IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 4, Issue 1, June 2024

Draw2Web: From Hand-Drawn Images to HTML + CSS

Mr. Roshan Immanuel[1] and Mr. Rohan S[2]

Students, Department of Computer Science and Engineering (CSE)[1,2]

The Oxford College of Engineering (TOCE), Bengaluru, Karnataka, India

Visvesvaraya Technological University (VTU), Karnataka, India

roshan7156@gmail.com[1] and rohansati04@gmail.com[2]

Abstract: *The website design for the automated code generator begins with creating and uploading individual mock-ups images, which can be hand-drawn with the help of graphics and visual tools. The software programmer then converts the template to produce a standard HTML or CSS code. This process is usually done several times until a suitable model is obtained. The aim of this research is to automate the process of generating code from the drawing model. Computer vision techniques are used to create the plans and then techniques like DL (Deep Learning) and ML (Machine Learning) are used to create the proposed system.*

Keywords: HTML, CSS, Object Detection, Selective Recognition, LSTM, CNN

I. INTRODUCTION

Since the rise of the internet and the development in the website development and technology, it has been noticed that a substantial increase in the use of websites for business, such as Blogs, E- commerce, product pages etc. Having a good website design increases the customer churn rate which in turn increases the revenue of the businesses. No matter what the field, it is become a mandatory to have a well-functioning website to attract customers and increase the revenue, the usage of the websites vary differently on the requirement of the E-Commerce, the functional websites for advertising and productmarketing and others to provide service.

As per a recent report, only 9.15% of India's population has a basic understanding ofcomputer knowledge which should be at least 45% more given the population of this country. The digital wave has impacted majority of youngsters and now slowly all small scaled and medium industrial go digital for survival, thedevelopment of modern day algorithm could help common public convert their hand-drawn website template into HTML code and a functioning website.

Providing a smooth interactive user experience to potential end user is the goal for most of the business, the prolonged process with multiple steps like prototyping, designing and user- testing. Large corporations have a separate team working for it for weeks and months whereas small and medium scaled businesses struggle having these resources and to provide a smooth user experience.

It is evident that Automatic Code Generation (ACG) is going to play a crucial and important role in the e-commerce market since industries have started utilizing it to increase their speed and lower their costs.

The biggest challenge right now is the scarcity of computing power, but gaining experiencesfrom concepts like Long Short-Term memories and the Convolution Neural Nets (CNN) has made ACG strategies possible. By adding human experiences and expertise to these, the generated websites will look more natural and industry-ready

II. RELATED-RESEARCH AND WORKS

Tony Beltramelli et al. [13], from Ulzard Technologies in Copenhagen, Denmark, published a paper on generating code from GUI screenshots. The core idea of this concept is to transform a GUI screenshot into computer code using conventional and recurrent neural networks, which are part of the deep learning methods, achieving an accuracy of 77% on platforms like iOS, Android, and web-based technologies.

A paper by Tuan Anh Nguyen and Christoph Csallner et al. [14] from the Computer Science and Engineering Department at the University of Texas Arlington, USA, focused on reverse-engineering mobile apps to identify various

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, June 2024

elements such as images, texts, and videos. Techniques such as optical character recognition (OCR) were used in REMAUI to produce user interfaces similar to different screenshots of Android and iOS.

Machine learning-based prototyping of graphical user interfaces for mobile apps was published by Kevin Moran, Carlos Bernal Cardenas, Michael Curcio, Richard Bonett, and Denys Poshyvanyk et al. [15]. The main purpose of this work is to transform a GUI into code by detecting GUI components from an artifact. Using concepts such as Automated Dynamic Analysis (ADA), Software Repository Mining (SRM), and deep convolutional neural networks (CNN), GUI components were accurately classified into different domain types. Algorithms such as the K-Nearest Neighbors (K-NN) algorithm were used to integrate prototypes from a GUI structure.

Prasang Gupta and Swayambodha Mohapatra et al. [10] introduced a pipeline to convert code from a hand-drawn image. However, implementing the different models required significant computational time and power to produce satisfactory output.

J. Canny et al. [11] proposed two main approaches useful for detection and localization in a mathematical format. For better detection, the two criteria were compared together with different datasets.

K.P. Moran et al. [12] provided a way for transforming mock-up images of GUI into code in three steps. An attention model was added to improve the detection of required and valid components.

III. DATASET

For us to generate our data-set, we used our own hand drawn images to train the data for our model. We chose images that contained four types of components when creating it for the experiment: text-box, drop-down, button, and checkbox. Then, from these images, were cropped each component and gathered them to train our CNN model.

