IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 20 September 2023, accepted 16 October 2023, date of publication 26 October 2023, date of current version 2 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3327743

RESEARCH ARTICLE

Webpage Validation by Visualizing Importance Area

YUYA INAGAKI[1], JUNKO SHIROGANE[2], (Member, IEEE), HAJIME IWATA[3], (Member, IEEE), AND YOSHIAKI FUKAZAWA[1], (Member, IEEE)

[1]School of Fundamental Science and Engineering, Waseda University, Shijuku-ku, Tokyo 169-8555, Japan
[2]School of Arts and Sciences, Tokyo Woman's Christian University, Suginami-ku, Tokyo 167-8585, Japan
[3]Faculty of Information Technology, Kanagawa Institute of Technology, Atsugi, Kanagawa 243-0292, Japan

Corresponding author: Junko Shirogane (junko@lab.twcu.ac.jp)

ABSTRACT Web designers should develop webpages that are easy to navigate and engage users. However, an aesthetic design does not guarantee usability. Users tend to avoid webpages with low usability, resulting in poor service and sales. One cause of inadequate usability is the difference between the intended content and what the users see on the webpage. On the other hand, visual saliency is the ease with which the human eye notices something. Visual saliency maps (VSMs) are used to recognize areas in an image where people pay attention. VSMs often represent the visual features of landscapes and human faces. Here, we propose a method to generate a webpage-specific VSM based on the layout. Our method computes and visualizes the visual salience level for each webpage element, allowing web designers to easily recognize webpage elements with a high visual prominence. An evaluation revealed that the quality of VSMs generated by our method is higher than those generated by existing methods. Moreover, the VSMs generated by our method are easier to recognize than those of existing methods.

INDEX TERMS Visual saliency, webpage, sightline, usability.

I. INTRODUCTION

Webpages are ubiquitous. As of January 2023, there are more than 1.1 billion websites providing information, and the number is increasing [1]. This is about 1.8 times more than the number of webpages in 2013 because the importance of webpages has increased as they can be viewed anytime and anywhere on a mobile device.

Web designers strive to create visually appealing webpages while ensuring that they are easily navigated and engaging. Even the most aesthetic designs can suffer from poor usability if users struggle to find the information they need. ISO 9241-11:2018 defines usability as the "extent to which a system, product, or service can be used by users to achieve specific goals with effectiveness, efficiency, and satisfaction within a specified context" [2]. A truly usable webpage empowers users to accomplish their purposes with ease and efficiency, satisfying their needs and expectations. However, webpages with subpar usability tend to drive away users, decreasing service quality and sales. For example, when WriteWork, which is a platform where students share papers, modified its layout after a heat map survey [3], its sales for paid services increased by 50%. This example highlights that the original webpage was not aligned with the web designer's intention due to a disconnect between the intended content and what the users saw.

A Visual Saliency Map (VSM) of the target webpage to the web designer can effectively resolve this issue [4]. A VSM represents the ease of paying attention using people's eyes and sightlines. The VSM allows a web designer to easily identify areas that attract users (i.e., important areas). If the important areas differ from the designer's intention, the webpage can be modified. Although many studies have generated VSMs for natural images such as sceneries and human faces [5], [6], [7], [8], few have generated webpage-specific VSMs.

Previously we proposed a method to generate webpage-specific VSMs [9]. A webpage contains various elements

The associate editor coordinating the review of this manuscript and approving it for publication was Sandra Baldassarri.

such as a title and images. These elements differ from the visual saliencies of natural images. Thus, our previous method combined a webpage layout with a VSM. The webpage elements were used to represent visual saliencies. However, humans tend to move their eyes when viewing a webpage [10]. An example of such tendencies is called the "f-bias" [11], in which the human eye moves from the top-left to the center area of a webpage. These tendencies must be considered to generate a webpage-specific VSM.

In this paper, we propose a method to generate a webpage-specific VSM by considering the tendencies of the human eye moving on a webpage. Although eye movement data when viewing a webpage is necessary to calculate the proper visual saliency, there are few existing eye movement datasets, especially for recent modern webpage design on the internet. Herein we create an original dataset, calculate the visual saliency by webpage elements using this dataset, and generate a visualization of the visual salient area of a webpage.

The rest of this paper is organized as follows. Section II overviews visual saliency. Section III describes related works. Section IV discusses the features of our method, while Section V explains the collected sightline data. Section VI details the generation of our webpage-specific VSM. Section VII evaluates our method. Finally, Section VIII concludes this paper.

II. VISUAL SALIENCY

Visual saliency indicates the ease of paying attention using people's eyes [12]. A more eye-catching element has a higher visual saliency. As a VSM represents the visual saliency of an image, many methods have been proposed to create a VSM for scenery or a human face. VSMs can be categorized into two types: a grayscale map and a heat map. A grayscale map represents the visual saliency using a gray scale, where a light-colored area indicates the area with high visual saliency, while a heat map denotes visual saliency by color.

Figure 1 shows an example of a VSM, where (a) is the input image of a cherry blossom, and (b) and (c) are the VSMs using a grayscale map and heat map, respectively. In the heat map, a warm color indicates an area with high visual saliency.

A model for generating a VSM is called a "VSM generation model." A VSM generation model produces a VSM by combining the characteristics of the human eye such as color, intensity, and orientation. Recently, many models using machine learning or neural networks have been proposed [13], [14], [15], [16].

(a) Input image (b) Visual saliency map (Gray scale) (c) Visual saliency map (Heat map)

FIGURE 1. Example of a VSM of an image [9].

III. RELATED WORKS

Many studies are related to our research such as VSM generation models for natural images and graphic designs, structure analyses of webpages, and VSM generation models for webpages. Various VSM generation models for natural images have been proposed. Itti et al. reported a well-known basic VSM generation model [17]. Their model extracts visual characteristics similar to perceptions of the human eye such as color, intensity, and orientation. Then it generates a VSM by weighting and summing these characteristics. This model is often used in research related to visual saliency.

Recently, many VSM generation models using deep learning have been proposed. That is, a VSM model was generated after learning from large datasets. SalNet used a convolutional neural network to predict visual saliency of an image [18]. SALICON [19], ML-NET [20], and Deep Gaze 2 [21] were also trained using convolutional neural networks. Deep Gaze 2 has been evaluated as the best model compared with other models by mit300 [22], which was a VSM benchmark of MIT. However, it is difficult to generate a VSM appropriate for webpages because these models are intended for natural images.

