Automatic generation of HTML code from hand-drawn images using machine learning techniques

Asha G[1], Chitralekha S[2], Keerthi Basavaraj N Math[3], Latha A[4]
[1,2,3] Students,CSE Department,Sri Krishna Institute of Technology, B'lore-560090, India
[4] Faculty CSE Department,Sri Krishna Institute of Technology, B'lore-560090, India

ABSTRACT:

A new area of intersection between machine learning and website development is the automatic generation of HTML code from hand-drawn images. This paper describes a modern way to create HTML code from hand-drawn designs. Unfortunately, developers may not be able to properly comment on their code due to lack of effort, lack of understanding, ignorance of the importance of code generation, or other reasons. As a result, the code may be poorly directed or inconsistent with the source code, making the software difficult to understand, reuse, and maintain. Developers are interested in automatically generating base code to address these code comment generation issues. The main purpose of this investigation is to examine the automatic generation of HTML code.

Keywords: Hand drawn designs,HTML code generation, website Development, Machine Learning.

I. INTRODUCTION:

Automatically generating HTML from hand-drawn sketches is a difficult endeavor due to the ambiguity and complexity required to decipher the user's intent. In recent years, researchers have proposed various approaches to address this issue. The purpose of this literature review is to provide an overview of the state of the art for automatically generating HTML code from hand-drawn sketches. Internet websites can speed up your daily work.

Among other things, implied mobile applications have in-screen animations and transitions between screens. This paper's drawback is that it cannot be used to produce HTML scripts.

Websites play an important role in today's organizations, societies, and many other industries. There are websites in every industry, including education to learn in the field of training, games, and many more. In this work, an algorithm for automatically producing HTML code for a hand-drawn mock-up of a web page was developed. It aims to identify the mock-up drawing's developed components and encode them in accordance with the web page hierarchy.

When the GUI components are found, they must be categorised. The second step is to classify the images. The RCNN model is used to implement the suggested system. RCNNcan identify several picture areas (up to 2000 different regions). HTML code will be created for a particular component when object identification and picture categorization have been completed.

The procedure Redraw converts mock-ups of mobile application displays into a structured XML code. Computer vision techniques are used to identify individual GUI components at the first stages of their implementation. The second stage consists of the identified components' placement according to their intended use, such as the toggle button, text area, etc.

II. METHODOLOGY:

Machine learning-based prototyping released in 2018 used CNNs and K nearest neighbors. This whitepaper introduces an approach to automating this process by allowing her to accurately prototype her GUI through his three task:Detection,classification,assembly. Limitation of this paper is that Both the detection of objects and the generation of HTML scripts are inappropriate.

Mobile Application Conversion, released in 2018, used computer vision technology methodology. P2A tools use computer vision techniques to develop animated mobile applications. Derived from mobile application screen design, P2A creates the source code for the application's user interface and other materials prepared to assemble and run on a smartphone. This paper drawback is that it is not able to generate HTML scripts.

System design informs us about system functionality from the user's perspective. It contains four important modules.

Send UI sketches or screenshots: In this module, the user sketches her interface and uploads the resulting image to the program.

Areas of interest: In this module, the system acquires the specified image and extracts the region of interest.

Element recognition: In this module, the system uses her RCNN model to identify HTML elements.

Released in 2019, Convolutional Neural Networks (CNN) for image detection and recognition is the technology used in this white paper. Models are developed to evaluate the performance of CNNs on image recognition and recognition datasets. The method is put into practice and its performance is evaluated on the MNIST and CIFAR-10 datasets. There are 99.6 curated models that use MNIST, and CIFAR-10 uses real-time data augmentation and impairments on CPU units, but this document has some limitations. Only suitable for object detection. Writing HTML scripts is inappropriate.

Released in 2021, automatic HTML code creation from mockup photos using machine learning approaches. The technology in this paper is CNN. This study aims to create code automatically from handwritten datasets. The suggested system uses a number of deep learning algorithms to analyze hand-drawn pictures using computer vision techniques. 96% accuracy and 73% validation accuracy are attained by the system. This paper's restriction to specified dataset photographs makes it unsuitable for use with your own images one of its weaknesses.

For our system survey we have referred various survey paper of different years ,Google scholar ,IEEE explorer many more in order to develop a project which can be useful for many users who doesn't have a minimum knowledge about the coding in our project we are using various algorithm like RCNN in order to find region of interest, CNN for element recognition and RESNET 50.

III. IMPLEMENTATION

1. **Image Capturing:**
In this module we are drawing web page designs in a page and taking that image using the camera.

2. **Preprocessing of Image:**
In this module we are using RGB_TO_GRAY function in opencv convert the color images into Gray color.

3. **Building and Prediction of HTML elements:**
In this module we are passing the gray image as input and we are building the RESNET-50 Model to predict the HTML elements.
   - We are building the model using following layers.
   - opt = SGD(lr=0.01,momentum=0.7)
   - resnet50_x= Flatten()(ResNet50_model.output)
   - resnet50_x= Dense(256,activation='relu')(resnet50_x)
   - resnet50_x= Dense(8,activation='softmax')(resnet50_x)
   - resnet50_x_final_model= Model(inputs=ResNet50_model.input, outputs=resnet50_x)
   - resnet50_x_final_model.compile(loss= 'categorical_crossentropy', optimizer= opt, metrics=['acc'])

4. **Generation of HTML code:**
In this model We are collecting the predicted HTML elements and we will generate the HTML code.

