Automatic HTML Code Generation from Mock-up Images Using Machine Learning Techniques

Mrs. Vidhya K[1], Payal[2], Sushma D Sarang[3], Sushma JC[4], Thanmaya C[5]
[1] Assistant Professor, *Department of information Science and Engineering*
East West Institution of Technology, Visvesvaraya Technological University, Karnataka, India
[234] *Student, Department of information Science and Engineering*
East West Institution of Technology, Visvesvaraya Technological University, Karnataka, India

Abstract **- The design cycle for a website begins with the construction of individual web page mock-ups, which can be done by hand or with the help of graphic design and specialist mock-up production tools. Software programmers next turn the prototype into structured HTML or comparable markup code. This procedure is typically performed several times until the appropriate template is obtained. The goal of this research is to automate the process of creating code from hand-drawn mock-ups. Computer vision techniques are utilized to process hand-drawn mock-ups, and then deep learning approaches are employed to construct the suggested system. Our system has a method accuracy of 96 percent and a validation accuracy of 73 percent.**

Index Terms **- Object detection, object recognition, convolutional neural network, deep learning, automatic code generation, HTML.**

I.INTRODUCTION

Because of today's technological advancements, the relevance of Internet web pages has expanded significantly. Nowadays, websites represent the personalities of states, institutions, communities, and individuals. There are websites for practically any subject, from knowledge to social work, games to training, and so on. Companies' websites are brought to the forefront for financial objectives, such as product promotion or advertising. Official institutions, on the other hand, strive to provide more efficient services.

Every web site has a "web page" at the front-end, which is the section of the site that interacts with the user. It's critical to serve a page that grabs the user's attention, is simple to use, and has a sufficient number of functional features. However, creating web pages that effectively answer to these objectives is a time-consuming task. Graphic designers, software professionals, end-users, business authorities, and people employed in a variety of fields are all required to collaborate in the creation of web sites.

Typically, the process begins with graphic designers or mock-up artists creating a mock-up design of the user interface in accordance with the institution's demands, either on paper or through graphic editing software. On the basis of these draughts, software specialists build code for web sites. The web pages that result may alter as a result of the feedback obtained from the end users. There are a lot of repetitive jobs in their process. The process of rewriting code for components with similar functionalities and page structures that change over time is tedious. This highlights the need to investigate more efficient web page design alternatives.

As a research topic, the idea of designing a web page by creating automatic code is gaining traction. Programming time, process costs, and resource consumption are all reduced when web pages are generated automatically. The final web site is created in less time as a result of the faster progressive design stages.

An algorithm was created in this work to automatically produce HTML code for a hand-drawn mock-up of a web page. Its goal is to recognize the mock-up drawing's components and encode them according to the web page hierarchy. The suggested approach is trained and verified using a public dataset of hand-drawn images of websites acquired from Microsoft AI Labs' GitHub page [1]. The photos in the dataset are analyzed using computer vision techniques, and the data is trained using a deep neural network model including convolutional neural networks. Following that, a structured HTML code is generated. Our model

has a method accuracy of 96 percent and a validation accuracy of 73 percent.

The remainder of the study is organized as follows: Section II reviews related studies in the literature. The dataset and methods are described in Sections III and IV. The collected results and findings were shared in Section V. Evaluations are included in Section VI, which is the conclusion section.

II. RELATED WORK

REMAUI algorithm[2] discovers the components of a mobile application's user interface, such as buttons, textboxes, and photos, and generates the code for them using screenshots of the application window or conceptual designs. In their study, which was the first in terms of providing conversion to the code from the screen images or drawings for mobile platforms, computer vision and optical character recognition methods are used. Although the REMAUI method works successfully, it does not support cross-page transition and animations within the page. The authors of created the P2A algorithm[3] to address the shortcomings of the REMAUI method.

