmap as the data representation for gaze-based activity recognition. A gaze heatmap is a spatial representation of an aggregation of gaze points over a certain window of time. It provides an overview of the eye movements and indicates the regions in the visual scene at which subject's attention is located. As an example, Figure 12 shows the gaze heatmaps that are generated from the aggregated gaze points captured by the eye tracker. The color of the heatmap indicates the density of the subject's visual attention on the normalized 2D scene. To generate a gaze heatmap, we take the gaze points aggregated in each sensing window as the inputs, and create the 2D histogram of the gaze points based on their normalized coordinates. Then, we perform a 2D convolution operation with a Gaussian kernel on the histogram to generate the gaze heatmap. In our implementation, the resolution of the histogram is 128, and the width of the Gaussian kernel is 1. The final gaze heatmap has a size of 128×128.

CNN-based classifier. We design a convolutional neural network (CNN)-based classifier for gaze-based activity recognition. Table 3 shows the network architecture of the classifier. We choose this shallow design over deeper models (e.g., ResNet and VGGNet) to prevent overfitting when a small-scale dataset is used for model training [18]. The input to the classifier is a 128×128 gaze heatmap.

Table 3: The network design of the CNN-based classifier.

Layer	Size In	Size Out	Filter
conv1	128 × 128 × 1	128 × 128 × 32	3 × 3, 1
pool1	128 × 128 × 32	64 × 64 × 32	2 × 2, 2
conv2	64 × 64 × 32	64 × 64 × 32	3 × 3, 1
pool2	64 × 64 × 32	32 × 32 × 32	2 × 2, 2
conv3	32 × 32 × 32	32 × 32 × 32	3 × 3, 1
pool3	32 × 32 × 32	16 × 16 × 32	2 × 2, 2
flatten	16 × 16 × 32	8192	
fc	8192	128	
fc	182	4	

Note that while conventional hand-crafted feature-based classifiers [8, 9] may also benefit from the synthesized data generated by EyeSyn, we choose the CNN-based design due to its superior ability in extracting spatial features from the gaze signal [12].

6 EVALUATION

In this section, we first perform a signal level evaluation to assess the similarity between the actual and the synthesized gaze signals. Then, we investigate how the synthesized signals can be used to improve the performance of gaze-based activity recognition.

6.1 Data Collection

We collect a gaze dataset, denoted as **VisualProcessingActivity**, for the evaluation. The study is approved by our institution's Institutional Review Board. Two different eye tracking devices, the Pupil Labs [55] and the Magic Leap One [2], are used in the data collection, which allows us to evaluate our work with real gaze signals captured by heterogeneous devices. Eight subjects participate in the study: four subjects leverage the onboard eye tracker in the Magic Leap One, while the others use the Pupil Labs for eye movement collection. Both devices capture eye movements with a sampling frequency of 30Hz. The subjects can move freely during the experiment. Specifically, the subjects who are wearing the Pupil Labs are sitting in front of a 34-inch computer monitor at a distance of 50cm. The visual stimulus for each of the activities is displayed on the monitor. The resolution of the display is 800×600. We conduct the manufacturer's default on-screen five-points calibration for each of the subjects. For the Magic Leap One, the stimuli are rendered as virtual objects placed on blank white walls around a room at head height. The virtual objects are 50cm×50cm in size, and their distances to the subjects are 1 to 1.5m. We perform the built-in visual calibration on the Magic Leap One for each subject.

For both devices, we ask the subjects to perform each of the four activities, i.e., *Read, Communicate, Browse*, and *Watch*, for five minutes. They can freely choose the stimuli that we have prepared:

- **Read**: we create three sets of text images from three digital reading materials that differ in both text layout and font size: a transcription of Richard Hamming's talk on "You and Your Research"; a chapter from the book "Rich Dad Poor Dad"; and a chapter from the book "Discrete Calculus".
- **Communicate**: seven monologue videos are prepared, including: three video clips extracted from an online interview with Anthony Fauci; two video clips extracted from the ACM Turing Award Laureate interview with Raj Reddy; and two online YouTube videos in which the speaker is giving advice on career development. All videos have only one speaker.

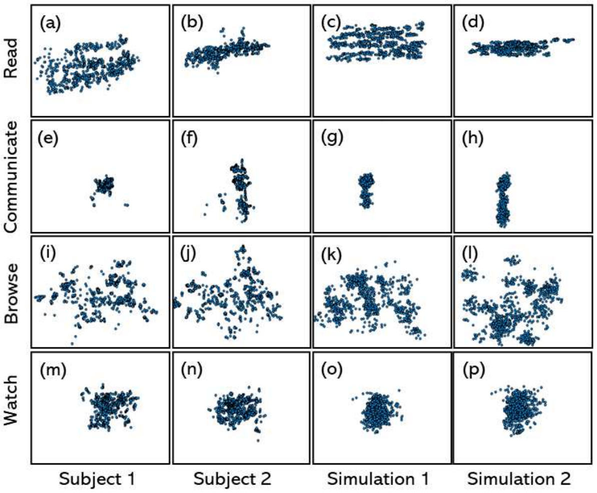


Figure 13: Comparison between the actual (left) and the simulated (right) gaze signals for the four activities. The four rows from top to bottom correspond to the four different activities: Read, Communicate, Browse, and Watch.

