Unsupervised Discovery of Salient Design Features of Websites

T.T. Kaluarachchi, D.M.S. Dissanayake, M.I.E. Wickramasinghe,
University of Colombo School of Computing, Sri Lanka

***Abstract* -- Web development is one of the fastest-growing fields in the IT industry. User Interface (UI) design of a website is critical in attracting new users, which helps businesses increase sales and revenue. A unique website design will encourage user interaction among website visitors and ensure that the time and resources spent on a webpage are worthwhile. Web designers create websites either by using pre-existing templates or by building them from scratch. The web designer's design skills heavily influence the overall appearance of a website. However, such websites do not always meet the client's expectations. As a result of these challenges and the ever-changing web development trends, the automatic website generation concept has emerged, which generates websites without relying on human interaction. In this concept, it is useful to understand how to classify websites based on their appearance and how to identify design features that distinguish websites. This study aims to develop a classification system for websites on the internet based on their salient design features.**

***Index Terms* - web designing, automatic website generation, templates, clustering, self-organizing-map**

I. INTRODUCTION

The *World Wide Web*, also known as *WWW*, *Web,* or *W3,* is a network of online content formatted in Hypertext Markup Language (HTML) and accessible via Hyper Text Transfer Protocol (HTTP). Any downloadable media can be defined as a web resource. Web pages are the building blocks of websites, usually a text file containing hypertext written in HTML or a similar markup language. This format allows users to easily navigate to other web resources via a Uniform Resource Locator (URL). In addition to text, web pages may contain images, video, audio and scripts that run within the user's web browser. Websites serve many purposes and can be used in various ways. A website can serve various purposes, such as a personal blog, an e-commerce website, an informational website, an online community website, a government website, a non-profit organization website, and so on. There are over 3.16 billion websites on the internet, according to the Netcraft Web Server Survey conducted in February 2023[1].

The UI is what connects a user to a web page. It focuses on anticipating what users need to do and ensuring that the interface has features that are easy to access, understand, and use to facilitate those actions. The UI design of a website is critical in attracting new users, which helps businesses increase sales and revenue. A unique website design will encourage user interaction among website visitors and ensure that the time and resources spent on a webpage are worthwhile. A literature review by Garett R. et al. [10] shows that seven website design elements: navigation, graphical representation, organization, content utility, purpose, simplicity, and readability can affect user engagement on websites. Website designers can use these design elements to implement best practices in their web designs with high user engagement such as keeping the interface simple, creating consistency and using common UI elements, strategically using colour and texture.

Web designers create websites either by using pre-existing templates or by building them from scratch. There are a number of free and paid templates available online. The web designer's design skills heavily influence the overall appearance of a website. However, such websites do not always meet the client's expectations. As a result of these challenges and the ever-changing web development trends, the automatic website generation concept has emerged, which generates websites without relying on human interaction [1].

Machine Learning is a broad algorithm-based study that enables computer/device/software to learn based on their own previous experiences. The main techniques used in machine learning are Supervised Learning and Unsupervised Learning. The main difference between supervised learning and unsupervised learning is that supervised method requires both input data and corresponding output labels, whereas unsupervised methods require only input data and no expected output labels.

Clustering is a technique used to group data points. A clustering algorithm can be used to classify each data point into a specific group. Data points in the same group should have similar

Correspondence: T.T. Kaluarachchi (E-mail: thisaranie@spc.cmb.ac.lk)

Received: 16-06-2024 **Revised**: 12-08-2024 **Accepted**: 09-09-2024

T.T. Kaluarachchi, D.M.S. Dissanayake, and M.I.E. Wickramasinghe are from the University of Colombo School of Computing (thisaranie@spc.cmb.ac.lk, sumedhedissanayake@gmail.com, mie@ucsc.cmb.ac.lk)

DOI: https://doi.org/10.4038/icter.v18i2.7301
The 2025 Special Issue contains the full papers of the abstracts published at the 24th ICTer International Conference.

[1] NetCraft February 2023 Web Server Survey: https://news.netcraft.com/archives/category/web-server-survey/.

properties/features. Data items in different groups should have relatively different properties.

Self-Organizing Map (SOM) [2] is an artificial neural network and is one of the most popular neural network models. It belongs to the category of competitive learning and is based on unsupervised learning. Hence, learning does not require human interaction and does not need to know about the characteristics of the input data. The SOM algorithm works on the concept of *Hebbian* Learning. *Hebbian* learning can be described as the strengthening of the synapse between two neurons when the neurons on either side of the synapse have highly correlated outputs. i.e., cells that fire together are wired together. This technique can be used to identify groups of functionally similar entities.

Automating the website design process using machine learning techniques can be a better solution to reduce practical problems and human errors in the design process [2]. Identifying the design features of websites is an essential part of this process. This study focuses on extracting design features of websites using unsupervised machine learning methods. To achieve the study-focused goals, the following research questions (RQs) were formulated.

- ***RQ1***: How can existing website designs be classified?
- ***RQ2:*** What are the basic features that uniquely identify a web design?
- ***RQ3:*** What are the effects of these basic features on overall design and classification?

The remaining of this paper is structured as follows: Section II presents a literature review on related studies. Section III focuses on the design features of the proposed classification system. Section IV presents the implementation. Section V evaluates the experiments conducted. Section VI discusses the limitations. Finally, Section VII brings the research to a conclusion by examining the future directions.

II. LITERATURE REVIEW

In this section, recent research studies related to automatic Graphical User Interface (GUI) generation, image feature extraction, and web design are summarized.

A. Related Studies

Moran et al. [3] proposed a machine learning-based approach to prototyping GUIs in mobile applications. Their approach consisted of three main phases: detection, classification, and assembly. First, computer vision techniques and mock-up metadata are used to detect GUI components from a screen capture of a mobile app. Then, GUI components are classified using Convolutional Neural Networks (CNNs), automated dynamic analysis, and software repository mining. Finally, the assembling phase is performed after creating a hierarchical GUI using the K-nearest-neighbours (KNN) algorithm. It was implemented on Android applications and achieved 91% of accuracy for GUI component classification. The approach is able to accurately detect and classify GUI components in a mock-up artifact, generate a hierarchy of GUI components, and create apps that are visually similar to the originals.

