

NOCTELIAs Guide für Ai ART

Was ist Stable Diffusion?

Stable Diffusion ist eine Technik oder ein Algorithmus, der in der Bildverarbeitung und Computergrafik verwendet wird. Es handelt sich um einen fortgeschrittenen Prozess zur Generierung von hochwertigen Bildern, insbesondere bei der Verbesserung von Bildern mit geringer Qualität oder Rauschen.

Bei der stabilen Diffusion wird ein mathematisches Modell angewendet, um das Bild allmählich zu glätten und gleichzeitig wichtige Details beizubehalten. Dieser Prozess basiert auf der Diffusionsgleichung, die eine Ausbreitung von Informationen im Bild simuliert. Durch kontrollierte Diffusion werden Unregelmäßigkeiten und Rauschen verringert, während gleichzeitig die Bildstruktur und feine Details bewahrt werden.

Stable Diffusion bietet eine leistungsfähige Methode zur Verbesserung von Bildern, insbesondere in Bereichen wie der Bildrestaurierung, Rauschunterdrückung und der Erzeugung von hochwertigen Visualisierungen. Es ermöglicht eine präzisere Bildanalyse und -interpretation sowie die Schaffung ästhetisch ansprechender visueller Ergebnisse

Was ist Python?

Python ist eine hochrangige, interpretierte Programmiersprache, die sich durch ihre einfache Syntax und Lesbarkeit auszeichnet.

Entwickelt von Guido van Rossum in den späten 1980er Jahren, wurde Python entworfen, um einen Fokus auf Code-Lesbarkeit und Produktivität zu legen.

Python ist eine vielseitige Programmiersprache, die für eine Vielzahl von Anwendungsbereichen geeignet ist. Sie wird häufig in der Webentwicklung, Datenanalyse, künstlichen Intelligenz, maschinellen Lernen, wissenschaftlichen Berechnungen und Automatisierungsaufgaben eingesetzt.

Eine der Stärken von Python ist die große Auswahl an Bibliotheken und Frameworks, die es bietet. Diese Bibliotheken erweitern die Funktionalität von Python und ermöglichen Entwicklern, komplexe Aufgaben mit weniger Code zu erledigen. Beispiele für beliebte

Python-Bibliotheken sind NumPy, pandas, TensorFlow, Django und Flask.

Python ist plattformunabhängig, was bedeutet, dass Python-Programme auf verschiedenen Betriebssystemen wie Windows, macOS und Linux ausgeführt werden können. Python unterstützt auch eine breite Palette von Programmierparadigmen, darunter prozedurale Programmierung, objektorientierte Programmierung und funktionale Programmierung.

Dank seiner Benutzerfreundlichkeit und Flexibilität ist Python sowohl bei Anfängern als auch erfahrenen Entwicklern beliebt. Es bietet eine reichhaltige Community, die Tutorials, Dokumentationen und Unterstützung bereitstellt, was es zu einer großartigen Wahl für Programmieranfänger und professionelle Entwickler macht.

Was ist Torch?

Torch, auch bekannt als PyTorch, ist ein Open-Source-Maschinenlern-Framework, das für die Entwicklung und Bereitstellung von künstlichen neuronalen Netzwerken verwendet wird. Es basiert auf der Programmiersprache Python und bietet eine Vielzahl von Tools und Funktionen, die es Entwicklern ermöglichen, komplexe maschinelle Lernmodelle zu erstellen, zu trainieren und zu optimieren.

Torch wurde von Facebook AI Research entwickelt und zeichnet sich durch seine Flexibilität, Einfachheit und Effizienz aus. Es unterstützt sowohl das Training von Modellen auf CPUs als auch auf GPUs, was die Verarbeitung großer Datenmengen beschleunigt. Torch bietet auch eine einfache Integration mit anderen Bibliotheken und Frameworks wie NumPy und TensorFlow.

Ein Hauptmerkmal von Torch ist sein dynamisches Berechnungsgraph-System. Im Gegensatz zu statischen Berechnungsgraphen ermöglicht das dynamische Berechnungsgraph-System von Torch eine effiziente und flexible Modellierung. Es erlaubt Entwicklern, Modelle schrittweise aufzubauen und während der Ausführung Änderungen vorzunehmen, was die experimentelle Entwicklung und das Prototyping erleichtert.

Torch bietet eine breite Palette von Funktionen und Modulen für maschinelles Lernen, einschließlich vorgefertigter Verlustfunktionen, Optimierungsalgorithmen und Netzwerkarchitekturen. Es unterstützt auch eine Vielzahl von

Anwendungen wie Bilderkennung, Sprachverarbeitung, Zeitreihenanalyse und mehr.

Dank seiner leistungsstarken Funktionen und der aktiven Community hat sich Torch zu einem beliebten Framework in der Forschung und im industriellen Umfeld entwickelt. Es bietet umfangreiche Ressourcen, Dokumentationen und Beispiele, um Entwicklern beim Einstieg und der Weiterentwicklung ihrer maschinellen Lernprojekte zu unterstützen.

