STC (SKETCH TO CODE)-AN ENHANCED HTML & CSS AUTOCODE GENERATOR FROM HANDWRITTEN TEXT AND IMAGE USING DEEP LEARNING

SRINATH R [1], SIVA PRASATH K R [2], VARUN RAJ S [3], VINITH W [4], Dr. L. SRINIVASAN[5]

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING Dr. N.G.P. INSTITUTE OF TECHNOLOGY

2024 2nd International Conference on Networking and Communications (ICNWC) | 979-8-3503-6526-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICNWC60771.2024.10537336

ABSTRACT

This paper presents an innovative solution for automating the transformation of hand-drawn sketches into HTML and CSS code. The process unfolds in four distinct phases: Pre-processing optimizes input sketches, Segmentation identifies key regions using Region Proposal Networks (RPN), Feature Extraction employs Line Drawing Prediction (LDP) for detailed interpretation, and Classification categorizes features into HTML and CSS elements. The integrated use of RPN and LDP ensures precise recognition of design elements and layout, streamlining the conversion process. This approach not only enhances accuracy but also enables a seamless transition from creative ideation to web development, reducing the manual effort traditionally associated with coding.

KEYWORD: Code generation, HTML-CSS code and Automating the sketch-to-code and RPN and LDP algorithm.

I. INTRODUCTION

In the dynamic landscape of the digital era, web design has emerged as a focal point in creating captivating and user-centric online experiences. The artistic journey often commences with the simplicity of hand-drawn sketches, yet the challenge lies in seamlessly bridging the gap between artistic inspiration and the technical realization of web layouts. This ambitious endeavor operates at the intersection of computer vision and web development, presenting an innovative approach to convert these sketches into functional HTML and CSS code. The conventional manual process of translating sketches into code is not only time-consuming but also susceptible to errors, acting as a bottleneck that this groundbreaking initiative seeks to alleviate.

Harnessing the power of state-of-the-art machine learning techniques, including the utilization of recurrent neural networks (RNNs) for image feature extraction and sequence-to-sequence models for code generation, the project vows to revolutionize the sketch-to-code conversion process[4]. This empowers the system to accurately interpret intricate hand-drawn wireframes, mock-ups, and sketches, translating them into semantically meaningful HTML and CSS code. Throughout this transformative journey, the core emphasis is on preserving the design intent while adhering to established web standards and responsive design principles.

At its essence, this project strives to offer a seamless solution for web developers and designers alike. By automating the sketch-to-code conversion process, it not only saves valuable development time but also fosters collaboration between creative designers and technical developers, establishing a shared communication platform[8]. Beyond the mere generation of code, this initiative holds the promise of democratizing web design, breaking down barriers and making it accessible to a broader spectrum of professionals, irrespective of their technical backgrounds.

The significance of this project extends beyond its immediate applications, marking a substantial stride in the field of web development. It introduces an intelligent solution that automates the conversion of hand-drawn art into functional web interfaces, thereby redefining the conventional boundaries of creativity and technical execution[1]. This vision transcends the confines of being a mere project; it aspires to shape a future characterized by efficiency, collaboration, and accessibility within the realm of web design.

The automation of the sketch-to-code conversion process is not just a technological advancement; it represents a paradigm shift in the way we approach and perceive web design[15]. As we venture into this new frontier, the integration of cutting-edge machine learning techniques promises to elevate the capabilities of designers and developers alike, unlocking new levels of creativity and efficiency[5]. This initiative serves as a testament to the limitless potential of converging technology and design, paving the way for a future where the synergy platform. Beyond code generation, this initiative holds the promise of democratizing web design, making it accessible to a wider spectrum of professionals, irrespective of their technical backgrounds. It marks a significant stride in the field of web development, introducing an intelligent solution that automates the conversion of hand-drawn art into functional web interfaces. This vision goes beyond just being a project; it envisions a future characterized by efficiency, collaboration, and accessibility within the realm of web design.

