A Scanner Deeply: Predicting Gaze Heatmaps on Visualizations Using Crowdsourced Eye Movement Data

Sungbok Shin, Sunghyo Chung, Sanghyun Hong, and Niklas Elmqvist, *Senior Member, IEEE*

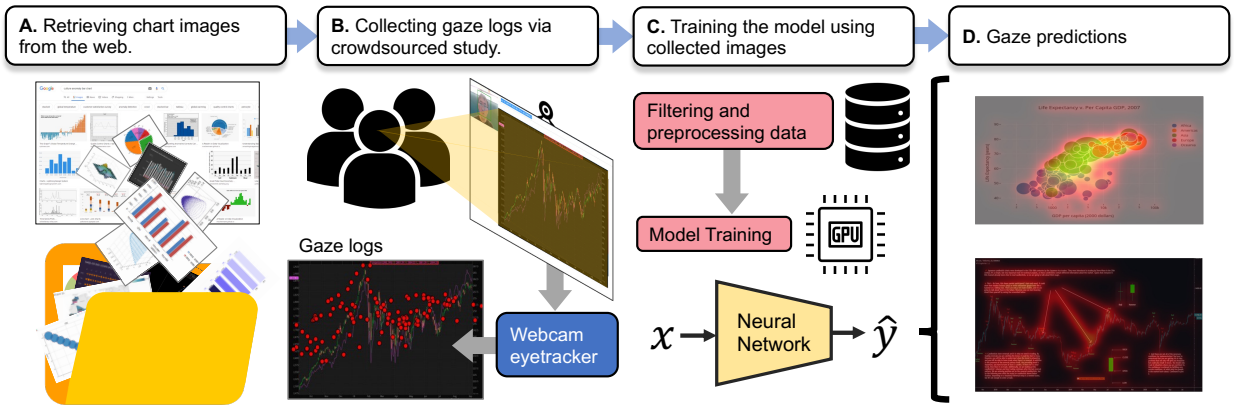

Fig. 1: **Summary of the Scanner Deeply pipeline.** Pipeline for developing a gaze prediction model that, given an image as an input, produces a saliency map for an image containing a visualization. First, we collect more than 10,000 images that contain one chart per image (A). Second, using the collected images we conduct a crowdsourced study on Amazon Mechanical Turk to gather gaze logs using a webcam eyetracker (B). Third, using the image collection with gaze log annotations, we train a model called SimpleNet, a CNN-based neural network model (C). Gaze predictions are shown in (D) in the form of saliency maps.

Abstract—Visual perception is a key component of data visualization. Much prior empirical work uses eye movement as a *proxy* to understand human visual perception. Diverse apparatus and techniques have been proposed to collect eye movements, but there is still no optimal approach. In this paper, we review 30 prior works for collecting eye movements based on three axes: (1) the *tracker* technology used to measure eye movements; (2) the *image stimulus* shown to participants; and (3) the *collection methodology* used to gather the data. Based on this taxonomy, we employ a webcam-based eyetracking approach using task-specific visualizations as the stimulus. The low technology requirement means that virtually anyone can participate, thus enabling us to collect data at large scale using crowdsourcing: approximately 12,000 samples in total. Choosing visualization images as stimulus means that the eye movements will be specific to perceptual tasks associated with visualization. We use these data to propose a SCANNER DEEPLY, a virtual eyetracker model that, given an image of a visualization, generates a gaze heatmap for that image. We employ a computationally efficient, yet powerful convolutional neural network for our model. We compare the results of our work with results from the DVS model and a neural network trained on the Salicon dataset. The analysis of our gaze patterns enables us to understand how users grasp the *structure* of visualized data. We also make our stimulus dataset of visualization images available as part of this paper's contribution.

Index Terms—Gaze prediction, visualization, webcam-based eye-tracking, crowdsourcing, deep learning.

1 INTRODUCTION

Your eyes are not just windows to your soul, but also to your ability to read and perceive the visual content on a computer screen. Thus, the ability to detect, track, and predict one's eye movements is used in numerous fields such as computer vision [32, 63], human-computer interaction [27, 48], and natural language processing [2, 12]. This is doubly true for data visualization, where tracking the user's gaze on an interactive chart is key to understanding sophisticated mechanisms behind how humans perceive them [3, 35]. However, current eyetracker hardware is bulky and costly, thus preventing widespread use, and the resulting data is often large in scope and difficult to interpret. Recent progress in machine learning and computer vision has made web-based eyetracking using standard webcams feasible [33, 52, 55]. However, such techniques tend to be less accurate than specialized eyetracking hardware and are sensitive to varying lighting conditions, webcam placement, and even the user's appearance.

At the same time, universally available eyetracking would undoubtedly be highly useful during the design and development phases of a data visualization, allowing the designer to quickly gauge the appearance and salience of a visual representation. What if you could, for example, figure out whether a peak in data stands out, or if a relationship between two items can be seen in the visual clutter? However, to the best of our knowledge, such functionality does not yet exist.

In this paper, we propose a virtual eye tracker using deep learning—a SCANNER DEEPLY—that, given an image of a visualization as an input, automatically generates a gaze heatmap for the image. Figure 1 outlines our approach for developing this deep learning-based eye tracker. We first gathered a large corpus of some 11,000 visualization images from the web. Then we collected eyetracking data at scale by conducting a crowdsourced user study on Amazon Mechanical Turk that leverages an existing webcam-based eyetracking technique [55] to collect gaze logs for our visualization corpus. We used these images and the annotated

- *Sungbok Shin and Niklas Elmqvist are with University of Maryland, College Park. E-mails: sbshin90, elm@umd.edu.*
- *Sunghyo Chung is with Kakao Corp. E-mail: shawn.chung@kakaocorp.com.*
- *Sanghyun Hong is with Oregon State University. E-mail: sanghyun.hong@oregonstate.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

Table 1: **Taxonomy of prior work.** We categorize the 30 papers that collect eye movement data with three dimensions: (1) the tracker, (2) image stimulus, and (3) collection methodology. Note that the papers are listed in chronological order. We found that there is no work that employs the same collection approach as ours, i.e., collecting webcam-based crowdsourcing eye movements on chart images.

	Title (shortened)	Year	Eyetracker	Image Stimulus	Collection Methodology
1	A Model of Saliency-Based Visual Attention for Rapid... [26]	1998	eyetracker	natural	lab experiment
2	Using Eye Tracking to Investigate Graph Layout... [25]	2007	eyetracker	charts	lab experiment
3	Visual Perception of Parallel Coordinate Visualizations [64]	2009	eyetracker	charts	lab experiment
4	Learning to Predict Where Humans Look [30]	2009	eyetracker	natural	lab experiment
5	Eye Movements During Mindless Reading... [59]	2010	eyetracker	texts	lab experiment
6	Evaluation of Traditional, Orthogonal, and Radial... [6]	2011	eyetracker	charts	lab experiment
7	Findings while Investigating Visualizations for... [35]	2012	eyetracker	table	lab experiment
8	User-Adaptive Information Visualization - Using Eye... [66]	2013	eyetracker	charts	lab experiment
9	Predicting Affect from Gaze Data During Interaction... [28]	2014	eyetracker	texts	lab experiment
10	A Crowdsourced Alternative to Eye-tracking for... [34]	2015	mouse/cursor	charts	crowdsourcing
11	TurkerGaze: Crowdsourcing Saliency with Webcam... [71]	2015	webcam	natural	crowdsourcing
12	Constructing Models of User and Task Characteristics... [16]	2015	eyetracker	charts	lab experiment
13	SALICON: Saliency in Context [29]	2015	mouse/cursor	natural	lab experiment
14	Predicting Confusion in Information Visualization... [40]	2016	eyetracker	charts	lab experiment
15	Do graph readers prefer the graph type most suited to... [67]	2016	eyetracker	charts	lab experiment
16	WebGazer: Scalable Webcam Eye Tracking Using User... [55]	2016	webcam	natural	crowdsourcing
17	Eye Tracking for Everyone [37]	2016	phone camera	eye gaze	crowdsourcing
18	Beyond Memorability: Visualization Recognition and... [3]	2016	eyetracker	charts	lab experiment
19	Zone out no More: Mitigating Mind Wandering... [14]	2017	eyetracker	texts	lab experiment
20	Learning Visual Importance for Graphic Designs and ... [10]	2017	mouse/cursor	charts	crowdsourcing
21	SearchGazer: Webcam Eye Tracking for Remote... [54]	2017	webcam	web	crowdsourcing
22	Fauxvea: Crowdsourcing Gaze Location Estimates for... [18]	2017	mouse/cursor	charts	crowdsourcing
23	Saliency Revisited: Analysis of Mouse Movements... [68]	2017	mouse/cursor	natural	lab experiment
24	Patterns of Attention: How Visualizations are Read... [49]	2017	eyetracker	charts	lab experiment
25	Revisiting Video Saliency: A Large-scale Benchmark... [70]	2018	eyetrackers	natural	lab experiment
26	Exploring Visual Attention and Saliency Modeling... [56]	2018	eyetrackers	natural	lab experiment
27	Predicting Visual Importance Across Graphic Design... [15]	2020	mouse/cursor	natural/graphics	crowdsourcing
28	Visual Saliency Model Based on Crowdsourcing Eye... [11]	2020	webcam	natural	crowdsourcing
29	TurkEyes: A Web-Based Toolbox for Crowdsourcing... [52]	2020	mouse/cursor/zoom	natural	crowdsourcing
30	Gaze-driven Adaptive Interventions for MSNV... [41]	2021	eyetracker	charts	lab experiment
–	SCANNER DEEPLY (OUR WORK)	2022	**webcam**	**charts**	**crowdsourcing**

gaze logs to train a state-of-the-art convolutional neural network-based model [58] that predicts gaze heatmap when an image is given as an input. Based on the images created from Scanner Deeply, we present its qualitative features and evaluate our work using a model trained on the Salicon dataset and the DVS model. We also present a preprocessing method that effectively removes noisy gaze dots that improves the quality of webcam-based gaze dots. Our work demonstrates that gaze patterns on visualizations are task- and domain-specific.