Torch wird hauptsächlich zur Entwicklung und Bereitstellung von künstlichen neuronalen Netzwerken (KNN) verwendet. Es bietet eine Vielzahl von Funktionen und Werkzeugen, die es Entwicklern ermöglichen, komplexe maschinelle Lernmodelle zu erstellen, zu trainieren und zu optimieren. Hier sind einige Anwendungen und Einsatzbereiche von Torch:

1. Maschinelles Lernen: Torch wird verwendet, um Modelle für maschinelles Lernen in verschiedenen Anwendungsbereichen zu entwickeln, darunter Bilderkennung, Sprachverarbeitung, natürliche Sprachverarbeitung, Empfehlungssysteme, Zeitreihenanalyse und vieles mehr.
2. Forschung: Torch wird von Wissenschaftlern und Forschern häufig eingesetzt, um neue Modelle und Algorithmen im Bereich des maschinellen Lernens zu erforschen und zu entwickeln. Es ermöglicht die schnelle Prototypenerstellung, die Experimentation mit neuen Ideen und die einfache Anpassung vorhandener Modelle.
3. Computer Vision: Torch bietet umfangreiche Funktionen für die Bilderkennung und Bildverarbeitung. Es wird verwendet, um KNN-Modelle für Aufgaben wie Objekterkennung, Segmentierung, Klassifizierung, Gesichtserkennung und mehr zu entwickeln.
4. Natürliche Sprachverarbeitung: Torch unterstützt die Verarbeitung von natürlicher Sprache und wird für Aufgaben wie Spracherkennung, Textklassifizierung, Sentiment-Analyse, maschinelle Übersetzung und Textgenerierung eingesetzt.
5. Verstärkendes Lernen: Torch bietet Unterstützung für die Entwicklung von verstärkenden Lernmodellen. Diese Modelle lernen durch Interaktion mit einer Umgebung und werden in Anwendungen wie Spielagenten, Robotik und Entscheidungsfindung eingesetzt.
6. Transfer Learning: Torch ermöglicht das Transferieren von gelerntem Wissen von einem vortrainierten Modell auf ähnliche

Aufgaben oder Datensätze. Dies beschleunigt den Lernprozess und ermöglicht die effiziente Nutzung vorhandener Modelle. Torch wird von einer aktiven Community unterstützt und bietet umfangreiche Ressourcen, Bibliotheken und Tools, die Entwicklern helfen, ihre maschinellen Lernprojekte umzusetzen. Es ist ein leistungsstarkes Framework, das Flexibilität, Effizienz und Experimentierfreudigkeit bietet.

Was ist Git?

Git ist ein verteiltes Versionskontrollsystem, das entwickelt wurde, um die Verfolgung von Änderungen in Projekten zu erleichtern. Es ermöglicht Entwicklern, den Verlauf ihres Codes zu überwachen, Änderungen zu verfolgen, verschiedene Versionen zu verwalten und problemlos mit anderen Entwicklern zusammenzuarbeiten. Hier sind einige Hauptmerkmale und Verwendungszwecke von Git:

1. Versionskontrolle: Git zeichnet jede Änderung im Code auf, sodass Entwickler den gesamten Verlauf eines Projekts verfolgen können. Dies erleichtert das Zurückverfolgen von Fehlern, das Wiederherstellen früherer Versionen und das Verwalten von Branches für parallele Entwicklungsarbeiten.
2. Kollaboration: Git ermöglicht die nahtlose Zusammenarbeit mehrerer Entwickler an einem Projekt. Jeder Entwickler kann lokal Änderungen vornehmen und diese dann mit anderen teilen, indem er sie auf einen gemeinsamen Git-Server hochlädt. Git verwaltet die Zusammenführung der Änderungen automatisch und löst Konflikte, die bei gleichzeitigen Änderungen auftreten können.
3. Branching und Merging: Git ermöglicht das Erstellen von Branches, um verschiedene Entwicklungslinien parallel zu verfolgen. Entwickler können neue Features oder Experimente in separaten Branches entwickeln, ohne die Hauptentwicklungslinie zu beeinträchtigen. Das Merging von Branches ermöglicht es, die Änderungen aus verschiedenen Branches zusammenzuführen und so neue Funktionen oder Fehlerbehebungen in den Hauptcode zu integrieren.
4. Rückverfolgbarkeit und Rollback: Mit Git kann man Änderungen verfolgen, wer sie gemacht hat und warum sie gemacht wurden. Dies erleichtert die Überprüfung des Codes, das Finden von

Fehlern und das Rückgängigmachen von Änderungen, wenn etwas schiefgeht.

5. Open-Source-Community: Git ist das am weitesten verbreitete Versionskontrollsystem und wird von einer großen und aktiven Open-Source-Community unterstützt. Dies bedeutet, dass es viele Ressourcen, Tools und Plugins gibt, die die Verwendung von Git erleichtern und erweitern.

Git bietet eine effektive Möglichkeit, den Codeverlauf zu verwalten, die Zusammenarbeit zu verbessern und die Entwicklungsarbeit effizienter zu gestalten. Es ist heute zu einem unverzichtbaren Werkzeug für Entwickler auf der ganzen Welt geworden.

Was ist ein Checkpoint?

