

STML(Sketch to Markup Language)

Harish Naik

*Faculty of engineering and technology
Jain deemed to be university
Bangalore, India
harishnaikbm012@gmail.com*

Rishav Raj

*Faculty of engineering and technology
Jain deemed to be university
Bangalore, India
rishav1101@gmail.com*

Sanidhya Jain

*Faculty of engineering and technology
Jain deemed to be university
Bangalore, India
sanidhyax@gmail.com*

H.Srinivasa

*Faculty of engineering and technology
Jain deemed to be university
Bangalore, India
srinivash48@gmail.com*

Denish Goklani

*Faculty of engineering and technology
Jain deemed to be university
Bangalore, India
denish.goklani15@gmail.com*

Abstract—Abstract—Web development encompasses many stages, which involve designing the website done by making a wireframe first, after which the code for the frontend and backend is appropriately written over many iterations until the developers arrive at a fully functional and satisfying website for their need. We see how every website starts with the basic idea and structure of the website that can be described using a wireframe. It shows the basic elements and gives developers an idea about the structure of the website. The result is sent to the developers to create the boilerplate code that places all the elements into place accordingly. The task of converting the wireframe to HTML code is tedious and time-consuming. At present, the boilerplate code is mostly written manually. Currently, the users have to write the HTML code for structuring the elements on the webpage, which leads to a lot of redundant work and users investing their precious time into it. Generally, where the structure of web pages is the same, users would tend to maintain a copy of the pre-existing boilerplate code for reusability. In most cases, the boilerplate code differs for web pages despite the code for the HTML elements being the same. In such situations, the work becomes redundant. However, this process can be automated to ease the world of the web developer community. To make this process effortless, we propose a machine learning model. It will be trained to identify specific symbols and shapes as elements, recognize texts from the wireframe. The model will take in wireframe images as input through the web application. The idea is to process input, recognize each element in the wireframe using the Open Source Computer vision library (OpenCV). Identified elements will each have a corresponding code in the backend. Once the elements on the wireframe are identified, the corresponding code is written into an HTML file. The user gets the HTML file as an output. The creation of the boilerplate code for a website becomes much less time-consuming, which gives developers the freedom to test out many different designs and layouts before opting for the one that best suits their needs.

I. INTRODUCTION

Machines has been a major technological advance of humankind. In daily life, right from doing our basic calculations

to a more efficient AI. Combining of machines with the AI algorithm accommodating into a single system is useful to negotiate against challenging environments. Nowadays, intelligent systems are one of the fastest expanding fields of scientific research. Due to their abilities, it can substitute humans in many fields. Applications include surveillance, industrial automation, reconnaissance, planetary exploration, construction, entertainment, personal services, museum guides medical care, transportation and so on, as well as many other industrial and nonindustrial applications..

II. EASE OF USE

A. Problem Definition

To provide the front-end developers a system that helps in converting an imaginable concept of a website from a sketch to a pre-rendered and editable webpage. Accomplished by using machine learning algorithms and file system algorithm, this will ensure satisfaction and on-point solution for the people who take hours and hours to design a simple form factor for a large website.

III. RELATED WORK

With the development of remote scanning apps which ease up the tasks of scanning and uploading the documents instead of depending on an external hardware device. The recent work shows the development of similar system developed by Microsoft, where they have developed a software known as Sketch2Code [1], where it only sees the structure of an image and gives a skeleton code to develop the said website, but it only gives a basic structure where there is no styling. Coming to the point of developing the image scanning is done using OpenCV, which is a python library and with inbuilt tools it scans and does most of the work. At the same time, being the trend of having both mobile and web application for any software, hence with the recent advancements of mobile application frameworks like Flutter, Kotlin it gives an edge to

any developer to create a user-friendly application. Without image processing and pattern recognition, the mathematical constraints related to the image synthesizing and inbuilt library recognition. To this end, the work in shows a generic approach to generate general notions. Due to the formulation of the Natural Language Processing (NLP), drawback, the computation is too slow to execute in an exceedingly receding horizon fashion, which is required for robust execution.