Contributions. To sum up, the contributions of our work are

- Our Scanner Deeply pipeline, which collects large-scale chart images and uses webcam-based eyetracker, a low-technology requirement, to collect gaze dots and train the model on a neural network. Our approach is based on a taxonomy of 30 prior works.
- A preprocessing technique to effectively remove noise from raw eye movement data.
- Qualitative and quantitative evaluations of our work compared with other gaze prediction models.
- The disclosure of the dataset upon the acceptance of our work.

2 OVERVIEW

We review works that study visual perception via human eye movements. As summarized in Table 1, diverse apparatus and techniques have been proposed to collect eye movements data. We taxonomize those works based on three axes: tracker technology, image stimulus, and collection methodology. We detail why some choices are not practically desirable given the recent research trends. We conclude this section with a discussion on our choices.

2.1 Understanding Visual Perception via Eye Movements

Research on understanding human perception using eye movement started more than three decades ago [26], and the methods for collecting eye movements have made advances over time. One form of early works that involve studies with eye movement data explore different human reactions and patterns as a lab study. Eye movement data are typically collected using eyetracking hardware. Examples of these experiments include understanding mind wandering patterns when reading text [59], how people react under tabular visualizations [35], and how people make sense of unfamiliar visualizations [43]. Another form of early work aims at developing gaze prediction models. First introduced by Itti et al. [26], there has been many attempts to build models predicting gaze views of natural images, e.g., models using concepts from information theory to predict salience on natural images [5, 22].

As eyetracker started to gain more popularity in human perception research than the past, there has been growing demand for gaze prediction models with high quality. This brought light to numerous data-driven gaze prediction models [47, 53] and also to large-scale gaze logs on natural image datasets [30]. Two most commonly used image datasets for training and evaluating saliency models are (1) CAT2000 benchmark [8] and (2) Salicon [29]. CAT2000 contains 2,000 images from 20 different categories, and Salicon contains 10,000 images drawn from MS COCO [44]. Salicon provides highly varied and natural images along with ground-truth fixation annotations. With rapidly developing deep learning techniques, the performance of saliency maps on the benchmark datasets have also rapidly ameliorated [13, 38, 39]. Furthermore, as crowdsourced platforms became available, various methods have been developed to democratize the eyetracking process, such as cursor-based eyetrackers [33], or webcam-based eyetrackers [55, 71].

Initial work in the visualization community on gaze prediction builds models using natural image datasets [19]. Inspired by the works showing that humans focus on texts while analyzing charts, Matzen et al. proposed DVS [50], a gaze prediction model that utilizes a linear combination of a model based on natural images [22] and text optimizers [50]. However, it has been unclear whether those models based on natural images are effective on visualization images. In evaluation (§5.1), we

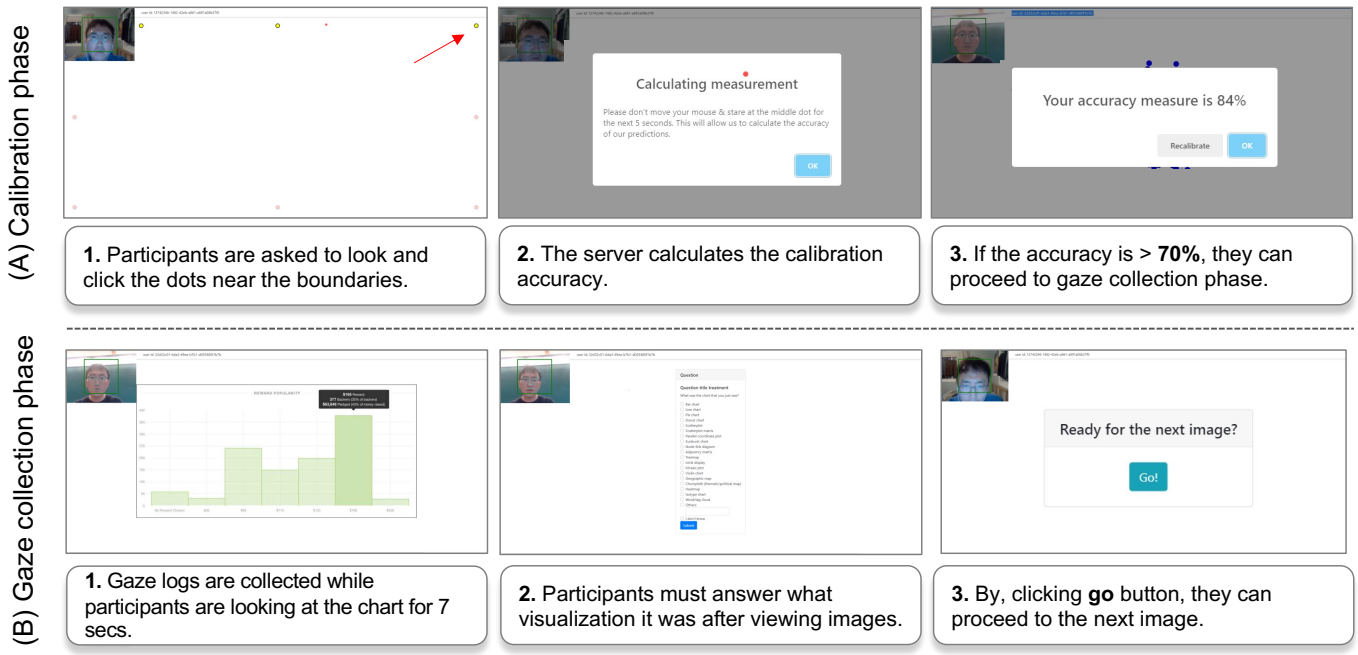

Fig. 2: **Calibration and gaze collection.** Participants click on 8 dots that are placed near the boundaries of the screen. Each dot turns yellow once the user has clicked on it five times (A-1). Then, the calibration accuracy is measured by making users look at the middle dot at the center of the screen (A-2). If the accuracy is above 70%, then participants can proceed to the gaze collection phase (A-3). In the gaze collection phase, participants are asked to look at the chart for 7 seconds, during which time the user's gaze is tracked (B-1). Then, they are asked to answer what type of chart the image is (B-2). After they submit the answer, participants look at the next image by clicking "Go" button at the question "Ready for the next image?" We customize the code WebGazer developed by Papoutsaki et al. [55] in creating the website.

show that such a model is less effective than our models trained on a visualization dataset, which necessitates developing a task-specific—i.e., focusing on data visualization—data corpus. It also has been unknown whether neural networks that improve the performance of gaze prediction tasks in natural image domains, are effective for visualization images, especially for charts. Our work aims at addressing this knowledge gap by collecting a large-scale gaze dataset of charts and training neural networks for gaze prediction on the data.

2.2 Choice of Apparatus and Techniques in Prior Work

Building gaze prediction models requires a corpus of eye movement data. Prior work therefore utilizes diverse apparatus and techniques for collecting datasets. Here, we review 30 prior works and assess the advantages and disadvantages of their collection methods. We collect those papers from visualization (IEEE TVCG, IEEE VIS, etc.), computer vision (IEEE CVPR, IEEE ICCV, etc.), and human-computer interaction (ACM CHI, ACM UIST, etc.). We consider studies comprehensively, i.e., they deal with eye movements on visualizations, natural images, texts, and webpages. We then evaluate their choices based on three axes: (1) the *tracker* that is used to track eye movement of a participant, (2) the *image stimulus*, or the type of images that is shown to participants, and (3) the *collection methodology* that is used to gather the data from participants. We have listed those works in Table 1.

Eyetracker. Understanding one's eye movement patterns provide us with various information about the attention the person is focusing on. For example, the fixation data about an image play as indicators of important regions within the image. In the field of visualization, by knowing which part of the chart would people's attention be most focused, the visualization designer can slightly change her design to better meet her intentions. There exist three types of methods that are used for eyetracking: specialized eyetracking hardware, indirect measurement methods, and general-purpose webcams.

To begin with, the benefit of eyetracking hardware is that it provides an accurate measurement of one's eye gaze. As a result, it has been extensively used in lab experiments. However, the quality of gaze dots are largely dependent on their prices. The performance of some of the inexpensive ones is somewhat questionable, and high-end eyetrackers can cost upwards of $20,000. For this reason, and also because of the difficulty in recruiting participants for laboratory sessions, experiments involving eyetracking devices are generally known to be expensive.

As an alternative, there have been attempts to collect gaze information without directly tracking the user's eyes. One approach is the cursor-based gaze collection technique. This is based on the assumption that the movement of cursor is correlated with the eye movement when looking at a screen [24]—or at least that participants can reliably be asked to use the mouse cursor in this way. Cursor-based eyetrackers are effective in conveying the person's location of visual attention [24]. Because of this characteristics, various research collects gaze dots using this approach [29, 33, 34]. However, cursor-based approaches do have limits and are not a completely accurate replacement for eyetracking devices. It loses some of its information, because it cannot track the person's eye when she is moving her eyes without moving the cursor.

There is another alternative that is being studied—the use of general-purpose webcams as an eyetracking method. However, webcams have one infamous hurdle that needs to be addressed: consistent calibration of the eye to the screen. This is mainly because calibration in webcams are very sensitive to even small head motions. Jiang et al. [29] argue that collecting large-scale dataset via general-purpose webcams is not possible, especially in an uncontrolled setting. Because of this, with current technology, gaze collection can only be maintained for a short period of time after calibration [71]. Furthermore, although it is still imperfect technology, we think that it is by far the best method to measure directly one's eyes, affordably, and still yield large-size data.