- **Browse**: we randomly select a subset of 200 images from a public painting image dataset [22] that contains 7,937 images of famous paintings. During the data collection, for each of the subjects, we randomly select 30 images from the subset and show each of the selected images to the subject for 10 seconds.
- **Watch**: we randomly pick six short documentary videos from the online video series "National Geographic Animals 101". Each video lasts 5 to 6 minutes.

The details of the stimuli used in the data collection can be found at https://github.com/EyeSyn/EyeSynResource.

6.2 Signal Level Evaluation

6.2.1 Setup. In this evaluation we leverage the Pupil Labs eye tracker to collect gaze signals from two subjects when they are performing the four activities. For each of the activities, we give the *same visual stimuli* to the two subjects, and ask them to perform each of the activities for 30 seconds. The stimuli used in this experiment are: (1) a page of text in the book "Rich Dad Poor Dad" for Read; (2) an interview video with Anthony Fauci for Communicate; (3) an image of a Paul Cezanne painting for Browse; and (4) a documentary video from the National Geographic series for Watch.

For gaze simulation, the scale parameters d, w, and h (defined in Figure 3) are set to 50cm, 40cm, and 30cm, respectively. Identical visual stimuli are also used as the inputs for gaze synthesis.

6.2.2 Signal Comparison. The scatter plots in Figure 13 compare the real gaze signals with the synthetic signals. The dots in each of the images are the 900 gaze points displayed in a normalized 2D plane (with X and Y coordinates ranging from 0 to 1). The four rows from top to bottom correspond to the gaze signals for Read, Communicate, Browse, and Watch, respectively. The two columns on the left correspond to the actual gaze signals of the two subjects; the two columns on the right are the synthesized signals generated in two simulation sessions.

First, the difference between the gaze signals of the two subjects demonstrates *the heterogeneity in human visual behavior*, even in the case where the *same visual stimuli and the same eye tracker* were used in the data collection. For instance, the gaze points shown in Figure 13(a) cover a wider range in the Y direction than the gaze points shown in Figure 13(b). This indicates that Subject 1 reads faster than Subject 2 (i.e., Subject 1 reads more lines in 30 seconds). Similarly, the gaze points in Figure 13(e) are clustered in a single area, which indicates that Subject 1 fixates his visual attention on a single facial region of the speaker in the monologue video. By contrast, the three clusters in Figure 13(f) indicate that Subject 2 switches her attention among the three facial regions of the speaker.

Second, by comparing the synthesized signal with the real gaze signal, we make the following observations for each of the activities:

- **Read:** Figures 13(a-d) show that the distinct "left-to-right" reading pattern [13, 14] in the actual gaze signals is well reproduced in the simulated signals. Figures 13(c,d) show that the diversity in reading speed is also well captured in the simulated signals.
- **Communicate:** As shown in Figures 13(g,h), similar to the real gaze signal, the synthesized gaze points are clustered in three areas that correspond to the three facial regions of the speaker. The results show that the VerbalGaze model introduced in Section 4.3 can effectively replicate the actual visual behaviors of "fixating on and switching attention between different facial regions" [25].
- **Browse:** Figures 13(i-l) indicate that the geometric patterns of the gaze signals when subjects are browsing the painting are well reproduced by the StaticScene model introduced in Section 4.4. Specifically, in both real and synthesized signals, the gaze points are clustered at different saliency regions of the painting.
- **Watch:** In the stimuli used for the Watch activity, the most salient object appears frequently at locations that are close to the center of the scene. Thus, for both real and synthesized eye movement signals, the gaze points are densely located around the center of the 2D plane. As shown in Figures 13(m-p), this geometrical pattern is well simulated by the DynamicScene model.

Overall, our results demonstrate the feasibility of using EyeSyn in synthesizing realistic eye movement signals that closely resemble the real ones. More specifically, our models can not only *replicate the distinct trends and geometric patterns* in the eye movement signal for each of the four activities, but can also *simulate the heterogeneity among subjects.* The latter is important as a synthesized training dataset that captures the heterogeneity in eye movements can potentially overcome the domain shift problem in gaze-based activity recognition and ensure better classification accuracy [12].

6.3 Performance in Activity Recognition

Below, we leverage the synthetic and real gaze datasets, *SynGaze* and *VisualProcessingActivity*, to investigate how EyeSyn can be used to improve the performance of gaze-based activity recognition. Specifically, we consider the few-shot learning scenario, where we aim to train the CNN-based classifier (Section 5.2) such that it can quickly adapt to new subjects with only K training instances ($K \in \{1, 2, 3, 5, 10\}$ is a small number) for each of the four activities.

We perform the evaluation on the VisualProcessingActivity dataset in the leave-one-subject-out manner, which has been used in previous studies [9, 12]. Specifically, we regard the data collected from one subject as the *target set* and the data collected from the remaining subjects as the *source set.* The single subject in the target set simulates the scenario where the system is deployed to a new subject with limited real gaze samples available for training (K samples per class). We denote the simulated gaze dataset SynGaze as the *synthetic training set* in our evaluation. The sensing window size is 30s with 50% overlap between consecutive windows.