A. Contribution

This paper shows dynamic advancements for scanning systems application which combine the image processing of inbuilt library on a web and mobile interface. Our main contribution is a whole-body development of a wireframe of a website and planning framework which considers the additional styling via CSS and scripting languages. The notion planner relies on an in-built library which considers of all the pattern defined and with the Machine Learning models. These optimized scanning are tracked by a ML model on a user interface which considers the constraints and hence first give the image processed and then generates an HTML code. To the best we can predict, this work shows for the first time an advanced wireframe of website generation easing up the job of a developer. Furthermore, we can show the whole-body structure of a front-end is produced just by a rough sketch done by user without changing any of the principles of web semantics

IV. OBJECTIVE AND METHODOLOGY

A. Objective

Our main objective is to bridge the gap between client's dream and developer's skills.

a. a. What are we doing? At our platform, the user will draw whatever they want to see in their website and we will generate the wireframe i.e., the skeleton of the website along with the HTML code. The user will first scan its image and then upload it at our platform, then with machine learning model we will analyze the specific patterns depicting the images and then will get our preferred data in the form of the code.

b. b. How are we superior? Existing models only generate the form like setup of a website, which contains only the heading tags and the data field, ours model will be analyzing the whole setup of a full-fledged website's wireframe. The existing model only deals with basic, minimal or even no styling, our model will not only style it but also, we will try to incorporate the styling frameworks. Our model is time saving, the existing model sometimes takes around 10-15 minutes to run the basic setup of just a form, ours will take the same amount of time and gives the wireframe with the demanded styling. The User Experience is going to be far more superior to the models in the market.

B. Methodology

Algorithm Input: The model will require a dataset that contains diverse samples of wireframe sketches for training and testing. The input dataset will consist of sketches of components of the web user interface drawn by different individuals. A set of predefined rules and symbols will have to be adhered to while the components are being drawn. This will ease the process of component recognition in the images. Image Analysis: Once the Input has been provided it is time to extract useful information from the wireframe, but first, the image will have to be pre-processed, which will consist of converting the image to grayscale, noise reduction and edge detection, and other techniques to make the HTML elements detection process easier in the Pattern recognition stage. Pattern Recognition: Once the image has been pre-processed and analyzed for basic image analysis components. Pattern Recognition will be done to recognize patterns for all the HTML elements/ Web components from the wireframe sketches of the training dataset. There will be a specific symbol or a shape for specific web components that will be drawn. Processing: 1. The inbuilt library that will be designed by the developer will contain certain sets of pre-set data and patterns that will be required by the model to analyse it. In our model, we will be defining a basic set of structures and patterns, let's say for our form detection we will be defining: - 1) The labels, text field, buttons, check boxes, list and other multiple setup will be fed into it before hand to make analyse, as of now we will be focusing on the edge detection models of Canny-Edge Detection to detect all the edges and match it with the data fed, if the accuracy will be more than the threshold then it will be detected, otherwise will show a failed detection message. The other labels and buttons functionality will be a detected using the library built or already existing library of python models like Tesseract and Ease text detection which contains the setup of a) Pre-processing the image using HED of any edge detector and some perspective transformations. b) Recognizing the text areas in the image using east text detector and reading it using Pytesseract. Pre-processing of the uploaded image in our model can be used to improve the efficiency of the result. Then, it will be followed by the noise reduction part to get the clearer image. It will be followed by contour detection where it will differentiate the big, small and other contours. Now, as the pre-processing part is finished, we will shift to label detection. 2) Since we need to find the text areas inside the image. We can use EAST model to locate the text in the image, the resulted resize image need to go through a model to have the detected score and geometry of the boundary. The output produced will be used to detect the contours and regions. We could use pytesseract which is free. 3) Now, we will have to detect all our text labels in the image which is done by EAST text detector, the image 1 shows the Neural Network part of how this works.

TABLE I
NOMENCLATURE USED

HTML	Hypertext Markup Language
NLP	Natural Language Processing
OpenCV	Open Source Computer Vision Library
ML	Machine Learning
API	Application Package Interface
NLP	Natural Language Processing
HED	Holistically Nested Edge Detection
PIL	Python Imaging Library
EAST	Efficient and Accurate Scene Text
OCR	Optical Character Recognition

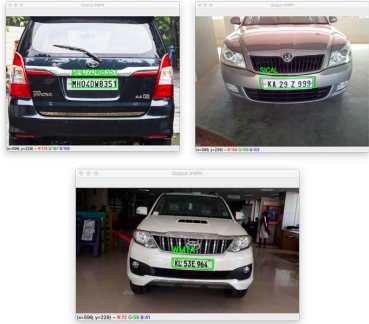


Fig. 1. Vehicle number plate identification.

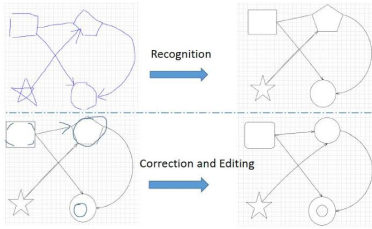


Fig. 2. Label Detection using Neural Network.

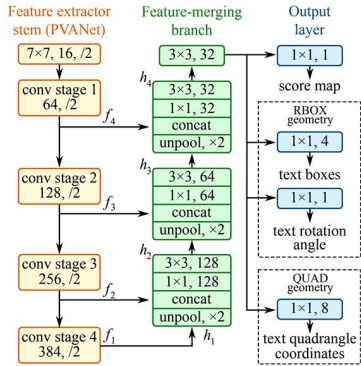


Fig. 3. CV2 using Canny-Edge to understand sketch over image.