Image Stimulus. Two types of data can be used to predict gaze for visualization studies: visualization images, and natural images.

For the former, stimulus is often confined to a particular type of chart (e.g., node-link diagrams, parallel coordinate diagrams, etc.), and the size of the data corpus is typically small, not exceeding 300 images. Recently, a large-scale visualization dataset, called the MASSVIS dataset [7], has been developed. The dataset contains more than 5,000 different types of static visualizations from four topics: government, infographic, news, and science.

Natural image datasets are also referenced to imitate gaze dots about chart images. For example, despite being collected on a different domain of images and not intended to be used for visualization images, the reaction to low-level features (e.g., color, contrast, motion, etc.) of an image in natural image datasets and chart images is similar on certain tasks (e.g., exploration tasks) [56]. Furthermore, since early large-scale gaze datasets are collected using natural image datasets as the stimulus, their results have often been deployed in predicting saliency in visualizations.

Collection Methodology. There are two main choices for collection methodology: lab experiment and crowdsourcing.

Lab experiments are optimal for collecting small-scale but high-quality eye movement data. As mentioned before, these lab experiments are mainly done using eyetracking hardware. The accuracy and quality of eyetracking dots are high as it deploys a specialized eyetracker and the experiment can be fully controlled. However, collecting gaze dots via lab experiments is expensive, impractical and time-consuming to involve a large number of participants.

Crowdsourced platforms such as Amazon Mechanical Turk (AMT) or Innocentive are beneficial in that it is easier to recruit at a relatively affordable price. Consequently, many researchers use crowdsourced platforms to obtain human-involved datasets in a large scale. However, existing technologies for tracking eye movements are not as accurate as real eyetracking hardware.

2.3 Our Approach

Based on our assessments, the most desirable approach is to develop a gaze prediction model that provides the highest accuracy, but at the same time using the least possible resources (i.e., reducing time and cost). Neural network models trained on large-scale natural image datasets achieves superior performance over other methods. We therefore aim to apply this approach to visualization in order to construct a gaze prediction model that we call a SCANNER DEEPLY.

To do this, we choose to gather *chart images* as our *image stimulus*, as we hypothesize this will lead to a more specialized model with better performance for chart input. For the *collection methodology*, we choose to use a *crowdsourced platform* to be able to collect sufficiently large number of data samples for training neural network models. To reduce the confusion of a model, we only collect eyetracking data for images that contain a single static chart.

We set a task that can provide answers on the generability of tasks. Matzen et al. [50], while describing the DVS, mention the possibility of a general-purpose gaze prediction model. However, several researchers provide empirical evidence that gaze patterns are task-specific. For example, Yarbus' [72] and Michal and Franconeri's [51] works on gaze research suggest that human attentions are guided by the task she is conducting. Prior work [50] showed that the ideal visualization has a strong overlap between the regions (1) that are most likely to draw the viewer's attention (bottom-up) and (2) the regions that convey important information (top-down) about a task. We choose a task that satisfies both conditions and can also be evaluated with a gaze dataset collected in a short time (7 seconds). Specifically, we ask participants to figure what type of chart it is after they view the chart for 7 seconds.

Another question remains—it is not clear how large the image stimulus dataset should be to yield a sufficient amount of eye movement data. To the best of our knowledge, the largest known visualization dataset is the MASSVIS dataset [7] which contains 5,000 static visualization images. However, the dataset is not suitable for our task as images in the infographics category have more than two charts per image, and the subset includes images far smaller than 5,000. For comparison, the Salicon dataset [29] has gaze annotations for 10,000 images, while they are natural images. It means the amount of images in MASSVIS is not sufficient for training neural networks. To that end, we decide to create our own image dataset that contains 10,000 images.

For the *tracker*, we choose a low-technology requirement consistent with our crowdsourced platform. Between cursor-based method and webcams, we choose to go with webcams. The reason is two-fold. First, webcams are the only method to collect gaze dots that are actually directly measured from one's eyes in a crowdsourced platform. The argument in this decision is not at all in manifesting the superiority of current webcam-based gaze collection over other methods. Our intention is just to reinforce that it is a choice worth studying because of the benefits we would gain if conducted successfully. Second, the use of webcams as the tracker, chart images as the stimulus on a crowdsourced platform is a choice that has not yet been investigated and verified in the literature. This may be because of the intrinsic issue that there may be noise in gaze dots collected from webcams and that consequently the result is unpredictable. However, if we can devise a method that can successfully ease the impact of noise and successfully train the model, then this can also lead to another contribution.

Given the gaze dataset obtained from crowdsourcing, we train SimpleNet, a neural network devised by Reddy et al. [58]. SimpleNet is a state-of-the-art saliency map generating model that utilizes convolutional neural network architectures (e.g., ResNets [23]), to generate gaze heatmap. It requires less computational resources to train or predict than existing networks designed for gaze prediction.

Table 2: **Example queries for collecting chart images from the web.** Each query (on the right) is a combination of two keywords from our topical analysis of papers (on the left) and a chart type (in the middle).

Topic Keywords	Chart	Search Query
culture, anomaly	bar	culture anomaly bar chart
normal, politics	line	politics normal line chart
artist, history	pie	history artist pie chart
global, media	tree	global media tree diagram
meeting, crime	heatmap	meeting crime heatmap
sports, outlier	spider	sports outlier spider map
newyork, time	bubble	newyork time bubble chart
market, medical	scatter	market medical scatterplot
museum, weather	violin	museum weather violin

3 TASK-SPECIFIC CROWDSOURCED DATA COLLECTION

Here, we delineate the steps taken to collect task-specific gaze dots using webcams on a crowdsourced platform. We first describe how we crawled chart images from the web. Then we describe the processes taken to conduct the crowdsourced study on Mechanical Turk.

3.1 Visualization Image Collection

The goal in collecting chart images is to collect a dataset that aligns with the distribution of charts that can be found in the web, so that the trained model becomes as versatile as possible. To collect crowdsourced eyetracking data at scale, we first needed a large-scale dataset of visualization images. While the large-scale MASSVIS dataset contains 5,000 images, this may not be sufficient for training a neural network. Furthermore, the dataset also contains images containing multiple individual charts. We base our scale on the Salicon dataset, which contains 10,000 gaze dots. Accordingly, we choose 10,000 images as our target quantity.

We gather images by querying Google Image[1] using three keywords each time, where two keywords are chosen from 379 keywords derived from the topical analysis [1] of papers in scientific communities over the past 10 years (2011–2020), and the last keyword stands for the type of charts (*e.g.*, line chart, bar chart, or heatmap). Examples of these keywords are shown in Table 2. Through a series of search queries, we gathered 280,000 images from the web. These images contain not only charts, but also natural images pertaining to the topic. We only kept the images that (1) have one chart in an image, (2) both width and height of an image are larger than 400 pixels, (3) have heights or widths less than 4 times of the other, and (4) are without texts whose sizes are unreadably small. We also removed all duplicates.

In the end, we were able to retrieve 10,960 chart images. To roughly identify the distribution of charts, we randomly sampled 1,000 of the images, and counted them by their chart types. Fig. 3 shows the 10 most frequent chart types in a random sample of 1,000 images in our

[1] `https://images.google.com`

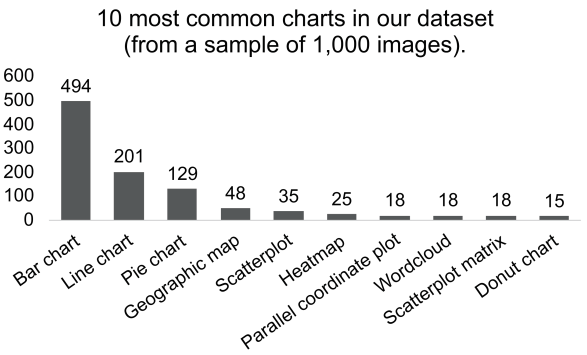

Fig. 3: **Chart type popularity.** We show the 10 most popular charts from a set of 1,000 samples randomly chosen from our image collection. This provides an estimate on the chart type distribution in our dataset.

collection. We can observe that bar charts, line charts, and pie charts are the three most commonly used graphs on the web.

3.2 Gaze Collection Setup

We collect eyetracking data using crowdsourcing on AMT. We design the procedure with three goals in mind: (1) obtaining high quality gaze dots, (2) obtaining large dataset, and (3) lessening the burden of participants. The task involved participants identifying the kind of chart after looking at the image for a few seconds; while they did so, we tracked their gaze using a webcam eyetracker.

Eyetracking Mechanism We created our experimental platform with the help of WebGazer, an eyetracking technique developed by Papoutsaki et al. [55]. Prior to finalizing our experiment environment, we conducted pilot studies with members of our research group at the University of Maryland to find the optimal settings. We identified two issues in deploying general-purpose webcams as eyetrackers: (1) the sensitivity and accuracy for webcam eyetrackers, and (2) maintaining a high level of concentration from participants.

Calibration in webcams is sensitive to slight movements of the head. Even state-of-the-art webcam eyetrackers exhibit poor robustness against changing head posture. This is exacerbated when testing in the wild, i.e., using a crowdsourcing platform on the internet. During their experiment with webcams, Xu et al. [71] note that eyetracking requires frequent calibration, and to minimize the number, the viewing time must be short. To alleviate this problem, we limit our eyetracked tasks to 7 seconds per image and conduct calibration every 6 images.

