How Stable Diffusion Could Develop as a Mainstream Consumer Product

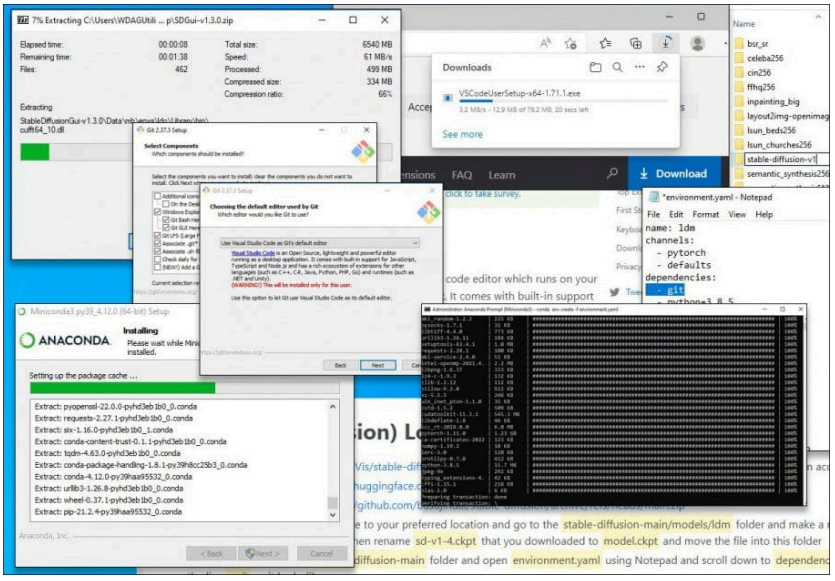
Originally published at https://web.archive.org/web/20220922163600/https://www.unite.ai/how-stable-diffusion-could-develop-as-a-mainstream-consumer-product/

September 15, 2022

Ironically, Stable Diffusion, the new AI image synthesis framework that has taken the world by storm, is neither stable nor really that 'diffused' – at least, not yet.

The full range of the system's capabilities are spread across a varying smorgasbord of constantly mutating offerings from a handful of developers frantically swapping the latest information and theories in diverse colloquies on Discord – and the vast majority of the installation procedures for the packages they are creating or modifying are very far from 'plug and play'.

Rather, they tend to require command-line or BAT-driven installation via GIT, Conda, Python, Miniconda, and other bleeding-edge development frameworks – software packages so rare among the general run of consumers that their installation is frequently flagged by antivirus and anti-malware vendors as evidence of a compromised host system.



Only a small selection of stages in the gauntlet that the standard Stable Diffusion installation currently requires. Many of the distributions also require specific versions of Python, which may clash with existing versions installed on the user's machine – though this can be obviated with Docker-based installs and, to a certain extent, through the use of Conda environments.

Message threads in both the SFW and NSFW Stable Diffusion communities are flooded with tips and tricks related to hacking Python scripts and standard installs, in order to enable improved functionality, or to resolve frequent dependency errors, and a range of other issues.

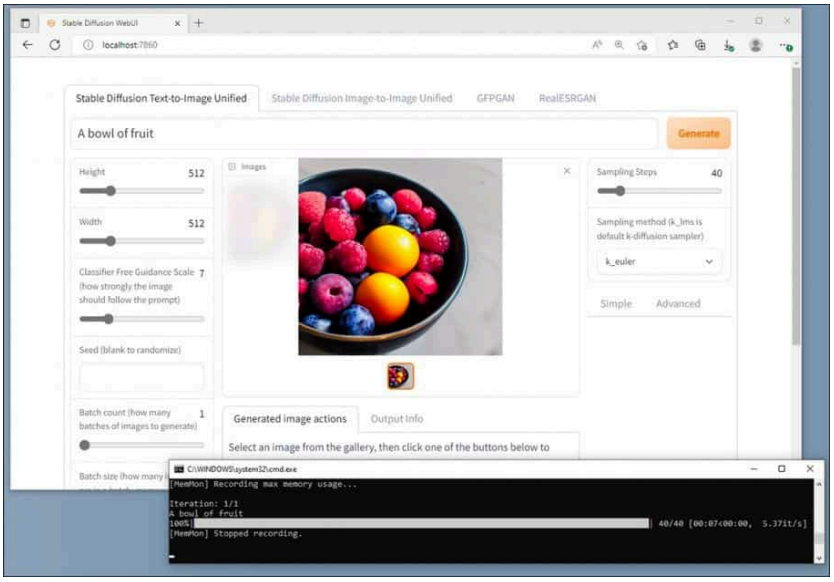
This leaves the average consumer, interested in creating amazing images from text prompts, pretty much at the mercy of the growing number of monetized API web interfaces, most of which offer a minimal number of free image generations before requiring the purchase of tokens.

