JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

LIVE WEBPAGE GENERATION FROM WIREFRAM USING DEEP LEARNING

Sukshma RD[1], Savitha K V[2],Bhavyashree R[3], Sweatha R[4],Indresh B S[5]
[1]Assistant Professor,[2]Assistant Professor,[3]Assistant Professor, [4]Assistant Professor, [5]Web Developer, SocExpert, Mysuru, Karnataka.

Abstract: A beginning phase of creating applications for users is making a wireframe to design the connection point. A wireframe had been made it is given to the programmer to carry out code. Creating boiler plate standard UI code is tedious works yet at the same time requires an experienced. Designer. Wireframes are visual portrayals of UIs, representation of any visual plan or marking components. UI (UI) and client experience (UX) designers them in the beginning phases of client confronting application improvement to frame the page design and convey what things it should to have. While trying to reduce the effect of web design on the general process of web development, new research headings have been examined that tackle the believability of UI developer to convey a more exact and tangible item in a more shorter time. One of a most encouraging methodologies is to computerize this interaction by utilizing AI. To accomplish this top notch results, we really want an adequately enormous dataset with many examples. On a given information bitmap distinguishes UI components like pictures, texts, holders, and records, by means of PC vision and optical person acknowledgment (OCR) strategies. In this approach we uses the image captioning model to detect and extract text and image format from the given input images. It could be used in several scenarios like prototype designs, mock up designs. Therefore, our contribution will hopefully impact the future of UI design in general, and websites in particular.

Keywords— Wireframes. Webpage generation, Deep Learning, Web Mining. Synthetic Datasets.

I. INTRODUCTION

Since their development, people have been utilizing many kinds of tools to achieve different tasks in a less complex manner. The imagination of the human brain prompted the development of various machines. AI (ML) is utilized to show machines how to effectively deal with the data more effectively. The motivation behind AI is to gain from the information. Many studies Have been finished on the most proficient method to cause machines to advance without anyone else without being expressly customized .Numerous mathematicians and developers apply a

Several ways to deal with find the solution of this issue which are having enormous data collections.

To make such enhanced UIs, the improvement interaction of portable applications regularly integrates non-developers. Client experience (UX) planners and modify, craftsmen plan, realistic and streamline each screen of the UI with a blend of prototyping methods. Some Normal proto-composing model methods including paper and pencil and pixel-based idea drawings are made in Photoshop or comparative visual computerization apparatuses.

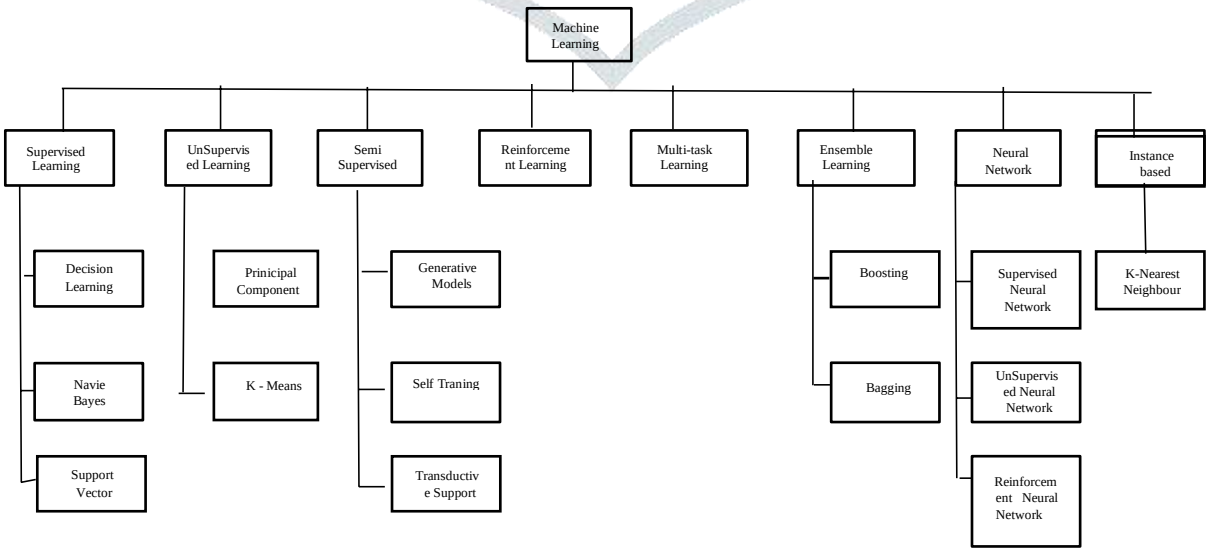
Our aim is to use cutting edge technologies to solve real word business problem. In our scenario we took the design workflow and we have to utilize modern applied science to overcome significant timing creating and testing web pages, main objective of the project is to reduce the time of making mock up screens from wire frames. This project uses the image captioning model to detect and extract text and image format from the images. It could be used in several scenarios like prototype designs, mock up designs.

