e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:01/ January-2022 Impact Factor- 6.752 www.irjmets.com

AUTOMATED HTML CODE GENERATION FROM HAND DRAWN IMAGES USING MACHINE LEARNING METHODS

Gayatri Vitkare[*1], Rutuja Jejurkar[*2], Sammyaka Kamble[*3], Yogeshwari Thakare[*4], Prof. A.P.Lahare[*5]

[*1,2,3,4,5]Department Of Information Technology, Pune Vidhyarthi Griha's College Of Engineering And SSD Institute Of Management, Nashik, Maharashtra, 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 creation tools. Software programmers next transform the mockups into structured HTML or comparable markup code. This procedure is typically repeated several times until the desired template is obtained. The goal of this review is to make the development process of constructing code from hand-drawn mock-ups more automated. Computer vision techniques are utilized to process hand-drawn mock-ups, and then deep learning approaches are employed to construct the suggested system. The process of producing a website begins with the construction of a rough draught of each web page, which can be done using design software or by hand. After that, equivalent code is created for the web page draught. The process is repeated until the required web page is created and the user is happy with the results. This process is complex, costly, and time-consuming. As a result, the proposed system is designed to automate this process. As an input, a hand-drawn picture of a shape is supplied, which is analysed and numerous components discovered. After the components have been identified, they are cropped and recognized using deep learning CNN algorithms. When the component equivalent is recognized. The HTML builder algorithm is used to create HTML code.

Keywords: Object Identification, Object Recognition, Machine Learning With Convolutional Neural Networks (CNN), HTML Code Generation.

I. INTRODUCTION

The importance of the Internet web sites has increased considerably due to the progress made in today's technology. Nowadays web sites reflect the face of the states, institutions, communities, people, etc. Web sites are available in almost every field, from knowledge to social work, from games to training and many more. Web sites created by companies come to the fore for financial reasons for product marketing or advertising purposes. On the other hand, official institutions aim to provide more efficient services. Developers are responsible for building client-side software based on a Graphical User Interface (GUI) mock-up prepared by a designer. The implementation of GUI code, on the other hand, takes time and prevents engineers from focusing on the real functionality and logic of the product they are developing. Furthermore, the computer Furthermore, each target runtime system's computer languages for creating such GUIs are distinct, leading in languages used to create such GUIs are unique to each target runtime system, resulting in laborious and repetitious effort when the product is intended to function on various platforms employing native technologies. We provide a model trained end-to-end with stochastic gradient descent to produce variable-length strings of tokens from a single GUI picture as input while technical and educational to simulate sequences and spatiotemporal visual information in this paper. In this study, an algorithm has been developed to automatically generate the HTML code for hand-drawn mock-up of a web page. It is aimed to recognize the components created in the mock-up drawing and to encode them according to the web page hierarchy.

