International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 06 | June - 2021 ISSN: 2582-3930

Html Code Generation using CNN Algorithm

Harshada Khairnar[1], Prof.K.N.Shedge[2]
Department of Computer Engineering, SVIT, Nashik,Maharashtra India
Assistant Professor, Dept. of Computer Engineering, SVIT, *Nashik,Maharashtra, India SPPU Pune, India*

***Abstract*— Website creation starts with creating rough draft of each web page either with the help of designing tools or drawing by hand. Equivalent code is then generated for draft of web page. Above process is carried out until desired web page is constructed and user is satisfied with the results. This process is repetitive, expensive and time consuming. Hence, proposed system aims to automate this process. Hand drawn image of form is given as an input and it is processed and various components are detected. Once the components are detected they are cropped and these components are recognized using deep learning CNN methods. On recognition of the component equivalent HTML code is constructed using HTML builder algorithm.**

***Index Terms*—Object detection, object recognition, deep learning, HTML**

I. INTRODUCTION

The process of automatic code generation based on the hand drawn mockup images of Graphical User Interface created by designers is expensive in practice. There are various real time software applications that provide facility to generate code for user interfaces such as Android Studio, Xcode and Ecllipse IDE. These software applications are quite expensive and cumbersome. It is not easy to convert user interface drawings into working user interface code as development process requires time and proper skills. At present programmer perform this task manually which is costly and error prone [3]. Developing GUI based application to automatically generate code will help developers to save time on GUI implementation.

Proposed system presents an approach that constructs HTML code for hand drawn GUI image. It involves processes namely detection, dilation, erosion, classification and assembly of an image [4]. The initial process involves detecting borders of various GUI components such as text boxes, labels, drop down list, check box, button, etc. It involves 2 morphological processes: erosion and dilation. Dilation adds pixels to the image that expands the image and erosion removes pixel on object boundaries.

Once the GUI components are detected, they need to be classified. Thus second task is image classification. The proposed system is implemented using RCNN model. RCNN can detect multiple regions of image (up to 2000 different regions). Once object detection and image classification is performed, HTML code will be generated for specific component [3], [4], [6].

II. PROBLEM STATEMENT

To generate HTML code from hand drawn image of web form using object detection and machine learning algorithms.

III. LITERATURE SURVEY

A. Generating Code from GUI Screenshots
Tony Beltramelli, Ulzard Technologies, Copenhagen, Denmark published a paper [6] dedicated to generating code from a GUI Screenshots. A system is used to transform GUI screenshots into computer code. Deep learning methods such as Convolutional and Recurrent Neural Networks were used for automatic generation of code from an input image with accuracy of 77% for platforms like iOS, Android and web-based technologies.

B. Reverse Engineering Mobile Application

Tuan Anh Nguyen, Cristoph Csallner, Computer Science and Engineering Depatment, The University of Texas at Arlington, USA published a paper [3] dedicated to Reverse Engineering Mobile Application. Computer vision and optical character recognition (OCR) techniques are used in REMAUI to identify various elements such as images, texts, containers and lists on a given input. 448 screenshots of iOS and Android applications were used in this system. User interfaces generated by REMAUI were similar to the original screenshots

C. Machine Learning Based Prototyping of Graphical User Interfaces for Mobile Apps

Kevin Moran, Carlos Bernal C‟ardenas, Michael Curcio, Richard Bonett, Denys Poshyvanyk published paper [4] dedicated to Machine Learning Based Prototyping of Graphical User Interfaces for Mobile Apps. This system is used to transform a GUI into code. Initially, components of a GUI were detected from an artifact. Then, GUI components were accurately classified into domain specific types (e.g. toggle-button) using automated dynamic analysis, software repository mining and deep convolutional neural networks. Finally, prototype application can be combined from a GUI structure generated using K-nearest neighbors algorithm. This was implemented for Android system.