Secondly, to check if participants are concentrating, we add attention trials throughout the experiment to determine if participants are concentrating. We select 400 attention images. These attention images are all bar charts given the assumption that all participants know what a bar chart is, and hence can answer questions without difficulty. Furthermore, as Figure 3 shows, the bar chart is also the most common visual representation in our dataset. If the user does not correctly answer the attention trial, then we stop the participant from further proceeding with the experiment. This fact—that participants remain focused on the trials—is clearly communicated at the outset of the experiment.

Participants. We recruit participants who are fluent in English, do not have color blindness or any other kind of color vision deficiency, and are at least 18 years old. We let participants be aware that the experiment is about charts. Furthermore, we ask them to use computing device with a webcam, have monitors with a screen resolution or 1280×720 or higher, and use Google Chrome or Microsoft Edge web browsers (these requirements are imposed by our eyetracking software).

Apparatus. We build a website for collecting data while participants perform pre-defined tasks. While they are doing so, we track their eye movements. We use JavaScript and jQuery to build our website and use Python Flask v1.1.1 to run it on our university-hosted virtual machine (VM). We allocate 2 GB RAM and 32 GB storage to the VM.

3.3 Participant Task

Below we describe steps taken for collecting gaze dots from participants on a crowdsourced platform.

Getting Consent. Participants are first presented with a consent form, including a brief introduction to what the study is about and how the study is run. Afterwards, we lead participants to a website hosted on our server. The server was created by the authors and is run on a Linux-based virtualization environment provided by the department of Computer Science at the University of Maryland.

Calibration. Prior to conducting the experiment we provide several suggestions to participants to help them get past the accuracy threshold during the calibration phase: (1) conducting the experiment in a well-lit environment, (2) situating their heads within the green box shown on the camera view, and (3) trying not to change their head posture during the experiment, as it will decrease the calibration accuracy.

After participants allow access to the camera, the calibration can proceed. Fig. 2 (A) illustrates the steps required for calibration. Calibration takes place in the following manner: Participants are asked to hold their head steady and place their head on the box shown in the camera view. Then, they are asked to synchronously look at and click at 8 dots placed near the margins of the browser five times each. The gaze dot, or the estimated gaze location of the participant, is represented by a small moving red-colored dot. Every time a target button is clicked, the opacity of it gets lower until it eventually turns into yellow to signal that it has been fully clicked 5 times. These clicks adjust the calibration between the participant's eyes and the browser window. After all dots turn yellow, a new red-colored button shows up at the center of the browser to measure the accuracy of the calibration. The participants are required to look at the dot until it turns yellow. Accuracy is measured by the proportion of gaze dots that are placed near the center and those that are not. As described in the previous part, if that proportion is higher than 70%, then participants can proceed to the gaze collection phase. If the accuracy is below 70%, then participants must repeat the process until the accuracy rises above the threshold.

Data Collection. The next step is where the gaze log collection process starts. Fig. 2 (B) explains the steps taken to collect gaze logs. Once participants click the button "Go" from the question "Are you ready?," an image appears at the center of the screen. The image appears for 7 seconds. During the 7-second period, gaze logs are collected at a refresh rate of 20 Hz. After 7 seconds have passed, the screen changes into a 20-multiple choice question that polls the chart type. The participants can choose to answer between the choices, select "I don't know," or provide a new answer after choosing "other."

As stated before, we compose one unit block as one calibration followed by 6 gaze trials. The intention is to keep one human intelligence task (HIT) under 3 minutes, including the calibration. From our pilot study, collecting gaze logs from a full block averaged around 1 minute and 10 seconds. Among the 6 trials in a block, one is an attention trial and 5 are used for the experiment. After each HIT is complete, we collect a packet from the server that contains information on (1) the gaze log information, (2) width, height, and size of the browser (e.g., the exact displacement of the image, the size of the screen, etc.), (3) answers to the questions asked, and (4) the name of the image. We do not capture nor reference videos from the webcam; the latter is clearly stated to participants prior to participating in the experiment.

One HIT is composed of three blocks. Based on our pilot study, it took approximately 3 minutes and 33 seconds to complete one HIT. Based on the targeted rate of $15 per hour upon successful completion of the experiment, we pay each participant $0.90 per HIT. For those that perform more than one HIT, we compensate them in the form of bonuses, at the rate of $0.30 per block. We gave compensation to any participants that conducted the experiment even if only partially. We paid participants no more than five days after the date of submission.

3.4 Collection Results

For the 10,960 images, we ran this collection process until we had at least one annotation per image, resulting in 12,504 gaze dots. It took 60 days to collect the data. We exclude the gaze dots from participants

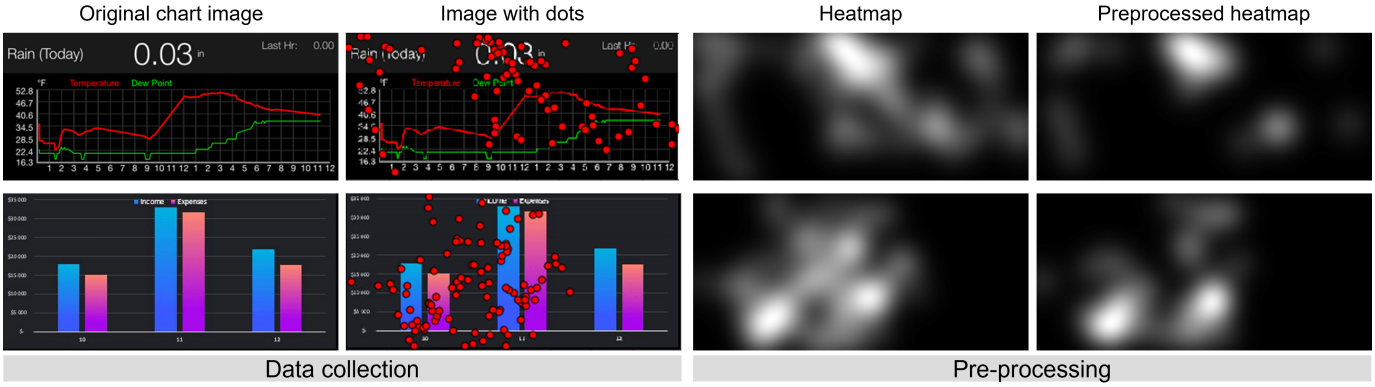

Fig. 4: **Illustration of procedures for constructing oracle heatmaps.** We use the final heatmaps for training our neural networks. Gaze dots collected by using webcams contain rich information that enables further analysis, e.g., temporal changes in human visual perception (see §5.4 for the analyses), but they are also noisy to learn. We address this challenge by carefully pre-processing gaze dots. The second column shows the dots in the raw data from webcams, and we blur dots and remove some of them not in the human's area of focus (see §4.2 for details).

who submitted incorrect identification numbers. The dataset consists of gaze logs from 9,157 correct responses and 3,347 incorrect responses. Examples of gaze logs collected from our study are shown in Fig. 4. The success rate of each HIT task was 71.2%. The unsuccessful HITs are due to (1) incompleted tasks (21.3%) and (2) false answers (7.5%).

4 A Scanner Deeply

We now propose a Scanner Deeply, a virtual eyetracker that utilizes a deep neural network (DNN) for gaze prediction. Scanner Deeply takes a visualization image as an input and automatically generates a gaze heatmap for the image. We first discuss our choice of a DNN: we employ SimpleNet [58], a compact DNN architecture designed for gaze prediction. We then describe how we preprocess our dataset to prepare the training data, i.e., data filtration and saliency map generation. We finally describe how we train SimpleNet on the preprocessed data.

4.1 SimpleNet: A DNN for Gaze Prediction

We have two criteria for choosing a DNN architecture. First, the architecture should be fast and computationally efficient at inference time. Most DNN architectures that are used in prior work [39, 69] for gaze prediction are complex, i.e., they contain millions of model parameters, which increases the operational costs. To run inferences with those models, we require special hardware (e.g., GPUs or hardware accelerators). Reducing a DNN's post-training operations thus allows users and practitioners to deploy the Scanner Deeply to diverse computing environments, ranging from servers and personal computers to devices with limited computational resources, e.g., IoT or mobile devices. Second, while reducing the costs of post-training operations, we choose the architecture that provides state-of-the-art performance in gaze prediction tasks. Deep and complex architectures [45] typically offer better performance. However, we aim to find a shallow, simpler network that can achieve on-par performance. Considering the criteria, we employ SimpleNet to implement the Scanner Deeply.

Figure 5 illustrates the SimpleNet architecture adapted for our pipeline. SimpleNet employs an encoder-decoder architecture. The encoder extracts latent representations (often referred to as features) from an input, and the decoder reconstructs the input from the latent vectors. In the figure, SimpleNet extracts a 2048-dimensional vector from a visualization image as a latent representation. From this vector, the decoder architecture generates a gaze map. Multiple image classification models, such as VGGNet [65], ResNet [23], and PNASNet [46], can be used as an encoder; we choose ResNet because it offers the highest performance. The decoder is composed of two deconvolutional layers. SimpleNet utilizes the U-Net structure [62], which helps to improve the performance further by incorporating the information from earlier layers when the decoder reconstructs a gaze map.

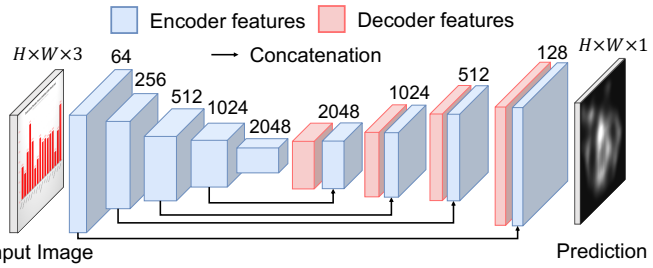

Fig. 5: **Our SimpleNet architecture.** The pipeline accepts image input and produces an output image. In our case, the input is a visualization and the output is the gaze map. SimpleNet follows an encoder-decoder structure similar to U-Net [62]. Compared to DNNs from prior work [29], SimpleNet requires less computation.

