

Cloud Application for the Generation of Static Websites Through the Recognition of Wireframes using Artificial Intelligence

Cesar Gutierrez¹, Rodrigo Lara¹ and Daniel Subauste¹

¹ Universidad Peruana de Ciencias Aplicadas, Prolongación Primavera 2390, Lima, 15023, Perú

Abstract

Nowadays, companies need to have a presence on the Internet to offer their products or services. This involves high costs and long lead times, as well as specialized personnel in web development. Therefore, we propose the implementation of a solution that allows the generation of static web pages from hand-drawn drawings. This solution allows users to automate the process of creating HTML and CSS code, reducing time and cost. In this research, a model based on the standard nomenclature of the basic wireframes of a web page was trained and then ordered using a tree-based algorithm. The results show a reduction in the time and cost invested by developers in the wireframe to source code transformation process. Also, the acceptance of users who have no knowledge of HTML and CSS is evident, as they find the tool a simple way to generate web pages. to generate web pages.

Keywords

Computer Vision, Wireframe, Web Page, N-ary tree

1. Introduction

In recent years, companies have increased the need to have a website, due of covid-19 pandemic where they have had to develop or update their own web pages to have more presence on the Internet [1]. However, for web development, it is necessary to invest time and money, as well as in pre-development designs.

These previous designs are represented through a wireframe or mockup. The former is a low-fidelity version of the product that is hand-drawn or made through software, while the latter is a high-fidelity design that includes colors, images, and text and consumes more resources to create than the first [2]. The opportunity to have a tool that allows the automatic generation of static web pages from a handmade design will allow more users to have in less time and at a lower cost a website that allows them to have a presence on the Internet.

There are currently proposals to solve this problem, but they are limited, that is, they present as functionality to enter a wireframe image and generate HTML code. In our research we propose additional functionalities that allow the user to develop a customized web page.

In general, the main contributions of this document are summarized below:

- A web application was built to allow the generation of web pages from pattern recognition in wireframes. The proposed solution improves the transformation process from wireframes to user interface since it reduces time and costs.
- A model was trained using Azure Custom Vision, which allows the recognition of previously standardized components of a web page.

CISETC 2021: International Congress on Educational and Technology in Sciences, November 16-18, 2021, Chiclayo, Peru
EMAIL: u201611510@upc.edu.pe (C. Gutierrez); u201614682@upc.edu.pe (R. Lara); daniel.subauste@upc.pe (D. Subauste)
ORCID: 0000-0002-5126-882X (C. Gutierrez); 0000-0001-8315-5275 (R. Lara); 0000-0003-1131-1384 (D. Subauste)



© 2021 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

- A tree-based algorithm was built for the ordering of components by rows and columns using the Bootstrap grid.
- A series of experiments were conducted on a group of users to evaluate the performance of our proposal. The transformation results are more accurate compared to other solutions.

This paper is organized in 7 sections. In section 2, the context is developed. Section 3 describes the work related to our proposal. Section 4 presents the web solution (Wire2web), detailing the model training guidelines, the implementation process and the algorithms used. Section 5 explains the validation of the proposed web solution. Finally, section 6 presents the conclusions.

2. Context

2.1. Artificial Intelligence (AI)

The term artificial intelligence (AI) refers to any human-like intelligence exhibited by a computer, robot, or other machine [3]. The main research fields of AI include expert system, machine learning, pattern recognition, natural language understanding, and so on. In addition, there are application fields of AI, such as virtual reality, machine translation, computer vision, etc. The latter being the one we will use in our solution [4].

2.2. Computer Vision

Computer vision is one of the branches of computer science that has experienced a remarkable growth in recent years, both in face and object detection. It also presents a sequence of stages in which the image is processed at different levels, in addition to taking actions or making recommendations based on that information [5]. Computer vision needs to be trained with a large amount of data until it identifies distinctions and finally recognizes images.

2.3. Web Page

Web pages are documents that are written in HTML and can be stored on a computer or on a remote web server [6]. These are divided into two types. Firstly, static web pages have the main functionality of being informative and are stored as simple files that are then served by a web server [7]. Secondly, dynamic web pages allow a web page to communicate with a server and change its content without visiting a new page or updating the previous one and offer greater interactivity with visiting users [8].

2.4. Wireframe

A wireframe is a static, low-fidelity representation of a final product, and is made up of several visual components, represented in a simplified way, that aim to show the location of each of them together [9].

2.5. N-ary Tree

An n-ary tree, of height h , is a tree whose nodes that are at a maximum distance of $h - 1$ from the root, have n child nodes, these children are known as leaves since there are no nodes below them [10].

