

Factors Involved in Artificial Intelligence-based Automated HTML Code Generation Tool

1st Sanjay Agrawal
Dept. of Computer Engineering,
MMIT, Lohgaon, Pune, India
sanjay.agrawal@mmmit.edu.in

2nd Swapnil Suryawanshi
Dept. of Computer Engineering
MMIT, Lohgaon, Pune, India
Swapnil.suryawanshi@mmmit.edu.in

3rd Vikas Arsude
Dept. of Computer Engineering
MMIT, Lohgaon, Pune, India
vikas.arsude@mmmit.edu.in

4th Nilesh Maid
Dept. of Computer Engineering
MMIT, Lohgaon, Pune, India
nilesh.maid@mmmit.edu.in

5th Madhuri Kawarkhe
Dept. of Information Technology,
Jawaharlal Nehru Engg. College, MGM University,
Aurangabad, India
madhurikawarkhe@gmail.com

Abstract—Automated code generation or automatic programming is a widely researched topic these days. The auto code generation fascinates the young researchers towards this field more than ever. The Automatic code generation is the process to generate the valid programming code in various languages and makes programming easy for the next generations. Automatic coding is to build a machine which can write code. The code writing process involves several wearisome steps tearing down a process into small instructions, allotting specific memory locations or to specify the system interrupts. It can also be able to manage the input-output buffers and many other things. After observing these steps as mentioned above, we implement mathematical routines, a subroutine library, and sorting programs. In this paper, we look closely towards the factors involved in Automatic programming. We studied more extensive programming and code writing process to understand how these factors affect the process and which can help anyone build a better machine for auto programming. As we programmed manually first we observed the process and tried to invent ways to abstract these steps to combine them into a higher-level programming language. This includes the development of interpreters, assemblers, compilers, and finally, the automatic code generator programs designed to operate with the process as mentioned earlier, that is, automatic programming.

Keywords— *Web development, Artificial Intelligence, Deep Learning, Machine Learning.*

I. INTRODUCTION

We are living in an era of tech where the technology we have access to is way too advanced than what we used to have in the 90s era. The advancement in the area of automation is also no expectation. Automation is used in almost every industry these days, whether it is manufacturing industries, Medical industries, or Auto industries.

Automatic programming or the automatic code generation has been a goal of computer science artificial intelligence (AI) since the first programmer came face to face with the challenges of programming. As befits such a long-term goal, it has been a moving target continually shifting to reflect the growing expectations. Much of what was initially conceived of as automatic programming was

achieved long ago in 1958. On the other hand, current expectations regarding its potential are often based on an idealised view of reality and are probably unachievable. Nevertheless, several significant developments are appearing in research efforts and commercially available systems.

With the help of advanced technologies like Artificial Intelligence and Machine Learning in our tool, we have been able to achieve a continuous upgrading of the overall performance for recognising and conversion of user's input. The process of training of our tool begins with notes or data, to search for shapes and text in drawings and make a healthier decision. In this proposed automation technology concept, we will use various applications of machine learning and artificial intelligence. Here it will take a hand-drawn design or prototype as input and generated the front-end code in HTML [1] So at the end of the discussion which can be directly evaluated and further validate outputs by the engineers, and data scientists. This paper summarises the factors involved in the automatic code generation process.

II. RELATED WORK

Recently proposed framework expresses that, picture subtitle age has been for quite some time saw as a connect to associate Natural Language Processing and Computer Vision [2]. Ordinarily, this issue was tackled by profoundly handmade frameworks. The work in the related papers concentrated on the essential item, properties and relational words identification in pictures and afterwards utilises conceivable realistic models to comprehend related subtitles [2]. When creators connote imaged by using object recommendations and properties, the ground-breaking language parsing models to produce inscriptions are applied at that point. In any case, the predefined formats and language punctuation limit crafted by the frameworks above. Investigation of the source code has been one of the center research fields of programming building. For example, much research has been led to finding symmetric parts from source documents, and mining code vacillates in source code catalogues [3]. Source code of proposed programming design invariably contains auto-created code, i.e., parser generators yield. The dynamic auto-produced code once in a while hampers source code study. Consider, if we distinguish code twins from source code containing auto-created code, a

reasonable number of code twins are found from auto-produced code. Code twins saw from hand-created code ought to be paid no notification on the grounds of that their check is quite littler than the code twins found from auto-produced code. That may lead designers to lose a portion of the vital examines by twining identification apparatuses. We can likewise consider mining source code storehouses. This framework tracks code components; for example, classes and techniques are through many cycles. The dynamic auto-produced code may require more opportunity to get code components. The code remarks, for the most part, speak to themselves as they are auto-produced code. In this way, we can utilise apparatuses as UNIX GSrep to discover and evacuate the auto-created codes. Be that as it may, developers do erase those code remarks early. Checking records individually physically is the best way to investigate and wipe out the auto-created code barring code remarks.

