Automatic HTML Code Generation from Hand Drawn Images using Machine Learning Methods

Dr. Malatesh S H[1], Miss. Bhavanashree R[2], Miss. M Keerthana[3], Miss. Monisha B[4], Mr. Vishwanath R, Dr. Aruna M G[6]

[1]*Professor and Head of Department, Department of CSE, M. S. Engineering College, Bangalore, India*

[2,3,4,5]*Student, Department of CSE, M. S. Engineering College, Bangalore, India*

[6]*Associate Professor, Department of CSE, M. S. Engineering College, Bangalore, India*

***Abstract*—The production of individual web page mock-ups, which can be done by hand or with the aid of graphic design and professional mock-up creation tools, is the first stage in the website design process. The mock-ups are then converted into structured HTML or similar mark-up code by software engineers. Typically, this method is performed multiple times until the required template is achieved. The purpose of this review is to make the process of developing code from hand-drawn mock-ups more automated. Hand drawn mock-ups are processed using computer vision techniques, and the recommended system is built using deep learning techniques.**

The building of a preliminary drawing of each web page, which can be done using design tools or by hand. After that, corresponding code for the web page draught is written. This procedure is difficult, expensive, and time-consuming. Consequently, the suggested system will automate this operation. A hand-drawn drawing of a form is provided as input, which is analysed, and several components revealed. After the components have been discovered, deep learning CNN techniques are used to crop and recognize them. When the matching component is identified.

I. INTRODUCTION

Because of today's technological advancements, the relevance of Internet web pages has expanded significantly. Nowadays, websites represent the personalities of nations, institutions, communities, and individuals. There are websites for practically any subject, from information to social work, games to training, and so on. Companies' websites are brought to the forefront for financial objectives, such as product promotion or advertising. Developers are in charge of creating client-side software based on a designer's mock-up of the Graphical User Interface (GUI). The implementation of GUI code, on the other hand, requires time and diverts engineers' attention away from the product's core functionality and logic. In addition, the computer Furthermore, each target runtime system's computer languages for designing such GUIs are separate, resulting in onerous and repetitive effort when the product is meant to function on several platforms using native technologies. In this research, we present a model that has been trained end-to-end with stochastic capable to reduce to construct variable-length tokens strings from a single GUI picture given input, yet staying technically and informational. An algorithm was created in this work to automatically produce HTML code for a hand-drawn mock-up of a web page. Its goal is to detect the mock-up picture's components and encode them according the web page structure.

II. EXISTING SYSTEM

Transforming a graphical user interface screenshot created by a designer into computer code is a typical task conducted by a developer in order to build customized software, websites, and mobile applications. In this paper, we show that deep learning methods can be leveraged to train a model end-to-end to automatically reverse engineer user interfaces and generate code from a single input image with over 77% of accuracy for three different platforms (i.e. iOS, Android and web-based technologies).

Sketch2Code uses AI to convert hand-written drawings to working HTML prototypes. Designers share ideas on a whiteboard, then changes are shown in the browser instantly. We can use Computer Vision to build a system that understands what a designer has drawn on a whiteboard, then translates that understanding to HTML code. This way we can generate HTML code directly from a hand-drawn image.

A. Disadvantages of Existing System:

In the CNN based application accuracy is less than 75%.Sketch2Code uses the object detection API and Microsoft API which is more expensive.

III. PROPOSED SYSTEM

As a first step of designing of website is start to built the mock-up images for the particular web pages by operated with the hands or using mock-up developer tools. It is efficiently used for the developer to transferring web pages mock-up to the coding. It's generating the proposed system to creating the wireframe to the layout interfaces there are two techniques mostly used first is computer vision and second is deep systematic analysis. The automatic code generation is time reducing and cost effective. We have design structured an outline the design.

A.Advantages:

It gives 95% and above Accuracy. Our application is developed using FRCNN it consumes less time.