Our project is basically an image captioning Deep learning model which takes wireframes as input data and gives HTML code as output .And the raise middleware which acts as a mediator between user and deep learning model. Page is an over and over written in HTML, that is find in internet browser. A page can recover by entering an UTL marked into a program's location block .A site page obliges text, illustration and hyperlinks to another pages and documents .A site page often times used to offer data to a watcher, along with pictures or recordings to help portray significant subjects. A site page similarly utilized as a method to offer item or administrations to watchers. Blend of the pages establishment in a site, some of the firms do 100's of mock ups before connecting to real world of data or scenarios, for example Unisys which has its own product called Cargo Portal Service which is the one of world largest cargo management portal and it supply daily 100's of the cargo firms for scheduling the flights and cargo shipment details in day by day basis, but in last they made the decision by moving into new designs which would be 400 pages first they have to achieve their static design from designers using tools like in vision and they have to get it converted into mock up screens from the front end developer and then those screens would be integrated into backend APIs using programmers, like this real world example we can take our imagination to next level by assuming how the resources are being used for same kind of work which is nothing but cost pooling. So this is what makes us to think of coming up with this kind of solution.

In this work, we propose the development of a configurable and extensible tool —– Web Wire

——capable of generating images of hand-drawn-like wireframes from real websites by relying on the rendered DOM of an HTMLdocument as well a son additional information provided by the CSS to extract the layout of a given webpage. It subsequently uses computer-generated imagery (CGI) techniques to draw the page mimicking free-hand sketches, thus producing a wireframe. We then compare the results with two alternative approaches — CSS Based and MockGen — and inquire humans how they perceive the resemblance of the produced sketch with a real website. We conclude that Web Wire is better at representing and generalizing real- world examples, with 90.63% of participants (vs. 63.23% and 9.86%) stating that the wireframes resembled real websites and 57.37% (vs. 32.57% and 57.17%) being convinced that the results were drawn by a human. With the levels of realism attained with our approach, we believe the results of this work can be directly applied to improve the quality and accuracy of the learning phase of current machine learning algorithms, possibly by reducing the required fine-tuning phase with real sketches and thus allowing them to generalize better and produce higher quality outcomes. Therefore, our contribution will hopefully impact the future of UI design in general, and websites in particular.

The delivery in dataset of web-sites which can could used to prepare and assess the model methodologies. Further, we have planned a clever assessment structure which permits observational assessment by making the engineered draws. Our assessment shows that our profound learning approach model plays out our traditional PC vision approach and we can reason that profound learning is the most encouraging bearing to the future exploration.