III. SYSTEM ARCHITECTURE

To improve the effectiveness of the web advancement process by diminishing the remaining task at hand toward the front-end developer. The underlying advancement stages expect clients to draw many wire-frames before concluding the last structure. The concluded wire-frame is the contribution to the framework in Image format. the proposed framework will take a screen capture of the wire-frame as an input to the framework, and produce an auto-created code from that structured mock-up and convert them into produced code record as the yield to the end-client.

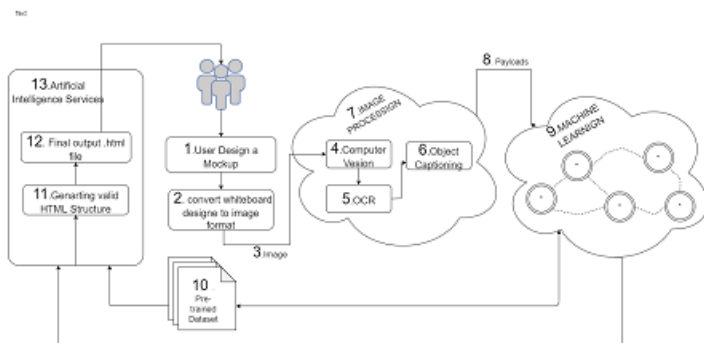


Fig. 1. System Architecture Diagram

Text represented in the figure 1,

1. User designs a mockup
2. Convert mockup into image format
3. Image
4. Computer Vision
5. OCR
6. Object Captioning
7. Image Processing
8. Payloads
9. Machine Learning module

10. Pre-trained datasets.
11. Generating a valid HTML Structure
12. Final output .html file
13. Artificial Intelligence module

IV. SYSTEM ARCHITECTURE EXPLANATION

Here we are using Deep Learning and Artificial Intelligence services to understand the handwritten mock-ups, and translate them into the HTML code. The user will first provide a screenshot with the design mock-ups as an input to the tool. The image processing will extract the meaningful information from the image. Here, the two sections for "Image Recognition" and "Code Generation" are working as the input provided. In image recognition, the input data is matched with the pretrained datasets in the storage. If the shapes or texts from the input are not recognised by the tool as those are different from the shapes of the components used in HTML, the tool ignores them and processes only the known shapes and text and gives an output for further generation of the HTML structure[2]. The Artificial Intelligence services will take the production of the previous two steps as input. Artificial Intelligence will compare the data received with the pre-trained data models. Once the shapes and text are recognised as valid, then it generates a structure for creating an HTML file. Finally, the HTML code file is generated using the defined attributes by Artificial Intelligence. Let us have a look at different modules used in the tool as represented in figure 2.

A. Image Processing

The image processing module includes the initial programs requires for the tool, which are as follows 1. OpenCV, 2. OCR, 3. Object Captioning. These three works together to extract the content of the uploaded images.[1] OpenCV extracts the shapes and their relative positions in the image and saves it for further processing.

B. Data Prepossessing

Data preprocessing may be a data processing technique that involves transforming data into a clear format. Real world data is usually incomplete, inconsistent, and lacking in certain behaviours or trends, and is probably going to contain many errors. Data preprocessing may be a proven method of resolving such issues. Data preprocessing prepares data for further processing. Data preprocessing is employed database driven applications like customer relationship management and rule-based applications (like neural networks).