4.2 Preprocessing Webcam-based Eyetracking Data

Even with our efforts to maintain a high level of calibration during gaze collection process, the resulting gaze dots from webcam-based eyetrackers contain noise. Noise is mainly attributed to the dispersion of gaze dots. Such dispersion happens as participants make diminutive movements of their heads, or because poor or shifting light conditions introduce errors in the eyetracking software. At worst, noisy ones exist in locations far from where the chart is located within an image, and they prevent the DNNs from learning patterns in the data. (see Table 3)

To address this issue, we filter out gaze dots not in a *saliency zone*. Following the prior work [22, 31, 60], we define the saliency zone as areas in an image that are highly disparate in RGB values in contrast to its neighboring pixels. We preprocess gaze dots as follows: At each pixel location, we first compute color differences between the location and all the neighboring pixels within a 5-pixel radius. We perform this step over all the pixels in an image and remove gaze dots at each pixel location where the sum of the color differences is less than 10 (10 is the threshold we empirically find). We then blur the fixations by using a Gaussian filter that has a standard deviation of 5 and set the boundaries of the salient zone. In Fig. 4, we show examples of our preprocessed gaze dots. We exclude 160 images that this filtration process removes all the gaze dots from our dataset, leading to 12,344 gaze maps, consisting of 9,040 correct and 3,304 are incorrect ones. Note that our preprocessing does not change the input resolution.

4.3 Training SimpleNet on Our Crowdsourced Dataset

We implement Scanner Deeply with Python 3.9 and PyTorch v1.10.[2] To train our SimpleNet models, we use a machine equipped with Intel i7-6700K CPU with 32GB RAM and 2 NVIDIA GTX 1080 Ti GPUs. Note that, once trained, we only use CPUs to offer gaze prediction.

[2] `https://pytorch.org`

Datasets. We compose our dataset as follows: Among the 9,040 correctly-responded visualization images, we randomly pick 70% of them as our training data and use the rest 30% as the testing set. This split leads to 6,328 training and 2,712 testing images. Next, we pair each visualization image with a gaze map preprocessed by the method described in §4.2. Those maps are the oracles the model should generate. All the visualization images and saliency maps are scaled to 256×256. Our model produces gaze predictions with the same size, and we re-scale them back to the original resolutions for visualization.

Objective Function. We train SimpleNet to minimize the perceptual difference between gaze predictions and the oracle saliency maps. To measure the difference, we employ the Kullback-Leibler (KL) divergence, an objective function commonly-used in literature [17]. We also examine other metrics for measuring perceptual differences, e.g., ℓ_p-distances or normalized scanpath saliency (NSS) [42], proposed by prior work [61]. While our model minimizes those metrics, we observe in our manual analysis that gaze predictions generated with KL divergence are better than the cases of using others.

Hyper-parameters. We use the Adam [36] optimizer to train our models. We set the batch size to 32 and the learning rate to 10^{-4}. We also set the weight decay to 10^{-4}. We train our models for 40 epochs; at each epoch, we compute the KL divergence of our model on the testing set and store the one that minimizes the metric over the 40.

5 Evaluation

Here, we evaluate the Scanner Deeply. We first show the prediction performance of the Scanner Deeply and the baselines (§5.1). We then compare the gaze predictions qualitatively to illustrate unique characteristics the Scanner Deeply captures (§5.2 and §5.3). We lastly show the benefit of using webcam-based eyetackers by analyzing temporal changes in human visual perception with the Scanner Deeply (§5.4).

5.1 Prediction Performance of the Scanner Deeply

We evaluate the performance of the Scanner Deeply. The purpose of this experiment is to show that for predicting gaze heatmaps for a specific task, it is desirable to train models on a task-specific dataset. To this end, we train (1) SimpleNet on Salicon [29], gaze maps for a set of natural images and (2) use the DVS model [50][3] designed to perform generally well on predicting gaze maps for visualizations. We employ five metrics for measuring perceptual similarity between gaze predictions and gaze maps on our 2,712 testing images: two location-based metrics, i.e., AUC-Judd (AUC-J) [30] and NSS, and three similarity-based metrics, i.e., KL-Div, SIM, and CC [9]. Except for KL-Div, the higher a metric is, the more gaze predictions are similar to oracles. Table 3 shows our results.

Scanner Deeply shows better performance over the two baselines. We first observe that the Scanner Deeply exhibits 5–15% higher performance than SimpleNet trained on the Salicon dataset. Considering that the training dataset is the only difference between the two models, it is important to use the dataset collected from the same domain (i.e., visualization) for high-quality gaze prediction. Compared with the DVS model, we find that the Scanner Deeply achieves 8–18% improvements. This result implies that within the same domain, a model trained on a task-dependent dataset can perform better on a specific task than a model built for a general-purpose prediction.

We also assess the impact of the preprocessing step on the performance of the Scanner Deeply (Ours vs. Ours†). Across the board, the similarity metrics from the models trained on preprocessed gaze maps are better than those trained on un-preprocessed maps. This confirms our hypothesis that filtering out gaze dots in the image area with no stimulus helps a model focus more on the important low-level features. We find that the improvements are larger for the location-based similarity metrics (i.e., AUC-J and NSS), which implies that preprocessing encourages a model to ignore the unimportant parts in visualization.

[3] https://github.com/LauraMatzen/DVS

Table 3: **Performance evaluation of Scanner Deeply by comparing it with gaze prediction models proposed by prior work.** We show the performance using five metrics. We report each metric's mean and standard deviation over the five runs. The value shown on top is the mean, and on the bracket below is the standard deviation over the five runs. We compare with DVS [50] and Salicon [29]. The only difference between Salicon and the Scanner Deeply is in the dataset. We show that a model using datasets collected for the specific task and domain on a specific task can show improved performance over a general-purpose gaze prediction model. We also compare ours with Scanner Deeply before pre-processing. This is represented as Ours†.

Metric		Ours	Ours†	DVS	Salicon
Location-based	(↑) **AUC-J**	0.784 (0.002)	0.751 (0.002)	0.737 (0.005)	0.741 (0.002)
	(↑) **NSS**	0.716 (0.004)	0.625 (0.006)	0.489 (0.005)	0.537 (0.009)
Distribution-based	(↓) **KL-Div.**	1.283 (0.035)	1.333 (0.053)	1.507 (0.022)	1.443 (0.063)
	(↑) **CC**	0.403 (0.166)	0.373 (0.186)	0.276 (0.136)	0.301 (0.193)
	(↑) **SIM**	0.429 (0.002)	0.388 (0.014)	0.373 (0.012)	0.392 (0.003)

† SimpleNet trained on the dataset w/o the preprocessing step in §4.2.

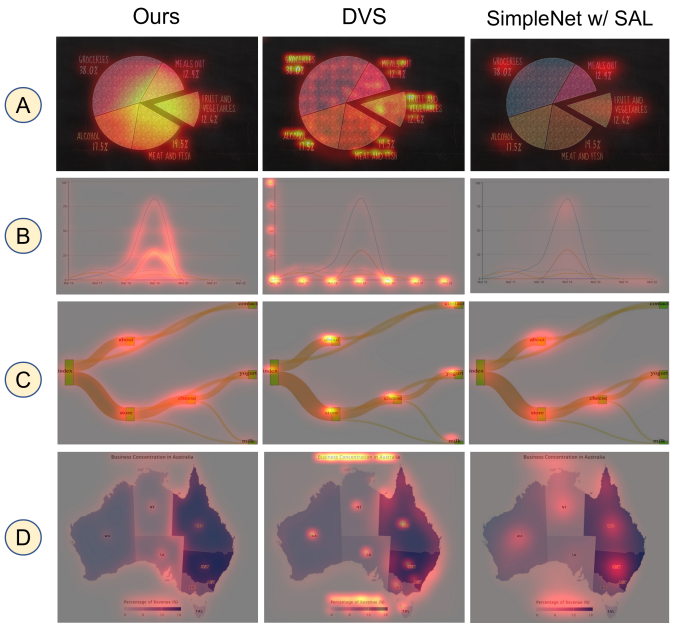

Fig. 6: **Visualization of gaze predictions.** We show the gaze predictions of three charts from ScannerDeeply, DVS, and SimpleNet trained on the Salicon dataset. DVS and Salicon focus more on text, while our model focuses more on distinct areas in a chart.

5.2 How Does the Scanner Deeply Perceive Charts?

We now present unique characteristics of how the Scanner Deeply perceives visualization images compared with our baselines. We make this comparison by analyzing how those three models perceive three different stimuli within an image: (1) charts (visualized data information), (2) textual information, and (3) low-level features [56]. In Fig. 6, we illustrate gaze predictions generated by three models for four images.

