

Intégration de ChatGPT à Stable Diffusion localement

Fonctionnement d'une IA (Intelligence Artificielle)

Déjà avant toute chose, je vais vous expliquer le principe même d'une IA (Intelligence Artificielle) afin de ne pas vous perdre dans la suite de mes explications.

Une IA est un programme informatique qui peut effectuer des tâches qui nécessitent normalement l'intelligence humaine, telles que la reconnaissance de la parole, la vision, la prise de décision et la résolution de problèmes.

Le fonctionnement d'une IA est basé sur l'apprentissage automatique (machine learning). Le but de l'apprentissage automatique est de permettre à l'IA d'apprendre à partir de données, plutôt que de programmer explicitement toutes les règles et les décisions. En d'autres termes, l'IA apprend à partir d'exemples, comme un enfant apprendrait à identifier un chat ou un chien en voyant des photos.

Les données d'apprentissage sont introduites dans un modèle d'apprentissage automatique, qui est un ensemble d'algorithmes mathématiques qui apprennent à partir de ces données. Le modèle d'apprentissage automatique ajuste ensuite ses paramètres pour améliorer la précision de ses prédictions et de ses décisions.

C'est ce qu'on appelle l'entraînement du modèle. Une fois que le modèle est entraîné, il peut être utilisé pour prendre des décisions ou effectuer des tâches en temps réel, en utilisant de nouvelles données en entrée.

Par exemple, un modèle d'IA entraîné à jouer à un jeu vidéo pourrait être utilisé pour jouer contre des humains en temps réel.

Prenons un autre exemple avec celui de la reconnaissance d'images. Supposons que nous voulons construire une IA qui peut reconnaître les images de chats et les images de chiens. Pour ce faire, nous allons utiliser un modèle d'apprentissage automatique. Nous allons lui donner beaucoup d'exemples d'images de chats et d'images de chiens, pour qu'il puisse apprendre à les différencier. Le modèle d'apprentissage automatique va commencer par examiner chaque image et identifier les caractéristiques importantes, telles que la forme des oreilles, la forme du visage, la couleur des poils, etc.

Ensuite, le modèle va utiliser ces caractéristiques pour prendre une décision : est-ce une image de chat ou une image de chien ?

Au début, le modèle va souvent se tromper. Mais au fur et à mesure que nous lui donnons plus d'exemples et qu'il apprend à partir de ses erreurs, il deviendra de plus en plus précis.

Une fois que le modèle est suffisamment précis, nous pouvons l'utiliser pour identifier les images de chats et de chiens dans de nouvelles images. Nous donnons simplement au modèle une nouvelle image, et il nous dit si c'est une image de chat ou une image de chien.

Description de StableDiffusion

Bien, maintenant que vous avez une meilleure approche sur les IA, je vais vous expliquer le fonctionnement de StableDiffusion qui est une IA de génération d'image à partir de texte.

Je vais tenter de vous expliquer son fonctionnement le plus brièvement possible. Stable Diffusion est une intelligence artificielle dite de text to image, c'est-à-dire de génération d'image à partir d'un texte que l'on appelle un prompt.

Cette IA propose d'autres services que nous aborderons que très peu tel que l'img2img qui est une technique qui permet de transformer une image en une autre image ayant un aspect différent, tout en conservant le contenu général de l'image originale. Par exemple, on peut transformer une image de jour en une image de nuit, une image d'un style artistique en un autre style artistique, ou encore transformer une image de basse résolution en une image de haute résolution.

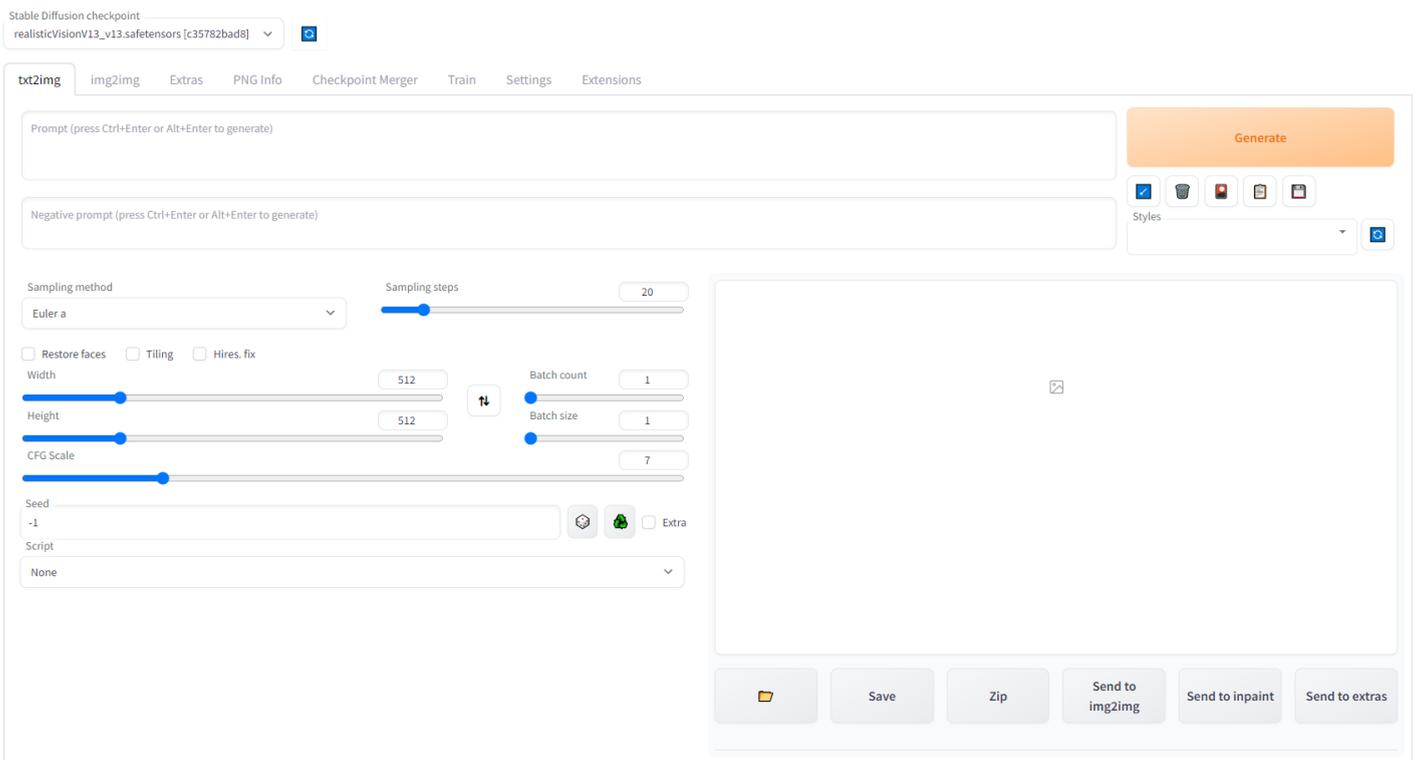
A l'instar de certaines IA, Stable Diffusion est totalement libre de source, c'est-à-dire que tout le monde, à condition d'avoir la configuration matériel requise, peut l'installer localement et le modifier à sa guise.

Objectif du projet StableDiffusion

Le but de ma première approche avec cette IA était de pouvoir l'installer localement ce que j'ai réussi à faire. Dans un premier temps je l'ai installé sans interface graphique, c'est-à-dire que je rentrais mes prompts directement dans l'invite de commande (il permet d'entrer des commandes afin de réaliser des tâches de base avec interface graphique telle que l'exécution d'un logiciel par exemple). Le résultat de ce prompt apparaissait dans un répertoire dédié pour.

Puis satisfait du résultat, j'ai commencé à chercher une manière plus jolie, plus esthétique, pour cela, j'ai opté pour une stable diffusion Automatic1111 qui intègre tout le code à l'installation sans n'avoir à exécuter quasiment aucune commande supplémentaire et qui intègre une interface graphique.

Voici une brève vue de son interface sans rentrer dans les sous-menus :

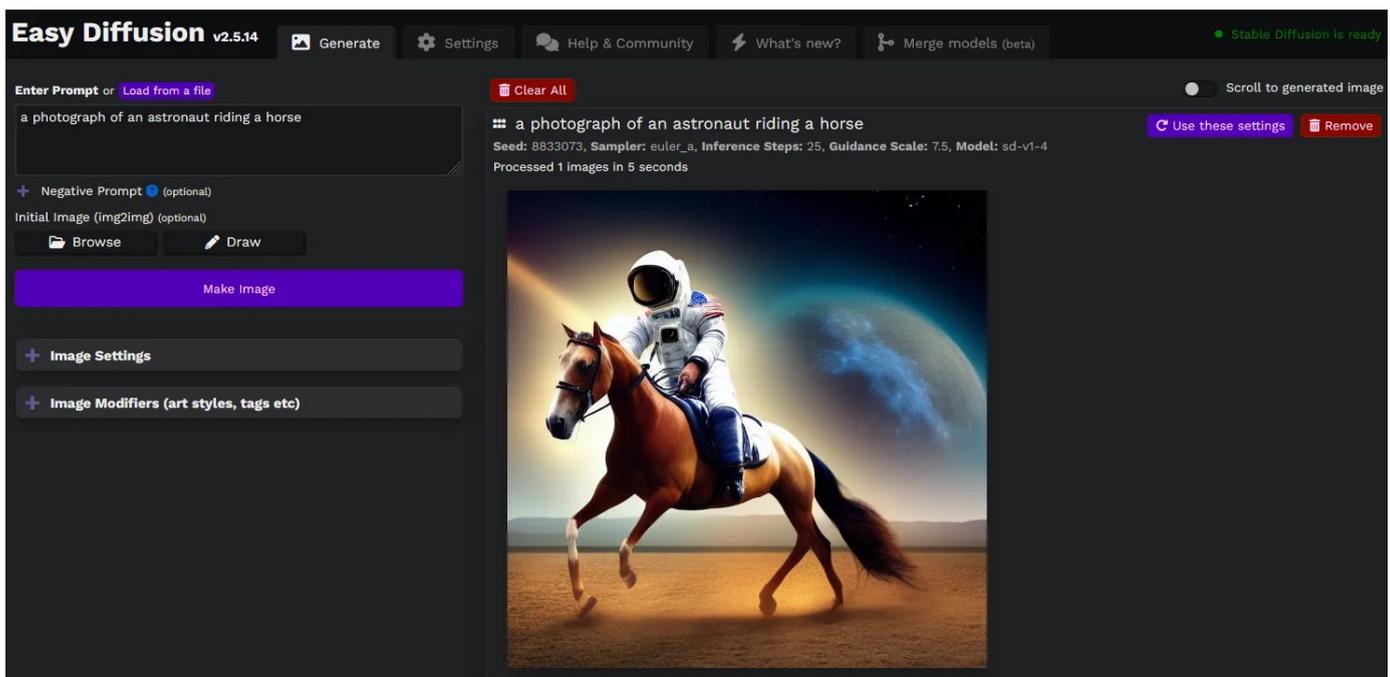


Cela n'est pas très compréhensible, mais je vais vous expliquer plus en détail dans la suite de ce rapport, sur une autre version.