VSM generation models have also been proposed for graphic designs. Bylinskii et al. classified the targets into two types: graphic designs and data with text and tables. Additionally, they classified the collected datasets into two types: those used to create a neural network model and those used to generate VSMs [23]. Bylinskii et al. generated higher quality VSMs than those using existing VSM generation models. However, their method did not analyze webpage structures, and a webpage-specific VSM is difficult to generate.

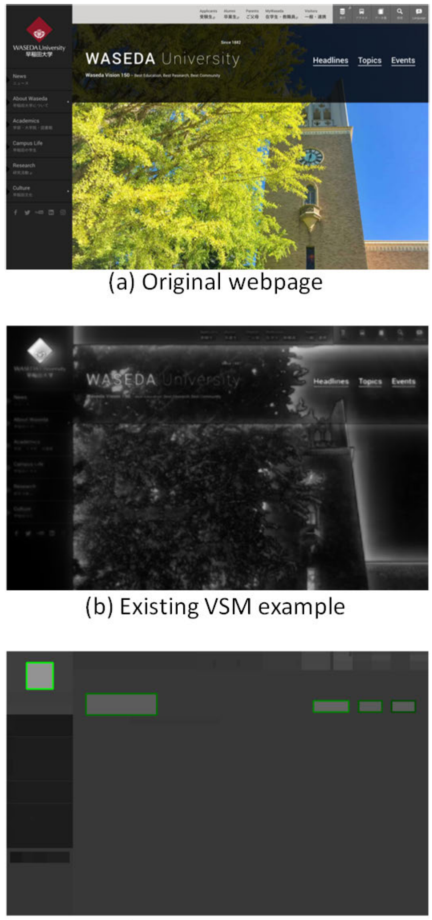
For structure analysis of webpages, Nonaka et al. proposed a method to analyze webpage structures and tags and to extract how-to information [24]. They compared this method with existing methods and showed effective results. Cai et al. proposed a method to identify the relationships among webpage contents based on visual representations [25]. This method extracted webpage structures. Although these methods can analyze webpage structures, they do not show areas with a high visual saliency.

Some researchers have investigated VSM generation of webpages and applications. Human attention can be classified by a bottom-up factor or a top-down factor. The bottom-up factor indicates colors, intensity, and orientation. By contrast, the top-down factor indicates memories and knowledge based on past experiences. Shen et al. combined these factors and proposed a method to generate a VSM for a webpage [26]. They considered that the top-down factors were position bias (e.g., tendency to pay attention to upper-left areas) and face bias (e.g., tendency to pay attention to human faces). Then they used multiple kernel learning [27], integrated all characteristics, and generated a VSM for a webpage. Zheng et al. proposed a method to prepare a visual saliency prediction model to identify areas to which users pay attention in various task conditions [28]. They combined a task-specific prediction model with a free browse prediction model without tasks to create a task-driven VSM generation model. However, these studies used VSMs based on pixels. By contrast, our study assumes that a VSM for a webpage must be represented by webpage elements.

IV. FEATURES

Our method generates a VSM for a webpage by combining an existing VSM generation model and the webpage structure. To predict the visual saliency level by webpage elements, we collected a sightline dataset. As described in Section III, human attention is classified by bottom-up and top-down factors. For a bottom-up factor, drastically different colors, intensities, or orientations in one area from those in other areas indicate that humans pay attention. For a top-down factor, when humans have knowledge about an area, they pay attention to it. Compared with natural images, a webpage has various elements such as photos, texts, and logos. These elements make it difficult to predict visual saliency. Meanwhile, when viewing a webpage, human sightlines tend to concentrate in the top left and center, which is called the f-bias. Hence, the f-bias must be considered to predict visual saliency. This requires sightline data, but there are few datasets available. Moreover, none of the datasets are on recent modern webpage designs. Consequently, we created an original dataset, which can be used to generate VSMs for both traditional and modern designs.

Existing VSMs use grayscale representations to show pixels, making it challenging to identify high visual saliency areas. Users tend to rely on webpage elements such as tags of images, titles, and links to recognize important areas. Unlike existing VSMs, our approach focuses on webpage elements. This allows the visual saliency levels for webpage elements to be calculated, ranked, and displayed. The VSMs generated by our method are called Visual Saliency Area Maps (VSAMs). VSAMs help web designers effectively identify areas with a high visual saliency. Figure 2 [30] compares an existing VSM and our VSAM. (b) and (c) are VSM and VSAM that are generated from (a) by using an existing VSM generation method and our method, respectively. In our VSAM (Figure 2(c)), areas with different visual saliency levels are depicted as green boxes and gray colors. The different shades of gray correspond to different saliency levels, providing a clear representation that garners users' attention.



(a) Original webpage

(b) Existing VSM example

(c) VSAM example

FIGURE 2. Example of an existing VSM and a VSAM.

V. SIGHTLINE DATASET

To precisely predict visual saliency considering f-bias, sightline dataset, which reflects how users see webpages, must be prepared. We created an original dataset because existing ones are not based on modern webpage design. The first step was to collect webpage screenshots from the website, which had many links to Japanese websites. The links were classified into 27 categories such as corporate, food, and politics. For each category, we randomly selected 10 websites for a total of 270 websites. Then we captured screenshots of their top pages.

The second step was to acquire sightline data from 35 subjects using these screenshots. The subjects included 27 males and 8 females, who ranged in age from 19 to 26. All subjects used the internet in their daily lives and were unimpaired people. Each subject viewed 20 screenshots in a random order. Thus, each screenshot was seen by five different subjects. The experiment began with a calibration. Then the subject viewed a screenshot for ten seconds followed by a black screen for five seconds. This process was repeated until all screenshots were viewed. The subjects saw the screenshots freely as we did not provide any instructions prior to viewing. Their sightline data were recorded by Tobii Pro Nano [29].

The final step was to analyze the collected data. We classified the screenshot tendencies into seven layout patterns (Table 1). The layout patterns were defined by referring to books and web articles on sightlines. Figure 3 shows examples of the layout patterns. For the analysis, the 270 screenshots were randomly divided into a ratio of 4:1 (216:54). The 216 screenshots were used for analyzing sightline data, and the remaining 54 were used for the evaluations described in Section VII.

TABLE 1. Seven layout patterns.