6.3.1 Methods. We consider five strategies to train the CNN-based classifier for the few-shot learning scenario:

(S1) Real data + Image-based data augmentation: we use the few-shot samples, i.e., $4 \times K$ samples, from the target set to train the classifier and test it using the remaining data in the target set. This represents the scenario where we only have the data collected from the target subject. Moreover, we apply the ImageDataGenerator [66] in Keras to perform standard image-based data augmentation techniques during the training. Specifically, we apply horizontal and vertical shifts with the range of (-0.3, 0.3) to the input gaze heatmaps to simulate shifts of the gaze signal in both X and Y directions; we apply rotation augmentation with the range of (-10, 10) degrees to simulate variance in the gaze signal due to different head orientations; finally, we leverage the zoom augmentation with the range of (0.5, 1.5) to simulate the effects of different viewing distances.

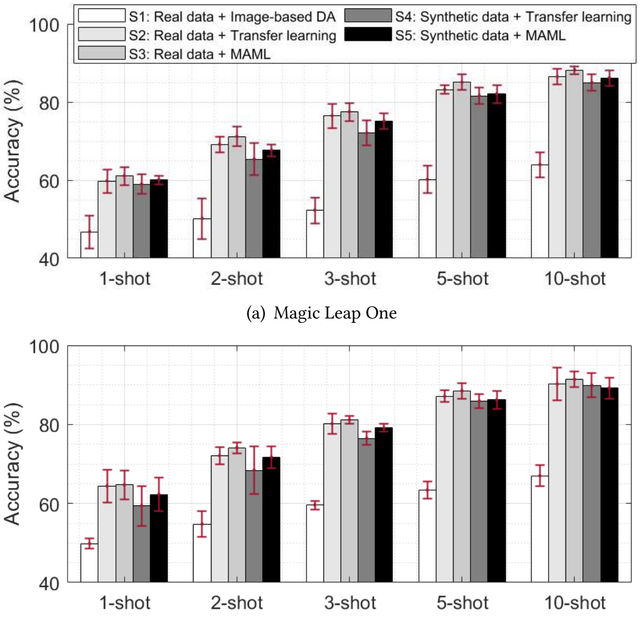
(S2) Real data + Transfer learning: we first train the CNN-based classifier on the source set. Then, we employ transfer learning [67] to transfer the trained model to the target set. In brief, we freeze the pre-trained weights of all the convolutional layers in the DNN architecture (shown in Table 3), and fine-tune the fully connected layers using the few-shot samples from the target set. This strategy represents the scenario where we have the access to the gaze samples collected from the other subjects during training. This method has been widely used for domain adaptation with few-shot instances [10, 17].

(S3) Real data + MAML: we apply the model-agnostic meta learning (MAML) [68] to train the classifier on the VisualProcessingActivity dataset. Specifically, we use the *source set* to train the classifier in the meta-training phase, and fine-tune it with the few-shot instances from the target set in the adaptation phase [68]. The MAML-based strategy is the state-of-the-art solution for few-shot gaze-based activity recognition [12]. Similar to strategy S2, this strategy also assumes the availability of the source set during the training.

(S4) Synthetic data + Transfer learning: we train the classifier on the synthetic training set, and leverage transfer learning to fine-tune the fully connected layers of the classifier using the few-shot samples from the target set. In contrast to strategies S2 and S3, it requires only the synthesized gaze data for training, and only the few-shot real gaze samples are needed during the fine-tuning stage.

(S5) Synthetic data + MAML: we apply MAML on the synthetic training set during the meta-training phase. Then, in the adaptation phase, we fine-tune all layers of the classifier by using the few-shot samples from the target set. Similar to strategy S4, we do not need any real gaze samples in the pre-training stage.

6.3.2 Overall result. The performance of the five learning strategies with different numbers of shots (K) is shown in Figure 14. Figures 14 (a) and (b) are the averaged accuracy over all the subjects who use the Magic Leap One and the Pupil Labs in the data



(a) Magic Leap One

(b) Pupil Labs

Figure 14: Accuracy of different training strategies in the few-shot learning scenario with gaze data collected from (a) Magic Leap One and (b) Pupil Labs.

collection, respectively. The error bar is the standard deviation across the subjects. We make the following observations.

First, strategy S1 achieves the worst accuracy in all examined cases, as the limited training samples lead to overfitting, which indicates that *standard image-based data augmentation cannot simulate the diversity in gaze signals even for the same subject.* By contrast, using the synthetic gaze signals for training, the transfer learning and MAML-based strategies, i.e., S4 and S5, improve upon the accuracy of S1 by 17.9% and 19.5% on average, respectively.

Second, leveraging the synthetic gaze dataset for training, S4 and S5 achieve good accuracy on datasets collected from both the Magic Leap One and the Pupil Labs. Moreover, since the two datasets are collected from different subjects in different environments, the results demonstrate the capability of the proposed models in capturing such diversity and improve the robustness of the classifier in heterogeneous sensing conditions.