CONTEXT

The software development process includes several different phases. Typical top-down approaches tend to introduce some precedence to these phases. First comes the undertaking definition, where groups and partners work near accumulate every one of the prerequisites, like its motivation, objectives, interest group, and highlights. After investigation, the task scope plan is considered, which frames explicit exercises and expectations, alongside unambiguous timetables. Then, the software will be designed, developed, tested, and deployed. For the scope of this dissertation, we will focus on the development of websites and the design and implementation stages. Designers are one of the most valuable and influential members of a web development team. Designing an accessible and usable website requires understanding users and how they may interact with its interface, i.e., the user experience. Human-Centered Design (HCD) is an approach particularly concerned with this concept, where designers strive to fulfill human needs when designing, testing, and iterating to achieve the ideal design. In a beginning phase of creating client confronting applications, for example, sites, they start with outlining wireframes on paper as per the venture's prerequisites, as exemplified in Figure 1.1. A wireframe is a low-devotion configuration report, which outwardly addresses a UI, deprived of any visual computerization, styles, or marking components. It is a speedy approach to prototyping and communicating thoughts, for the most part was hand-drawn.

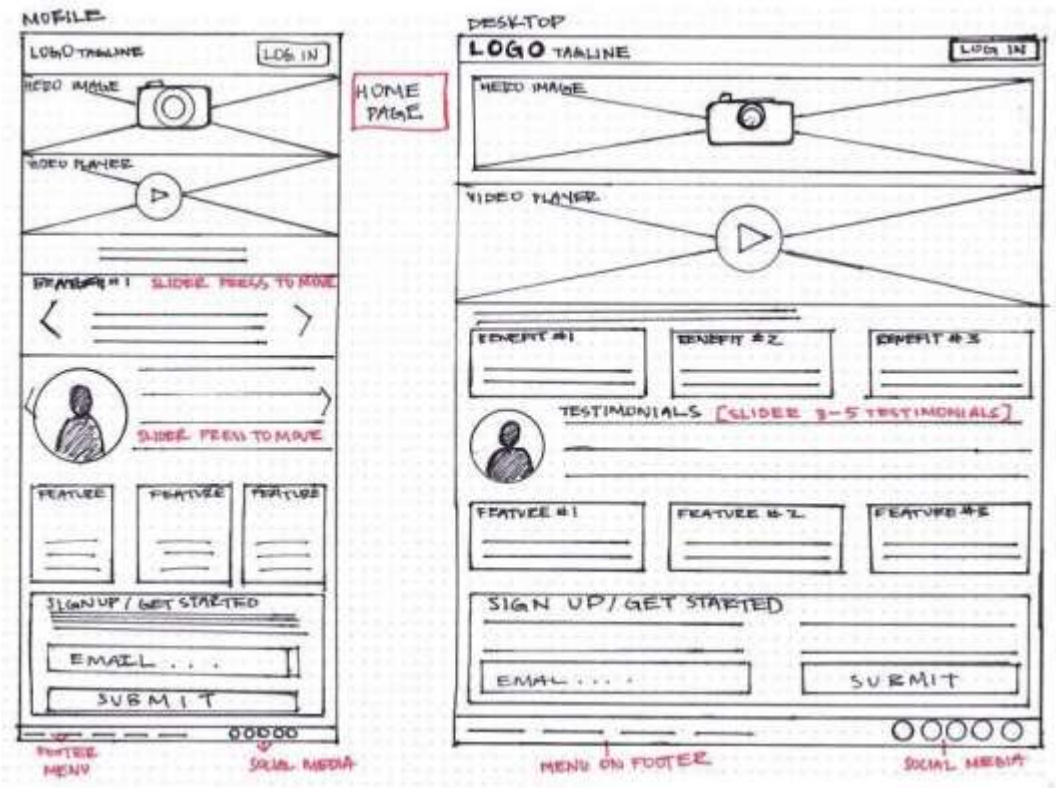


Figure 1.1 Hand-drawn paper wireframes for Taskly, a project management app. This figure represents the homepage of its website on mobile and desktop devices.

As wireframes don't mirror the look and feel of a site page, just the diagram, creators have two other options, especially (a) plan a high-loyalty model utilizing complex devices or (b) resort to engineers to transform their underlying plan into code. The last option holds many dangers, as engineers don't frequently have planning abilities, and the majority of the visual style is absent. Thusly, they as a rule seek after the main other option, which ensures that engineers will obviously comprehend what to carry out without squandering energy on plan choices. When the plan is supported, the designers will change over the high-constancy model into code. This stage is frequently relentless and excess (since a similar work is changed over starting with one organization then onto the next), and it requires a decent joint effort between the planner and the engineer. At long last, the site goes through testing and audit. Tragically, it may not match the requests of partners, so originators should return to the wireframes, and designers should carry out those changes. This cycle rehashes however many times depending on the situation, and when everything is endorsed, the site is fit to be sentoff.