Additionally, nearly all of these web-based offerings refuse to output the NSFW content (much of which may relate to non-porn subjects of general interest, such as 'war') which distinguishes Stable Diffusion from the bowdlerized services of OpenAI's DALL-E 2.

'Photoshop for Stable Diffusion'

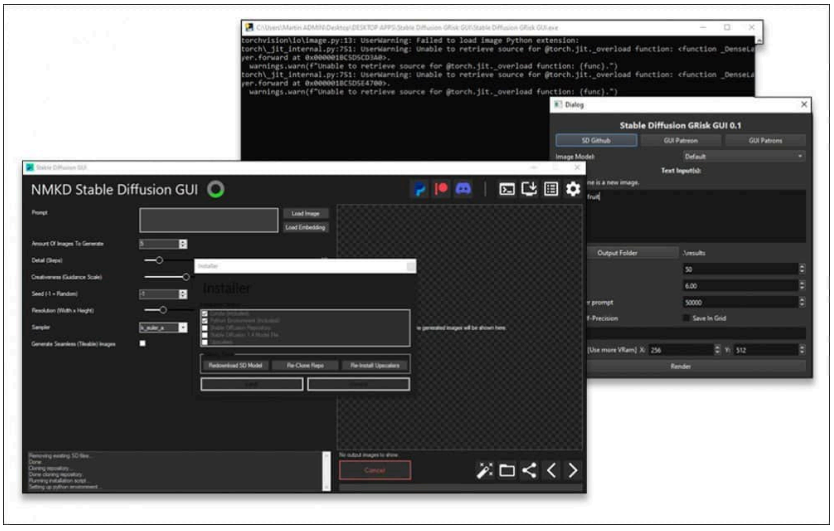
Tantalized by the fabulous, racy or other-worldly images that populate Twitter's #stablediffusion hashtag daily, What the wider world is arguably waiting for is *'Photoshop for Stable Diffusion'* – a cross-platform installable application that folds in the best and most powerful functionality of Stability.ai's architecture, as well as the various ingenious innovations of the emerging SD development community, without any floating CLI windows, obscure and ever-changing install and update routines, or missing features.

What we currently have, in most of the more capable installations, is a variously elegant web-page straddled by a disembodied command-line window, and whose URL is a localhost port:



Similar to CLI-driven synthesis apps such as FaceSwap, and the BAT-centric DeepFaceLab, the 'prepack' install of Stable Diffusion shows its command-line roots, with the interface accessed via a localhost port (see top of image above) which communicates with the CLI-based Stable Diffusion functionality.

Without doubt, a more streamlined application is coming. Already there are several Patreon-based integral applications that can be downloaded, such as GRisk and NMKD (see image below) – but none that, as yet, integrate the full range of features that some of the more advanced and less accessible implementations of Stable Diffusion can offer.



Early, Patreon-based packages of Stable Diffusion, lightly 'app-ized'. NMKD's is the first to integrate the CLI output directly into the GUI.

Let's take a look at what a more polished and integral implementation of this astonishing open source marvel may eventually look like – and what challenges it may face.

Legal Considerations for a Fully-Funded Commercial Stable Diffusion Application

The NSFW Factor

The Stable Diffusion source code has been released under an extremely permissive license which does not prohibit commercial re-implementations and derived works that build extensively from the source code.

Besides the aforementioned and growing number of Patreon-based Stable Diffusion builds, as well as the extensive number of application plugins being developed for Figma, Krita, Photoshop, GIMP, and Blender (among others), there is no *practical* reason why a well-funded software development house could not develop a far more sophisticated and capable Stable Diffusion application. From a market perspective, there is every reason to believe that several such initiatives are already well underway.

Here, such efforts immediately face the dilemma as to whether or not, like the majority of web APIs for Stable Diffusion, the application will allow Stable Diffusion's native NSFW filter (a fragment of code), to be turned off.

'Burying' the NSFW Switch

Though Stability.ai's open source license for Stable Diffusion includes a broadly interpretable list of applications for which it may *not* be used (arguably including pornographic content and deepfakes), the only way a vendor could effectively prohibit such use would be to compile the NSFW filter into an opaque executable instead of a parameter in a Python file, or else enforce a checksum comparison on the Python file or DLL that contains the NSFW directive, so that renders cannot occur if users alter this setting.

This would leave the putative application 'neutered' in much the same way that DALL-E 2 currently is, diminishing its commercial appeal. Also, inevitably, decompiled 'doctored' versions of these components (either original Python runtime elements or compiled DLL files, as are now used in the Topaz line of AI image enhancement tools) would likely emerge in the torrent/hacking community to unlock such restrictions, simply by replacing the obstructing elements, and negating any checksum requirements.

