

DrAI

P2431522

Kategorie: ICT & Digital - “Vertrauenswürdige Künstliche Intelligenz”

Schule: HTL-Neufelden

Adresse: 4120 Neufelden, Höferweg 47

Tel: +43 7282-5955

E-Mail: info@htl-neufelden.at

Projektkoordinator:

Name: Samuel Nösslböck

Tel: +43 669 1427 54 44

E-Mail: samuel.noesselboeck@gmail.com

Projektbetreuer:

Name: Dipl. -Ing. Peter Rachinger

Tel: +43 664 21 08 59 7

E-Mail: pe.rachinger@htl-neufelden.at

Projektteilnehmer:

Name: Rene Schwarz

Tel: +43 677 63 08 75 03

E-Mail: rene3.schwarz@gmail.com

Kooperationspartner:

Name: Ars Electronica Center Linz

Projekthomepage: <https://github.com/SamuelNoesslboeck/DrAI>

1. Inhaltsverzeichnis

1. Inhaltsverzeichnis.....	2
2. Projektdokumentation.....	3
2.1. Produktentstehung und -planung.....	3
2.1.1. Ideenfindung.....	3
2.1.2. Recherche und Innovation.....	4
2.1.3. Reifung und Lizenzierung.....	5
2.1.4. Projektplanung und Meilensteine.....	6
2.1.5. Zeitplan und derzeitiger Stand.....	7
2.2. Inhaltliche Beschreibung.....	8
2.2.1. Finales Konzept und Aufgabe.....	8
2.2.2. Roboter.....	9
2.2.2.1. Design.....	9
2.2.2.2. Interaktion mit dem*der Benutzer*In.....	9
2.2.2.3. Mechanik.....	10
2.2.2.4. Elektronik.....	10
2.2.2.5. Steuerungssoftware.....	10
2.2.3. Künstliche Intelligenz und Software.....	11
2.2.3.1. Einzigartigkeit.....	11
2.2.3.2. Parametrisierung.....	11
2.2.3.3. Ergebnisse.....	11
2.2.3.4. Entstehung / Experimente.....	13
2.2.3.4.1. Eigenes Neuronales Netzwerk.....	14
2.2.3.4.2. StablePipe.....	16
2.2.3.4.3. InterStablePipe.....	18
2.2.3.4.4 InterStableCloud.....	21
2.2.3.4.5 InterStableLLM Pipeline.....	25
2.2.3.4.6 Finale Umsetzung, InterStableLLMRLLine.....	28
2.2.4. Ausblick, Entwicklungspotential und Wirtschaftlichkeit.....	29
2.3. Projektkoordination.....	30
2.3.1. Kompetenzen des Teams.....	30
2.3.2. Kooperation und Betreuung.....	31
2.3.3. Kosten.....	32
3. Literaturverzeichnis.....	33
4. Bildverzeichnis.....	35

2. Projektdokumentation

2.1. Produktentstehung und -planung

2.1.1. Ideenfindung

Von klein auf waren wir das, was umgangssprachlich “Bastler” genannt wird, in den letzten Jahren vor allem in den Bereichen Robotik und KI. Da wir uns schon seit der Unterstufe kennen und uns gegenseitig immer wieder Fotos und Fortschritte unserer Basteleien zeigten, war die Findung eines Projektteams keine Aufgabe von großer Schwierigkeit.

Das Ziel war es, ein Projekt umzusetzen, das wirklich an die Grenzen geht, deshalb wurde sich für eine Zusammenarbeit mit einem Museum entschieden. Durch die guten Kontakte des Beratungslehrers, Herrn Dipl.-Ing. Peter Rachinger, konnte ein Meeting mit dem FutureLab des *Ars Electronica Centers (AEC)* organisiert werden. Die Aufgabe seitens des *AEC* nach der Auflistung persönlicher Interessen lautete:

Es sollte ein Projekt erstellt werden, welches den Besuchern Künstliche Intelligenz näherbringt.

In Absprache mit dem Projektpartner und des Projektbetreuers wurde über zahlreiche Konzepte diskutiert. Da die Möglichkeiten äußerst vielfältig sind, standen Systeme zur Debatte, die selbst Musik komponieren und Marionetten, die Gesten imitieren, bis sich schlussendlich auf die Idee geeinigt wurde, analoge Zeichnungen mit neuen KI-Technologien zu erstellen. Nach einigen weiteren Überlegungen entstand die finale Idee: ein von KI unterstützter, zeichnender Roboter.

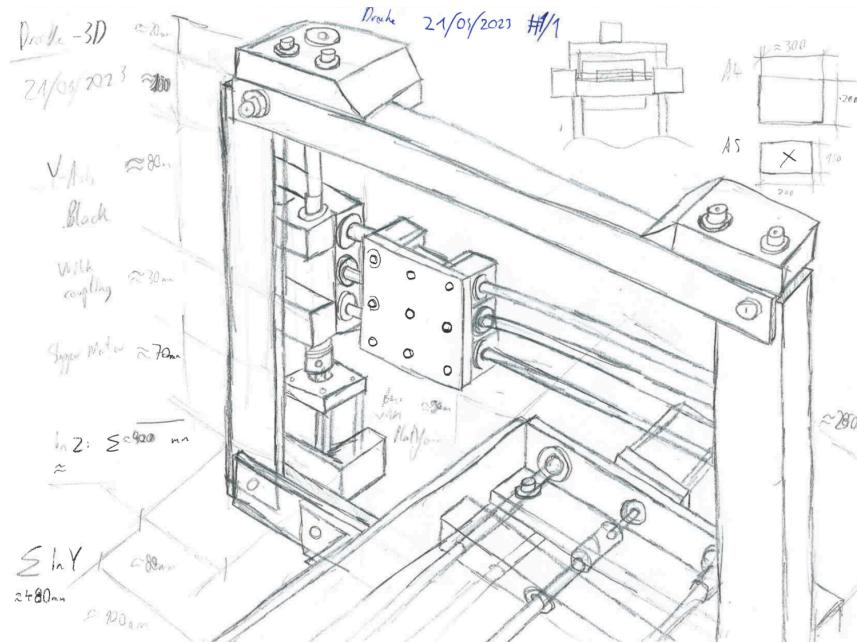


Abbildung 2.1.1.1 Erste Konzeptskizze

2.1.2. Recherche und Innovation

Eine Internetrecherche in Zusammenarbeit mit dem Projektleiter ergab, dass es zwar schon Zeichenroboter gab, die auf gewisse Datensätze trainiert wurden, jedoch funktionierten diese nur auf kleine Datensätze oder Zeichnungen und wiesen nicht die benötigte gewünschte Vielfältigkeit und Kreativität auf. Auch waren viele der existierenden Konzepte nicht für den Museumsbetrieb geeignet.

Die anfängliche Idee des Projektes war, zu einer bestehenden Zeichnung etwas Kreatives hinzuzufügen und die Besucher*innen mit der KI kollaborieren zu lassen. Das Projekt sollte zur Demonstration von kreativer Zusammenarbeit von Künstlicher Intelligenz und Mensch dienen.

Zu Beginn des Projektes existierte die Meinung, künstliche Intelligenz werde in den nächsten Jahren zu einem neuen Werkzeug für die Menschheit, wie die Erfindung des Taschenrechners. Bereits zu dieser Zeit existierten diverse Algorithmen, mit denen sich durch menschliches Einwirken Herausragendes entwickeln lässt. Durch Programme wie *Github Copilot* lassen sich komplexe Programme in beliebigen Programmiersprachen durch eine Textbeschreibung entwickeln. Oft genügt eine einfache Beschreibung des Lösungsansatzes und schon liefert ein *Large Language Model (LLM)* wie *GPT-4* den Programmcode. Was jedoch hierbei wichtig ist, ist die Beschreibung des Lösungsansatzes. Neuronale Netzwerke sind nur so gut wie ihre Trainingsdaten. Befindet sich das Problem nicht in den Datensätzen, so fällt es dem Programm meist schwer, eine komplexe Lösung zu finden, die auch wirklich funktioniert. Hierbei wird die menschliche Kreativität benötigt, um einen Lösungsansatz zu finden, der dann beschrieben und in Code umgesetzt wird.

Mit Bildern ist es das gleiche, es wird menschliche Kreativität benötigt, um kreative Textbeschreibungen für Bilder zu finden, die im Anschluss gezeichnet werden. Je weiter die Forschung des Algorithmus fortschritt, desto mehr festigte sich die Erkenntnis, dass diese These inkorrekt ist. Der Programmcode hat einerseits die Möglichkeit, aus Bestehendem etwas Neues zu erzeugen und andererseits aus einem weißen Blatt Papier ohne textliche Beschreibungen oder menschliches Einwirken kreative Bilder zu malen. Es ist eine philosophische Frage, ob Maschinen kreativ sein können oder nicht, jedoch sehen die Resultate sehr vielversprechend aus. Die Frage, ob KI kreativ sein kann, wird eher zur Frage, wann KI kreativer wird, als die Menschheit es derzeit ist.

Der zunächst beschriebene Algorithmus versucht die menschliche Kreativität nachzuahmen, indem er generative Künstliche Intelligenz mit zufällig in das Eingangsbild generierten Inhalten kombiniert. Die Innovation liegt in dieser Interpretation menschlicher Kreativität als genau dieser Kombination von zufälligen Impulsen in der menschlichen Wahrnehmung, abgebildet durch die Zufallsgeneratoren, und der Erkennung von Mustern, abgebildet durch KI.

2.1.3. Reifung und Lizenzierung

Im Laufe des Projekts und mehreren Meetings mit dem *AEC* nahm das Projekt vor allem in Design-Aspekten Form an. Eine dieser visuellen Anpassungen ist zum Beispiel der Rahmen, der durch ein Oktagon ersetzt wurde.