Pattern name	Description
Single column	Contents are ordered vertically in a column
Two column	Contents are ordered vertically in two columns
Three column	Contents are ordered vertically in three columns
Multi column	Contents are ordered vertically in over four columns
Card layout	Contents are ordered like cards
Broken grid	Contents are arranged in various position
Full screen	A content is shown as a full screen

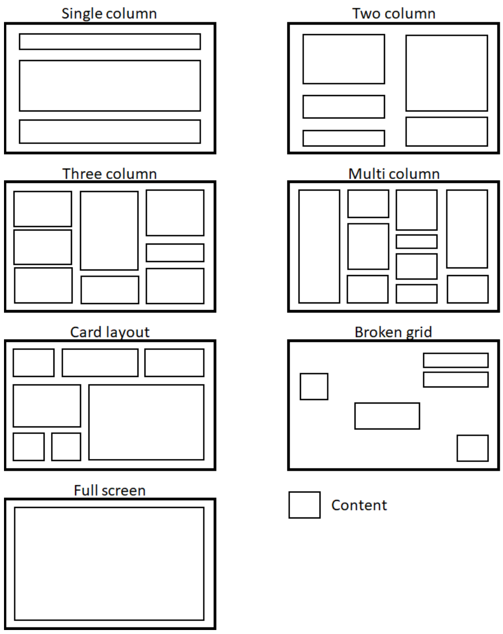


FIGURE 3. Examples of layout patterns.

As described above, the subjects viewed each screenshot for ten seconds. According to the tracking of the sightline coordinates, the sightlines moved a lot. Thus, to analyze the movement tendency by layout pattern, we initially divided the sightline data into five groups using two-second intervals (0-2, 2-4, 4-6, 6-8, and 8-10 seconds) and specified the gaze points in these time ranges. Next, we calculated the average of the coordinates by screenshot and time range (average coordinates) and the average of the coordinates of all the screenshots by layout pattern and time range (median points). The median points indicated the average coordinates of the sightlines by the layout patterns and the time ranges. The results show that sightline movement can be followed in a layout pattern.

Moreover, we drew circles that included the average coordinates of all screenshots and the time ranges by the layout pattern. Table 2 shows the x- and y-coordinates of the median points and the circle radii by time range. Figure 4 shows the circles and the trajectories of the median points by layout pattern. The pink circles indicate that the average coordinates of all screenshots and the time ranges are included, while red lines indicate the trajectories of the median points. The central coordinates are the median points, and the "×" marks are the average coordinates of the time ranges.

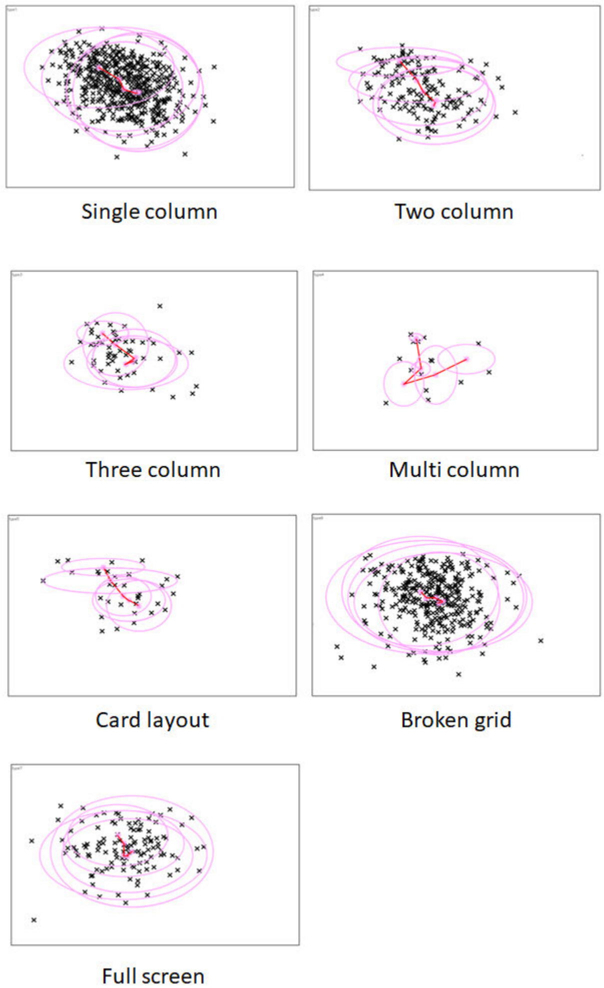


FIGURE 4. Circles and trajectories of the median points.

If the webpage contents are included in the pink circles in Figure 4, users will likely see the contents. For all the layout patterns, sightlines tend to move from the top-left to the center area over time, which is called the f-bias. However, the sightline movements differ by the layout pattern. In column layouts (e.g., one, two, three, or multi columns), the sightlines move a lot from the top left to the center over time. Meanwhile, in modern design layouts (e.g., broken grid and full screen), the sightlines do not move a lot over time. Instead, they concentrate in the center area. These results demonstrate that our method generates a VSAM considering sightline movements by the layout pattern.

TABLE 2. Median points and circle radii.

Layout pattern	Time range	Median point (X)[px]	Median point (Y)[px]	Radius (X) [px]	Radius (Y) [px]
Single column	0-2s	556.53526	368.23793	444.82839	237.78349
	2-4s	680.65559	438.25877	466.89552	308.36171
	4-6s	707.52981	496.78213	349.64802	278.91648
	6-8s	768.12999	511.91153	408.86725	338.98615
	8-10s	788.44816	515.60635	389.02887	352.11794
Two column	0-2s	537.90889	327.89538	371.11744	84.03475
	2-4s	638.07192	436.23222	396.09261	131.80240
	4-6s	673.81014	500.41143	382.18434	202.55096
	6-8s	748.26660	563.43498	338.23595	204.41473
	8-10s	738.18496	583.13415	359.40345	226.29391
Three column	0-2s	553.66790	375.01389	156.00054	74.33003
	2-4s	630.30464	442.51196	197.01774	190.90868
	4-6s	740.487670	524.98878	250.85640	172.198448
	6-8s	692.25816	558.33862	371.49920	154.66621
	8-10s	729.13148	544.05520	271.23804	152.74239
Multi column	0-2s	622.93684	403.41291	39.81908	23.29113
	2-4s	652.95710	587.20462	57.61500	43.52946
	4-6s	550.20828	680.73009	119.84877	132.47447
	6-8s	741.04658	624.54582	129.73321	175.20079
	8-10s	926.99540	530.66825	172.67673	87.60673
Card layout	0-2s	569.89335	310.07291	248.88284	51.79706
	2-4s	612.14546	391.96698	399.64150	78.47047
	4-6s	688.60131	481.16412	166.22500	119.72984
	6-8s	723.86810	507.66301	216.52019	128.41249
	8-10s	782.71513	540.81882	194.26017	152.85102
Broken grid	0-2s	645.74007	458.44231	466.02712	326.28638
	2-4s	682.06057	494.16684	596.61873	335.88933
	4-6s	740.43617	500.64521	522.82219	322.04717
	6-8s	783.37446	520.62385	530.18073	292.10090
	8-10s	759.12313	529.81126	360.04529	305.44824
Full screen	0-2s	640.26711	423.96134	298.25324	184.01572
	2-4s	673.68514	472.42480	359.91272	197.11066
	4-6s	672.59751	542.48711	497.06624	264.81229
	6-8s	694.48142	545.09160	389.88440	221.83996
	8-10s	722.54375	520.62472	484.22990	333.35191