In the end, the vendor may choose to simply repeat Stability.ai's warning against misuse that characterizes the first run of many current Stable Diffusion distributions.

However, the small open source developers currently using casual disclaimers in this way have little to lose in comparison to a software company which has invested significant amounts of time and money in making Stable Diffusion full-featured and accessible – which invites deeper consideration.

Deepfake Liability

As we have recently noted, the LAION-aesthetics database, part of the 4.2 billion images on which Stable Diffusion's ongoing models were trained, contains a great number of celebrity images, enabling users to effectively create deepfakes, including deepfake celebrity porn.

From our recent article, four stages of Jennifer Connelly over four decades of her career, inferred from Stable Diffusion.

This is a separate and more contentious issue than the generation of (usually) legal 'abstract' porn, which does not depict 'real' people (though such images are inferred from multiple real photos in the training material).

Since an increasing number of US states and countries are developing, or have instituted, laws against deepfake pornography, Stable Diffusion's ability to create celebrity porn could mean that a commercial application that's not entirely censored (i.e. that can create pornographic material) might still need some ability to filter perceived celebrity faces.

One method would be to provide a built-in 'black-list' of terms that will not be accepted in a user prompt, relating to celebrity names and to fictitious characters with which they may be associated. Presumably such settings would need to be instituted in more languages than just English, since the originating data features other languages. Another approach could be to incorporate celebrity-recognition systems such as those developed by Clarifai.

It may be necessary for software producers to incorporate such methods, perhaps initially switched off, as may aid in preventing a full-fledged standalone Stable Diffusion application from generating celebrity faces, pending new legislation that could render such functionality illegal.

Once again, however, such functionality could inevitably be decompiled and reversed by interested parties; however, the software producer could, in that eventuality, claim that this is effectively unsanctioned vandalism – so long as this kind of reverse engineering is not made excessively easy.

Features That Could Be Included

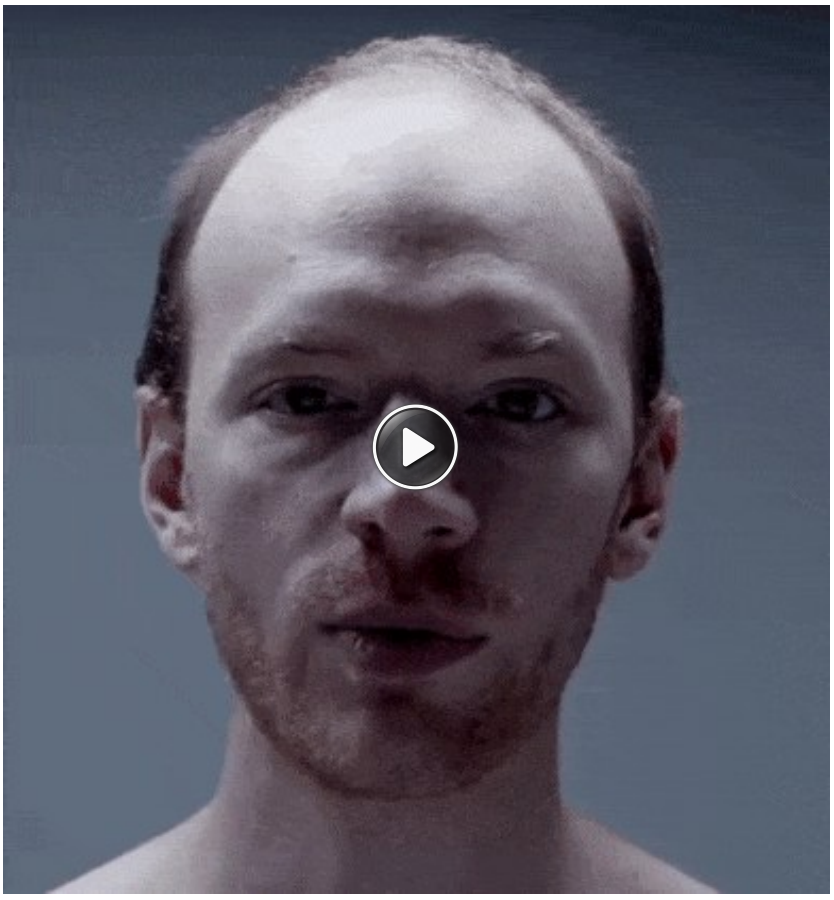
The core functionality in any distribution of Stable Diffusion would be expected of any well-funded commercial application. These include the ability to use text prompts to generate apposite images (text-to-image); the ability to use sketches or other pictures as guidelines for new generated images (image-to-image); the means to adjust how 'imaginative' the system is instructed to be; a way to trade off render time against quality; and other 'basics', such as optional automatic image/prompt archiving, and routine optional upscaling via RealESRGAN, and at least basic 'face fixing' with GFPGAN or CodeFormer.

