BACHELORARBEIT

Nutzung von Stable Diffusion zur Datensatzgenerierung für die Bilderkennung von Verschmutzungen in Sanitärbereichen

vorgelegt am 24. Juni 2024

Martin Tasch, [redacted]

Erstprüferin: Prof. Dr. Tessa Taefi
Zweitprüfer: M.Sc. Cedrik Fuchs

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Department Medientechnik
Finkenau 35
22081 Hamburg

Zusammenfassung

Diese Arbeit untersucht die Anwendung von Stable Diffusion zur Generierung von Datensätzen für die Bilderkennung, speziell im Sanitärbereich. Die Methodik umfasst die Anpassung von Stable Diffusion für diese spezifische Anwendung mittels einer Low Rank Adaptation (LoRA) und nutzt das Realistic Vision 5.1 Modell zur Erstellung photorealistischer Bilder. Die Arbeit demonstriert die Machbarkeit der automatisierten Datensatzerstellung mit Stable Diffusion und erzielt vielversprechende Ergebnisse mit einer Validierungsgenauigkeit von über 90 % bei den generierten Datensätzen, bewertet durch ein trainiertes InceptionV3 Bilderkennungsmodell. Verbesserungspotentiale wurden identifiziert, darunter die Notwendigkeit, die realistische Darstellung der Verschmutzung durch einen größeren und realistischeren Trainingsdatensatz zu verbessern und das Modell mittels LoRA weiter anzupassen. Zukünftige Arbeiten sollten außerdem die Erstellung eines größeren, ausgewogeneren Evaluationsdatensatzes in Betracht ziehen, um aussagekräftigere Ergebnisse zu erzielen. Das Entwickelte Framework zur automatisierten Datensatzgenerierung ist unter folgendem Link auf Github verfügbar: https://github.com/marttasch/StableDiffusion_generateDataset

Abstract

This thesis investigates the application of Stable Diffusion to generate datasets for image recognition, specifically in the sanitation domain. The methodology includes the adaptation of Stable Diffusion for this specific application using a Low Rank Adaptation (LoRA) and utilises the Realistic Vision 5.1 model to generate photorealistic images. The work demonstrates the feasibility of automated dataset generation with Stable Diffusion and achieves promising results with a validation accuracy of over 90 % for the generated datasets, evaluated by a trained InceptionV3 image recognition model. Areas for improvement were identified, including the need to improve the realistic representation of contamination with a larger and more realistic training dataset and to further customise the model using LoRA. Future work should also consider the creation of a larger, more balanced evaluation dataset to achieve more meaningful results. The developed framework for automated dataset generation is available on Github at the following link: https://github.com/marttasch/StableDiffusion_generateDataset

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Codeblöcke

Abkürzungsverzeichnis

API	Application Programming Interface
GB	Gigabyte
KB	Kilobyte
MB	Megabyte
RAM	Random Access Memory (Arbeitsspeicher)
VRAM	Video Random Access Memory (Grafikspeicher)
UI	User Interface
CNN	Convolutional Neural Network
GPT	Generative Pre-trained Transformer
PCA	Principal Component Analysis
CLIP	Contrastive Language-Image Pretraining
LDM	Latent Diffusion Model
LoRA	Low Rank Adaptation
SD	Stable Diffusion
SDXL	Stable Diffusion XL
txt2img	Text-to-Image
img2img	Image-to-Image
FN	False Negative
FP	False Positive
TN	True Negative
TP	True Positive

1 Einleitung

1.1 Hintergrund und Motivation

Die Gewährleistung von Hygiene und Sauberkeit in öffentlichen Sanitäreinrichtungen sind von allgemeinem Interesse und stellen ein essentielles Anliegen der öffentlichen Gesundheit und des Wohlbefindens dar. Schlechte Arbeitsbedingungen, ein Mangel an Arbeitskräften und eine chronische Unterfinanzierung sind jedoch Herausforderungen mit denen die Reinigungsdienstleister:innen konfrontiert sind (Bergfeld, 2022). Die manuelle Überwachung und Reinigung von Sanitärbereichen ist zeitaufwendig und unterliegt subjektiven Bewertungsfehlern (Bergfeld, 2022; Mezil, 2023). Automatisierte Lösungen, insbesondere bildbasierte Erkennungssysteme, könnten den Reinigungsprozess effizienter und objektiver gestalten und somit die Arbeitsbedingungen und das Wohlbefinden der Nutzer:innen steigern (Bergfeld, 2022; Eurofound & Mandl, 2021).

Bereits vorhandene „Smart Restroom“ Systeme, die auf IoT-Geräten basieren, bieten Unterstützung bei der Überwachung von Verbrauchsmaterialien und der Geruchskontrolle, um Reinigungsprozesse zu verbessern (Chaudhari & Bagde, 2021; See-To et al., 2023). Eine automatisierte visuelle Überprüfung und objektive Bewertung der Sauberkeit fehlt jedoch.