Der Checkpoint ist der Kern der Bildverarbeitung und hat Auswirkungen auf die entwickelten Bilder. Die grundlegende Stable Diffusion hat einige Modelle, wobei die beliebteste Version die 1.5 ist. Wenn nicht angegeben wird, auf welchem ursprünglichen Modell sie basiert, handelt es sich wahrscheinlich um das 1.5-Modell. Das ursprüngliche 1.5-Modell ist ein realistisches Modell, das für verschiedene Zwecke verwendet werden kann und nicht speziell optimiert ist. Daher ist es üblich, einen benutzerdefinierten Checkpoint zu verwenden, den du bevorzugst, anstatt das rohe 1.5-Modell selbst zu verwenden. Wenn du ein Anime-Fan bist (wie ich), gelten die Regeln etwas anders. Alle Anime-Modelle verwenden NAI (Novel AI) als Basis, einen ursprünglich kostenpflichtigen Dienst, der sich weiterentwickelt hat und mittlerweile für jede Einrichtung recht zugänglich ist. Du verwendest normalerweise nicht NAI selbst, außer für das Lora-Training (dazu später mehr), aber beliebte Anime-Modelle für die tatsächliche Verwendung wären zum Beispiel die "Anything" und "Abyss Orange"-Serien von Checkpoints.

Was ist ein VAE?

VAE ist sehr wichtig, und wenn dir dieser Begriff nichts sagt, LIES BITTE WEITER. Ein VAE ist die zweite Hälfte eines jeden Checkpoints. Ohne VAE sehen deine Bilder ausgewaschen und farblos aus. VAE sind das getrennte Farbtraining eines Checkpoints. Die meisten Checkpoints haben ein empfohlenes VAE, aber VAEs sind austauschbar, sodass du dein bevorzugtes VAE mit verschiedenen Checkpoints

verwenden kannst. Achte nur darauf, dass immer eines aktiv ist, damit deine Bilder nicht verwässert aussehen. Anmerkung: VAEs können in den Einstellungen aktiviert werden. Manchmal ist ein Neustart erforderlich, um es zu aktivieren. Außerdem kannst du in der "[info] Quicksettings-Liste" nachsehen und es an die Spitze deiner Konsole hinzufügen, wenn du "automatic1111" verwendest.

Was ist eine Lora?

Ein LORA ist eine Low-Rank-Adaption-Datei. Es fügt deiner KI neue Konzepte und Begriffe hinzu, die ursprünglich nicht enthalten waren. Hier kommt die individuelle Anpassung einer Person wirklich zum Ausdruck. LORAs werden in Prozentangaben auf deinen Checkpoint angewendet, um ihr individuelles Training dem verwendeten Checkpoint hinzuzufügen. Wenn du Bilder erstellst, musst du auf die verschiedenen Prozentsätze der einzelnen LORAs achten. Du kannst nicht eine große Menge an LORAs mit ":1" (=100%) verwenden.

Was sind Hypernetwork und Textual Inversion?

Hypernetwork und Textual Inversion ähneln Lora, da es sich um separate Trainingseinheiten handelt, die du in deinen Checkpoint einfügen kannst. Hypernetwork folgt ebenfalls dem Prozentsystem, während Textual Inversion auf ein einzelnes Wort beschränkt ist. Textual Inversion ist normalerweise ein Thema und erfordert keine weiteren Eingaben, nachdem du es in das Eingabefenster eingegeben hast.

Was sind Prompts?

Denke daran, dass es positive und negative Prompts gibt. Normalerweise generieren die meisten Bilder gut mit kurzen Prompts, aber du wirst Dinge wie "highres" in den positiven Prompts sehen. Für Eigenschaften gibst du Dinge wie "lowres" in die negativen Prompts ein. Es gibt auch eine Vielzahl von Einbettungen (embeddings), die du verwenden kannst. Einbettungen ähneln der Textual Inversion, werden aber normalerweise für die negativen Prompts verwendet. Finde deine bevorzugte Mischung von negativen Einbettungen und gib sie in das negative Prompt-Fenster ein oder gib sie einfach selbst ein. Bei den positiven Prompts, wenn du ein Lora verwendest, lies und verwende unbedingt die aufgeführten Trigger, falls vorhanden.

Ein Leitfaden von der Installation bis zum Lernen

Nachdem du alles gelesen hast, könntest du jetzt versuchen, ein LORA zu erstellen, um eigene individuelle Bilder zu kreieren, falls das Herunterladen von Bildern anderer nicht genug für dich ist. Für leistungsfähige Desktops empfehle ich die Kohya-Tools für das KI-Training. Kohya nutzt dein Gerät, um einen Checkpoint für realistisches Training auf SD 1.5 pruned oder für Anime-Training auf NAI pruned zu trainieren. Wenn du dein eigenes Gerät nicht verwenden möchtest, gibt es einen Trainer von hallowstrawberry in Google Colab.

Warum diese Modelle und was bedeutet "pruned"?

"Pruned" bedeutet, dass der Checkpoint vereinfacht wurde und speziell für das Training entwickelt wurde. Ich empfehle das Training auf den genannten Modellen aufgrund ihrer Kompatibilität. Einige Checkpoints haben auch benutzerdefinierte pruned-Varianten, aber wenn du auf solchen trainierst und es auf einem anderen Checkpoint ausprobierst, wird das Ergebnis schlechter aussehen. SD 1.5 ist der Ursprung aller Checkpoints und mit allen Realismus-Modellen kompatibel. Gleiches gilt für NAI und Anime-Modelle, wobei NAI auch rückwärts mit einigen Realismus-Modellen funktionieren kann, jedoch nicht optimal darauf ausgelegt ist.