The limitations of this study are (i) supporting only single screens in mobile apps, (ii) tying the KNN hierarchy construction is tied to a specific screen size, and (iii) supporting only a distinct set of stylistic details such as background colour, font colour, and font sizes.

Before writing front-end code for web applications, web developers review UI mock-ups produced by graphic designers and reuse repetitive web elements. Bajammal et al. [4] developed a tool named *VizMod* that generates reusable web components from website mock-up designs. A visual analysis of the mock-up and unsupervised learning of visual cues are used to create reusable web components. First, the mock-up is converted into a set of visual elements, which are then used to identify potential web element instances, such as an image with text as a single UI component. Finally, density-based clustering, an unsupervised learning method, is performed to integrate potential web element instances into GUI components in a parameter-free fashion [5]. *VizMod* was evaluated and compared to real-world mock-ups and expert web developers, resulting in an average precision and recall of 94% and 75%, respectively.

B. Image Feature Extraction

Venetti et al. [6] conducted a study on unsupervised feature learning using SOM. It learns the features of natural images using the SOM neural network to classify natural images. They used the CIFAR-10 dataset, which contains 60,000 small natural images with labels. They showed that the accuracy of the classification can be improved by using appropriate normalization and fine-tuning methods. Furthermore, the study says that using multiple levels of SOM networks can learn more features of visual patterns in images.

The study by Arai K. [7] introduced an image clustering method on density maps derived from SOM. They showed that their SOM-based image clustering method gives better results than k-means clustering. Their method was tested on simulated and real satellite imagery data, and the separability between clusters for the dataset was 16% longer than k-means clustering.

Some research studies are related to image feature classification using improved SOM. Abdelsamea et al. [8] introduced an improved SOM method to classify mammographic images based on texture feature representation. The weight update procedure of their method is different from the normal SOM and the classification depends on the class reliability of the nodes. Their result showed high accuracy compared to classical SOM method and other state-of-art classifiers.

C. Web Designing

Doosti et al. [9] conducted an in- depth study of the history of web design by analyzing automated techniques used to describe the features of websites. A CNN was trained to classify the websites into 26 subject areas and 4 design eras. They developed an approach that uses Hidden Markov Models (HMMs) to represent changes in features over time and generated new website designs after training the CNN with 17,000 snapshots of

websites. The results showed that the new images "look" very similar to the original designs.

One of the most important factors influencing user engagement on a website is its proper design. Garett et al. [10] conducted a literature review to demonstrate how design elements can be used to create an effective website. They reviewed 20 different design elements to determine what influences user engagement. Seven design elements were chosen to influence user engagement on websites.

Website responsiveness has become an essential feature in web development. It allows you to view the website in different ways depending on the screen size. A survey by Mohamed and Ondago [11] demonstrates the state-of-the-art technical aspects of responsive web design. To make the fluid grid concept more practical, they propose a three-category classification. They include Frame-based Solutions (FBS), Support-based Solutions (SBS), and Algorithm-based Solutions (ABS).

III. DESIGN

A. Dataset Preparation

First, an image set of screen captures of websites is created. Next, image processing techniques are applied to pre-process the images in the dataset. The dataset is a 2D matrix with each row representing a pixel value (feature) of an image. This experiment does not focus on colour images. As a result, colour images are converted to grayscale. The number of columns in the dataset equals the number of pixels in an image, while the number of rows equals the number of images in the image set.

B. Conducting Experiments

Experiments are carried out with various image pre-processing methods, including edge detection, morphological operations, image smoothing techniques, and other matrix manipulation methods. The primary goal of these experiments is to identify a better pre-processing method that produces better classification results with SOM. To facilitate these experiments, a MATLAB program with multiple modules was developed.

C. Evaluation

After training the SOM, a U-matrix is generated that shows the *Euclidean* distance between neighbouring neurones on the map. Clusters can be identified by analyzing the U-matrix. The Best Matching Unit (BMU) values for each data item can be represented in the same U-matrix. Then, it is possible to determine which data items belong to each cluster.

To assess the quality of the SOM, we can calculate the Topographic error (TE) which measures how well the topographic structure of the data is preserved on the map. The TE can be computed by finding the first and the second BMUs and measuring their positions. If both are adjacent, we can conclude that the topology is conserved for the given input. Furthermore, by visualizing the u-matrix, we can determine whether the first, second, and third BMUs are adjacent to each other on the map.

Quantification Error (QE) is another quality metric for SOM. The QE is calculated by adding the distances between the input data items and the node weights. To obtain a better model for the dataset, the QE and TE must be minimized during the SOM training process in each experiment.

IV. IMPLEMENTATION

The list of the top-ranked websites was selected from the Amazon Alexa [12]. A bash script was written to capture screen captures of a specified list of websites.Next step was to The web browser used was Firefox 72.0.2. After opening the web page, it waits 1 minute for it to fully load. Then, a system function is invoked to capture the screen and navigate to the next website. Unloaded and partially loaded websites were manually filtered. The lists were repeated several rounds increasing the time interval for each capture. This method captured 1500 images from the top-ranking list.

Some websites were ignored when capturing screen captures because they do not load properly. Websites that could not be found or returned a 404 error were ignored. Websites with critical security warnings were ignored because the web browser blocked such webpages with a warning. Webpages that took a long time to load were ignored. Some of the most popular links were advertisement redirect links rather than actual webpages. These pages were also ignored.

Following the pre-processing stage, the screen captures of the websites were resized to 130x224 while maintaining the line structure. Training images with higher resolution was difficult because it required a significant amount of memory to generate the SOM model, which was difficult to provide with a typical computer used for the research.

V. EXPERIMENTS

A. Clustering Manually Created Wireframe Like Images

The primary goal of this research is to classify websites based on their appearance. A wireframe primarily affects the appearance of a website. A wireframe of a website can be represented with horizontal and vertical lines. To identify how the wireframe can be used to cluster websites, a set of images was used to train a SOM neural network to analyse the clusters. The images were assembled by drawing horizontal and vertical lines.