Chart Perception. A predominant characteristic of gaze predictions produced by the Scanner Deeply is that it focuses on the shape (or structures) of charts within images, while the baseline models focus more on the texts. Fig. 6 shows that gaze predictions from DVS and SimpleNet models, trained with Salicon, highlight legends, numbers, or titles. In contrast, the Scanner Deeply focuses on a slice in the pie chart (A), peaks in the line chart (B), branching points in the sankey tree (C), and edges in the geographical maps (D). This means that the

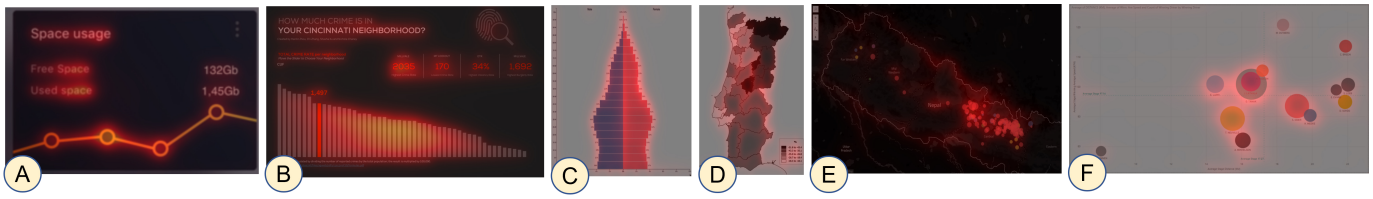

Fig. 7: **Gaze predictions from ScannerDeeply for different types of visualizations.** We show examples of the six common chart types in our datasets, *i.e.*, (from the left) a line chart, a bar chart, a population pyramid, a heatmap on a geographic map, a scatterplot on a geographic map, and a scatterplot. We analyze them in §5.3.

Scanner Deeply has the ability to emphasize different parts of images depending on the visualization tasks. The task here is to determine the type of a chart. It also means that in our data collection, participants have looked at the entire chart to perform the task, not focusing only on a specific part (see in §5.3 and 5.4 for our detailed analysis).

If we take a closer look at the gaze predictions from the DVS model (see the figures in the middle column), the model explicitly captures texts the most. The result implies that DVS is designed to capture both the low-level features (via the GBVS model) and texts (via text optimizers), but in fact, the text optimizers seem to be a dominant component. Surprisingly, from the figures in (B), DVS predicts that humans will mainly focus on reading the x-/y-axis. However, it is unlikely to be true as we use charts for reducing the perceptual complexity.

We further show that the model learned from natural image datasets (e.g., Salicon) focuses on low-level features the most. If the shape of the chart is easily noticeable with respect to its environment, the model focuses on the chart (see the figures in rows (A) and (C)); otherwise, it emphasizes other areas, e.g., texts. Here, being noticeable means how much contrast the chart is in terms of its color or shapes compared to the other parts of an image, which accounts for human perception of natural images. However, our results suggest that it might not be suitable for understanding human perception for a specific task.

Text Perception. Our findings are in stark contrast to the prior work [49] showing humans mostly focus on textual information in a visualization. In Scanner Deeply, the concentration of gaze on text is somewhat similar to, or sometimes even weaker than that on the shape of the chart. We hypothesize that the importance of textual information in visualization may differ by the designated task. If the task is to read visualization closely and comprehensively, textural information will carry a lot of importance. But, the overall structure of visualization becomes more critical if the task is to classify visualization types. This finding is in line with the results presented by Polatsek et al. [56].

Oftentimes, the model trained on the SALICON dataset also produces gaze predictions that focus on textual information. However, we argue that it is because the saliency is largely determined by the level of contrast in the low-level features (e.g., colors) of the text with respect to the chart background. Note that we typically use distinct colors in texts for perceptual clarity. Thus, neural networks—that are good at capturing such input differences—will learn them during training and reflect the differences in the gaze predictions.

Perception of Low-level Features. Charts in images are generally drawn so that they can stand out from the backgrounds visually. As we discussed above, this is the main reason that all three models capture low-level features. But, there is a noticeable difference between the Scanner Deeply and the baseline models. Scanner Deeply utilizes different levels of prioritization in capturing low-level features.

Suppose our model only emphasizes the low-level features that are similar to the baselines (see gaze predictions conducted by SimpleNet with Salicon dataset in Fig. 6). In (A), the prediction should focus on every component in the image with white color (as it has the highest contrast to the black background). However, our model focuses on a specific slide in the pie charts. It does not mean that we de-prioritize low-level features; the Scanner Deeply does prioritize low-level features when it is needed. In (C), humans need to focus on the texts at branching points to understand the sankey tree chart. In this case, our model emphasizes the texts in black, which contrasts with the yellow-colored background the most.

5.3 What Structure Does the Scanner Deeply Focus?

We now analyze further the gaze predictions generated by the Scanner Deeply. We specifically focus on the structures (or shapes) our model primarily focuses on visualization images. We present examples for five most popular chart types: line chart, bar chart, pie chart, geographics map, and scatterplot. We show those examples in Fig. 7.

Bar Charts. For bar charts in Fig. 7 (B), the model mainly focuses on the bars of the chart, though the gaze is focused on the center of the chart. This is not an unusual phenomenon, as chart images are generally located at the center of the image. Variants of bar charts also follow a similar pattern. For example, in population pyramids (see Fig. 7 (C)), the gaze covers the entire shape of the chart.

Pie Charts. In Fig. 6 (A), the gaze prediction is on the center of the chart. It checks the angles of each pie and looks at the boundaries. Most pie charts have labels next to the pie, attracting the gaze. However, the concentration of gaze is usually less than that of the chart.

Line Charts. The model reads the line that passes through the chart for line charts. When the color of the line stands out from its environment, the saliency follows the line until it ends (see Fig. 6 (B)). When there are multiple lines in the chart, it follows all of these lines. However, if the line is similar to the background color, then it often fails to follow the line. For example, when a line is drawn on a grid plane with similar color, it confuses distinguishing between the grid and the line, and it follows both the grid and the line.

Geographic Maps. We have two types of geographic maps: (1) maps containing textual information and (2) superimposed visual components. In the first one (in Fig. 7 (D)), the gaze prediction focuses on the boundaries and texts with large font. In the second one (in Fig. 7 (E)), the prediction focuses on parts of the map with more visual components than parts with fewer components, e.g., if there are dots on the map, the gaze is focused on where the concentration is high.

Scatterplots. Generally, in scatterplots (see Fig. 7 (F)), our model focuses on each dot, with the highest concentration of interest in areas with the highest concentration of dots.

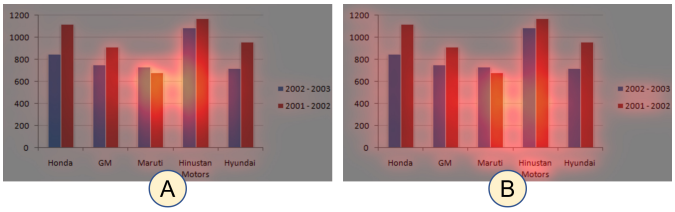

Fig. 8: **Temporal eye movements in gaze predictions.** By training SimpleNet on datasets representing different time scale, ScannerDeeply can predict human visual perception over time. In (A), we train our model on the gaze maps collected from the first 0.7 seconds, while we train our models on the gaze maps for the entire 7 seconds. Our model predicts that human eyes will focus first on the distinct areas in charts and then focus on text components (e.g., legends) later on.

5.4 Analyzing Temporal Patterns in Human Perception

Our webcam-based data collection offers unique information over other datasets: we collected eye movements *over time*. We recorded eye movements every 0.05 seconds over 7 seconds in total, which allows

us to analyze temporal patterns in human visual perception. Using this advantage, we evaluate an intriguing hypothesis: It has been known that humans first scan the overview of a visualization image and then perceive its details. To this end, we train our neural network on different datasets containing 0–t seconds, where $t \in [1, 7]$.

Fig. 8 illustrates the gaze predictions for an image from our model trained with 0.7s period data (left) and the model trained with the full 7s data (right). We first observe that the Scanner Deeply, trained on the 0.7s data, looks at the center of the charts. We found two reasons: (1) We made participants reset their eyes at the center; thus, they started by looking at the center. (2) We also see that to understand the chart type (i.e., bar), participants do not need to look at other information.

However, when trained with the entire data, the Scanner Deeply starts focusing on other details, such as legends or texts in the x-axis where the highest bars are. This confirms our (and also a hypothesis well-known to the visualization community) that human perceives an overview first, and then details come. Our findings also raise questions about the community's practice in gaze predictions. Most prior work [29, 50] offers a single gaze map for each image, which may lead to missing opportunities to understand human perception in more depth.

6 Discussion

In this section, we discuss the implications of our work in two ways: (1) an assessment of our data collection methodology, (2) two possible approaches one can consider for gaze estimation. Finally, we conclude by discussing the limitations and plans for our future work.

6.1 Assessing Crowdsourced, Webcam-based Gaze

We collected a large-scale dataset of gaze fixation points on visualization images using a crowdsourced platform through general-purpose webcams. There is clearly a benefit in terms of time and cost when using a crowdsourced platform than when conducting a lab study. We spent approximately $900 to collect around 12,000 gaze logs. This yielded approximately $0.07 to $0.08 per gaze log. This is about 40% more expensive than cost expectations using TurkEyes [52]. However, one must take into account that our payment was based on a targeted rate of $15 per hour, whereas that of TurkEyes was $10.

There was an obstacle during the experiment—the time it takes to collect sufficient amount of data. While our initial expectation of collecting period was 30 days, it took more than 50 days to collect 10,000+ gaze logs. We attribute this to the fact that the task of collecting eye movement data on webcams is a challenge when using general-purpose webcams. Participants are required to conduct frequent calibrations, potentially multiple times, if they did not reach a particular threshold. We struggled to collect large-scale data at a fast, constant pace despite compensating at the relatively high rate of $15 per hour.

6.2 A Priori or A Posteriori? A Better Method for Gaze Estimation

Our results show that gaze patterns on visualizations are task-specific and domain-specific. This goes in line with the experiment conducted by Polatsek et al. [56]. In developing a gaze prediction model, they propose a gaze prediction model based on an a *priori* approach where the model is made as a combination of an image- and an object-based saliency model, similar to the DVS model that combines a model for natural images with a text identifier. In contrast, our work takes a *posteriori* approach that trains a neural network on empirical gaze data. Here, we discuss advantages and disadvantages for both approaches.