3. Related Works

In this section we examine the main research related to our project. First, with respect to automating the process of transforming hand-drawn drawings to source code, we found articles proposing software solutions.

One research is the one shown in sketch2code [11] where they develop a system capable of generating web pages from hand-drawn sketches. This research proposes the following process: dataset development, model training and application implementation. Such proposal generates a significant impact on our research since we perform a similar process.

Another research is the one shown in Pix2code [12], which is a system based on convolutional and recurrent neural networks that allows code generation from a GUI screenshot as input. That research proposes a model, which we took as a reference to realize the wireframe standards for our project. In addition, the application implemented by [12] is named Uizard, which presents several functionalities that we take as a reference, such as uploading a wireframe, editing the generated view, and relating views.






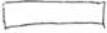






4. Proposal

This section will show the solution development process. For this purpose, we propose five processes divided into two stages. In the first stage, we explain the process for the construction of our dataset and the model training using computer vision techniques. In the second stage, the implementation of the web application is explained, as well as the algorithm used and the functionalities that the application will present. The 5 processes will be explained below.

4.1. Data Set

First, we evaluated the composition of the wireframes, where we obtained as a result that these drawings are composed of components. Second, we performed component standardization, where we investigated about the most used components in web pages. These standards were obtained as a reference from Justinmind, Uizard and Scketch2code, which are platforms where wireframes are designed or used. Finally, 12 components were obtained, which were represented in handmade drawings or also known as wireframes.

Table 1
Standard wireframe components

Components	Hand drawing	Components	Hand drawing	Components	Hand drawing
Circle image		Square image		Text	
Text Area		Input Number		Input	
Button		Combo box		Radio Button off	
Radio Button on		Checkbox on		Checkbox off	

4.2. Model Training

To develop the model, training was carried out to recognize the components of the wireframes. To carry out this process, a cloud service that uses computer vision to detect objects was used. In addition,

the processing capacity and the price-capacity ratio were taken as variables to choose this service. Then we concluded that Azure Custom Vision will be used since it meets the requirements that our project needs.

To train the model, two iterations were carried out. In the first one, thirty wireframe images were added to the dataset. While, in the second iteration, seventy additional wireframe drawings were added with different colors and ways of capturing images, which allowed our model to be more accurate in detecting components.

At the end of each iteration, we obtained indicators, such as precision, recall and mAP. Accuracy indicates the fraction of identified images that were correct, Recall the fraction of real images that were correctly recognized and finally mean Average Precision (mAP) the overall accuracy of the object detector in finding a component. The results of each iteration show that with the second iteration all three indicators improved, making the model more stable.

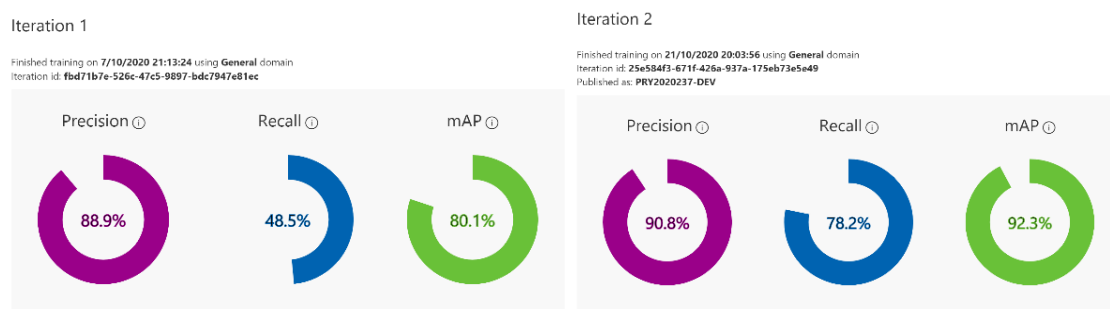


Figure 1. Results of the two iterations. Source: Azure Custom Vision.

4.3. Model consultation

To consult the trained model, you must have a photo of a hand-drawn wireframe. This must be uploaded to the application. It is then converted to base64 and sent to the Azure Custom Vision service for analysis. The request returns a JSON with a structure defined by each component. Each one contains the probability, position, tag name, width, and height. Finally, each component was used by a tree-based algorithm to sort them into rows and columns and have a better distribution of these.

Result in JSON format of the Azure Custom Vision of the Text component.

```
{
  "probability": 0.875464261,
  "tagId": "1932c95f-ed4a-4675-bde4-c2457e1389e6",
  "tagName": "Text",
  "boundingBox": {
    "left": 0.453497916,
    "top": 0,
    "width": 0.2523211,
    "height": 0.8738168
  }
}
```

4.4. Algorithm

This tree-based algorithm was developed to display the distribution of detected wireframe components in rows and columns for better visualization by the user. This development was divided into two processes.