That's a pretty 'vanilla install'. Let's take a look at some of the more advanced features currently being developed or extended, that could be incorporated into a full-fledged 'traditional' Stable Diffusion application.

Stochastic Freezing

Even if you reuse a seed from a previous successful render, it is terribly difficult to get Stable Diffusion to accurately repeat a transformation if *any part* of the prompt or the source image (or both) is changed for a subsequent render.

This is a problem if you want to use EbSynth to impose Stable Diffusion's transformations onto real video in a temporally coherent way – though the technique can be very effective for simple head-and-shoulders shots:



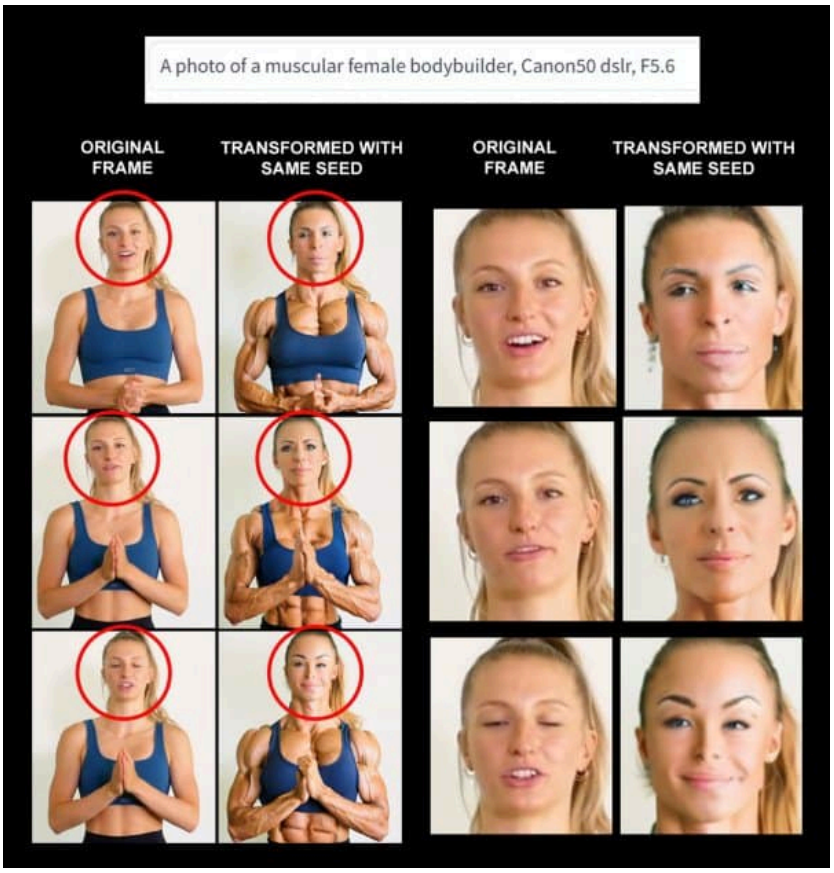
CLICK TO VIEW ANIMATED GIF

Limited movement can make EbSynth an effective medium to turn Stable Diffusion transformations into realistic video. Source: https://streamable.com/u0pgzd

EbSynth works by extrapolating a small selection of 'altered' keyframes into a video that has been rendered out into a series of image files (and which can later be reassembled back into a video).

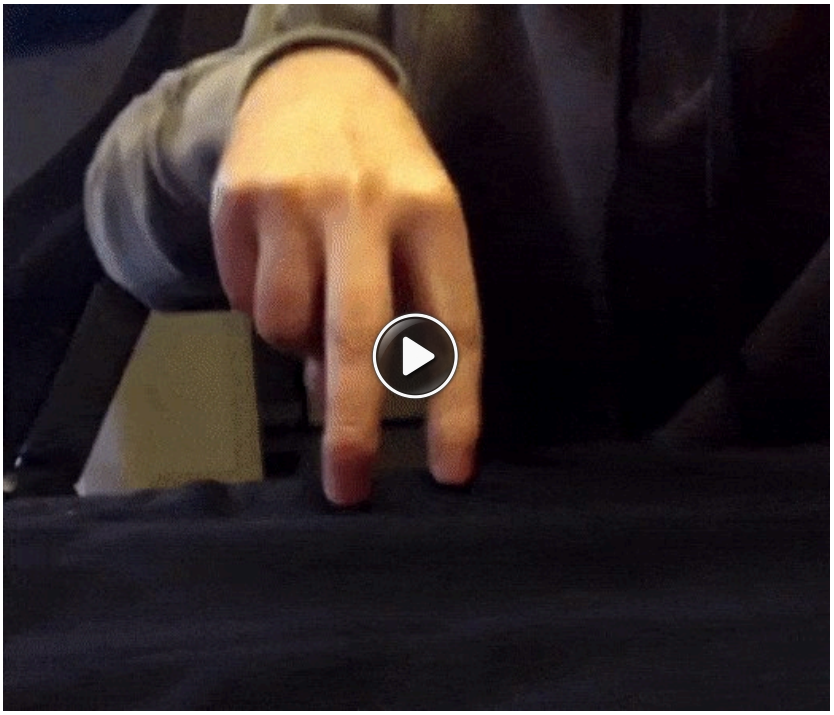
https://www.youtube.com/embed/eghGQtQhY38

