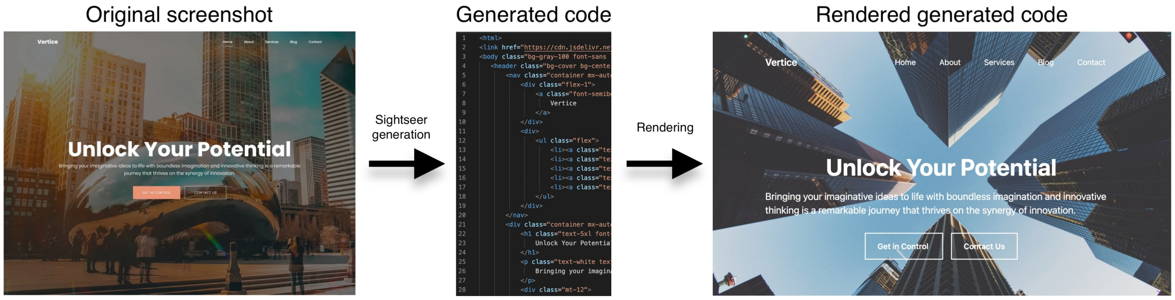
Unlocking the conversion of Web Screenshots into HTML Code with the WebSight Dataset

Hugo Laurençon, Léo Tronchon, Victor Sanh

Hugging Face

Using vision-language models (VLMs) in web development presents a promising strategy to increase efficiency and unblock no-code solutions: by providing a screenshot or a sketch of a UI, a VLM could generate the code to reproduce it, for instance in a language like HTML. Despite the advancements in VLMs for various tasks, the specific challenge of converting a screenshot into a corresponding HTML has been minimally explored. We posit that this is mainly due to the absence of a suitable, high-quality dataset. This work introduces WebSight, a synthetic dataset consisting of 2 million pairs of HTML codes and their corresponding screenshots. We fine-tune a foundational VLM on our dataset and show proficiency in converting webpage screenshots to functional HTML code. To accelerate the research in this area, we open-source WebSight.

Dataset: `https://huggingface.co/datasets/HuggingFaceM4/WebSight`



```
<html>
<link href="https://cdn.jsdelivr.ne
<body class="bg-gray-100 font-sans
    <header class="bg-cover bg-cente
        <nav class="container mx-aut
            <div class="flex-1">
                <a class="font-semib
                    Vertice
                </a>
            </div>
            <div>
                <ul class="flex">
                    <li><a class="te
                    <li><a class="te
                    <li><a class="te
                    <li><a class="te
                    <li><a class="te
                </ul>
            </div>
        </nav>
        <div class="container mx-aut
            <h1 class="text-5xl font
                Unlock Your Potentia
            </h1>
            <p class="text-white tex
                Bringing your imagin
            </p>
            <div class="mt-12">
```

Bringing your imaginative ideas to life with boundless imagination and innovative thinking is a remarkable journey that thrives on the synergy of innovation.

Figure 1: Comparison of an original web page (input) on the left, and the rendering of the code generated by our model - Sightseer - (output) on the right. To match the visual aspect of the original webpage, the model picked a suitable image background from `www.unsplash.com`

1 Introduction

Current advancements in vision-language models (VLMs) have significantly improved their capabilities, enabling them to master a variety of tasks including image captioning, question answering, and optical character recognition (OCR) (OpenAI et al., 2023; Team et al., 2023; Hong et al., 2023; Liu et al., 2024a). Despite these achievements, the task of converting screenshots of websites or web components into usable HTML code—a process highly valuable to web developers—remains relatively unexplored, particularly in the open-source community. The development and open-source release of a model capable of such a conversion could unlock new AI-powered tools for UI developers, facilitating the creation of no-code modules and plugins for design tools like Figma. For instance, the ability to rapidly transform a design sketch into a functional UI component and code could significantly increase the iteration pace for UI developers.

We posit that the primary challenge for VLMs to achieve proficiency in this specific task does not stem from the inherent difficulty of the task itself. Rather, it is the lack of a large, high-quality, dataset of pairs of HTML codes and their associated screenshots that poses the primary obstacle. In fact, VLMs are commonly trained on web-scale datasets of image-text pairs (Schuhmann et al., 2022; Gadre et al., 2023) or multimodal web documents (Laurençon et al., 2023; Zhu et al., 2023). Having such a dataset of screenshots-HTML pairs as an open and accessible artifact would significantly accelerate research in this area by enabling the community to inspect the data, its limitations and improve upon the dataset. Consequently, our initial focus is on developing a dataset useful for the fine-tuning of VLMs for this task. To accomplish this, several strategies can be considered:

1. *Leveraging existing webpages and their HTML codes.* The vast repository of HTML files available online (and often captured in web crawls like Common Crawl) presents a tempting resource for generating pairs of screenshots and corresponding HTML codes by simply rendering the HTML and capturing the output. However, this approach poses significant challenges. HTML files found on the web are often laden with noise like comments, scripts or data, and can be excessively lengthy, encompassing a very large number tokens, sometimes even exceeding the maximum sequence length of most current models. This complexity hinders a model's ability to accurately learn the correlation between the contents of a screenshot and the underlying HTML syntax. Additionally, HTML codes frequently incorporate references to external JavaScript (JS) or Cascading Style Sheets (CSS) scripts, or rely on files located in separate directories. This dependency complexifies the creation of a self-contained HTML file that faithfully reproduces the intended design in a screenshot. Given these obstacles, we opted to forego this method in favor of a more controlled approach.

2. *Synthesizing HTML codes with Large Language Models (LLMs).* The most recentlarge language models, especially those trained extensively on programming languages, show remarkable proficiency in generating high-quality code applicable to various domains, including website development. This capability opens the door to artificially create a vast corpus of HTML codes using a LLM specialized in coding which has been further fine-tuned to follow instructions. By adapting the prompts, we can introduce specific constraints to the code generation process, such as controlling the topic, the text length or the image placement in the websites. This level of control not only ensures the production of relevant HTML code but also makes them more suitable for VLMs by providing the models with cleaner, more concise, and structured data that models can be effectively trained on. Our study adopts this approach.

In response to the identified gap, we develop WebSight, a comprehensive synthetic dataset comprising 2 million examples of HTML code paired with corresponding screenshots. Leveraging this dataset, we proceed to fine-tune our forthcoming foundational VLM of 8 billion parameters, notably enhanced by robust OCR capabilities, to obtain the specialized model Sightseer. This fine-tuning process yields promising outcomes, demonstrating the model's proficiency in converting webpage screenshots into functional HTML code. Remarkably, the model also exhibits the versatility to adapt to untrained scenarios, such as transforming handwritten sketches into functional HTML code. To accelerate advancements in this direction, we open source WebSight.

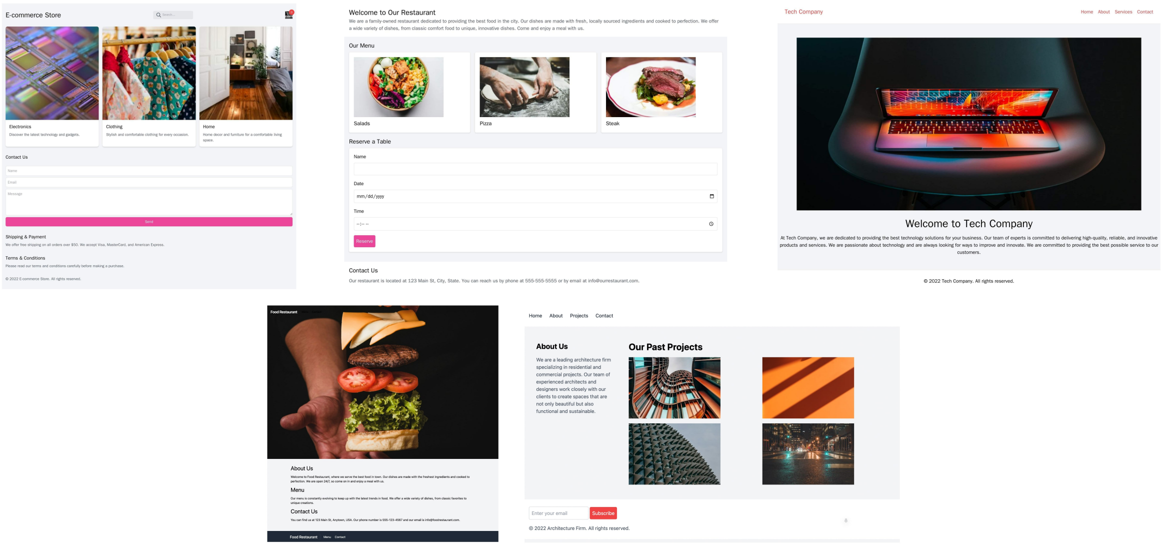
2 Related work

Nguyen and Csallner (2015) uses a classical pipeline of interface elements recognition (images, texts, containers, etc.) with computer vision and optical character, followed by heuristics to generate code on these detections. The authors show the effectiveness of this approach on mobile UIs. Beltramelli (2017) introduces an end-to-end method for generating computer code from graphical user interface (GUI) screenshots using deep learning. The model, trained end-to-end, can generate code for different platforms (iOS, Android, and web) from a single input image. It uses convolutional and recurrent neural networks to interpret GUI screenshots and generate corresponding code. In Lee et al. (2023), authors pre-train a VLM to convert masked screenshots of web pages into simplified HTML, and show the effectiveness of this training objective to pretrain foundational VLM that transfers well to a variety of downstream tasks. Similar to Sightseer, their model accepts images of varying resolutions as input.

In our recent beta release of WebSight-v0.1, we provided a dataset with 823K synthetic pairs of screenshots and associated HTML + traditional CSS code. In the current version of WebSight discussed in this paper (v0.2), we introduce significant improvements. First, WebSight-v0.2 replaces the colored rectangles used as image placeholders in WebSight-v0.1 with real images that match the website's content. Additionally, we adopt Tailwind CSS to streamline the code and facilitate the creation of visually appealing designs. Other notable upgrades include 2.5x the dataset size, offering higher resolution screenshots, and providing richer metadata.

WebSight-v0.1 has already proven to be a helpful resource. In Design2Code (Si et al., 2024), the authors create a benchmark for evaluating VLMs at generating HTML code given a screenshot. They also fine-tune an 18B-parameter VLM on WebSight-v0.1, after observing that models trained on synthetic examples outperform those trained on longer, more complex real-world code data.

3 Construction of the dataset



E-commerce Store
Search...
Electronics
Discover the latest technology and gadgets.
Clothing
Stylish and comfortable clothing for every occasion.
Home
Home decor and furniture for a comfortable living space.
Contact Us
Name
Email
Message
Send
Shipping & Payment
We offer free shipping on all orders over $50. We accept Visa, MasterCard, and American Express.
Terms & Conditions
Please read our terms and conditions carefully before making a purchase.
© 2022 E-commerce Store. All rights reserved.
Welcome to Our Restaurant
We are a family-owned restaurant dedicated to providing the best food in the city. Our dishes are made with fresh, locally sourced ingredients and cooked to perfection. We offer a wide variety of dishes, from classic comfort food to unique, innovative dishes. Come and enjoy a meal with us.
Our Menu
Salads
Pizza
Steak
Reserve a Table
Name
Date
mm/dd/yyyy
Time
--:-- --
Reserve
Contact Us
Our restaurant is located at 123 Main St, City, State. You can reach us by phone at 555-555-5555 or by email at info@ourrestaurant.com.
Tech Company
Home
About
Services
Contact
Welcome to Tech Company
At Tech Company, we are dedicated to providing the best technology solutions for your business. Our team of experts is committed to delivering high-quality, reliable, and innovative products and services. We are passionate about technology and are always looking for ways to improve and innovate. We are committed to providing the best possible service to our customers.
© 2022 Tech Company. All rights reserved.
Food Restaurant
About Us
Welcome to Food Restaurant, where we serve the best food in town. Our dishes are made with the freshest ingredients and cooked to perfection. We are open 24/7, so come on in and enjoy a meal with us.
Menu
Our menu is constantly evolving to keep up with the latest trends in food. We offer a wide variety of dishes, from classic favorites to unique creations.
Contact Us
You can find us at 123 Main St, Anytown, USA. Our phone number is 555-123-4567 and our email is info@foodrestaurant.com.
Food Restaurant
Menu
Contact
Home
About
Projects
Contact
About Us
We are a leading architecture firm specializing in residential and commercial projects. Our team of experienced architects and designers work closely with our clients to create spaces that are not only beautiful but also functional and sustainable.
Our Past Projects
Enter your email
Subscribe
© 2022 Architecture Firm. All rights reserved.

Figure 2: Examples of synthetic web pages present in WebSight.

Overview of the strategy Our synthetic HTML code generation process involves two key steps for maximizing diversity and quality. First, we employ a smaller language model to generate a variety of website themes and designs. These creative outputs serve as the foundation for the next stage, where they are fed into the prompts of a larger language model, mostly trained on code data. This LLM then generates the final HTML code, ensuring that our dataset encompasses a wide range of styles while generating high-quality codes.

Generating diverse website concepts We employ Mistral-7B-Instruct (Jiang et al., 2023) to generate several million unique website concepts and designs with the prompt:

> Generate diverse website layout ideas for different companies, each with a unique design element. Examples include: a car company site with a left column, a webpage footer with a centered logo. Explore variations in colors, positions, and company fields. Don’t give any explanations or recognition that you have understood the request, just give the list of 10 ideas, with a line break between each.

Here are 5 examples of generated concepts:

> 1) Fashion Brand: A visually stunning layout with a full-width, rotating image carousel showcasing their latest collections, a bold, center-aligned logo, and a bottom navigation menu. The color palette is inspired by the latest fashion trends.
> 2) Restaurant Chain: A design with a mouth-watering header image of a popular dish, a top navigation menu with pages for location, menu, and order online. The footer displays social media icons, a newsletter sign-up form, and contact information.
> 3) Consulting Firm: A clean, professional design with a full-width image and a tagline, a left column for the main navigation menu, and a right column for featuring key staff members and their service offerings. A muted color palette evokes trust and expertise.
> 4) Real Estate Agency: A user-friendly design with a header featuring a property image and search functionality, a top navigation menu with links to the various property listings. The page includes a grid layout for listing images, and a footer with contact information and social media icons.
> 5) Education Platform: A design with a wide, hero image, a centered logo, and a top navigation menu featuring links for courses, pricing, and about the company. The site includes a feature section where students can share testimonials and success stories. A warm, approachable color palette is used to inspire confidence and engagement.

Opting for Tailwind CSS over traditional CSS Generating visually diverse and appealing designs requires more than just pure HTML. However, to simplify the learning process of VLMs, employing standalone code is preferable to managing separate files. In this context, Tailwind CSS emerges as an ideal solution. This utility-first framework allows creating unique designs by providing a wide array of utility classes, enables direct styling within the HTML document, and eliminates the need for external style files. Tailwind CSS offers an extensive array of predefined classes that mirror various CSS properties. By integrating these utility classes into HTML elements, we can efficiently style web pages, resulting in concise code that is easier for VLMs to learn from.

Using a code specialized LLM to generate the HTML codes To generate the final HTML codes, we leverage Deepseek-Coder-33b-instruct (Guo et al., 2024), a state-of-the-art language model mostly trained on code data and fine-tuned to follow instruction. We use the prompt:

> Code a complete website with a good design in HTML and Tailwind CSS about this: {concept}
> Write the code inside a tag <body>.
> Write real and long sentences about the business. NEVER USE sentences starting with Lorem ipsum, NEVER.
> You don’t have to include images, but if you do, use only this source "https://source.unsplash.com/random/WxH/?keyword", by replacing ‘W‘ and ‘H‘ in the URL by the desired width and height, and ‘?keyword‘ by a keyword describing the picture, for example "https://source.unsplash.com/random/300x200/?gym" for an image about gym of size 300x200, or "https://source.unsplash.com/random/100x200/?cake" for an image of a cake of size 100x200.

An initial challenge was the text-only nature of our outputs, contrasting with the real websites containing many images. The task of integrating images into an HTML code seems hard, especially when trying to look for images related to the context of the web page. However, we discovered an effective solution through photo stocks like `https://source.unsplash.com/`, which offers the capability to dynamically generate images based on keywords, thus providing images of any size and relevant to any specified topics.

After a filtering step in which we discard web pages with insufficient text, generic content or images not aligning with the website's topic, we finally ended up with 2 million web pages.

Screenshot capture process We use Playwright[1] to visualize and capture the output of our generated HTML codes. We ensure that screenshots encompass the entire web page, regardless of its length. As a result, our dataset features screenshots in a wide range of resolutions. This diversity in image size and format is useful for enhancing the robustness of our model.

Visualization of WebSight examples Five examples present in WebSight are shown in Figure 2.

4 Fine-tuning a foundation vision-language model on WebSight

Model prerequisites for webpage conversion For a model to accurately convert webpage screenshots into HTML code, it necessitates several capabilities. These include advanced OCR to transcribe text from images, spatial understanding to arrange elements on the page, and object recognition abilities to replicate images similar to those in the input with the strategy explained above.

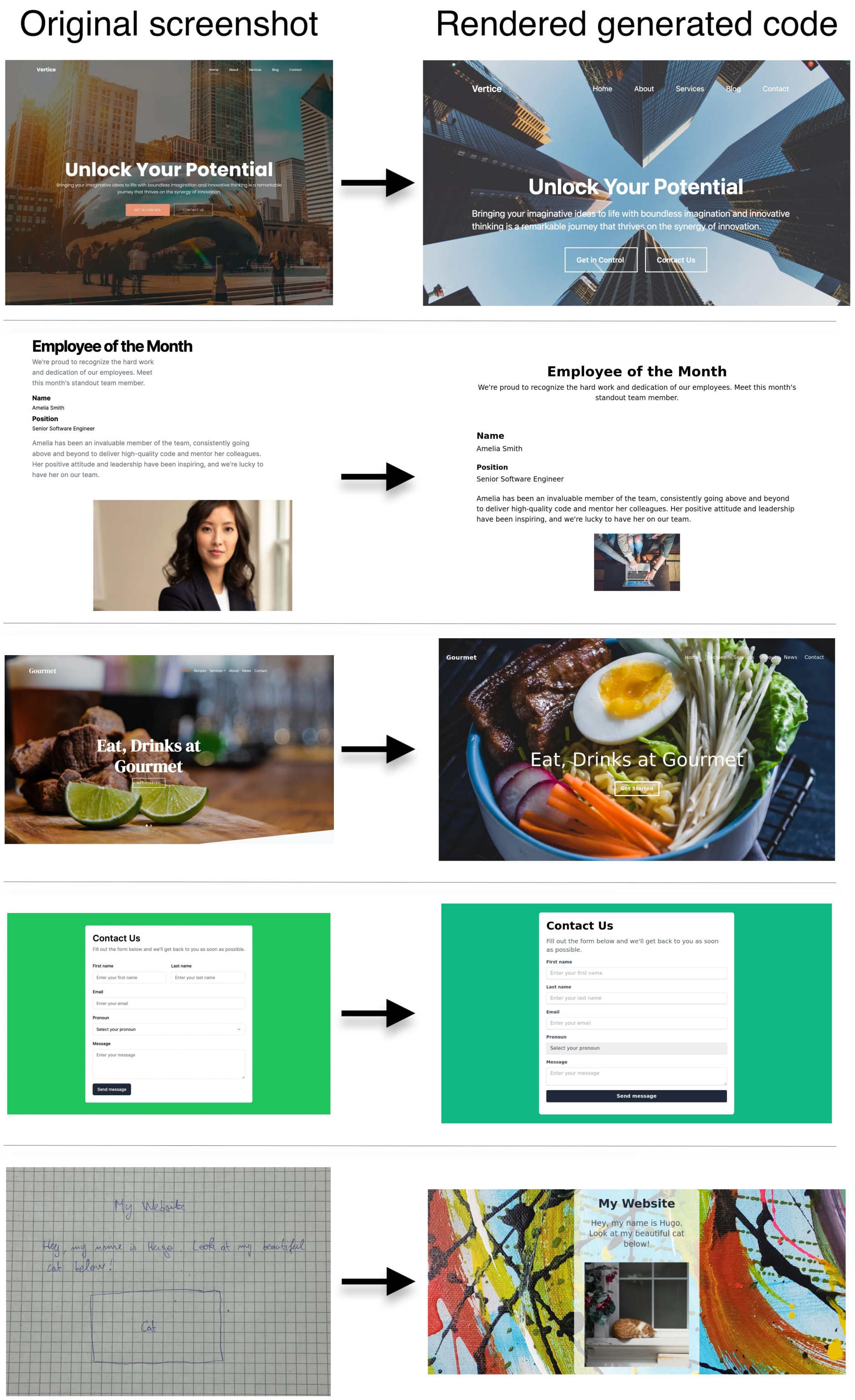
We use our forthcoming foundation VLM as the base model. It is built upon Mistral-7B (Jiang et al., 2023) and SigLIP-SO400M (Zhai et al., 2023), and is using the Patch n' Pack strategy (Dehghani et al., 2023) to preserve the original aspect ratio of the input images, with a resolution of up to 980 pixels for each side.

This base model was trained mostly on OBELICS (Laurençon et al., 2023), synthetic captions of image/text pairs datasets, and a combination of OCR datasets (Biten et al., 2022).

Further insights into the model's architecture and its training process will be detailed upon its release.

Fine-tuning on WebSight For the fine-tuning, instead of unfreezing all the weights, which requires lowering significantly the learning rate for a stable training, we use the parameter efficient DoRA method (Liu et al., 2024b) with a rank 64. We use the same learning rate that was chosen during the pre-training, 10^{-4}, while seeing 2016 examples per iteration, for a total of 1100 iterations, representing a bit less than one epoch.

We find that the validation loss is not a good indicator of the trained model and in particular the quality of generated codes in real-world cases. Consequently, we perform checkpoint selection by manually inspecting generated samples rather than relying on the validation loss. Despite the validation loss continuing to decrease significantly over several epochs, it did not translate into an increased ability to generalize to websites that differ from those in the training dataset.



Bringing your imaginative ideas to life with boundless imagination and innovative thinking is a remarkable journey that thrives on the synergy of innovation.

Employee of the Month

We're proud to recognize the hard work and dedication of our employees. Meet this month's standout team member.

Name

Amelia Smith

Position

Senior Software Engineer

Amelia has been an invaluable member of the team, consistently going above and beyond to deliver high-quality code and mentor her colleagues. Her positive attitude and leadership have been inspiring, and we're lucky to have her on our team.

Contact Us

Fill out the form below and we'll get back to you as soon as possible.

First name: Enter your first name

Last name: Enter your last name

Email: Enter your email

Pronoun: Select your pronoun

Message: Enter your message

Send message

My Website

Hey, my name is Hugo. Look at my beautiful cat below!

Figure 3: Comparison of an original web page (input) on the left, and the rendering of the code generated by our model, Sightseer, (output) on the right.

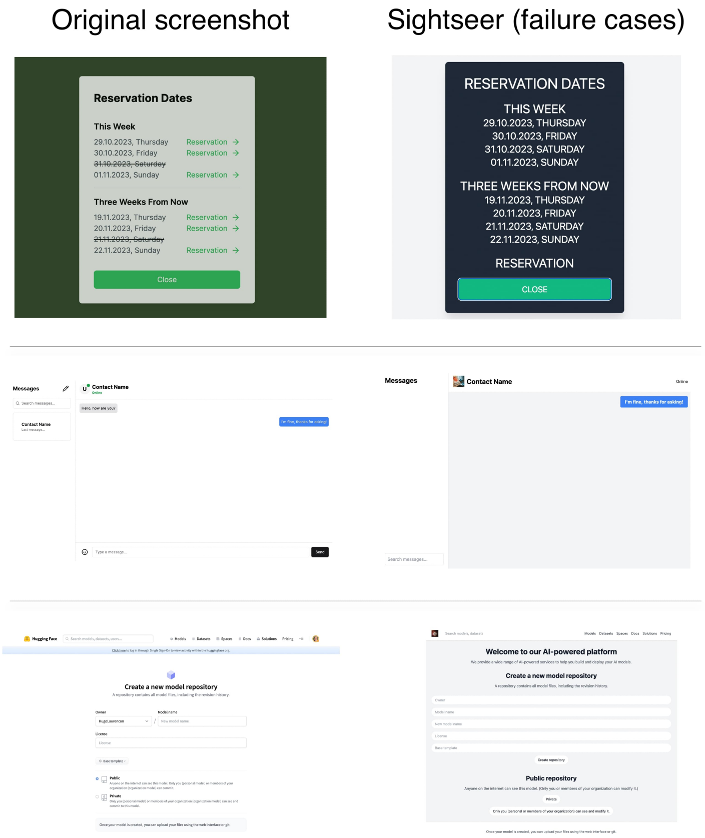
5 Qualitative evaluation

5.1 Results on different screenshots

Figure 3 showcases various outputs from Sightseer when fed with simple website designs. Notably, in instances where the input contains a limited amount of text, this text tends to be accurately preserved in the output.

Remarkably, Sightseer sometimes exhibits the capability to generalize beyond its training dataset to websites that differ significantly in appearance, as evidenced by its conversion of a handwritten website sketch into functional HTML code.

5.2 Failure cases



Reservation Dates

This Week

29.10.2023, Thursday Reservation →
30.10.2023, Friday Reservation →
~~31.10.2023, Saturday~~
01.11.2023, Sunday Reservation →

Three Weeks From Now

19.11.2023, Thursday Reservation →
20.11.2023, Friday Reservation →
~~21.11.2023, Saturday~~
22.11.2023, Sunday Reservation →

RESERVATION DATES

THIS WEEK
29.10.2023, THURSDAY
30.10.2023, FRIDAY
31.10.2023, SATURDAY
01.11.2023, SUNDAY

THREE WEEKS FROM NOW
19.11.2023, THURSDAY
20.11.2023, FRIDAY
21.11.2023, SATURDAY
22.11.2023, SUNDAY

RESERVATION

Search messages...
Contact Name
Last message...
Contact Name
Online
Hello, how are you?
I'm fine, thanks for asking!
Type a message...

Contact Name
Online
I'm fine, thanks for asking!
Search messages...

Search models, datasets, users...
Click here to log in through Single Sign-On to view activity within the huggingface org.

Create a new model repository
A repository contains all model files, including the revision history.

Owner: HugoLaurencon / Model name: New model name
License: License
Base template
○ Public — Anyone on the internet can see this model. Only you (personal model) or members of your organization (organization model) can commit.
○ Private — Only you (personal model) or members of your organization (organization model) can see and commit to this model.

Once your model is created, you can upload your files using the web interface or git.

Search models, datasets

Welcome to our AI-powered platform
We provide a wide range of AI-powered services to help you build and deploy your AI models.

Create a new model repository
A repository contains all model files, including the revision history.

Owner
Model name
New model name
License
Base template

Public repository
Anyone on the internet can see this model. (Only you or members of your organization can modify it.)
Only you (personal or members of your organization) can see and modify it.

Once your model is created, you can upload your files using the web interface or git.

Figure 4: Examples where Sightseer-generated codes fall short in replicating the intended design.

In our analysis, Sightseer struggles with complex website layouts, excessive text, or designs significantly divergent from its training data.

In some instances, generated code includes elements such as images, text, or buttons that fail to appear upon rendering. This can result from issues like text colored identically to its background or incorrect syntax use, suggesting that Sightseer has not fully mastered the HTML + Tailwind CSS syntax.

[1] https://github.com/microsoft/playwright

While the model produces visually more attractive websites, it sometimes produces errors not observed in our initial model[2] trained on WebSight-v0.1, which used traditional CSS instead of Tailwind CSS. As a more recent framework than traditional CSS, Tailwind CSS has less frequent occurrence in the pre-training data of the base LLM, and we hypothesize that the LLM has bigger challenges in fully mastering its syntax. We posit that starting with a foundational VLM pre-trained with text-only HTML + Tailwind CSS in the mixture of data could significantly enhance Sightseer's translation accuracy, and we are exploring related strategies to achieve this improvement.

6 Conclusion

In this work, we introduce WebSight, a large synthetic dataset of 2 million pairs of HTML codes and corresponding renderings, and Sightseer, a vision and language model with OCR ability fine-tuned on WebSight, as contributions towards automating the conversion of webpage screenshots to HTML code. By leveraging synthetic data generation and fine-tuning a high-capacity base VLM on the dataset, we demonstrate a viable path to accelerate UI development tasks and enhance no-code solutions with increasingly more powerful AI-powered tools. By open-sourcing WebSight, we aim to foster further innovation and research in this area.

References

Tony Beltramelli. pix2code: Generating code from a graphical user interface screenshot. *arXiv preprint arXiv:1705.07962*, 2017.

Ali Furkan Biten, Rubèn Tito, Lluis Gomez, Ernest Valveny, and Dimosthenis Karatzas. Ocr-idl: Ocr annotations for industry document library dataset, 2022.

Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim Alabdulmohsin, Avital Oliver, Piotr Padlewski, Alexey Gritsenko, Mario Lučić, and Neil Houlsby. Patch n' pack: Navit, a vision transformer for any aspect ratio and resolution, 2023.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets, 2023.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. Deepseek-coder: When the large language model meets programming – the rise of code intelligence, 2024.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxuan Zhang, Juanzi Li, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. Cogagent: A visual language model for gui agents, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[2] https://huggingface.co/HuggingFaceM4/VLM_WebSight_finetuned

Hugo Laurençon, Lucile Saulnier, Leo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M Rush, Douwe Kiela, Matthieu Cord, and Victor Sanh. OBELICS: An open web-scale filtered dataset of interleaved image-text documents. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL `https://openreview.net/forum?id=SKN2hflBIZ`.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2Struct: Screenshot parsing as pretraining for visual language understanding. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 18893–18912. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/lee23g.html`.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024a. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation, 2024b.

Tuan Anh Nguyen and Christoph Csallner. Reverse engineering mobile application user interfaces with remaui (t). *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 248–259, 2015. URL `https://api.semanticscholar.org/CorpusID:7499368`.

OpenAI et al. Gpt-4 technical report, 2023.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022.

Chenglei Si, Yanzhe Zhang, Zhengyuan Yang, Ruibo Liu, and Diyi Yang. Design2code: How far are we from automating front-end engineering?, 2024.

Gemini Team et al. Gemini: A family of highly capable multimodal models, 2023.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023.

Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi. Multimodal c4: An open, billion-scale corpus of images interleaved with text. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL `https://openreview.net/forum?id=tOd8rSjcWz`.