VI. VSAM GENERATION METHOD

Figure 5 overviews our method. It begins when a web designer inputs a URL (Uniform Resource Locator) of the target webpage and its layout pattern. Then our system generates a VSM, analyzes the structure of the webpage, and combines the VSM and structure. Next, our system analyzes areas with high visual saliencies by webpage elements of HTML (Hyper Text Markup Language) and generates a VSAM. Finally, our system visualizes the VSAM.

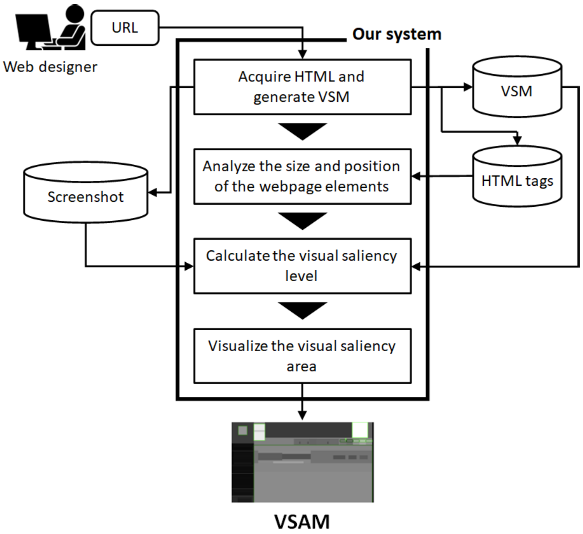


FIGURE 5. System structure.

Our method consists of the following four steps.

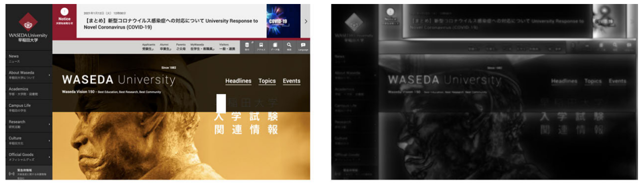
Step 1 Acquire HTML and generate the VSM
Step 2 Analyze the size and position of webpage elements
Step 3 Calculate the visual saliency level
Step 4 Visualize the visual saliency area

A. ACQUIRE HTML AND GENERATE THE VSM

Our system acquires a screenshot and HTML file using a web scraping technique from the inputted URL to generate a VSM of the screenshot using the VSM generation model proposed by Itti et al. [17]. This model is a simple VSM generation model. The human eye can recognize color, intensity, and orientation. This model extracts these properties from an image, weights them, and calculates their weighted sum to generate a VSM. Although this model generates a VSM using red, blue, and green channels of the RGB (Red-Blue-Green) color model, our system generates the VSM as a grayscale image to simplify the VSM calculation. Because the visual saliency level of webpage areas can be represented sufficiently by grayscale tones in our VSAM, generating the VSM as a grayscale image does not affect the representation of the visual saliency level. Figure 6 shows an example of a generated VSM, where the left image (a) is the inputted screenshot of the top page of Waseda University [31] and the right image (b) is the generated VSM.

B. ANALYZE THE SIZE AND POSITION OF WEBPAGE ELEMENTS

Our system extracts webpage elements represented by the seven tag types from the HTML file acquired in Section VI-A(Table 3). The webpage elements include the size and position. The position is defined using the top-left point of a web browser as the reference point (0, 0), and the horizontal and vertical distances from the reference point are defined as the coordinates of the webpage element. Figure 7 shows examples using the homepages of Waseda University [31] and Yahoo! JAPAN [32]. The webpage elements are bordered in this figure.

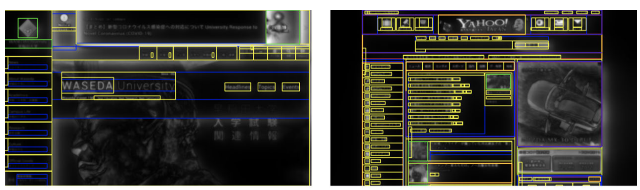


(a) Input: Waseda University web page (b) Output: Waseda University web page

FIGURE 6. Example of the generated VSM.

TABLE 3. Tags to extract webpage elements.

Tag	Use
<div>	Block element grouping
<h1>	Highest heading
<h2>	Second highest heading
<h3>	Third highest heading
<a>	Link to other page
<span>	In-line element grouping
<p>	Paragraph
<input>	Input element
<img>	Image



Waseda University web page (2021/01/17) Yahoo! JAPAN (2021/01/17)

FIGURE 7. Examples of the acquired size and position.

C. CALCULATE THE VISUAL SALIENCY LEVEL

Using the VSM (Section VI-A) along with the size and position (Section VI-B), our system calculates the visual saliency levels of webpage elements by considering the layout pattern of the target webpage. First, our system reads the VSM. Recently, high-definition displays show a dot in an image using pixels. Because a screenshot of the target webpage is recorded in a higher resolution than non-high-definition displays, the sizes and positions of webpage elements acquired in Section VI-B differ from those in the screenshot. Thus, our system determines the resolution of the

Algorithm 1 Calculation Algorithm for the Visual Saliency Density

function get_element_salient_level(start_x, start_y, end_x, end_y)
  if (end_x - start_x) > 0 and (end_y - start_y) > 0 **then**
    clipped = Element.canvas.cv2[start_y:end_y, start_x:end_x] // Webpage element extraction
    element_occupancy = $\frac{(end\_x - start\_x) \times (end\_y - start\_y)}{width * height}$ // Calculation of webpage element occupancy
    salient_level = $\frac{get\_element\_total\_saliency(cliped)}{get\_total\_saliency() \times element\_occupancy}$ // Calculation of visual saliency level for webpage element
    return salient_level
  else
    return 0
  end if
end function

function get_total_saliency
  clipped = Element.canvas.cv2[0:height, 0:width] // Whole webpage
  total_saliency_per_row = np.sum(clipped, axis=0) // Calculation of sum of colors in a row
  total_saliency = np.sum(total_saliency_per_row, axis=0) // Calculation of sum of colors in a column
  return np.uint8(total_saliency)
end function

function get_element_total_saliency(clipped)
  total_saliency_per_row = np.sum(clipped, axis=0) // Calculation of sum of colors in a row
  total_saliency = np.sum(total_saliency_per_row, axis=0) // Calculation of sum of colors in a column
  return np.uint8(total_saliency)
end function

screenshot based on the number of pixels per dot. Second, the area corresponding to a webpage element from the VSM (Section VI-B) is extracted to calculate the visual saliency level of the webpage element (Section VI-A). Third, the integral of the visual saliency level is calculated by pixels for an extracted area. Fourth, the average of the visual saliency level for the extracted area is calculated. Finally, the integral is divided by the average to give a visual saliency density for the extracted area (webpage element). Algorithm 1 shows the algorithm of these calculations.