IV. EXISTING SYSTEM

The importance of Internet websites have increased due to progress in technology. Websites reflects states, institutions, communities and people which are used for product marketing or advertising purpose. Any website is the interface from which user interacts with the system. Hence, it‟s very important to design interactive web pages which attract attention of users. In existing system, software experts and designers along with the end-users work together for developing design for web page. Draft of design of user interface is created by designers either on paper or graphics tools. Then based on this draft the code for the web pages is developed by software experts. There may be changes in web pages according to the feedback received by the end users and this is repetitive task. Rewriting code for similar components is tedious task. Hence, to automate the production of web pages is all about Code Generation using CNN Algorithm.

V. PROPOSED SYSTEM

The system is used to digitize the work of generation of web form. Hand-drawn image of a web form is given as an input to the system and equivalent HTML code is to be generated.

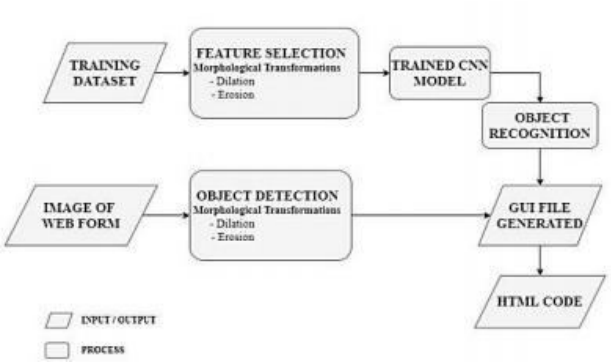

Fig 1: System Overview

A. Object Detection

Dataset [5] consists of various components like textbox, buttons, and labels. These components in an image are detected using contour detection and morphological transformation. Contour detection is used to draw borders outside the component. Small objects are removed using morphological erosion so that only substantive objects are obtained. Gaps in objects are filled using morphological dilation so that objects become more visible. These techniques make object more visible and clear.

B. Object Recognition

The dataset is trained using Convolutional Neural Network. Input image of web form is then given to this trained model which recognizes components in an image.

C. Generate DSL Code with .gui File Extension

Figure 2 shows the content of DSL code with .gui file extension of an image of a web form. This .gui file is generated for the recognized image which contains information about various components in an image. This file gives a base to develop a HTML code for the given form.

```
<START>
header
{
btn-inactive , btn-inactive
}
row
{
quadruple { small-title , text , btn-orange }
quadruple { small-title , text , btn-orange }
quadruple { small-title , text , btn-orange }
quadruple { small-title , text , btn-orange }
}
row
{
quadruple { small-title , text , btn-orange }
quadruple { small-title , text , btn-orange }
quadruple { small-title , text , btn-orange }
quadruple { small-title , text , btn-orange }
}
<END>
```

Fig 2: Format of .gui file

D. HTML Code Generation

HTML code is then generated using the .gui file generated in previous step for the recognized components. The aim is to encode the recognized components based on the web page. First, templates for a header and footer are created. Then detect number of items on each row and map the labels of the components with their codes. Finally, combine the header, body and footer.

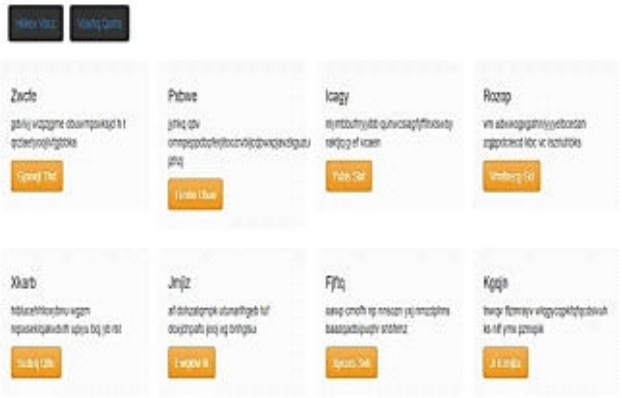

Fig 3: HTML code generated

ADVANTAGES

1. Quick Development of front-end of website or mobile GUI.
2. Cost and time effective as time required to develop the front-end is minimized.
3. Easy to process digital image.