In the example below, which features almost no movement at all from the (real) blonde yoga instructor on the left, Stable Diffusion still has difficulty maintaining a consistent face, because the three images being transformed as 'key frames' are not completely identical, even though they all share the same numeric seed.



Here, even with the same prompt and seed across all three transformations, and very few changes between the source frames, the body muscles vary in size and shape, but more importantly the face is inconsistent, hindering temporal consistency in a potential EbSynth render.

Though the SD/EbSynth video below is very inventive, where the user's fingers have been transformed into (respectively) a walking pair of trousered legs and a duck, the inconsistency of the trousers typify the problem that Stable Diffusion has in maintaining consistency across different keyframes, even when the source frames are similar to each other and the seed is consistent.



CLICK TO VIEW ANIMATED GIF

A man's fingers become a walking man and a duck, via Stable Diffusion and EbSynth. Source: https://old.reddit.com/r/StableDiffusion/comments/x92itm/proof_of_concept_using_img2img_ebsynth_to_animate/

The user who created this video commented that the duck transformation, arguably the more effective of the two, if less striking and original, required only a single transformed key-frame, whereas it was necessary to render 50 Stable Diffusion images in order to create the walking

trousers, which exhibit more temporal inconsistency. The user also noted that it took five attempts to achieve consistency for each of the 50 keyframes.

Therefore it would be a great benefit for a truly comprehensive Stable Diffusion application to provide functionality that preserves characteristics to the maximum extent across keyframes.

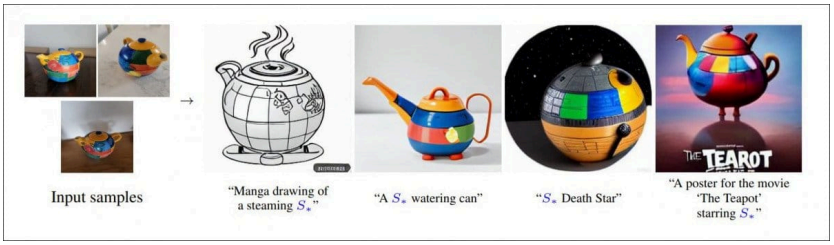
One possibility is for the application to allow the user to 'freeze' the stochastic encode for the transformation on each frame, which can currently only be achieved by modifying the source code manually. As the example below shows, this aids temporal consistency, though it certainly does not solve it:

CLICK TO VIEW ANIMATED GIF

One Reddit user transformed webcam footage of himself into different famous people by not just persisting the seed (which any implementation of Stable Diffusion can do), but by ensuring that the stochastic_encode() parameter was identical in each transformation. This was accomplished by modifying the code, but could easily become a user-accessible switch. Clearly, however, it does not solve all the temporal issues. Source: https://old.reddit.com/r/StableDiffusion/comments/wyeoqq/turning_img2img_into_vid2vid/

Cloud-Based Textual Inversion

A better solution for eliciting temporally consistent characters and objects is to 'bake' them into a Textual Inversion – a 5KB file that can be trained in a few hours based on just five annotated images, which can then be elicited by a special *'*'* prompt, enabling, for instance, a persistent appearance of novel characters for inclusion in a narrative.



Images associated with apposite tags can be converted into discrete entities via Textual Inversion, and summoned up without ambiguity, and in the correct context and style, by special token words. Source: https://huggingface.co/docs/diffusers/training/text_inversion

Textual Inversions are adjunct files to the very large and fully trained model that Stable Diffusion uses, and are effectively 'slipstreamed' into the eliciting/prompting process, so that they can participate in model-derived scenes, and benefit from the model's enormous database of knowledge about objects, styles, environments and interactions.

However, though a Textual Inversion does not take long to train, it does require a high amount of VRAM; according to various current walkthroughs, somewhere between 12, 20 and even 40GB.

Since most casual users are unlikely to have that kind of GPU heft at their disposal, cloud services are already emerging that will handle the operation, including a Hugging Face version. Though there are Google Colab implementations that can create textual inversions for Stable Diffusion, the requisite VRAM and time requirements may make these challenging for free-tier Colab users.