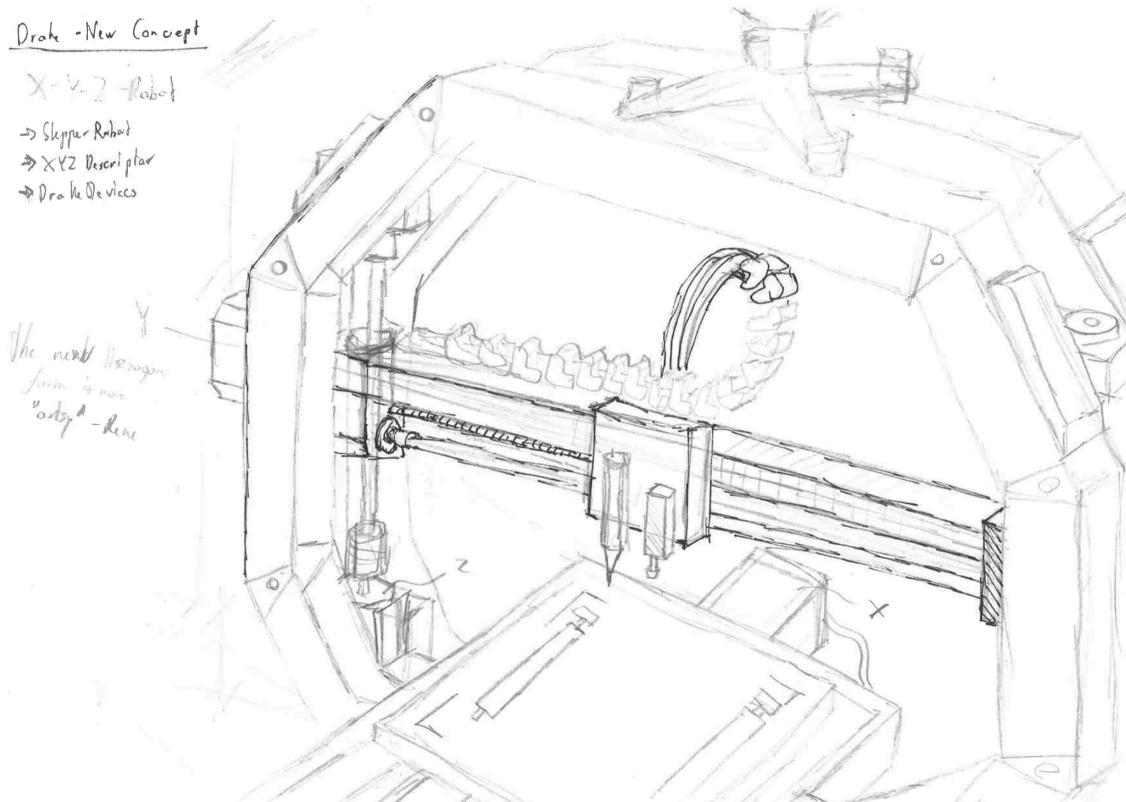


Abbildung 2.1.3.1 Skizze Octagon

Mit einem relativ unberührten Grundkonzept entstand dann im September 2023 ein museumstauglicher Roboter, der im Sommer 2024 ein Teil der Ausstellung werden soll. Das anfänglich eher nebensächliche KI-Projekt entwickelte sich zu einer intensiven Auseinandersetzung mit den Grenzen der KI und den vielen philosophischen Fragen zum Thema.

Das gesamte Projekt sowie das Konzept sind öffentlich und stehen der Allgemeinheit frei zur Verfügung. Die Dokumentation auf *Github* wird auf Englisch geführt, aus eigenem Interesse sowie auf Wunsch des *AEC*.

2.1.4. Projektplanung und Meilensteine

In eigenem Interesse und zugunsten des *AEC* wird das Projekt öffentlich auf [Github](#) in dem auf dem Titelblatt angeführten [Repository](#) geführt.

The screenshot shows the GitHub repository page for 'DrAI'. At the top, there are buttons for 'Unpin', 'Unwatch', 'Fork', and 'Star'. Below the header, there's a search bar and a 'Code' button. The main area displays a list of commits from 'SamuelNoesslboeck' with details like date and message. To the right, there are sections for 'About', 'Releases', and 'Contributors'.

About
Diploma project of Samuel Nösslböck and Rene Schwarz started in 2023
robot ai draw exhibition ars-electronica

Releases 1
Phase I - First version (Latest) on Oct 2, 2023

Contributors 2
SamuelNoesslboeck Samuel Nössl böck
SchwarzRene

Commit	Message	Date
SamuelNoesslboeck sync	new images	2 weeks ago
code	new images	2 weeks ago
construction	new comp	4 days ago
documentation	sync	3 hours ago
drake_printer_files @ a1731e9	sync	3 hours ago
electronics	improved docs	3 months ago
export	added export folder	9 months ago
sketches	new images	2 weeks ago
standard	updated covery	4 months ago
.gitignore	better documentation	2 weeks ago
.gitmodules	updated submodules	4 months ago
LICENSE-CERN	Update licenses	6 months ago
LICENSE-MIT	Update licenses	6 months ago
README.de.md	sync	3 weeks ago
README.md	better documentation	2 weeks ago
a1_drake_without_cover.asm	Hole distance of H8-Lager changed	4 months ago
a_drake.asm	merging branches for current cad status	4 months ago
a_drake.cfg	merging branches for current cad status	4 months ago

Abbildung 2.1.4.1 Github-Website

Des weiteren wurden folgende Meilensteine definiert:

- 1. Planung:** Das Grundkonzept sollte gut durchdacht und dadurch mögliche Fehler gleich am Anfang vermieden werden. Aufgrund der hohen Komplexität des Projektes, besonders in der Software, wurde hier viel Zeit eingeplant.
- 2. Bau:** Der Bau und die Fertigung des Roboters sind durch die hohe Anzahl der 3D-Druckteile eher eine anspruchsvolle Aufgabe für gute Versionierung. Unsere Fertigungstoleranzen sind hoch, was uns zeitlich viel Spielraum gibt.
- 3. Kombination:** Bis zu dieser Phase waren die beiden Teile Software und Hardware nahezu völlig getrennt, was sich in dieser Phase drastisch ändert. Jetzt wird alles kombiniert und fertiggestellt, um erste Tests durchzuführen.
- 4. Tests:** In der Testphase werden mögliche Fehler und Probleme ausgeglichen und behoben. Am Ende dieser Phase ist der Roboter fertig und bereit für die Ausstellung.

2.1.5. Zeitplan und derzeitiger Stand

Der Zeitplan wird ebenso über ein *Github*-Projekt organisiert. Um diesen grob zu veranschaulichen, dient die folgende Grafik:



Abbildung 2.1.5.1 Zeitplan

Der derzeitige Stand ist, sehr kurz zusammengefasst, folgender:

Roboter:

Derzeit befindet sich der Roboter am Ende der Bauphase. Der Aluminiumrahmen des Roboters ist bereits gefertigt, jedoch mussten starke zeitliche Verzögerungen in Kauf genommen werden. Die komplette Mechanik ist fertiggestellt sowie nahezu alle Designelemente.

Die Elektronik und Steuerungssoftware wird nach und nach integriert, erste Bewegungen sind bereits möglich.

Software:

Die Künstliche Intelligenz ist vollständig programmiert und entspricht den Anforderungen unseres Auftraggebers. Da sich die Entwicklungszeit beschränkt, wurde derzeit noch kein Parameter-Finetuning in Erwägung gezogen. Dies würde es ermöglichen, die Kreativität der Künstlichen Intelligenz noch weiter zu entwickeln und zu verbessern.

2.2. Inhaltliche Beschreibung

2.2.1. Finales Konzept und Aufgabe

Die Aufgabe des Projektes ist es, den Besucher*Innen im *Ars Electronica Center* Künstliche Intelligenz näherzubringen. Hierzu wurde final folgender Ablauf definiert:

1. Der User zeichnet eine Zeichnung mit einem Stift auf ein Blatt Papier
2. Der Roboter erstellt mithilfe einer Kamera eine digitale Version davon
3. Die Künstliche Intelligenz erkennt, was gezeichnet wurde, und fügt kreativ Dinge hinzu
4. Das Generierte wird zu einer Liste an Linien konvertiert und diese werden vom Roboter gezeichnet
5. Dieser Prozess kann nach Belieben wiederholt werden, beide Parteien können immer weiter Dinge hinzufügen

Dies führt dazu, dass der*die Besucher*In in den Prozess integriert und mit der Künstlichen Intelligenz gemeinsam eine Zeichnung entwickelt. Die Kreativität beider Parteien soll in den Vordergrund gerückt werden.

Unser Projekt wurde in zwei Teilbereiche unterteilt: Roboter (Hardware) und Künstliche Intelligenz (Software). Hierbei ist die Künstliche Intelligenz die wesentliche Innovation des Projektes, weshalb besonderer Fokus auf diesen Teil gelegt wurde.

2.2.2. Roboter

Die Hauptaufgabe des Roboters ist es, das digitale Bild auf ein Blatt Papier zu transferieren und eine möglichst einfache, interaktive Kommunikation mit dem*der Nutzer*in zu ermöglichen. Zusätzlich soll dieser Zeichenprozess möglichst schnell erfolgen, um die Aufmerksamkeitsspanne der Benutzer*innen nicht zu sehr auszureizen.

2.2.2.1. Design

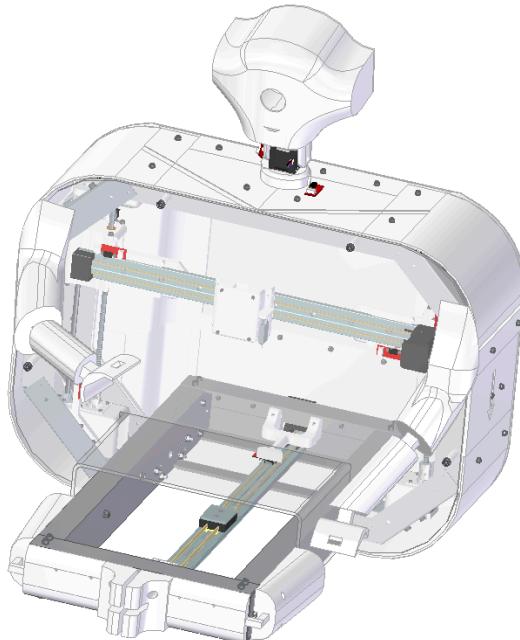


Abbildung 2.2.2.1 Fertiggestellter Roboter im CAD

Das Design des Roboters soll niedlich und vertraut wirken. Der Roboter fertigt die Bilder mithilfe der Kamera in seiner linken Hand an, zeichnet in seinem Bauch und soll über einen Kopf verfügen, der den*die Benutzer*In während des Prozesses anschauen soll. Um mehr Interaktion mit dem*der Besucher*In zu etablieren.