II. LITERATURE SURVEY

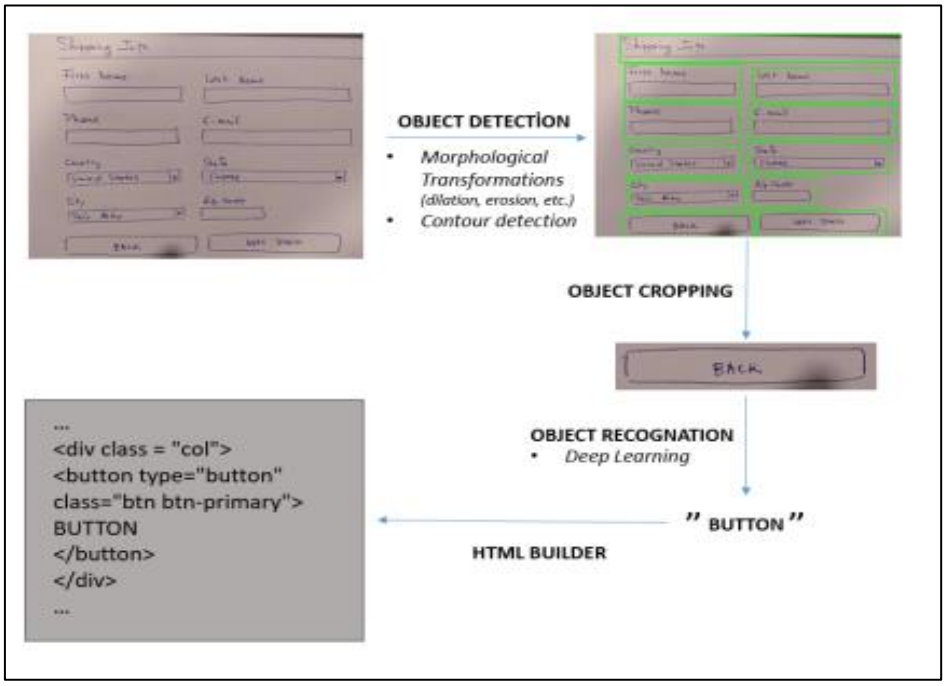
1] A large gap occurs between graphic designers' conceptual designs and functional UI code while building the UI code of a mobile application. Programmers often close this gap by rewriting the conceptual drawings in code, which is time-consuming and costly. To fill this need, we pioneered the first method for automatically reverse engineering mobile app user interfaces (REMAUI). REMAUI uses computer vision and OCR methods to identify UI components on a given input image. REMAUI-generated UIs were close to the originals in our tests on 488 screenshots of over 100 major third-party programmes. We intend to expand the export process to more

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:01/ January-2022 Impact Factor- 6.752 www.irjmets.com

platforms, including iOS and cross-platform JavaScript frameworks. Currently, REMAUI separates each input screen into its own programme. We want to provide a graphical notation that users can use to connect many inputs screen designs, which REMAUI can then utilize to produce a single application with several screens and matching screen transitions. REMAUI will be integrated with tools that produce mobile app functionality using keyword-based code search [44] or high-level models.) We want to index a screenshot corpus by executing REMAUI on it and saving the intermediate results of REMAUI. By exposing this index through a query interface, a user might look for screenshots based on their structure and features.

2] Building classifiers for picture detection and recognition on multiple datasets using various machine learning had advanced significantly in recent years. Deep learning, in particular, has improved accuracy across a variety of datasets. The below are summaries of some of the works. SVM (support vector machines), KNN (K-nearest Neighbour), and CNN were used to evaluate the performance on MNIST datasets (convolutional neural networks). On that platform, the Multilayer Perceptron didn't function well since it didn't attain the global minimum but rather stayed caught in the local optimum and couldn't reliably distinguish digits 9 and 6.Other classifiers performed well, and it was decided that by building the model on the Keras platform, performance on CNN may be enhanced. On the MNIST dataset, implement DNN (deep neural networks), DBF (deep belief networks), and CNN (convolutional neural networks) and compare the results. According to the research, DNN fared the best, with an accuracy of 98.08 percent, while others had error rates and execution times that differed.

3] The study was classified into four categories. In the first stage, image processing methods such as erosion, dilation, and contour detection were being used to search for things in the input image. Following that, the identified objects were cropped, and the associated components were labelled using the trained CNN model. Finally, the model's output was transformed to HTML code using the HTML Builder script.Figure 1 depicts the suggested method.



Once the input data has already been read, it is transformed to grey scale style. Gaussian Blur was then reapplied to them using a 3x3 rectangle kernel. Following the thresholding step, the contour detection technique was used to create a rectangle in order to identify the objects basis of morphological changes. In this approach, the components of the uploaded picture have been detected. Cropping was used to transfer the identified components to the CNN model.