Lastly, we compare the accuracy of the strategies that use real (S2 and S3) and synthetic (S4 and S5) gaze signals for training. The accuracy differences are further summarized in Table 4. As shown, for all examined cases, we see a negligible accuracy drop when using synthetic data for training. Specifically, for the data collected from the Magic Leap One and the Pupil Labs, we see only 0.8% to 4.2%, and 0.3% to 4.0% accuracy drop, respectively. Moreover, when the number of shots $K \geq 5$, the accuracy deficiency for transfer learning and MAML are less than 2% and 3%, respectively.

Note that the small accuracy gains achieved by S2 and S3 rely on a labor-intensive process to collect eye movement data from the other subjects. Based on our own experience, due to the calibration, experiment setup, instruction, and device failure, it takes more than 40 minutes to collect 20 minutes of gaze data with satisfactory quality from a single subject. Indeed, the labor-intensive and privacy-compromising [16] process has prohibited the collection of large-scale eye movement datasets, which is evidenced by the fact that *the sizes of current public gaze-based activity datasets are on the order of a couple of hours* [8, 9, 12]. By contrast, leveraging the massive gaze data simulated from the already-available images and videos for training, S4 and S5 eliminate the labor-intensive data collection and require only few-shot instances from the target subject for fine-tuning the classifier.

Table 4: The accuracy difference (in %) between the use of real and synthetic gaze signals for classifier training.

Eye tracker	Method	Number of shots (K)				
		10	5	3	2	1
Magic Leap One	Transfer learning (S2-S4)	1.5	1.6	4.2	3.8	0.8
	MAML (S3-S5)	2.0	3.0	2.4	3.5	1.2
Pupil Labs	Transfer learning (S2-S4)	0.3	1.2	3.7	3.3	4.0
	MAML (S3-S5)	2.2	2.3	2.1	2.4	2.4

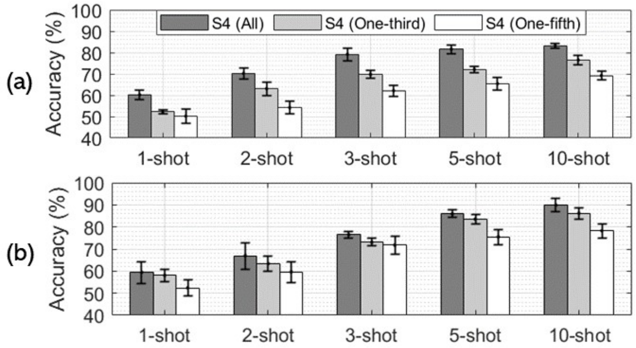


Figure 15: Accuracy with different sizes of synthetic data used in the training. The classifier is tested on the data collected from: (a) Magic Leap One and (b) Pupil Labs.

6.3.3 Impact of synthetic data size and sensing window size. Below, we examine how the amount of synthetic data used in training and the sensing window size will affect the recognition accuracy. We use strategy S4 as the training method in this evaluation.

First, we evaluate the recognition accuracy given different sizes of synthetic data used in training. Specifically, we use one-fifth, one-third, and all of the synthetic signals in *SynGaze* (in Section 5.1) to train the CNN-based classifier. Then, for each of the subjects, we apply transfer learning to fine-tune the classifier using few-shot (K) gaze samples from the corresponding target set. The results are shown in Figure 15. We observe that the accuracy increases with the size of synthetic data used in training. Note that, since we are using diverse image and video stimuli as the inputs for gaze simulation, a larger synthetic dataset indicates a higher diversity of input stimuli. Thus, the results indicate that the scalability of EyeSyn to diverse visual stimuli is crucial for the final recognition accuracy: taking the ready-to-use public image and video datasets as the inputs, EyeSyn can readily simulate a massive amount of diverse gaze signals, i.e., the 185 hours of data generated in the current work, to ensure good recognition accuracy.

Finally, we examine the impact of sensing window size on the recognition performance. As shown in Figure 16, for all the examined few-shot scenarios, the accuracy increases with the window size, as a larger sensing window contains more information about

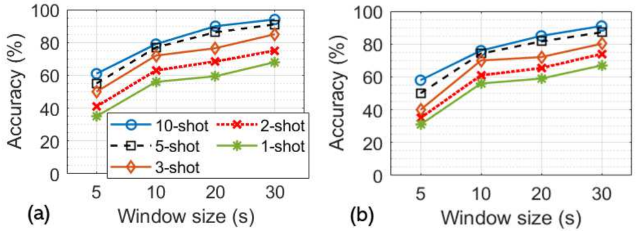


Figure 16: Accuracy with different sensing window sizes. The classifier is tested on the data collected from: (a) Magic Leap One and (b) Pupil Labs.

eye movements. Moreover, with a window size of five seconds, the accuracy drops significantly. This is because the five-second window is too short to contain enough distinct eye movement patterns. In fact, based on the statistics shown previously in Table 1 and Figure 8(b), a five-second window may contain only two fixation points, which is insufficient for activity recognition.

Overview: Our results demonstrate that the synthetic data can be incorporated with either transfer learning or MAML to achieve good recognition accuracy with only few-shot gaze instances required from the target sensing scenario (i.e., a new subject). More importantly, without sacrificing the recognition accuracy, the proposed work eliminates the need for the expensive and privacy-compromising large-scale eye movement dataset that is required by current state-of-the-art solutions [8, 12] for classifier training.