2.2.2.2. Interaktion mit dem*der Benutzer*In

Unser Auftraggeber hat uns klare Rahmenbedingungen, was die Kommunikation mit dem*der User*in betrifft, gegeben. Die benötigten Eingabemöglichkeiten sollten auf ein Minimum reduziert werden, weshalb ein einziger Knopf zum Starten des Prozesses verwendet wird. Ansonsten werden keine weiteren Eingabemöglichkeiten benötigt.

2.2.2.3. Mechanik

Um die Achsen in X- und Y-Richtung bewegen zu können, wurden maßgefertigte Achsen der Firma *Igus* verbaut. Dies ermöglicht schnelles Einbauen und eine genaue Fertigung. Auch erspart es uns Zeit und ist weniger Konstruktionsaufwand.

Um die Z-Achse anzusteuern, wurden Rundführungen mit Gewindespindeln kombiniert. Hierbei ist die Z-Achse links und rechts gelagert und durch den beidseitigen Antrieb ist es möglich, Fertigungsungenauigkeiten auszugleichen.

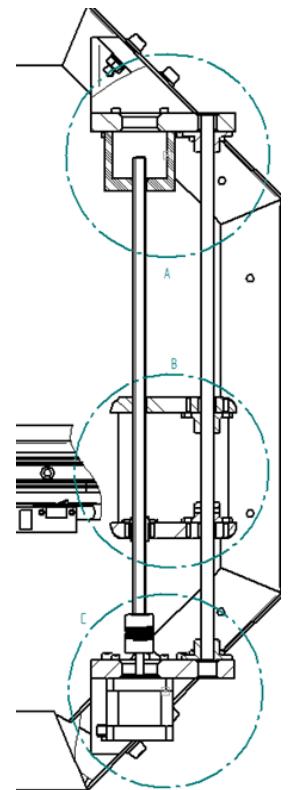


Abbildung 2.2.2.4.1
Mechanik

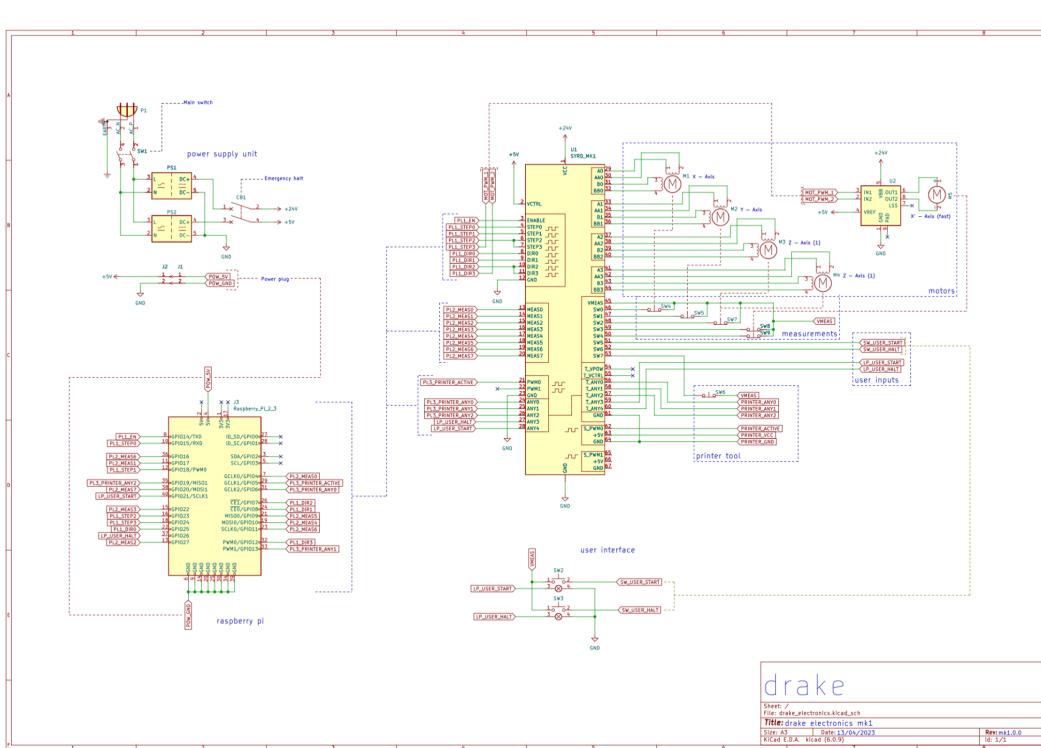


Abbildung 2.2.2.4.2 Elektronik

Die Elektronik wird durch einen Raspberry Pi gesteuert, der mit der Kamera und anderen IOT Geräten verbunden ist. Um hohe Geschwindigkeiten während des Zeichenprozesses zu ermöglichen, wurden Hochleistungs-Steppercontroller eingebaut und für die Referenzierung wurden an den Achsen Endschalter verbaut.

2.2.2.5. Steuerungssoftware

Die Steuerungssoftware ist vollständig in der Systemprogrammiersprache *Rust*, eine eher neue am Markt, programmiert. *Rust* ist durch ihre enorme Effizienz und Stabilität vor allem in der hardwarenahen Programmierung immer weiter verbreitet.

2.2.3. Künstliche Intelligenz und Software

Das Ziel der Künstlichen Intelligenz besteht darin, das existierende Bild zu erkennen, zu interpretieren und schlussendlich einen kreativen Beitrag zu leisten. Wie bereits im Kapitel 2.2.1. erwähnt, ist der Prozess zeitlich etwas begrenzt und sollte deshalb effizient und schnell sein.

2.2.3.1. Einzigartigkeit

Bestehende Algorithmen erzielen herausragende Resultate, wenn es darum geht, aus einer Textbeschreibung ein Bild oder Text zu generieren. Der *DrAI* Algorithmus ermöglicht es, dass diese ursprüngliche Textbeschreibung nicht zwingend notwendig ist, um weiteres Material zu erstellen. Es ist einerseits möglich, auf einer “leeren Leinwand” etwas zu erzeugen, andererseits kann, basierend auf einem bestehenden Bild, etwas generiert werden. Algorithmen dieser Art könnten auch zur eigenen Inspiration dienen, indem andere Denkweisen der KI offenbart werden. In manchen Fällen ist es eine Herausforderung, die Kreativität und Einzigartigkeit der Bilder zu erkennen, bei eindeutigen Skizzen des Users oder der Userin fällt jedoch das Ergebnis klar und deutlich aus.

2.2.3.2. Parametrisierung

Mithilfe diverser Parameter lässt sich effektiv in den Algorithmus eingreifen. Vereinfacht gesagt gibt es Parameter wie den Grad der Kreativität, Abstraktheit, Menge, Dichte und viele weitere.

Die Parameter wurden vorerst dem Auftraggeber entsprechend angepasst, jedoch können sie jederzeit abgeändert werden. Im Konzept des Algorithmus sind auch Möglichkeiten wie dynamische Parametrisierung enthalten, das heißt Anpassung bestimmter Parameter wie zum Beispiel Menge und Dichte des zu Generierenden auf Basis der bereits vorhandenen Menge in einem bestimmten Bereich des Bildes.

2.2.3.3. Ergebnisse

In den folgenden Abbildungen sind Tests und Experimente mit dem *DrAI* Algorithmus dargestellt. Das obere Bild ist von uns oder Freund*innen digital angefertigt worden, das jeweils untere das Ergebnis nach der Verarbeitung durch den Algorithmus.

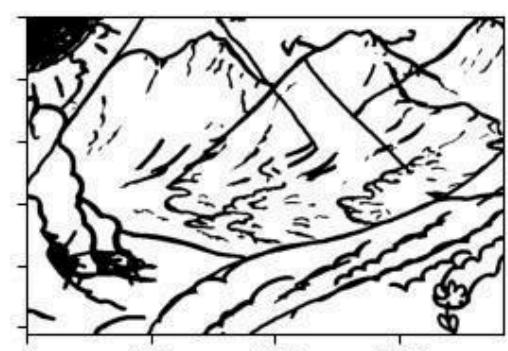
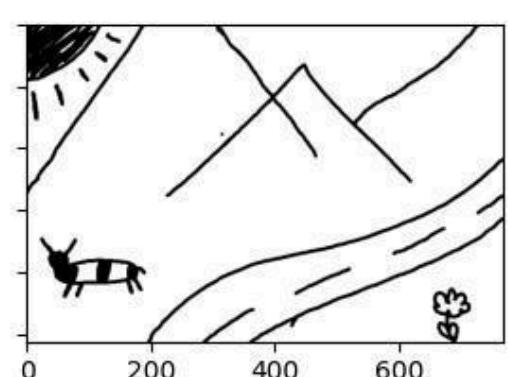
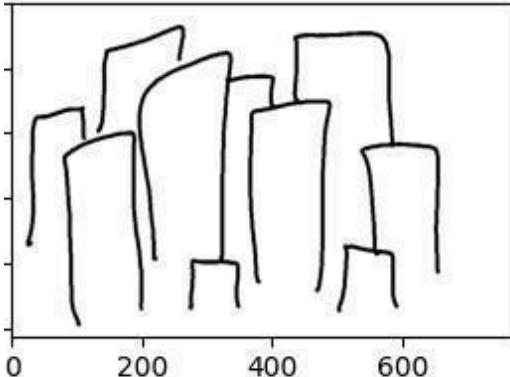
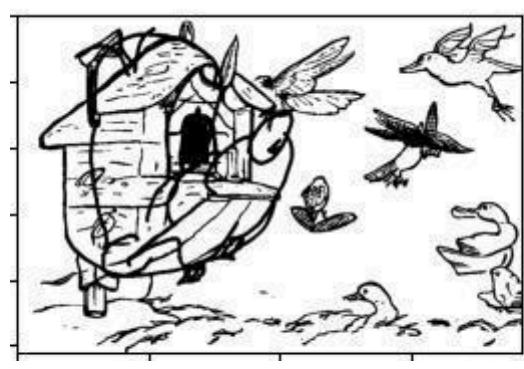
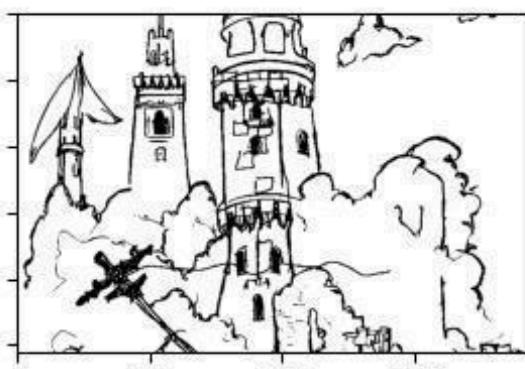
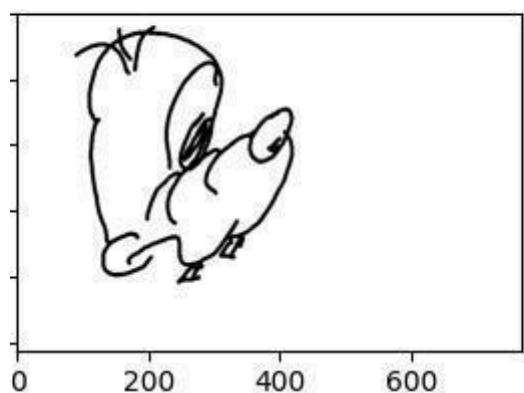
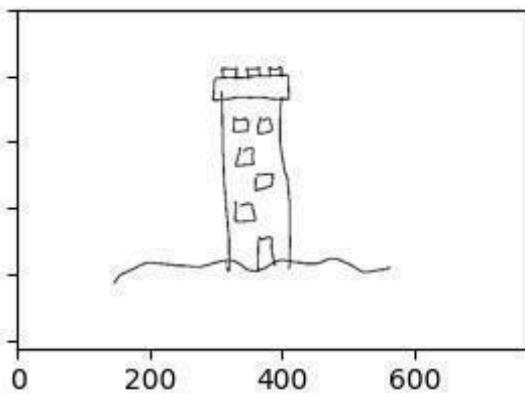