Problem

As expressed over, the model plan cycle rehashes as requested, which can be asset and tedious. Consequently, various arrangements have been considered, endeavoring to decrease the effect that this cycle has on the general course of web improvement .The best way to deal with robotize this cycle is to utilize AI. In any case, to accomplish top notch results, an adequately huge dataset with many examples is required. Gathering such a dataset ends up being one of the greatest difficulties because of the restricted accessibility of organized datasets that contain clarified wireframe outlines made by individuals. The response ordinarily is to utilize engineered datasets for preparing, trailed by a (potential) ensuing tweaking with genuine situations. Current arrangements can be separated into three distinct methodologies. The arrangements that (a) utilization a mockup generator that places significant level components haphazardly in a 2D plane, (b) track down genuine sites, and naturally sketch them through PC vision strategies that separate the undeniable level design from screen captures, and (c) take a current dataset of engineered sites and alter each CSS template. These solutions face numerous challenges, mainly the need to collect a dataset that truly reflects real websites and mimics free-hand sketches of wireframes so that the trained model can achieve more reliable and accurate results. Despite showing promising results, the synthetic dataset generators (i)did not create wireframes that resembled real-world examples,

(ii) did not generalize well, or (iii) used complex and unpredictable methods to create the wireframes (e.g., the use of computer vision techniques to deduce the structure of a webpage)

Motivation

Creators frequently utilize free-hand representations of wireframes to convey their thoughts with the group and partners. These portrayals permit them to convey their contemplations in a short measure of time. Be that as it may, one downside of this approach is the actual portrayal, which may not address the look and feel of the last page. Subsequently, a high- constancy model should be planned with the goal that designers make no suppositions on visual styles nor squander energy on them. It for the most part takes a few emphases to accomplish the ideal outcome as it may not match the client's vision, causing a critical exercise in futility. Late works endeavored to decrease this cycle with the utilization of AI and made different manufactured dataset generators. The authors focused on the machine learning methodologies employed and architecture rather than on the development of a generator capable of producing a whole and accurate dataset. Although the segeneratorshadan overall satisfactory performance, we believe that there is a method that will increase both quality and variety of the generated wireframes, thus being able to create the needed dataset for proper training. By working on existing arrangements of code age from wireframes, we will give a quicker plan emphasis, openness, and the requirement for no go betweens (like engineers). Generally, originators will be fit for conveying a more practical and unmistakable item.

Objectives

This dissertation seeks to explore whether there is a better method capable of generating hand- drawl like wireframes that resemble real-world websites and designs. Hence, it intends to further develop existing AI strategies of shortening the model advancement cycle by adding to the examination field of model age from wireframes. By the end of this research, the tool developed should

(a) rely on the overall structure of an existing webpage in a more predictable manner (which may imply initially discarding computer vision techniques),

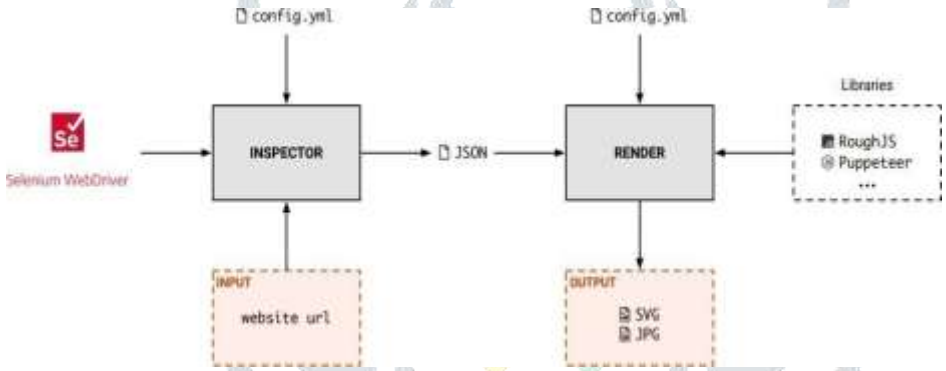
(b) be able to generalize authentic wireframes ,and

