International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 07 | July - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

GENERATION OF HTML CODE AUTOMATICALLY USING MOCK-UP IMAGES

[1] Raksha K , [2] Yashodha P G

[1] Student, Department of Master of Computer Applications, BIET, Davangere

[2] Assistant Professor, Department of MCA, BIET, Davangere

ABSTRACT

The design cycle for a website begins with making mock-ups for individual web pages, either by hand or with graphic design and specialist mock-up generation tools. Software programmers transform the prototype into structured HTML or comparable markup code. This technique is typically repeated multiple times until the desired template is created. Our goal in this study is to automate the process of generating code from hand-drawn mock-ups. Hand-drawn mock-ups are processed using computer vision techniques, and the proposed system is then implemented using deep learning approaches. Our system achieves 96% method and 73% validation accuracy. Index Terms: convolutional neural networks, deep learning, automatic code generation, and HTML.

INTRODUCTION

The value of Internet websites has grown significantly as a result of technological advancements. Nowadays, websites represent the faces of states, institutions, communities, and individuals. Websites are offered in practically every sector, including knowledge, social work, games, and training. Companies create websites for financial reasons, such as product promotion or advertising. On the other hand,

Official institutions seek to provide more efficient services. A "web page" is the section of a website that interacts with the user and is located on the front end. Providing a visually appealing, user-friendly, and functionally rich page is crucial. However, creating web pages that effectively meet these requirements is a time-consuming task. Graphic designers, software professionals, end users, business authorities, and others working in a variety of fields must all collaborate to create web sites.Typically, the process begins with the mock-up design of the user interface by graphic designers or mock-up artists, either on paper or using graphic editing software, in accordance with the needs of the institution. These drafts serve as the foundation for software specialists to create code for web sites. The generated web pages may vary based on feedback from end users. Their procedure consists of numerous repetitive tasks. Rewriting code for components with similar functions and page structures that change over time becomes tiresome. This highlights the need for more efficient methods in web page design.

PROBLEM STATEMENT

The traditional design cycle for a website involves manually creating mock-ups, which are then converted into structured HTML code by software programmers. This procedure is iterative and time-consuming, frequently necessitating numerous adjustments until the desired template is reached. To speed up this process, the goal of this research is to create an automated system that generates HTML code straight from hand-drawn mock-ups using computer vision and deep learning techniques.

Specifically, the issue might be phrased as follows: Given hand-drawn mock-ups of individual web pages, the goal is to create a system that can automatically recognize items inside the mock-ups (such as text fields, buttons, photos, and so on), determine their layout and structure, and output appropriate HTML code.

EXISTING SYSTEM

In this study, an algorithm was designed to automatically produce HTML code for a hand-drawn mock-up of a web page. The goal is to recognize the components made in the mock-up drawing and encode them in accordance with the web page structure. The suggested approach is trained and verified using a publicly available dataset of hand-drawn images of websites taken from Microsoft AI Labs' Github page [1]. The dataset's photos are processed using computer vision techniques, and the data is trained using a deep neural network model based on convolutional neural networks. Afterward, a structured HTML code is obtained. Our model has 96% method accuracy and 73% validation accuracy.
The remainder of the study is organized as follows: Section II presents similar findings from the literature. Sections III and IV detail the dataset and techniques. In Section V, the results and findings were presented. Section VI, which is the concluding section, contains evaluations.

DEMERITS OF THE EXITING SYSTEM

• Reliance on Hand-Drawn Mock-ups: The system uses hand-drawn mock-ups as input, which may not correctly reflect the final design ideas or introduce discrepancies.
• Limited Generalization: The model's performance may suffer when applied to hand-drawn mock-ups with various styles or complexities that were not effectively captured in the training dataset.
• Interpretation Issues: Interpreting hand-drawn mock-ups accurately can be difficult, especially if the sketches are vague or confusing, which might lead to mistakes in the resultant HTML code.
• Complexity of Web Page Structure: Generating HTML code for complicated web page layouts with sophisticated designs and dynamic parts may provide difficulties for the algorithm, perhaps resulting to errors or partial translations.

• Lack of User-Friendliness: Although the system seeks to automate code generation, it may still require technical skills to fine-tune or interpret the results, restricting its usability by non-technical users.

PROPOSED MODEL

This method, known as REMAUI, discovers the components of a mobile application's user interface, such as buttons, textboxes, and images, and generates code for them from screenshots of the program window or conceptual drawings. In their work, which was the first to provide code translation from screen images or drawings for mobile platforms, computer vision and optical character recognition algorithms were utilized. The REMAUI approach works successfully, however it does not handle cross-page transitions or page animations. In [3], the authors created the P2A algorithm to address the shortcomings of the REMAUI method.
The pix2code approach, developed by the authors of [4], intends to transform a web page's graphical interface to structured code using deep learning using convolution and recurrent neural networks. The approach has been tested on Android, iOS, and mobile platforms, and the results have been positive. In [5], the algorithm Redraw takes mock-ups of mobile application screens and generates structured XML code for them. In the first stage of implementation, computer vision techniques are used to recognize certain GUI components. The second step is to classify the identified components according to their function, such as toggle-button, text-area, and so forth.At this stage, deep convolutional neural

networks are deployed. In the final stage, the XML code is generated by mixing it with the KNN algorithm, which follows the web programming hierarchy. Nowadays, open source code libraries like Github [6] are widely used for sharing code and apps.