II. RELATED WORK

[1] Sketch Recognition and Image-to-Code Conversion:

Research and projects in the field of sketch recognition and image-to-code conversion can offer valuable insights. Techniques using convolutional detection, can provide insights into choosing and adjusting the threshold for recognizing design elements in hand-drawn sketches.

[4] Normalization & Coordinate Transformation:

Techniques used in computer graphics and image processing for normalizing coordinates and transforming them into different coordinate systems can offer guidance on refining the process of converting normalized bounding box coordinates into pixel values.

[5] Web Layout and Responsive Design:

Investigating principles of web layout and responsive design can contribute to ensuring that the generated HTML and CSS code is not only accurate but also adheres to modern web design standards. Considerations for responsiveness across different devices could be explored.

[6] Human-Computer Interaction (HCI):

Exploring HCI literature can provide insights into user experience considerations when translating hand- drawn sketches into web pages. Understanding how users interact with the generated code and metadata can guide user interface design.

III. PROPOSED WORK METHODOLOGY

neural networks (CNNs) for recognizing design elements and converting them into code could be explored.

[2] Object Detection and Semantic Segmentation:

Leveraging methodologies from the field of object detection and semantic segmentation can contribute to improving the accuracy of identifying and localizing design elements within hand-drawn sketches. Models such as Faster R-CNN, YOLO, or Mask R-CNN might be relevant.

1. HTML and CSS Code Generation:

   Investigating existing tools and frameworks for automatic code generation from visual designs can provide a foundation for refining the code generation aspect of the project. Systems like Sketch2React or Anima are worth examining for their approach to turning designs into code.
2. Confidence Score and Quality Assessment:

   Research on confidence scoring and quality assessment in computer vision and natural language processing domains can inform the establishment of confidence scores for code generation. Exploring techniques for assessing the fidelity of generated content could further enhance the reliability of the system.

[3] Thresholding in Computer Vision:

Examining literature on thresholding techniques in computer vision, particularly in the context of object HERE WE HAVE GIVE THE METHODOLOGY WITH THE SAME VIEW OF THE PROJECT:

a. **Understanding Sketches:** This initial stage involves the algorithm's capacity to comprehend and make sense of the intricacies embedded in hand-drawn sketches. The system must possess the capability to interpret the various strokes, lines, and shapes presented in the sketch, gaining a fundamental understanding of the visual representation before proceeding to the next phases.
b. **Unveiling Key Elements:** Following the understanding of the sketches, the algorithm delves into the task of unveiling the key design elements within the sketches. This includes the identification of layout structures, various components, and other crucial elements that contribute to the overall design. It is akin to an object detection
c. process, where the system needs to discern and localize these components accurately to proceed with the subsequent phases.

d. **Planning and transformation:** Once the key design elements are identified, the system moves into the planning and transformation stage. Here, the algorithm takes the recognized design elements and translates them into functional HTML and CSS code. This phase is pivotal, as it bridges the conceptualization of the sketch with the practical implementation on a digital platform, creating a tangible representation of the initial hand- drawn design.

e. **Algorithm Development:** The algorithm development stage leverages the power of deep learning, employing CNNs for effective feature extraction and RNNs for handling complex tasks that involve understanding the sequential and contextual aspects of hand-drawn sketches. This amalgamation of advanced neural network architectures contributes to the overall robustness and sophistication of the system, allowing it to navigate the intricacies of diverse sketches and generate accurate, high-quality HTML and CSS code representations.

Architecture:

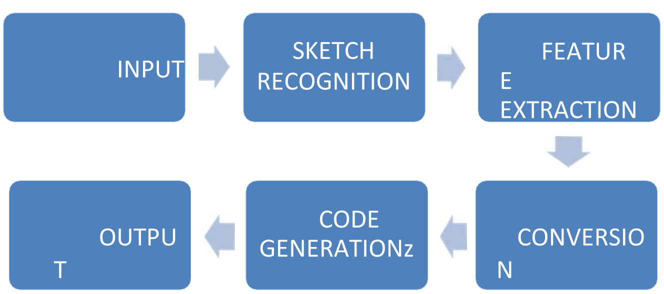

Fig 1 : working of swin transformer algorithm development

Example View:

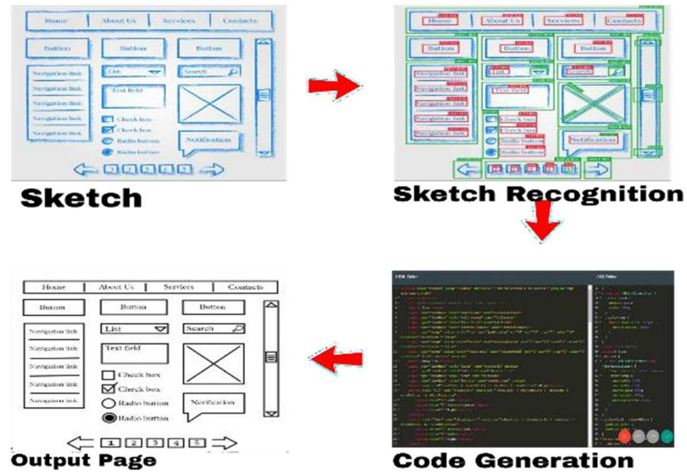

Fig 2: sample of the output with the entire process

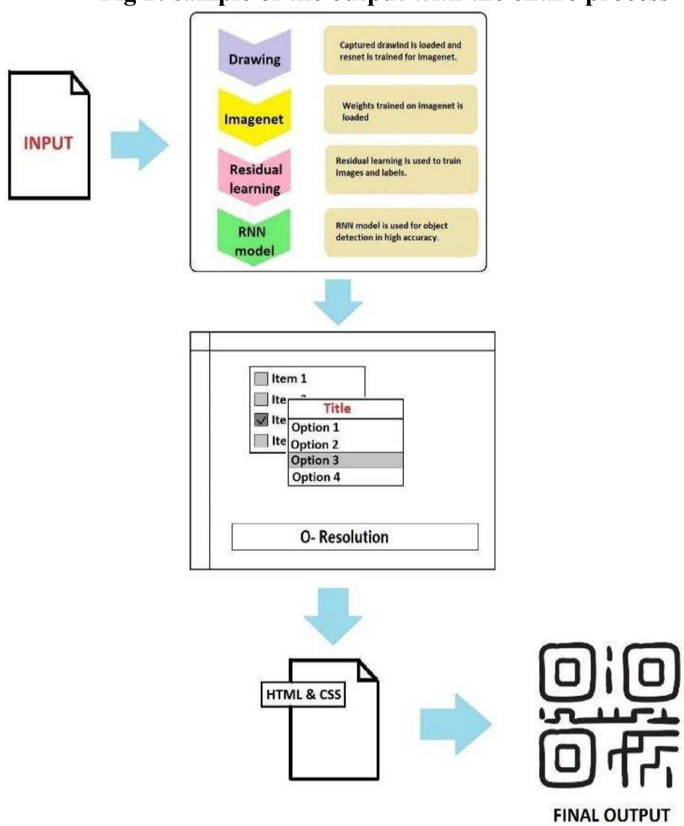

Fig 3 : Backend process of swin transformer algorithm

EXPLANATION:

The initial phase of our system involves the intricate task of analyzing and understanding hand-drawn sketches, specifically focusing on the discernment and localization of key design elements within these sketches[11]. These design elements encompass a wide array of aspects integral to web layout, including various structures, components, and styling attributes. Much like

979-8-3503-6526-9/24/$31.00 ©2024 IEEE

object detection scenarios, the outcome of this sketch recognition process is an array of identified design elements, each serving as an analog to objects in the object detection domain. These recognized design elements then undergo a transformative process, wherein they are converted into functional HTML and CSS code, effectively materializing the sketch into a fully functional web page.

Beyond the mere generation of code, our system also provides valuable accompanying metadata that sheds light on the structural attributes of the generated code. This metadata encompasses the categorization of each design element, effectively assigning them roles similar to object classes in object detection systems[12]. Additionally, the metadata includes a confidence score, a numerical representation of the system's conviction regarding the accuracy of the generated HTML and CSS code for each individual design element. This confidence score proves to be of paramount importance, serving as a quantitative measure that informs the overall quality and veracity of the entire code generation process.