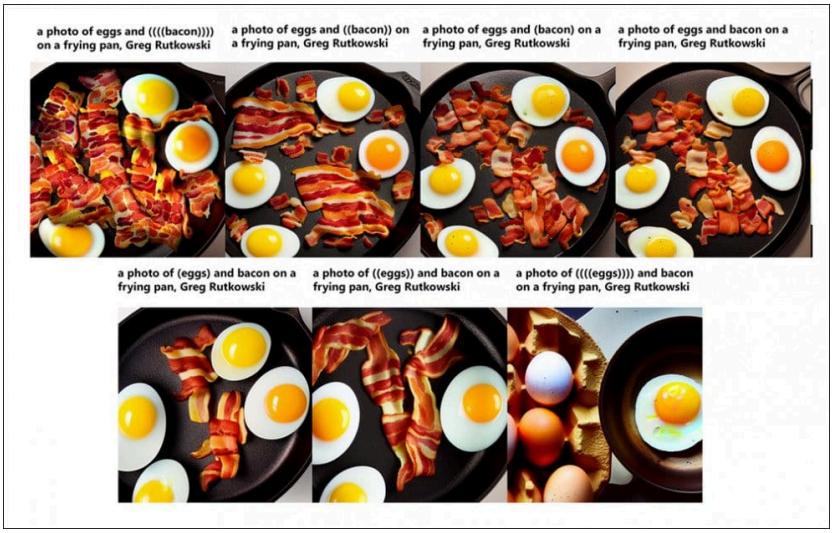
For a potential full-blown and well-invested Stable Diffusion (installed) application, passing this heavy task through to the company's cloud servers seems an obvious monetization strategy (assuming that a low or no-cost Stable Diffusion application is permeated with such non-free functionality, which seems likely in many possible applications that will emerge from this technology in the next 6-9 months).

Additionally, the rather complicated process of annotating and formatting the submitted images and text could benefit from automation in an integrated environment. The potential 'addictive factor' of creating unique elements that can explore and interact with the vast worlds of Stable Diffusion would seem potentially compulsive, both for general enthusiasts and younger users.

Versatile Prompt Weighting

There are many current implementations that allow the user to assign greater emphasis to a section of a long text prompt, but the instrumentality varies quite a lot between these, and is frequently clunky or unintuitive.

The very popular Stable Diffusion fork by AUTOMATIC1111, for instance, can lower or raise the value of a prompt word by enclosing it in single or multiple brackets (for de-emphasis) or square brackets for extra emphasis.



Square brackets and/or parentheses can transform your breakfast in this version of Stable Diffusion prompt weights, but it's a cholesterol nightmare either way.

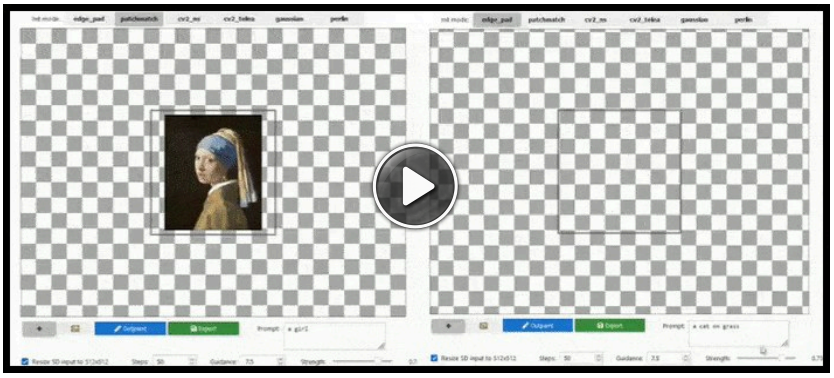
Other iterations of Stable Diffusion use exclamation marks for emphasis, while the most versatile allow users to assign weights to each word in the prompt through the GUI.

The system should also allow for negative prompt weights – not just for horror fans, but because there may be less alarming and more edifying mysteries in Stable Diffusion's latent space than our limited use of language can summon up.

Outpainting

Shortly after the sensational open-sourcing of Stable Diffusion, OpenAI tried – largely in vain – to recapture some of its DALL-E 2 thunder by announcing 'outpainting', which allows a user to extend an image beyond its boundaries with semantic logic and visual coherence.

Naturally, this has since been implemented in various forms for Stable Diffusion, as well as in Krita, and should certainly be included in a comprehensive, Photoshop-style version of Stable Diffusion.



CLICK TO VIEW ANIMATED GIF

Tile-based augmentation can extend a standard 512×512 render almost infinitely, so long as the prompts, existing image and semantic logic allow for it. Source: https://github.com/lkwq007/stablediffusion-infinity

Because Stable Diffusion is trained on 512x512px images (and for a variety of other reasons), it frequently cuts the heads (or other essential body parts) off of human subjects, even where the prompt clearly indicated 'head emphasis', etc..