IV. SYSTEM ARCHITECTURE

The system "design" is defined as the process of applying various requirements and permits it physical realization. Various design features are followed to develop the system the design specification describes the features of the system, the opponent or elements of the system and their appearance to the end-users. A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and nonfunctional requirements, and may include a set of use cases that describe user interactions that the software must provide.

It is very important in a SRS to list out the requirements and how to meet them. It helps the team to save upon their time as they are able to comprehend how are going to go about the project. A SRS can also be defined as a detailed description of a software system to be developed with its functional and non-functional requirements. The software requirement specification document is consistent with all necessary requirements required for project development. To develop the software system, we should have a clear understanding of Software system. To achieve this, we need continuous communication with customers to gather all requirements. A good SRS defines how the Software System will interact with all internal modules, hardware, and communication with other programs and human user interactions with a wide range of real-life scenarios. It is very important that testers must be cleared with every detail specified in this document in order to avoid faults in test cases and its expected result.

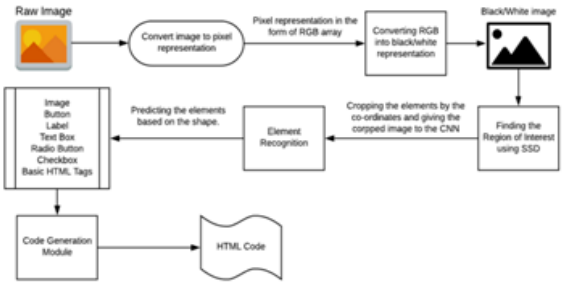


Fig 4.1 System Architecture

V. METHODOLOGY

The technology is used to automate the web page creation process. The system is given a hand-drawn image of a web form as input, and corresponding HTML code is to be created. There are four key phases to any process of developing robotic HTML code. The first step is object detection, which involves scanning the picture and using image processing algorithms such as LSTM, DSL, and so on. After then, the objects which have been identified are snipped. Eventually, the outcome of the model is converted to HTML code. 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 customize the CSS for the website's front end.

Gathering Dataset: Data is one of the most valuable resources today's businesses have. The more information you have about your customers, the better you can understand their interests, wants and needs. This enhanced understanding helps you meet and exceed your customers' expectations and allows you to create messaging and products that appeal to them.

Image Capturing: In this module we are drawing web page designs in a page and taking that image using the camera. It refers to the process of acquiring or collecting images to be used as training data for training a machine learning model. Image capturing is a crucial in computer

vision tasks, where the model learns to understand and interpret visual data.

Pre-processing of Images: Pre-processing of images refers to a set of operations or transformations applied to the images before they are used as input to a machine learning model. These operations aim to enhance the quality, standardize the data, or extract relevant features to improve the model's performance. Preprocess the images before feeding them into ResNet-50. This typically involves resizing the images to the input size expected by the model and applying normalization techniques. In this module we are using RGB_TO_GRAY function in opencv convert the color images into Gray colour.

Building and Prediction of HTML elements: This mapping could be based on specific conditions or algorithms that determine the appropriate HTML element based on the input data. In this module we are passing the gray image as input and we are building the RESNET-50 Model to predict the HTML elements.ResNet-50 is a deep learning model primarily used for image classification tasks rather than directly building or predicting HTML elements.

Generation of HTML Code: Using JavaScript or any other server-side language, generate or dynamically modify the HTML code to include the desired elements. In this model We are collecting the predicted HTML elements and we will generate the HTML code.

VI. IMPLEMENTATION

Implementation is the stage of the project where the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Modules:

1. Image Capturing
2. Pre-processing of Image
3. Building and Prediction of HTML elements
4. Generation of HTML Code

Module Descriptions:

1. Image Capturing:

Image processing is a field of computer science and digital signal processing that focuses on analyzing and manipulating images. It involves using various algorithms and techniques to enhance, transform, or extract information from digital images.

