iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** IV **Month of publication:** April 2022

DOI: https://doi.org/10.22214/ijraset.2022.40675

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IV Apr 2022- Available at www.ijraset.com

A Deep Learning Approach for Generating Mark-up Code from Sketch Images

Bhavesh Lohana[1], Muskan Tanna[2], Gautam Pamnani[3], Tanish Sahijwani[4], Rohini Temkar[5]
[1, 2, 3, 4, 5]*Department of Computer Engineering, Vivekanand Education Society's Institute of Technology, Chembur, Mumbai-400074, India*

Abstract: ***User Interface (UI) design is an important part of software development. Creating an intuitive and engaging user experience is a key goal for businesses of all sizes and is a process driven by rapid prototyping, design, and user testing cycles. It requires a significant amount of money and effort just to build a production-grade website. It's difficult to generate code from photos. The insight goal is to use modern Deep Learning algorithms to significantly simplify the design workflow and enable any business to quickly create and test web pages. The proposed Deep Learning model consists of a Convolutional Neural Network (CNN) encoder segment and a Gated Recurrent Network (GRU) decoder segment which is trained on a custom database of wireframe sketches and their corresponding code. The network will produce the HTML code, corresponding to the sketch image that is fed into the proposed model.***

Keywords: Computational Neural Network, Deep Learning, Machine Learning, Gated Recurrent Unit, Mark-Up Code Generation

I. INTRODUCTION

UI Prototyping is an integral part of application development. It helps give an insight into how the users will interact with the application. With some experience in web development, it is known that it takes a lot of time to design and code a website from scratch. To build a webpage, a group of individuals from various backgrounds must collaborate. The process begins with the creation of mock-up images, which can be done on paper or graphically. Designers face challenges to convert designs into code. A person needs to have a sound knowledge of Web Development aspects such as HTML, CSS, UI, UX, Color Theory, etc. They spend a lot of time designing Graphical User Interface (GUI) instead of actual logic. The use of Machine Learning algorithms to build code from sketches is a comparatively recent area of research. This might help tackle multiple obstacles faced by the application development team. To tackle this problem, a Deep Learning algorithm could help to effectively code a website just from the sketch drawn for the desired website design. Automatic production of web pages reduces programming time, operation cost, and resource consumption. With rapid progressive design stages, the final website is produced in a shorter time. An algorithm has to be developed to automatically generate the HTML code for hand-drawn mock-ups of a website. It is aimed to determine the components created in the mock-up drawing and to encode them according to the web page hierarchy.

II. LITERATURE SURVEY

Tony Beltramelli in [1] used Convolutional Neural Network(CNN) as an encoder to perform unsupervised feature learning. The decoder used is a stack of two LSTM layers of 512 cells each. The model was trained on batches of 64 image-sequence pairs. This approach had an accuracy of 77%. The shortcomings of this approach were training on a small dataset.

Alexander Robinson in [2] used another approach by turning images into black and white counterparts of the sketch and then having two different approaches i.e. classical Computer Vision(CV) technique and Deep Learning Segmentation such as CNN, ANN or R-CNN. It was trained on 250 wireframe sketches and their corresponding website code. The first and second approaches had an average precision of 0.6024 and 0.7138.

In another related topic, [3] by Siva Natarajan, Christoph Csallner used REMAUI(Reverse Engineering Mobile Application User Interfaces) and OCR.

Carlos Bernal-Cardenas, Michael Curcio Richard Bonett, Kevin Moran and Denys Poshyvanyk in [4] uses CNN can be effectively trained to classify images of GUI-Components from a mock-up. Then that was used by an iterative K-nearest-neighbors (KNN) algorithm and Computer Vision technique on mined GUI metadata and screenshots to translate into code.

Tiago Bouc ̧as and Antonio Esteves in [5] use two approaches. The first approach is the hybrid architecture of CNN and two RNNs as the encoder-decoder architecture.