(c) draw wireframes in a way that mimics hand-drawn sketches. Lastly, it should outperform existing approaches and be able to easily generate a dataset. We will end this dissertation by conducting an empirical study, where participants will evaluate the wireframes generated by our tool and alternative approaches ,thus determining which has a better performance .Besides, we will analyze the quality and accuracy of the wireframes when compared to current designs.

IMPLEMENTATION

Web Wire is divided into two modules: (a) Inspector, which inspects the webpage and extracts

its UI elements, and (b) Render, which draws these elements in an SVG canvas and saves it as an image. Together, they convert real websites into hand-drawn wireframes by analyzing the webpage's source code and DOM using browser automation tools. Due to the flexibility and lack of enforcement of good practices in HTML, we decided to enable users to configure each module. For this reason, the tool can accommodate different websites and corner cases, as well as the customization of the drawing process, such as add new type faces and choose to use the website's text content or dummy text 1. 1https://www.npmjs.com/package/lorem-ipsum Web Wire 43 In Fig, we illustrate the high-level overview of Web Wire. First, the Inspector receives as input an URL of a webpage. Then, the elements are classified, and their properties are extracted from the web page using Selenium and according to the configuration file. Lastly, the webpage information (e.g., URL and size), the selected elements, and their properties are saved in a JSON file. In the next phase ,the Render receives this file and renders it a sa drawing using existing libraries and by following several heuristics, which will be explained in detail throughout. In the end, the wireframe is exported as two different files: SVG and JGP.



High-level overview of Web wire

To efficiently generate multiple wireframes for different websites, we developed a script that takes a starter JSON and creates a collection of wireframes for each URL using the above modules, as described in Algorithm 1. The starter file includes a random seed, a list of design styles, and a list of websites. The latter is composed of website objects, where each must have an URL (url), the number of wireframes to generate (draws), and optionally, a list of styles (styles).A style defines specific properties of a wire frame design, such as roughness and stroke width. These properties will be further explored. The script takes this file and parses it to a class object, where each website has an identifier. The properties defined with in a style will be the ones used in the design, despite the configuration and defaults of Render. For example, when style1 sets the font property to Kalama, it indicates that the adopted font will always be Kalama for every wireframe that gets this style. The undefined properties, such as roughness, will be set by Render. There are instances where a website(a)has more than one style assigned, and (b) has no styles. For (a), a style will be chosen randomly for each wireframe sketch, whereas (b) all wireframes will have its design properties set by Render. Therefore, the wireframe so fwebsites1 and 4 will have its design set by Render. The wireframes of 2 will

pick either style1 or style2 for each sketch, and of 3 will always pick style1. In the end, a total of 13 wireframes will be generated for four websites.

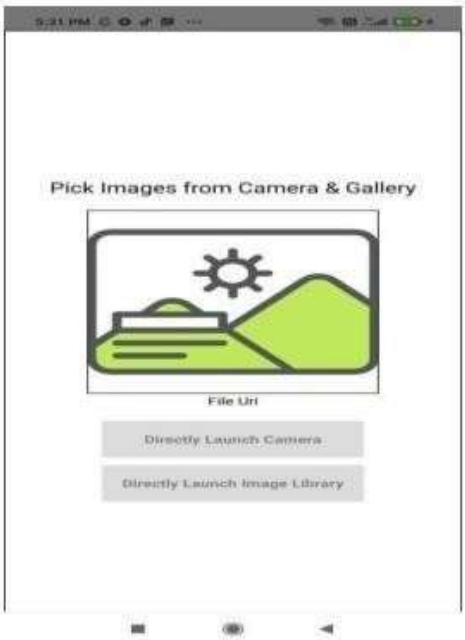
The problem we faced while doing falls under program synthesis and we have leveraged the hand written to code fact to start with, actually there is well named standard given to this kind of requirement in machine learning fields is image captioning model which combines the image and text together which is mainly used to describe the image, so we came up this proof that we could generate text (HTML Code) from the source image and use it as markup tags for webpage. So getting the right dataset is important that we have to get the thousands of hand written data sets to train this model and we have started this with the pix2code paper which is open source and had around 1000+ screenshot with design and generated web pages and each dataset consist of bootstrap elements like buttons, text boxes and dives the source code for each sample consists of token which corresponds to HTML and CSS code snippets. We have leveraged the image captioning model which consists of 3 parts mainly