7 DISCUSSION

7.1 Limitations

Although EyeSyn embodies several psychology findings in the literature, its current design cannot fully replicate the complex mechanisms of human visual processing to synthesize eye movements for all subject groups. For instance, people with neurodevelopmental or mental disorders, such as autism spectrum disorder [69], schizophrenia [70], or social anxiety disorder [71], may exhibit atypical eye movement patterns in social interactions, e.g., avoiding direct eye contact with the communication partner. Moreover, decision making in visual attention is affected by many cognitive factors, such as the mental workload of the subject [7], the reward of different visual saliency [72], and the current cognitive task [73]. These cognitive factors have highly diverse impacts on eye movements [33]. The current design of the proposed generative models did not take these factors into account. In fact, the implementation of a generalized model is still an open challenge in visual behavior modeling research [28, 74], as it is difficult to have a one size fits all model that can synthesize visual attention for all subject groups and possible cognitive cases. Solving this problem requires future endeavors to integrate knowledge from various disciplines, such as psychology, neuroscience, and the social sciences.

7.2 Future Directions

EyeSyn can be readily extended to cover more complex scenarios by embodying the atypical eye movement characteristics of different subject groups in its design. For instance, current works in the neuropsychology literature [75, 76] have shown that individuals with autism spectrum disorder exhibit reduced visual attention to social and semantic stimuli, e.g., faces, but focus more on non-social and low-level stimuli, e.g., vehicles. To model this behavior, we can extend the current saliency-based fixation estimation method by taking the social and semantic properties of the underlying stimuli into account, e.g., we can assign a higher weight to fixation points that are associated with non-social and low-level stimuli, and vice versa. Similarly, subjects with schizophrenia are known to have strikingly different eye movement patterns during smooth pursuit (a type of eye movement in which the eyes remain fixated on a moving object) and visual search [77, 78]. For instance, when conducting smooth pursuit to track a moving stimulus using their eyes, the gaze positions for subjects with schizophrenia often lag behind the moving stimulus, as the speed of their eye movements cannot keep up with that of the moving visual target [78] due to the lesions in the superior temporal sulcus [79]. Thus, to model this atypical eye movement pattern in scene perception, we can introduce a lag when associating the coordinates of the selected salient location with the simulated gaze points. Overall, we believe the current design of EyeSyn can serve as an important first step towards a more comprehensive suite of models for eye movement synthesis.

7.3 Potential Applications

EyeSyn can also benefit applications that feature animated characters or avatars [80], such as video games [29, 30], social conversational agents [28], and photo-realistic facial animation for virtual reality [81–83]. In these applications, the virtual avatars should have realistic eye movements that are consistent with the ongoing activity and the visual stimuli. The gaze signals synthesized by EyeSyn can be used as the inputs of the avatar model to produce realistic eye movements for the facial animation. EyeSyn can also be used to estimate spatial-temporal attention when a user is viewing different visual stimuli [84, 85]. The estimated fixation locations and saccade trajectories can further serve as the inputs for attention-adaptive systems to improve user perceived quality in services such as webpage loading [86], gaze-contingent rendering [87], and foveated rendering in virtual and augmented reality [88, 89].

8 CONCLUSION

In this work we present EyeSyn, a novel suite of *psychology-inspired generative models* that leverage only publicly available images and videos to synthesize a *realistic* and *arbitrarily large* eye movement dataset for DNN training. Our evaluation demonstrates the efficacy of EyeSyn in replicating the distinct patterns in actual gaze signals, as well as in simulating the gaze diversity that results from different measurement setups and subject heterogeneity. Using gaze-based museum activity recognition as a case study, we show that a CNN-based classifier trained by the synthetic gaze signals can achieve 90% accuracy, without the need for labor-intensive and privacy-compromising data collection.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers and the shepherd for their insightful comments and guidance. This work was supported in part by NSF grants CSR-1903136 and CNS-1908051, NSF CAREER Award IIS-2046072, and an IBM Faculty Award.

REFERENCES

[1] "Eye tracking on HoloLens 2," https://docs.microsoft.com/en-us/windows/mixed-reality/design/eye-tracking.

[2] "Magic Leap One," https://www.magicleap.com/en-us/magic-leap-1.

[3] "VIVE Pro Eye," https://www.vive.com/eu/product/vive-pro-eye/.

[4] N. Valliappan, N. Dai, E. Steinberg, J. He, K. Rogers, V. Ramachandran, P. Xu, M. Shojaeizadeh, L. Guo, K. Kohlhoff *et al.*, "Accelerating eye movement research via accurate and affordable smartphone eye tracking," *Nature Communications*, vol. 11, no. 1, pp. 1–12, 2020.

[5] E. Wood and A. Bulling, "EyeTab: Model-based gaze estimation on unmodified tablet computers," in *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*, 2014, pp. 207–210.

[6] Y. Sugano, X. Zhang, and A. Bulling, "AggreGaze: Collective estimation of audience attention on public displays," in *Proceedings of the ACM Annual Symposium on User Interface Software and Technology*, 2016, pp. 821–831.