Although our previous paper specified the average of the visual saliency levels for a webpage element, the visual saliency level for a big-sized webpage element became low [9]. To resolve this issue, this paper calculates the visual saliency density for a webpage element. Consequently, the visual saliency levels for webpage elements with different sizes can be calculated.

1) WEIGHTING BY SIZE

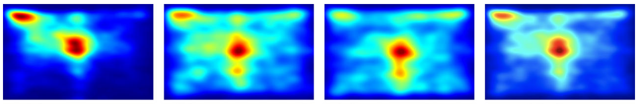
Because the visual saliency level for a small-sized webpage element becomes excessively high in algorithm 1, the size of a webpage element is weighted. Concretely, the weight of webpage elements under a certain size such as icons, is decreased. According to the guidelines of Microsoft Windows, 32 px × 32 px is the standard icon size [33]. According to the Human Interface Guidelines of Apple Inc., 32 px × 32 px or 64 px × 64 px are the standard icon sizes in iOS and macOS [34]. Thus, our system defines an area size of a webpage element less than 64 px × 64 px as an icon (icon element) and assigns a low weight. Algorithm 2 shows the weighting algorithm. For the weight value, visual saliency levels of icon elements are calculated many times using various values to ensure that the calculated visual saliency level is near the actual level.

Algorithm 2 Weight Algorithm by Element Size

function apply_size_bias(salient_level, start_x, start_y, end_x, end_y)
  element_area = (end_x - start_x) × (end_y - start_y) // Calculation of area size of webpage element
  if element_area > 64 × 64 **then**
    salient_level = salient_level
  else
    salient_level = salient_level × 0.5
  end if
  return salient_level
end function

2) WEIGHTING BY POSITION AND LAYOUT PATTERN

Webpages differ from natural images, and sightlines by humans move from the top left to the center (i.e., f-bias). Figure 8 shows the heat maps of sightline data in which subjects viewed webpage screenshots for ten seconds (Section V). Figures 8(a)-(c) are the heat maps divided by three seconds, and (d) is the merged heat map for sightline data of the entire ten seconds. According to (d), the top-left and center areas clearly receive attention, while according to (a)-(c), subjects' sightlines move from the top-left to the center area. These results confirm that the f-bias exists in webpages with modern design and the areas receiving attention change over time.



(a) 0ms-3000ms (b) 3000ms-6000ms (c) 6000ms-9000ms (d) All time

FIGURE 8. Heat maps from the experiment.

To reflect such biases in VSAMs, the position of a webpage element is weighted based on the layout pattern. Previously, we weighted the top-left and center areas [9], but the VSAM was not appropriately represented because a pseudo bias was adapted to all layout patterns of webpages. In this paper, we analyze the tendencies of sightline data by layout pattern and weighted the webpage elements accordingly. Figure 4 represents the movement of the sightline data by layout pattern, where the ellipses include all "×" marks except those that are far off. When the center of a webpage element is included in an ellipse, the webpage element is likely to be seen.

According to Figure 4, sightlines for column layouts change over time from the top-left to the center area, while sightline movement in modern layouts remains around the center area. Algorithm 3 weights webpage elements based on these features by position and layout pattern. In this algorithm, the visual saliency area of the layout pattern for the target webpage is specified based on Table 2. Then the *judge_inside_ellipse* function evaluates whether the central coordinate of each webpage element in the target webpage is included in the specified visual saliency area. Finally, if the central coordinate is included, a weight *weight_position* is multiplied by the visual saliency level of the area. Consequently, the visual saliency levels by webpage element are determined.

D. VISUALIZE THE VISUAL SALIENCY AREA

Our system depicts the visual saliency areas (Section VI-C) by generating a VSAM, where the webpage elements with high visual saliency are represented by a light-dark change.

1) RANK DETERMINATION FOR THE VISUAL SALIENCY LEVEL

Our system arranges the webpage elements in descending order based on the calculated visual saliency levels (Section VI-C). A web designer defines the number of ranks (the default is ten), and the webpage elements are ranked. A webpage element (parent element) often includes some webpage elements (child elements) such as "<div><p>sentence</p></div>" (<div> is a parent element, while <p> is a child element). In this case, when the visual saliency level of a child element is high, the parent element of the child element becomes high. Consequently, other child elements also become high. That

Algorithm 3 Weight Algorithm by Position and Layout Pattern

```
function apply_position_bias(salient_level, start_x, start_y, end_x, end_y, type)
    weight_position ← Weight for position bias
    bias_info_list ← List of coordinates of median point
    center_x = (start_x + end_x) / 2 // x-coordinate of median point
    center_y = (start_y + end_y) / 2 // y-coordinate of median point
    for <bias_info_list> do
        if judge_inside_ellipse(center_x, center_y, bias_info) then
            return salient_level × weight_position
        else
            return salient_level
        end if
    end for
end function

function judge_inside_ellipse(center_x, center_y, bias_info)
    return Return whether median point exists in ellipse
end function
```

is, these containment elements are included multiple times in the rankings. To resolve this problem, when a webpage element has a high rank, the parent element is excluded from the ranking. Moreover, extremely long and thin webpage elements such as decorations are excluded as they should not have high ranks.

2) VSAM GENERATION

A VSAM is generated by filling webpage elements with 256 levels of gray based on the calculated visual saliency levels (Section VI-C). Because a single-precision floating point format represents the calculated visual saliency levels, the levels must be represented by values of 0-255. Our system assigns a value of 255 to the webpage elements with the highest saliency level and converts the other values of the visual saliency levels to the other 256 levels of gray. That is, when the value of the highest visual saliency level is v_h, the value of the other visual saliency levels is v_o, and the calculated 256 levels of gray is g_o, the calculation formula is $g_o = (v_o/v_h) \times 255$ (g_o is rounded off to the closest whole number). Then all webpage elements are filled with the assigned/calculated number of grays. Additionally, light green borders are added to the ranked webpage elements (Section VI-D1). Due to the border, webpage elements with higher visual saliency levels can be easily distinguished. Figure 9 shows an example of a VSAM.