Compared to a prior approach, our data-driven, a posteriori approach requires a large-scale dataset and substantial computing power to train a sophisticated neural network model. However, there is no guarantee that a priori-based model will provide solid performance in the general case. For simple low-level tasks, a priori-based models can predict gaze views as well as a data-driven a posteriori model without difficulty. Issues will arise when dealing with tasks that require a priori knowledge about human perception where no such knowledge exists. This may occur when the model encounters an entirely new type of chart not covered by existing perceptual models. It also becomes a problem when balancing between known factors is required. For example, while conventional knowledge suggests that people focus on textual information in a chart, the overview task in our experiment yielded more focus on chart shape than text. For both of these examples, a data-driven a posteriori model uses sheer computing power to learn from the large-scale dataset.

On the other hand, an a priori model for gaze prediction is built on operational and interpretable perceptual knowledge. Such knowledge can be informed by existing and ongoing research in perceptual psychology [21, 59]. The neural network implementing our a posteriori model, in contrast, is opaque and not understandable to humans [20]. Furthermore, new training datasets and models may be needed to cover all conceivable tasks, datasets, charts, and domains to guarantee robust performance for the deep learning method. Nevertheless, one important question we raise for future work is whether neural networks, such as our Scanner Deeply model presented in this paper, can teach us anything about perceptual psychology in humans.

6.3 Limitations and Future Work

As shown in our qualitative and quantitative experiments, gaze prediction models that are task-specific and domain-specific are desirable. Even if several works (e.g., Yarbus' [72] and Michal and Franconeri's [51]) suggest that the task guides human attention she is conducting, most of the recent work (shown in §2) on predicting gaze heatmaps focuses on collecting and developing models generally working well. That being said, building such models for individual tasks may require iterative efforts. However, we highlight that our work presented a method and desirable choices for the crowdsourced data collection approach in conjunction with the low-technology requirement.

We emphasize that our paper's datasets and methodology shed some light on interesting future work directions. As gaze prediction on visualization is an active area of study, we first envision extending our approach for various tasks as future work. We also want to highlight that the scalability issues in building prediction models for different tasks may inspire interesting future work. For example, one could build foundational models for gaze predictions, similar to models developed in computer vision (CLIP [57] or GPT-3 [4]), and then fine-tune those models for multiple downstream tasks. We further envision that exploring the feasibility of few-shot learning in this context to tackle data scarcity issues could be promising for future work.

Separately, in contrast to the datasets presented by the prior work, our dataset contains gaze dots collected over 7 seconds, which may reflect temporal human eye movements. We further envision future work using our dataset to conduct in-depth analysis of temporal eye movements to understand human perception better.

Homage

> Once a guy stood all day shaking bugs from his hair. The doctor told him there were no bugs in his hair. After he had taken a shower for eight hours, standing under hot water hour after hour suffering the pain of the bugs, he got out and dried himself, and he still had bugs in his hair; in fact, he had bugs all over him. A month later he had bugs in his lungs.
>
> – *A Scanner Darkly* (1977), Philip K. Dick (1928–1982)

Data Availability

Our crowdsourced dataset and source code for reproducing all of our experiments can be found in `https://osf.io/spw49/?view_only=7fde2fc5e51f4d6682805c1dbb7420f6`.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. This work was partially supported by the U.S. National Science Foundation award IIS-1908605. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the funding agency.

REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003. doi: 10.5555/944919.944937

[2] C. E. Bonhage, J. L. Mueller, A. D. Friederici, and C. J. Fiebach. Combined eye tracking and fMRI reveals neural basis of linguistic predictions during sentence comprehension. *Cortex*, 68:33–47, 2015. doi: 10.1016/j.cortex.2015.04.011

[3] M. A. Borkin, Z. Bylinskii, N. W. Kim, C. M. Bainbridge, C. S. Yeh, D. Borkin, H. Pfister, and A. Oliva. Beyond memorability: Visualization recognition and recall. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):519–528, 2016. doi: 10.1109/TVCG.2015.2467732

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Proceedings of the Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901. Curran Associates, Inc., 2020.

[5] N. Bruce and J. Tsotsos. Saliency based on information maximization. In *Proceedings of the Advances in Neural Information Processing Systems*, vol. 18. MIT Press, 2005.

[6] M. Burch, N. Konevtsova, J. Heinrich, M. Hoeferlin, and D. Weiskopf. Evaluation of traditional, orthogonal, and radial tree diagrams by an eye tracking study. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2440–2448, 2011. doi: 10.1109/TVCG.2011.193

[7] Z. Bylinskii, M. A. Borkin, N. W. Kim, H. Pfister, and A. Oliva. Eye fixation metrics for large scale evaluation and comparison of information visualizations. In M. Burch, L. Chuang, B. Fisher, A. Schmidt, and D. Weiskopf, eds., *Proceedings of the Workshop on Eye Tracking and Visualization*, pp. 235–255. Springer, Cham, 2017. doi: 10.1007/978-3-319-47024-5_14

[8] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba. MIT saliency benchmark. http://saliency.mit.edu/.

[9] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand. What do different evaluation metrics tell us about saliency models? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(3):740–757, 2019. doi: 10.1109/TPAMI.2018.2815601

[10] Z. Bylinskii, N. W. Kim, P. O'Donovan, S. Alsheikh, S. Madan, H. Pfister, F. Durand, B. Russell, and A. Hertzmann. Learning visual importance for graphic designs and data visualizations. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, p. 57–69. ACM, New York, NY, USA, 2017. doi: 10.1145/3126594.3126653

[11] S. Cheng, J. Fan, and Y. Hu. Visual saliency model based on crowdsourcing eye tracking data and its application in visual design. *Personal and Ubiquitous Computing*, pp. 1–18, 2020. doi: 10.1007/s00779-020-01463-7

[12] K. Conklin and A. Pellicer-Sánchez. Using eye-tracking in applied linguistics and second language research. *Second Language Research*, 32(3):453–467, 2016. doi: 10.1177/0267658316637401

[13] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. SAM: Pushing the limits of saliency prediction models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1971–19712. IEEE, Piscataway, NJ, USA, 2018. doi: 10.1109/CVPRW.2018.00250

[14] S. K. D'Mello, C. Mills, R. Bixler, and N. Bosch. Zone out no more: Mitigating mind wandering during computerized reading. In *Proceedings of the International Conference on Educational Data Mining*. ERIC, 2017.

[15] C. Fosco, V. Casser, A. K. Bedi, P. O'Donovan, A. Hertzmann, and Z. Bylinskii. Predicting visual importance across graphic design types. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, p. 249–260. ACM, New York, NY, USA, 2020. doi: 10.1145/3379337.3415825

[16] M. Gingerich and C. Conati. Constructing models of user and task characteristics from eye gaze data for user-adaptive information highlighting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1728–1734. AAAI Press, Palo Alto, CA, USA, 2015.

[17] J. Goldberger, S. Gordon, and H. Greenspan. An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 487–493 vol.1, 2003. doi: 10.1109/ICCV.2003.1238387

[18] S. R. Gomez, R. Jianu, R. Cabeen, H. Guo, and D. H. Laidlaw. Fauxvea: Crowdsourcing gaze location estimates for visualization analysis tasks. *IEEE Transactions on Visualization and Computer Graphics*, 23(2):1042–1055, 2017. doi: 10.1109/TVCG.2016.2532331

[19] M. J. Haass, A. T. Wilson, L. E. Matzen, and K. M. Divis. Modeling human comprehension of data visualizations. In *Proceedings of the International Conference on Virtual, Augmented and Mixed Reality*, pp. 125–134. Springer, Cham, 2016. doi: 10.1007/978-3-319-39907-2_12

[20] D. Haehn, J. Tompkin, and H. Pfister. Evaluating "graphical perception" with CNNs. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):641–650, 2018. doi: 10.1109/TVCG.2018.2865138

[21] T. Hamner and G. Vivanti. Eye-tracking research in autism spectrum disorder: What are we measuring and for what purposes? *Current Developmental Disorders Reports*, 6(2):37–44, 2019. doi: 10.1007/s40474-019-00158-w

[22] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Proceedings of the Advances in Neural Information Processing Systems*, pp. 545–552. MIT Press, Cambridge, MA, USA, 2006.

[23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778. IEEE, Piscataway, NJ, USA, 2016. doi: 10.1109/CVPR.2016.90

[24] J. Huang, R. White, and G. Buscher. User see, user point: Gaze and cursor alignment in web search. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 1341–1350. ACM, New York, NY, USA, 2012. doi: 10.1145/2207676.2208591

[25] W. Huang. Using eye tracking to investigate graph layout effects. In *Proceedings of the International Asia-Pacific Symposium on Visualization*, pp. 97–100. IEEE, Piscataway, NJ, USA, Feb 2007. doi: 10.1109/APVIS.2007.329282

[26] 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*, 20(11):1254–1259, 1998. doi: 10.1109/34.730558

[27] R. J. Jacob and K. S. Karn. Commentary on section 4 - eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In *The Mind's Eye*, pp. 573–605. North-Holland, Amsterdam, 2003.