IV. SYSTEM ARCHITECTURE

We use computer vision technology to identify the GUI images which represent the HTML components. Once detected, these components will be grouped into different categories like text-box, buttons, and label. We used DL andCNN to perform the categorization. Finally, the output is formatted into XML programming code, adhering to the web programming framework. The author employed SUJSE, a search programming software, enabling users to correct the GUI using simple hand-drawn keywords [5][2]. For us to generate our dataset, we utilized the Microsoft AI Lab dataset [7][1].

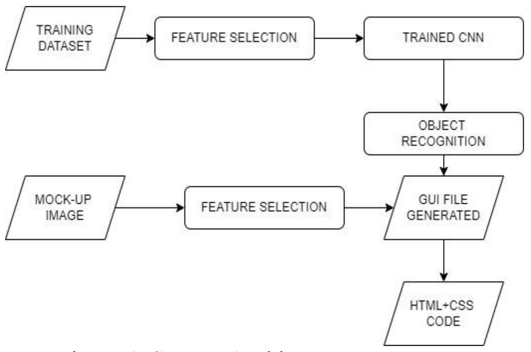

Figure 1: System Architecture

Dataset: the models by creating our own hand drawn dataset. We select an image format that includes dropdown checkboxes, toggle buttons,and text boxes, among other features. Next, we feed each component to our CNN model for training after cropping it.

Methods: This is done in four sections. In first section the input is fed into the objection detection algorithm. The detected components are then cropped. We used Tensor Flow library for the whole object detection process. Finally, the HTML builder technique is used to turn the process' output into HTML code.

Object Recognition: The components from our component dataset to transfer the model. Four different types of components make up a dataset: textboxes, checkboxes, buttons, dropdown menus, and more. This basic computer vision job has numerous applications, such as picture retrieval, video surveillance, and self-driving cars. The items may be Reorganizing cropped components is accomplished by feeding the neural network (CNN) with the cropped component.

Object detection: Thereafter converting the user-inputted image into an array format and obtaining the file in PNG format. Components are discovered by creating a 3x3 rectangle kernel outside the object and using a counter detection technique to detect it. Thecaptured parts are then clipped before proceeding to the CNN model.

CNN model: The CNN consists and has multiple filtering layers, including convolution layers with 4*4 kernels and a max pooling process with 2*2 kernels for extraction. After factorization, the extraction correlation is the BILSTM layer. The dropout layer comes last and is followed by the pool full connected layer;these layers all work together to train the model. The keras library is directly importedinto our code to complete the procedure.

HTML Builder: The Bootstrap framework, the identified components were successfully transformed to HTML code following the reorganization process. The coordinates fromthe counter-finding algorithm's output were used to help with every step of this operation. The HTML builder algorithm first generates the website's header and footer design, and thenuses the component's coordinate to find and select the number of components in a row. After that, we labeled each component's code according to its template code, and the HTML code for the body section was produced. Lastly, the HTML code was formed by combining the header and footer.

Input/Output Design: Our primary goal is to translate the hand- drawn mock-ups, which include text boxes, buttons, and pictures, into HTML code so that we can create a website template or front end that matches the conceptual drawing of the hand-drawn mock-up. We employed Machine- Visionary Techniques (MVT) to transform this hand-drawn artwork into the HTML code for the front-end template. CNN model, cropping, object recognition, and so forth. The author is working on an object detection technique to recognize the hand-drawn images

The object rearrangement approach developed uses a CNN model to train our intended system using a bunch of data that we evaluated on the iOS platform. It functions well and produces have been acquired. The author implements a HTML builder algorithm, which uses the object reorganization technique to turn the recognized item using the model (CNN) into HTML code.

V. METHODOLOGY

The methodology talks about the six importantsteps namely:

- Image-Collection
- Pre-processing Images
- Image Segmentation
- Features Extraction
- Custom Classifier Training
- Code Generator

Image-Collection

We start every project with a selection of carefully chosen hand-drawn mock-up pictures.This set of mock-ups is our main source of data.Mock-ups, as opposed to actual photos, give us precise control over the visual components, which makes them perfect for custom classifier training.