Das Erstellen eigener Checkpoints für Stable Diffusion ist ein spannender Weg, um kreative Möglichkeiten zu erkunden und ein tieferes Verständnis der künstlichen Intelligenz zu gewinnen. In diesem Artikel werde ich euch einen umfassenden Leitfaden bieten, der euch Schritt für Schritt zeigt, wie ihr mit dem Programm Kohya eigene Checkpoints erstellen könnt. Ihr werdet lernen, wie ihr das Programm installiert, wie ihr damit arbeiten und eure eigenen Checkpoints trainieren könnt. Bitte beachtet, dass ich hierbei von einer Windows-Installation ausgehe, weil ich selbst Windows nutze.

TERMINI- Again...

DATASET:

Der Datensatz von Bildern sollte eine Vielzahl von Blickwinkeln und Entfernungen von deinem Zielobjekt haben. Wenn du ein Charakter-Training durchführst, solltest du sicherstellen, dass du sowohl Nahaufnahmen des Gesichts als auch Ganzkörper- und Ausschnittsbilder verwendest, damit die KI das Aussehen und das Erscheinungsbild wirklich verstehen kann. Datensätze können auch mit unterschiedlichen Werten trainiert werden, damit qualitativ hochwertigere Bilder mehr Gewicht im Training haben. Dies ist auch der Weg, wie du verschiedene Begriffe in einer einzelnen Lora trainierst, wenn du eine Lora entwirfst, die ein Auslösewort oder einen Auslösesatz hat (ein erforderliches Wort oder eine erforderliche Phrase, um eine Lora zu aktivieren). Nicht alle Lora benötigen jedoch ein Auslösewort. Zum Beispiel können die meisten Stile einfach nur durch Lora aktiviert werden. Wenn du einen Datensatz mit einem Auslösewort erstellst, achte darauf, dass das Auslösewort das erste Wort in deiner Tag-Prompt-Liste ist.

Training

Für die verschiedenen Lora-Typen ist es wichtig zu beachten, dass sie unterschiedlich sind und separat aufgeschlüsselt und trainiert werden müssen: Charaktere, Stile, Hintergründe und tatsächliche Konzepte. Am wichtigsten sind bei Kohya oder Google Colab drei Dinge: Epochen, Text-LR und Unet-LR.

Lernraten:

Die Lernraten oder LR sind der Hauptfaktor, der von Lora zu Lora angepasst und geändert werden muss. Zum Beispiel haben Stile im Durchschnitt etwas niedrigere Lernraten als Charaktere, weil sie allgemeiner sind. Um eine gute Mischung für dein Startprojekt zu finden, empfehle ich, einfach die gleichen Werte wie jemand zu verwenden, der etwas Ähnliches gemacht hat, das dir gefällt. Du kannst auch immer wieder verschiedene Werte ausprobieren, bis du zufrieden bist (aber beginne wirklich mit einem ähnlichen Wert, den du magst). Du kannst die Informationen zu jeder Lora mit der "sd-webui-additional-networks" Erweiterung anzeigen, die auch im Erweiterungstab hinzugefügt werden kann.

Epochen:

Eine Epoche ist die Anzahl der Male, die deine Daten normalerweise überprüft werden. Normalerweise würdest du jedes einzelne Bild etwa 10 Mal in den Trainingsschritten überprüfen lassen. Durch Hinzufügen von mehr Epochen wird die KI den gesamten Datensatz beispielsweise 10 Mal betrachten (ich verwende für alles tendenziell 10 Epochen), insgesamt also 100 Mal. Durch die Verwendung von minimalen Epochen erhältst du eine minimale Stärke für deine Lora. Aus meiner persönlichen Erfahrung würde ich nicht weniger als 8 Epochen oder mehr als 12 Epochen empfehlen.

Vorbereitung:

Gehe auf deine Festplatte zB.: [C:] und lege dort einen Ordner namens "AI" an. In diesem Ordner legst du 2 weitere an einen mit dem Namen "stable-diffusion-webui" und einen mit dem Namen "Kohya". Dort entpackst du später die Dateien. Das dient rein zur Struktur. Du kannst sie natürlich auch frei umplatzieren wo du möchtest dennoch wird dich, mit der Zeit, Stable und Co zu ordnung zwingen da viele Bilder auch viel Speicher bedeutet.

Abschnitt 1: Die Installation von Kohya

Kohya-ss Install: https://github.com/bmaltais/kohya_ss

- Installiere [Python 3.10](#)
 - Stelle sicher, dass das Kästchen angekreuzt ist, um Python zur Umgebungsvariable "PATH" hinzuzufügen.
- Installiere [Git](#)
- Installiere [Visual Studio 2015, 2017, 2019, and 2022 redistributable](#)

Kohya benötigst du, um zB eigene Stile in dein Stable Diffusion zu laden. Diese dienen wie eine "Datenbank" an Bildern und greifen daraufhin zurück. Du kannst dir das vorstellen, als würdest du mit Kohya deiner Ki beibringen so zu zeichnen wie zu du oder fotorealistic Bilder zu generieren oder Bilder von Katzen zu perfektionieren, sodass sie genau so aussehen wie dein Haustier. Deiner Fantasie sind keine Grenzen gesetzt.