Donc, là, nous avons la version 1 (V1) de stable diffusion, celle sortie en août 2022. Cette version nous intéresse moins aujourd'hui, car la version 2 (V2) est sortie tout récemment, cette dernière est bien plus complète (hormis quelques petites exceptions).

Je ne vais pas me pencher longtemps sur la version 1 car je ne m'en suis pas servi pour réaliser le sujet de ce thème technique, mais je voulais quand même l'aborder afin d'avoir une première approche.

À présent, nous allons voir la version 2 de Stable Diffusion, celle qui nous intéresse donc.



Pour l'installer, je suis allé sur le GitHub du projet puis téléchargé tous les fichiers.

Notre objectif est d'intégrer ChatGPT (une IA de génération de texte) sur cette interface de StableDiffusion, donc de rajouter une case dans laquelle nous entrons un mot-clé afin qu'un prompt détaillé, généré par ChatGPT, ressorte pour générer l'image.

Le principal point fort de notre démarche est qu'elle n'as que très peu voir pas du tout était abordé et que par conséquent aucune aide à part nos propres compétences et internet ne pourront être utilisés.

Mise en place du projet

Dans un premier temps, il m'a fallu analyser le code fourni dans le projet afin de pouvoir le modifier à ma guise. Car oui, par défaut, le code n'est pas modifiable et dès lors que nous le modifions, il se met à jour automatiquement afin de retrouver sa version de base.

Voici tout le répertoire de stable diffusion 2 afin de le lancer en local, comme vous pourrez le remarquer, il se trouve le dossier "models" (le fichier modèle est en fait la base de données dans laquelle l'IA va puiser ses ressources afin de générer l'image), dans le reste se trouve des fichiers pour installer l'IA, notamment dans le dossier scripts où nous allons retrouver la source du problème.

Dans ce dossier, se trouve deux fichiers .bat (sous fichiers qui permettent de lancer l'application), ces derniers sont "on_env_start.bat" et "on_sd_start.bat". Le problème est que lors de l'exécution du fichier "Start Stable Diffusion UI.cmd"(fichier principal à exécuter afin de lancer l'application), le script qui s'exécute, fait appel au fichier "on_env_start.bat", celui-ci fait appel au fichier "on_sd_start.bat", ces derniers effectuent des copies de fichiers, ce qui écrase les fichiers enregistrés avec notre nouvelle configuration, il suffit donc de supprimer ces lignes et l'IA est totalement modifiable à notre guise, cela inclut l'interface graphique ce qui nous intéresse le plus.

Nom	Modifié	Type	taille
installer_files	24/02/2023 11:36	Dossier de fichiers	
models	24/02/2023 11:46	Dossier de fichiers	
plugins	24/02/2023 11:46	Dossier de fichiers	
profile	24/02/2023 11:46	Dossier de fichiers	
scripts	24/02/2023 14:48	Dossier de fichiers	
sd-ui-files	24/02/2023 11:40	Dossier de fichiers	
stable-diffusion	24/02/2023 11:50	Dossier de fichiers	
tmp	24/02/2023 16:10	Dossier de fichiers	
ui	24/02/2023 16:03	Dossier de fichiers	
.tmp1	24/02/2023 16:10	Fichier TMP1	1 Ko
.tmp2	24/02/2023 16:10	Fichier TMP2	0 Ko
CreativeML Open RAIL-M License	06/02/2023 19:25	Fichier	15 Ko
Developer Console.cmd	24/02/2023 11:40	Script de comman...	2 Ko
How to install and run.txt	06/02/2023 19:25	Document texte	1 Ko
LICENSE	06/02/2023 19:25	Fichier	5 Ko
Start Stable Diffusion UI.cmd	24/02/2023 16:09	Script de comman...	1 Ko

Voici les lignes à supprimer afin de pouvoir avoir accès à toute l'IA :

```
@xcopy sd-ui-files\ui ui /s /i /Y /q
@copy sd-ui-files\scripts\on_sd_start.bat scripts\ /Y
@copy sd-ui-files\scripts\bootstrap.bat scripts\ /Y
@copy sd-ui-files\scripts\check_modules.py scripts\ /Y
@copy "sd-ui-files\scripts\Start Stable Diffusion UI.cmd" . /Y
@copy "sd-ui-files\scripts\Developer Console.cmd" . /Y
```

```
@copy sd-ui-files\scripts\on_env_start.bat scripts\ /Y
@copy sd-ui-files\scripts\bootstrap.bat scripts\ /Y
@copy sd-ui-files\scripts\check_modules.py scripts\ /Y
```

```
1 @echo off
2
3 cd /d %~dp0
4 set PATH=C:\Windows\System32;%PATH%
5
6 @rem set legacy installer's PATH, if it exists
7 if exist "installer" set PATH=%cd%\installer;%cd%\installer\Library\bin;%cd%\installer\Scripts;%cd%\installer\Library\usr\bin;%PATH%
8
9 @rem Setup the packages required for the installer
10 call scripts\bootstrap.bat
11
12 @rem set new installer's PATH, if it downloaded any packages
13 if exist "installer_files\env" set PATH=%cd%\installer_files\env;%cd%\installer_files\env\Library\bin;%cd%\installer_files\env\Scripts;%cd%\instal
14
15 set PYTHONPATH=%cd%\installer;%cd%\installer_files\env
16
17 @rem Test the bootstrap
18 call where git
19 call git --version
20
21 call where conda
22 call conda --version
23
24 @rem Download the rest of the installer and UI
25 call scripts\on_env_start.bat
26
27 @pause
```

Bien, maintenant que nous avons accès total à l'IA, nous allons pouvoir commencer nos expérimentations.

Pour commencer, j'ai testé de générer une image avec cette nouvelle version, sauf qu'il s'avérerait que cette nouvelle version soit plus restrictive que la première, en faisant quelques recherches, j'en ai conclu que cela venait du modèle (la base de données téléchargée en local), qui avait été mis à jour pour ne plus afficher le visage des personnalités connus pour semble-t-il éviter les deep fake, et même sans parler de personnalité, les images générées sont clairement moins jolie que l'ancienne version.

J'ai donc téléchargé d'autres modèles qui ne sont pas natifs à l'IA, c'est-à-dire des modèles réalisés par d'autres utilisateurs.

Après plusieurs tests, j'en ai trouvé deux qui paraissent plutôt bien faits.

Donc j'ai opté pour le modèle "Realistic Vision V1.3" qui est un modèle permettant de réaliser des images réalistes et intégrant les personnalités connues.

Et "mdjrny-v4" style qui est un préfixe à utiliser dans son prompt, il est tout droit inspiré de l'IA Midjourney qui est aussi une IA de génération d'images de base payante au bout d'un certain nombre d'utilisation et généralement non-utilisable en local, à la différence ce modèle est gratuit et utilisable en local.

Realistic Vision V1.3

rihanna wearing a blue jacket (dimension : 512 / 512) :



rihanna wearing a blue jacket (dimension : 768 / 768) :



mdjrny-v4

mdjrny-v4 style rihanna wearing a blue jacket (dimension : 512 / 512) :



mdjrny-v4 style rihanna wearing a blue jacket (dimension : 712 / 712) :



Pour commencer, faisons la comparaison entre les deux modèles, l'un est fait pour avoir un aspect réel et détaillé (Realistic Vision V1.3), l'autre a plus un effet cartoon (mdjrny-v4 style).

J'ai pris l'exemple d'une star, car souvent, c'est là où cela pêche et là où c'est censé pêcher sur la version de Stable Diffusion V2 alors que dans la V1 le problème ne se pose pas. En fait, si nous utilisons le modèle de base intégré à la V2 de Stable Diffusion, ce modèle est très restreint. L'avantage, là, c'est que je peux profiter de la nouvelle interface de la V2 avec en bonus la possibilité de pouvoir générer des images en 768 / 768 et générer des images impossibles de générer à la base en gardant le modèle intégré à la V2.

Vous remarquerez que pour le modèle de Midjourney, j'ai ajouté le préfixe "mdjrny-v4 style " devant mon prompt, c'est tout simplement pour éviter des

complications notamment la non prise en compte de certains éléments du prompt comme vous le voyez ci-dessous :

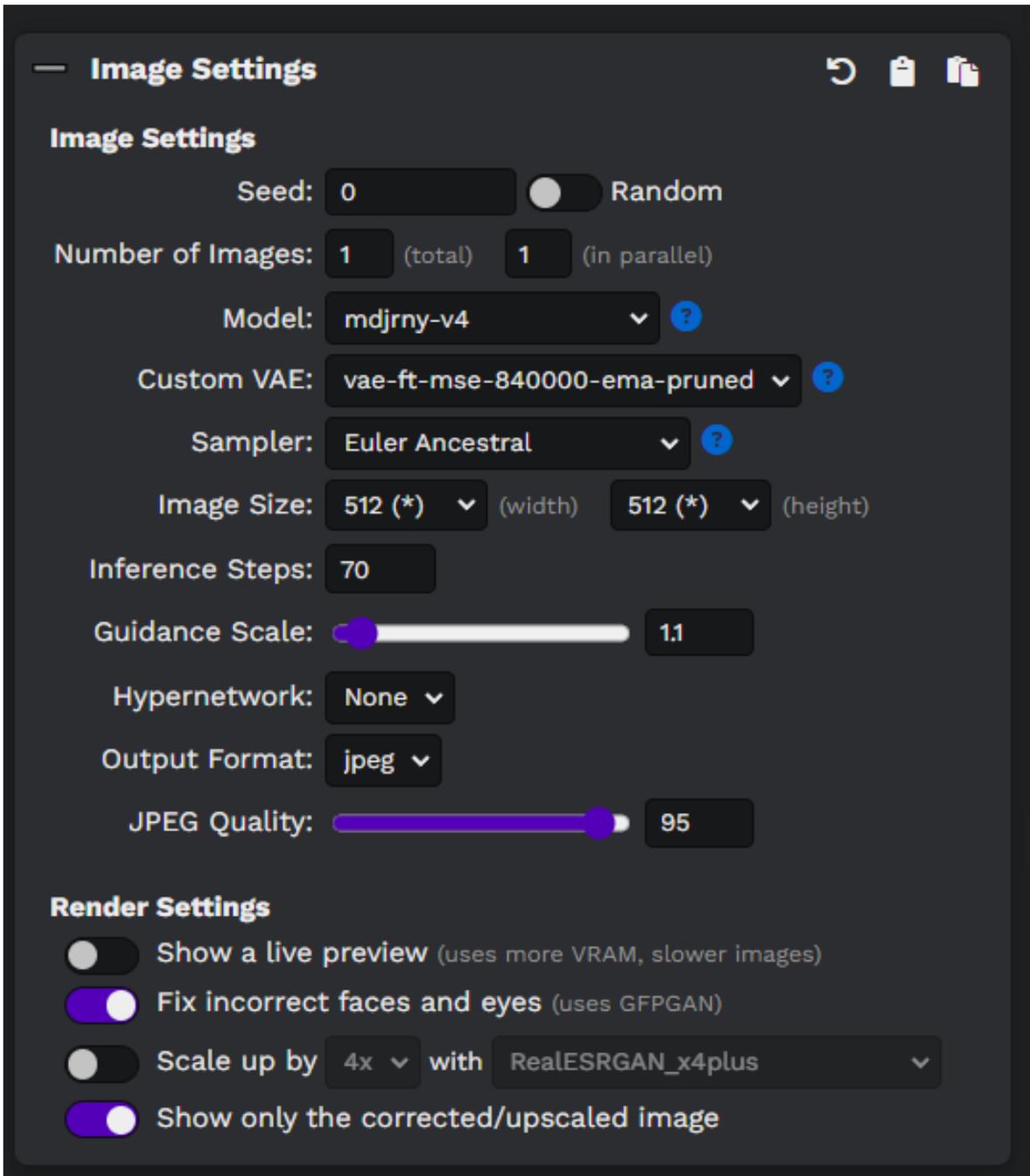
rihanna wearing a blue jacket :