VII. EVALUATION

We conducted two experiments to investigate our method. One assessed the appropriateness of determining the visual

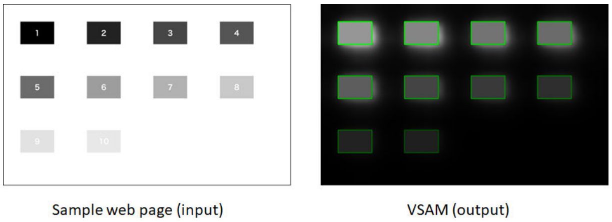


FIGURE 9. Example of VSAM generation.

saliency levels of webpage elements and the other evaluated the effectiveness of VSAMs.

A. APPROPRIATENESS OF DETERMINING VISUAL SALIENCY LEVELS

To evaluate the appropriateness of determining visual saliency levels, we analyzed the sightline data of 54 webpage screenshots. The 54 screenshots were part of the original 270 screenshots (Section V). We used three visual saliency metrics as evaluation criteria: Area Under the Curve (AUC), Linear Correlation Coefficient (CC), and Kullback-Leibler divergence (KL) provided by Bylinskii et al. [35]. These metrics compare the VSM under evaluation (target VSM) to the appropriate VSM (baseline VSM). AUC, which is a popular metric for VSM generation models, is evaluated by a value from 0 to 1. A value of 0.5 means it is random, and a high value indicates a high-quality target VSM. CC calculates the correlation coefficient between the target VSM and the baseline VSM. The baseline VSM is created by the sightlines of subjects. The correlation coefficient ranges from -1 to 1. A high value indicates a high-quality target VSM. KL is a metric that analyzes the similarity of the probability distributions between the target VSM and the baseline VSM. It has a value from 0 to 1. If the probability distribution is the same, KL has a value of 0. Thus, a value closer to zero indicates a high-quality target VSM.

In this experiment, we generated VSAMs from the 54 webpages and compared VSAMs that were considered appropriate (baseline VSAMs) with the following four types of VSAMs and VSMs:

- VSAMs of our generation model (our VSAMs)
- VSAMs that the VSM generation model changed to ML-NET (originally, our VSAM generation model used the VSM generation model by Itti [17]) (our VSAM with ML-NET)
- A VSM by SalNet by Pan [18]
- A VSM by ML-NET by Cornia et al. [36]

Figure 10 shows part of these VSAMs. The baseline VSMs and VSAMs were created using the sightline data from the subjects. These were assumed as the baseline VSMs and VSAMs in this evaluation. VSAMs depend on the generation model (Figure 10). In both VSMs and VSAMs, a lighter color indicates an area that users pay attention to. The experiment reveals that our VSAMs are similar to the baseline VSAMs.

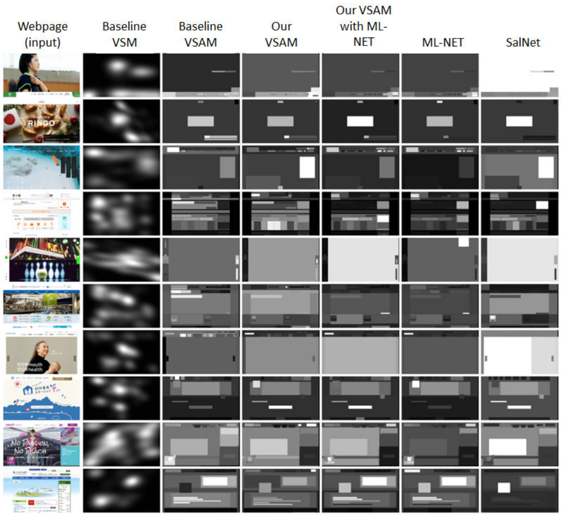


FIGURE 10. Part of VSMs and VSAMs for evaluating appropriateness.

All VSAMs were evaluated using the three metrics. Table 4 shows the average calculated values. Our VSAM shows the best value by AUC and KL. In CC, although our VSAM has the third best value, our VSAM with ML-NET gives the best value. Because CC calculates the correlation coefficient between the target VSM and the baseline VSM, it analyzes differences in brightness between VSAMs. Because the overall visual saliency levels of our VSAMs are low, the correlation coefficients are likely low. However, our VSAM gives the best values by AUC and KL. Our VSAM generation model is appropriate to generate a VSAM of a webpage since the differences between the values of our VSAM and the higher values by CC are small.

TABLE 4. Average values of metrics evaluations.

	AUC	CC	KL
Our VSAM	**0.78612**	0.53475	**0.12348**
Our VSAM with ML-NET	**0.74319**	0.55167	0.13691
SalNet	0.66270	0.36863	0.14679
ML-NET	0.74279	0.54798	0.13777

B. EFFECTIVENESS OF VSAMS

We evaluated the recognizability of the visual salient areas by comparing our VSAM and VSMs by an existing model [17], which is a basic VSM generation model. We selected two webpages randomly from the Japan Web Design Gallery [37] and generated the VSMs and VSAMs (Figure 11).

The experiment involved undergraduate and graduate university students ranging in age from 19 to 25. They used the internet daily. We provided an overview of VSMs and VSAMs. Then the subjects looked at two VSAMs and two VSMs, and answered our interview questions (Table 5).

TABLE 5. Questions after viewing VSM and VSAM.

	Questions	Answer style
Q1	Recognizability of VSM	5 point likert scale [38]
Q2	Recognizability of VSAM	5 point likert scale [38]
Q3	Difficultly of recognizing in VSM	Free writing
Q4	Ease of recognizing in VSM	Free writing
Q5	Difficultly of recognizing in VSAM	Free writing
Q6	Ease of recognizing in VSAM	Free writing
Q7	Map to easily recognize visual salient area	VSM or VSAM

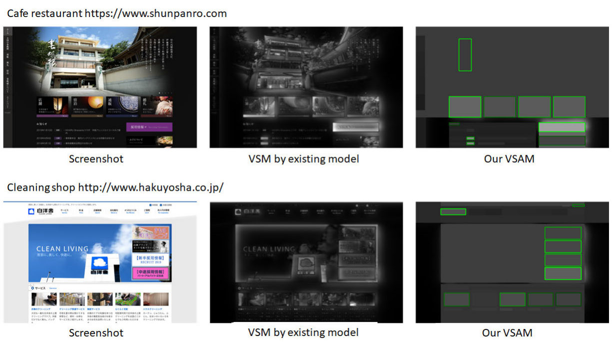


FIGURE 11. Sample VSMs and VSAMs for evaluating the effectiveness.