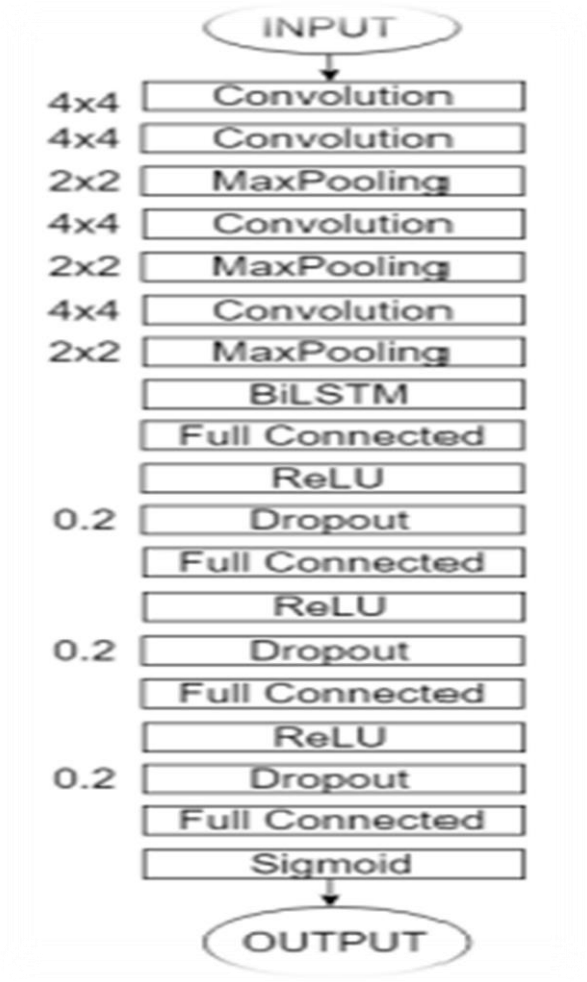
In the phases of morphological features, 8 iteration dilation with a 4x4 rectangle kernel was implemented. With a 3x3 ellipse kernel, the erosion process was used for 4 repetitions, and the dilation process was used for 2 iterations with a 1x10 rectangle shaped kernel. Finally, a 10x1 rectangle shaped kernel was used in the

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:01/ January-2022 Impact Factor- 6.752 www.irjmets.com

dilatation operation. The model was trained using our component dataset chunks. It is made up of four major types of components, as previously stated: textboxes, drop-down menus, buttons, and checkboxes.

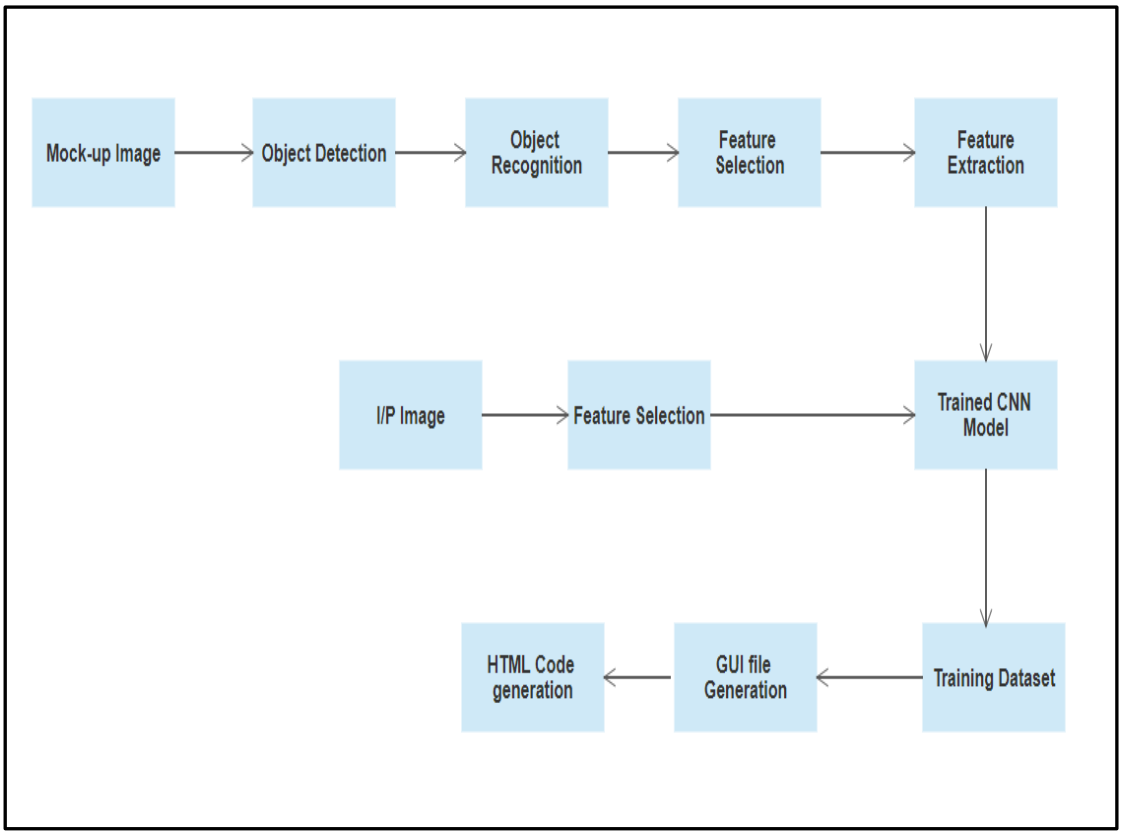
After the model was trained, the loss function was trained for 200 epochs using Binary Cross Entropy and RMS Prop methodologies with a batch size of 64. After that, the component recognition procedure used the cropped elements from the preceding stage as input. For feature extraction, we utilized several convolution layers with 4x4 kernels, similar to our CNN Model, and thereafter max pooling operations using 2x2 kernels. We apply the BiLSTM layer to catch the relationship of the collected features after the feature vectorization procedure. After all, we utilised a 20% ratio of Full Connected Layers to Dropout Layers to achieve the classification aim.