Two sets of images were generated for training and testing purposes of the SOM. The training image set consists of 40 images (see Fig. 1.), while the testing image set consists of 30 images (see Fig. 2). For the training image set, the first 20 images were created by drawing lines, and the remaining 20 were created by slightly moving each line. The testing image set consists of 20 images from the training image set with minor variations and 10 images with randomly placed horizontal and vertical lines.

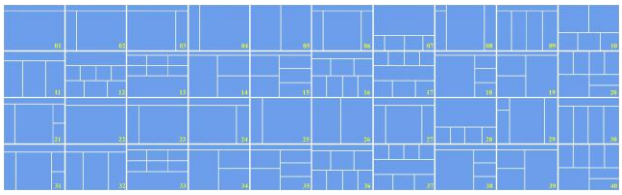

Fig. 1. Training Image Set Created by Drawing Horizontal and Vertical Lines

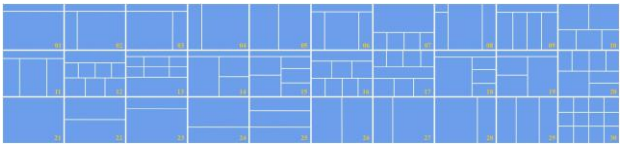

Fig. 2. Testing Image Set Created by Drawing Horizontal and Vertical Lines

1. Experiments for Grayscale and Binarized Images

The dataset for this experiment was created by first converting images to grayscale and then to binary images to highlight lines in the image against the background.

Fig. 3 depicts the U-matrix of the result, while Fig. 4 depicts the U-matrix in three dimensions. The U-matrix identifies several clusters. Fig. 5 depicts the shaded U-matrix, which clearly identifies the clusters. The BMUs for the input dataset are depicted in Fig. 6. The number in the figure represents the index of the input data item.

It examined the first, second, and third BMUs for each input item to ensure that the topology was preserved. Fig. 7 depicts the first three hits. The first, second, and third BMUs are labelled in white, green, and yellow, respectively. Fig. 8 depicts the hits from the training dataset, along with a preview of the input images. Fig. 9 depicts the hits from the testing dataset, along with a preview of the testing images.

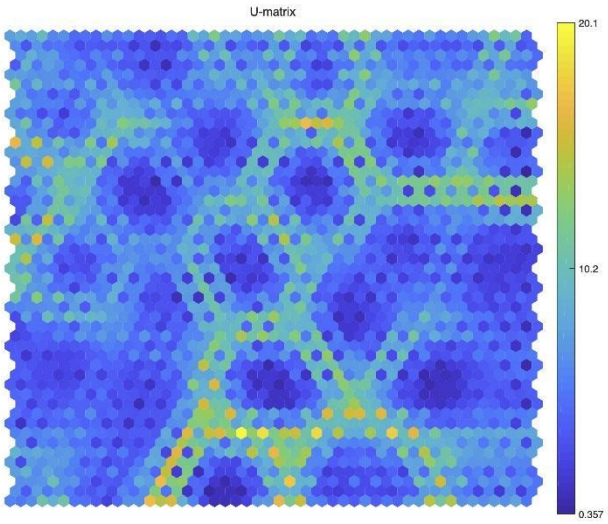

Fig. 3. Experiment U-matrix with Grayscale and Binarized Dataset

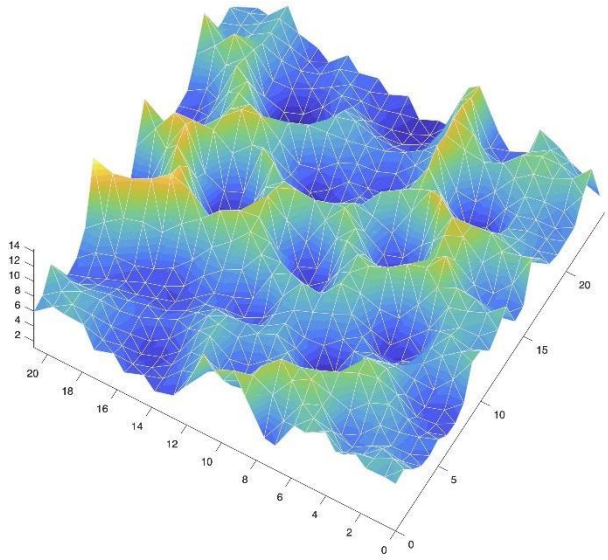

Fig. 4. 3D View of Experiment U-matrix with Grayscale and Binarized Dataset

2. Experiments with Different Initialization Methods

The initialization method of the SOM can influence the final clusters. Several experiments were conducted to visualize the cauterization of different initiation methods. Four initialization methods namely random initialization, zero initialization, incremental initialization, and incremental-normalized initialization were considered. Fig. 10 depicts the U-matrix diagrams with the BMU values of the input data for each initialization method.

Random initialization: This is the default initialization method provided by the SOM toolbox. The values for each feature are uniformly distributed between the minimum and maximum values of the data set.

Zero initialization: Before training, all weight values in the SOM are set to zero.

Incremental initialization: Before training, all weight values in the SOM are set to zero. The value increases from left to right and top to bottom on the map. Each map node has the exact same weight. Fig. 11 depicts a sample map of this initialization method.

Incremental-normalized initialization: Values are distributed similarly to the incremental initialization method, and all values are scaled between 0 and 1. Fig. 12 depicts a sample map of this initialization method.

3. Experiments with the Dilation Filter

In this experiment, the dilation filter was applied after the images were converted to binary format. The reason for dilating images is to make the lines more visible and to fill in any small holes. Fig.13 shows the U-matrix and the results of this experiment.

4. Experiments with Smoothing the Images

This experiment was performed by applying Gaussian filter to smooth the lines in the images. It adds some notification to the image and updates the matrix values of the lines in the images,

which decrease with the distance to the center of the line. A dilation filter was also applied before smoothing the image to make the lines more visible. Fig. 14 depicts the U-matrix and the experimental hits.