[7] L. Fridman, B. Reimer, B. Mehler, and W. T. Freeman, "Cognitive load estimation in the wild," in *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–9.

[8] N. Srivastava, J. Newn, and E. Velloso, "Combining low and mid-level gaze features for desktop activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, p. 189, 2018.

[9] K. Kunze, Y. Utsumi, Y. Shiga, K. Kise, and A. Bulling, "I know what you are reading: Recognition of document types using mobile eye tracking," in *Proceedings of the ACM International Symposium on Wearable Computers*, 2013, pp. 113–116.

[10] H. Wu, J. Feng, X. Tian, E. Sun, Y. Liu, B. Dong, F. Xu, and S. Zhong, "EMO: Real-time emotion recognition from single-eye images for resource-constrained eyewear devices," in *Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services*, 2020, pp. 448–461.

[11] S. Ahn, C. Kelton, A. Balasubramanian, and G. Zelinsky, "Towards predicting reading comprehension from gaze behavior," in *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*, 2020, pp. 1–5.

[12] G. Lan, B. Heit, T. Scargill, and M. Gorlatova, "GazeGraph: Graph-based few-shot cognitive context sensing from human visual behavior," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, 2020, pp. 422–435.

[13] K. Rayner, "The 35th Sir Frederick Bartlett lecture: Eye movements and attention in reading, scene perception, and visual search," *Quarterly Journal of Experimental Psychology*, vol. 62, no. 8, pp. 1457–1506, 2009.

[14] G. Öquist and K. Lundin, "Eye movement study of reading text on a mobile phone using paging, scrolling, leading, and RSVP," in *Proceedings of the ACM International Conference on Mobile and Ubiquitous Multimedia*, 2007, pp. 176–183.

[15] M. K. Eckstein, B. Guerra-Carrillo, A. T. M. Singley, and S. A. Bunge, "Beyond eye gaze: What else can eyetracking reveal about cognition and cognitive development?" *Developmental Cognitive Neuroscience*, vol. 25, pp. 69–91, 2017.

[16] J. Li, A. R. Chowdhury, K. Fawaz, and Y. Kim, "Kalεido: Real-time privacy control for eye-tracking systems," in *Proceedings of the USENIX Security Symposium*, 2021, pp. 1793–1810.

[17] S. A. Rokni, M. Nourollahi, and H. Ghasemzadeh, "Personalized human activity recognition using convolutional neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[18] T. Gong, Y. Kim, J. Shin, and S.-J. Lee, "MetaSense: Few-shot adaptation to untrained conditions in deep mobile sensing," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, 2019, pp. 110–123.

[19] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional GANs," *arXiv preprint arXiv:1706.02633*, 2017.

[20] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series generative adversarial networks," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[21] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 12 104–12 114.

[22] "Best artworks of all time dataset," https://www.kaggle.com/ikarus777/best-artworks-of-all-time.

[23] "Noisy and Rotated Scanned Documents Dataset," https://www.kaggle.com/sthabile/noisy-and-rotated-scanned-documents.

[24] K. Rayner, "Eye movements in reading and information processing: 20 years of research." *Psychological Bulletin*, vol. 124, no. 3, p. 372, 1998.

[25] J. Jiang, K. Borowiak, L. Tudge, C. Otto, and K. von Kriegstein, "Neural mechanisms of eye contact when listening to another person talking," *Social Cognitive and Affective Neuroscience*, vol. 12, no. 2, pp. 319–328, 2017.

[26] B. 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*, vol. 7, no. 14, pp. 4–4, 2007.

[27] J. Nie, Y. Hu, Y. Wang, S. Xia, and X. Jiang, "SPIDERS: Low-cost wireless glasses for continuous in-situ bio-signal acquisition and emotion recognition," in *Proceedings of IEEE/ACM International Conference on Internet-of-Things Design and Implementation*, 2020, pp. 27–39.

[28] K. Ruhland, S. Andrist, J. Badler, C. Peters, N. Badler, M. Gleicher, B. Mutlu, and R. Mcdonnell, "Look me in the eyes: A survey of eye and gaze animation for virtual agents and artificial systems," in *Proceedings of Eurographics State of the Art Reports*, 2014, pp. 69–91.

[29] S. H. Yeo, M. Lesmana, D. R. Neog, and D. K. Pai, "Eyecatch: Simulating visuomotor coordination for object interception," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1–10, 2012.

[30] S. P. Lee, J. B. Badler, and N. I. Badler, "Eyes alive," in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, 2002, pp. 637–644.

[31] A. Duchowski, S. Jörg, A. Lawson, T. Bolte, L. Świrski, and K. Krejtz, "Eye movement synthesis with 1/f pink noise," in *Proceedings of the ACM SIGGRAPH Conference on Motion in Games*, 2015, pp. 47–56.

[32] A. Duchowski, S. Jörg, T. N. Allen, I. Giannopoulos, and K. Krejtz, "Eye movement synthesis," in *Proceedings of the ACM Symposium on Eye Tracking Research & Applications*, 2016, pp. 147–154.

[33] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 185–207, 2012.