4.4.1. Components sorting by rows

First, the algorithm detects the components from top to bottom. This comparison is made with respect to the "top" property provided by the Azure Custom Vision service. After that, it checks if any component is inside its section (red lines) "Fig. 2". Also, it adds a margin (yellow lines) "Fig. 2", to detect components that are within the margins and determine whether they belong to the same section.

In addition, if the height of the component found is greater than the components that are within the same section, this will be the element of comparison. If there are no more elements to compare within the section, a row is assigned, and the elements of the next lower sections are analyzed.

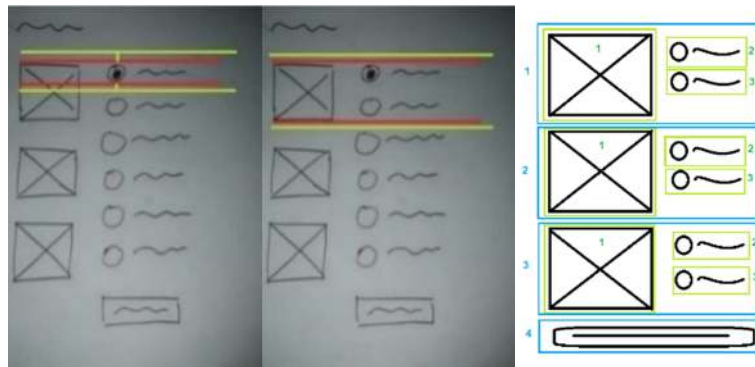


Figure 2. Analysis and representation of the wireframe image by rows.

Second, once all the components have been detected and assigned to a specific row, they are added to the tree, that is, each node is the row, and the child nodes (leaves) are the detected components. Finally, the tree generated in the first process has a hierarchical structure and is level 3.

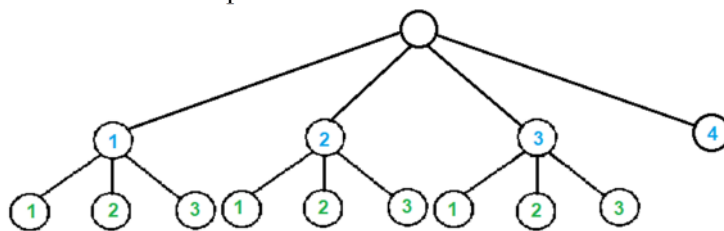


Figure 3. Tree generated from a wireframe by rows.

4.4.2. Components sorting by columns

When the tree reached level three in height. In the third level a comparison is made between the child nodes with the same parent. This comparison is made from left to right with the "Left" field which is obtained by the Azure Custom Vision service. For example, for the first row: if node 2 has node 3 within its range, then they are joined in the same column.

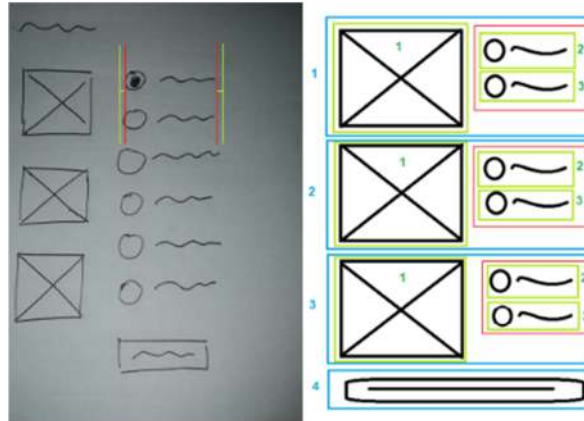


Figure 4. Analysis and representation of the original image by rows and columns.

For the example shown, once all the columns within each row were detected, the tree must be in 4 levels as follows:

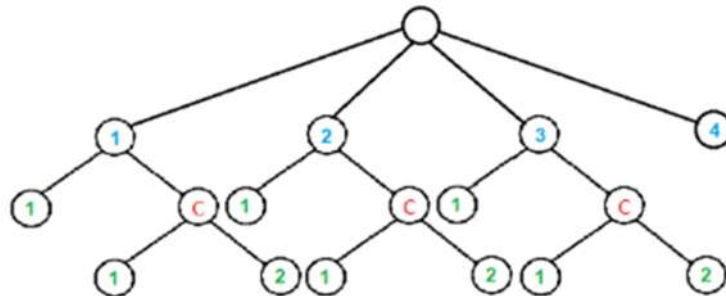


Figure 5. Tree generated from a wireframe - Second stage.