Pseudo Code:

```
1. Collect the results from previous module
2. Generate the HTML code with the help of HTML code generator.
3. Save Generated HTML
4. Execute the HTML code
5. Display HTML page
```

IV. DATA FLOW DIAGRAM:

Level-0

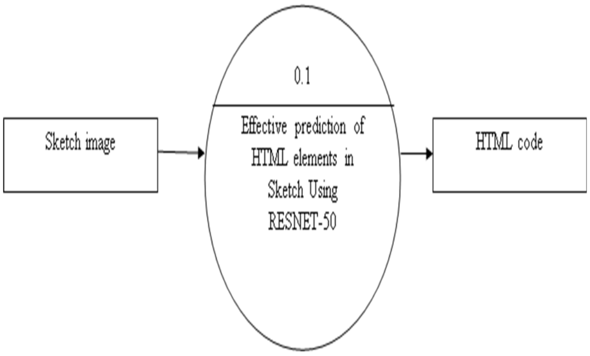
Sketch image
0.1
Effective prediction of HTML elements in Sketch Using RESNET-50
HTML code

Fig : 1.1

The above diagram represents the overall process of the project. The input is a sketch image. The system predicts the HTML elements in the sketchand generates the resulting HTML code using the RESNET50 deep learning algorithm.

Level-1

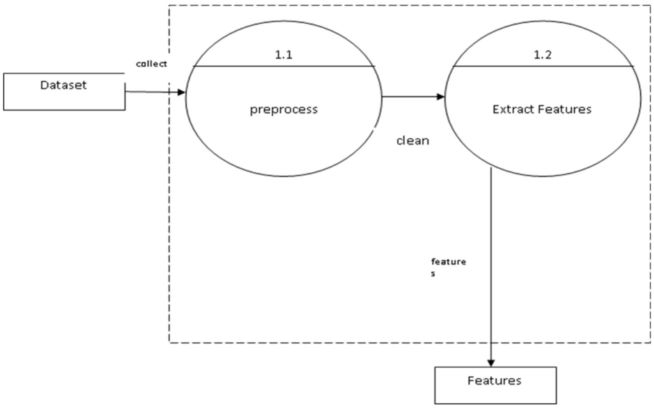

Fig : 1.2

The above diagram shows the first step of the project. As input, use a dataset of sketch images. The system pre processes the data and uses image processing techniques to extract importan telements.

Level-2:

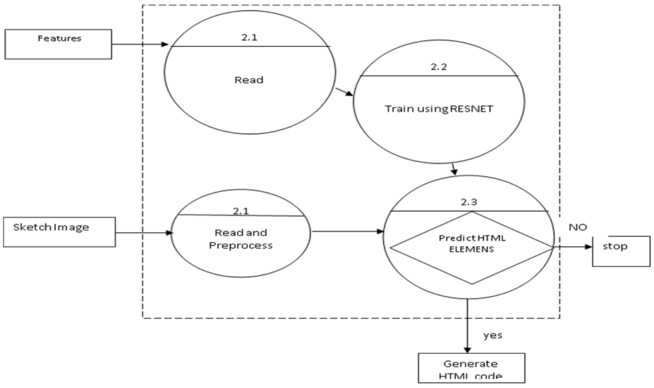

Fig : 1.3

The above diagram describes the final stages of the project. Use level 1 features and a sketchimage as input. The system applies the Resnetclassifier and generates HTML code.

V. RESULTS:

After a review of previous papers from various years, each work aims to automatically produce HTML code from hand-drawn graphics using a different technique. After a review of previous papers from various years, each work aims to automatically produce HTML code from hand-drawn graphics using a different technique. Many projects failed to generate automatic html code for the own images of the users according to their expectations and also suitable only for the specific datasets that has been trained and doesnot able to generate a code for HTML scripts. But in our project we are using RESNET 50 which is 50 layer deep which can browse the significant code in depth and can produce a specific HTML code according to the user expectations. Investigative techniques include breaking down drawings into their component pieces, looking fortext and pictures, and compiling layout information.

Others have concentrated on exploring different feature extraction methods and modelling their architecture in order to increase the accuracy of machine learning models.

For example, rule-based techniques and natural language processing are two of his examples of hybrid systems that integrate machine learning with other feature extraction techniques and model architectures. These hybrid strategies incorporate heuristics and domain-specific information to improve the accuracy of the generated HTMLcode.

VI. CONCLUSION:

To summarise, it is a challenging but crucial task for web design automation to generate automated HTML code from hand-drawn designs. Although machine learning algorithms are the most popular approach presently being utilised to address this issue, there is still potential for improvement in terms of accuracy and efficiency. Hybrid strategies that incorporate machine learning with other techniques may hold the key to the future of this discipline. Overall, by offering adetailed overview of the most recent techniques for automatically creating HTML code from handwritten drawings, this literature review servesas a helpful tool for academics and industry professionals working in the subject.

VII. ACKNOWLEDGEMENT

We would like to thank,<Latha A> & <Dr.Shantharam Nayak> for their valuable suggestion,expert advise and moral support in the process of preparing this paper.

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] 19Sep 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 InternationalConference 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]. Shweta Patil1, Rutuja Pawar, Shraddha Punder , Jacob John Generation of HTML Code using Machine Learning Techniques from Mock-Up Images Department of Computer Engineering, Pillai HOC College of Engineering and Technology, Rasayan,
India , Journal 2021

[10].Harshada Khairnar,rof.D.S.Thosar, Prof.K.N.Shedge HTML Code Generation using CNN Algorithm Department of Computer Engineering, SVIT, Nashik,Maharashtra India Assistant Professor,Dept.of Computer Engineering,SVIT, Nashik,Maharashtra, India SPPU Pune, India Journal 2021.