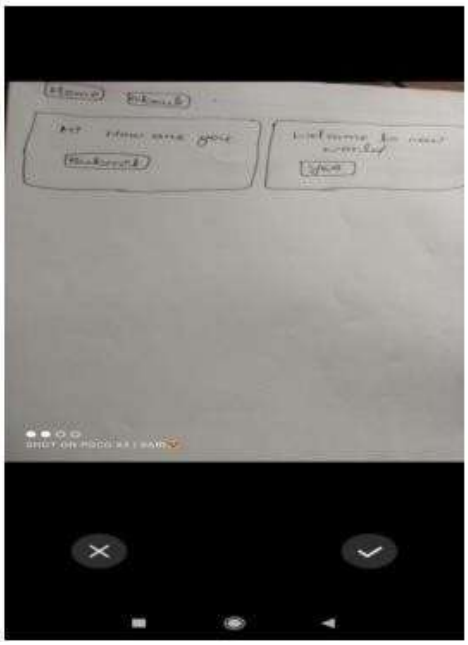
a. A computer vision that uses CNN to extract image features from the image

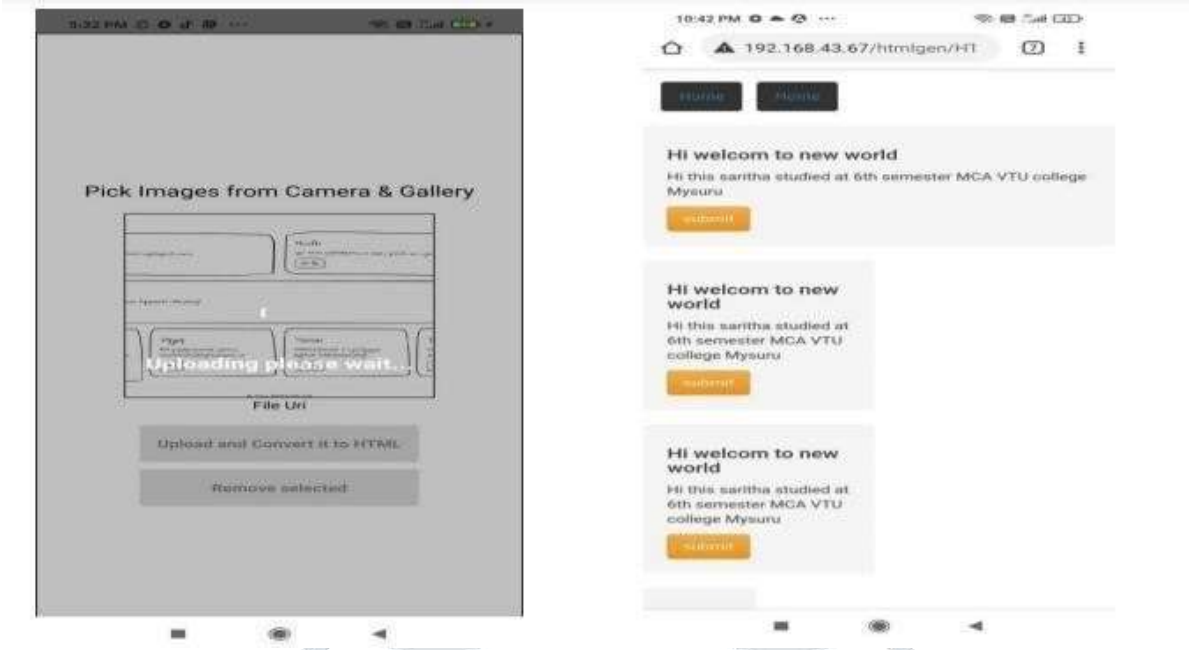
b. A language model (GRU gated recurrent unit) which encode the token