Figure 12 shows the result for Q1 (about VSM) and Q2 (about VSAM). Nine subjects answered "very recognizable" and "recognizable" for Q2. For Q1, although four subjects answered "recognizable," five subjects answered "difficult to recognize" or "very difficult to recognize." Thus, our VSAMs can recognize visual salient areas more effectively than the VSMs from existing models.

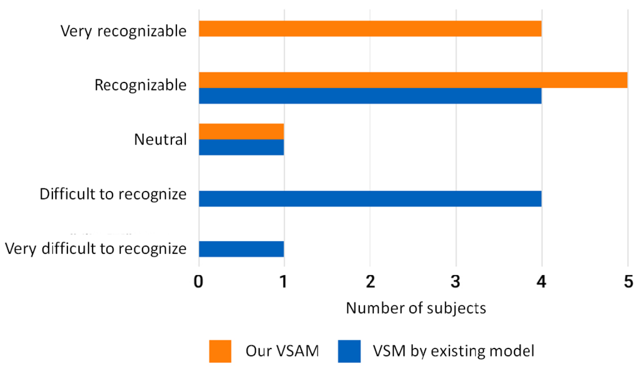


FIGURE 12. Result of the answers for Q1 and Q2 [9].

According to Q3-Q6, for VSMs by the existing model (Q3-4), subjects answered that the borders of webpage elements were difficult to recognize, the highest visual salient area was difficult to recognize, and the visual saliency levels of webpages were difficult to compare. For our VSAMs (Q5-6), subjects answered that these problems were improved. However, for our VSAMs, subjects also answered that because the VSAMs did not show the original webpages, it is necessary to see the screenshots together with the VSAMs, and the differences of gray for webpage elements were difficult to distinguish. These problems were due to filling the webpage elements and not showing webpage contents. To resolve these problems, it is necessary to provide strategies to overlap the webpage screenshots on the VSAMs for easily switching between the webpage screenshots and the VSAMs. Additionally, the rank numbers must be assigned to webpage elements on the VSAMs.

For Q7, all subjects indicated that they could recognize visual salient areas by our VSAMs more easily than the VSMs from existing models. Although our VSAMs can be improved, it is easier to recognize visual salient areas using our VSAMs compared to VSMs from existing models.

C. THREATS OF VALIDITY

All the subjects in the sightline participating in the data collection and evaluations were Japanese, young, and undergraduate/graduate university students. Generally, the results of experiments differ by subjects' characteristics such as age, work, culture, and language. Additionally, the sample webpages used for the sightline data and evaluations were written in Japanese. Thus, if subjects have different characteristics or the sample webpages are in another language, our method may not be able to generate an appropriate VSAM. As described in Section VI-C1, our method treats a webpage element less than 64 px $\times$ 64 px as an icon and assigns a low weight. Consequently, our method will not recognize such webpage elements as high visual salient areas even if they are important.

VIII. CONCLUSION

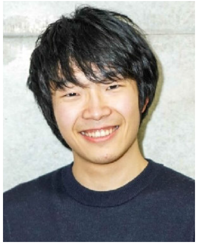
We proposed a method to visualize the visual salient areas of webpages with an emphasis on webpage layout patterns. First, we collected the sightline data of webpages with various layout patterns, analyzed the data, and confirmed the f-bias and tendencies of sightlines by layout patterns. Considering these biases, tendencies, and webpage-specific features, our method calculated the visual saliency levels by every webpage element to generate a VSAM.

The evaluations show that our method can generate a higher quality VSAM than the existing VSM generation model, and the visual salient areas generated by our VSAM are more recognizable than those generated by the existing VSM. That is, our VSAM is appropriate for webpages. Using our method, a web designer can easily recognize webpage elements with high visual saliency when developing a webpage.

In the future, we aim to improve the usefulness of VSAMs. In our VSAM, the visual saliency levels are based on webpage elements, but the webpage contents are not displayed. However, the subjects answered that it is necessary to see screenshots along with the VSAMs to identify the correspondences between a VSAM and the webpage contents.