Abbildung 2.2.3.3 Ergebnisse

2.2.3.4. Entstehung / Experimente

Die nachfolgenden Kapitel behandeln bestehende Algorithmen und schlussendlich die verwendete Kombination und Verbesserung von ihnen für den benötigten Anwendungsfall.

2.2.3.4.1. Eigenes Neuronales Netzwerk

Da in diesem Anwendungsfall keine Flächen mit Farbe bemalt werden sollten, wurde die Idee geboren, ein eigenes *Generative Adversarial Network* (Wikipedia, 2017) zu trainieren. Dies ist eines der ersten vielversprechenden Konzepte im Bereich Generative AI. Hierbei werden zwei Netzwerke trainiert: der Generator und der Discriminator. Der Generator erzeugt aus einem Random Noise oder einem Eingangsbild ein modifiziertes Ausgangsbild, der Discriminator hingegen erhält als Eingangsparameter ein echtes oder ein generiertes Bild und wird darauf trainiert, zu erkennen, ob das Eingangsbild echt oder generiert ist. Im besten Fall lernt der Generator, realitätsnahe Bilder zu erzeugen, damit ein Discriminator nicht mehr zwischen real und generiert unterscheiden kann.

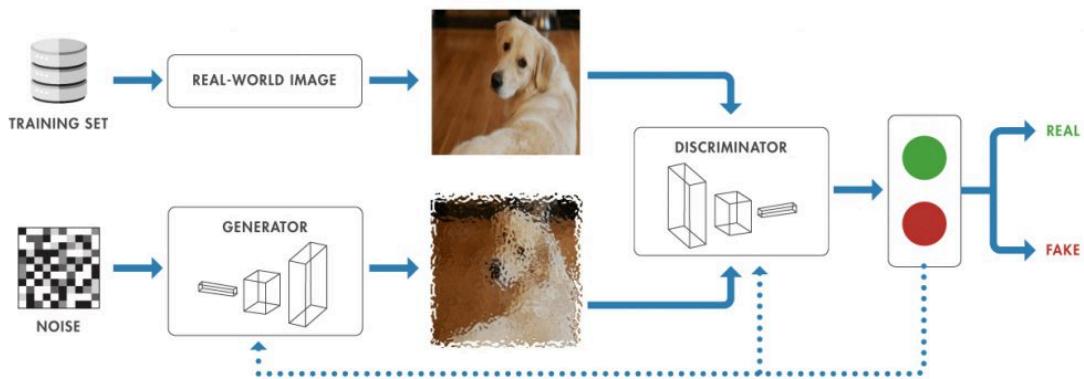
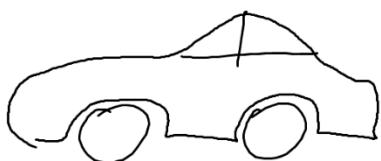


Abbildung 2.2.3.4.1 GAN-Architektur

Da nur beschränkt Rechenleistung zur Verfügung steht, wurde vereinbart, nur Landschaftsbilder als Trainingsdaten zu verwenden. Die Eingaben des Benutzers oder der Benutzerin werden als reine Umrisse betrachtet. Um die Trainingsdaten der Eingangsdaten des Benutzers / der Benutzerin anzupassen, wurden aus den Landschaftsbildern die Kanten mithilfe des *Holistically-Nested Edge Detection* (Saining Xie, Zhuowen Tu, 2015) Algorithmuses Kanten extrahiert.

Da es die Aufgabe des Roboters ist, Konturen zur Originalzeichnung hinzuzufügen, wurden Bildbereiche / Linien aus dem Originalbild entfernt. Der Generator sollte lernen, die entfernten Linien zu rekonstruieren. Um nun die menschliche Kreativität nachzuahmen, werden zufällig normalverteilte Werte dem Bild hinzugefügt.



Originalbild (Output für den User)



Eingangsbild (Input KI)

Abbildung 2.2.3.4.2 GAN-Trainingsdaten

Die Vorteile dieses Algorithmuses wären:

- Geringer Rechenaufwand im Betrieb
- Gute *Parametrisierung*, besonders im benötigten Anwendungsfall

Die Nachteile jedoch wären:

- Hoher rechenaufwand in der Trainingsphase
- Aufwendige Beschaffung von Daten
- Geringer Grad an Kreativität
- Hohe Spezialisierung notwendig (z.B. nur Landschaftsbilder)

Die Resultate erwiesen sich als wenig vielversprechend. Es war kein Hauch von Kreativität zu erkennen und der Algorithmus drohte, rein schwarze Bilder zu generieren, da der Discriminator zu gut erkannte, welches Bild echt und welches generiert wurde. Dies führte dazu, dass der Generator nicht mehr effizient lernen konnte und sich die Ergebnisse wenig bis kaum verbesserten.

2.2.3.4.2. StablePipe

2.2.3.4.2.1. Stable Diffusion

Um die Probleme eines selbst trainierten Algorithmus zu lösen, wurde *Stable Diffusion* benutzt, ein von *Stability AI* vor trainiertes Netzwerk. Dieser Algorithmus benutzt Text, um Bilder zu erstellen. Hierbei wird wie bei einem *Generative Adversarial Network* aus einem zufällig normalverteilten Bild ein für uns menschliches Bild erzeugt. Der Algorithmus wird trainiert indem zu einem für das menschliche Auge sinnhaften Bild normalverteilte Werte hinzugefügt werden, sodass das Bild über Zeit an Sinnhaftigkeit verliert. Ziel des Netzwerkes ist es nun, über den Verlauf der Zeit den *Diffusion-Prozess* umzukehren und basierend auf dem diffundierten Bild und dem Text Prompt das Originalbild wiederherzustellen.

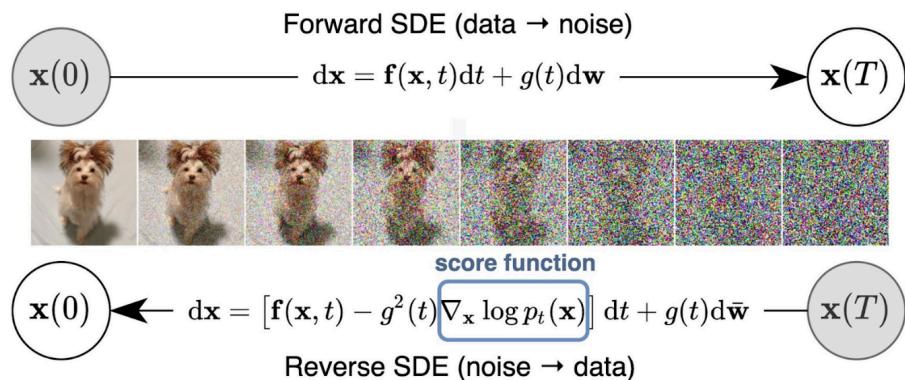


Abbildung 2.2.3.4.2.1 Stable Diffusion Prozess

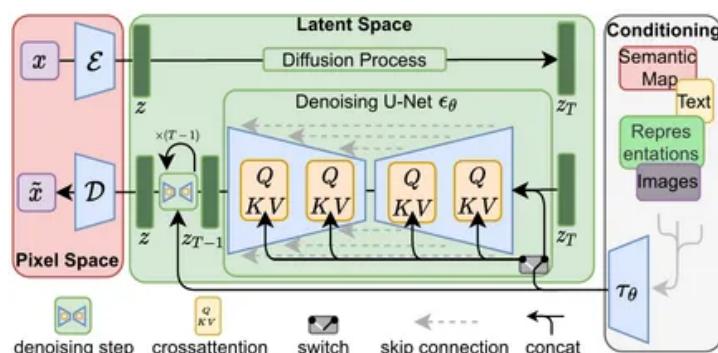


Abbildung 2.2.3.4.2.2 Stable Diffusion Architecture

Die erste Version dieses Netzwerkes wurde auf dem *Laion2B-en* (Laion, n.d.) Datensatz trainiert. Dies ermöglichte es dem Netzwerk eine Vielzahl an Bildern zu konstruieren und viele Bereiche, wie zum Beispiel Landschaften, Städte und vieles mehr, basierend auf Textbeschreibungen zu rekonstruieren. Je einfallsreicher die Textbeschreibung, desto einfallsreicher das generierte Bild.

2.2.3.4.2.1. StablePipe Pipeline

Mithilfe des *Stable Diffusion* Algorithmus, beschrieben im Absatz 2.2.3.4.2.1, ist es möglich, aus einem bestehenden Bild und einer Textbeschreibung ein neues Bild zu generieren. Die *Stable Pipeline* malt zusätzlich zufällig generierte Linien auf das Originalbild. Durch den

Diffusionsprozess sollten die Linien verschwinden und nach dem *Forward Process* würden danach komplexe Zeichnungen entstehen. Die Resultate waren weniger vielversprechend, da der Diffusion Prozess keine konkrete Textbeschreibung besaß und so keine komplexen Formen entstanden.