c. A language mode (GRU) which is decode and takes the input from last 2 steps and predicts the next token. To ready our model with training we had divided sources into sequence of tokens and input for this model also a sequence with source image and source image is processed with CNN and once the tokens generated from the model compiler converts it into HTML code which can be rendered in any web browser. so we did it only with limited vocabulary and exactly 16 types of tags were used while building this model and we can call this as mode vocabulary we could enhance this by adding more to it and these components are only bootstrap components and we have used bleu to evaluate our model and this is a common metric to check the efficiency of the task a perfect score is 1.0 so the right element is placed from the right design if this fails we have to enhance it and from out dataset we got around 0.76 which is not bad.

RESULTS







Conclusion

Throughout this dissertation, we faced numerous challenges that not only made us understand how vast the world of web development is but also how vital design is during this process. Reverse engineering websites allowed us to learn the many form so fHTML code, layouts, and styles.

Although at the beginning of this work, our ultimate goal was unclear, after research, we quickly noticed how urgent the development of a tool capable of generating hand-draw-like wireframes was. This tool would enable researchers to create their dataset, thus being able to accurately train machine learning models that would produce code from wireframe images. Related works avoided creating wireframes from existing websites' source code, and since no one dared to try, we were highly motivated to pursuit his approach and prove them wrong. We developed a tool, Web Wire, that (a) extracts the layout of a webpage by inspecting the generated DOM and by relying on additional information provided by the CSS, and (b) given the extracted data, it sketches the UI elements in a brand new white canvas, which ultimately becomes a wireframe. Web Wire performed better that alternative approaches when generating wireframes that resembled real websites and designs, as well as mimicking free-hand sketches made by humans. Lastly, we believe a hybrid approach, containing both our method and computer vision techniques, would significantly improve the accuracy and precision of the detection and classification of UI elements from websites. In this way, we would rely on computer vision in cases where HTML is too ambiguous. Besides, theme though do f drawing Wireframes could be enhanced by adding newan notations and types of element sketches(which promotes the generalization of the dataset) and by producing uneven lines in all shapes(which gives the wireframes a more human-made feel). Hopefully, this work will be continued and expanded in a way that will improve existing solutions of code generation from wireframes, thus providing a faster design iteration , accessibility, and efficient distribution of resources and time.

REFERENCES

[1] A.Antonacopoulos,S.Pletschacher,D.Bridson,andC.Papadopoulos,"ICDAR2009page segmentationcompetition,"inProc.10th InternationalConferenceonDocumentAnalysisand Recognition (ICDAR). IEEE, Jul. 2009, pp.1370–1374.

[2] Microsoft AI Labs, [online] Available: https://github.com/Microsoft/ailab/tree/master/Sketch2Code/model/images.

[3] T. A. Nguyen and C. Csallner, "Reverse Engineering Mobile Application User Interfaces with REMAUI (T)", 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 248-259, nov2015.

[4] https://cs.stanford.edu/people/karpathy/deepimagesent/

[5] https://getbootstrap.com/2.3.2/ [5]https://en.wikipedia.org/wiki/Domain- specific_language

[6] S. Chatty, S. Sire, J.-L. Vinot, P. Lecoanet, A. Lemort, and C. P. Mertz, "Revisiting visual interface programming: creating GUI tools for designers and programmers," in Proc. 17th Annual ACM Symposium on User Interface Software and Technology (UIST). ACM, Oct. 2004, pp.267–276.

[7] Alexander Robinson University of Bristolar15247@bristol.ac.uksketch2code: Generating a website from a papermockup.

Bárbara Sofia Lopez de Carvalho Ferreira da Silva. Automatic Generation of Synthetic Website