Là où cela devient intéressant, c'est que je peux ajouter ce préfixe sur un prompt utilisant un autre modèle comme realistic :



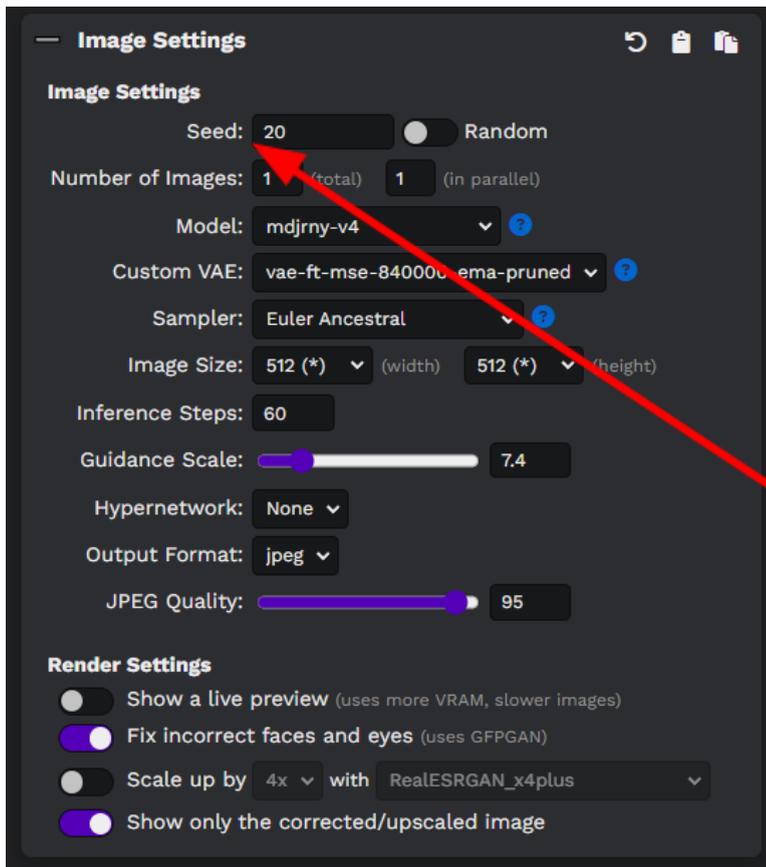
Pour réaliser ces images, j'ai utilisé ces paramètres :



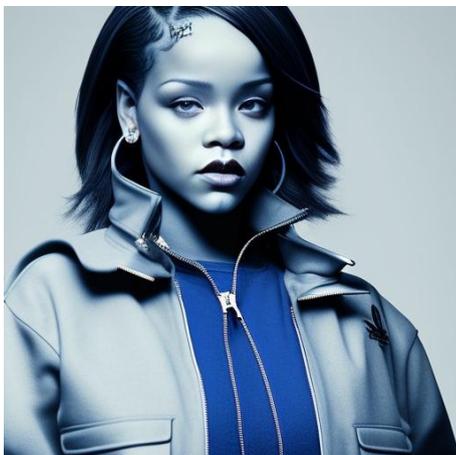
Je vais vous détailler leurs utilités.

Le Seed : il permet de donner une identité à l'image, par exemple si je donne le seed 20 à une image, bien que tous les paramètres soient les mêmes, l'image sera différente car elle n'aura pas la même identité.

Dans ce cas, j'ai toujours utilisé le même seed afin de comparer deux images avec le préfixe mdjrny-v4 style et sans, avec le même model. Mais par exemple, en gardant le même prompt et les mêmes paramètres, nous pouvons voir que le résultat est différent en changeant le seed :



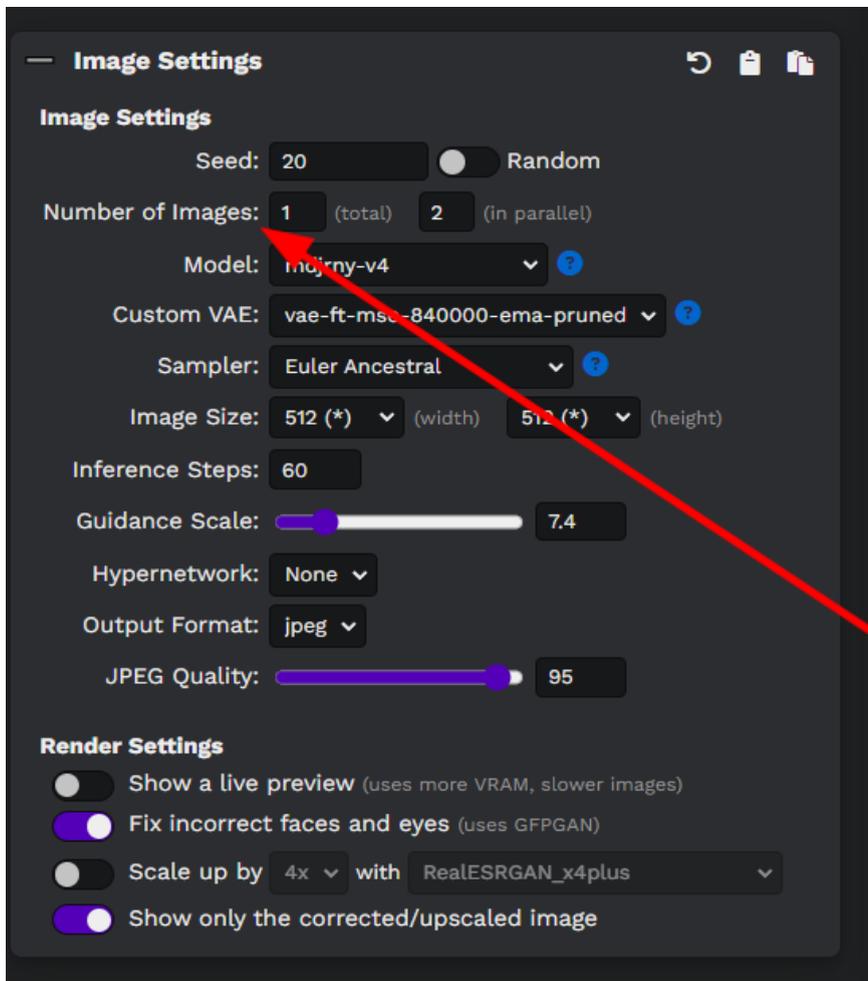
Seed généré en 20 en 512 / 512



Seed 20 généré en 768 / 768



Number of Images : il permet de générer plusieurs images simultanément avec plusieurs variations pour le même seed



Model : L'interface utilisateur de stable diffusion utilise ce qu'on appelle des modèles pour créer les images. Les modèles sont formés à l'aide de nombreuses images et descriptions d'images. Lors de l'installation, un modèle par défaut est téléchargé, le

modèle sd-v1-4. Comme vu juste avant, d'autres modèles existent. Certains d'entre eux utilisent sd-v1-4 comme base et ont ensuite été formés sur des images supplémentaires, tandis que d'autres modèles ont été formés à partir de zéro.

Custom VAE : En bref, il améliore les images générées. Un VAE est formé pour certains aspects de l'image, le VAE par défaut fourni (vae-ft-mse-840000-ema-pruned) améliore les yeux dans les images générées.

Sampler : Un sampler et une méthode d'échantillonnage qui permet de corriger certaines imperfections, il existe beaucoup de samplers différents, mais j'ai décidé d'utiliser Euler Ancestral, car après comparaison, c'est celui qui génère les meilleures images.

Image Size : C'est tout simplement la taille de l'image. En général, une image plus grande nécessitera plus de ressources et de temps pour être générée, mais elle peut également contenir plus de détails et être plus facilement reconnaissable par l'utilisateur final. À l'inverse, une image plus petite nécessitera moins de ressources et de temps, mais elle peut également manquer de détails et être moins reconnaissable.

Inference Steps : L'inférence steps (ou étapes d'inférence) dans un modèle d'IA de "text to image" se réfère au processus de génération d'images à partir de texte. Voici une explication détaillée de chacune des étapes courantes de l'inférence :

1. Pre-traitement du texte : cette étape implique le nettoyage et la préparation du texte d'entrée avant qu'il ne soit utilisé pour générer une image. Le texte peut être nettoyé en retirant les mots ou les caractères inutiles, les signes de ponctuation et les fautes d'orthographe. Le texte est également normalisé pour garantir une cohérence dans le formatage et la structure.
2. Encodage du texte : après le pré-traitement, le texte d'entrée est transformé en une représentation numérique qui peut être comprise par le modèle d'IA. Cette étape peut impliquer l'utilisation de techniques de traitement du langage naturel (NLP) pour extraire des informations sémantiques du texte, telles que des concepts clés ou des relations entre les mots.
3. Décodage de l'image : une fois que le texte est encodé, le modèle d'IA utilise une technique de décodage pour générer une image correspondante. Les techniques de décodage peuvent varier en fonction du modèle spécifique utilisé, mais elles impliquent généralement la génération de pixels pour chaque partie de l'image.
4. Post-traitement de l'image générée : après la génération de l'image, elle est soumise à une étape de post-traitement pour améliorer la qualité et la précision. Cette étape peut inclure des techniques telles que la suppression du bruit ou la correction des couleurs.