2.2.3.4.3. InterStablePipe

2.2.3.4.3.1. Clip-Interrogator

Da der *Stable Diffusion Algorithmus* (Absatz 2.2.3.4.2.1) einen Text Prompt benötigt um kreative Bilder zu erzeugen und um zu wissen, wovon das Bild handelt, wird ein *Interrogator* (Pharmapsychotic, n.d.), benutzt, um aus dem vom Benutzer gezeichnetem Bild eine Beschreibung zu erhalten.

2.2.3.4.3.2. Line Extraction Software (LES)

Ein digitales Bild kann auf verschiedenste Weisen repräsentiert werden. Die bekanntesten Darstellungen sind *RGB*, *HSV* oder *Grayscale*, ersteres wird erzeugt von *Stable Diffusion*. Die *RGB*-Darstellung setzt sich aus einem roten, einem grünen und einem blauen Kanal zusammen und ergibt ein für das menschliche Auge in Farbe sichtbares Bild. Da nur Linien aus dem Bild extrahiert werden, wird es zu schwarz-weiß konvertiert. Jedes Pixel wird in dieser Darstellung entweder durch den Wert 0 (schwarz) oder 1 (weiß) repräsentiert. Im ersten Schritt wird ein Bild erstellt, welches die Pixel darstellt, die im Originalbild nicht bemalt wurden, im generierten Bild jedoch schwarz sind. Der Algorithmus extrahiert mithilfe des *Canny-Edge-Detection-Algorithmus* nun Kanten aus der Grafik. Diese werden nun zu Polygone mit Hilfe der *OpenCV2* Bibliothek verbunden, welche dann, mit einer definierten Stiftgröße in Pixel, in das Originalbild gezeichnet werden. Dieser Prozess wiederholt sich so oft, bis alle Pixel, die der *Stable Diffusion* Algorithmus generiert hat und im Originalbild nicht gezeichnet waren, gezeichnet wurden. Das Output wird dann an die Steuerungssoftware in der Form einer langen, meistens mit mehreren zehntausenden Einträgen, Liste an Linien weitergegeben.

Der Zeichenprozess wird dann optimiert, indem der kürzest mögliche Weg zwischen Linien kalkuliert wird und die rekursive Struktur ermöglicht sogar die effiziente Bemalung von Flächen.

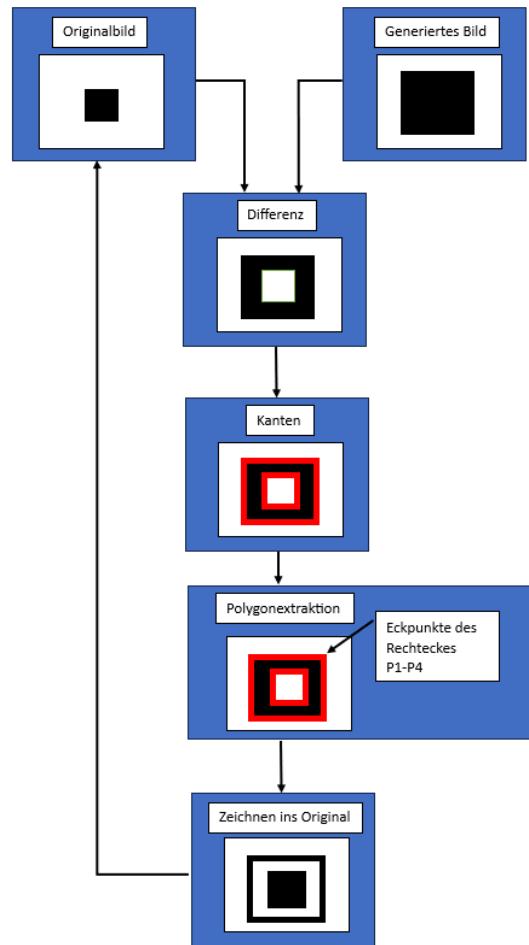


Abbildung 2.2.3.4.3.2.1 Line Extraction Software

2.2.3.4.3.3 InterStablePipe Pipeline

Um zu einer neuen Zeichnung zu kommen wird zuerst aus dem Bild eine Beschreibung mittels des *Interrogators* (Absatz 2.2.3.4.3.1) erzeugt. Diese wird dann in den *Stable Diffusion* Algorithmus (Absatz 2.2.3.4.2.1) gegeben und ein neues Bild wird generiert. Aus diesem werden nun Linien mittels der LES (Absatz 2.2.3.4.3.2) extrahiert, welche dann an den Roboter gesendet werden können.

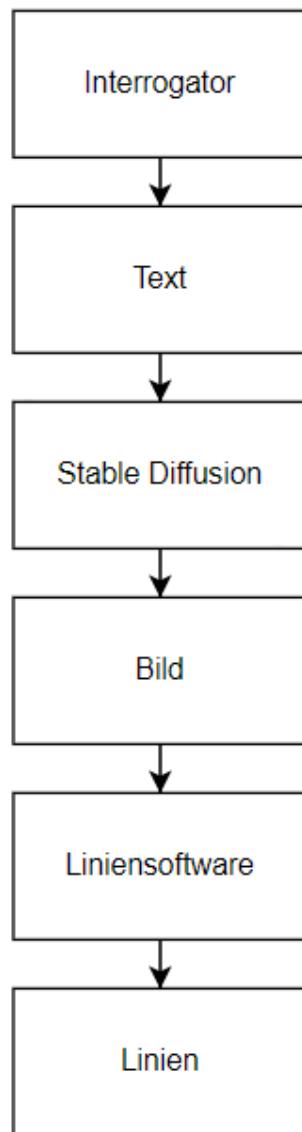


Abbildung 2.2.3.4.3.3 InterStablePipe Pipeline

Dieser Algorithmus hat zwei Hauptprobleme:

1. Es wird ein neues Bild generiert, welches mit dem Thema des Benutzers übereinstimmt, jedoch verschwinden die originalen Linien, welche initial gezeichnet
2. Der Algorithmus überlegt sich nichts Neues, was er hinzufügen kann

2.2.3.4.4 InterStableCloud

2.2.3.4.4.1 Wordcloud

Im Bereich des *Natural Language Processing* werden Wörter in Vektoren umgewandelt, damit mathematische Operationen mit ihnen durchgeführt werden können. Ein Effekt der dann auftritt ist, dass wenn Wörter sich ähneln, es auch ihre Vektoren tun. Ein Musterbeispiel der *Wordcloud* (Wikipedia, 2024) ist, wenn der Vektor des Wortes "King" mit dem Vektor des Wortes von "Women" addiert wird, so kommt der Vektor von "Queen" heraus. Um aus Wörtern Vektoren erzeugen zu können, wird ein Wort in einen *One-Hot Vektor* verwandelt. Ein *One-Hot Vektor* besteht vollständig aus Null-Werten bis auf eine Eins, deren Index die Position in einer Liste aus allen Wörtern des Trainingsmaterials repräsentiert.

Trainiert wird dieser Algorithmus, indem der *One-Hot Vektor* mit den Spalten "n" (n = Anzahl an bekannten Wörtern in den Trainingsdaten) mit einer zu trainierenden Matrix "n" x "m" multipliziert wird, "m" ist die hierbei Anzahl an Vektor Spalten der *Wortvektoren*. Nun werden "t" Wörter vor und nach dem Wort in Vektoren verwandelt und miteinander addiert. Als Summe sollte, wie in Abbildung 2.2.3.4.4.1 illustriert, das Original herauskommen,

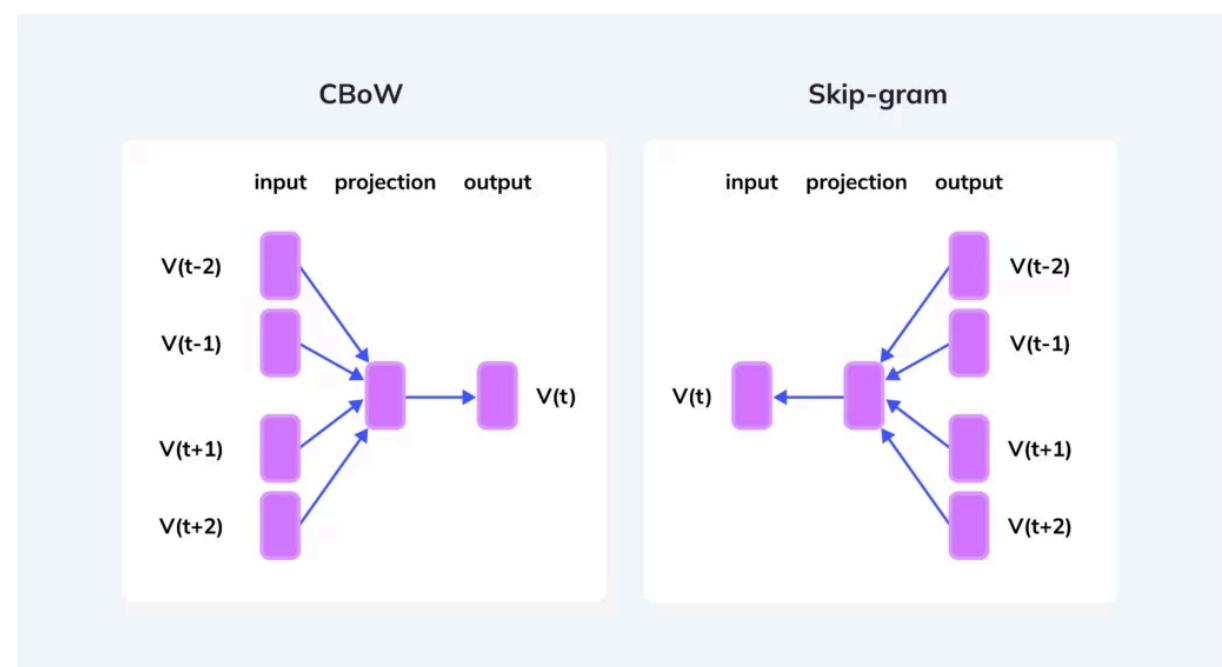


Abbildung 2.2.3.4.4.1 Word Cloud-Training

2.2.3.4.4.2 InterStableCloud Pipeline

Die *InterStableCloud Pipeline* ist eine Erweiterung der im Absatz 2.2.3.4.3 beschriebenen *InterStable Pipeline*. Die Probleme, die in 2.2.3.4.3.3 beschrieben werden, werden durch zwei Änderungen teilweise behoben.