Typical examples of Stable Diffusion 'decapitation'; but outpainting could put George back in the picture.

Any outpainting implementation of the type illustrated in the animated image above (which is based exclusively on Unix libraries, but should be capable of being replicated on Windows) should also be tooled as a one-click/prompt remedy for this.

Currently, a number of users extend the canvas of 'decapitated' depictions upwards, roughly fill the head area in, and use img2img to complete the botched render.

Effective Masking That Understands Context

Masking can be a terribly hit-and-miss affair in Stable Diffusion, depending on the fork or version in question. Frequently, where it's possible to draw a cohesive mask at all, the specified area ends up getting in-painted with content that does not take the entire context of the picture into account.

On one occasion, I masked out the corneas of a face image, and provided the prompt *'blue eyes'* as a mask inpaint – only to find that I appeared to be looking through two cut-out human eyes at a distant picture of an unearthly-looking wolf. I guess I'm lucky it wasn't Frank Sinatra.

Semantic editing is also possible by identifying the noise that constructed the image in the first place, which allows the user to address specific structural elements in a render without interfering with the rest of the image:

Changing one element in an image without traditional masking and without altering adjacent content, by identifying the noise that first originated the picture and addressing the parts of it that contributed to the target area. Source: https://old.reddit.com/r/StableDiffusion/comments/xboy90/a_better_way_of_doing_img2img_by_finding_the/

This method is based on the K-Diffusion sampler.

Semantic Filters for Physiological Goofs

As we've mentioned before, Stable Diffusion can frequently add or subtract limbs, largely due to data issues and shortcomings in the annotations that accompany the images that trained it.

Just like that errant kid who stuck his tongue out in the school group photo, Stable Diffusion's biological atrocities are not always immediately obvious, and you might have Instagrammed your latest AI masterpiece before you notice the extra hands or melted limbs.

It is so difficult to fix these kinds of errors that it would be useful if a full-size Stable Diffusion application contained some kind of anatomical recognition system that employed semantic segmentation to calculate whether the incoming picture features severe anatomical deficiencies (as in the image above), and discards it in favor of a new render before presenting it to the user.

Of course, you might want to render the goddess Kali, or Doctor Octopus, or even rescue an unaffected portion of a limb-afflicted picture, so this feature should be an optional toggle.

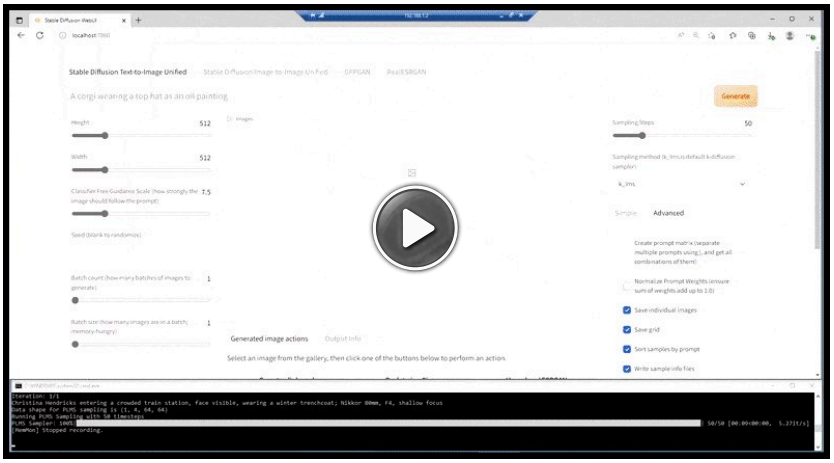
If users could tolerate the telemetry aspect, such misfires could even be transmitted anonymously in a collective effort of federative learning that may help future models to improve their understanding of anatomical logic.

LAION-Based Automatic Face Enhancement

As I noted in my previous look at three things Stable Diffusion could address in the future, it should not be left solely to any version of GFPGAN to attempt to 'improve' rendered faces in first-instance renders.

GFPGAN's 'improvements' are terribly generic, frequently undermine the identity of the individual depicted, and operate solely on a face that has usually been rendered poorly, as it has received no more processing time or attention than any other part of the picture.

Therefore a professional-standard program for Stable Diffusion should be able to recognize a face (with a standard and relatively lightweight library such as YOLO), apply the full weight of available GPU power to re-rendering it, and either blend the ameliorated face into the original full-context render, or else save it separately for manual re-composition. Currently, this is a fairly 'hands on' operation.