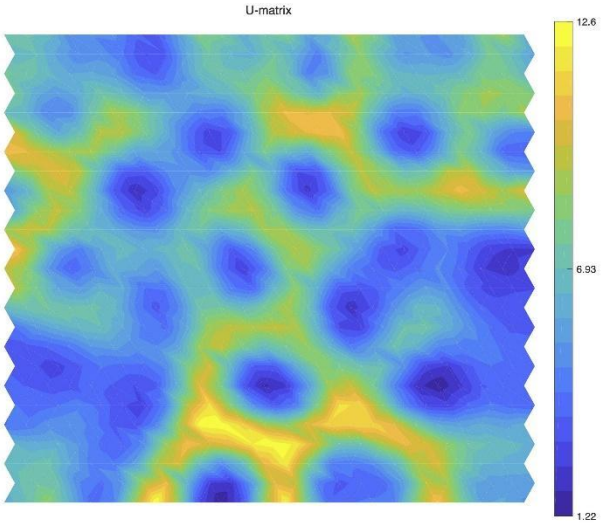

Fig. 5. Shaded U-matrix of Experiment with Grayscale and Binarized Dataset

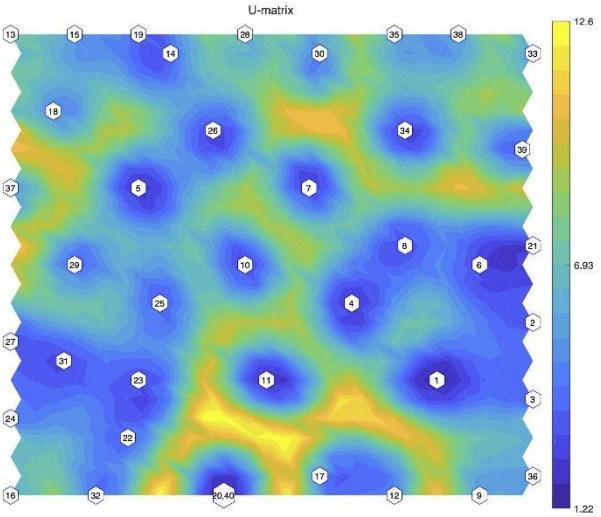

Fig. 6. BMUs of the Training Dataset

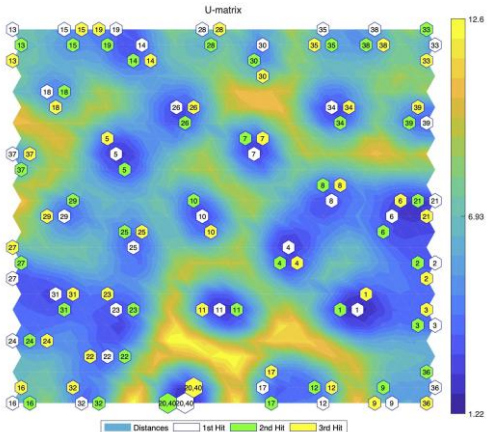

Fig. 7. First, Second and Third BMUs of the Training Dataset

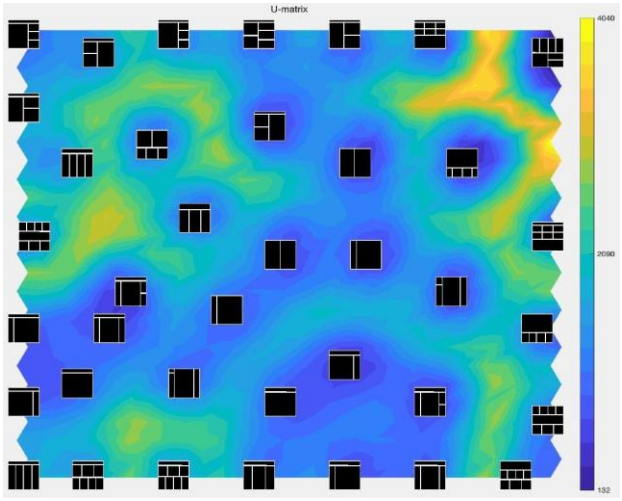

Fig. 8. The BMUs of the Training Dataset with a Preview

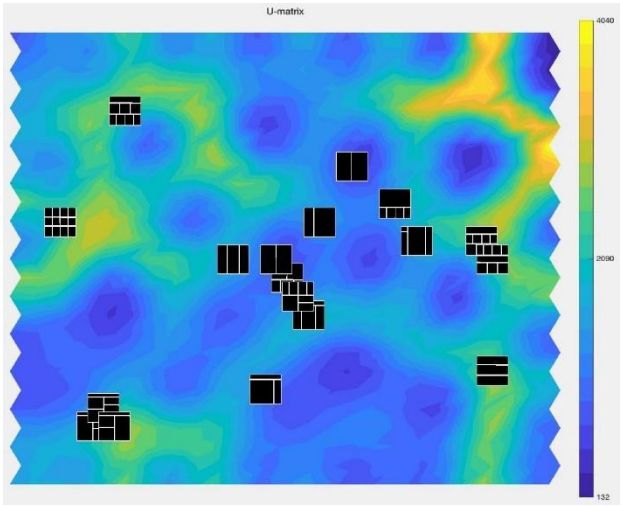

Fig. 9. BMUs of the Testing Dataset with a Preview

B. Clustering Screen Captures of Top 100 Websites

1. Experiments with Grayscale Screen Captures

The dataset was created by converting screen captures of websites into grayscale images. This experiment used no other special image processing methods. Fig. 15 depicts the U-matrix with BMUs of grayscale images. Fig. 16 depicts the U-matrix with previews of BMUs in the dataset.

2. Experiments with Sobel Edge Detection Algorithm

This experiment involved extracting borders from screen captures of websites. Sobel is a general edge detection algorithm. The Sobel method was chosen because it is effective in detecting horizontal and vertical edges. The images were dilated after applying the Sobel filter, and then the Gaussian filter was applied.