The second approach includes a You Only Look Once(YOLO) network and a layout algorithm. After testing it on the same dataset, the first and second approaches had an accuracy of 71.30% and 88.28% respectively. The models were trained on 1100 images.

Akash Wadje and Rohit Bagh in [6] used Artificial Neural Network(ANN) and Multilayer Perceptron Networks(MPN) created and used a dataset by finding websites and manually sketching them and manually sketching websites and building matching websites. The HTML code is not generated directly by their approach, instead Domain Specific Language is generated.

Piyush Agrawal, Subham Banga, Vanita Jain, Rishabh Kapoor and Shashwat Gulyani in [7] in which they use RetinaNet detection architecture, it uses 50-layer ResNet variant. The complexity of the model was increased when some other variants were implemented. Feature Pyramid Networks (FPN) is used to obtain information and achieve classification. The dataset consists of 10 different UI components such as Button, Link, Image, Paragraph, etc. For training, 2001 samples of components from 149 sketches are used. The training was done on 50 epochs resulting in high accuracy with low inference time.

Harish Naik, Rishav Raj, Sanidhya Jain, H.Srinivasa and Denish Goklani in [8] used an approach where they pre-processed the image by converting it into grayscale for easier pattern recognition. Python's OpenCV library is used to extract the labels, shapes and symbols. To extract the text, Tesseract OCR has been used. The dataset consists of component sketches of web user interfaces.

Shraddha Punder, Shweta Patil, Rutuja Pawar, Jacob John in [9] have a similar approach as [1] as they used CNN to encode the input image and which is then fed into a stack of two LSTM layers of 128 cells each. The decoder has two LSTM layers with 512 cells each, as a stack. The Softmax layer has been used to perform multi-class characterization.

Vaishnavi Kalbande, Kajal Meshram, Samiksha Somnath, Raksha Deshmukh, Mrunali Mohod in
[10] had a similar approach to [9] where object detection and recognition was applied to an image to extract distinct kinds of components such as buttons, checkboxes, etc. After that, the cropped components were fed into a CNN model with factorization by Bidirectional LSTM (BiLSTM). The HTML code was generated with the help of the Bootstrap framework.

JonnadulaNarasimha Rao, Gajula Harish, Annapurna das, Y. Tejasvi in [11] have used object detection and pruning techniques on the image, as well as Gaussian function to reduce the noise. The approach they have used is similar to [10] where the CNN model is used with the BiLSTM layer. The paper also uses pix2code's[1] algorithm to generate the output which is then used to create the HTML output using the Bootstrap framework. This approach showed an accuracy of 90%.

Yanbin Liu, Qidi Hu, Kunxian Shu in [12]. In this approach, pix2code[1] which consists of CNN and LSTM is replaced by CNN and BiLSTM. This approach uses a stack of two BiLSTM(128 cells each). And the decoder consists of a stack of BiLSTM with 512 cells each. The approach improved the pix2code framework as the accuracy reached 85%.

Jieshan Chen, Chunyang Chen, Zhenchang Xing, Xin Xia, Liming Zhu, John Grundy, Jinshui Wang in [13] used an approach consisting of 3 steps. First, the dataset is built using real-application UI designs using automatic GUI exploration. Then, a CNN autoencoder is used for encoding the semantics of UI designs from the dataset. Then, the output of UI designs is embedded into vector space using the encoder.

Yuntian Deng, Anssi Kanervisto, Jeffrey Ling, Alexander M. Rush in [14] uses an approach for Image-to-LaTeX markup generation where CNN is used to extract image features that are placed in a grid. Each row of the grid is then encoded using a Recurrent Neural Network (RNN). The decoder for this approach is also defined on RNN. The dataset used consists of real-world expressions written in LaTeX.

The comparative study of the proposed model with existing models is discussed in Table 1.