REFERENCES

[1] NetCraft. *April 2023 Web Server Survey*. Accessed: May 3, 2023. [Online]. Available: https://news.netcraft.com/archives/category/web-server-survey/
[2] *Ergonomics of Human-System Interaction—Part 11: Usability: Definitions and Concepts*, Standard ISO 9241-11:2018, 2018.
[3] *WriteWork Essays & Writing Guides for Students*. Accessed: May 3, 2023. [Online]. Available: https://www.writework.com/
[4] J. D. Still and C. M. Masciocchi, "A saliency model predicts fixations in web interfaces," in *Proc. MDDAUI*, Atlanta, GA, USA, 2010, pp. 25–28.
[5] H. Takimoto, T. Kokui, H. Yamauchi, M. Kishihara, and K. Okubo, "Image modification based on a visual saliency map for guiding visual attention," *IEICE Trans. Inf. Syst.*, vol. 98, no. 11, pp. 1967–1975, 2015, doi: 10.1587/transinf.2015EDP7087.
[6] L. Marchesotti, C. Cifarelli, and G. Csurka, "A framework for visual saliency detection with applications to image thumbnailing," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep. 2009, pp. 2232–2239, doi: 10.1109/ICCV.2009.5459467.
[7] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2006, pp. 1–8.
[8] B. Li, W. Xiong, and W. Hu, "Visual saliency map from tensor analysis," in *Proc. AAAI*, Toronto, ON, Canada, 2012, pp. 1585–1591, doi: 10.1609/aaai.v26i1.8327.
[9] Y. Inagaki, H. Iwata, J. Shirogane, and Y. Fukazawa, "Visualization method of important regions by combination of webpage structures and saliency maps," in *Proc. 15th Int. Conf. Softw. Technol.*, 2020, pp. 204–211.
[10] G. Buscher, E. Cutrell, and M. R. Morris, "What do you see when you're surfing? Using eye tracking to predict salient regions of Web pages," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, New York, NY, USA, Apr. 2009, pp. 21–30, doi: 10.1145/1518701.1518705.
[11] J. Nielsen. *F-Shaped Pattern for Reading Web Content (Original Study)*. Accessed: May 3, 2023. [Online]. Available: https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content-discovered/
[12] L. Itti, "Visual salience," *Scholarpedia*, vol. 2, no. 9, p. 3327, 2007, doi: 10.4249/scholarpedia.3327. [Online]. Available: http://www.scholarpedia.org/article/Visual_salience
[13] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 5455–5463, doi: 10.1109/CVPR.2015.7299184.
[14] Y. Liu, D. Zhang, Q. Zhang, and J. Han, "Part-object relational visual saliency," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3688–3704, Jul. 2022, doi: 10.1109/TPAMI.2021.3053577.
[15] A. Kroner, M. Senden, K. Driessens, and R. Goebel, "Contextual encoder–decoder network for visual saliency prediction," *Neural Netw.*, vol. 129, pp. 261–270, Sep. 2020, doi: 10.1016/j.neunet.2020.05.004.
[16] M. Tliba, M. A. Kerkouri, B. Ghariba, A. Chetouani, A. Çöltekin, M. S. Shehata, and A. Bruno, "SATSal: A multi-level self-attention based architecture for visual saliency prediction," *IEEE Access*, vol. 10, pp. 20701–20713, 2022, doi: 10.1109/ACCESS.2022.3152189.
[17] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998, doi: 10.1109/34.730558.
[18] J. Pan, E. Sayrol, X. Giro-I-Nieto, K. McGuinness, and N. E. O'Connor, "Shallow and deep convolutional networks for saliency prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 598–606, doi: 10.1109/CVPR.2016.71.
[19] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "SALICON: Saliency in context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1072–1080, doi: 10.1109/CVPR.2015.7298710.
[20] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an LSTM-based saliency attentive model," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5142–5154, Oct. 2018, doi: 10.1109/TIP.2018.2851672.
[21] M. Kümmerer, T. S. A. Wallis, and M. Bethge, "DeepGaze II: Reading fixations from deep features trained on object recognition," 2016, *arXiv:1610.01563*.
[22] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba. *MIT Saliency Benchmark*. Accessed: May 3, 2023. [Online]. Available: http://saliency.mit.edu/
[23] 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 *Proc. 30th Annu. ACM Symp. User Interface Softw. Technol.*, New York, NY, USA, Oct. 2017, pp. 57–69, doi: 10.1145/3126594.3126653.
[24] R. Nonaka, T. Yumoto, M. Nii, and Y. Takahashi, "How-to information search by lightweight analysis of Web pages," in *Proc. 3rd Int. Universal Commun. Symp.*, Tokyo, Japan, Dec. 2009, pp. 350–354, doi: 10.1145/1667780.1667853.
[25] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, "Extracting content structure for web pages based on visual representation," in *Proc. Asia–Pacific Web Conf.* Berlin, Germany, 2003, pp. 406–417, doi: 10.1007/3-540-36901-5_42.
[26] C. Shen and Q. Zhao, "Webpage saliency," in *Proc. ECCV*, 2014, pp. 33–46, doi: 10.1007/978-3-319-10584-0_3.
[27] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *J. Mach. Learn. Res.*, vol. 9, no. 83, pp. 2491–2521, 2008. [Online]. Available: https://www.jmlr.org/papers/v9/rakotomamonjy08a.html
[28] Q. Zheng, J. Jiao, Y. Cao, and R. W. H. Lau, "Task-driven webpage saliency," in *Proc. ECCV*, Munich, Germany, 2018, pp. 300–316, doi: 10.1007/978-3-030-01264-9_18.
[29] Tobii. *Tobii Pro Nano*. Accessed: May 3, 2023. [Online]. Available: https://www.tobii.com/products/eye-trackers/screen-based/tobii-pro-nano
[30] *Waseda University*. Accessed: Dec. 10, 2018. [Online]. Available: https://www.waseda.jp/
[31] *Waseda University*. Accessed: Jan. 10, 2021. [Online]. Available: https://www.waseda.jp/
[32] *Yahoo! JAPAN*. Accessed: Jan. 10, 2021. [Online]. Available: https://www.yahoo.co.jp/
[33] Microsoft Corporation. *Icons (Design Basics)*. Accessed: May 3, 2023. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/uxguide/vis-icons
[34] Apple. *App Icons*. Accessed: May 3, 2023. [Online]. Available: https://developer.apple.com/design/human-interface-guidelines/app-icons
[35] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 3, pp. 740–757, Mar. 2019, doi: 10.1109/TPAMI.2018.2815601.
[36] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "A deep multi-level network for saliency prediction," 2020, *arXiv:1609.01064*.
[37] *Japan Web Design Gallery*. Accessed: Jan. 12, 2018. [Online]. Available: http://japan.2-wg.com/
[38] J. Russell and R. Cohn, *Likert Scale*. Warsaw, Poland: Books On Demand, 2012.



YUYA INAGAKI received the B.E. and M.E. degrees in information and computer science from Waseda University, Tokyo, Japan, in 2019 and 2021, respectively. His research interests include visual saliation and webpage usability.

JUNKO SHIROGANE (Member, IEEE) received the B.E., M.E., and D.E. degrees in information and computer science from Waseda University, Tokyo, Japan, in 1997, 1999, and 2002, respectively. She joined the Media Network Center, Waseda University, as a Research Assistant, in 2000; and the Department of Communication, Tokyo Woman's Christian University, as a Lecturer, in 2003, where she is currently an Associate Professor with the School of Arts and Sciences. Her research interest includes support tools for development of software with graphical user interface. She is a member of IPSJ, IEICE Japan, JSSST, and ACM.

HAJIME IWATA (Member, IEEE) received the B.E., M.E., and D.E. degrees in information and computer science from Waseda University, Tokyo, Japan, in 2002, 2004, and 2008, respectively. He joined the Media Network Center, Waseda University, as a Research Assistant, in 2005; and the Department of Network and Communication, Kanagawa Institute of Technology, as a Assistant Professor, in 2008, and an Associate Professor, in 2017. His research interest includes support tools for learning operating method of applications. He is a member of IPSJ, IEICE Japan, and ACM.

YOSHIAKI FUKAZAWA (Member, IEEE) received the B.E., M.E., and D.E. degrees in electrical engineering from Waseda University, Tokyo, Japan, in 1976, 1978, and 1986, respectively. He joined the Department of Computer Science, Sagami Institute of Technology, as a Lecturer, in 1983; and the Department of Electrical Engineering, Waseda University, as an Associate Professor, in 1987, where he is currently a Professor with the Department of Information and Computer Science. His research interests include software engineering, program optimization, and computer aided design. He is a member of IPSJ, IEICE Japan, JSSST, and ACM.