1. Änderung bei *WordCloud* (Absatz 2.2.3.4.4.1)

Um das Problem der Kreativität zu beheben, wird der Text aus dem Bild mit Hilfe des *Interrogators* extrahiert. Nun werden aus dem generierten Text die Hauptwörter extrahiert. Zu jedem Hauptwort werden nun “n” ähnliche Wörter in der Wordcloud gesucht. Hierbei wird das Hauptwort in einen Vektor verwandelt. Nun kann zu jedem bekannten Wort die Vektor Distanz berechnet werden. Um nun ein ähnliches Wort zu bekommen, wird ein zufälliges Wort ausgewählt, welches im Rahmen einer definierten Vektor-Distanz liegt. Dies ermöglicht es, zum Beispiel aus dem Hauptwort “Haus”, Wörter wie “Garten”, “Türe”, oder “Fenster” zu erhalten. Durch die zufällige Auswahl wird jedesmal etwas anderes gezeichnet und je kleiner die Vektor Distanz, desto mehr hat das Wort mit dem Hauptwort zu tun. Zu guter Letzt werden die Wörter zur Textbeschreibung des Originalbildes hinzugefügt.

2. Änderung bei *Img2Img*

Da die in Absatz 2.2.3.4.3.3 beschriebene Pipeline das Problem aufweist, das Originalbild nicht zu berücksichtigen, dient nicht nur die neu generierte Beschreibung als Input für den *Stable Diffusion* Algorithmus, sondern auch das Originalbild. Nun können durch *Strength* und *Prompt Guidance* die neu generierten Bilder kontrolliert werden. Je höher die beiden Werte sind, desto kreativer wird das Ausgangsbild. Der Effekt dieser beiden Werte ist illustriert in Abbildung 2.2.3.4.4.2.1. Zu sehen ist, dass durch Steigerung der beiden Werte mehr zum Originalbild generiert wird.

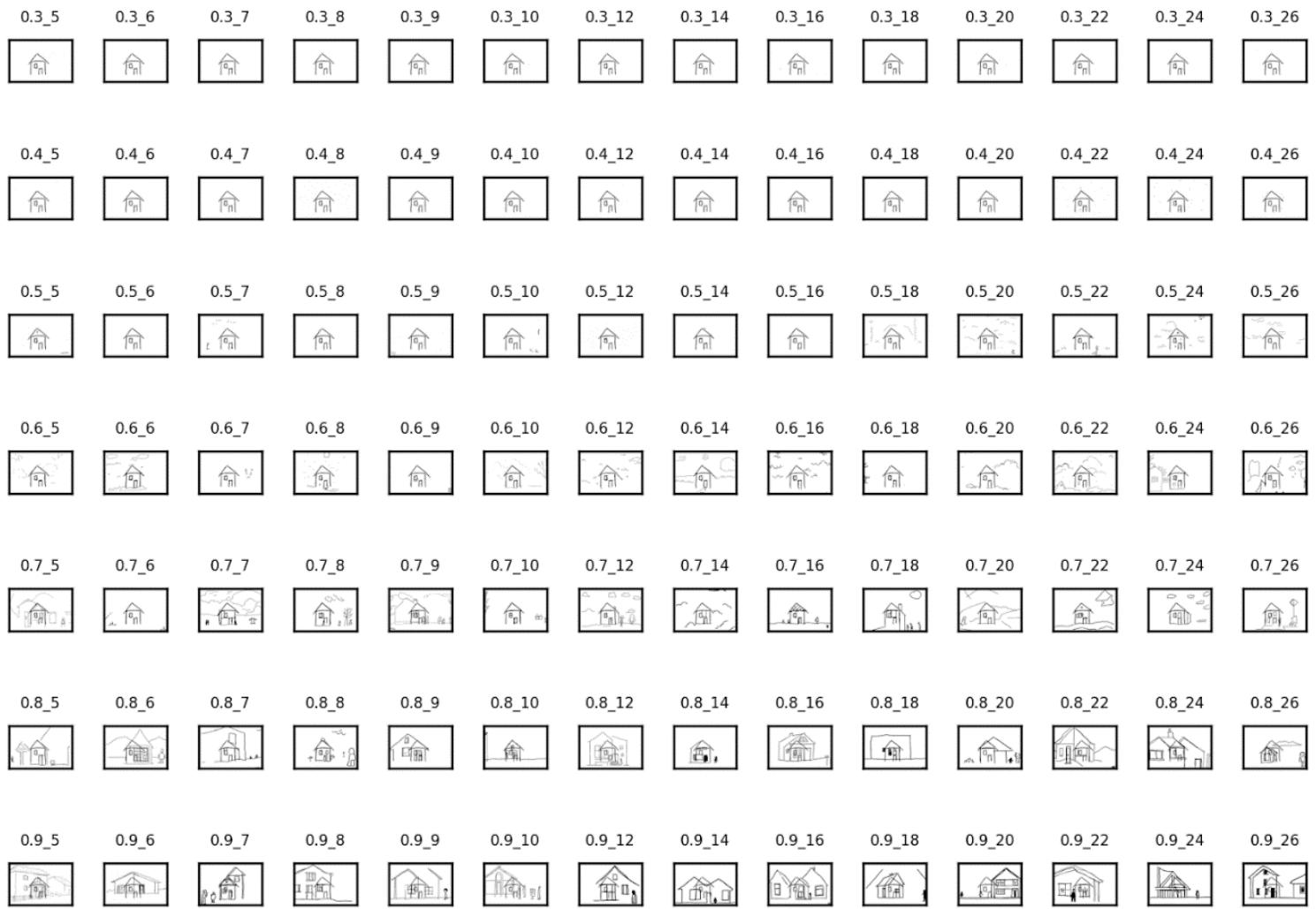


Abbildung 2.2.3.4.4.2.1 Guidance Scale

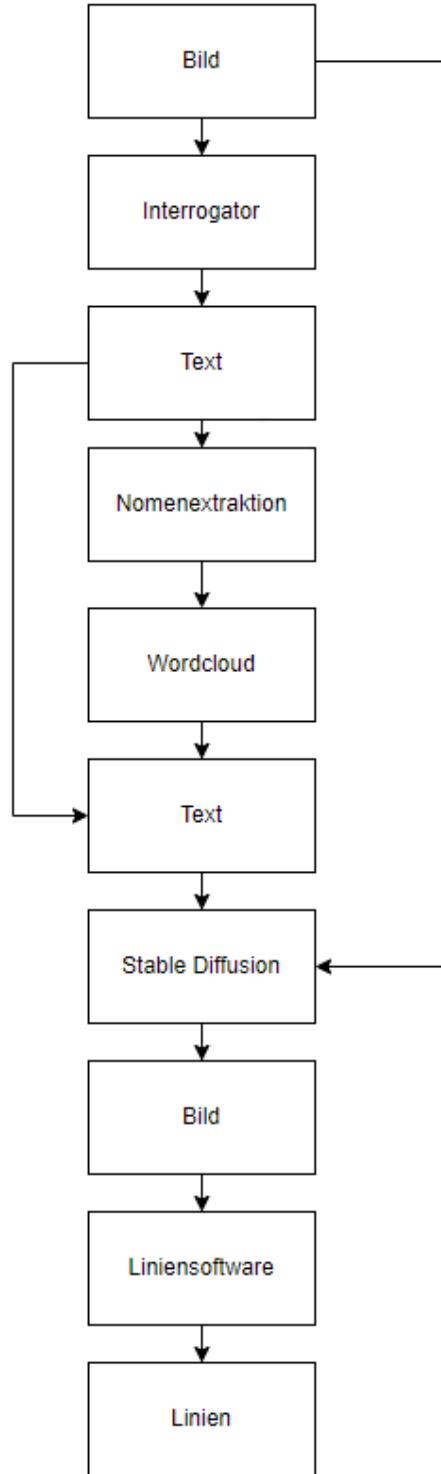


Abbildung 2.2.3.4.4.2.2 InterStableCloud Pipeline

Der Algorithmus liefert erste brauchbare Resultate, jedoch entsprechen die Resultate nicht dem gesetzten Ziel, da nur manche Bilder das Maß an Kreativität aufweisen, das vom Auftraggeber gewünscht wurde.

2.2.3.4.5 InterStableLLM Pipeline

2.2.3.4.5.1 Motivation

Bekommt ein Mensch die Aufgabe, ein Bild kreativ zu erweitern, so erfolgt dies in mehreren Schritten:

1. Er erkennt wovon das Bild handelt
2. Er überlegt sich, was hinzufügen werden könnte, oft durch zufällige Impulse in der Wahrnehmung
3. Er zeichnet dies auf das Papier mit dem Originalbild im Hinterkopf und verknüpft dabei die neuen Impulse durch Bekanntes, vertraute Bilder und Muster

Dieses Schema kann man dank neuester Technologie, welche zu Beginn dieser Experimente noch nicht im Trend lag, auch auf eine Maschine übertragen.

1. Ein *Interrogator* (Absatz 2.2.3.4.3.1) erkennt wovon das Bild handelt
2. Ein *Large Language Model* (Absatz 2.2.3.4.5.2) “überlegt” sich kreative Beschreibungen
3. *Stable Diffusion* zeichnet das Neue auf das Originalbild

2.2.3.4.5.2 Large Language Model

Ein *Large Language Model* (Wikipedia, 2024) versucht, das nächste Satzzeichen / Wort vorherzusagen. *Language Models* bauen derzeit auf den *Transformer* Algorithmus auf. Zuerst verwandelt ein *Large Language Model* Wörter in Vektoren, danach werden diese Vektoren durch eine Reihe von Attention Layer und Feed Forward Layer verändert. Diese Vektoren werden so lange trainiert, bis das *LLM* lernt, das nächste Wort mit einer gewissen Wahrscheinlichkeit vorherzusagen. Forscher haben herausgefunden, dass, wenn diese Netzwerke auf Milliarden von Wörtern trainiert werden, sie den Text verstehen und so Texte in menschlicher Qualität schreiben können. Die bekanntesten *LLMs* sind *ChatGPT*, *Palm*, *Llama*, *Mistral*, und viele weitere.

Ein Problem dieser Netzwerke ist, dass sie hohen Rechenaufwand aufweisen und meist sehr viel *VRAM* benötigen. Je nach Anzahl der Parameter, welche in großer Zahl die Qualität verbessern, wird mehr von beidem benötigt.

Durch *Quantisierung* (geringere Anzahl an Bits pro Zahl) ist es möglich, solche Netzwerke auf leistungsschwächeren Computern auszuführen. Hierbei wird jedoch Geschwindigkeit und Performance eingebüßt.