Table I
Comparative Study Of Code Generating Model

Approaches	Encoder	Decoder	Pros	Cons	Accuracy
pix2code: Generating code from a graphical user interface screenshot	CNN	LSTM	Provides higher accuracy in image recognition problems.	Small dataset for training and fewer parameters	77%
Sketch2code Generating a website from a paper mockup	CV + ANN/CNN/R-CNN	ANN/CNN/R-CNN	Color detection	The wire-frame generated varied from the original sketch, some being highly similar while some less	60%-71%

Converting Web Pages Mockups to HTML using Machine Learning	CNN + RNN	RNN	Covered a wide range of HTML elements with better precision. The approach identified the elements and their coordinates perfectly.	Half of the layers were freezed due to limited memory. Lack of data to train.	71.30%-88.28%
Pre-programmed Web Page Implementation For Mockup Image	CNN + Gaussian Function	BiLSTM	Gaussian functions helps reduce the noise.	The model is difficult to train.	90%
Improving pix2code based Bi-directional LSTM	CNN + BiLSTM	BiLSTM	Better accuracy than using LSTM as decoder[1]	BiLSTM are prone to overfitting	85%
Proposed Method	CNN + GRU	GRU	GRU is less complex and faster as it only has two gates that are update gate and reset gate, unlike LSTM.	The only notable con for the proposed system would be limited resources to train the model.	90%

III. METHODOLOGY

The basic flow of the proposed Deep Learning model has been described below. All the modules are elaborated in detail as follows:

A. Data Collection

A dataset consisting of hand-drawn wireframe sketches and their HTML code equivalents has been considered for the proposed system. For starters, pix2code's [1] dataset which contains 1750 screenshots of generated websites and their corresponding source codes will be used.

Features of this dataset were:

1) Each image in the dataset has sketched websites that are made up of a few simple Bootstrap elements like buttons, text boxes, and divs.
2) The source code consists of tokens from a Domain Specific Language (DSL).
3) A compiler will be used to translate from the DSL to working HTML code as
4) Each token corresponds to a snippet of HTML and CSS.

B. Data Pre-Processing

All the wireframe images will be converted into grayscale and sized at 224x224 for consistency. And the corresponding source code tokens will be tokenized with <START> and <END>. The dataset will be divided into two sections, training and testing consisting of 80% and 20% of the dataset respectively.

C. Proposed Methodology

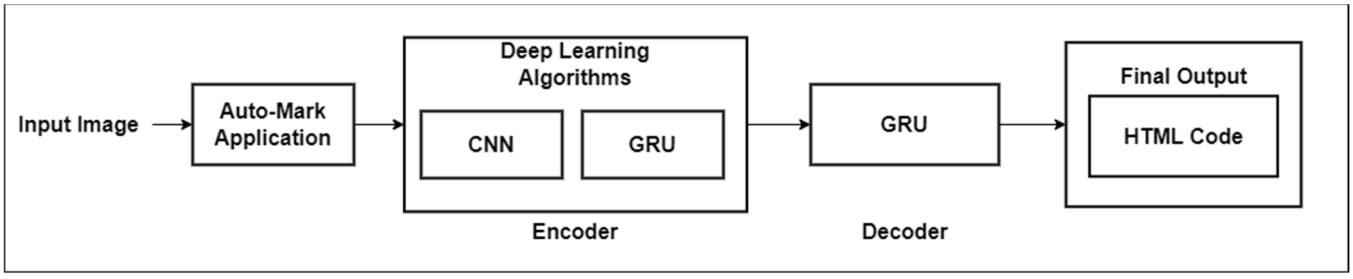


Figure 1. Proposed System Design

D. Model Selection

As seen from the above block diagram, the architecture will begin with what is typically an Encoder for the model. The encoder segment consists of CNN and GRU. Here, CNN is significant for image segmentation as it preserves various features present in the dataset. This makes the data appropriate for training and testing. A GRU will also be employed besides this as it will consider all the HTML tags corresponding to the input image. The GRU will solve the vanishing gradient problem which usually occurs in networks with LSTM units. Hence, we will be going forward with GRU instead.

After the encoding segment, the result will be forwarded to a GRU which essentially forms our decoder segment. The essence of this GRU decoder will be to decode the output from the encoder, process it, validate it against the provided HTML tags to check for loss and accuracy & finally to emit the output of the model.