Hast du alle Notwendigen Programme heruntergeladen. Starte die Kohya-ss Installation: https://github.com/bmaltais/kohya_ss lade die Datei herunter und packe sie in deinen Kohya Ordner. Extrahiere sie und starte **die gui-user.bat** . Diese befindet sich relativ unscheinbar zwischen den anderen Dateien. Starte nun Kohya und Kohya hat einige Fragen an dich. ES IST WICHTIG DIESE FRAGEN GUT ZU LESEN UND EVENTUELL ZU RECHERCHIEREN. Wenn dies eine Erstinstallation ist, antworte mit "NO", wenn du gefragt wirst, ob du vor der Installation vorherige Versionen von Torch und zugehörigen Dateien deinstallieren möchtest.

Aktualisierung:

Die folgenden Befehle funktionieren vom Stammverzeichnis des Projekts aus, wenn du Skripte nicht ausführen möchtest

```
git pull
.\venv\Scripts\activate
pip install --use-pep517 --upgrade -r requirements.txt
```

Checkpoint, Lora, Model? - Na was denn nun?

Erlaube mir, eine Analogie aus der Welt der Bücher zu verwenden:

Modell - Enzyklopädie

LoRa - ein zusätzlicher Eintrag in der Enzyklopädie, der als "Post-it"-Notiz darin versteckt ist.

Textual inversion - eine Liste von Seiten mit Einträgen in der Enzyklopädie, die mit dem Konzept, das du zu verstehen versuchst, zusammenhängen.

Vielleicht könnte man jetzt schon rückschließen, wie man welche Methode benutzt, doch bedenkt, ihr müsst nicht immer das Rad neu erfinden, gerade nicht bei Loras. Seht euch genau an, was ihr zB an eurem Stil verändern wollt und ändert dann das Post it ab. Stellt euch das wie eine Leinwand mit Grundmotiv vor, welche der Checkpoint ist, das kann ein total vanilla oder 08/15 Checkpoint sein. Denkt euch nun das Zeichenmittel zb Acryl, das legt ihr mit eurer Lora fest [könnt ihr festlegen, es gibt allerdings auch schon checkpoints die einen Artstyle haben aber dazu komm ich gleich] nun malt euer

Bild und je nachdem wie ihr eure Lora trainiert habt oder auf was ihr sie trainiert habt wird euer zb gewünschtes portrait auch aussehen. - Ist sehr vereinfacht erklärt natürlich gibt es da viel viel viel mehr Möglichkeiten die ich vermutlich selbst nicht einmal kenne, aber das ist wie ich es aktuell benutze. Nun stellt euch vor, ihr habt bereits einen Checkpoint der weis wie ihr malt und was ihr so darstellt und ihr merkt dass die Gesichter eher "Meh" werden, dann erstellt ihr einen Lora mit nur Portraits und verwendet den mit dem Checkpoint eures Syls. So sieht es aus , als würdet ihr über das verhundste Gesicht auf dem Bild ein neues Papier drauf kleben, das besser aussieht.

Wie mache ich das jetzt genau?

Vorweg: Vorbereitung. Such dir nun aus ob du einen checkpoint doer eine Lora erstellen willst. Egal in welchem FALL gehts du in dein Verzeichnis mit den ordnern und erstellst:

C:\Ai\Khoya\Checkpoint_training

danach erstellst du: [DEINNAME steht für einen von dir gewählten Namen]

C:\Ai\Khoya\Checkpoint_training\DEINNAME

Darin bedinfen sich 4 Weitere Ordner:

-> images [Da kommen deine KOMPRIMIERTEN BILDER REIN!]

-> logs [Zwischenspeicher "Logbuch"]

-> model [da wird dein Endprodukt rein generiert.]

-> source [Für dich als Bildersammlung, welche du verwendet hast.]

so: bis auf den Ordner images ignorieren wir die anderen Ordner einfach. IN den Images Ordner legst du einen Ordner an und ACHTUNG, die Benennung ist hier wichtig, denn er besteht aus einer Zahl einem "_" und dem frei wählbaren Namen deines Projektes. zb 100_Noctelia. Die Zahl im vorderen Block beschreibt, wie genau sich die KI deine Bilder ansehen soll, wobei 100x schon echt viel ist. Ich habe bei 224 Bilder im Ordner 100_Noctelia und x 100 ansehen mit ner Nvidia1070 8,4h gebraucht um meine erste Lora zu erstellen. Dämlich, aber möglich. Also versucht es vielleicht am Anfang mit weniger Bildern und weniger "Begutachtungs-Durchgängen".

Habt ihr den Ornder XXX_XXXXX erstellt nehmt nun eure Bilder und komprimiert sie zb auf der Seite:

BIRME: https://www.birme.net/?target_width=512&target_height=512

dort könnt ihr auch Ausschnitte anpassen beim Komprimieren eurer Bilder. Ihr könnt aber auch Bulk Resize:

<https://bulkresizephotos.com/en> nehmen. Ladet die .rar oder .zip nun in euren XXX_XXX Ordner und entpackt.