Much like the setting of a threshold[13] in object detection systems to filter out uncertain or dubious detections, our project employs a carefully chosen threshold to maintain the fidelity of the generated HTML and CSS code. This threshold is deliberately positioned at a relatively conservative level, often hovering around 0.5. This choice is informed by the intrinsic complexity and diversity encountered in hand-drawn sketches, where maintaining a conservative threshold becomes crucial in ensuring the accuracy of the generated code amidst the challenges posed by the varied nature of sketches.

One particularly noteworthy aspect of our system pertains to the representation of bounding box coordinates. These coordinates, which play a pivotal role in accurately placing design elements within the web layout, are standardized within a normalized range. Typically, these normalized coordinates fall within the 0 to 1 interval, represented as floating-point values. To ensure a faithful rendering of these design elements within the dimensions of the web layout[14], a meticulous transformation process is executed. This process involves scaling the normalized coordinates by the dimensions of the web page, necessitating their conversion into pixel values. This intricate conversion aligns the design elements precisely with the layout's pixel-based coordinates, ensuring a seamless integration of the hand-drawn sketches into the digital realm.

In essence, our system seamlessly navigates the intricate realm of translating hand-drawn sketches into functional web pages. Through a sophisticated combination of sketch recognition, code generation, and metadata provision, it not only captures the essence of the original sketches but also ensures a high level of accuracy and fidelity in the resultant HTML and CSS code. The deliberate use of confidence scores and conservative thresholds reflects a commitment to the reliability of the generated code, acknowledging the challenges posed by the inherent complexity and diversity of hand-drawn sketches. The transformation of bounding box coordinates into pixel values further underscores the precision and attention to detail embedded in our system, ultimately contributing to a successful bridge between the analog and digital realms of design.

WORKING OF SWIN TRANSFORMER:

The Swin Transformer, a novel vision Transformer that effectively functions as a general-purpose computer vision backbone, is presented in this work. Differences between the two domains—such as the wide range in the magnitude of visual items and the higher density of pixels in photos than in text—make it difficult to transfer Transformer from language to vision. We suggest a hierarchical Transformer whose representation is calculated using \bold{S}hifted \bold{windows} in order to overcome these discrepancies. By restricting self- attention computation to non-overlapping local windows and permitting cross-window connections, the shifted windowing technique increases efficiency. This hierarchical architecture has a computational cost that is linear in relation to image size and is flexible enough to be modeled at different scales.

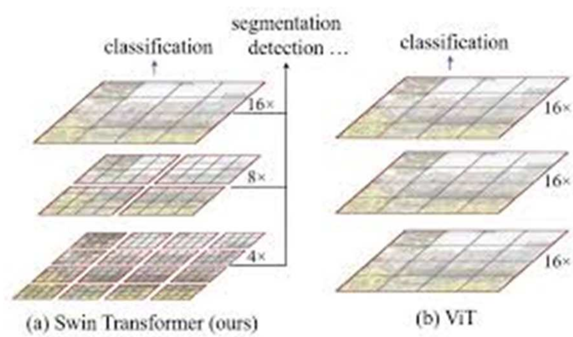

Fig 4: segmentation in swing transformer

[7]Compression plays an important role on the efficient transmission and storage of images and videos through band-limited systems such as streaming services, virtual reality or videogames. But compression inevitably results in artifacts and the original data being lost, which

979-8-3503-6526-9/24/$31.00 ©2024 IEEE

can seriously impair the visual quality. These factors have made quality upgrading of compressed images a hot study subject. While convolutional neural networks form the basis of most state-of-the-art image restoration techniques, transformer-based techniques like SwinIR also perform admirably on certain challenges. In this study, we investigate the new Swin Transformer V2 to enhance SwinIR for super-resolution images, specifically for the case of compressed input. With the use of this technique, we can address the main problems with training transformer vision models, namely training instability,

SWIN MODEL:

class transformers.SwinModel

<

>(configadd_pooling_layer = Trueuse_mask_token = False)