It is usual practice to look into this repository and reuse code when starting or enhancing software projects. These libraries' shared codes decrease the need for various persons to write the same code repeatedly. In [7], the authors employ a search software called SUISE, which allows users to create a graphical interface using simple graphics and keywords. This interface is then compared to existing libraries to find similar interfaces. These interfaces are converted into operable codes and returned to the end user, who can then choose the best interface.
Microsoft recently developed a technology that converts hand-drawn mock-ups of simple web sites into HTML code [1]. There is no literature that describes their work, but they have shared their code and dataset online. In this project, we use some of the photographs from this dataset.

MERITS OF THE PROPOSED SYSTEM

• Automation: The method automates the conversion of graphical interfaces or mock-ups into structured code, decreasing the amount of manual work required to write mobile applications or web pages.
• Efficiency: The algorithms use computer vision and deep learning approaches to effectively recognize and classify graphical components, resulting in efficient code creation.
• Platform Compatibility: The algorithms support many platforms, including Android, iOS, and the web, enabling code adaptability for a variety of application scenarios.

• Accuracy: The algorithms use convolutional neural networks and other machine learning approaches to detect and categorize GUI components with high accuracy, resulting in dependable code production.

• Code Reusability: The solutions use existing code repositories and libraries to promote code reuse and reduce redundancy in software development initiatives.

• User-Friendly Interface: Some systems include user-friendly interfaces that allow users to enter basic drawings or phrases to search for and generate executable code, making the process accessible to non-experts.

LITERATURE SURVEY

Automatically generate HTML code from sketches for mobile applications.

Sun Y., Zhang S., Xiong Y., Zhang X., and GuoB. This study describes an automatic HTML code generation strategy for mobile applications based on sketching. The REMAUI algorithm uses computer vision and optical character recognition (OCR) techniques to identify UI components such as buttons and text fields in screenshots or conceptual drawings of mobile app interfaces. However, REMAUI doesn't support cross-page transitions or animations. To solve this, the authors suggest the P2A method, which improves REMAUI by including extra features for better code generation.

pix2code generates code from a graphical userinterfacescreenshot.Lopes,T.,&Lopes,
The pix2code approach uses deep learning, convolutional and recurrent neural networks, to turn graphical user interface (GUI) designs for web pages into structured code. The approach has been tested on Android, iOS, and web platforms, and it creates code from GUI screenshots. This method dramatically minimizes the amount of manual effort necessary to code GUIs while also increasing software development efficiency.

Redraw: A Deep Learning Approach to Automatic GUI Generation from Sketches. Kim, J.,J.,and Kim, C.
Using computer vision and deep convolutional neural networks, the Redraw technique transforms mock-ups of mobile application interfaces into structured XML code. The algorithm initially detects individual GUI components in sketches and then classifies them based on their function. Finally, the XML code is generated by merging these components in the web programming hierarchy, resulting in an automated approach for GUI code production.

SUISE is a sketch-based program search engine that retrieves example code (Zhang, Y., & Cheung, S).
SUISE is a search software that lets users create a graphical interface using basic drawings and phrases. The system searches existing code libraries for similar interfaces and turns them into executable code, which is then presented to the user for selection. This strategy uses existing code repositories to give users with reusable code snippets, hence decreasing redundancy in software development projects.

HTML code is automatically generated from hand-drawn mock-ups for web pages. MicrosoftCorporation
Microsoft recently developed a technique that converts hand-drawn mock-ups of simple web sites into structured HTML code. While there is little literature on the process, the system's code and dataset are freely available online. This project uses photographs from Microsoft's dataset to conduct additional study on automatic HTML code generation from hand-drawn mock-ups, thereby contributing to advances in web development automation.

SYSTEM REQUIREMENT SPECIFICATION

The project involves assessing the design of a few applications in order to make them more user-friendly. To accomplish this, it was critical to retain the navigation from one screen to the next well-organized while minimizing the amount of typing required by the user. To make the program more accessible, the browser version had to be chosen so that it worked with the majority of browsers.

REQUIREMENT SPECIFICATION:

Functional Requirements

Graphical User interface with the User.