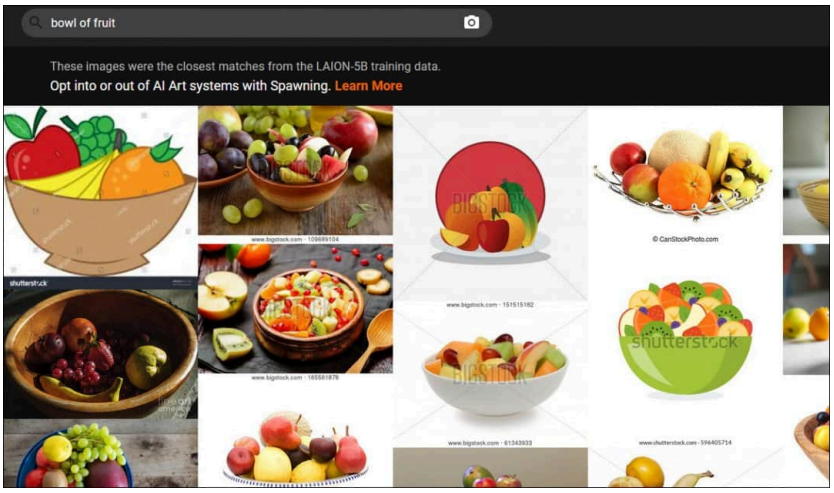


CLICK TO VIEW ANIMATED GIF

In cases where Stable Diffusion has been trained on an adequate number of images of a celebrity, it's possible to focus the entire GPU capacity on a subsequent render solely of the face of the rendered image, which is usually a notable improvement – and, unlike GFPGAN, draws on information from LAION-trained data, rather than simply adjusting the rendered pixels.

In-App LAION Searches

Since users began to realize that searching LAION's database for concepts, people and themes could prove an aide to better use of Stable Diffusion, several online LAION explorers have been created, including haveibeentrained.com.



The search function at haveibeentrained.com lets users explore the images that power Stable Diffusion, and discover whether objects, people or ideas that they might like to elicit from the system are likely to have been trained into it. Such systems are also useful to discover adjacent entities, such as the way celebrities are clustered, or the 'next idea' that leads on from the current one. Source: https://haveibeentrained.com/?search_text=bowl%20of%20fruit

Though such web-based databases often reveal some of the tags that accompany the images, the process of generalization that takes place during model training means that it is unlikely that any particular image could be summoned up by using its tag as a prompt.

Additionally, the removal of 'stop words' and the practice of stemming and lemmatization in Natural Language Processing means that many of the phrases on display were split up or omitted before being trained into Stable Diffusion.

Nonetheless, the way that aesthetic groupings bind together in these interfaces can teach the end user a lot about the logic (or, arguably, the 'personality') of Stable Diffusion, and prove an aide to better image production.

Conclusion

There are many other features that I'd like to see in a full native desktop implementation of Stable Diffusion, such as native CLIP-based image analysis, which reverses the standard Stable Diffusion process and allows the user to elicit phrases and words that the system would naturally associate with the source image, or the render.

Additionally, true tile-based scaling would be a welcome addition, since ESRGAN is almost as blunt an instrument as GFPGAN. Thankfully, plans to integrate the txt2imghd implementation of GOBIG are rapidly making this a reality across the distributions, and it seems an obvious choice for a desktop iteration.

Some other popular requests from the Discord communities interest me less, such as integrated prompt dictionaries and applicable lists of artists and styles, though an in-app notebook or customizable lexicon of phrases would seem a logical addition.

Likewise, the current limitations of human-centric animation in Stable Diffusion, though kick-started by CogVideo and various other projects, remains incredibly nascent, and at the mercy of upstream research into temporal priors relating to authentic human movement.

For now, Stable Diffusion video is strictly psychedelic, though it may have a much brighter near-future in deepfake puppetry, via EbSynth and other relatively nascent text-to-video initiatives (and it's worth noting the lack of synthesized or 'altered' people in Runway's latest promotional video).

Another valuable functionality would be transparent Photoshop pass-through, long since established in Cinema4D's texture editor, among other similar implementations. With this, one can shunt images between applications easily and use each application to perform the transformations that it excels at.

Finally, and perhaps most importantly, a full desktop Stable Diffusion program should be able not only to swap easily between checkpoints (i.e. versions of the underlying model that powers the system), but should also be able to update custom-made Textual Inversions that worked with previous official model releases, but may otherwise be broken by later versions of the model (as developers at the official Discord have indicated could be the case).

Ironically, the organization in the very best position to create such a powerful and integrated matrix of tools for Stable Diffusion, Adobe, has allied itself so strongly to the Content Authenticity Initiative that it might seem a retrograde PR misstep for the company – unless it were to hobble Stable Diffusion's generative powers as thoroughly as OpenAI has done with DALL-E 2, and position it instead as a natural evolution of its considerable holdings in stock photography.

First published 15th September 2022.