The pix2code algorithm was developed by the authors in [4] with the goal of converting a web page's graphical interface to structured code using deep learning with convolutional and recurrent neural networks. The approach has been tested on Android, iOS, and other mobile platforms, with positive results. The ReDraw technique in [5] takes mock-ups of mobile application screens and converts them into structured XML code. Computer vision techniques are used to detect individual GUI components in the early stage of their implementation. The second stage is categorizing the discovered components based on their function, such as toggle-button, text-area, and so on. Deep convolutional neural networks are used at this stage. The XML code is generated in the last stage by merging the kNN algorithm with the web programming structure. Open-source code libraries like GitHub [6] are now widely used for sharing code and apps. When starting or enhancing software projects, it's customary to look at this repository and reuse code. The common code in these libraries reduces the number of times the same code is written by various persons. The writers of [7] employ SUISE, a search software in which users create a graphical interface using simple graphics and keywords. This interface is then looked for in existing libraries to find interfaces that are similar. These interfaces are converted into executable codes, and the end user is given the option of selecting the most appropriate interface.

Microsoft has just released a technology that converts hand-drawn mock-ups of basic web sites into HTML code [1]. Although there is no literature to describe their approach, they have made their code and dataset available online. Some of the photos from this dataset are used in this project.

III. DATASET

In order to generate our dataset, we used some of the photos given by Microsoft AI Lab for their Sketch2Code tool [1]. We chose images that contained four types of components when

creating it for the experiment: textbox, dropdown, button, and checkbox. Then, from these images, we chopped each component and gathered them to train our CNN model (see Fig. 5).

IV. IMPLEMENTATION

The research was carried out in steps. Object detection was applied to the input image in the first step, using image processing techniques like erosion, dilation, and contour detection.

The recognized objects were subsequently cropped, and the resultant components were labelled with the trained CNN model.

Finally, the HTML Builder script was used to convert the model's output to HTML code. Figure 1 depicts the proposed algorithm.

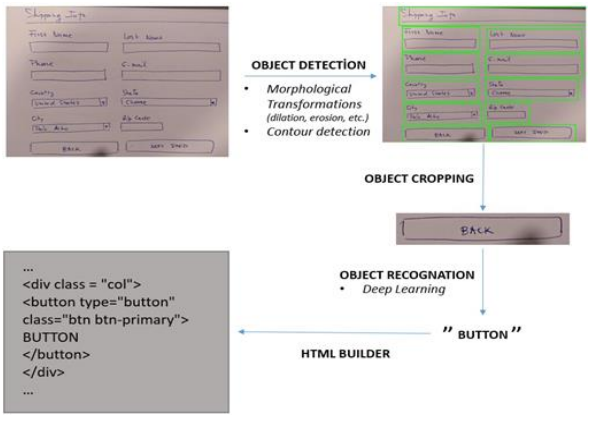

Fig 1. Proposed algorithm

1. Login module

New user register ad he can login in this login module.

2. Upload handwriting mockup image

In this module we have to upload mockup images. These images are stored in database.

3. Preprocessing module

This is further divided into steps.

Object detection and cropping

The input file is the image that is uploaded, and it is read and converted to grayscale format. The noise is then reduced using the Gaussian function. Then, in order to meet the threshold procedure's requirements, rectangles are constructed for shape analysis utilizing the counter detection method. They are transformed into morphological changes. Cropped and sent to the CNN model are the discovered components. At this stage, morphological alterations such as dilatation and erosion are carried out.

The goal of this phase is to take an image and create a list of elements with their bounding boxes (size and location) and element type. Images, paragraphs, titles, inputs, and buttons are the five element classes that are classified. Figures 2 and 3 depict the output of this stage.