En somme, l'inférence steps est le processus de génération d'images à partir de texte dans un modèle d'IA de "text to image". Les étapes d'inférence sont essentielles pour garantir que le modèle puisse comprendre le texte d'entrée et générer une image correspondante qui est de haute qualité et précise.

Guidance Scale : le paramètre Guidance Scale contrôle le degré de précision avec lequel Stable Diffusion suivra l'invite lors de la génération d'images. Une valeur plus élevée obligera l'IA à être plus stricte et à suivre l'invite de près, tandis qu'une valeur plus faible lui donnera une plus grande liberté de création.

Hypernetwork : l'hypernetwork est chargé de garder en mémoire les images que le système a précédemment générées. Cela signifie que lorsque l'utilisateur fournit une nouvelle entrée, le système peut utiliser ses connaissances antérieures pour créer une image plus précise. Les hypernetworks permettent au système d'apprendre plus rapidement et de s'améliorer au fil du temps.

Output Image : C'est le format de sortie de l'image, en l'occurrence jpeg, mais il existe aussi png par exemple.

JPEG Quality : C'est la qualité qu'aura l'image jpeg en sortie.

Render Settings : Il est composé de plusieurs paramètres :

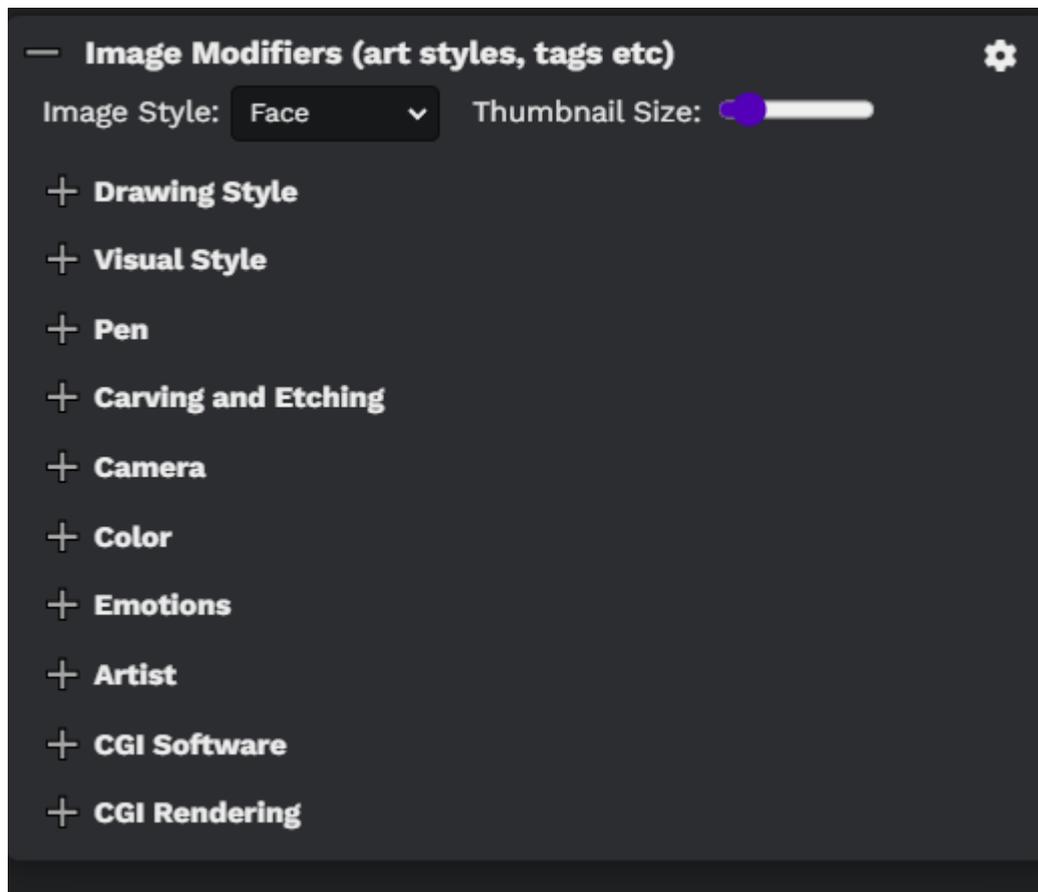
Show a live preview, il permet d'avoir un aperçu en direct lors de la génération de l'image, Fix incorrect faces and eyes, il permet de corriger les imperfections sur le visage et les yeux,

Scale up by, il permet de générer l'image en 2 ou 4 fois plus grand sans perte de qualité, je ne l'ai pas activé dans mon exemple, mais ce paramètre est vraiment bien.

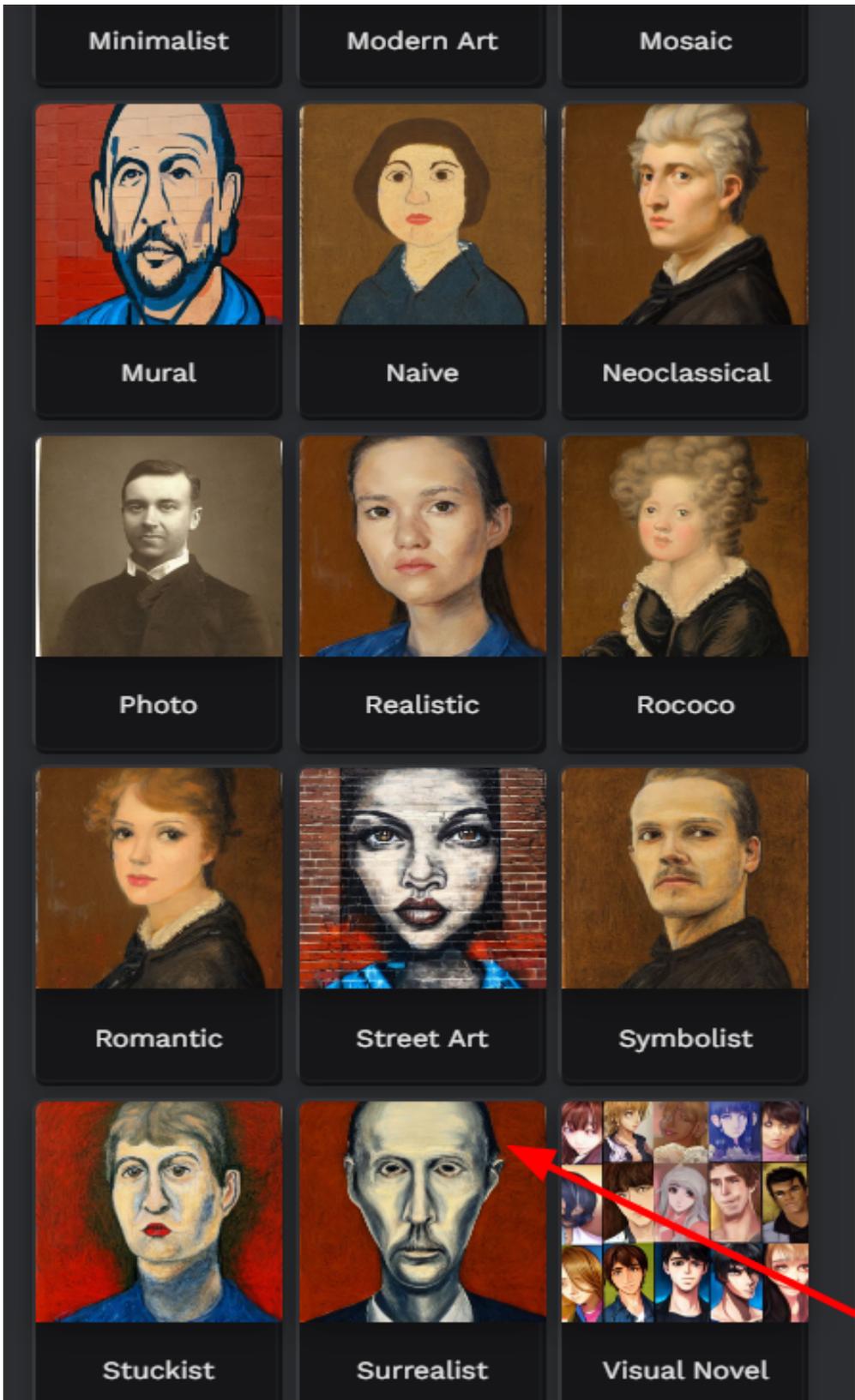
Nous avons vu les paramètres de l'image, maintenant nous allons voir les paramètres de style qui est une des nouveautés de la V2 de StableDiffusion.

Essaie des différents paramètres de style intégré à la V2 de Stable Diffusion, test avec le même seed afin de voir l'infinité de variations possibles, test de plusieurs styles en même temps avec résultat positif.

Voilà comment se composent les paramètres pour modifier le style de l'image :



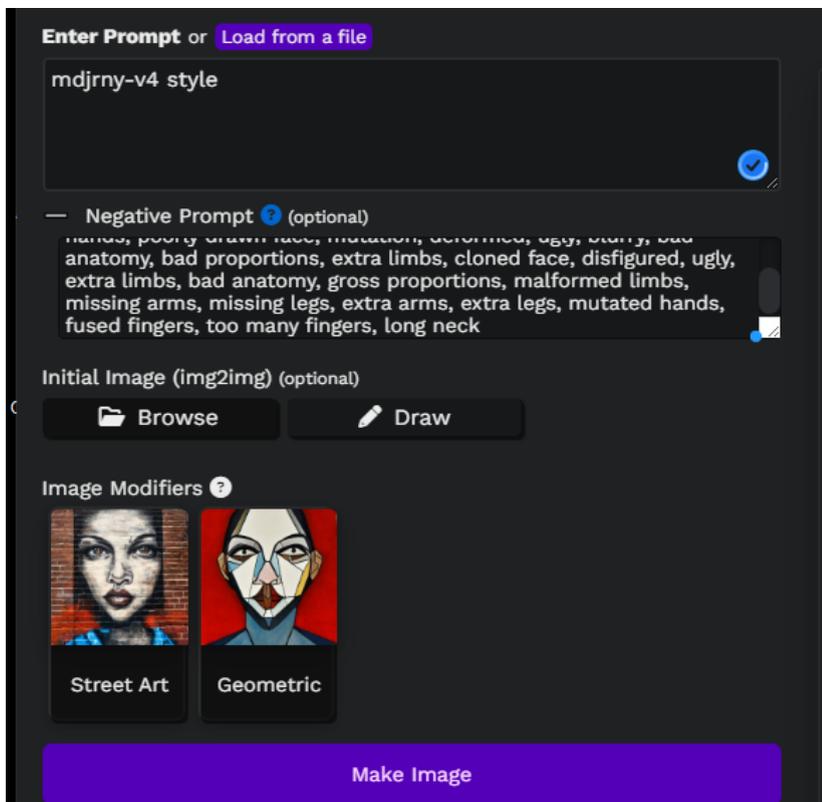
Dans cet exemple, je vais utiliser le paramètre Visual Style dans lequel nous pouvons voir beaucoup de styles, dans cet exemple nous allons utilisé en premier lieu le style Street Art :