Finally, this tree is saved in the database in JSON format, to be used later in other functionalities of the developed application.

4.5. Results and Functionalities

The result is the source code generated in HTML and CSS which uses the Bootstrap grid to display the rows and columns in an orderly fashion. On the other hand, the developed application allows grouping these views within a project, as well as making changes to each view, either by editing each attribute, adding new elements to the generated view, choosing a theme for the entire project, and allowing the download of the project in a .zip file.

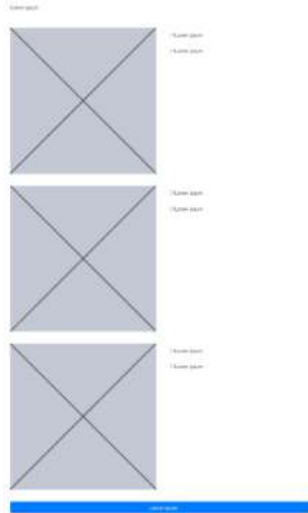


Figure 6. HTML generated from wireframe.

5. Validation

In this section, we will detail the results obtained by testing the application and the feedback obtained through the questions asked to the users. A total of 20 users were interviewed.

First, a detailed explanation of the project was given to each user. Then, a URL of the deployed web application was sent. Then, each user logged in through a browser using their PC and went through the entire flow, from creating a profile to downloading one or more projects.

Finally, users had to answer a questionnaire based on their experience with the application. A validation was also performed to measure the time and cost-effectiveness of using the application versus traditional development by a programmer. To do this, three developers implemented a static two-view web page. Then, these same developers made the same web page using the proposed application. Having as initial design the same wireframes.

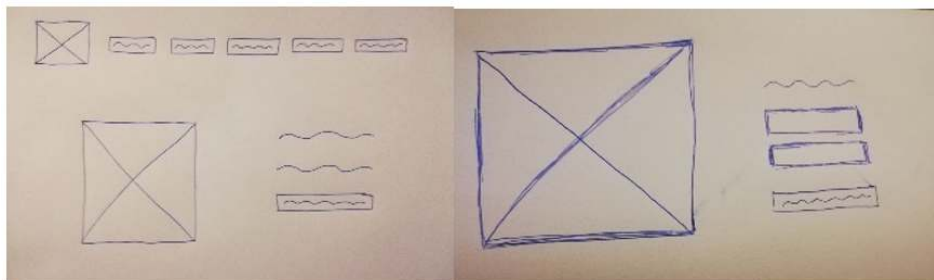


Figure 7. Wireframes used for validation.

Table 2 and 3 below show the results obtained:

Table 2.

Times obtained from validation.

Users	Traditional method	Using the proposal	Time saved
Developer 1	50,25 min	11,17 min	39,08 min – 77,77%
Developer 2	42,57 min	9,72 min	32,85 min – 77,18%
Developer 3	46,05 min	13,34 min	32,71 min – 71,03%

Table 3.

Costs obtained from validation

Users	Traditional method	Using the proposal	Cost saved
Developer 1	3.81 USD	0.85 USD	2.96 USD – 77,77%
Developer 2	3.23 USD	0.74 USD	2.49 USD – 77,18%
Developer 3	3.49 USD	1.01 USD	2.48 USD – 71,03%

The results show that for the development of a static web page, the proposal reduces the implementation time and cost for a developer by 70 to 80 percent.

Figure 8 shows the web page made manually using HTML and Bootstrap.

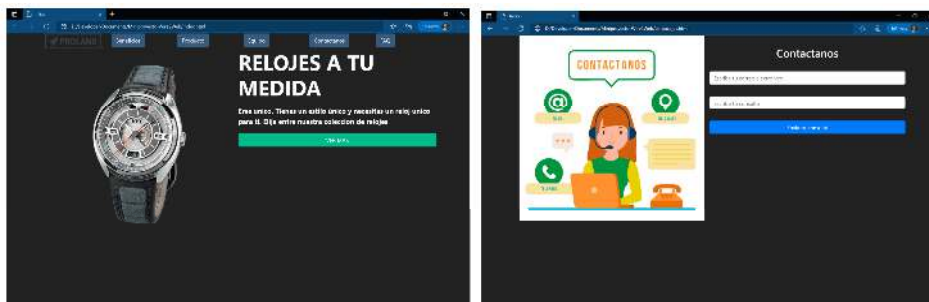


Figure 8. Static web page made by the developer 2 with the traditional method.

On the other hand, Figure 9 shows a web page using the proposed application.