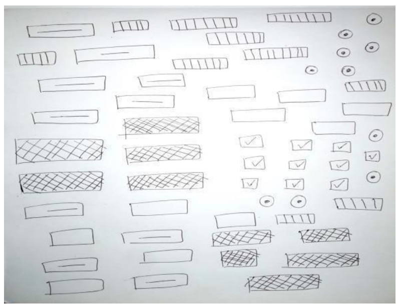

Figure 2: Sample mock-up image

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, June 2024

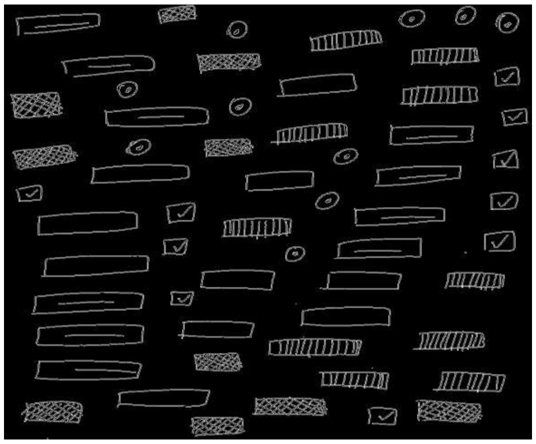
Figure 3: Pre-processed Grey-Scale

Pre-processing Images

Converted to Gray-scale:

For this we need to do pre-processing and it requires our colour input to grey scaled for further analysis. Gray-scale images reduce complexity while preserving important pixel information. This step ensures that the next steps in this stage go smoothly.

Elimination of Noise:

Noise can be proposed into the images during the scanning or drawing process. We use signal-to-noise ratio-boosting methods.

Image Enhancement:

Increasing the contrast and changing the brightness helps make features easier to see. We carefully develop our display models to where important features are visible.

Image Segmentation

It is very much important for the model to segment all weights of the particular input image structure. We need to do extraction of all the features and group the data sets to our target features (such as text, buttons or icons).

Feature Extraction

We now concentrate on obtaining significant and useful characteristics from the divided mock-up areas. The fact that these images are hand-drawn pictures makes texture analysis a vital part. The Gray-Level Co-occurrence Matrix (GLCM) is employed to record inter-pixel spatial relationships. Important characteristics consist of:

- **Contrast**: It measures the intensity variations between neighboring pixels.
- **Entropy**: Quantifies randomness or disorder in pixel values.
- **Energy**: Reflects the uniformity of pixel values. Homogeneity: Describes the similarity of neighboring pixel pairs.

Custom Classifier Training

We use our personally customized data models to train our classifiers. All the required field types, such as buttons, fields, and names, correspond to at least one user interface. CNNs and LSTM are by far the best choice for this purpose. Our goal is to create products that can only be recognized by the unique artwork and features found in our models (Figure 4).

Code Generator

Our class is trained to analyze user interface features in the new model's image. With this information we can create HTML and CSS code snippets. The transition from concept design (model) to implementation (code).

VI. RESULT

The model converts the existing hand drawnimages which are uploaded by the user in the format of hand drawn images that are saved and kept in the local system the model pre- processes the images. This project tries to use machine learning to change and convert hard- drawn images mock images into HTML and CSS code this shows how major improvements in automatic web development.

The system uses screenshots or images uploaded by the user, note these images areuploaded by the user's hand-drawn images and only selected inputs by the user are accepted and converted into automated code. The architecture is divided into 2 major components in which the input and mock-up image recognition and automated code generation by the model.

The model uses complex ML and DL concepts like CNN ACG and LSTM for object detectionand filtering so that only relevant objects are detected for the HTML and CSS code generation. The model produces valid skeleton code for both HTML and CSS for the inputted mockups.

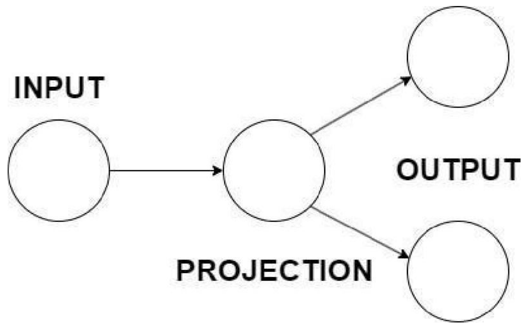

Figure 4: CNN Partition

The CNN model works based on 3 partitions which are input, projection and output (Figure 5). The system would compare the provided data with the pre-trained data sets to recognize the valid shapes and input boxes. The unmatched inputs are ignored and only focus on the allowed components. The selected recognition makes sure that only valid components and chosen attributes are selected for the following generation of the code to give the respected output required for us.