In unserem Fall wurde zu Beginn *Llama* (Meta AI, 2024) verwendet, um *Llama 7B* zu quantisieren und auf den uns zur Verfügung gestellten Rechner laufen zu lassen. Hierbei ist jedoch das Problem, dass *Llama* den Prompt evaluieren muss, bevor die Wahrscheinlichkeit des nächsten Wortes errechnet wird. Dies dauert pro Token, also ein einziges Satzzeichen, 200 Millisekunden. In unserem Fall dauert dieser Prozess je nach Bildbeschreibung zwischen 5-25

Sekunden. Des Weiteren wurde *LLama* hauptsächlich auf der CPU exekutiert, was wiederum den Prozess verlangsamt.

Am 27.09.23 wurde *Mistral 7B* (Mistral AI, 2024) veröffentlicht. Dieses Netzwerk zeigt bessere Performance wie *Llama-7b*. Des Weiteren ist es möglich, durch die Python Library *Hugging Face*, gewisse Layer des Netzwerkes auf die GPU zu verlagern und so den Prozess zu beschleunigen. Ein weiterer Vorteil gegenüber *Llama-7b* ist, dass keine Prompt Evaluierung benötigt wird.

2.2.3.4.5.3 InterStableLLM Pipeline

Um nun noch kreative Textbeschreibungen zu bekommen, wird anstelle der in *InterStableCloud Pipeline* (Absatz 2.2.3.4.4.2) verwendeten *Wordcloud* ein *Large Language Model* verwendet. Hierbei ist die Textbeschreibung für das *LLM* wie folgt:

```
You are a very creative artist.  
Think of something extraordinary.  
Think of something what you can add to this image prompt to make  
it magic and special  
Think in a way that inspires people and which makes them happy  
Do not describe the style of the new image mention that no areas  
should be filled and the style is black and white linestyle  
Be very creative for example to a house you could add a garden or  
to a car a trailer  
You can even draw a truck out of a car  
The description should be something which makes sense  
The image description is
```

Durch die Kombination aus verschiedenen Netzwerken ist es nun möglich, kreative Textbeschreibungen zu erhalten, jedoch zeichnet der *Stable Diffusion* Algorithmus nur dort, wo der Benutzer gezeichnet hat. Zeichnet der Besucher nur auf einer Hälfte des Papiers, so zeichnet auch der Algorithmus nur auf dieser Seite.

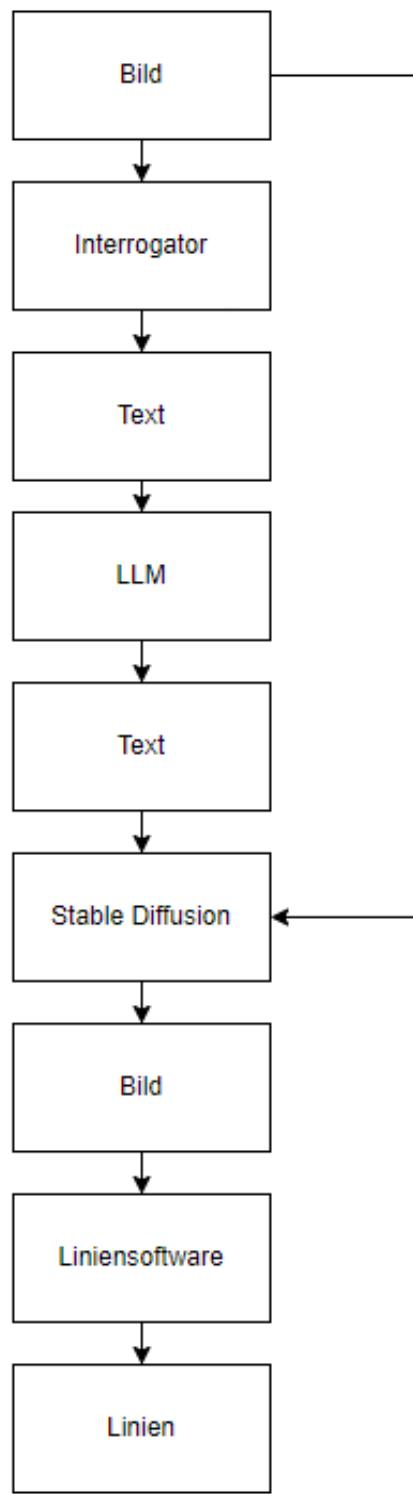


Abbildung 2.2.3.4.5.3 InterStableLLM Pipeline

2.2.3.4.6 Finale Umsetzung, InterStableLLMRLLine

2.2.3.4.6.1 R-Line

Die *InterStableLLM Pipeline* erfüllt bis auf das eher monotone Ergebnis, beschrieben in Absatz 2.2.3.4.5.3 alle Kriterien. *InterStableLLMRLLine* versucht die menschliche Kreativität nachzuahmen, indem er generative Künstliche Intelligenz mit zufällig in das Eingangsbild generierten Inhalten kombiniert. Die Innovation liegt in dieser Interpretation menschlicher Kreativität als genau dieser Kombination von zufälligen Impulsen in der menschlichen Wahrnehmung, abgebildet durch die Zufallsgeneratoren, und der Erkennung von Mustern, abgebildet durch KI.

2.2.3.4.6.2 InterStableLLMRLLine Pipeline

Die *InterStableLLMRLLine Pipeline* basiert auf dem gleichen Prinzip wie die *InterStableLLM Pipeline* beschrieben in Absatz 2.2.3.4.5.3 nur wird hierbei das Originalbild durch den *R-Line Algorithmus* beschrieben im Absatz 2.2.3.4.7.1 verändert. Zu sehen ist die Pipeline in der Abbildung 2.2.3.4.7.2. Sie kann noch durch *Hyperparameter Tuning* und *Prompt-Engineering* verbessert werden, jedoch liefert die derzeitige Version bereits angemessene Resultate.

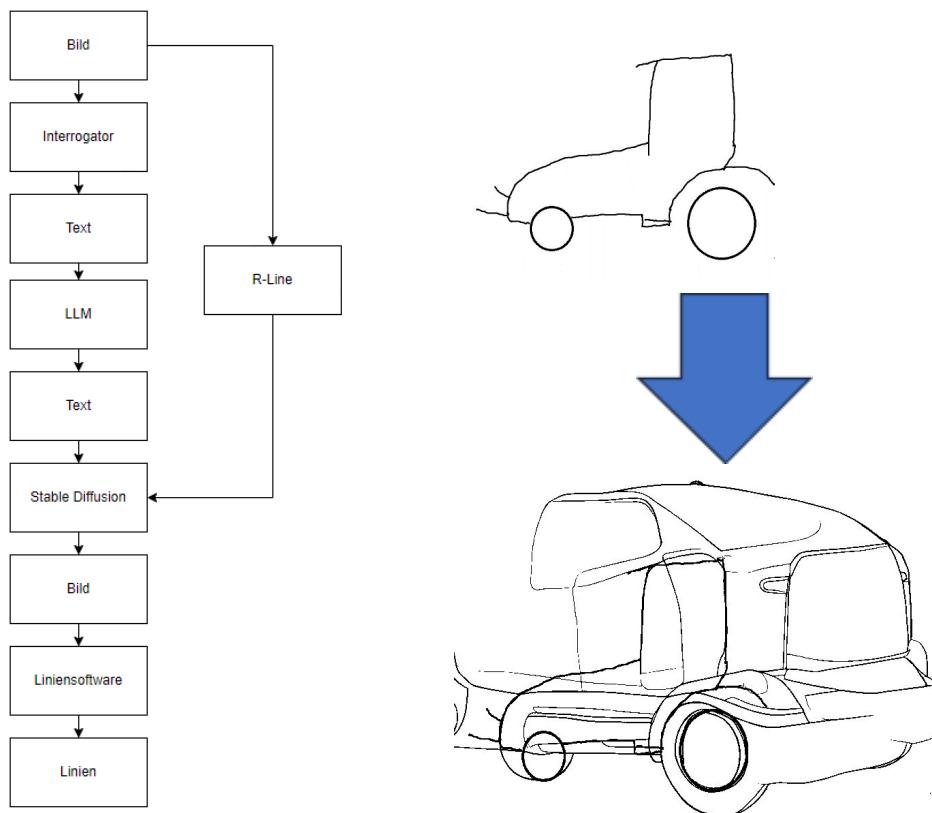


Abbildung 2.2.3.4.7.2 InterStableLLMRLLine Pipeline

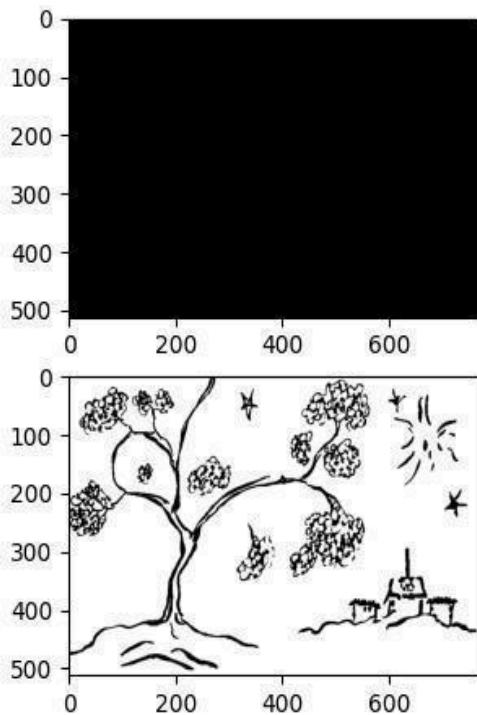
2.2.4. Ausblick, Entwicklungspotential und Wirtschaftlichkeit

Durch den Projektpartner und die Positionierung unseres Projektes wird die Erkenntnis, beschrieben im Absatz Neuheitsgrad, an viele Menschen weitergegeben und es kann breite Masse inspiriert werden, dies ist genau das Ziel dieses Projektes.

Das Konzept lässt sich jedoch enorm erweitern und einsetzen:

Die Pipeline kann mit verschiedenen Formen von Medien gekoppelt und trainiert werden, das heißt, es muss nicht wie in dieser Anwendung eine Bild-zu-Bild Pipeline verwendet werden. Es kann auch Audio-zu-Audio, Video-zu-Video oder Kreuzungen daraus verwendet werden. Die KI wird überall ihren kreativen Beitrag leisten.