[34] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[35] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Reviews Neuroscience*, vol. 2, no. 3, pp. 194–203, 2001.

[36] R. J. Peters and L. Itti, "Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[37] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proceedings of IEEE International Conference on Computer Vision*, 2009, pp. 2106–2113.

[38] Y. Zhai and M. Shah, "Visual attention detection in video sequences using spatiotemporal cues," in *Proceedings of the ACM International Conference on Multimedia*, 2006, pp. 815–824.

[39] J. Li, Y. Tian, T. Huang, and W. Gao, "Probabilistic multi-task learning for visual saliency estimation in video," *International Journal of Computer Vision*, vol. 90, no. 2, pp. 150–165, 2010.

[40] Z. Hu, C. Zhang, S. Li, G. Wang, and D. Manocha, "SGaze: A data-driven eye-head coordination model for realtime gaze prediction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 5, pp. 2002–2010, 2019.

[41] 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*, vol. 27, no. 10, pp. 5142–5154, 2018.

[42] Y. Zhu, G. Zhai, X. Min, and J. Zhou, "The prediction of saliency map for head and eye movements in 360 degree images," *IEEE Transactions on Multimedia*, vol. 22, no. 9, pp. 2331–2344, 2020.

[43] F. Vitu, J. K. O'Regan, and M. Mittau, "Optimal landing position in reading isolated words and continuous text," *Perception & Psychophysics*, vol. 47, no. 6, pp. 583–600, 1990.

[44] K. Rayner and G. W. McConkie, "What guides a reader's eye movements?" *Vision Research*, vol. 16, no. 8, pp. 829–837, 1976.

[45] K. Rayner, S. C. Sereno, and G. E. Raney, "Eye movement control in reading: A comparison of two types of models," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 22, no. 5, p. 1188, 1996.

[46] L. G. Lusk and A. D. Mitchel, "Differential gaze patterns on eyes and mouth during audiovisual speech segmentation," *Frontiers in Psychology*, vol. 7, p. 52, 2016.

[47] T. Foulsham, "Eye movements and their functions in everyday tasks," *Eye*, vol. 29, no. 2, pp. 196–199, 2015.

[48] S. Vassallo, S. L. Cooper, and J. M. Douglas, "Visual scanning in the recognition of facial affect: Is there an observer sex difference?" *Journal of Vision*, vol. 9, no. 3, pp. 11–11, 2009.

[49] M. Freeth, T. Foulsham, and A. Kingstone, "What affects social attention? social presence, eye contact and autistic traits," *PloS One*, vol. 8, no. 1, p. e53286, 2013.

[50] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vision Research*, vol. 40, no. 10-12, pp. 1489–1506, 2000.

[51] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[52] J. Najemnik and W. S. Geisler, "Optimal eye movement strategies in visual search," *Nature*, vol. 434, no. 7031, pp. 387–391, 2005.

[53] "Skin and Bones application in the Smithsonian National Museum of Natural History," https://naturalhistory.si.edu/exhibits/bone-hall.

[54] Z. Liu, G. Lan, J. Stojkovic, Y. Zhang, C. Joe-Wong, and M. Gorlatova, "CollabAR: Edge-assisted collaborative image recognition for mobile augmented reality," in *Proceedings of ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2020, pp. 301–312.

[55] "Pupil Labs eye tracker," https://pupil-labs.com/.

[56] D. Aks, G. Zelinsky, and J. Sprott, "Memory across eye-movements: 1/f dynamic in visual search," *Journal of Vision*, vol. 1, no. 3, pp. 230–230, 2001.

[57] N. J. Kasdin, "Discrete simulation of colored noise and stochastic processes and 1/f power law noise generation," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 802–827, 1995.

[58] K. Holmqvist, M. Nyström, and F. Mulvey, "Eye tracker data quality: what it is

and how to measure it," in *Proceedings of the ACM Symposium on Eye Tracking Research & Applications*, 2012, pp. 45–52.

[59] J. Johnsson and R. Matos, "Accuracy and precision test method for remote eye trackers," *Test Specification of Tobii Technology*, 2011.

[60] "Tesseract Open-Source OCR," https://opensource.google/projects/tesseract.

[61] R. Smith, D. Antonova, and D.-S. Lee, "Adapting the Tesseract open source OCR engine for multilingual OCR," in *Proceedings of the International Workshop on Multilingual OCR*, 2009, pp. 1–8.

[62] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. I–I.

[63] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proceedings of the ACM Symposium on Eye Tracking Research & Applications*, 2000, pp. 71–78.

[64] K. Rayner and M. Castelhano, "Eye movements," *Scholarpedia*, vol. 2, no. 10, p. 3649, 2007.

[65] "iMet Collection Artwork Dataset," https://github.com/visipedia/imet-fgvcx.

[66] "Keras ImageGenerator," https://keras.io/api/preprocessing/image/#imagedatagenerator-class.

[67] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proceedings of the International Conference on Neural Information Processing Systems*, 2014, pp. 3320–3328.

[68] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of International Conference on Machine Learning*, 2017, pp. 1126–1135.