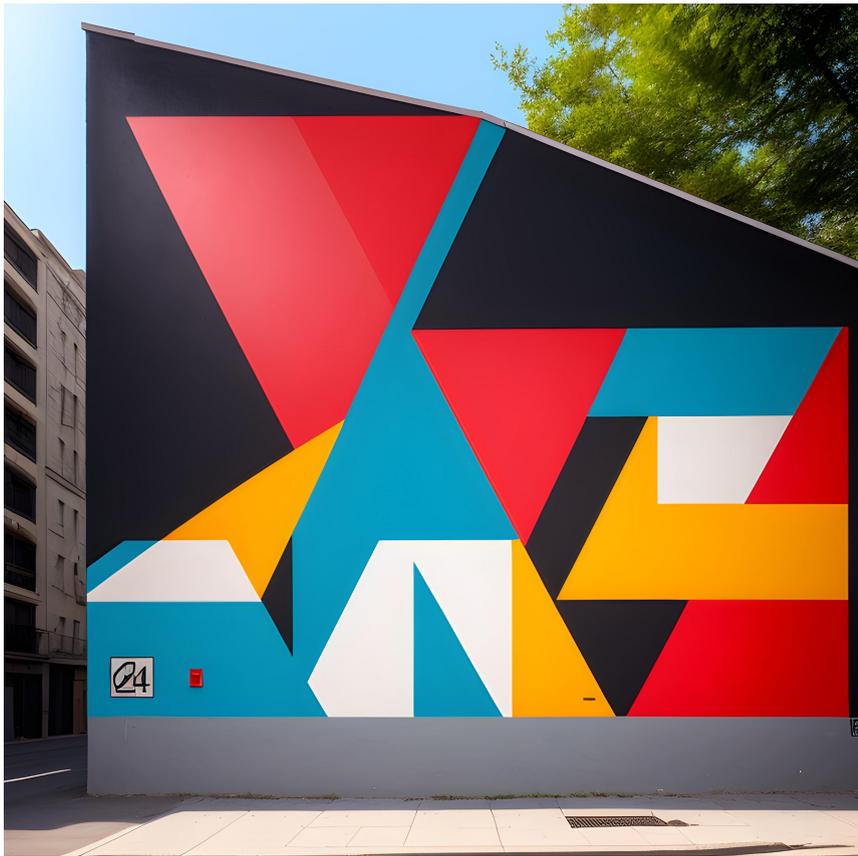
Résultat :



Puis j'ai rajouté le style Geometric en plus du style Street Art :



Résultat :



Comme vous pouvez le constater, les deux styles se sont mélangés ce qui donne ce résultat plutôt abstrait. Le but est de rendre chaque image unique et inédite, nous pouvons facilement imaginer que tout type d'art peut être recopié à la perfection et amélioré, par exemple, il est possible d'utiliser le style de Van Gogh ou de Picasso et généré des œuvres totalement inédites. Il y a un réel risque pour les métiers de l'art, mais également pour plein d'autre métier, d'ailleurs vu que nous y sommes, j'ai demandé à ChatGPT de me citer 5 métiers susceptibles d'être impacté par les IA dans l'avenir :

1. Opérateurs de centres d'appels et de services clients
2. Employés de la banque et du secteur financier
3. Opérateurs de saisie de données et de transcription
4. Traducteurs et interprètes
5. Métier de l'art

Je précise que l'IA m'a spécifié le fait que les métiers de l'art ne pourront pas être totalement remplacés, car la touche humaine est très importante et l'émotion de l'œuvre également, cependant l'IA pourra servir comme inspiration ou pour automatiser des tâches numérique (dans le cas de retouche photo par exemple). Bien sur cette liste est à prendre avec des pincettes.

Je pourrais rajouter encore plein de style en plus, mais cela s'éloignerait de plus en plus de la requête de base, le but est juste de montrer que des styles assez poussés peuvent être ajoutés à notre résultat pour créer des variations uniques.

En parallèle, j'ai demandé à ChatGPT de me réaliser des prompts, très détaillés pour voir jusqu'où stable diffusion peut aller.

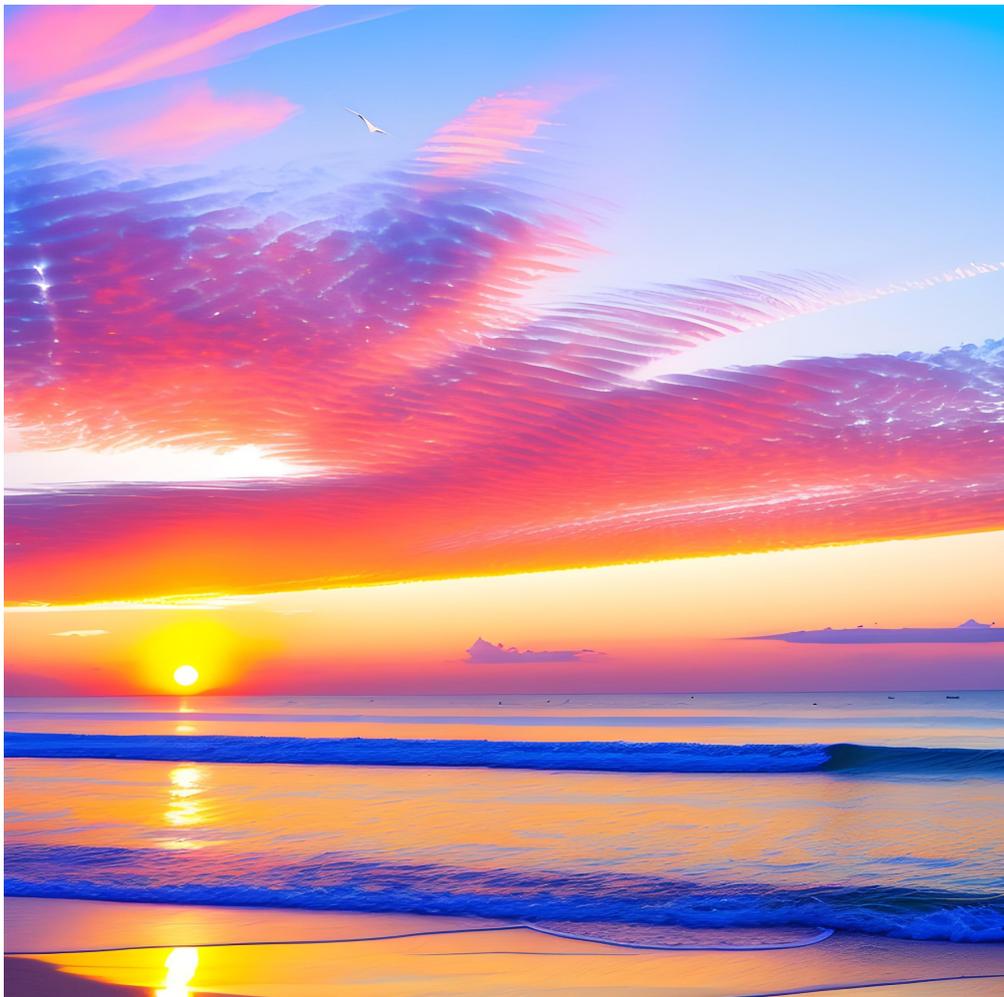
Ces premiers tests m'ont permis de me faire une idée des résultats qui pourront être générés grâce aux prompts réalisés par ChatGPT.

Voici des exemple de prompts et leurs résultat :

mdjrny-v4 style Imagine a breathtaking sunset on a tropical island, where the sky is a fiery orange and pink. The palm trees sway gently in the breeze, and the sand is a soft, golden white. You can see the crystal clear water of the ocean, with small waves lapping against the shore. A flock of seagulls fly overhead, their wings reflecting the warm light of the setting sun. Your task is to capture the beauty and tranquility of this tropical paradise in your painting

Prompt négatif (les éléments à éviter) :

Imagine a dark, foreboding forest where the trees are twisted and gnarled, casting deep shadows on the forest floor. The sky above is a deep, murky green, and the air is thick with the scent of decay. You can hear the distant howl of wolves and the rustling of underbrush as some unseen creature moves through the forest. Your task is to capture the sense of fear and danger that permeates this haunted forest. Avoid using any bright or cheerful colors in your painting.



mdjrny-v4 style Picture a vibrant and colorful parrot, with feathers in shades of brilliant blues, greens, and yellows. The bird is perched on a branch, its head turned slightly to the side, as if listening to something in the distance. You can see the

intricate details of its feathers, with each individual strand and color variation captured in vivid detail. In the background, you can see the lush foliage of a tropical rainforest, with hints of sunlight filtering through the leaves. Your task is to create a portrait of this magnificent bird that captures its beauty and unique personality, using a rich and vivid color palette.



Satisfaits de ces résultats, nous allons maintenant voir comment intégrer ChatGPT à notre projet.

Pour commencer, j'ai d'abord tenté d'installer ChatGPT localement, cela fut un succès.

Pour le moment, il se lance dans un localhost (accès au serveur local afin d'avoir un aperçu graphique de l'application) différent de celui de Stable Diffusion, mais ses fichiers se trouvent dans le même projet.

Pour l'installation de chatgpt en local, j'ai suivi une vidéo YouTube qui explique le pas-à-pas à adopter et tout le fonctionnement des modèles de chatgpt.