Möchtet ihr nun einen Checkpoint erstellen: öffnet nun Kohya. Geht nun auf den Reiter: Dreambooth und unter dem Reiter: Source Model Wählt bei dem Datei Symbol **runwayml/stable-diffusion-v1-5** aus das sollte auch unter Model Quick Pick stehen. Habt ihr diese Datei noch nicht ladet sie euch bei Huggingface herunter und packt sie in euren Stable-difusion Ordner unter Models. LINK zur Version 1.5 <https://huggingface.co/runwayml/stable-diffusion-v1-5/tree/main>

klickt nun auf den Reiter FOLDERS:

und tragt bei den passenden Tags eure erstellten Folder ein. Dh Images bei Images etc. habt ihr einen Folder nicht [wird der Regularisation sein] - den hab ich auch nicht und ich habe keine Ahnung wofür der sein soll. xD

Tragt nun bei Model Output Name EUEN PROJEKTNAMEN EIN.

Jetzt gehen wir zu den Trainings Parametern:

Train Batch Size 1 - Ändert es nicht, außer ihr habt ein massives Vieh an Grafikkarten im PC oder seid ein Studio. Alles über 1 ist für sehr Leistungsstarke Rechner gedacht mit ner guten GPU und massig Vram. Ich tel mit euch meine Einstellungen die ich mit meiner 1070 erwirkt hab also ist da definitiv luft nach oben und das müsst ihr für euch selbst ausknobeln, jedoch können meine Werte ruhig als Anfänger Richtwerte dienen. Epoch beschreibt wie oft ihr Zwischenspeicher und Schritte ihr Geht, bei mir hat sich ein Wert zwischen 1-3 als stabil herausgestellt. Gute Karten sollen mehr Epochen zulassen können.

Ignoriert erstmal die anderen Werte, geht nun zum Optimizer. Bei mir hat **AdamW8bit** nicht funktioniert, ich habe alles mit **LoIn** generiert. Wie gut, schlau oder dämlich das war, kann ich euch noch nicht sagen, aber ich konnte es zumindest generieren.

Geht nun auf MAX Resolution uns stellt 512,512 ein. Wenn ihr ne bessere Karte habt passt das auch bitte wieder an.

So jetzt fehlt noch ein wichtiger Schritt. Nachdem ihr euer Archiv, sprich eure Bilder in dem XXX_XXXX Ordner habt und sie komprimiert habt müsst ihr die Bilder sinnig umbenennen Markiert alle und drückt F2 und gebt ihnen am besten den Namen eures Projekts . Euer PC erstellt automatisch eine Nummernreihe daraus und benennt die Dateien um. Kontrolliert nochmal, ob irgendwelche Bilder dabei sind, die falsch zugeschnitten oder blurry sind, denn diese könnten euren Putput mindern.

Gene nun noch einmal in dein Kohya und klicke unter dem Reiter UTILITIES/CATIONING/W14 CAPTIONING

Hier wollen wir nun, dass sich die KI einmal unsere Bilder ansieht, um ihnen Text zuzuweisen, sprich eine Beschreibung der Bilder, sprich einen Prompt. Dazu wähle deinen XXX_XXXX Ordner aus. Alle andere Einstellungen -> ignorieren. Drücke nun auf Caption images und in deinem XXXX_XXXX Ordner befindet sich neben jeder .PNG nun eine TXT welche den Prompt beinhaltet.

Du könntest dir wie ich die mühe machen und jede TXT öffnen und auf Sinnigkeit überprüfen ODER du machst es dir mit diesem tool einfacher:

TAG MANAGER:

<https://github.com/starik222/BooruDatasetTagManager/releases/tag/v1.6.3>

lade diesen runter und achte darauf die **ALLINCLUDE.Zip** downzuloaden! Entpacke nun irgendwo [Empfohlen im AI Ordner] auf deinem Pc diese datei und führ sie mit **BooruDatasetTagManager.exe** aus. Gehe auf File/loadFolder und lade deinen XXX_XXXX Folder rein. Das Programm erkennt die TAGs und die Bilder und zeigt dir Gesammelt ALLE Tags. Es ist selbsterklärend und einfach zu bedienen. Du wirst es lieben.

Hast du die Tags angepasst kannst du nun in Kohya auf DREAMBOOTH zurück gehen und auf Train Model klicken. Und es Sollte alles klappen beobachte dabei immer den Controller ob fehlermeldungen auftauchen.

Hast du einen CUDA Out of Memory ERROR - oder so ähnlich, reduziere zB die Bildgröße oder deine Epochen.

..UND BEI EINER LORA?

Selbes Prinzip. Grundsätzlich ist ca. das Gleiche einzustellen. Ich würde dir nur auf jeden Fall empfehlen, für eine Lora wesentlich ähnliche und thematisch gleiche Bilder zu nehmen. Die sollten auch deutlich weniger sein. Also wenn eine Ki dein Gesicht lernen soll, sind 25 Bilder ausreichend . Am besten aus verschiedenen Winkeln und Posen, verschiedener Kleidung, Hüten etc. Ein Großenminimum bildet wieder 512,512.

TIPP: Logos Watermarks und Signaturen entfernen am besten. So Pirateriemäßig es sich anhört aber die Ki lernt sonst auch dir jedes mal ein fremdes Wasserzeichen drauf zu machen.

Abschnitt 2: Die Grundlagen von Stable Diffusion

Stelle sicher, dass die erforderlichen Abhängigkeiten erfüllt sind und befolgen Sie die verfügbaren Anweisungen sowohl für NVidia (empfohlen) als auch für AMD-GPUs.

~Alternativ kannst du auch Online-Dienste wie Google Colab verwenden.~

Liste der Online-Dienste

Installation unter Windows 10/11 mit NVidia-GPUs mithilfe des Release-Pakets lade : **sd.webui.zip von v1.0.0-pre** herunter und extrahiere den Inhalt.