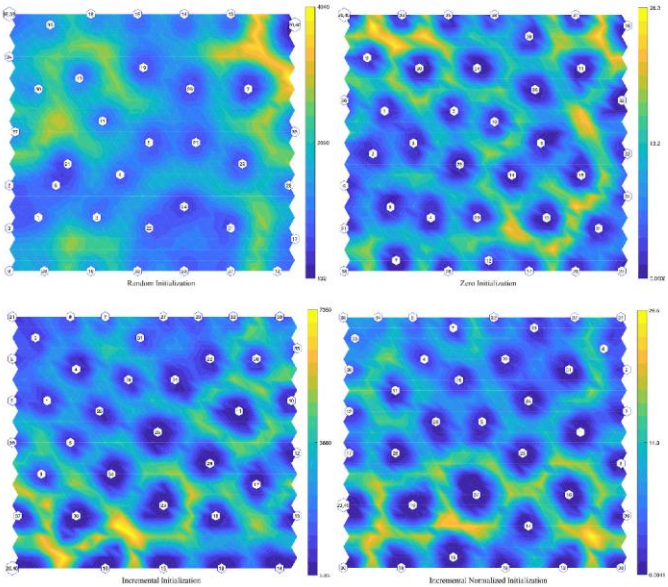

Fig. 10. U-matrix Diagrams with BMUs in the Training Dataset for Random Initialization, Zero Initialization, Incremental Initialization, and Incremental-Normalized Initialization

Fig. 17 shows the U-matrix with BMUs, while Fig. 18 shows the U-matrix with BMU previews from the dataset.

3. *Experiments with Gradient Magnitude and Direction*

In this experiment, the data set is generated by detecting lines in an image and determining the magnitude and direction of the line gradient in the image. After applying the Sobel edge detection filter, the magnitudes of the gradients and the directions of the lines were obtained, and then the horizontal and vertical lines were selected. A small area removal filter was used to reduce noise in small line segments. Using that method, small lines can be removed by specifying a threshold value, as each line should contain the number of pixels in that area. Fig. 19 depicts the U-matrix with BMUs and Fig. 20 depicts the U-matrix with previews of BMUs in the dataset.

0	14	28	43	57	71	85	99	113	128
14	28	43	57	71	85	99	113	128	142
28	43	57	71	85	99	113	128	142	156
43	57	71	85	99	113	128	142	156	170
57	71	85	99	113	128	142	156	170	184
71	85	99	113	128	142	156	170	184	198
85	99	113	128	142	156	170	184	198	213
99	113	128	142	156	170	184	198	212	227
113	128	142	156	170	184	198	212	227	241
128	142	156	170	184	198	213	227	241	255

Fig. 11. Sample Initialization Values of Incremental Initialization

0	0.054902	0.1098	0.16863	0.22353	0.27843	0.33333	0.38824	0.44314	0.50196
0.054902	0.1098	0.16863	0.22353	0.27843	0.33333	0.38824	0.44314	0.50196	0.55686
0.1098	0.16863	0.22353	0.27843	0.33333	0.38824	0.44314	0.50196	0.55686	0.61176
0.16863	0.22353	0.27843	0.33333	0.38824	0.44314	0.50196	0.55686	0.61176	0.66667
0.22353	0.27843	0.33333	0.38824	0.44314	0.50196	0.55686	0.61176	0.66667	0.72157
0.27843	0.33333	0.38824	0.44314	0.50196	0.55686	0.61176	0.66667	0.72157	0.77647
0.33333	0.38824	0.44314	0.50196	0.55686	0.61176	0.66667	0.72157	0.77647	0.83529
0.38824	0.44314	0.50196	0.55686	0.61176	0.66667	0.72157	0.77647	0.83137	0.8902
0.44314	0.50196	0.55686	0.61176	0.66667	0.72157	0.77647	0.83137	0.8902	0.9451
0.50196	0.55686	0.61176	0.66667	0.72157	0.77647	0.83529	0.8902	0.9451	1

Fig. 12. Sample Initialization Values of Incremental-Normalized Initialization

4. *Experiments with Small Regions Removing Method*

When considering the lines in a website wireframe, the average length of the horizontal and vertical lines differs. As a result, it is not reasonable to use the same threshold value in both directions during the small region removal step. This experiment involved creating the dataset by removing the small regions with two different threshold values for horizontal and vertical lines. It produces images by removing noises more precisely. Fig. 21 depicts the U-matrix with previews of BMUs in the dataset.

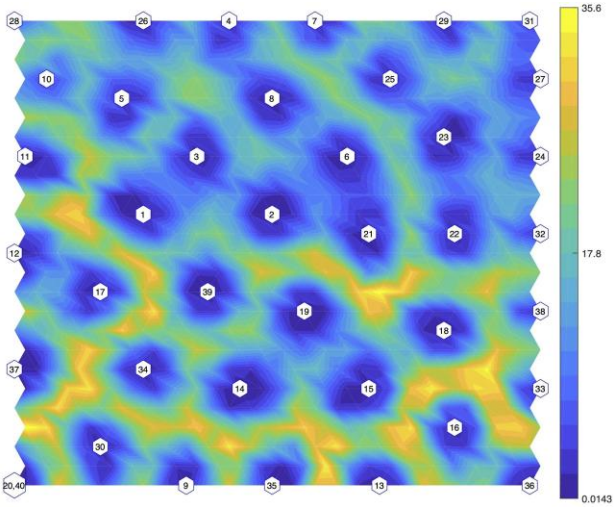

Fig. 13. U-matrix with BMUs in the Training Dataset of Experiment with Dilation

C. *Clustering Screen Captures of Top 1000 Websites*

In this experiment, multiple image processing techniques used in previous experiments were combined to extract lines representing the wireframe of a website. Fig. 22 depicts the U-matrix with previews of BMUs in top thousand images. The distances between the nodes in the neural network are used to

generate a dendrography. Fig. 23 highlights the seven most prominent clusters. Fig. 24 depicts the clustering hierarchy with sample input images and their previews.