There are several common tasks in image processing, including:

Image processing techniques can be applied in various domains, including medical imaging, surveillance systems, remote sensing, robotics, and entertainment (such as special effects in movies or image editing software). Various software libraries and tools, such as OpenCV, MATLAB, or Python libraries like Pillow and scikit-image, are commonly used to implement image processing algorithms. In this module we are collecting cocoon dataset from the kaggle.com 2. Preprocessing of Image:

In this module we are using RGB_TO_GRAY function in opencv convert the color images into Gray color.

Pseudo Code:

1. Input the image.
2. Read Image.
3. Convert to gray.
4. Normalize the image.
5. Save gray imag

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 image elements**.** ResNet-50 is a deep learning model primarily used for image classification tasks rather than directly building or predicting HTML elements. However, you can leverage ResNet-50 or any other pre-trained image classification model to perform tasks related to HTML elements indirectly.

Pseudo Code:

1. Input the preprocessed image.
2. Build the RESNET50Architecture.
3. Train the dataset using .fit()
4. Save the model using save()
5. Predict the HTML elements using predict()

4. Generation of HTML Code

ResNet-50 is a deep learning model primarily used for image classification tasks rather than directly building or

predicting HTML elements. However, you can leverage ResNet-50 or any other pre-trained image classification model to perform tasks related to HTML elements indirectly.

VII. DATA FLOW DIAGRAM

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart. 7.1 Data Flow Diagram - Level 0

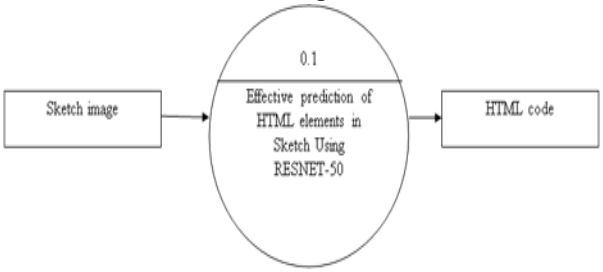


Fig.7.1 DFD-Level 0

Level: 0 describes the overall process of the project. We are using image as input. System will use a deep learning algorithm to predict the disease in the silk waroms.

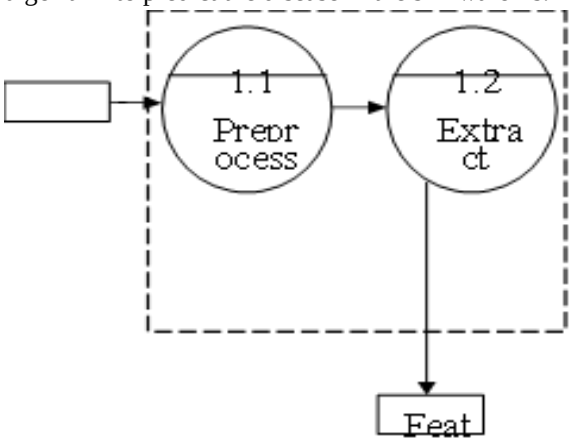


Fig.7.2 DFD-Level 1

Level: 1 describes the first step process of the project. We are passing silkworm image as input. System will read and preprocess the image and extract the color co-relations in pixels then extract the image features.

7.3 Data Flow Diagram - Level 2

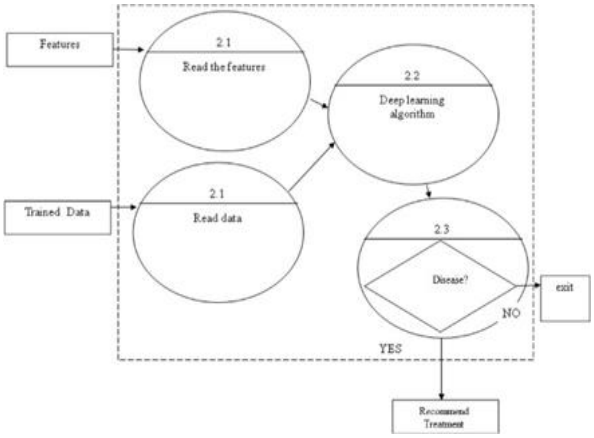
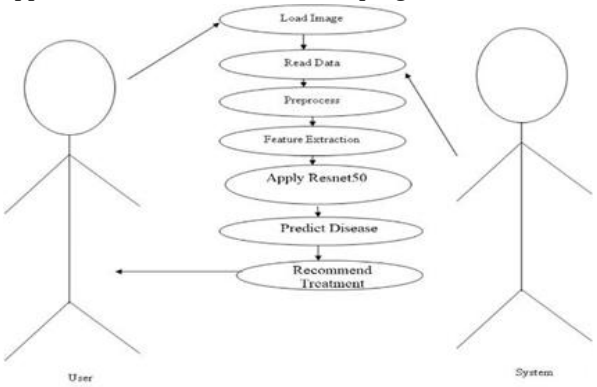