IV. RESULT & DISCUSSION

For evaluation measures, we are going to use Bilingual Evaluation Understudy, also known as BLEU. It's a measure for analyzing machine-translated text automatically. The BLEU score is a value between 0 and 1 that reflects whether closely the machine-translated text matches a set of high-quality reference translations. A value of 0 means that the machine-translated result has no overlap with the reference translation (low quality), while a value of 1 signifies that the overlap is ideal (high quality).

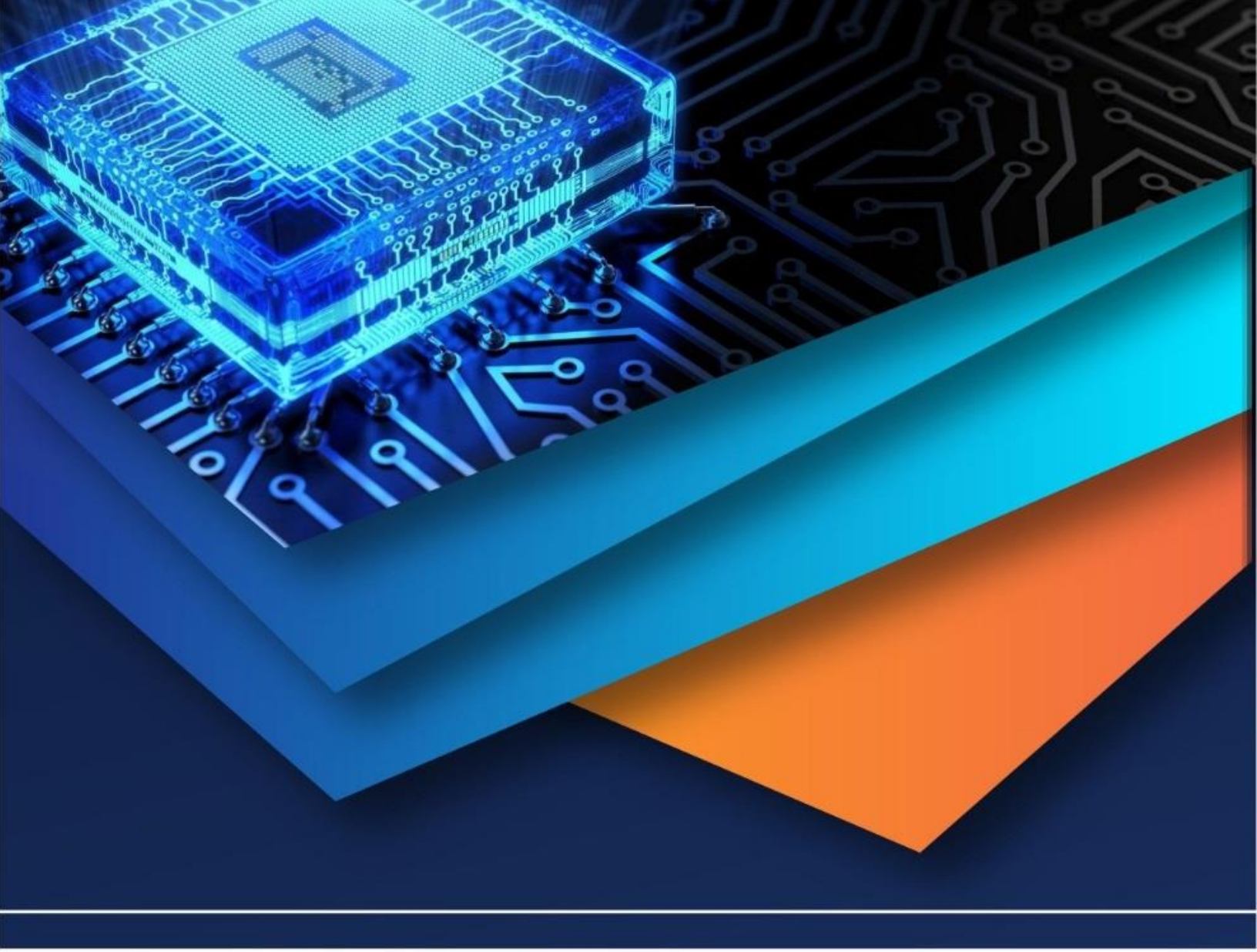
After the extensive study of work done before and the detailed working of the model's encoder and decoder, the predicted accuracy of the proposed system would be well over 90%.