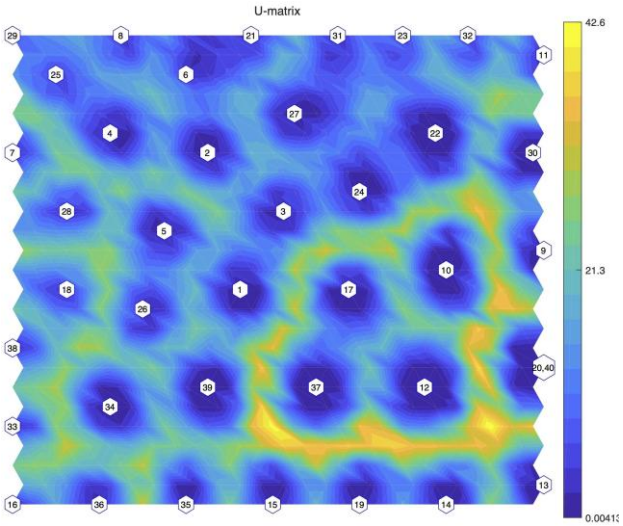

Fig. 14. U-matrix with BMUs in the Training Dataset of Experiment with Smoothing

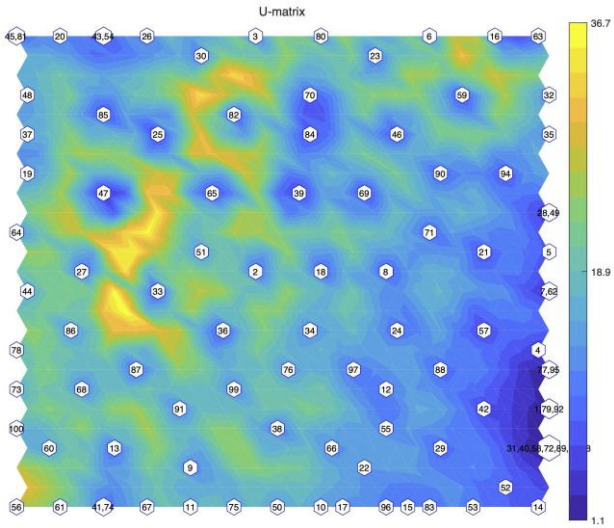

Fig. 15. U-matrix with BMUs in the Training Dataset of Experiment with Grayscale Screen Captures

Fig. 25 shows that each cluster contains similar types of websites. Websites with a home page, a simple search box, and a button are clustered in Cluster 5. All URLs with webpages similar to the Google search page are clustered there. Cluster 3 contains websites similar to e-commerce websites that contain a banner and sample product images. Websites with more horizontal lines are clustered in Cluster 7. Cluster 6 contains websites that display thumbnail images on the home page, including amazon.com.

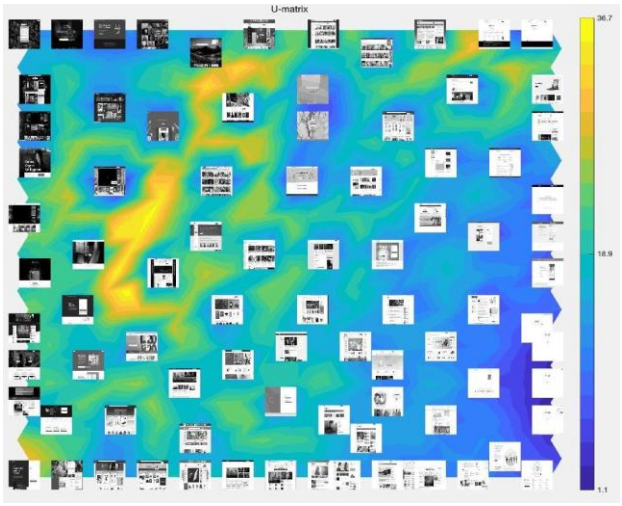

Fig. 16. U-matrix with Previews of BMUs in the Experiment with Grayscale Screen Captures

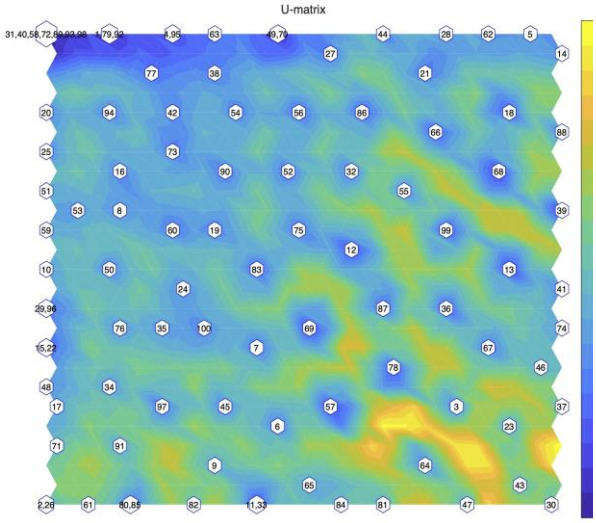

Fig. 17. U-matrix with BMUs in the Training Dataset of Experiment with Sobel

VI. LIMITATIONS

Websites that do not properly represent are removed from the dataset since they were incapable of capturing the intended design accurately. Only the lines that are part of the wireframe of a website are considered in this research. The model is only intended for the first thousand of websites on the list of rankings. Since building of the model required a large amount of memory, which was difficult to render on a standard computer, the screen captures of the websites are resized to a lower resolution after the reprocessing stage.

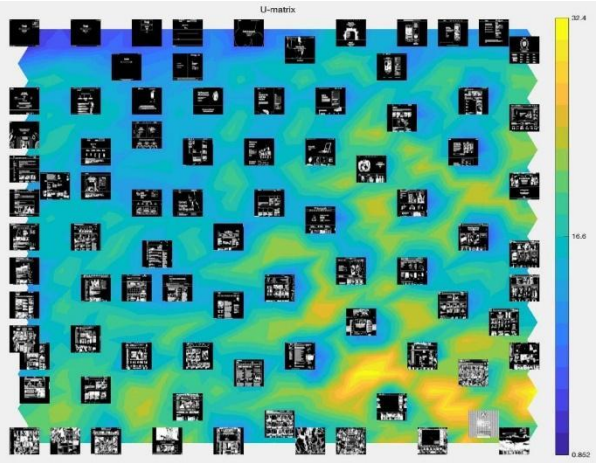

Fig. 18. U-matrix with Previews of BMUs in the Experiment with Sobel Filter

VII. CONCLUSION AND FUTURE WORK

This research paper presents some experiments conducted to develop a classification system for websites available on the Internet based on their salient design features. According to the findings, the wireframe lines are a fundamental element that aids in uniquely identifying a website design. Similarly, SOM is an effective approach for cluster detection in classification.