LIMITATIONS:

1. Input should be an image.
2. Attractive designs in input image are difficult to detect and create.
3. The system may not able to detect all GUI components.

VI.ALGORITHM DETAILS

Algorithm 1 - 1 HTML Builder Algorithm

The aim of an algorithm is to recognize the components detected from mock up im- ages and to encode them according to the web page hierarchy. The images are pro- cessed using a deep neural network model involving convolutional neural networks that is used to train the data. After detection of proper components from image, structured **HTML** code is obtained for an input image. Working of an Algorithm –

1.Created the templates for a header and a footer.

2.Detected number of items on each of the rows with coordinates of the compo- nents.

3.Map the labels of the components to their template codes.

4.Body section of the HTML code is obtained.

5.Finally the header, body and footer sections need to be combined.

Algorithm 2 – CNN

CNN stands for Convolution Neural Network. CNN works by extracting features from images based on single or

multiple classifier. CNN involves 2 steps:

1. Feature Extraction
2. Classification

VII.IMPLEMENTATION

OVERVIEW OF PROJECT MODULES

Module 1 - Registration and Login

In this module the user will first register to the system. System should allow only registered user to log into the system.

Module 2 - Object Detection

In this module various components in an image are detected using image process-ing techniques like Contour Detection andMorphologicalTransformation such as Erosion and Dilation.

Module 3 - Recognize the Component

The detected objects are recognized using CNN model and also equivalent .GUI file is generated

Module 4 - HTML Code Generation

HTML code is then generated using the .GUI file generated in previous module forthe recognized components.

VIII.CONCLUSION

Website is an important part in this growing world of technology as it is useful for product marketing and advertising purpose. Hence, it is important to build a website which will attract large number of users. In existing system building a website is a time consuming process. The proposed system takes hand drawn images of web form and convert it into HTML code. Hence, the system will consume less time and resources as compared to existing system.

REFERENCS

[1] Tolga Ensari, Alper Sezen Mustafa Dagtekin, “Automatic HTML Code Generation from Mock-up Images Using Machine Learning Techniques”, IEEE, 2019.

[2] S. Natarajan and C. Csallner, “P2A: A Tool for Converting Pixels to Animated Mobile Application User Interfaces”, IEEE, 2018

[3] Tuan Anh Nguyen, Christoph Csallner, Computer Science and Engineering Department, The University of Texas at Arlington, USA, “Reverse Engineering Mobile Application”, IEEE, 2015

[4] Kevin Moran, Carlos Bernal-C,,ardenas, Michael Curcio, Richard Bonett, Denys Poshyvanyk, “Machine Learning-Based Prototyping of Graphical User Interfaces for Mobile Apps”, IEEE, 2018

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

[6] Tony Beltramelli, UIzard Technologies, Copenhagen, “Generating Code from a GUI Screenshot“, 2017.

[7] 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.

[8] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2625–2634, 2015.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional

Neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[10] John Canny. 1986. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8, 6 (Nov. 1986), 679–698.

[11] R. Plamondon and S.N. Srihari. 2000. Online and off-line handwriting recognition: A comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 1 (Jan. 2000), 63–84.

[12] ZhouWang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing 13, 4 (April 2004), 600–612.

[13] V. Lelli, A. Blouin, and B. Baudry, "Classifying and qualifying gui defects," in 2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST), April 2015, pp. 1–10.

[14] R. N. Zaeem, M. R. Prasad, and S. Khurshid, "Automated generation of oracles for testing user-interaction features of mobile apps," in Proceedings of the 2014 IEEE International Conference on Software Testing, Verification, and Validation, ser. ICST "14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 183–192