[28] N. Jaques, C. Conati, J. M. Harley, and R. Azevedo. Predicting affect from gaze data during interaction with an intelligent tutoring system. In *Intelligent Tutoring Systems*, pp. 29–38. Springer International Publishing, Cham, 2014. doi: 10.1007/978-3-319-07221-0_4

[29] M. Jiang, S. Huang, J. Duan, and Q. Zhao. SALICON: Saliency in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1072–1080. IEEE, Piscataway, NJ, USA, 2015. doi: 10.1109/CVPR.2015.7298710

[30] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2106–2113. IEEE, Piscataway, NJ, USA, 2009. doi: 10.1109/ICCV.2009.5459462

[31] T. Kadir and M. Brady. Scale saliency: a novel approach to salient feature and scale selection. In *Proceedings of the International Conference on Visual Information Engineering*, pp. 25–28, 2003. doi: 10.1049/cp:20030478

[32] S. Karthikeyan, Thuyen Ngo, M. Eckstein, and B. S. Manjunath. Eye tracking assisted extraction of attentionally important objects from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3241–3250. IEEE, Piscataway, NJ, USA, 2015. doi: 10.1109/CVPR.2015.7298944

[33] N. W. Kim, Z. Bylinskii, M. A. Borkin, K. Z. Gajos, A. Oliva, F. Durand, and H. Pfister. BubbleView: An interface for crowdsourcing image importance maps and tracking visual attention. *ACM Transactions on Computer-Human Interaction*, 24(5), 2017. doi: 10.1145/3131275

[34] N. W. Kim, Z. Bylinskii, M. A. Borkin, A. Oliva, K. Z. Gajos, and H. Pfister. A crowdsourced alternative to eye-tracking for visualization understanding. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems*, p. 1349–1354. ACM, New York, NY, USA, 2015. doi: 10.1145/2702613.2732934

[35] S. Kim, Z. Dong, H. Xian, B. Upatising, and J. S. Yi. Does an eye tracker tell the truth about visualizations?: Findings while investigating visualizations for decision making. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2421–2430, 2012. doi: 10.1109/TVCG.2012.215

[36] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization.

arXiv preprint arXiv:1412.6980, 2014.

[37] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. Eye tracking for everyone. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2176–2184. IEEE, Piscataway, NJ, USA, 2016. doi: 10.1109/CVPR.2016.239

[38] M. Kümmerer, L. Theis, and M. Bethge. Deep Gaze I: Boosting saliency prediction with feature maps trained on ImageNet. In *Proceedings of the International Conference on Learning Representations*, 2015.

[39] M. Kümmerer, T. S. A. Wallis, L. A. Gatys, and M. Bethge. Understanding low- and high-level contributions to fixation prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4799–4808. IEEE, Piscataway, NJ, USA, 2017. doi: 10.1109/ICCV.2017.513

[40] S. Lallé, C. Conati, and G. Carenini. Predicting confusion in information visualization from eye tracking and interaction data. In *Proceedings of the International Joint Conference on Artificial Intelligence*, p. 2529–2535. AAAI Press, Palo Alto, CA, USA, 2016.

[41] S. Lalle, T. Wu, and C. Conati. Gaze-driven links for magazine style narrative visualizations. In *Short Proceedings of the IEEE Visualization Conference*, pp. 166–170. IEEE, Piscataway, NJ, USA, 2020. doi: 10.1109/VIS47514.2020.00040

[42] O. Le Meur and T. Baccino. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior Research Methods*, 45(1):251–266, 2013. doi: 10.3758/s13428-012-0226-9

[43] S. Lee, S.-H. Kim, Y.-H. Hung, H. Lam, Y.-A. Kang, and J. S. Yi. How do people make sense of unfamiliar visualizations?: A grounded model of novice's information visualization sensemaking. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):499–508, 2016. doi: 10.1109/TVCG.2015.2467195

[44] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pp. 740–755. Springer International Publishing, Cham, 2014. doi: 10.1007/978-3-319-10602-1_48

[45] A. Linardos, M. Kümmerer, O. Press, and M. Bethge. DeepGaze IIE: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12919–12928. IEEE, Piscataway, NJ, USA, 2021. doi: 10.1109/ICCV48922.2021.01268

[46] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision*, pp. 19–35. Springer International Publishing, Cham, 2018. doi: 10.1007/978-3-030-01246-5_2

[47] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Inferring human gaze from appearance via adaptive linear regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 153–160. IEEE, Piscataway, NJ, USA, 2011. doi: 10.1109/ICCV.2011.6126237

[48] P. Majaranta and A. Bulling. *Eye Tracking and Eye-Based Human–Computer Interaction*, pp. 39–65. Springer London, London, 2014.

[49] L. E. Matzen, M. J. Haass, K. M. Divis, and M. C. Stites. Patterns of attention: How data visualizations are read. In *Augmented Cognition. Neurocognition and Machine Learning*, pp. 176–191. Springer International Publishing, Cham, 2017. doi: 10.1007/978-3-319-58628-1_15

[50] L. E. Matzen, M. J. Haass, K. M. Divis, Z. Wang, and A. T. Wilson. Data visualization saliency model: A tool for evaluating abstract data visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):563–573, 2018. doi: 10.1109/TVCG.2017.2743939

[51] A. L. Michal and S. L. Franconeri. Visual routines are associated with specific graph interpretations. *Cognitive research: principles and implications*, 2(1):1–10, 2017. doi: 10.1186/s41235-017-0059-2

[52] A. Newman, B. McNamara, C. Fosco, Y. B. Zhang, P. Sukhum, M. Tancik, N. W. Kim, and Z. Bylinskii. TurkEyes: A web-based toolbox for crowdsourcing attention data. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, p. 1–13. ACM, New York, NY, USA, 2020. doi: 10.1145/3313831.3376799

[53] J. Pan, E. Sayrol, X. Giro-I-Nieto, K. McGuinness, and N. E. O'Connor. Shallow and deep convolutional networks for saliency prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 598–606. IEEE, Piscataway, NJ, USA, 2016. doi: 10.1109/CVPR.2016.71

[54] A. Papoutsaki, J. Laskey, and J. Huang. SearchGazer: Webcam eye tracking for remote studies of web search. In *Proceedings of the ACM Conference on Human Information Interaction and Retrieval*, p. 17–26. ACM, New York, NY, USA, 2017. doi: 10.1145/3020165.3020170

[55] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays. WebGazer: Scalable webcam eye tracking using user interactions. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3839–3845. AAAI Press, Palo Alto, CA, USA, 2016.

[56] P. Polatsek, M. Waldner, I. Viola, P. Kapec, and W. Benesova. Exploring visual attention and saliency modeling for task-based visual analysis. *Computers & Graphics*, 72:26–38, 2018. doi: 10.1016/j.cag.2018.01.010

[57] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, vol. 139, pp. 8748–8763. PMLR, 2021.

[58] N. Reddy, S. Jain, P. Yarlagadda, and V. Gandhi. Tidying deep saliency prediction architectures. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 10241–10247. IEEE, Piscataway, NJ, USA, 2020. doi: 10.1109/IROS45743.2020.9341574

[59] E. D. Reichle, A. E. Reineberg, and J. W. Schooler. Eye movements during mindless reading. *Psychological Science*, 21(9):1300–1310, 2010. doi: 10.1177/0956797610378686

[60] L. Renninger, J. Coughlan, P. Verghese, and J. Malik. An information maximization model of eye movements. In *Proceedings of the Advances in Neural Information Processing Systems*, vol. 17. MIT Press, Cambridge, MA, USA, 2004.

[61] N. Riche, M. Duvinage, M. Mancas, B. Gosselin, and T. Dutoit. Saliency and human fixations: State-of-the-art and study of comparison metrics. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1153–1160. IEEE, Piscataway, NJ, USA, 2013. doi: 10.1109/ICCV.2013.147

[62] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241. Springer International Publishing, Cham, 2015. doi: 10.1007/978-3-319-24574-4_28

[63] R. Shi, N. K. Ngan, and H. Li. Gaze-based object segmentation. *IEEE Signal Processing Letters*, 24(10):1493–1497, 2017. doi: 10.1109/LSP.2017.2739200

[64] H. Siirtola, T. Laivo, T. Heimonen, and K. Räihä. Visual perception of parallel coordinate visualizations. In *Proceedings of the IEEE International Conference on Information Visualisation*, pp. 3–9. IEEE, Piscataway, NJ, USA, 2009. doi: 10.1109/IV.2009.25

[65] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, 2015.

[66] B. Steichen, G. Carenini, and C. Conati. User-adaptive information visualization: Using eye gaze data to infer visualization tasks and user cognitive abilities. In *Proceedings of the ACM Conference on Intelligent User Interfaces*, p. 317–328. ACM, New York, NY, USA, 2013. doi: 10.1145/2449396.2449439

[67] B. Strobel, S. Sass, M. A. Lindner, and O. Köller. Do graph readers prefer the graph type most suited to a given task? insights from eye tracking. *Journal of Eye Movement Research*, 9(4):1–15, 2016. doi: 10.16910/jemr.9.4.4

[68] H. R. Tavakoli, F. Ahmed, A. Borji, and J. Laaksonen. Saliency revisited: Analysis of mouse movements versus fixations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6354–6362. IEEE, Piscataway, NJ, USA, 2017. doi: 10.1109/CVPR.2017.673

[69] E. Vig, M. Dorr, and D. Cox. Large-scale optimization of hierarchical features for saliency prediction in natural images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 2798–2805. IEEE, Piscataway, NJ, USA, 2014. doi: 10.1109/CVPR.2014.358

[70] W. Wang, J. Shen, F. Guo, M.-M. Cheng, and A. Borji. Revisiting video saliency: A large-scale benchmark and a new model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4894–4903. IEEE, Piscataway, NJ, USA, 2018. doi: 10.1109/CVPR.2018.00514

[71] P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao. TurkerGaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.

[72] A. L. Yarbus. Eye movements during perception of complex objects. In *Eye Movements and Vision*, pp. 171–211. Springer, Boston, MA, USA, 1967. doi: 10.1007/978-1-4899-5379-7_8