V. CONCLUSION

Converting web page mock-ups to their mark-up code with minimum time and labour cost has become a significant topic in recent years when artificial intelligence has been rapidly revolutionizing the industry by entering almost every field. In this proposed system, it has been planned to develop a system that takes hand-drawn web page mock-ups and gives a structured HTML code. To that end, a dataset consisting of images containing various hand-drawn sketches of web page designs will be used to give us the desired HTML/CSS code with high accuracy. By automating the process of converting mock-up images to basic GUI code, the proposed system is able to bridge the gap between developers and designers. This allows the designers to explore the designs, whereas the developers focus on the application's performance rather than its appearance and orientation.

REFERENCES

[1] T. Beltramelli, “pix2code: Generating code from a graphical user interface screenshot”, 2017.
[2] Robinson A., “Sketch2code Generating a website from a paper mockup”, May 2019.
[3] S. Natarajan and C. Csallner, “P2A: A Tool for Converting Pixels to Animated Mobile Application User Interfaces”, MOBILESoft, 2018.
[4] 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, 2018.
[5] Tiago Bouc¸as and Antonio Esteves, “Converting Web Pages Mockups to HTML using Machine Learning”, 16th International Conference on Web Information Systems and Technologies, WEBIST, 2020.
[6] Akash Wadje and Rohit Bagh, “Sketch2Code: From Sketch Design on Paper to Website Interface”, IJIRT, Volume 7 Issue 1, June 2020.
[7] Vanita Jain, Piyush Agrawal, Subham Banga, Rishrabh Kapoor, and Shashwat Gulyani. ”Sketch2Code: Transformation of Sketches to UI in Real-time Using Deep Neural Network.”, October 2019.
[8] Harish Naik, Rishav Raj, Sanidhya Jain, H.Srinivasa and Denish Goklani, “STML(Sketch to Markup Language)”, 2020.
[9] Shweta Patil, Rutuja Pawar, Shraddha Punder, Jacob John, “Generation of HTML Code using Machine Learning Techniques from Mock-Up Images”, International Research Journal of Engineering and Technology (IRJET), Volume: 07 Issue: 03, March 2020
[10] Vaishnavi Kalbande, Kajal Meshram, Samiksha Somnath, Raksha Deshmukh, Mrunali Mohod, “Automatic HTML Code Generation from Mock-Up Images Using Machine Learning Techniques”, International Research Journal of Engineering and Technology (IRJET), Volume: 08 Issue: 06, June 2021.
[11] JonnadulaNarasimha Rao, Gajula Harish, Annapurna das, Y. Tejasvi, “Pre-programmed Web Page Implementation For Mockup Image”, Journal of Information and Computational Science, Volume 10 Issue 3, 2020.
[12] Yanbin Liu, Qidi Hu, Kunxian Shu, “Improving pix2code based Bi-directional LSTM”, IEEE International Conference on Automation, Electronics and Electrical Engineering, 2018.
[13] Jieshan Chen, Chunyang Chen, Zhenchang Xing, Xin Xia, Liming Zhu, John Grundy, Jinshui Wang, “Wireframe-Based UI Design Search Through Image Autoencoder”, ACM Trans. Softw. Eng. Methodol. 29, 3, Article 19, July 2020.
[14] Yuntian Deng, Anssi Kanervisto, Jeffrey Ling, Alexander M. Rush, “Wireframe-Based UI Design Search Through Image Autoencoder”, ICML, 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY