# WebDiffusion: Machine Learning Meets the World Wide Web

https://webdiffusion.ai/

Yasir Zaki⋆, Joseph Hong⋆, Matteo VarvelloΔ
⋆ New York University Abu Dhabi, Δ Nokia Bell Labs
⋆ United Arab Emirates, Δ United States of America
yasir.zaki@nyu.edu,joseph.hong@nyu.edu,matteo.varvello@nokia.com

## ABSTRACT

Text-to-image generation is a recent advancement in machine-learning which has gained enormous popularity in the summer of 2022, especially after the release of *stable diffusion*. Many applications have been proposed, from AI-powered movies to medicine. In this paper, we argue that the World Wide Web is the perfect use-case for AI-powered image generation. A webpage, at its core, is a collection of text and images linked with some semantic meaning. Web developers can use tools like stable diffusion to automatically generate the images they need. Further, such images do not have to be transferred to the clients, as they can be locally generated by a browser via a small textual prompt. The main contribution of this paper is `WebDiffusion`, a tool that allows users to emulate a Web leveraging stable diffusion model for image generation. We use WebDiffusion to evaluate the accuracy of AI-powered image generation for the Web, as well as its performance. We crowdsourced about 900 pairs of eyeballs to assess the quality of 409 AI-generated images from 25 full webpages. We find that 70-80% of the images are scored as either "Ok" or higher, and no AI-generated webpage was ever scored less than "Ok". We also find that AI-generated images including people, especially when hands and faces are in the foreground, are responsible for most low scores. Performance wise, most webpages see a size reduction of multiple MBytes, at the cost of a few seconds of slowdown, 20% CPU increase and a 80% occupied powerful GPU (Nvidia Tesla A40).

## 1 INTRODUCTION

The summer of 2022 has seen a sudden surge in popularity of text-to-image generation models that generate images from textual descriptions [5, 31, 37–39]. Stable diffusion [32], launched in August 2022, is the first free and open-source model which has allowed a plethora of researchers, developers, and tech enthusiasts to develop novel AI applications like interior design [12], art [30], and even automated video generation [3].

Images are a vital component of the Web: for example, the median webpage today is comprised of 900 KB of images (or about 44% of its total size) [25]. Websites need images to showcase a product, or to add visual cues to a story. Some websites instead use images to keep the user engaged, and break the text flow. It follows that web developers do not always need a specific image for their webpage, and rely on services such as Pixabay [7] to identify "inspiring" images.

In this paper, we argue that text-to-image generation can be a new component of the Web. The idea is to allow Web developers to *describe* their images, which are then automatically generated by the browser. This approach offers several advantages. First, it speeds up Web development by removing the hassle of finding images and managing their copyrights. Second, it can reduce the weight (in bytes) of a webpage; this can offer significant cost savings for the content provider, as well as data savings for the users. Last but not least, by being image-centric this approach has virtually no impact on Webcompat, differently from solutions which aim at removing redundant or unused JavaScript code [26, 27, 36].

There are, however, potential concerns which need to be investigated. First and foremost, *what is the quality of AI-generated webpages?* This is a challenging research question as AI generated images are not expected to be equivalent to an input image, especially when such image does not exist. In the context of the Web, however, we have a large corpus of webpages for which images have been already selected, making it the perfect data-set to study how well text-to-image generation currently works. Second, *what is the overhead imposed on the clients?* The bandwidth savings discussed above come at the cost of extra work required at the client-side and could, depending on the client hardware, slow down the user experience.

In this paper, we set out to answer the above research questions. To do so, we have developed `WebDiffusion` – currently live at *https://webdiffusion.ai/* – a tool which allows users to emulate a Web augmented with stable diffusion model for image generation. WebDiffusion implements two techniques for image annotation. A *client-based* approach which emulates a browser implementation, *i.e.,* with no access to the original image. A *server-based* approach with full access to the Web content, thus emulating the role of a developer eager to adopt such new technology. Given a target webpage, WebDiffusion is capable to crawl its content, produce AI-based images, and perform real time experiments with unmodified browsers. The latter is achieved by introducing a *proxy* which replaces Web image requests with textual prompts to feed to a *stable diffusion server*. Last but not least, WebDiffusion integrates with Prolific [35], a popular crowdsourcing platform, to gather worldwide eyeballs reporting on the quality of Web images produced by stable diffusion.

We evaluate WebDiffusion via a combination of in-lab experiments and user studies. We start by crawling the top 500 webpages from Trancos' top million list [13]. After filtering,

we end up with 200 webpages from which we collected and automatically annotated 1,870 images. Next, we crowdsourced 1,000 people to evaluate the quality of these annotations, estimating a success rate of 80% for the client-side approach, and close to 90% for the server-side approach. We then randomly selected 25 webpages (accounting for 409 unique images) and generated both client-based and server-based versions of each image, as well as of each webpage. We then crowdsourced about 884 people (10 per image/webpage, on average) to score the quality of the AI-generated content. We find that 70-80% of the images are scored as either "ok" or higher ("good" and "very good"), and 95% of webpages are scored as "ok" or higher. The server-based approach achieves the best performance, thanks to the additional contextual information available. We further investigate which image features, e.g., landscape versus a celebrity, have the largest impact on high and low scores. We find that images including people, especially when hands and faces are in the foreground, achieve the lowest scores. Performance wise, most webpages see a size reduction of multiple MBytes, at the cost of a few seconds of slowdown, 20% CPU increase, and the need of a powerful GPU (Nvidia Tesla A40) occupied at 80%.

## 2 BACKGROUND AND RELATED WORK

**Text-to-Image Generation** – A variety of text-to-image generation models have been released over the last few years. These models face the so-called *generative learning trilemma* [44], *i.e.,* they can only satisfy two out of three requirements that real-world applications of generative models should possess: high-quality outputs, output sample diversity, and computationally fast and inexpensive sampling. Image generation usually requires high-quality outputs, thus general adversarial network (GAN) or diffusion models are preferred.

Thanks to recent developments, diffusion models are currently outperforming GAN models [20]. These gradually add Gaussian noise to the input, and afterwards perform a reverse denoising process, which reverses this process and outputs an image [44]. GLIDE and DALL-E 2 [31, 37] both use classifier-free diffusion models as a part of their image generation process. We found that these diffusion models produce very realistic outputs and have large mode coverage, ideal for image generation, but tend to be very slow due to the number of iterations required throughout the denoising processes.

This is the issue that the latent diffusion model [39], the foundation of *stable diffusion* [32], addresses. By transforming the diffusion model to be compatible with a compressed image representation in a latent space, latent diffusion allows the retention of high-quality outputs on a wide variety of image-generation tasks quickly and in a computationally efficient manner [32]. Fulfilling the three aforementioned requirements, latent diffusion has potential for real-world applications, which is our motivation for selecting stable diffusion as the image generation model for this paper.

**Image Generation for the Web** – To the best of our knowledge, no previous work has yet attempted to automatically generate Web images via any form of AI. The closest approach to our proposal is [41], where the authors propose to replace Web images with *similar* content, e.g., a similar picture of the Golden Gate bridge taken from a different angle. Their rationale is to "race" similar Web images to speedup webpage loads. The similarity with our work stands in the idea that Web images are not always "strict", aka developers might be willing to accept similar images to the original if this comes with some performance benefits, e.g., bandwidth savings. The key difference with our work is that we propose to eliminate rather than substitute images, trading CPU/GPU cycles (to generate the images) for bandwidth savings (both client and server side).

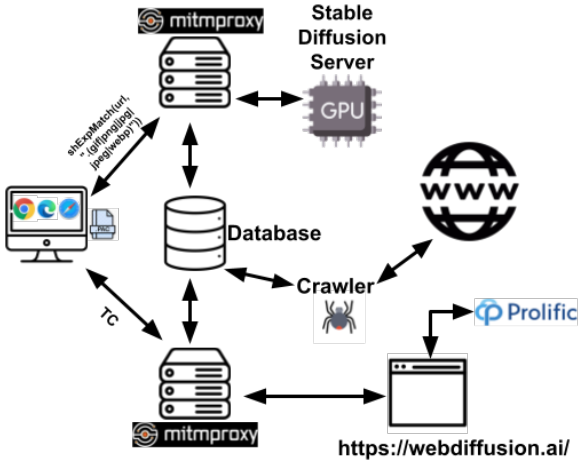



**Figure 1: A visualization ot WebDiffusion, our tool to evaluate the applicability of text-to-image generation for the Web.**

## 3 WEBDIFFUSION

This section presents `WebDiffusion`, a tool which allows users to explore and emulate a Web leveraging stable diffusion for image generation. At high level, WebDiffusion consists of the four components visualized in Figure 1. First, a *crawler* which makes local copies of actual webpages, while annotating embedded images with textual prompts using two techniques discussed next. Second, a *proxy* system to replay webpages while replacing images with textual prompts to feed to a *stable diffusion server*, allowing the emulation of an AI-powered Web without requiring an in-browser implementation. Finally, a *Web interface* which collects human feedback on the accuracy of AI-generated Web images. In the remainder of this section, we describe each component of WebDiffusion.

### 3.1 Crawler

The first task of the WebDiffusion's crawler is to generate local copies of webpages. These webpages can be analyzed to investigate the feasibility of image annotation techniques discussed below, as well as "replayed" to benchmark the performance of AI-generated webpages. The crawler consists of a Selenium [10] application which, for a given webpage, records headers and data returned for each HTTP(S) request

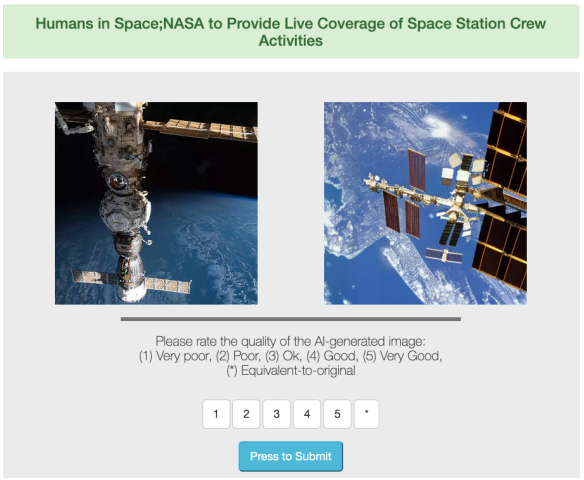



**Figure 2: WebDiffusion's GUI. The top of the figure shows the text originated using the client-based approach, *i.e.,* eventual *alt text* and nested divs information. The center of the figure shows both the original and the AI-generated images, randomly placed left or right. The bottom of the figure allows study participants to score the quality of this AI-generated image, between 1 (very poor) and 5 (very good). Participants can also score '*', in case they cannot distinguish which figure is AI-generated.**

along with its duration. Crawled webpages are stored in a database so that they can then be *replayed* via the proxy.

Next, the crawler runs the following annotation schemes to obtain textual prompts for each image embedded in a given webpage. Each textual prompt is also stored in the database along with the technique used for its derivation.

**Client-Based**: This technique aims at being deployed today without any server-side support, meaning that it works solely relying on *contextual* information, e.g., `alt text` sometimes provided along with an image (about 50% of the time [33]). The trade-off is a potential lack of accuracy as the browser is forced to "guess" the content of an image without having access to it.

This mechanism works as follows. For each image tag in a page's HTML that has either a `src` or `background-image: url` attribute in its CSS styling, we look for its `alt text` (or `alt tag`) whose goal is to provide a description of an image in case the actual image is missing. In addition, we also iterate through the different `div` elements associated with the image looking for potential text within $< p >$, $< h1 >$, $< h2 >$, $< h3 >$ and $< h4 >$ tag elements, and then iterate through their children nodes to extract the text associated with them. This is often where the title and the description of an image are stored. We accumulate all the text associated with these different elements into forming the client-based textual prompt to be used to generate the image. It also recursively traverses the parent div nodes of the image div to extract all the text related to the image; it stops when it discovers the presence of another image div and reverts back to previous node.

**Server-Based**: This technique implies some server side support, e.g., from Web developers, to annotate images. It extends the client-based approach with image annotation obtained via the popular `image2prompt` [9, 21, 22], which uses stable diffusion to derive a textual prompt related to an input image. While in the future Web developers might directly annotate their images, leveraging `image2prompt` allows us to build an approximation which can be evaluated today without support from Web developers.

## 3.2 Proxy System

In theory, stable diffusion can be integrated in any open source Web browser. In practice, this is a time consuming effort which can anyway lead to suboptimal performance as browsers are very complex software. Instead, WebDiffusion offloads image generation to an external machine (stable diffusion server) and leverages a proxy system to intercept and operate on Web traffic (see Figure 1). This setup allows WebDiffusion to control how images should be served, either in a classic way or generated by the stable diffusion server.

As shown in Figure 1, legacy clients – mobile or desktop running any browser – are instructed to forward all their HTTP(S) traffic via `mitmproxy` [17]. This requires installing mitmproxy's root CA (Certificate Authority) to handle HTTPS traffic. Note that this step is only required for testing purposes, as a browser implementation would not require these steps. We assume all traffic is HTTP/2 [16].

Our proxy system consists of two `mitmproxy` proxies: one handling only requests for images, and one serving all the other requests. These two proxies are set in the testing client using a simple Proxy Auto-Configuration (PAC) file [8] with a regular expression matching requests for images towards one proxy and vice-versa.[1] This is necessary to allow the flexibility to emulate network connectivity (realized via Linux `tc` [19]) at the link between legacy clients and the proxy not handling image requests. In fact, the network characteristics of this link should not affect the images generated by the stable diffusion server, since we are emulating a client-side implementation. This is ensured as long as a fast connection (1 Gbps) is available between client, image proxy, and stable diffusion server.

For each request received, our proxy system performs a database lookup. For non-images, this lookup returns the content to be served. For images, the database lookup might return some textual prompt for image generation. This depends on the experiment running, e.g., some prompts might be missing when testing the client-based approach (see Section 3). In the presence of a textual prompt, the proxy contacts the stable diffusion server which generates the image. The stable diffusion server can be configured with different hardware to emulate different client capabilities. When the

[1]Note that this also requires to set the browser's `network.proxy.autoconfig_url.include_path` to true so as to enable full URL visibility in the PAC file with HTTPS.

image is generated, it is returned to the proxy which forwards it back to the client.

### 3.3 Stable Diffusion Server

Given the prompts derived using any of the above techniques, we then rely on stable diffusion for automated image generation. We chose stable diffusion for the reason discussed in Section 2, and because of its open source nature. We deployed the stable diffusion implementation of `replicate.com` using their `stability-ai/stable-diffusion` model [39]. It is a latent diffusion model that uses a fixed, pre-trained text encoder (CLIP ViT-L/14). The implementation supports running the model in a Docker container [34] with a cog [2] web server for serving the images. To create prompts from images, as needed by the server-based technique explained earlier, we have also deployed the `methexis-inc/img2prompt` model from `replicate.com`, which uses a slightly adapted version of the CLIP Interrogator notebook by @pharmapsychotic. CLIP Interrogator [1] uses the OpenAI CLIP models to test a given image against a variety of artists, mediums, and styles to study how the different models see the content of the image. It also uses the Bootstrapping Language-Image Pre-training (BLIP) [29] to generate an image caption to be used as a text prompt.

Generally speaking, the *quality* of images produced by stable diffusion increases with the number of inference steps, which in turn requires a longer generation time. The number of inference steps relates to how many denoising steps the model will use to improve the quality of the generated images. Generally it is recommended to use 50 denoising steps, which is usually sufficient to generate high-quality images. While experimenting with stable diffusion in the context of the Web, we realized that some textual prompts benefit more than others from more iterations, e.g., human faces versus a landscape. In this paper, we settle for a constant number of iterations (20) for fairness between images from different webpages. This number was chosen as a good compromise between quality and speed (about 2 seconds image generation time on our machine). From the different web images that were generated, we have noticed that the above number of inference is enough for generating a large portion of the web images, where any images that does not contain a human face had a good quality (for example food, landscape, objects, animals, etc.). However, an interesting direction for future work is to explore some dynamic stable diffusion settings based on the content of a textual prompt, and potentially additional webpage information. We refer the reader to Section 5 for more details. Apart from the number of inference all other stable diffusion parameters were kept to the default values: prompt strength (0.8), guidance scale (7.5), width and height (512x512 pixels). We randomly generated the seed to have a different images generated each time.

Our stable diffusion server currently runs on a Linux server equipped with Nvidia Tesla A40 GPU [6] with a 48 GB GDDR6 with error-correcting code (ECC), 300 W maximum power consumption, and a 696 GB/s GPU memory bandwidth. The server specifications are: Intel@Xeon(R) Bronze 3204 CPU@1.90GHz x 12, with 16 GiB RAM, and running Ubuntu 20.04.5 LTS.

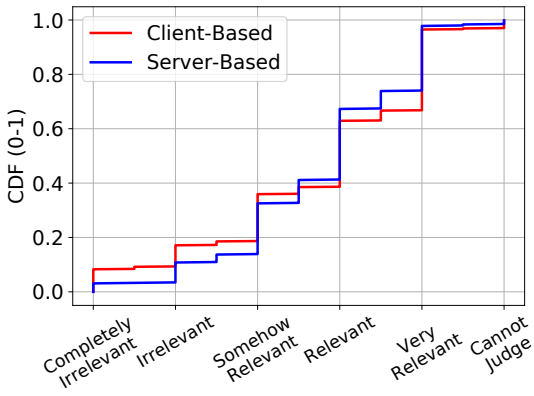



**Figure 3: CDF of how relevant client-based and served-based annotations are with respect to original images extracted from webpages (1,870 images from 200 webpages evaluated by 1,000 people each evaluating 10 random images).**

### 3.4 Web Interface

Last but not least, WebDiffusion offers a Web interface at *https://webdiffusion.ai/*. At this page, users can explore Web images or full webpages automatically generated using the above annotation schemes (see Appendix Figures 8, 9 and 10), while providing some feedback on the quality of the generated content. The Web interface accepts a parameter (`?type=[images, webpages]`) to control whether to show images or full webpages, and which annotation scheme to use – default is server-based, client-based is activated adding the suffix `_client`.

Figure 2 shows a visualization of WebDiffusion's Web interface when instrumented to show a image comparison. At the top of the page some textual prompt is offered to describe the images. This text was obtained using the techniques described above for image annotation. Next, two images are shown. One image is an original Web image pulled from a given website; the other image is originated via stable diffusion using the above textual prompt. Original and AI-powered images are randomly placed left or right. At the bottom of the page, the user is asked to rank the quality of AI-generated images between 1 (very poor) and 5 (very good). In case the user is not capable to distinguish the original and AI-generated image, we ask to mark their answer with a "*". We integrated WebDiffusion's Web GUI with Prolific [35], a popular crowdsourcing website which allows us to quickly scale user studies.

# 4 EVALUATION

This section sets out to evaluate the applicability of AI-powered text-to-image generation [5, 37, 39] to the Web. Our evaluation revolves around three main parts. First, we crawl 200 webpages and quantify the *feasibility* of the techniques discussed in the previous section for image annotation: client-based, and server-based. Next, we evaluate the *quality* of AI-generated webpages, or webpages where images are automatically generated via stable diffusion [39]. We do this by performing several user studies involving about 900 participants provided by Prolific [35]. Finally, we benchmark the performance of AI-generated webpages using the following metrics: bandwidth usage, CPU/GPU consumption, and Web performance metrics (SpeedIndex, Page Load Time, and 99% visual complete time [23]). We use WebDiffusion to perform all the experiments discussed in this section. We use WebPageTest [43] to both automate webpage loads and collect browser telemetry data. We choose Firefox since it allowed to configure the `network.http.connection-timeout` which was not possible in Chrome.

## 4.1 Feasibility of Images Annotation

We start our analysis by investigating the *feasibility* of automated image annotation, a key component of WebDiffusion (see Section 3.1). To do so, we crawl (using Selenium) the top 500 webpages from Tranco's top one million list. About 25% of these webpages fail to load; for the remainder 75% of webpages, the crawler achieves a 70% success rate (260 webpages); failures are due to webpage formatting, e.g., lack of `img` tags with clear `src` or `background-image` attributes. Next, we filter webpages hosting inappropriate or non-English content, given that stable diffusion currently operates only with English prompts. This results in 200 webpages with a total of 1,870 images for which some client-based annotation is produced. We finally run `image2prompt` to derive the extra annotation needed by the server-based scheme.

Client-based annotation is a best effort attempt to locate webpage text which should refer to an image. As such, potential mistakes are possible. The output of `image2prompt` is instead highly accurate – given their usage of BLIP image-captioning technique which is shown to outperform all state-of-the-art techniques [29] – but generic. For example, an image of the Golden Gate bridge might generate a label such as "a bridge with some clouds in the background"; conversely, the text in the webpage might describe more precisely which kind of bridge and its scenery (see Figure 7 in the appendix for additional examples on the difference between both texts).

We evaluate the correctness of the "client-based" annotation scheme using Natural Language Processing (NLP) of sentences similarity. We use the `bert-base-nli-mean-tokens` [11] model to generate text embeddings for both annotation schemes, each containing 768 values; we then compute the cosine similarity [28] between both texts' embeddings. The results show that both the median and average similarities are about 50% (with 75th percentile at 60%, and 25th percentile at 40%). This result is encouraging since it suggests some intersection between the two annotation schemes. Of course, perfect intersection is unlikely given that the output of `image2prompt` is highly generic. However, systematic misbehavior of our client-based annotation scheme would result in much lower cosine similarity values.

| | Images Server | Images Client | Webpages Server | Webpages Client |
|---|---|---|---|---|
| **Size** | 409 | 409 | 25 | 25 |
| **Participants** | 410 | 413 | 30 | 30 |
| **Responses** | 4,110 | 4,160 | 300 | 300 |
| **Age Range** | 19-63 | 19-63 | 20-40 | 18-52 |
| **Gender (M/F)** | 239/171 | 197/216 | 20/10 | 20/10 |
| **# Countries** | 29 | 26 | 9 | 10 |

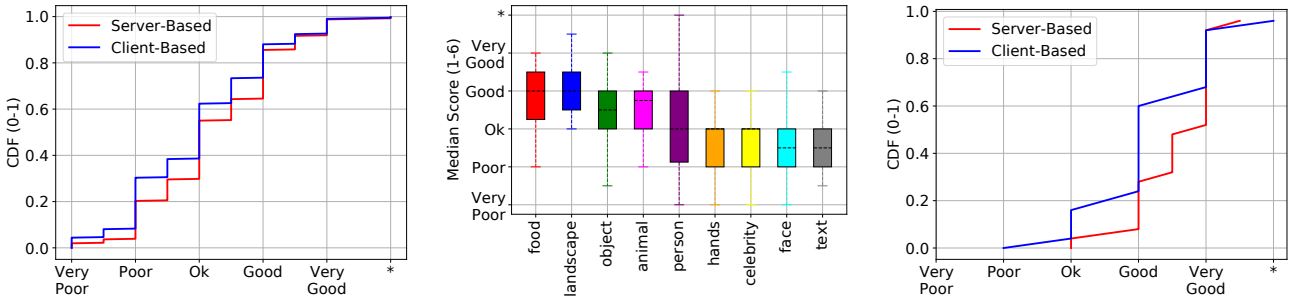
**Table 1: User study summary. *Server* is short for server-based, and *client* is short for client-based.**

To further comment on the quality of WebDiffusion's annotation schemes, we perform a user study. To do so, we extend WebDiffusion's Web interface (activated with parameter `?type=[scale, scale_prompt]`, see Section 3.4) to show a single (original) image along some descriptive text and ask for how *relevant* such text is to the image. Scores are in the range of "completely irrelevant" to "very relevant". We also allow study participants to respond "cannot judge", for the cases when participants are not knowledgeable of the image, e.g., when the text refers to a celebrity they do not know.

Figure 3 shows the Cumulative Distribution Function (CDF) of the median score (10,000 scores from 1,000 people) for the 1,870 images we have collected from 200 websites, distinguishing between the client-based" and "server-based" annotations shown on top of each image. The figure shows a success rate – score higher than "irrelevant" – of 80% for the client-side approach, and close to 90% for the server-side approach. This result has two important implications. First, not even the highly accurate `image2prompt` is capable of being 100% "relevant", according to our study participants. This is not unexpected, especially with no control on the input images which can assume a very abstract form (see Figure 7 in the Appendix for sample images with their text description). It follows that the accuracy of the "client-based" approach is quite high, as it is only 10% worse from the best attainable by a sophisticated AI.

## 4.2 Quality of AI-Generated Webpages

This section investigates the quality of AI-generated webpages using user feedback collected on Prolific [35]. We first collect feedback per image composing a webpage, and then focus on the full webpages. Table 1 summarizes the user study. We collect 10 user feedback for the two versions (client-based and server-based) of the 409 images extracted from 25 websites randomly selected from the 200 previously crawled. Next, we also collect 10 user feedback for two versions of each webpage, for a total of 8,840 user feedback. We ask Prolific to offer high quality participants which are fluent in English, so that they can understand the image descriptions. Among 884 study




(a) CDF of median score per image comparison, distinguishing between client and server-based image generation.

(b) Boxplot of median image scores as a function of their tags. Server-based image generation.

(c) CDF of median score per webpgage comparison, distinguishing between client and server-based image generation.

**Figure 4: Results from the user study. *Client-based* refers to image generated on the client side, *i.e.,* only using contextual information like `alt text`. *Server-based* refers to image generated on the server side, *i.e.,* assuming availability of both contextual information and original image.**

participants, the minimum *approval rate*, *i.e.,* the past rate of approval of their work, we recorded was 94%, with 80% of the participants having an approval rate of 99-100%.

We start by investigating the quality of AI-generated Web images. Figure 4(a) shows the CDF of the median score (from 10 users) among images, distinguishing between the "client-based" and "server-based" approach. The figure shows that, regardless of the approach, the majority of scores are *positive*, *i.e.,* "ok" or higher. Precisely, 70% of the median scores are better or equal than "ok" when considering the "server-based" annotation scheme. This number drops to 60% in the "client-based" approach. A reduction is expected given the more challenging conditions (lack of access to the original image) in which the client-based approach operates.