C. Machine Learning

Machine learning is an application of AI (Artificial Intelligence) that gives tools the power to learn automatically improve from experience without being explicitly programmed. The proposed tool usages the ML module to process and determine patterns of the images uploaded in the tool. The ML module will train the tool for the different models, and the shapes, etc. The data preprocessing is the core responsibility of the tool. [4]

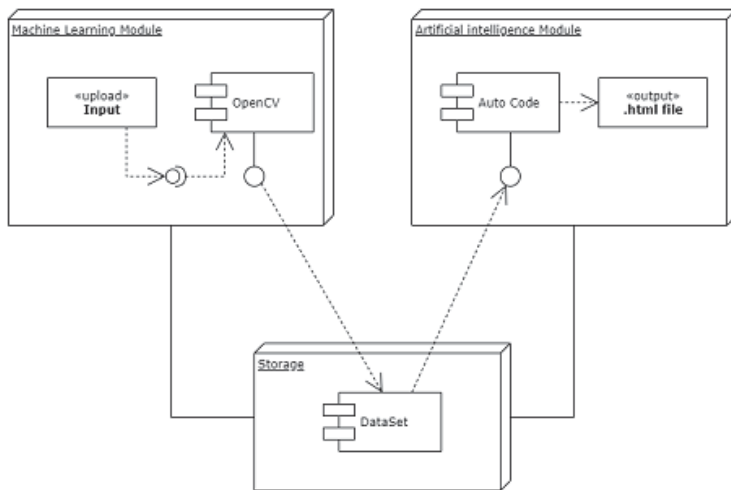


Fig. 2. Proposed Component Diagram

D. Artificial Intelligence

The AI module accepts the extracted and processed data by the ML module as the input. The AI module uses pretrained Datasets to work with the data. AI modules use algorithms along BLEU scores algorithms to compare the data with verification set and determine the accuracy of the processed data. The AI module can create the .html code with the most accuracy.[2] The Artificial neural network helps the AI module to build the .html code and validate it with the standard norm of the HTML programming by w3school.[7]

V. ALGORITHMS USED

- Bilingual Evaluation Understudy Score.
- Image Captioning.

A. Bilingual Evaluation Understudy Score.

The Bilingual Evaluation Understudy Score, or BLEU for short, is a measurement for assessing a produced sentence to a reference sentence. An ideal match brings about a score of 1.0, while a perfect bungle brings about a score of 0.0. The score was produced for assessing the expectations made via programmed machine interpretation frameworks. It isn't great, yet offers five convincing advantages: It is fast and economical to compute, It is straight forward, It is language autonomous, It associates profoundly with human assessment, After Considering these advantages, the BLEU score algorithm is used to validate the too. The Blue Scores algorithm are used to validate the actual output with the generated output by the tool.

B. Image Captioning

The image captioning algorithms are used in the tool to extract the shapes and recognize the shapes in the uploaded images to the tool. Image Captioning algorithm can help in the process of generating a textual description to caption the elements of the images. It uses both Natural Language Processing(NLP) and Computer Vision(CV) to generate the captions. The image captioning work with the 2D matrix. The algorithm will add a tag to the known shapes such as the text box, Buttons, grids layouts, etc. The captioned data is then provided to the Artificial Intelligence modules to help it generate the valid HTML code.

VI. RESULT ANALYSIS

The proposed tool can generate the desired output. So, we will analyse it here. The tool has three main modules which need to be working in synchronisation for the tool to operate properly. So, let us analyse the Machine Learning module, which helps the proposed tool to work on the initial tasks of image data extractions process. The ML module uses the OpenCV to read the images uploaded in .png format. The OpenCV works very effectively on PNG images, so the .png is the most suitable image format for the tool. The OCR is used to extract the exact text written in the pictures. However, due to its difficulties in understanding different languages. The handwritten text contains lots of variation every user of the tool may have different handwriting which makes it more complex to recognise the text written. Hence, makes the tool inefficient as it requires more processing time. So, we have to skip the OCR for the time being to increase the efficiency of the tool. The ML module is the heaviest one in the tool is requires the most processing power. The sample input given is provided in figure3.