Lines extracted from screen captures are one aspect considered in this study. Since a large number of features can be extracted from such screen captures, the proposed method can be used to discover additional design features in websites. Furthermore, a more refined classification can be achieved by incorporating more pre-processing techniques and adjusting the parameters.

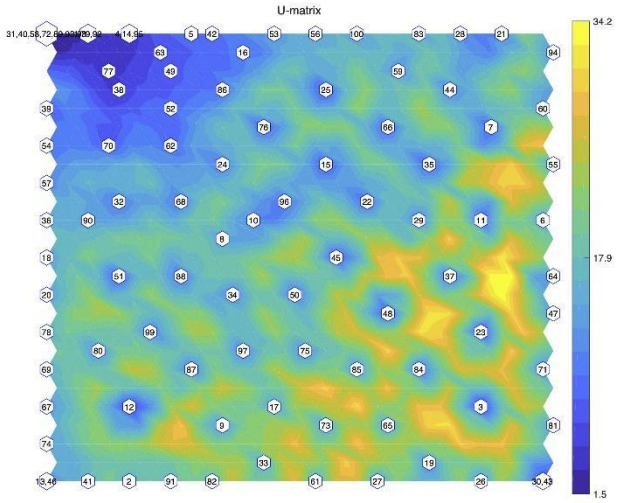

Fig. 19. U-matrix with BMUs in the Training Dataset of Experiment with Gradient Magnitude and Direction

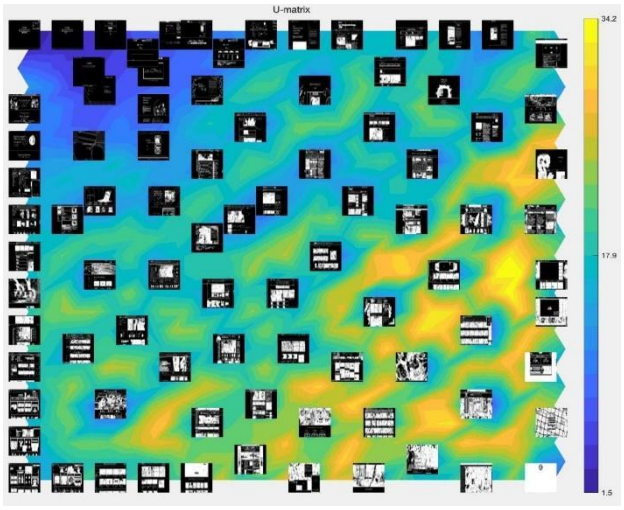

Fig. 20. U-matrix with Previews of BMUs in the Experiment with Gradient Magnitude and Direction

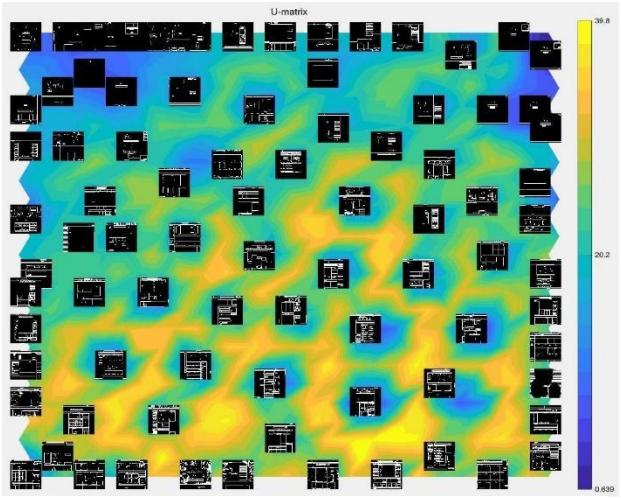

Fig. 21. U-matrix with Previews of BMUs in the Experiment with Small Regions Removing Method

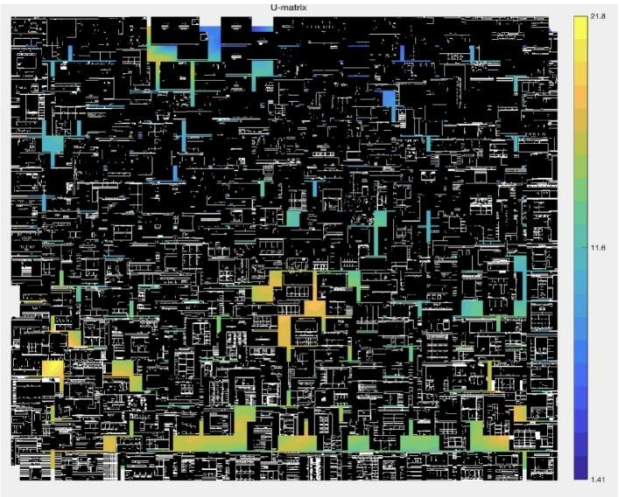

Fig. 22. U-matrix with Previews of BMUs in the Experiment with Small Regions Remving Method