Führe **update.bat** aus.

Führe **run.bat** aus.

[Solltest du Probleme beim Starten bzw ausführen der run.bat haben prüfe mal deinen Grafikkarten treiber, Phyton, Git. Sollten immer noch probleme auftreten kannst du versuchen das Programm zu forcen mithilfe von Powershell scripts: aber lies besser hier nochmal nach: <https://github.com/AUTOMATIC1111/stable-diffusion-webui/wiki/Install-and-Run-on-NVidia-GPUs>]

Automatische Installation unter Windows

Installiere **Python 3.10.6** (Neuere Versionen von Python unterstützen torch nicht -> stand Mai/23), aktiviere die Option "Python zum PATH hinzufügen".

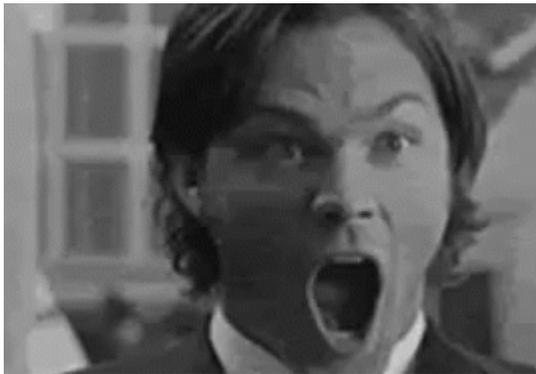
Installiere Git.

Lade das Repository **stable-diffusion-webui** herunter, z. B. durch Ausführen von `git clone`

<https://github.com/AUTOMATIC1111/stable-diffusion-webui.git>.

Führen Sie **webui-user.bat** im Windows Explorer als normaler Benutzer ohne Administratorrechte aus. Die `webui-user.bat` ist super versteckt und sieht auch nicht besonders aus, sie ist inmitten des gesamten Gewürschfels an Dateien. Achte darauf, dass du die richtige erwischst, sonst startet gar nichts.

Sollten dabei auch wieder Fehler auftreten...



... Panic.

Kleiner Scherz.

TXT to IMG

Offne dein Stabe und Kopiere die "IP" IN dein Browserfenster dann öffnet sich das programm. Du siehst den "txt2img" Tab, wenn du die GUI zum ersten Mal startest. Dieser Tab erfüllt die grundlegendste Funktion von Stable Diffusion: das Umwandeln eines Text-Prompts in Bilder. Wähle den gewünschten Stable Diffusion Checkpoint aus. Das muss kein eigener sein, du kannst welche aus dem Internet downloaden und in deinen MODEL Folder in Stable difusion ziehen, aber Erstbenutzer können das v1.5 Basismodell verwenden.

Gib einen Prompt ein, der beschreibt, was du auf den Bildern sehen möchtest. Hier ist ein Beispiel.

"A surrealist painting of a cat by Salvador Dali"

Gib die Breite und Höhe des Ausgabebildes an. Für ein v1-Modell solltest du mindestens eine Seite auf 512 Pixel setzen. Zum Beispiel kannst du die Breite auf 512 und die Höhe auf 768 für ein Hochformatbild mit einem Seitenverhältnis von 2:3 setzen.

Wähle die Batchgröße aus, also die Anzahl der pro Durchlauf generierten Bilder. Es ist ratsam, mindestens einige Bilder zu generieren, um den Prompt zu testen, da jedes Bild unterschiedlich sein wird.

Drücke abschließend die Schaltfläche "Generieren". Nach einer kurzen Wartezeit erhältst du deine Bilder! Standardmäßig erhältst du ein zusätzliches Bild mit zusammengesetzten Miniaturansichten.

Du kannst ein Bild in deinem lokalen Speicher speichern. Wähle dazu das gewünschte Bild mithilfe der Miniaturansichten unterhalb der Hauptbildfläche aus. Klicke mit der rechten Maustaste auf das Bild, um das Kontextmenü aufzurufen. Dort solltest du Optionen zum Speichern des Bildes oder zum Kopieren des Bildes in die Zwischenablage finden.

Das sind die grundlegenden Informationen, die du benötigst! Der Rest dieses Abschnitts erläutert jede Funktion detaillierter.

Der Stable Diffusion Checkpoint ist ein Dropdown-Menü zum Auswählen von Modellen. Du musst Modelldateien im Ordner `stable-diffusion-webui > models > Stable-diffusion` platzieren. Weitere Informationen zur Installation von Modellen findest du [hier](#).

Der **refresh button** neben dem Dropdown-Menü dient zum Aktualisieren der Modellliste. Sie wird verwendet, wenn du gerade ein neues Modell in den Modellordner gelegt hast und die Liste aktualisieren möchtest.

Texteingabefeld "Prompt": Gib an, was du auf den Bildern sehen möchtest. Sei detailliert und spezifisch. Verwende bewährte Schlüsselwörter. Eine kurze Liste findest du [hier](#), oder eine ausführlichere Liste im Prompt Generator.

Texteingabefeld "Negative Prompt": Gib an, was du nicht sehen möchtest. Du solltest einen negativen Prompt verwenden, wenn du v2-Modelle verwendest. Du kannst einen universellen negativen Prompt verwenden. Weitere Details findest du in [diesem Artikel](#).