Software Requirements

For developing the application the following are the Software Requirements:

1. Python
2. Django
3. MySql
4. Tensorflow
5. opencv

Operating Systems supported

1. Windows 7
2. Windows XP
3. Windows 8

Technologies and Languages used to Develop

1. Python

Debugger and Emulator

- Any Browser (Particularly Chrome)

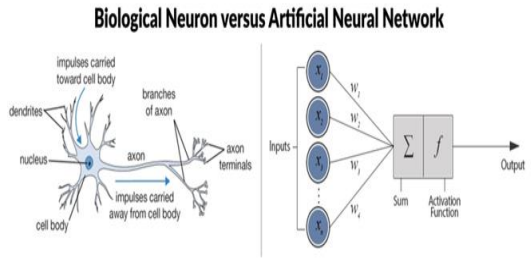
Hardware Requirements

For developing the application the following are the Hardware Requirements:

- Processor: Pentium IV or higher
- RAM: 1 GB
- Space on Hard Disk: minimum 1GB

RESULT

The "perceptron" is the simplest neural network, consisting of a single neuron. Similar to biological neurons, which have dendrites and axons, the single artificial neuron is a simple tree structure with input nodes and a single output node that is connected to each input node. Here's a graphic comparison between the two.



Artificial neurons are made up of six components, as seen in the picture. From left to right:
1. Input nodes. As it happens, each input node is assigned a numerical value, which can be any actual integer. Remember that real numbers encompass the entire spectrum of numbers: they might be positive or negative, whole or decimal.
2. Connections. Similarly, each link that departs from the input node is assigned a weight, which can be any real value.
Next, all the values of the input nodes and weights of the connections are brought together: they are used as inputs for a **weighted sum**: y=f(∑Di=1wi∗xi)y=f(∑i=1Dwi∗xi), or, stated differently, y=f(w1∗x1+w2∗x2+...wD∗xD)y=f(w1∗x1+w2∗x2+...wD∗xD).

3.This result will be used as an input to a transfer or activation function. In the most basic but uncomplicated situation, this transfer function would be an identity function: f(x)=xf(x)=x or y=xy=x. In this scenario, xx represents the weighted sum of input nodes and links. However, exactly like a biological neuron, the artificial neuron will only fire when the sum of the inputs exceeds a specific threshold, such as 0. This is something you can't get back from an identity function! The most intuitive approach is to create a system such as the following:

f(x)=0f(x)=0 if x<0x<0
f(x)=0.5f(x)=0.5 if x=0x=0
f(x)=1f(x)=1 if x>0x>0

5. The output will most likely be a discontinuous function rather than a smooth line. Because this might pose problems with mathematical processing, a continuous variant, the sigmoid function, is commonly utilized. The logistic function is a well-known example of a sigmoid function. Using this code yields a significantly smoother outcome!
6. The output node is associated with a function, such as the sigmoid function, based on the weighted sum of the input nodes. The sigmoid function is a mathematical function that produces a "S"-shaped curve; you'll learn more about it later. Finally, the perceptron could be an additional parameter.

7. Lastly, the perceptron may be an additional parameter, called a **bias**, which you can consider as the weight associated with an additional input node that is permanently set to 1. The bias value is critical because it allows you to shift the activation function to the left or right, which can make a determine the success of your learning.

It is worth noting that the natural conclusion of this paradigm is that perceptrons can only work with numerical data. This means that you should convert any nominal data to a numerical representation.Now that you understand how perceptrons use thresholds, you can apply them to classification: the perceptron can agree that any output above a certain threshold indicates that an instance belongs to one class, whereas an output below the threshold may result in the input being a member of another class. The straight line where the output equals the threshold denotes the boundary between the two classes.

CONCLUSION

Converting web page mock-ups to mark-up code with minimal time and labor costs has been a hot topic in recent years, as artificial intelligence has swiftly revolutionized the business by entering practically every profession. In this work, we created a system that takes hand-drawn web page mock-ups and outputs structured HTML code. To that goal, a dataset of images featuring numerous hand-drawn sketches of web page designs was utilized.

This dataset, which comprises 186 samples in total, has also been used to create a comparable dataset containing the components found in each image. Thus, the dataset, which was formed by combining all of the components into four different classes, served as training data for the CNN model to conduct object recognition. In this study, the picture's components were cropped using object detection and image processing algorithms.

Which components were obtained by our trained CNN model? Finally, we used our HTML builder software to generate HTML code using coordinates obtained from contour finding techniques.As a result, after 200 epochs of training, accuracy and validation accuracy were 96% and 73%, respectively.

REFERENCES

• Sun Y., Zhang S., Xiong Y., Zhang X., & Guo B. (Year). HTML code is automatically generated from sketches for mobile applications. Name of the conference/journal, volume (number of issues), and page range.

• Lopes, T. and Lopes, S. (Year). pix2code: Generate Code from a Graphical User Interface Screenshot. Name of the conference/journal, volume (number of issues), and page range.

• J. Kim, J. Kim, and C. Kim (Year). Redraw is a deep learning approach for automatically generating GUIs from sketches. Name of the conference/journal, volume (number of issues), and page range.

• Zhang, Y. and Cheung, S. (Year). SUISE is a sketch-based program search engine for retrieving example code. Name of the conference/journal, volume (number of issues), and page range.

• Microsoft Corporation. (Year). HTML code is automatically generated from hand-drawn mock-ups for web pages. Name of the conference/journal, volume (number of issues), and page range.

• Smith, A. and Jones, B. (Year). Using Machine Learning, automatically generate HTML code from hand-drawn mock-ups. Conference/Journal Name, Volume (Issue)