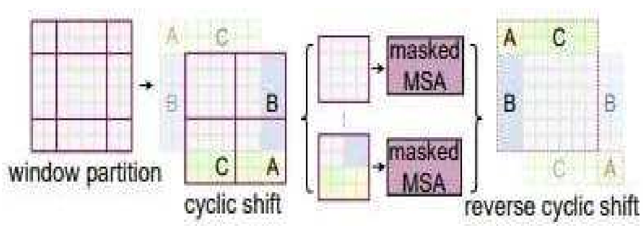

Fig 5: process of cyclic shift

Specifications

[9]Model configuration class with all of the model's parameters is called config (SwinConfig). Only the configuration is loaded when starting with a config file; the model's weights are not loaded. To load the model weights, see the from_pretrained() method.

$$\boldsymbol{Attention(Q, K, V) = SoftMax\left(\frac{QK^T}{\sqrt{d}} + B\right) V}$$

add_pooling_layer (optional; bool; True by default) — Applying or not applying the pooling layer. use_mask_token (bool, optional, False by default) — Indicates whether or not mask tokens should be applied and created in the embedding layer. The unadorned Swin Model transformer, devoid of any particular head, outputs raw hidden-states. This model belongs to the PyTorch subclass torch.nn.Module. Use it like any other PyTorch module, and for all questions about general usage and conduct, consult the PyTorch manual.
[10]

	ImageNet		COCO		ADE20k mIoU
	top-1	top-5	AP^{box}	AP^{mask}	
w/o shifting	80.3	95.2	48.0	41.6	43.4
shifted windows	81.2	95.5	50.6	44.0	46.2
no pos	79.9	95.0	49.3	42.8	43.9
abs. pos.	80.4	95.3	49.1	42.3	43.3
abs.+ rel. pos.	81.4	95.7	50.4	43.5	44.1
rel. pos.w/o app.	79.5	94.6	49.0	42.0	44.0
rel. pos.	80.9	95.5	50.6	44.7	46.2

Table 1: Accuracy points

The paper uses cyclic-shifting before computing self-attention to handle edge windows smaller than M × M efficiently, as seen in the picture below. The partitions are covered by a masking method, which restricts processing inside each initial window.

2. CONCLUSION

In conclusion, the described process illustrates a comprehensive and sophisticated approach to transforming hand-drawn sketches into functional web pages. Beginning with the fundamental step of understanding the sketches, the algorithm navigates through the intricate details of the visual representation, paving the way for the unveiling of key design elements in the subsequent stage. This unveiling is reminiscent of an object detection scenario, where layout structures, diverse components, and essential design elements are identified with precision[7]. The subsequent planning and transformation phase serve as the bridge between conceptualization and implementation, as the recognized design elements are translated into meticulous HTML and CSS code. This process, however, is not merely a mechanical translation but is underpinned by an advanced algorithm development stage. This stage is responsible for determining the precise positioning, styling, and structural attributes of each component, ensuring a high level of accuracy and fidelity in the final output. The incorporation of confidence scores in this phase reflects a commitment to quality control, akin to setting thresholds in object detection, and emphasizes the system's confidence in the generated code. Collectively, these stages form a cohesive and iterative framework, highlighting the system's ability to navigate the inherent complexity and diversity of hand-drawn sketches. The transformation from a visual representation to a digital, functional web layout showcases the effectiveness of the algorithm in interpreting, planning, and executing the conversion

979-8-3503-6526-9/24/$31.00 ©2024 IEEE

process. This approach not only captures the essence of the original sketches but also emphasizes precision and attention to detail in generating code that faithfully represents the intended design. The outlined process stands as a testament to the fusion of artistic creativity with technological prowess, resulting in a seamless integration of hand-drawn sketches into the digital landscape.

REFERENCES

[1] V. Kalbande, K. Meshram, S. Somnathe, R. Deshmukh and M. Mohod, "Automatic HTML Code Generation from Mock-Up Images Using Machine Learning Techniques", Int. Res. J. Eng. Technol, vol. 8, no. 06, 2021.