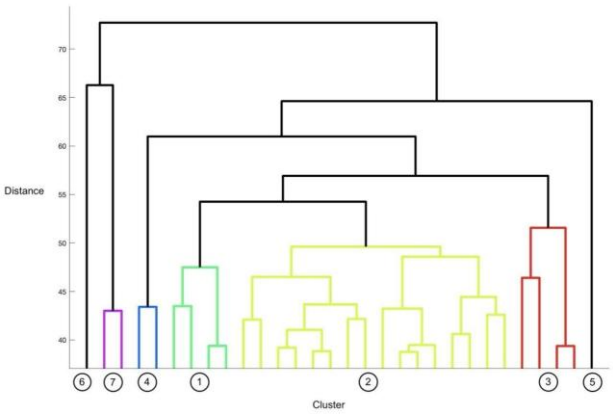

Fig. 23. Dendrography of the Clustering Websites

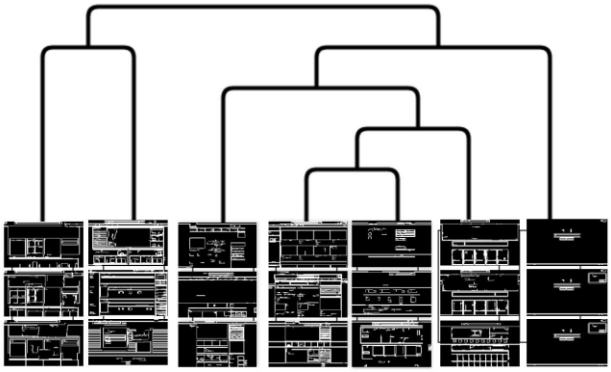

Fig. 24. Dendrography with Sample Inputs in Clusters

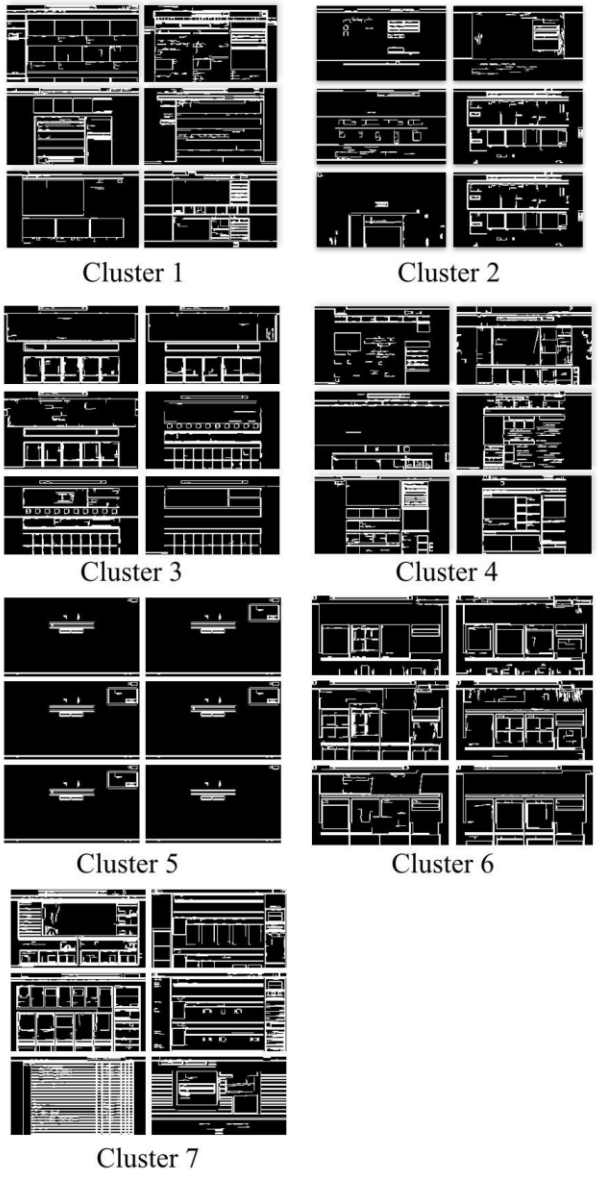

Fig. 25. Seven Clusters of Sample Input Images with Previews

REFERENCES

[1] T. Kaluarachchi, M. Wickramasinghe, *A systematic literature review on automatic website generation*, J. Comput. Languages 75 (2023), 101202, https://doi.org/10.1016/j.cola.2023.101202 ISSN 2590-1184.

[2] T. Kohonen, *Self-organized formation of topologically correct feature maps*, *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982. [Online]. Available: http://link.springer.com/10.1007/BF00337288

[3] K. Moran, C. Bernal-Cardenas, M. Curcio, R. Bonett, and D. Poshyvanyk, *Machine Learning-Based Prototyping of Graphical User Interfaces for Mobile Apps*, *arXiv:1802.02312 [cs]*, Feb. 2018, arXiv: 1802.02312. [Online]. Available: http://arxiv.org/abs/1802.02312

[4] M. Bajammal, D. Mazinanian, and A. Mesbah, *Generating reusable web components from mock-ups*, in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering - ASE 2018*. Montpellier, France: ACM Press, 2018, pp. 601–611. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3238147.3238194

R. J. G. B. Campello, D. Moulavi, and J. Sander, *Density-Based Clustering Based on Hierarchical Density Estimates*, in *Advances in*

[5] *Knowledge Discovery and Data Mining*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, vol. 7819, pp. 160–172.

[6] M. Venetti, I. Gallo, and A. Nodari, *Unsupervised Feature Learning using Self-organizing Maps*, In *Proceedings of the International Conference on Computer Vision Theory and Applications*. Barcelona, Spain: SciTePress - Science and and Technology Publications, 2013, pp. 596– 601.

[7] K. Arai, *Image Clustering Method Based on Density Maps Derived from Self-Organizing Mapping: SOM*, *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 7, 2012.

[8] M. Abdelsamea, M. H. Mohamed, and M. Bamatraf, *An Effective Image Feature Classification using an improved SOM*, *arXiv:1501.01723 [cs]*, Jan. 2015, arXiv: 1501.01723. [Online]. Available: http://arxiv.org/abs/1501.01723

[9] B. Doosti, D. J. Crandall, and N. M. Su, *A Deep Study into the History of Web Design*, in *Proceedings of the 2017 ACM on Web Science Conference - WebSci '17*. Troy, New York, USA: ACM Press, 2017, pp. 329–338. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3091478.3091503

[10] Garett, R., Chiu, J., Zhang, L., & Young, S. D. (2016). *A literature review: website design and user engagement*. Online journal of communication and media technologies, 6(3), 1.

[11] A. A. Mohamed and C. Ondago, *Responsive Web Design in Fluid Grid Concept Literature Survey*, p. 9.

[12] Top 1000 most visited websites in the World by Ahrefs organic search traffic estimates. January 2024, Available: https://ahrefs.com/top