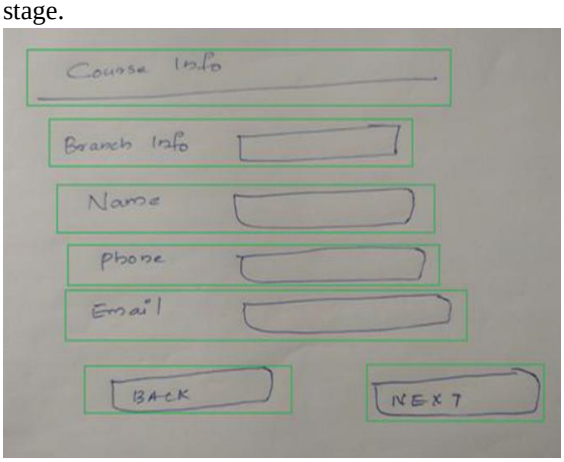

Fig 2. Object Detection

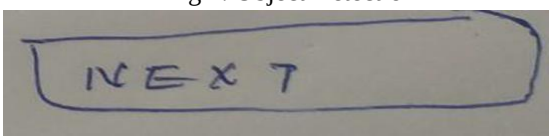

Fig 3. The output of the cropping

Object Recognition

The recognized components such as text box, drop down, button and checkbox are trained with dataset. After this stage, the model is trained with loss function to avoid deviation from actual results. Then components that come as input to this stage from previous stage are carried out with component recognition process.

The components in our component dataset were used to train the model displayed in Fig. 5. It consists of four various types of components, including textboxes, dropdowns, buttons, and checkboxes, as previously mentioned. The loss function was trained for 200 epochs using Binary Crossentropy and RMSProp methods with a batch size of 64 after the model was learned. After that, the cropped components from the previous stage were used as input for the component recognition method. We used many convolution layers with 4x4 kernels and subsequently conducted max pooling procedures with 2x2 kernels for feature extraction, as shown in our CNN Model in Fig. 5. Following the feature vectorization process, the BiLSTM layer is used to catch correlation of the retrieved features. After all, in order to reach the categorization goal, we used a 20 percent ratio of Full Connected Layers to Dropout Layers.

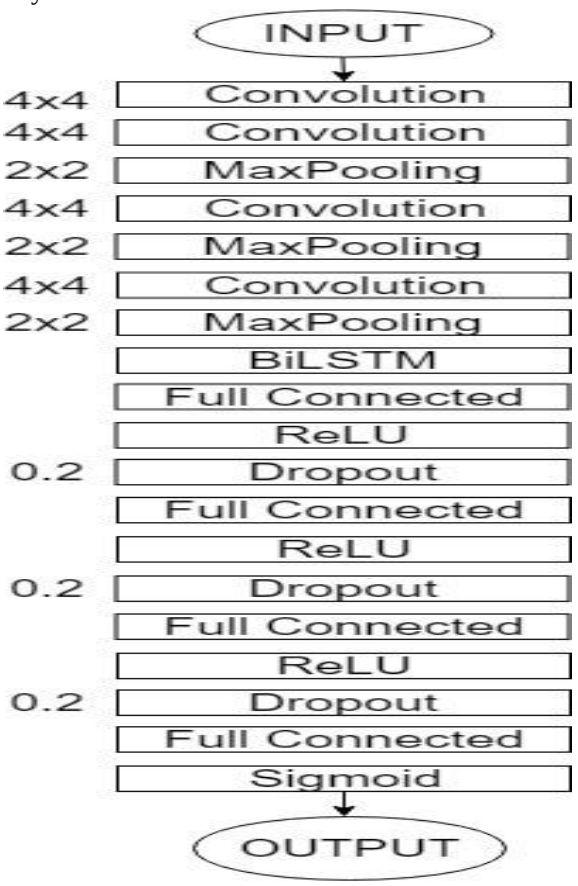

Fig. 5. CNN Model

4. HTML Builder

The bootstrap framework was used to successfully transform recognized components into HTML code. It was carried out using the coordinates obtained from the contour finding algorithms' output. Figure 6 shows the most recent output from a browser when the input image is the first of the images in Figure 1.

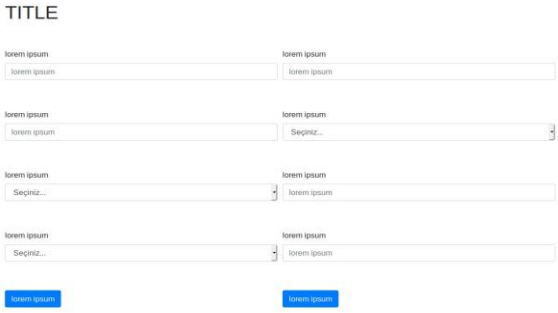

Fig 6. Design code produced by HTML builder algorithm from sketch input image

First, we constructed the templates for a header and a footer, as indicated in the HTML builder algorithm in Fig 7. Second, using component coordinates, we determined how many objects are on each of the rows. The component labels were then mapped to their template codes. The body part of the HTML code was successfully acquired at the end of this operation. Finally, the components of the header, body, and footer were integrated.