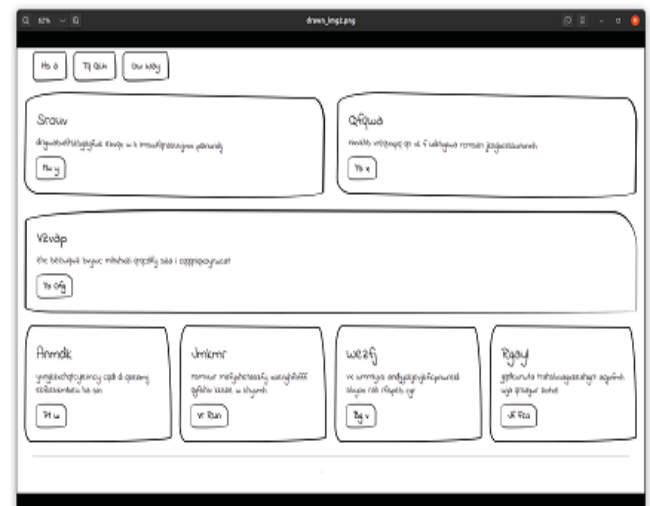


Fig. 3. Sample Input

The ML module can generate the Model weights, and we can use it in the next modules. Model weights are stores in .h format and used further as the pre-trained Datasets for future use. The next part of the proposed tool is the code generating process. The code generation starts with the accepting input from ML module. The ML module can filter the ambiguous data and processes it further with Artificial Intelligence module. The AI module uses pre-trained Datasets to work with the data. AI modules use algorithms along BLEU scores algorithms to compare the data with verification set and determine the accuracy of the processed data. The AI module can create the .html code with the most accuracy, and the new data may contain some ambiguous code. Still, overall, the AI module can work with reasonable accuracy when we compared it with a validation set. The ouput generated by tool is shown in figure 4.

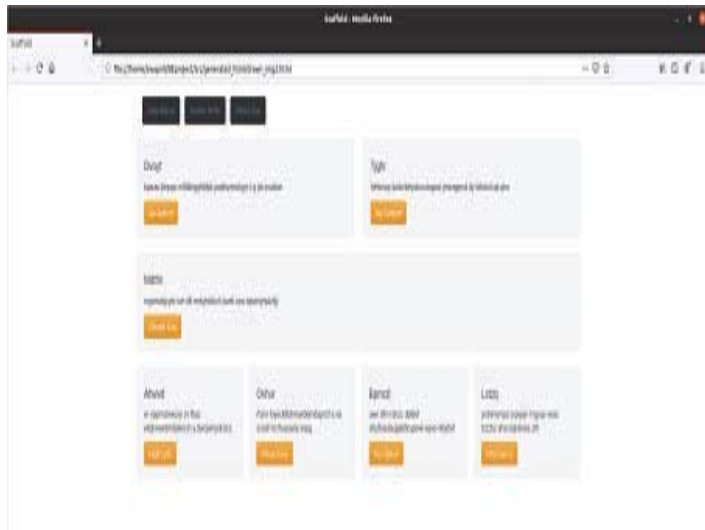


Fig. 4. Sample Output

VII. CONCLUSION

This paper reviews the factors Involved in artificial intelligence-based automated HTML code generation tool and analyse the results. Building an automated system would surely help and simplify the designing process. By using researches in image captioning and machine learning, this tool is able to take hand-drawn design mock-ups and convert them into working HTML code in a few seconds.

REFERENCES

- [1] Agrawal, S. A., Swapnil Suryawanshi, Vikas Arsude, and Nilesh Maid. "Artificial Intelligence based Automated HTML Code Generation Tool using Design Mockups." *Journal of Interdisciplinary Cycle Research* Volume XII, Issue III, March/2020
- [2] Senmao Ye, Nian Liu, Junwei Han, Senior Member, IEEE "Attentive Linear Transformation for Image Captioning" *JOURNAL OF LATEX CLASS FILES*, VOL. 14, NO. 8, AUGUST 2015
- [3] Kento Shimonaka, Soichi Sumi, Yoshiki Higo and Shinji Kusumoto "Identifying Auto-generated Code by Using Machine Learning Techniques" 2016 7th International Workshop on Empirical Software Engineering in Practice (IWESep) 13-13 March 2016
- [4] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.
- [5] Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem "Re-evaluating Automatic Metrics for Image Captioning" *arXiv:1612.07600v1 [cs.CL]* 22 Dec 2016
- [6] MD. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, Hamid Laga "A Comprehensive Survey of Deep Learning for Image Captioning" (Submitted on 6 Oct 2018 (v1), last revised 14 Oct 2018 (this version, v2))
- [7] Jicheng Fu ; Farokh B. Bastani ; I-ling Yen "Automated AI Planning and Code Pattern Based Code Synthesis" *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*0-7695-2728-0/06