Comme Stable Diffusion, afin que ChatGPT puisse récupérer ses données, il faut avoir un modèle. Chaque modèle à sa spécificité. Certains modèles sont plus dédiés à la traduction, certains a de la correction de code, certains a de la génération de code à partir d'un texte, certains pour du question answering (réponses aux questions sous forme de dialogue)

Voici tous les modèles existant et leurs spécialités :

text-davinci-003 : Le modèle text-davinci-003 est une version améliorée du modèle GPT-3 de OpenAI. Il est conçu pour effectuer des tâches de suivi d'instructions plus complexes que les modèles GPT-3 standard, tels que davinci, curie, babbage, qui sont destinés à être utilisés avec le point de terminaison de complétion de texte. Selon OpenAI, text-davinci-003 produit une écriture de qualité supérieure et peut traiter des instructions plus complexes, ce qui permet aux développeurs d'applications d'être encore plus créatifs avec les capacités du modèle. Bien que davinci et text-davinci-003 soient tous deux des modèles puissants, ils diffèrent sur certains points clés. text-davinci-003 est la version plus récente et plus capable, conçue spécifiquement pour les tâches de suivi d'instructions.

text-curie-001 : Le modèle text-curie-001 est un modèle de traitement de langage naturel (NLP pour Natural Language Processing, c'est un type de modèle d'apprentissage automatique conçu pour traiter et comprendre le langage naturel, c'est-à-dire le langage utilisé par les êtres humains dans leur communication quotidienne) développé par OpenAI. Il est connu pour sa puissance et sa rapidité, et il est capable de traiter des tâches NLP complexes telles que l'analyse de texte et la classification de sentiment. En comparaison avec le modèle Davinci qui est considéré comme le plus performant, le modèle text-curie-001 est moins puissant mais reste capable de traiter des tâches de manière très efficace, comme la classification de sentiment et le résumé de texte.

En somme, le modèle text-curie-001 est un modèle NLP performant et rapide qui peut traiter de manière efficace des tâches telles que la classification de sentiment et le résumé de texte.

text-babbage-001 : Le modèle text-babbage-001 est un modèle de traitement du langage naturel (NLP) développé par OpenAI et disponible sur la plateforme Azure

Cognitive Services. Ce modèle est capable d'effectuer des tâches simples de classification ainsi que de la recherche sémantique pour déterminer à quel point les documents correspondent aux requêtes de recherche. Bien que ce modèle soit moins performant que le modèle Davinci en termes de précision, il est plus rapide, ce qui peut être un avantage pour certaines applications. En général, il est recommandé aux utilisateurs de commencer avec le modèle Davinci tout en expérimentant, car il produit les meilleurs résultats et valide la valeur qu'Azure OpenAI peut offrir.

Code-davinci-002 : Le modèle Codex (nom complet: GPT-3 Codex) est un modèle de langage génératif créé par la société OpenAI. Le modèle a été entraîné sur une large variété de tâches de traitement de langage naturel à grande échelle, notamment la traduction automatique, la génération de texte, la réponse à des questions, etc. Le modèle Codex est particulièrement remarquable car il a été entraîné sur des exemples de code informatique, en plus du langage naturel. Cela signifie qu'il peut être utilisé pour générer automatiquement des lignes de code informatique à partir de descriptions textuelles en langage naturel. Cette fonctionnalité est très utile pour les développeurs de logiciels qui cherchent à accélérer leur processus de développement. Le modèle Codex est également connu sous le nom de "Davinci-002", car il fait partie de la série de modèles GPT-3 de OpenAI, et il est considéré comme la version améliorée de son prédécesseur, GPT-3. Il a été lancé en 2021 et est disponible via une API publique, mais l'accès est soumis à certaines restrictions et nécessite une autorisation de la part de OpenAI.

Dans mon cas, j'ai utilisé le modèle text-davinci-003, car il est très complet et que mon application n'est pas très volumineuse donc n'est pas gourmande et ne risque pas de ramer lors de son utilisation.

Une fois ChatGPT installé localement, voici à quoi cela ressemble :

localhost:3000

A brown, fluffy, bushy-tailed cat crouches atop a white, wooden chest of drawers, eyes wide with anticipation and wary concentration. Its thin tail, wrapped horizontally around its body, is raised slightly off the ground as the cat focuses intently on a small, blue feather that dangles just above its nose, suspended from a thin thread of black yarn. With its pink, whiskered snout and soft features, the cat looks ready to spring into action at a moment's notice, ready to pounce on the feather and claim it as its own at any second.

Nous avons dans cet exemple l'input (l'entrée) dans lequel est écrit "cat", le bouton pour rechercher et le résultat qui s'affiche en bas.

Le résultat de ce prompt :



À noter : l'utilisation de ce modèle est payante, je n'ai pas vraiment trouvé d'alternative à ce problème, j'ai donc pris la décision de faire comme ça jusqu'à qu'il y est une alternative.

Détail du prix en fonction du modèle :

- **Babbage**
 - \$0.0005 / 1K tokens
- **Curie**
 - \$0.0020 / 1K tokens
- **Davinci**
 - Most powerful
 - \$0.0200 / 1K tokens

Un token correspond à environ 4 caractères ou 0,75 mots.

Par exemple, le prompt généré en dessus faisait 118 tokens et 535 caractères, pour le savoir, j'ai utilisé un service de openai permettant d'estimer leur nombre.

Afin que le projet marche, il m'a été utile de bien comprendre ce que propose openai et comprendre tous les paramètres à ajuster tout comme stable diffusion, je dirais que la plus grosse partie du travail se trouve ici, sans cela, cela reviendrait à foncer tête baissée sans chercher à comprendre toute la techno utilisée pour en arriver là et tôt ou tard être bloqué par ce manque de connaissances.

Je vous mets ici la partie du code qui nous intéresse :

```
const response = await openai.createCompletion({
  "model": "text-davinci-003",
  "prompt": `Pretend you are a prompt text to image
writer, write the best prompt, write directly only the result,
write an extremely detailed result, ${message}`,
  "max_tokens": 1000,
  "temperature": 1,
});
```

Le code est amplement plus grand et s'étend sur plusieurs fichiers, mais nous voyons ici la partie pour paramétrer le résultat.

Décortiquons :

```
const response = await openai.createCompletion({
```

Le code utilise la bibliothèque openai pour accéder à l'API GPT-3 d'OpenAI, qui permet de générer du texte en utilisant des modèles de langage de pointe. La méthode createCompletion() de cette bibliothèque envoie une requête à l'API GPT-3. La réponse de cette méthode est stockée dans une variable response qui contient le texte généré par l'API. Le mot-clé await est utilisé pour attendre que la méthode createCompletion() soit terminée avant de continuer à exécuter le code. D'autres paramètres peuvent être utilisés pour spécifier le modèle de langage à utiliser, le nombre de réponses à générer, ou encore le degré de créativité souhaité.

```
"model": "text-davinci-003",
```

Cela permet de définir sur quel modèle souhaitons nous nous baser.

```
"prompt": `Pretend you are a prompt text to image writer,
write the best prompt, write directly only the result, write
an extremely detailed result, ${message}`,
```

'Pretend you are a prompt text to image writer, write the best prompt, write directly only the result, write an extremely detailed result', est un texte qui précède le prompt. Il s'agit d'une demande adressée à l'API (ChatGPT) pour générer un texte qui répond à cette demande.

`{message}` est une expression JavaScript qui insère la valeur de la variable `message` à l'intérieur de la chaîne de caractères.

La variable 'message' contient les mots-clés que l'utilisateur a saisis dans l'interface. En combinant ces deux parties, la chaîne de caractères prompt demande à l'API GPT-3 de générer un texte qui répond à la demande, ce texte est le prompt en réponse au mot clé.

Dans cette partie on dit à l'IA ce qu'on veut qu'elle fasse afin d'avoir le résultat attendu.

```
"max_tokens": 1000,
```

C'est le maximum de token que le résultat peut afficher, cela permet de limiter un trop grand résultat.

```
"temperature": 1,
```

Cela permet de créer des variations à la réponse afin que l'IA ne génère pas le même résultat pour un même mot clé et qu'à chaque fois qu'une requête est lancée avec le même mot clé, et bien que le résultat soit différent. A 0 le même résultat sera généré à chaque fois pour le même mot clé.

Bien, maintenant que nous savons qu'il est possible d'utiliser ChatGPT localement, il nous reste à l'intégrer à StableDiffusion.

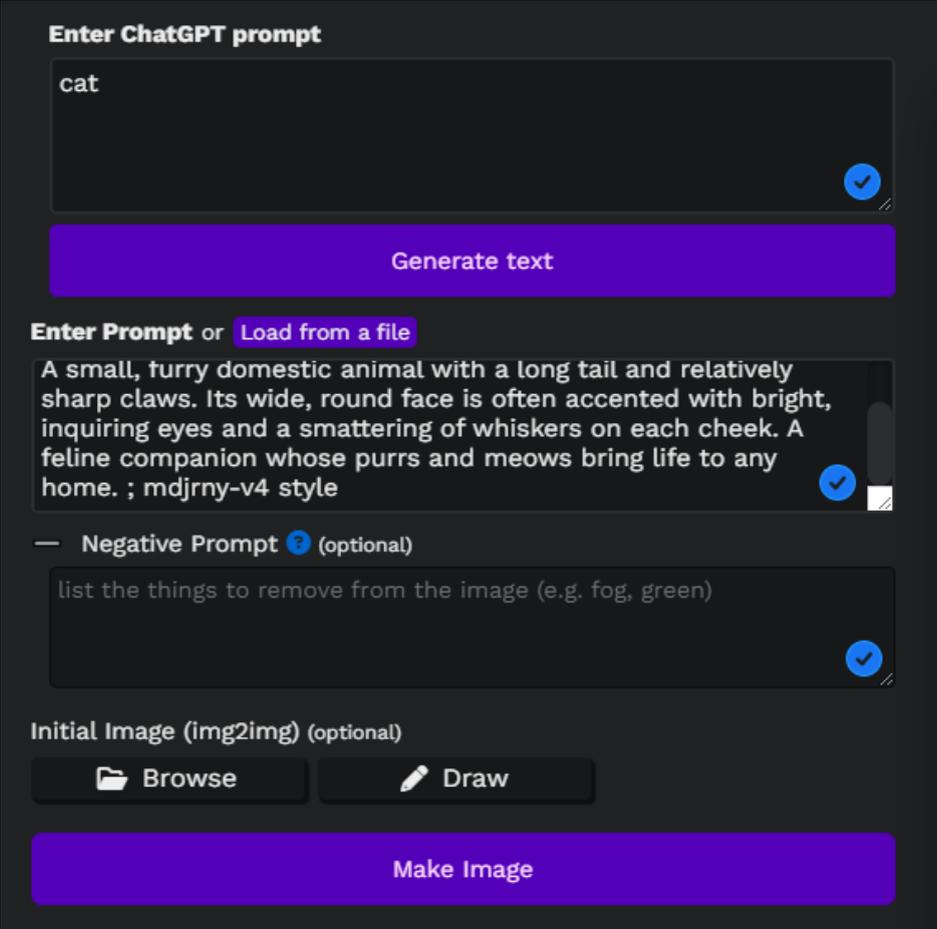
Je précise que cette partie ne faisait pas partie de mes connaissances, je me suis donc fait aider par l'alternant.