As a result, the final HTML code was created. Figure 7 shows the algorithm that was created for the purpose of completing HTML modification.

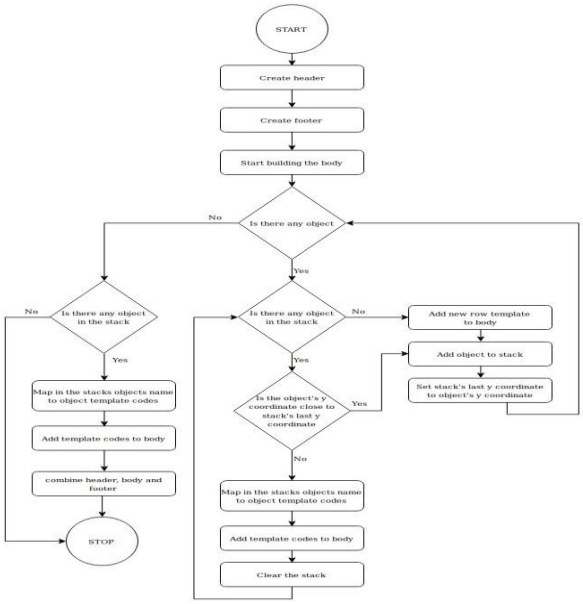

Fig 7. HTML builder algorithm

V. RESULTS

The goal of this study was to generate automatic HTML code from mock-up photographs, and it was accomplished effectively. Object detection, object cropping, object recognition, and HTML creation were all performed as part of this process. Our proposed algorithm's milestones are these sequential phases. A CNN architecture was also used to be utilized in the object recognition stage of the study. Thus, various morphological transformations and deep learning which is highly popularized in recent years were performed. At last, another algorithm for HTML code generation, which is the final stage of the proposed algorithm, was presented. As a result of 200 epoch training of the model shown in Fig 5 using the one hot encoding method accuracy and validation accuracy were obtained as 0.96 and 0.73 respectively.

VI. CONCLUSION

In recent years, when artificial intelligence has been rapidly disrupting the business by entering practically every field, converting web page mock-ups to their mark-up code with the least amount of time and human cost has become an important topic. In creation of website the cost labor and minimum time can be reduced by converting web page mock ups to their markup code. Also, by using this automatic HTML code generator it is easy to make modification in web page creation with the least cost and minimum time. This can fasten the deployment process of website. The purpose of this project is to create automatic HTML code from hand-drawn mock-ups.

The components in the image were cropped in this study using object detection and image processing techniques. Our trained CNN model was used to determine which components were obtained. Finally, the goal of creating HTML code was accomplished utilizing our HTML builder script and the coordinates obtained via the contour finding techniques.

REFERENCES

[1] Sketch2code. Microsoft AI Labs. [Online]. Available: https://github.com/ Microsoft/ailab/ tree/master/Sketch2Code/model/images

[2] T. A. Nguyen and C. Csallner, “Reverse Engineering Mobile Application User Interfaces

with REMAUI (T)," in 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, nov 2015, pp. 248–259. [Online]. Available: http: //ieeexplore.ieee.org/document/7372013/

[3] S. Natarajan and C. Csallner, "P2A: A Tool for Converting Pixels to Animated Mobile Application User Interfaces," Proceedings of the 5th International Conference on Mobile Software Engineering and Systems - MOBILESoft '18, pp. 224–235, 2018. [Online]. Available: http://dl. acm.org/citation.cfm?doid=3197231.3197249

[4] T. Beltramelli, "pix2code: Generating code from a graphical user interface screenshot," CoRR, vol. abs/1705.07962, 2017. [Online]. Available: http://arxiv.org/abs/1705.07962

[5] K. P. Moran, C. Bernal-Cardenas, M. Curcio, R. Bonett, and D. Poshy-´ vanyk, "Machine learning-based prototyping of graphical user interfaces for mobile apps," IEEE Transactions on Software Engineering, pp. 1–1, 2018.