[69] K. M. Dalton, B. M. Nacewicz, T. Johnstone, H. S. Schaefer, M. A. Gernsbacher, H. H. Goldsmith, A. L. Alexander, and R. J. Davidson, "Gaze fixation and the neural circuitry of face processing in autism," *Nature Neuroscience*, vol. 8, no. 4, pp. 519–526, 2005.

[70] S.-H. Choi, J. Ku, K. Han, E. Kim, S. I. Kim, J. Park, and J.-J. Kim, "Deficits in eye gaze during negative social interactions in patients with schizophrenia," *The Journal of Nervous and Mental Disease*, vol. 198, no. 11, pp. 829–835, 2010.

[71] F. R. Schneier, T. L. Rodebaugh, C. Blanco, H. Lewin, and M. R. Liebowitz, "Fear and avoidance of eye contact in social anxiety disorder," *Comprehensive Psychiatry*, vol. 52, no. 1, pp. 81–87, 2011.

[72] V. Navalpakkam, C. Koch, A. Rangel, and P. Perona, "Optimal reward harvesting in complex perceptual environments," *Proceedings of the National Academy of Sciences*, vol. 107, no. 11, pp. 5232–5237, 2010.

[73] V. Navalpakkam and L. Itti, "Modeling the influence of task on attention," *Vision Research*, vol. 45, no. 2, pp. 205–231, 2005.

[74] J. Gutiérrez, Z. Che, G. Zhai, and P. Le Callet, "Saliency4ASD: Challenge, dataset and tools for visual attention modeling for autism spectrum disorder," *Signal Processing: Image Communication*, vol. 92, p. 116092, 2021.

[75] G. Dawson, S. J. Webb, and J. McPartland, "Understanding the nature of face processing impairment in autism: Insights from behavioral and electrophysiological studies," *Developmental Neuropsychology*, vol. 27, no. 3, pp. 403–424, 2005.

[76] N. J. Sasson, J. T. Elison, L. M. Turner-Brown, G. S. Dichter, and J. W. Bodfish, "Brief report: Circumscribed attention in young children with autism," *Journal of Autism and Developmental Disorders*, vol. 41, no. 2, pp. 242–247, 2011.

[77] P. S. Holzman, L. R. Proctor, and D. W. Hughes, "Eye-tracking patterns in schizophrenia," *Science*, vol. 181, no. 4095, pp. 179–181, 1973.

[78] K. Morita, K. Miura, K. Kasai, and R. Hashimoto, "Eye movement characteristics in schizophrenia: A recent update with clinical implications," *Neuropsychopharmacology Reports*, vol. 40, no. 1, pp. 2–9, 2020.

[79] M. Dursteler and R. H. Wurtz, "Pursuit and optokinetic deficits following chemical lesions of cortical areas MT and MST," *Journal of Neurophysiology*, vol. 60, no. 3, pp. 940–965, 1988.

[80] K. Ruhland, C. E. Peters, S. Andrist, J. B. Badler, N. I. Badler, M. Gleicher, B. Mutlu, and R. McDonnell, "A review of eye gaze in virtual agents, social robotics and HCI: Behaviour generation, user interaction and perception," *Computer Graphics Forum*, vol. 34, no. 6, pp. 299–326, 2015.

[81] S.-E. Wei, J. Saragih, T. Simon, A. W. Harley, S. Lombardi, M. Perdoch, A. Hypes, D. Wang, H. Badino, and Y. Sheikh, "VR facial animation via multiview image translation," *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1–16, 2019.

[82] G. Schwartz, S.-E. Wei, T.-L. Wang, S. Lombardi, T. Simon, J. Saragih, and Y. Sheikh, "The eyes have it: An integrated eye and face model for photorealistic facial animation," *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 91–1, 2020.

[83] A. Richard, C. Lea, S. Ma, J. Gall, F. De la Torre, and Y. Sheikh, "Audio- and gaze-driven facial animation of codec avatars," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 41–50.

[84] Y. Li, P. Xu, D. Lagun, and V. Navalpakkam, "Towards measuring and inferring user interest from gaze," in *Proceedings of the ACM International Conference on World Wide Web Companion*, 2017, pp. 525–533.

[85] P. Xu, Y. Sugano, and A. Bulling, "Spatio-temporal modeling and prediction of visual attention in graphical user interfaces," in *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, 2016, pp. 3299–3310.

[86] C. Kelton, J. Ryoo, A. Balasubramanian, and S. R. Das, "Improving user perceived page load times using gaze," in *Proceedings of USENIX Symposium on Networked Systems Design and Implementation*, 2017, pp. 545–559.

[87] E. Arabadzhiyska, O. T. Tursun, K. Myszkowski, H.-P. Seidel, and P. Didyk, "Saccade landing position prediction for gaze-contingent rendering," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–12, 2017.

[88] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn, "Towards foveated rendering for gaze-tracked virtual reality," *ACM Transactions on Graphics*, vol. 35, no. 6, pp. 1–12, 2016.

[89] J. Kim, Y. Jeong, M. Stengel, K. Akşit, R. Albert, B. Boudaoud, T. Greer, J. Kim, W. Lopes, Z. Majercik *et al.*, "Foveated AR: Dynamically-foveated augmented reality display," *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1–15, 2019.