Sampling Method: Der Algorithmus für den Denoising-Prozess. Ich verwende DPM++ 2M Karras, da er eine gute Balance zwischen Geschwindigkeit und Qualität bietet. Weitere Details findest du in [diesem Abschnitt](#). Du solltest am besten alle sampler (diejenigen mit

einem "a") vermeiden, da ihre Bilder auch bei großen Abtastschritten instabil sind, was das Feinabstimmen des Bildes erschwert.

Sampling Steps: Anzahl der Abtastschritte für den Denoising-Prozess. Je mehr, desto besser, aber es dauert auch länger. 25 Schritte funktionieren in den meisten Fällen.

Breite und Höhe: Die Größe des Ausgabebildes. Für v1-Modelle solltest du mindestens eine Seite auf 512 Pixel setzen. Zum Beispiel kannst du die Breite auf 512 und die Höhe auf 768 für ein Hochformatbild mit einem Seitenverhältnis von 2:3 setzen. Wenn du das v2-768px-Modell verwendest, setze mindestens eine Seite auf 768 Pixel.

Batch-Anzahl: Anzahl der Durchläufe der Bildgenerierungspipeline.

Batch-Größe: Anzahl der zu generierenden Bilder pro Durchlauf der Pipeline.

Die Gesamtzahl der generierten Bilder ergibt sich aus der Batch-Anzahl multipliziert mit der Batch-Größe. Normalerweise änderst du die Batch-Größe, um die Geschwindigkeit zu erhöhen. Die Batch-Anzahl änderst du nur, wenn es Probleme mit dem Speicher gibt.

CFG-Skala: Classifier Free Guidance-Skala ist ein Parameter, um zu steuern, wie sehr das Modell deinem Prompt folgen soll.

1 - Ignoriere den Prompt weitgehend.

3 - Sei kreativer.

7 - Ein guter Kompromiss zwischen dem Folgen des Prompts und Freiheit.

15 - Halte dich stärker an den Prompt.

30 - Folge dem Prompt strikt.

Die Bilder unten zeigen die Auswirkungen der Änderung der CFG bei festgelegten Seed-Werten. Du solltest die CFG-Werte weder zu niedrig noch zu hoch setzen. Wenn der CFG-Wert zu niedrig ist, ignoriert Stable Diffusion deinen Prompt.

CFG 1



CFG 3

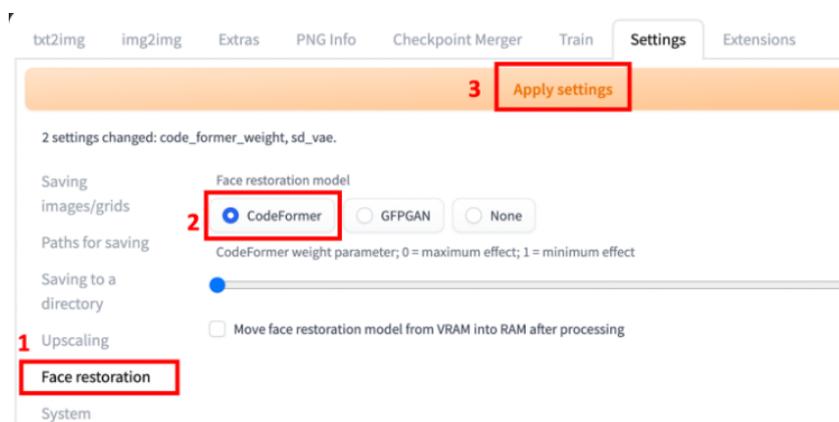


CFG 7



Seed: Der Seed-Wert wird verwendet, um den anfänglichen zufälligen Tensor im latenten Raum zu generieren. Praktisch gesehen kontrolliert er den Inhalt des Bildes. Jedes generierte Bild hat seinen eigenen Seed-Wert. Wenn AUTOMATIC1111 auf -1 gesetzt ist, wird ein zufälliger Seed-Wert verwendet. Ein häufiger Grund, den Seed zu fixieren, besteht darin, den Inhalt eines Bildes festzulegen und den Prompt anzupassen. Angenommen, ich habe ein Bild mit folgendem Prompt generiert.

RESTORE FACES



Du musst angeben, welches Gesichtsrestaurationsmodell du verwenden möchtest, bevor du "Gesichter wiederherstellen" verwenden kannst. Besuche zuerst den Einstellungen-Tab. Navigiere zum Abschnitt "Gesichtsrestaurierung". Wähle ein Gesichtsrestaurationsmodell aus. CodeFormer ist eine gute Wahl. Setze das CodeFormer-Gewicht auf 0 für maximale Wirkung. Vergiss nicht, auf die Schaltfläche "Einstellungen anwenden" zu klicken, um die Einstellungen zu speichern! Gehe zurück zum "txt2img"-Tab. Aktiviere "Gesichter wiederherstellen". Das Gesichtsrestaurationsmodell wird auf jedes von dir generierte Bild angewendet. Wenn du feststellst, dass die Anwendung den Stil auf den Gesichtern beeinflusst, möchtest du möglicherweise die Gesichtsrestaurierung deaktivieren. Alternativ kannst du den CodeFormer-Gewichtsparemeter erhöhen, um die Wirkung zu verringern.