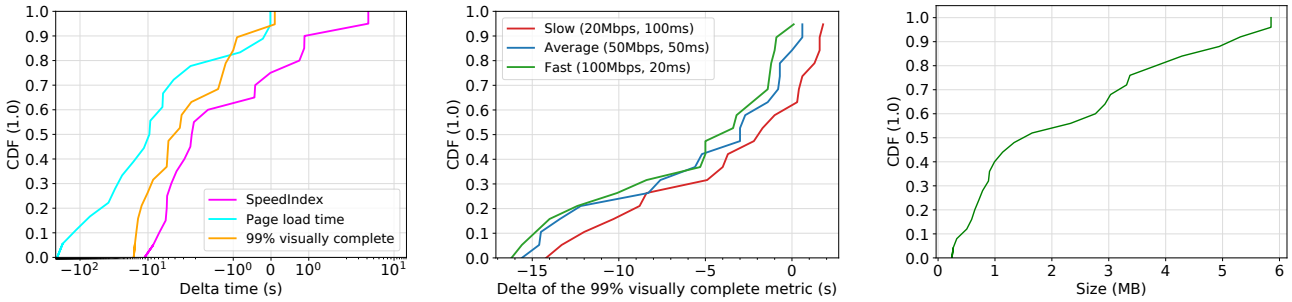
Next we dig deeper into the collected scores with respect to the *content* of the images. We manually tag each (original) image using the set of tags shown on the x-axis in Figure 4(b); we limit to a maximum of two tags per image, by considering the two predominant features. For example, the image in Figure 2 is labelled as "object". Note that we see the tags "person" and "face" which might seem redundant; however, these two tags are never used together, and their selection is based on whether a face is the main feature of an image. For example a closeup of a person would be labelled as "face", versus a group of people working is labelled as "person".

Figure 4(b) shows one boxplot per image label, where each boxplot contains the median scores of all the images tagged with such label. We only show results for the server-based annotation scheme since it shows a similar trend as the client-based approach. If we focus on the median, the figure shows that images containing "food" and "landscape" achieve the best performance, with a median score of "good" and rarely showing negative scores. Next, images labelled as "object" or "animal" also perform well, with median comprised between "ok" and "good". Most of the negative scores are instead due to images containing people, especially when hands and faces are in the foreground (median score is between "poor" and "ok"). A similar issue appears with images containing celebrities, which rarely achieve a score better than "ok". A much wider distribution is observed for images containing at least one "person". We conjecture that the low scores are due to the same reason discussed above for images labelled as "face" and "hands", which might still be noticeable and problematic. The opposite is instead responsible of the higher scores, e.g., when a group of people is shown walking from far away or from behind. Finally, images including some text are often problematic, due to the inability of stable diffusion to re-produce accurate text with respect to the original image – especially with an automated annotation.

Finally, Figure 4(c) shows the CDF of the median score (from 10 users) among webpages, distinguishing between the "client-based" and "server-based" approach. Compared to Figure 4(c), the overall score has improved, with only 5% of the scores being "poor" for the harder client-side approach. Indeed, 50% of the scores for the server-based approach are "very good" or "*", suggesting that study participants could not even detect which webpage was the AI-generated one. The reason for the score increase compared to image-to-image comparison is that webpages are more complicated, composed by a collection of text/images. While for the images study participants can focus on low level details, such as potential hands deformity, this is less likely on a webpage where eyeballs tend to move quickly, potentially ignoring some images. Still the client-based approach has an overall lower quality, with overall less "very good" scores, confirming the previous trend.

## 4.3 Client and Webpage Performance

This section compares the performance of original webpages with their WebDiffusion's counterparts—where images are not transferred but generated using stable diffusion. We also benchmark the cost of generating such images in terms of CPU/GPU usage, and bandwidth consumption. We evaluate




(a) CDF of the delta for SI, PLT, 99% VC. Slow network (100 Mbps, 20 ms RTT)

(b) CDF of 99% VC across slow, average, and fast network links.

(c) CDF of bandwidth savings.

**Figure 5: Performance evaluation of WebDiffusion-ed webpages versus original webpages.**

the previous 25 webpages under three network conditions emulated using `TC` (see Figure 1): *slow* (symmetric 20 Mbps and 100 ms RTT), *average* (symmetric 50 Mbps and 50ms RTT), and *fast* (symmetric 100 Mbps and 20ms RTT). These network conditions were selected assuming typical speeds of home connections [4], where we envision a powerful machine like our stable diffusion server could run. We instead ignore slower speeds, e.g., available in 3G, since a mobile phone's hardware is far from being capable of running such complex models. We use textual prompts derived with the "server-side" scheme, although we did not measure any difference in the duration of image generation using the two approaches.

**Web Performance:** We start by investigating the potential slowdown for WebDiffusion-ed webpages, with respect to three Web performance metrics: SpeedIndex (SI), Page Load Time (PLT), and the time at which the webpage was 99% visually complete (99% VC) [23]. SI measures how quickly the visible content of a webpage is displayed during a page load. PLT measures the total time it takes the browser to download and visualize the webpage, including content located below the fold. Finally, 99% VC measures the time the browser takes to visualize 99% of the above the fold webpage content.

Figure 5(a) shows the CDF of the *delta* for each metric, derived as the difference between the metric computed for the original webpage, and the metric computed for the WebDiffusion-ed webpage, *i.e.,* a negative value indicates a slowdown, and a positive value indicates a speedup. Each delta value in the plot is the median out of 10 runs; we setup *fast* network settings as a worst case for WebDiffusion-ed webpages. The figure shows median slowdowns of 10.5 seconds (PLT), 5 seconds (99% VC), and 2.3 seconds (SI). Surprisingly, 25% of the webpages have a faster SI when relying on stable diffusion. We visually investigated the webpage loads, and concluded that both SI and PLT are not representative metrics for these experiments. WebDiffusion-ed webpages are visually ready but lack images, which are slowly added as generated. This triggers an early SI which unfairly advantages WebDiffusion's evaluation. In contrast, the PLT metric unfairly disadvantages WebDiffusion's evaluation given that many webpages have a high number of images below the fold that the user might not even scroll to. Conversely, the 99% VC is more fair since it captures the time when all above-the-fold images are correctly visualized. Accordingly, in the remainder of this analysis we leverage the 99% VC metric.

Figure 5(b) shows shows the CDF of the delta for the 99% VC metric under three network conditions: slow, average, and fast. The figure shows that the slowdown of WebDiffusion-ed webpages reduces as more challenging network conditions are met, e.g., a 50% reduction at the median (from 5 seconds down to 2.2 seconds) when considering a slow connection (20Mbps, 100ms). This is intuitive since the images of WebDiffusion-ed webpages are not affected by the network slowdown. Indeed, 40% of webpages benefit from a speedup in presence of a slow network connection.

Finally, Figure 5(c) quantifies the bandwidth savings (MB) due to not transferring Web images when using stable diffusion. Overall, the majority of webpages we tested enjoy multiple MBs of savings (median of 1.3MB), and even more than 5MB for the 10% heaviest webpages. These savings are marginal for devices on WiFi, the likely connectivity scenario for the powerful devices we are assuming. However, these savings can be significant for the content providers. For example,

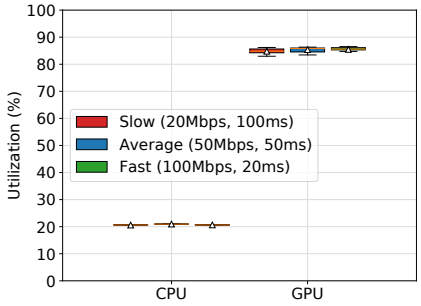



**Figure 6: Analysis of additional resources needed by a client when loading WebDiffusion-ed webpages.**

let's consider a popular newspaper with 1 million daily views; a median saving of 1.3 MB per view would account for daily savings of 1.3 TB.

**Resource Usage:** We now estimate the additional resources needed by a client when loading WebDiffusion-ed webpages. Figure 6 shows CPU/GPU usage measured at the stable diffusion server during the time it was busy generating images for each webpage. Each boxplot refers to a different network condition (slow, average, and fast). The figure shows a constant CPU usage of about 20% independent of the network conditions. As for the GPU utilization, the figure shows a steady utilization of about 85%, again almost independent of the network. GPU memory usage is also constant at 8 GB.

## 5 CONCLUSION AND DISCUSSION

This paper argues that the World Wide Web is an interesting application for text-to-image generation models, given that webpages consist of a collection of text and images, normally linked with some semantic meaning. Accordingly, this paper provides, to the best of our knowledge, the first investigation of the potential of integrating stable diffusion [32], a popular text-to-image generation model, in the Web. To do so we built and released `WebDiffusion`, a tool to emulate a Web leveraging stable diffusion model for image generation.

WebDiffusion uses a crawler to generate local versions of webpages, and augment them with textual prompts for their images. Such prompts are automatically derived using a *client-based* approach which emulates a browser implementation, *i.e.,* with no access to the original image, and a *server-based* approach with access to the original image, thus emulating the participation of a Web developer. WebDiffusion supports legacy browsers by offloading text-to-image generation to a stable diffusion server. This is achieved by a system of proxies which intercept network requests and trigger image generation, if needed. Last but not least, WebDiffusion offers a Web interface (https://webdiffusion.ai/) which integrates with Prolific [35], a popular crowdsourcing platform, to gather worldwide eyeballs reporting on the quality of Web images produced by stable diffusion.

We evaluate WebDiffusion via a combination of in-lab experiments and user studies involving up to 200 webpages (and 1,870 images) and 1,900 participants. We find that automated image annotation is mostly accurate, close to 90% for the server-side approach. We also find that 95% of webpages are scored "ok" or higher ("good" and "very good") even assuming the challenging client-based approach. Performance wise, most webpages see a size reduction of multiple MBytes, at the cost of few seconds slowdowns, 20% CPU increase, and the need of a powerful GPU (Nvidia Tesla A40) occupied at 85% during image generation.

The adoption of text-to-image generation models in the Web context is an interesting novel area of product/research, which opens up to the following potential follow-ups.

**Performance Optimizations:** This paper has highlighted two areas of improvements when applying text-to-image generation to the Web. First, the *quality* of images involving human subjects, notably face and hands deformation, and second the *speed* at which images are generated. Quality and speed are directly related to the number of iterations; in this paper we settle for a constant number of inference steps (20) for fairness between images of different webpages (see Section 3.3). An interesting research area is to explore stable diffusion's parameters as a function of the textual prompt content.

We experimented with modifying the seed and the number of inference steps of the model until the sampled output was artefact-free and acceptable. However, such a detailed sampling process could not be upscaled automatically given the difficulty in detecting the artefact or fit of the generated images without human intervention. Nonetheless, we concluded that dynamic parameters — modifying the seed and iterations parameters depending on image content — could also be a viable method to optimize the performance of WebDiffusion and reduce the artefacts in its output.

Indeed, we observed that stable diffusion often produces artefacts when generating images of faces and hands even with a large number of iterations ($> 100$, increasing the image generation time to above 15 seconds), especially when there are more than one subject in the produced image. We also noted that stable diffusion is capable of generating convincing and realistic portrayals of landscapes even with a low number of iterations (as low as 10, which can generate an image within a second). These results may be attributed to the fact that the stable diffusion model was trained on the LAION Aesthetic dataset [40], comprised of numerous paintings and artistic photography, meaning that the training set would need to be diversified in order to produce more realistic, artefact-free results. Alternatively, one could consider extending WebDiffusion to other text-to-image generation models. For example, the GFPGAN model is a popular method that addresses the issue of distorted faces. This model helps restore faces in samples and improves the overall realistic quality of the generated image [42].

Another area for optimization is the computation speed of the stable diffusion model. As stable diffusion is still a recent topic of exploration, there are constant developments in the areas of its computational efficiency and speed. One example of this is switching a large portion of the cross attention operations within the sampling process to FLASH attention, which can increase computation speeds by nearly 50 percent while decreasing VRAM usage [14, 24].

**Privacy-Preserving Advertisement:** On-line advertisement heavily relies on tracking, where user profiles and information are auctioned, in real time, to tens or hundred of ad networks. The privacy concerns of such approach are clear, and they have motivated the rise of adblockers and private models of advertising like the one offered by the Brave browser [18]. In Brave, user profiles are local and never leave the browser. Still, network requests are needed to fetch the ads to be shown, e.g., GET an image containing a sport car for a car enthusiast profile. These requests are answered in the cloud,

where some potential privacy leak is possible. Local ads generation remove even the slightest privacy leak from the, already private, Brave model.

**Broken URLs:** The Web is an ever-growing ecosystem and, as such, many webpages are often unreachable or miss some important content. The Wayback Machine [15] is an interesting approach to deal with broken pages, since it currently offers 751 billion saved webpages. Our client-based approach can help in such scenarios, where missing images can be recovered by mean of the contextual information available in the page.

## REFERENCES


[1] clip-interrogator. https://github.com/pharmapsychotic/clip-interrogator. Accessed: 2022-10-13.

[2] Cog: Containers for machine learning. https://github.com/replicate/cog. Accessed: 2022-10-13.

[3] Deforum stable diffusion. https://colab.research.google.com/github/deforum/stable-diffusion/blob/main/Deforum_Stable_Diffusion.ipynb#scrollTo=LBamKxcmNI7-.

[4] Etisalat internet and landline plans. https://www.etisalat.ae/b2c/eshop/viewProducts?category=homePlans&subCategory=cat1100023&catName=eLife_plan&listVal=eLife_plan&locale=EN. Accessed: 2022-10-13.

[5] Midjourney. https://www.midjourney.com/home/.

[6] NVIDIA A40 the world's most powerful data center gpu for visual computing. https://www.nvidia.com/en-us/data-center/a40/. Accessed: 2022-10-13.

[7] pixabay. https://pixabay.com/.

[8] Proxy auto-configuration (pac) file - http: Mdn. https://developer.mozilla.org/en-US/docs/Web/HTTP/Proxy_servers_and_tunneling/Proxy_Auto-Configuration_PAC_file.

[9] Replicate - img2prompt. https://replicate.com/methexis-inc/img2prompt.

[10] Selenium. https://www.selenium.dev/.

[11] Sentence-transformers/bert-base-nli-mean-tokens · hugging face. https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens.

[12] Stable diffusion, ai generated images. https://stablediffusion.fr/gallery/interior-design.

[13] tranco. https://tranco-list.eu/, journal=Tranco.

[14] Transformer for stable diffusion. https://nn.labml.ai/diffusion/stable_diffusion/model/unet_attention.html#section-45.

[15] Wayback machine. https://web.archive.org/.

[16] Hypertext transfer protocol version 2 (http/2). https://datatracker.ietf.org/doc/html/rfc7540, 2015.

[17] A free and open source interactive https proxy. https://mitmproxy.org/, 2021. Accessed: 2021-04-28.

[18] Brave: Secure, fast & private web browser with adblocker. https://brave.com/, Oct 2022.

[19] W. Almesberger et al. Linux network traffic control—implementation overview, 1999.

[20] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *CoRR*, abs/2105.05233, 2021.

[21] X. He and L. Deng. Deep learning for image-to-text generation: A technical overview. *IEEE Signal Processing Magazine*, 34(6):109–116, 2017.

[22] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or. Prompt-to-prompt image editing with cross attention control, 2022.

[23] T. Hoßfeld, F. Metzger, and D. Rossi. Speed index: Relating the industrial standard for user perceived web performance to web qoe. In , pages 1–6, 2018.

[24] W. Hua, Z. Dai, H. Liu, and Q. Le. Transformer quality in linear time. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of Machine Learning Research*, volume 162 of *Proceedings of Machine Learning Research*, pages 9099–9117. PMLR, 17–23 Jul 2022.

[25] C. Kelton, M. Varvello, A. Aucinas, and B. Livshits. Browselite: A private data saving solution for the web. In , pages 305–316, 2021.

[26] T. Kupoluyi, M. Chaqfeh, M. Varvello, W. Hashmi, L. Subramanian, and Y. Zaki. Muzeel: A dynamic javascript analyzer for dead code elimination in today's web. *arXiv preprint arXiv:2106.08948*, 2021.

[27] Lacuna-JDCE. Lacuna-jdce/lacuna: Lacuna is a javascript dead code elimination framework written in node.js. https://github.com/Lacuna-JDCE/Lacuna.

[28] B. Li and L. Han. Distance weighted cosine similarity measure for text classification. In H. Yin, K. Tang, Y. Gao, F. Klawonn, M. Lee, T. Weise, B. Li, and X. Yao, editors, , pages 611–618, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[29] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In , 2022.

[30] R. Metz. Ai won an art contest, and artists are furious | cnn business. https://edition.cnn.com/2022/09/03/tech/ai-art-fair-winner-controversy/index.html, Sep 2022.

[31] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *ArXiv*, abs/2112.10741, 2021.

[32] S. Patil, P. Cuenca, N. Lambert, and P. von Platen. Stable diffusion with diffusers. https://huggingface.co/blog/stable_diffusion.

[33] E. Portis and A. Ranganath. Media: 2022: The web almanac by http archive. https://almanac.httparchive.org/en/2022, Oct 2022.

[34] A. M. Potdar, N. D G, S. Kengond, and M. M. Mulla. Performance evaluation of docker container and virtual machine. *Procedia Computer Science*, 171:1419–1428, 2020. Third International Conference on Computing and Network Communications (CoCoNet'19).

[35] Prolific. A higher standard of online research. https://www.prolific.co/.

[36] G. Qiong and W. Li. An optimization method of javascript redundant code elimination based on hybrid analysis technique. In , pages 300–305, 2020.

[37] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022.

[38] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021.

[39] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In , pages 10684–10695, June 2022.

[40] C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki. LAION-400M: open dataset of clip-filtered 400 million image-text pairs. *CoRR*, abs/2111.02114, 2021.

[41] P. Wang, M. Varvello, C. Ni, R. Yu, and A. Kuzmanovic. Web-lego: trading content strictness for faster webpages. In , pages 1–10. IEEE, 2021.

[42] X. Wang, Y. Li, H. Zhang, and Y. Shan. Towards real-world blind face restoration with generative facial prior. In , pages 9168–9178, 2021.

[43] WPO-Foundation. Webpagetest - website performance and optimization test. https://www.webpagetest.org/, 2022. Accessed: 2022-10.

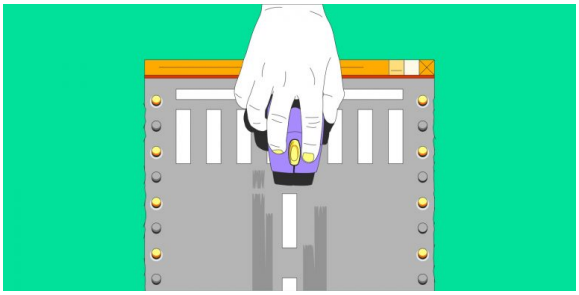
[44] Z. Xiao, K. Kreis, and A. Vahdat. Tackling the generative learning trilemma with denoising diffusion gans, 2021.


## A APPENDIX

In this appendix we show few images that received a low score during the user study associated with the feasibility evaluation (Section 4.1). In addition, we also show three webpages in three forms: original, AI-generated – using both the client-based and server-based annotation (Section 3.1).

(a) A golden statue of a person holding a sword; Akriti brass art wares antique brass metal lord ganesha on fan wall hanging for entrance door; living room; decorative

(b) A colorful tree with a man and a dog on it; Mozilla puts people before profit; creating products; technologies and programs that make the internet healthier for everyone. learn more about us more power to you.

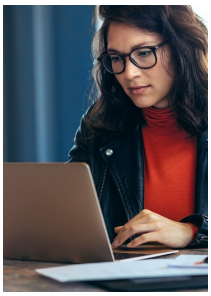


(c) A hand is holding a pencil in a drawing style; An effective landing page needs a layout; a way to implement it; clear; concise copy; and ongoing evaluation—all with the goal of driving visitors to your cta. how to make a landing page

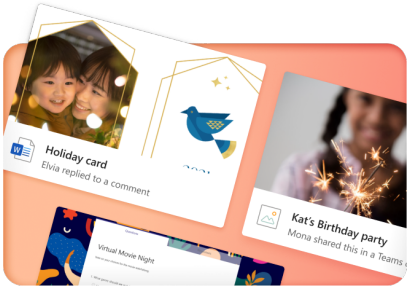
(d) A castle made out of colored plastic blocks; Picassotiles 60 piece set 60pcs magnet building tiles clear magnetic 3d building blocks construction playboards


(e) A man with a stack of cassettes on his head; By open mike eagle past and present blur on the new lp from the celebrated mc. read more a tape called component system with the auto reverse

(f) The front of a building with columns and a clock; Nih is the nations medical research agency; supporting scientific studies that turn discovery into health. nih-at-a-glance-whoweare.jpg who we are who we are