Recognized components were successfully translated into HTML code via the bootstrap framework. It was performed with the help of the coordinates from the result of the contour finding algorithms.



e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:01/ January-2022 Impact Factor- 6.752 www.irjmets.com

III. PROPOSED SYSTEM ARCHITECTURE



- **Detection of Objects**

Picture processing methods such as erosion, dilation, and contour detection were used to find objects in the input image. A dataset is a collection of photos that includes elements such as textboxes, buttons, and labels. Contour detection and morphological transformation are used to detect these elements in a picture. Outside the component, contour detection is employed to draw boundaries. Using morphological erosion, little things are eliminated, leaving only substantial objects. Objects are made more apparent by filling gaps in them using morphological dilatation. These strategies increase the visibility and clarity of an object.

- **Object Recognition**

The dataset was trained using a Convolutional Neural Network. The web form input picture is subsequently delivered to this trained model, which detects image components. Object detection is a difficult computer vision problem that requires predicting both the location and kind of items seen in a picture. One of the most advanced systems for object identification is the Mask Region-based Convolutional Neural Network, or Mask R-CNN.

- **Training data set**

A dataset is a collection of photos that includes elements such as textboxes, buttons, and labels. A dataset used to train a machine learning algorithm is referred to as a training model. The training model is utilized to process the input data through the algorithm in order to compare the processed output to the expected output. The model is modified based on the results of this association. "Model fitting" is the term for this iterative procedure. Machine learning techniques are used using the Kaggle dataset.

- **CNN**

The components collected were then labelled using the trained CNN model after the recognised items were cropped. Before being fed into the CNN, the input pictures are resized to 256 256 pixels (the aspect ratio is not kept) and the pixel values are normalised. There is no extra pre-processing. We only employed modest 3 3 receptive fields convolved with stride 1 to encode each input picture to a fixed-size output vector. To down-sample using max-pooling, these processes are done twice.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:01/ January-2022 Impact Factor- 6.752 www.irjmets.com

- **HTML Code Generation**

The .gui file created for the detected components is then used to produce HTML code. The goal is to encode the web page's identified components. First, header and footer templates are generated. Then, for each row, count the number of items and map the component names to their codes. Finally, put the header, body, and footer together.