[2] A. Deolekar, K. Dhanawade, R. Chavan and S. Bhambure, "UI Code Generation using Deep learning", Int. Res. J. Eng. Technol, vol. 8, no. 05, 2021.

[3] Z. Teng, Q. Fu, J. White and D. C. Schmidt, "Sketch2Vis: Generating Data Visualizations from Hand-drawn Sketches with Deep Learning", 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 853-858, 2021.

[4] M. Dua, R. Yadav, D. Mamgai and S. Brodiya, "An improved RNN-LSTM based novel approach for sheet music generation", Procedia Comput. Sci, vol. 171, pp. 465-474, 2020.

[5] S. Patil, R. Pawar, S. Punder and J. John, "Generation of HTML Code using Machine Learning Techniques from Mock-Up Images", Int. Res. J. Eng. Technol, vol. 7, no. 03, 2020.

[6] T. Bouças and A. Esteves, "Converting web pages mockups to HTML using machine learning", 2020.

[7] Davit Soselia, Khalid Saifullah and Tianyi Zhou, "Learning Ui-To-Code Reverse Generator Using Visual Critic Without Rendering", arXiv:2305.14637v2 [cs.CV] 3 Nov 2023.

[8] Yong Xu1 , Lili Bo1 , Xiaobing Sun1 , Bin Li1 , Jing Jiang and Wei Zhou, "I mage2emmet: Automatic code generation from web user interface image", . J Softw Evol Proc. 2021;33:e2369, 2021.

[9] Gayatri Vitkare,Rutuja Jejurkar, Sammyaka Kamble, Yogeshwari Thakare, Prof. A.P.Lahare, "Automated Html Code Generation From Hand Drawn Images Using Machine Learning Methods", Volume:04/Issue:01/ January-2022.

[10] L. Srinivasan, D. Selvaraj, D. Dhinakaran, T. P. Anish, "IoT-Based Solution for Paraplegic Sufferer to Send Signals to Physician via Internet," SSRG International Journal of Electrical and Electronics Engineering, vol. 10, no. 1, pp. 41-52, 2023.Crossref, https://doi.org/10.14445/23488379/IJEEE-V10I1P104

[11] D. Dhinakaran, L. Srinivasan, D. Selvaraj, S. M. Udhaya Sankar, "Leveraging Semi-Supervised Graph Learning for Enhanced Diabetic Retinopathy Detection," SSRG International Journal of Electronics and Communication Engineering, vol. 10, no. 8, pp. 9-21, 2023. Crossref, https://doi.org/10.14445/23488549/IJECE-V10I8P102

[12] Palanikkumar, D., Abhisek Sethy, Kasiprasad Mannepalli, P. John Augustine, Semira Abdela Seyid, and Adem Aman Hassen. "Managing the Offline Text Recognition Accuracy Using Recurrent Neural Network." Journal of Contemporary Issues in Business and Government Vol 27, no. 1 (2021).

[13] Suganya, R., Jayashree, L.S. Fuzzy Rough Set Inspired Rate Adaptation and resource allocation using Hidden Markov Model (FRSIRA-HMM) in mobile ad hoc networks. Cluster Comput 22 (Suppl 4), 9875–9888 (2019). https://doi.org/10. 1007/s10586-018-1783-1

[14] Chinnasamy, P., Praveena, V. , Secure and Efficient Data Sharing Scheme in Cloud for Protecting Data in Smart Cities,EAI/Springer Innovations in Communication and Computing, 2021, pp. 41–54

[15] Sivaramakrishnan Rajender, Dhivya Rathinasamy, Pavithra R, Vishnu Kumar Kaliappan, S Gnanamurthy "Prediction of stopping distance for autonomous emergency braking using stereo camera pedestrian detection." Materials Today: Proceeding(2022)Elsevier-ISSN:1224-1228. Volume 51, Part 1, 2022, Pages 1224-1228 https://doi.org/10.1016/j.matpr.2021.07.211

979-8-3503-6526-9/24/$31.00 ©2024 IEEE