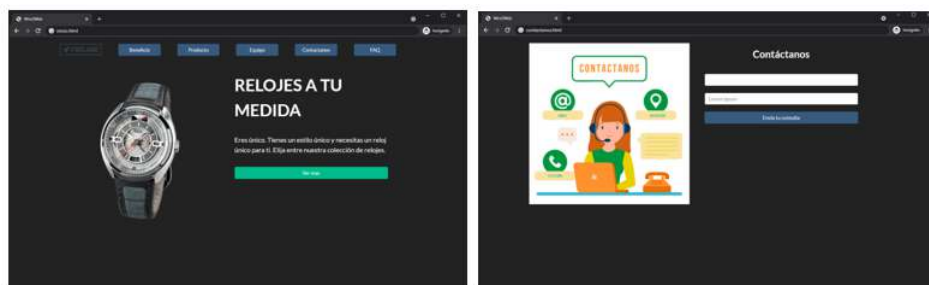


Figure 9. Static web page made by developer 2 using the proposed application.

6. Conclusions

After training the model, it can be concluded that for adequate training it is recommended to use at least fifty images per label. Because, in tests performed, the first iteration had a total of 30 images per label and as a result the model still did not detect some objects. Then, a second iteration was performed, and 70 more images were added, having a minimum of 70 images per tag and a maximum of 100 images per tag, where the result was favorable, since it improved the accuracy of recognition of web components.

Secondly, after performing the corresponding validations and the different tests, it was concluded that the detection of web page components, the transformation of a wireframe to HTML and CSS code, as well as the sorting by rows and columns using the proposed tree-based algorithm complied with the established requirements.

On the other hand, with respect to the validations with the group of users through the software tests and the survey conducted, it can be concluded that the solution, for 89% of the surveyed developers reduces the development time, having as a result that the average response was 4.45 within a response

range of 1 to 5. Also, it can be concluded for 83% of the interviewed developers, our solution allows them to reduce the implementation costs, having that the average response is 4.15 in a range of 1 to 5.

Finally, for future work it could be extended to more complex components like cards, navbars, sliders and iconography. As well as the recognition of mobile device components and code generation. In addition, the project allows the extension of the use of frontend development frameworks such as: Vuejs, React or Angular.

7. References

- [1] EL PAIS, "Casi la mitad de las empresas no tenía web antes de la pandemia, según un estudio | Pyme | Cinco Días," *Cinco Días*, 2021.
- [2] Justinmind, "Wireframes Vs Mockups: what's the best? - Justinmind," 3 2019.
- [3] IBM, Acelere su camino hacia la IA - Argentina | IBM, 2020.
- [4] X. Fu, The Application of Artificial Intelligence Technology in College Physical Education, Institute of Electrical and Electronics Engineers Inc., 2020, pp. 263-266. doi: 10.1109/ICBAIE49996.2020.00062.
- [5] J. Sigut, M. Castro, R. Arnay and M. Sigut, OpenCV Basics: A Mobile Application to Support the Teaching of Computer Vision Concepts, vol. 63, Institute of Electrical and Electronics Engineers Inc., 2020, pp. 328-335. doi: 10.1109/TE.2020.2993013.
- [6] MDN, HTML: básico Aprende sobre desarrollo web, 2020.
- [7] A. Anagnostopoulos, A. Z. Broder, E. Gabrilovich, V. Josifovski and L. Riedel, Web page summarization for just-in-time contextual advertising, vol. 3, 2011, pp. 1 - 32. doi: 10.1145/2036264.2036278.
- [8] A. Brown, C. Jay and S. Harper, "Tailored presentation of dynamic web content for audio browsers," *International Journal of Human Computer Studies*, vol. 70, no. 3, pp. 179-196. doi: 10.1016/j.ijhcs.2011.11.001, 3 2012.
- [9] J. Chen, C. Chen, Z. Xing, X. Xia, L. Zhu, J. Grundy and J. Wang, "Wireframe-based UI Design Search through Image Autoencoder," *ACM Transactions on Software Engineering and Methodology*, vol. 29, no. 3, pp. 1-33. doi: 10.1145/3391613, 7 2020.
- [10] F. Duque, A. Roldán-Correa and L. A. Valencia, "Accessibility Percolation with Crossing Valleys on n-ary Trees," *Journal of Statistical Physics*, vol. 174, no. 5, pp. 1027-1037. doi: 10.1007/s10955-019-02223-5, 3 2019.
- [11] A. Robinson, "Sketch2code: Generating a website from a paper mockup," 5 2019.
- [12] T. Beltramelli, "pix2code: Generating Code from a Graphical User Interface Screenshot," *EICS '18: Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, pp. 1-6. doi: 10.1145/3220134.3220135, 5 2017.