- **Pix2Code**

The method examines a screenshot of a graphical user interface (i.e., the layout of a web application interface) and determines what the underlying code looks like through an iterative process. The challenge of creating computer code in a certain programming language from a GUI snapshot is comparable to creating English textual descriptions from scene photography. In both cases, we wish to build variable-length token strings from pixel values. As a result, our problem may be divided into three sub-problems. To begin, there is a computer vision challenge of comprehending a given scene (in this example, the GUI picture) and inferring the items present, their IDs, 2 locations, and postures (i.e., buttons, labels, element containers). Second, reading text (in this example, computer code) and creating syntactically and semantically accurate samples is a language modelling challenge. Finally, the last task is to apply the answers to both preceding sub-problems to construct matching textual descriptions (i.e., computer code rather than English) of the items represented by these variables using the latent variables inferred via scene interpretation. Pix2code is made up of the following elements:

- LSTM (Long-Short Term Memory)
- CNN (Convolutional Neural Network)
- DSL (Domain specific Language)

LSTM:

The CNN Long Short-Term Memory Network, or CNN LSTM, is an LSTM architecture intended primarily for sequence prediction issues using spatial inputs such as pictures or videos. In Kera, we may create a CNN LSTM model by first creating the CNN layer or levels, then wrapping them in a Time Distributed layer, and last specifying the LSTM and output layers.

CNN:

A CNN is a Deep Learning method that can take an input image and assign priority to various aspects/objects in the image using learnable weights and biasesand distinguish between them. A CNN-based model encodes the GUI picture during training. Convolutional Neural Network (CNN) layers for feature extraction on input data are combined with LSTMs to support sequence prediction in the CNN LSTM architecture. On the front end, CNN layers are added, followed by LSTM layers with a Dense layer on the output to create a CNN LSTM.

DSL:

DSLs are computer languages (e.g., Markup, Programming, Modelling) that are designed for a specific domain but are frequently less feature-rich than full-featured computer languages. Using DSLs reduces the size of the search areas and reduces the complexity of the programming language that needs to be simulated. The actual textual value of the labels is neglected in this study since we are primarily concerned in the GUI layout, the various graphical components, and their connections.

IV. FUTURE SCOPE

• Increasing system versatility by allowing users to create front-end designs using a variety of image formats.

• Improving the system in order to produce more appealing designs.

• Improving system quality by adding a new function that allows users to customise the CSS for the website's front end.

V. CONCLUSION

Websites are an important aspect of today's rapidly evolving technological environment since they are beneficial for product marketing and advertising. As a result, it is critical to create a website that will attract a significant number of visitors. We devised a method for taking web page mock-ups, processing them, and producing structured HTML code. A picture-based data collection was used, which included several prototypes

of web page architectures. The CNN model is trained using this dataset. Building a website under the current system is a time-consuming task. Hand-drawn graphics of online forms are converted into HTML code using the suggested technique. As a result, the system will use less time and resources than the current system.

VI. REFERENCES

[1] Automatic HTML Code Generation from Mock-up Images Using Machine Learning Techniques978-1-7281-1013-4/19/$31.00 © 2019 IEEE

[2] Convolutional Neural Network (CNN) for Image Detection and Recognition 978-1-5386-6373-8/18/$31.00 ©2018 IEEE

[3] Reverse Engineering Mobile Application User Interfaces with REMAUI 978-1-5090-0025-8/15 $31.00 © 2015 IEEE DOI 10.1109/ASE.2015.32

[4] pix2code: Generating Code from a Graphical User Interface Screenshot arXiv:1705.07962v2 [cs. LG] 19 Sep 2017

[5] Norhidayu binti Abdul Hamid, Nilam Nur Binti Amir Sjarif, "Handwritten Recognition Using SVM, KNN and Neural Network", www.arxiv.org/ftp/arxiv/papers/1702/1702.00723

[6] Mahmoud M. Abu Ghosh; Ashraf Y. Maghari, "A Comparative Study on Handwriting Digit Recognition Using Neural Networks", IEEE, 2017

[7] 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/

[8] 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

[9] 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

[10] Krizhevsky, Sutskever and Hinton, "ImageNet classification with deep convolutional neural networks", Advances in Neural Information Processing Systems 25 (NIPS 2012), pp. 1106–1114, 2012.

[11] Christian szegedy, Wei liu, Yangqing Jia et al., "Going Deeper with Convolutions", Conference on Computer Vision and Pattern Recognition (CPVR) , IEEE explorer, Boston, MA, USA, 2015

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

[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, pages3128–3137, 2015.ue 1, Mar 2013, 59-66