Nous avons réussi à intégrer ChatGPT au projet StableDiffusion.

En fait, dans notre code ChatGPT, il y avait deux parties : l'une graphique et l'une où la réponse s'affichait uniquement.

Donc cela fait que nous avons deux localhost différents, localhost:3000 et localhost:3001.

Dans la partie stable diffusion, il y a une partie où l'on peut entrer nos mots clés voir des prompts, on appuie sur un bouton qui envoie une requête au localhost:3001 (partie non graphique), la réponse se met dans la partie pour entrer le prompt pour générer l'image, la réponse est donc un prompt généré par ChatGPT. Comme vu précédemment, on a dit à ChatGPT de générer des prompts avec ce qu'on lui donne. La partie "Enter ChatGPT prompt" a été rajoutée dans le projet.

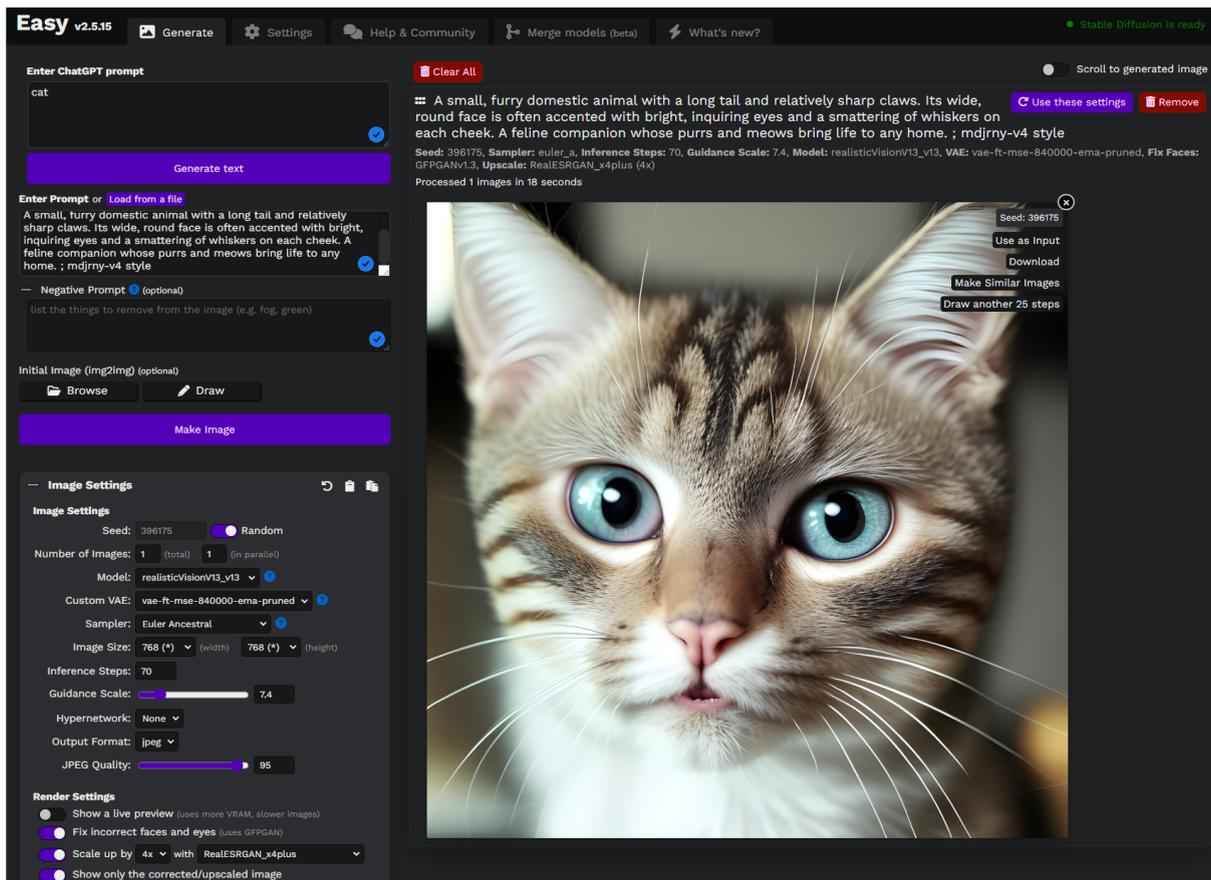


The image shows a dark-themed web interface for generating images. At the top, there is a section titled "Enter ChatGPT prompt" with a text input field containing the word "cat" and a blue checkmark icon. Below this is a purple button labeled "Generate text".

The next section is titled "Enter Prompt or Load from a file". It contains a text area with the following text: "A small, turry domestic animal with a long tail and relatively sharp claws. Its wide, round face is often accented with bright, inquiring eyes and a smattering of whiskers on each cheek. A feline companion whose purrs and meows bring life to any home. ; mdjrnv4 style". There is a blue checkmark icon to the right of the text area.

Below this is a section for "Negative Prompt (optional)" with a text input field containing the text "list the things to remove from the image (e.g. fog, green)". There is a blue checkmark icon to the right of the input field.

At the bottom, there is a section for "Initial Image (img2img) (optional)" with two buttons: "Browse" and "Draw". Below these buttons is a large purple button labeled "Make Image".



Afin que cette nouvelle partie fonctionne, il faut lancer la commande “node index.js” qui permet de lancer le fichier javascript qui exécute tous le code. Pour optimiser un maximum l’application, nous avons entré cette commande dans un fichier .bat (fichier exécutable), ce dernier se fait appelé par le fichier “Start Stable Diffusion UI.cmd” que nous avons vu précédemment, ce dernier permettant de lancer l’application en un clic en l’exécutant. Donc en cliquant sur “Start Stable Diffusion UI.cmd”, le fichier .bat contenant la commande “node index.js” s’exécute et permet de faire fonctionner l’option rajoutée.

Nous pourrions conclure sur cela mais j'aimerais vous montrer encore une options que StableDiffusion propose car mon but premier et de vous faire découvrir pleinement cet IA.

Je vais vous parler de l'option de img2mg de Stable Diffusion V2.

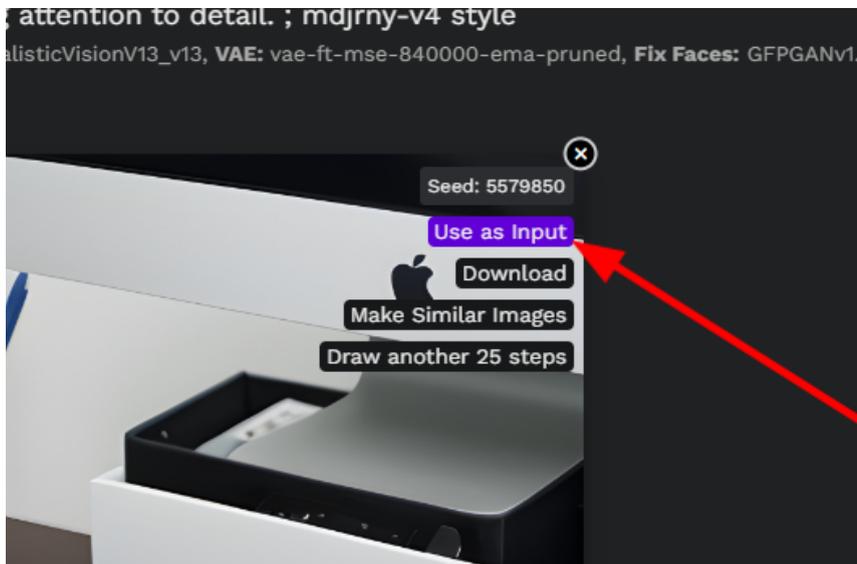
Le but de l'img2img est de modifier des éléments sur une image et de les supprimer ou de les remplacer.

Ce que j'ai fait, c'est que j'ai généré un prompt d'un bureau en marquant le mot "office" en mot clé, puis le prompt à été généré par ChatGPT.

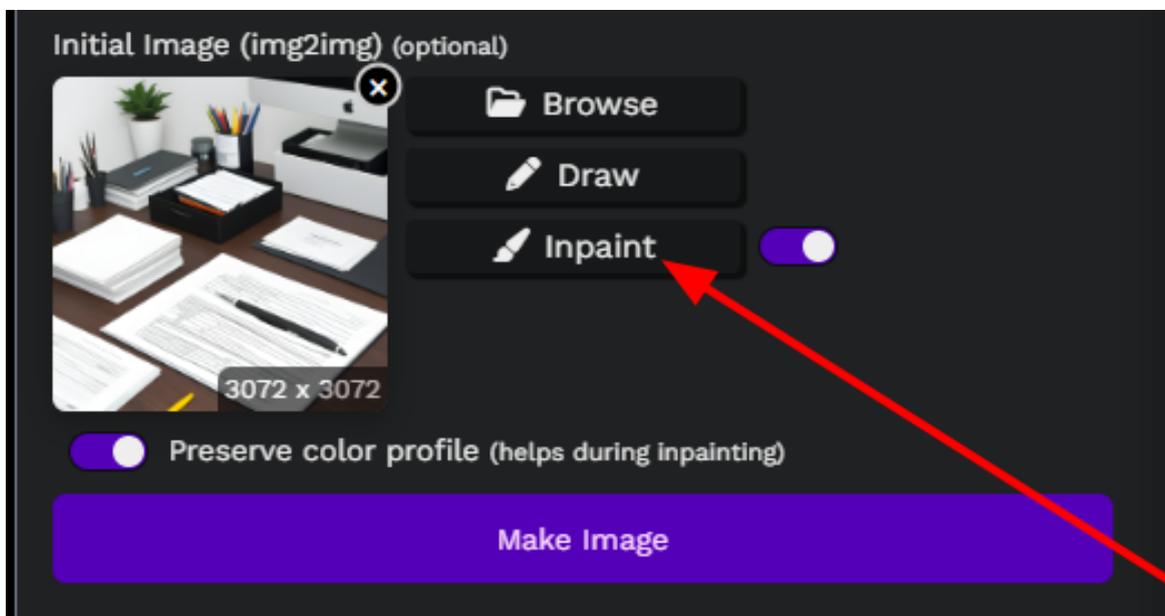
J'ai eu en résultat cette image :