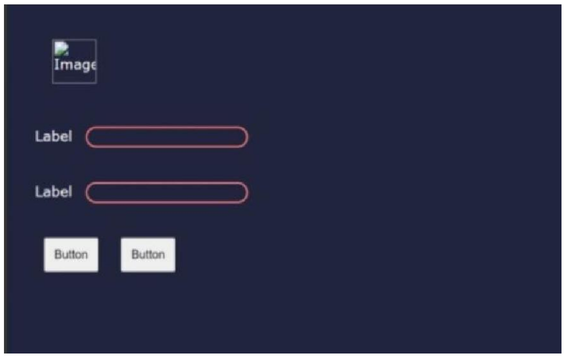

Figure 5: Output Webpage

After filtering the relevant shapes and inputs, the model takes over and uses the components that are selected to produce a well-structured HTML and CSS code. The model produces valid skeleton code for both HTML and CSS for the inputted mock-ups from the respected users and provides users with well- structured code for the user to customize to their requirements. The model achieves around 74% accuracy for object detection and 67% accuracy for code generation.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, June 2024

VII. CONCLUSION

This Project provides a drastic advancement in the field of automated web development by utilizing machine-learning (ML) and deep- learning (DL) concepts and models. The project compares the inputs and shapes from the mock images. This allows the user to save time in creating the basic creation of input boxes and texts rather than skipping the customization directly. The architecture uses two important constructs namely image recognition and code generation which ensures to production a satisfactory code from the pre-trained dataset and model also involves selective recognition to eliminate redundant inputs and provide a clean code for the user to proceed and also ensure the user robustness and reliability.

VIII. FUTURE WORKS

Enhanced Recognition would improve the model to provide better recognition of new inputs and better reliability to users. Integration of design tools for better customization of the code generated would ensure that the user could improve their website for better interactivity and also use of General Adversarial Nets in short known as GAN could be helpful to produce more sample data while the samples are under-fit.

REFERENCES

[1] A. Karpathy and L. Fei-Fei, “Deep visual- semantic alignments for generating image descriptions,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3128–3137.
[2] Senmao Ye, Nian Liu, Junwei Han, Senior Member, IEEE “Attentive Linear Transformation for Image Captioning” JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2015
[3] Jicheng Fu.; Farokh B. Bastani; I-ling Yen “Automated AI Planning and Code Pattern- Based Code Synthesis” Proceedings in the 18th IEEE International Conference on the topic of Tools with Artificial Intelligence(AI) (ICTAI'06) 0-7695-2728- 0/06
[4] Mert Kilickayya, Aykut Erdem, Nazli Ikizler- Cinbis, and Erkut Erdem “Re-evaluating Automatic Metrics for the Image Captioning “arXiv:1612.07600v1 [cs.CL] 22 Dec 2016
[5] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, “Scheduled sampling for sequence prediction with recurrent neural networks,” in Advances in Neural Information and of the Processing of Systems, 2015, pp. 1171–1179.
[6] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, “Babytalk: Understanding and generating simple image descriptions,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 12, pp. 2891–2903, 2013
[7] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollar, and C. L. Zitnick, “Microsoft coco captions: Data collection and evaluation server,” arXiv preprint arXiv:1504.00325, 2015.
[8] M. D. Zakir Hossain, Ferdous Sohel, Shiratuddin, Hamid Laga “A Comprehensive Survey of Deep Learning for Image and Image Captioning “(Submitted on 6 Oct 2018 (v1), last revised 14 Oct 2018 (this version, v2))
[9] Kento Shimonaka∗, Soichi Sumi∗, Yoshiki Higo∗ , and Shinji Kusumoto “Identifying Auto- generated Code by Using Machine Learning Techniques” 2016 7th International Workshop on Empirical Software Engineering in Practice (IWESEP) 13-13 March 2016
[10] Prasang Gupta and Swayambodha Mohapatra, "HTML Atomic UI Elements Extraction from Hand-Drawn Website Images using Mask-RCNN and novel Multi-Pass Inference Technique”, 2020.
[11] J. Canny, "A computational approach to edge detection," in IEEE Conference paper, 1986
[12] K. P. Moran, "Machine learning-based prototyping of graphical user interfaces for mobile apps," in IEEE Conference paper, 2018.
[13] Beltramelli, Tony. (2017). pix2code: Generating Code from a Graphical User Interface Screenshot.
[14] Nguyen, Tuan & Csallner, Christoph. (2015). Reverse Engineering Mobile Application User Interfaces with REMAUI (T). doi:248-259. 10.1109/ASE.2015.32.
[15] K. Moran, C. Bernal-Cárdenas, M. Curcio, R. Bonett and D. Poshyvanyk, "Machine Learning-Based Prototyping of Graphical User Interfaces for Mobile Apps," in IEEE Transactions on Software Engineering, vol. 46, no. 2, pp., doi: 10.1109/TSE.2018.2844788..