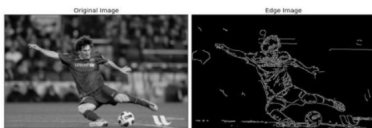


Fig. 4. System Architecture.

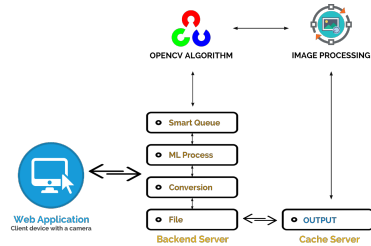


Fig. 5. Use Case Diagram.

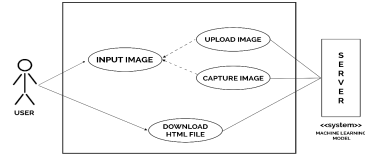


Fig. 6. Level 1 DFD.

SYSTEM DESIGN

C. System Architecture

Basically, the architecture starts from web application that is with client devices with a camera or a file system. The backend server is also connected to the OpenCV which is over the cloud that is also connected to the image processing Algorithm and all of these components are connected to the Cache server which is directly connected to the file system of the browser which helps in downloading of the file directly to the user.

1) *Interface Model*: Our Interface Model consists of a webpage, a minimal Ui so that users don't have to worry much to search for correct buttons. As of the future we might



Fig. 7. Level 2 DFD.

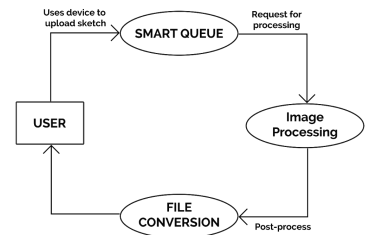


Fig. 8. Sequence Diagram.

bring more Ui enhancements and a mobile app for on-the-go experience.

2) *User Side*: User will have two options, either to use camera (Subject to availability) or access the browser-in-use file system to select the image and upload it to the server. While the processing unit works, the user is allowed to work on their system without being disturbed and as soon as the processing is over the user will get a prompt with options for the file to be allowed or not.

3) *Backend Server*: The backend is consisting of the CV2 algorithms and a HTML-conversion algorithm connected to key queue. The CV2 algorithm uses Canny- rEdge technology to understand the sketch over the image as shown in example below:

4) *Output*: The output will have the code and will be named as 'result.html'.

D. Hardware and Software Requirements

1) *Hardware Requirements*: Processor: Generic System requirements and some classes may have other software requirements which are not critically needed but required for smooth flow

V. CONCLUSION

The aim of this project is to give both the user and developers a platform to fulfil their part of getting their ideas a reality. An image to synthesize the data and convert the image template into a running website that facilitates a global view of the websites while synthesizing the given information. The main areas involved in this field are web development, concepts of machine learning, image processing, pattern recognition, and a software development. Smart systems in general and the machine learning in particular will continue with their evolution during the next years. Cognitive architecture, artificial intelligence, text analysis, and affective human–systems interaction will be increasingly incorporated into remote devices which are designed for many applications. Technological innovations such conversational skills, virtual assistance, and smart factories that have are already having a profound effect on the way industries function. Widespread adoption of advanced remote technologies and platform independent systems are expected. The latest trend confirms that commercialization of these kind of systems due to technological advancement in reducing time consumption is expected to drive the market over the next years and that open-source platforms together with reduction in work and time invested in designing the wireframe is expected to boost demand for this kind of innovations.

REFERENCES

- [1] Robinson, A. , "Sketch2code Generating a website from a paper mockup," ArXiv abs/1905.13750 (2019): n. pag.
- [2] HTML Atomic UI Elements Extraction from Hand-Drawn Website Images using Mask-RCNN and novel Multi-Pass Inference Technique Prasang Gupta and Swayambodha Mohapatra PWC US Advisory, Mumbai, India
- [3] Prof. S. A. Agrawal, Swapnil Suryawanshi, Vikas Arsude, Nilesh Maid Artificial Intelligence based Automated HTML Code Generation Tool using Design Mockups Department of Computer Engineering, Marathwada Mitra Mandal's Institute of Technology, SPP University, Pune, India.
- [4] Pawade, Dipti; Sakhapara, Avani; Parab, Sanyogita; Raikar, Divya; Bhujane, Ruchita; et al.i-Manager's Journal on Computer Science; Nagercoil Vol. 6, Iss. 2, (Jun/Aug 2018): 34. DOI:10.26634/jcom.6.2.15005.