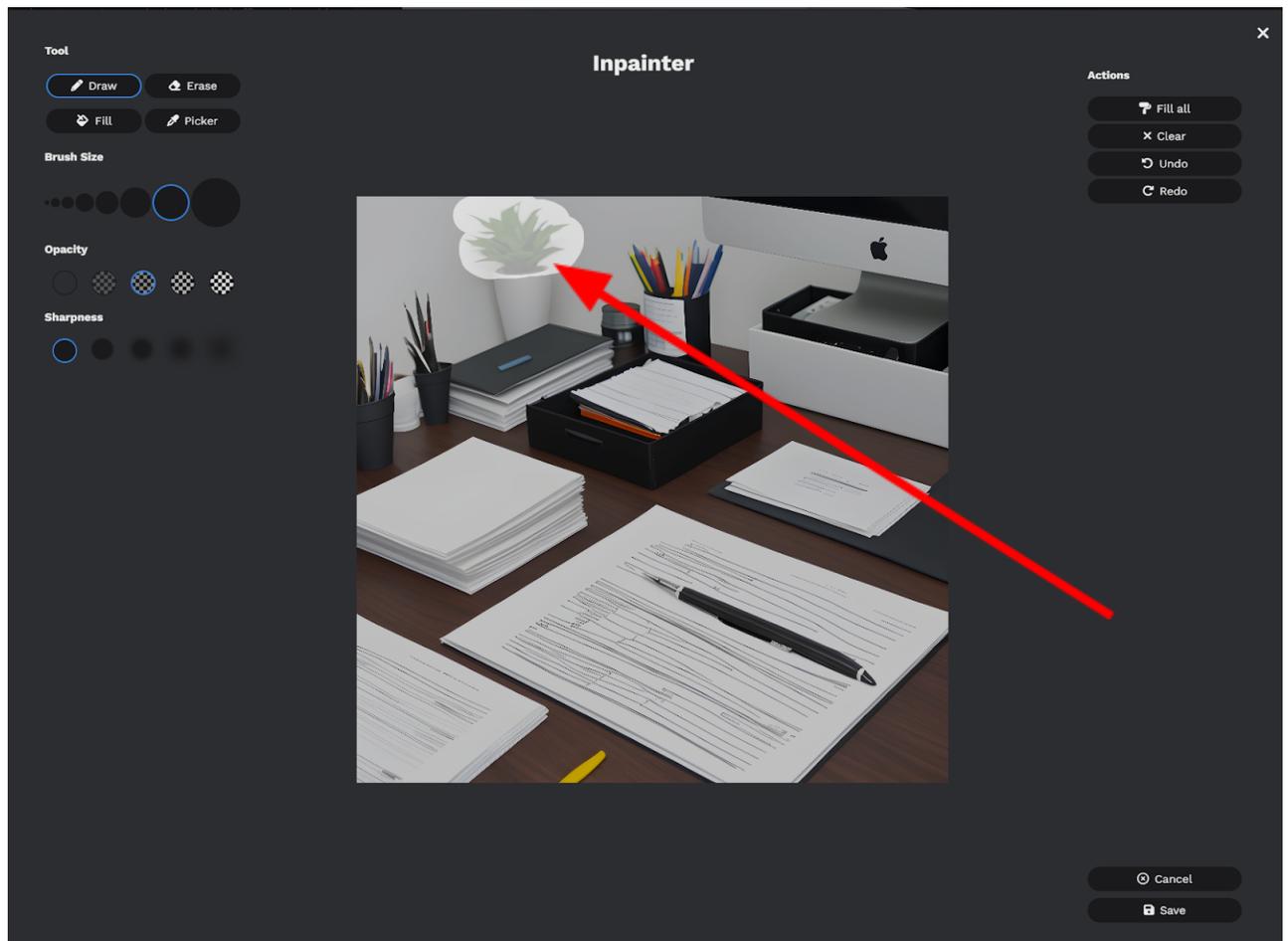
Ensuite j'ai voulu remplacer la plante dans le fond par un cactus, pour cela j'ai appuyé sur "Use as Input"



Puis j'ai cliquer sur "Inpaint"



J'ai dessiné l'endroit ou je voulais que mon cactus apparaisse :



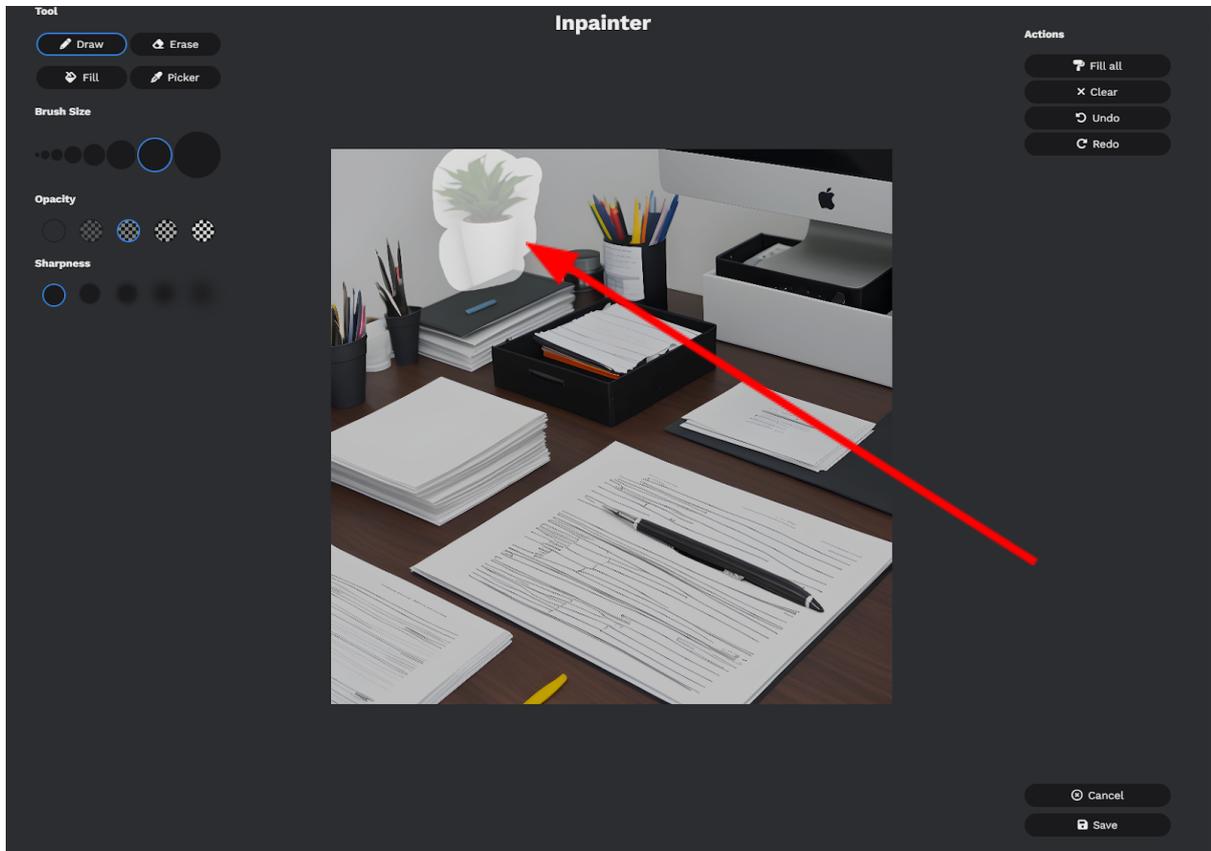
Maintenant il me resté à écrire "cactus" au début de mon prompt que j'ai séparé d'une virgule pour le distinguer du reste du prompt :

Cactus, An organised, methodical office worker with a tidy desk and organising stacks of paper and documents. Focused on completing tasks efficiently and on time, while paying attention to detail. ; mdjrny-v4 style

Résultat :



Mais on peut également supprimer l'élément de l'image, pour cela même principe sauf que cette fois ci au début de mon prompt, j'écris l'argument "No" ou "Remove". Donc là j'ai voulu supprimer la plante de mon image :



Dans mon prompt j'ai précisé que je ne voulais ni la plante ni le pot :

No plant, No pot, An organised, methodical office worker with a tidy desk and organising stacks of paper and documents. Focused on completing tasks efficiently and on time, while paying attention to detail. ; mdjrnv4 style

Résultat :



C'est plutôt bluffant, bien que beaucoup de logiciels de retouche photo permettent de faire cela, l'avantage de là c'est que cela m'a pris moins de 2 min en tout pour un résultat satisfaisant.

J'ai peaufiné le projet afin qu'on puisse le publier sur GitHub (rappelez vous c'est de là que le projet vient de base).

J'ai notamment modifier le fichier Readme qui permet à n'importe qui qui télécharge le projet d'accéder aux modifications apportées, j'ai expliqué la démarche, les modifications apportées et comment lancer l'application.

Voici son contenu :

```
This is a fork of "Easy Diffusion UI" without auto updates,
for manual customization purposes.

# How the app works

To launch this application, simply double-click on the "Start
Stable Diffusion UI.cmd" file to the root of the directory.
After you will be access to the app here :
http://localhost:9000/

# Changes

- Addition of the ChatGPT API to generate prompts with
keywords or little prompts.
- Addition of the green theme in the settings.

# ChatGPT

To create this, we used another server and connected it with
this app. When you click on "Generate Text" in the "Enter
ChatGPT Prompt" section, it sends a request to the server, and
the output of that request appears in the "Enter Prompt"
section. When the prompts are generated, the "; mdjrny-v4
style" prefix is added to the latter. This prefix serves to
correct any imperfections on the face, including eyes.

Disclaimer: To generate prompts with the ChatGPT API, you need
to enter your API key in the following parameters. It is
important to note that after a certain number of tokens, you
```

will be charged a fee. The amount depends on the model you used. There are different models with different functions and result capacities. You can find all of these models and their prices here: <https://openai.com/pricing#image-models>

In this app, we used the Davinci model, which is the most expensive but the best model for generating prompts. It is important to use a model that fits your needs and to inform yourself well beforehand in order to choose the most appropriate one.

You can change the model in the index.js file to the root of the directory.

For the Generate Text function to work, you must provide your API key, otherwise, if you click it, nothing happens.

To use your API key, please follow these steps :

Create an account on OpenAI: <https://platform.openai.com/>

Create a new API key:

<https://platform.openai.com/account/api-keys>

Delete the ".example" in your ".env.example" file and replace "ENTER_YOUR_API_KEY" by your API key in the file.

If you didn't provide the API key and you provide it afterwards, you have to launch the terminal and execute the command "node index.js" or restart the file Start Stable Diffusion UI.cmd .

Nous somme fin prêt à publier cette mise à jour de l'application.

A la fin de mon stage, j'ai réalisé une micro conférence sur les IA.

Voici le lien :

https://www.linkedin.com/posts/jonathan-chaffanjon-96683775_je-voudrais-profiter-d-e-cette-occasion-pour-activity-7044320308689854464-dm8k?utm_source=li_share&utm_content=feedcontent&utm_medium=g_dt_web&utm_campaign=copy