Auch könnte man die Software an Künstler oder Menschen, die kreative Ideen benötigen, verkaufen. Die Projektteilnehmer kennen Phasen im Leben, in denen man den menschlichen



Drang verspürt, etwas zu bauen, jedoch keine Idee hat, was umgesetzt werden könnte. Einen genialen Einfall erhält man meistens durch externe Inspiration. Zum Beispiel sieht man eine Spinne, so kann man diese als Inspiration für einen Roboter nutzen. Wäre jemand, wie ein Science-Fiction Drehbuchautor in eben der Situation, so könnte er sich ein paar Bilder generieren lassen, welche er als Aufenthaltsort in Filmen nutzen kann.

Es ist einerseits möglich, zu schon bestehende Bilder etwas hinzuzufügen, andererseits können auch ganz neue Bilder ohne irgendeinen menschlichen Einfluss entstehen. Die Software kann nicht nur von Science-Fiction-Drehbuchautoren genutzt werden. Es wäre nutzbar für jeden Menschen, der sich durch Bilder inspirieren lässt. Auch das repräsentative Bild dieses Projektes wurde durch unsere Software generiert.

Abbildung 2.2.6.1 White Picture

Würde man die Software weiterentwickeln, so könnte es zu dem Punkt kommen, an dem sie auch auf End-Devices wie Mobiltelefone übertragbar wäre und so könnte man sich jederzeit inspirieren lassen.

Jedoch kann man sich auch selbst mit der KI spielen, da es eine großartige Beschäftigung ist. Es wäre auch einsetzbar für Apps wie Snapchat. Hierbei könnten die Gesichter von Menschen auf die kreativste und unvorhersehbare Weise verändert werden. Die Software hat eine breite Anwendbarkeit. Einerseits zur Unterhaltung und andererseits, um Menschen zu unterstützen.

2.3. Projektkoordination

2.3.1. Kompetenzen des Teams

Wie schon im Absatz Ideenfindung beschrieben, haben wir eine Ausbildung im Bereich Automatisierungstechnik. Da der Wissensdurst der Projektentwickler durch diese Ausbildung nicht gestillt wird, beschäftigen wir uns auch in der Freizeit mit diversen Themen, die über das Schulwissen hinausragen.

Samuel Nösslböck hat sein Wissen im Bereich Roboterbau, Elektronik und Kinematik erweitert und auch Projekte wie einen eigenen [Roboterarm](#) oder diverse ferngesteuerte mobile Roboter umgesetzt.

Schwarz Rene hat sich in seiner Freizeit mit Künstlicher Intelligenz beschäftigt, da er schon immer Interesse am menschlichen Bewusstsein und Denken gezeigt hat. Auch hat er Projekte wie eine eigene Spracherkennung oder [Light-Weight Objekterkennung](#) umgesetzt.

Entsprechend der Kompetenzen wurden die Aufgaben folgendermaßen verteilt:

1. Samuel Nösslböck
 - a. *Steuerungssoftware und Elektronik*
 - b. *Entwicklung des ersten Konzeptes und Idee*
 - c. *Fertigung mit 3D-Druck*
 - d. *Finale Montage / Zusammenbau*
 - e. *Kommunikation nach Außen*
2. Schwarz Rene
 - a. *Entwicklung der KI*
 - b. *Design des digitalen Roboters*
 - c. *Diverse Fertigungen*
 - d. *Schriftliche Dokumentation*

2.3.2. Kooperation und Betreuung

Die effiziente Arbeitsteilung hat einen Großteil der benötigten Kooperation hinfällig gemacht, der Rest wurde durch Github ziemlich erleichtert. Andere Angelegenheiten wie zum Beispiel das Networking wurden durch eine frühe Einigung auf ein einheitliches Konzept, Definition der zu benützenden Schnittstellen, Ports etc. geregelt.

Die HTL Neufelden ist eine eher auf Maschinenbau spezialisierte Schule, weshalb die Betreuung im Bereich KI wenig unterstützend war. Auch der Projektpartner, das AEC, sieht uns als unabhängige Künstler, weshalb die Betreuung auf die Definition der Rahmenbedingungen hinausläuft.

2.3.3. Kosten

Die Kosten des Roboters belaufen sich auf rund 1500€ und werden vollständig von unserem Schulverein und dem *AEC* getragen.

Kosten

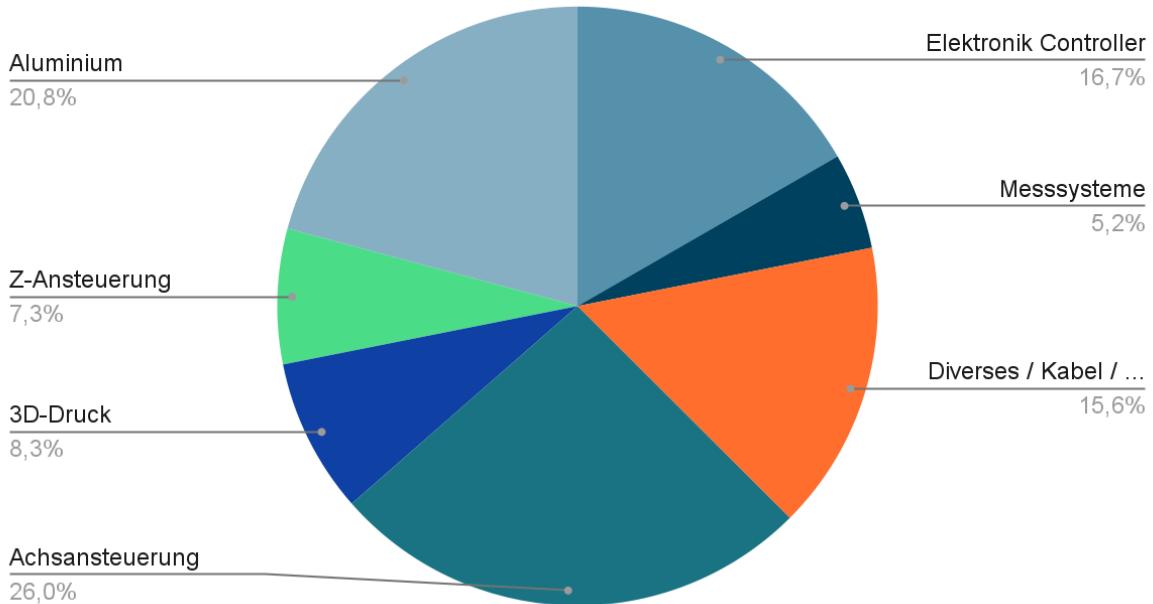


Abbildung 3.6 Kosten

3. Literaturverzeichnis

Stable diffusion online. (n.d.). <https://stablediffusionweb.com/>

Ai, M. (2024, January 31). *Mistral 7B*. Mistral AI | Open-weight Models. <https://mistral.ai/news/announcing-mistral-7b/>

Wikipedia contributors. (2024a, January 29). *LLAMA*. Wikipedia. <https://en.wikipedia.org/wiki/LLaMA>

Pharmapsychotic. (n.d.). *GitHub - pharmapsychotic/clip-interrogator: Image to prompt with BLIP and CLIP*. GitHub. <https://github.com/pharmapsychotic/clip-interrogator>

Welcome to Python.org. (2024, January 18). Python.org. <https://www.python.org/>

TensorFlow. (n.d.). *TensorFlow*. <https://www.tensorflow.org/>

NVIDIA CUDA toolkit - free tools and training. (n.d.). NVIDIA Developer. <https://developer.nvidia.com/cuda-toolkit>

Introduction | Langchain. (n.d.). https://python.langchain.com/docs/get_started/introduction

OpenCV. (2024, January 29). *OpenCV - Open Computer Vision Library*. <https://opencv.org/>

Hugging Face – The AI community building the future. (2023, December 15). <https://huggingface.co/>

Wikipedia contributors. (2023b, December 30). *Generative adversarial network*. Wikipedia. https://en.wikipedia.org/wiki/Generative_adversarial_network

LAION-5B: A NEW ERA OF OPEN LARGE-SCALE MULTI-MODAL DATASETS | LAION. (n.d.). <https://laion.ai/blog/laion-5b/>

Ahirwar, K. (2023, October 19). A very short introduction to diffusion models - Kailash Ahirwar - medium. *Medium*.

<https://kailashahirwar.medium.com/a-very-short-introduction-to-diffusion-models-a84235e4e9ae>

Wikipedia contributors. (2023, December 8). *Canny edge detector*. Wikipedia. https://en.wikipedia.org/wiki/Canny_edge_detector

Wikipedia contributors. (2024, January 30). *Word2Vec*. Wikipedia. <https://en.wikipedia.org/wiki/Word2vec>

Wikipedia contributors. (2024a, January 30). *Large language model*. Wikipedia. https://en.wikipedia.org/wiki/Large_language_model

Xie, S. (2015, April 24). *Holistically-Nested edge detection*. arXiv.org. <https://arxiv.org/abs/1504.06375>

Wikipedia contributors. (2024b, January 30). *Large language model*. Wikipedia.
https://en.wikipedia.org/wiki/Large_language_model

4. Bildverzeichnis

2.1.1.1 Erste Konzeptskizze.....	3
2.1.3.1 Skizze Octagon.....	5
2.1.4.1 Github-Website.....	6
2.1.5.1 Zeitplan.....	7
2.2.2.1 Fertiggestellter Roboter im CAD	9
2.2.2.4.1 Mechanik.....	10
2.2.2.4.2 Elektronik.....	10
2.2.3.3 Ergebnisse.....	12
2.2.3.4.1 GAN-Architektur.....	14
2.2.3.4.2 GAN-Trainingsdata.....	14
2.2.3.4.2.1 Stable Diffusion Prozess.....	16
2.2.3.4.2.2 Stable Diffusion Architecture.....	16
2.2.3.4.3.2.1 Line Extraction Software.....	19
2.2.3.4.3.3 InterStablePipe Pipeline.....	20
2.2.3.4.4.1 Word Cloud-Training.....	21
2.2.3.4.4.2.1 Guidance Scale.....	23
2.2.3.4.4.2.2 InterStableCloud Pipeline.....	24
2.2.3.4.5.3 InterStableLLM Pipeline.....	27
2.2.3.4.7.2 InterStableLLMRLLine Pipeline.....	28
2.2.6.1 White Picture.....	29
3.6 Kosten.....	32