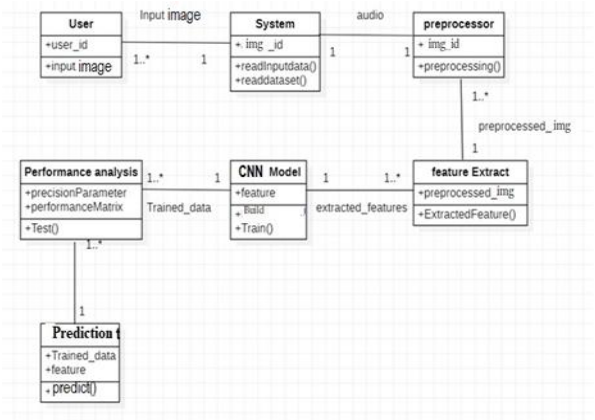
Fig.7.3 DFD-Level 2

Level: 2 describes the final step process of the project. We are passing extracted image features from level 1 and trained data as input. System will read features and load the trained model Detect types of disease and recommend the treatment of the disease.

7.4 Use Case Diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures. In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The class diagram is the main building block of object-oriented modeling Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.





7.4 Use Case Diagram

VIII. CONCLUSION

A critical aspect has been converting website mock-ups into mark-up code with less time and development expense. We built a technique in this work that receives web page mock-ups, processes them, and generates structured HTML code.

A collection of images was used, which included several mock-ups of web page architectures. Future research and expertise are needed to improve the code generation's efficiency. Such apps are needed in the industry and in our dailylives since every firm insists on making their laborious task easier and more efficient. The above study presents a system that turns picture data into an HTML webpage automatically.

REFERENCE

[1] Srinivas B, Khushi Kumari, Goverdan Reddy H, Niranjan N, Hariprasad S A and Sunil M P, "IoT based Automated Sericulture System": International journal of recent technology and engineering, July 2019.

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

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

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

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

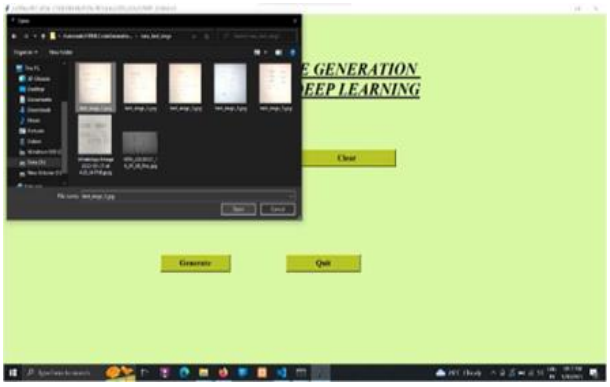
[6]. Norhidayu binti Abdul Hamid, Nilam Nur Binti Amir Sjarif, "Handwritten Recognition Using SVM, KNN and Neural Network", www.arxiv.org/ftp/arxiv/papers/1702/1702.00723

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

[8]. 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/

[9]. 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=31 97231.3197249.

IX. APPENDIX




Select Image
Label: 99%
TextBox: 99%
Label: 99%
Radio
Label: 99%
RadioButton: 99%
CheckBox
Label: 86%
Button: 99%