(g) A woman sitting at a table using a laptop computer; Visit cisco hybrid work index to understand the inclusive collaboration experiences driving hybrid work. the future of work is hybrid

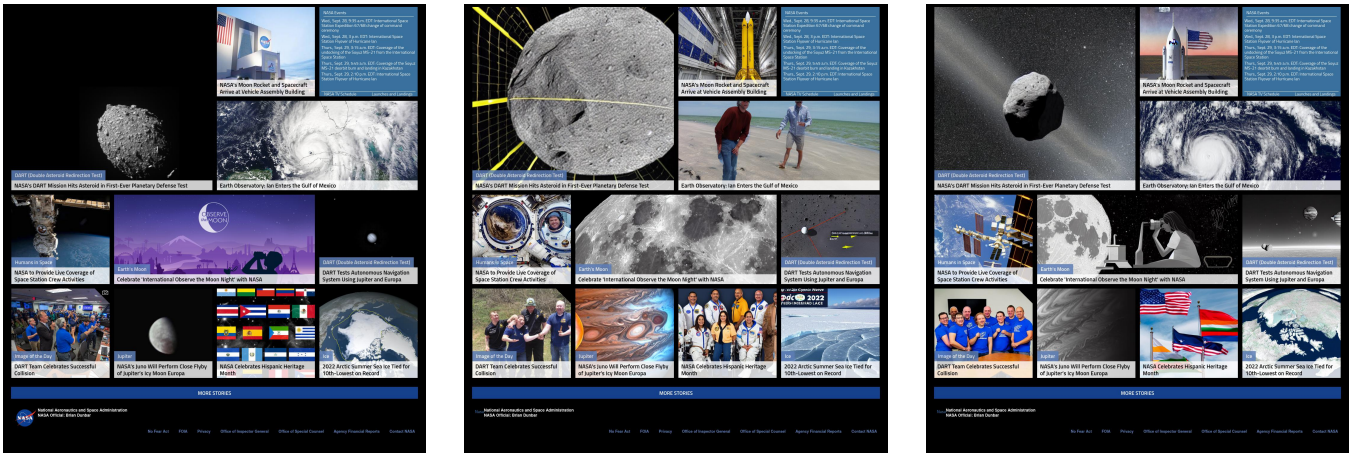



(h) A couple of pictures of a girl and a boy; Your files and memories stay safe and secure in the cloud; with 5 gb for free and 1 tb+ if you go premium store with confidence visual renderings demonstrating some of the content types that can be safely stored in office

(i) A woman standing in front of a large truck; Lady of the gobi;a rare woman among thousands of coal truck drivers fights for survival on the hazardous mining road from mongolia to china;watch now;25.13 coal truck driver; maikhuu; in the film lady of the gobi directed by khoroldorj choijoovanchig

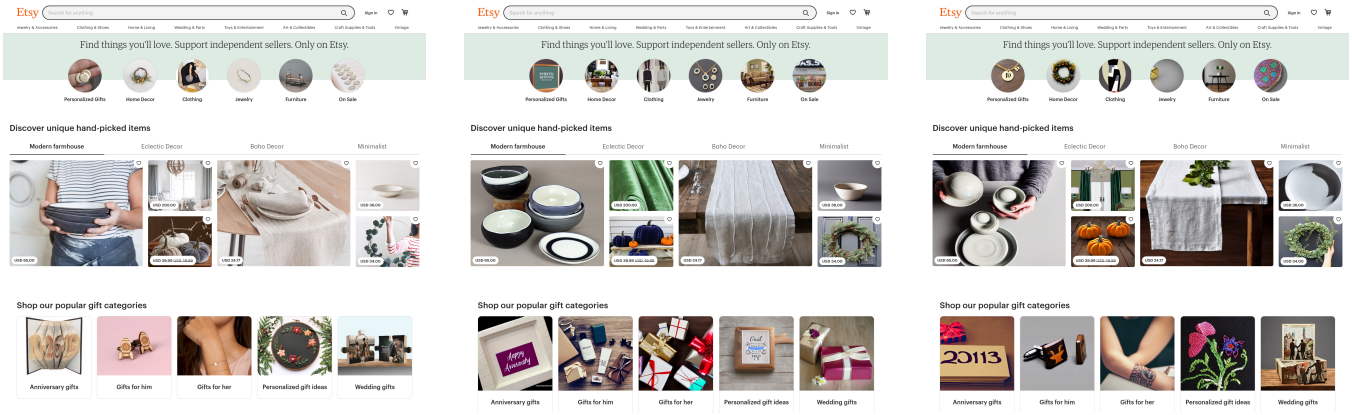

(j) A black and white photo with the words all about it; All about the white lotus season 2

**Figure 7: Several samples of Web images with their img2prompt (first sentence before the semicolon) and extracted text descriptions (text following the first semicolon) that were given a low score ("irrelevant" or "very irrelevant") by our study participants.**



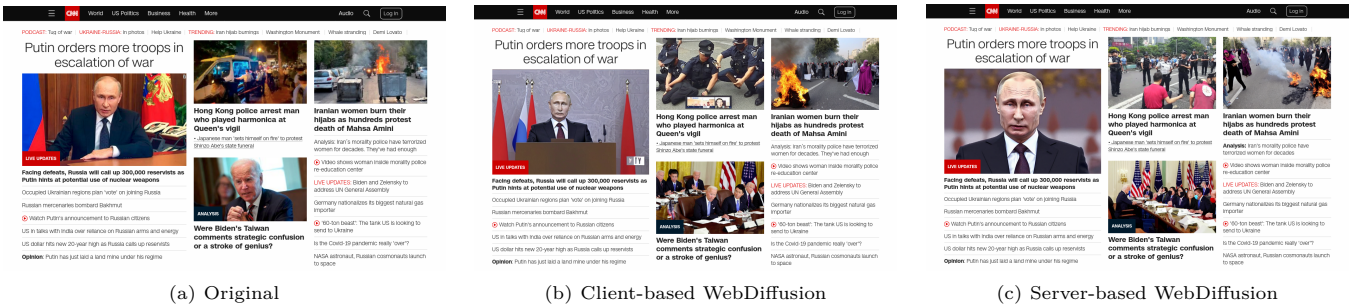
(a) Original (b) Client-based WebDiffusion (c) Server-based WebDiffusion

**Figure 8: Comparing the www.nasa.gov webpage with both their Client-Based and Server-Based WebDiffusion counterparts.**



(a) Original (b) Client-based WebDiffusion (c) Server-based WebDiffusion

**Figure 9: Comparing the www.etsy.com webpage with both their Client-Based and Server-Based WebDiffusion counterparts.**



(a) Original (b) Client-based WebDiffusion (c) Server-based WebDiffusion

**Figure 10: Comparing the www.cnn.com webpage with both their Client-Based and Server-Based WebDiffusion counterparts.**