Bodenreinigende Saug- und Wischroboter sind bereits mit Methoden zur Erkennung von Verschmutzungen ausgestattet, um den Boden gezielt und effizient zu reinigen (Canedo et al., 2021; Yun et al., 2022). Für die Erkennung von Verschmutzungen in Sanitärbereichen zur gezielten Reinigung wurde bisher scheinbar nur sehr wenig Forschung betrieben. Erste Reinigungsroboter für Sanitärbereiche handeln vortrainierte Routinen ab und sind nicht in der Lage, Verschmutzungen zu erkennen und gezielt zu reinigen oder auf unvorhergesehene Situationen reagieren zu können (Heater, 2020).

Ein Bilderkennungssystem, das Verschmutzungen in öffentlichen Sanitärbereichen effizient klassifizieren kann, könnte dazu beitragen, die Notwendigkeit manueller Kontrollen zu reduzieren und die Einsatzplanung von Reinigungskräften oder Reinigungsrobotern zu optimieren. Trotz der Verfügbarkeit fortschrittlicher Bilderkennungstechnologien ist die Forschung zur Sauberkeitserkennung in Sanitärbereichen noch begrenzt. Bislang konnte nur eine relevante Studie gefunden werden, die sich auf die Erkennung von Verschmutzungen an Pissoirs konzentriert (Jayasinghe et al., 2019).

Die Erstellung eines solchen Bilderkennungssystems erfordert einen ausführlichen Datensatz, der zum Training von Bilderkennungsmodellen verwendet werden kann. Die Erstellung eines solchen Datensatzes ist jedoch aufwendig und zeitintensiv, da auf konventionelle Weise die Bilder manuell gelabelt und gegebenfalls segmentiert werden müssen. Ein entsprechender öffentlicher Datensatz konnte nicht gefunden werden. Die Entwicklung von Bildgenerierungsmethoden, wie Stable Diffusion, bieten jedoch eine Möglichkeit zur Generierung von synthetischen Datensätzen mit segmentierten Labeln (Shenoda & Kim, 2023). Ein solcher Datensatz könnte zum Training von Bilderkennungsmodellen verwendet werden, um Verschmutzungen in Sanitärbereichen zu erkennen und zu klassifizieren.

Vor diesem Hintergrund zielt diese Arbeit darauf ab, mittels der Stable Diffusion Technologie einen umfassenden Datensatz zu generieren, der zum Training von Bilderkennungsmodellen für die Sauberkeitsbewertung verwendet werden kann. Die Forschungsfrage lautet daher: *„Wie kann die Technologie der Stable Diffusion genutzt werden, um einen realistischen und umfassenden Datensatz für die Schulung von Bilderkennungsmodellen zur effektiven Bewertung der Sauberkeit von Sanitärbereichen zu erstellen?“*

1.2 Struktur der Arbeit

Nachdem in diesem Kapitel zunächst die Hintergründe und die zugrunde liegende Motivation der Arbeit erläutert wurden, wird im folgenden Kapitel 2 der aktuelle Forschungsstand im Bereich der Bilderkennung zur Verschmutzungserkennung dargestellt und daraus die Forschungslücke abgeleitet. Kapitel 3 gibt einen Überblick über die Grundlagen, insbesondere der Technologie Stable Diffusion und dessen Aufbau und Funktionsweise. Sowie einige Grundlagen der Bilderkennung mittels Convolutional Neuronal Networks (CNN), die für das Verständnis der Methodik notwendig sind. Kapitel 4 stellt die Methodik der Datensatzgenerierung mittels Stable Diffusion und der Validierung der generierten Datensätze mittels eines CNN-Modells dar. In Kapitel 5 werden die Ergebnisse der Datensatzgenerierung und Validierung präsentiert und in Kapitel 6 werden diese diskutiert sowie Verbesserungspotenziale und zukünftige Arbeiten aufgezeigt. Zum Abschluss wird in Kapitel 7 ein Fazit gezogen und ein Ausblick gegeben.

2 Literatur Review

Dieses Kapitel widmet sich einer Durchsicht der relevanten Literatur mit dem Ziel, den aktuellen Stand der Forschung zu ermitteln, existierende Technologien und Methoden zu evaluieren und schließlich Forschungslücken zu identifizieren, die diese Arbeit zu schließen beabsichtigt. Abgeleitet aus dem Praxisproblem wurde dazu eine ausführliche Literaturrecherche in zwei Schritten durchgeführt. Zunächst wurde eine allgemeine Recherche zu automatisierten Reinigungssystemen und Bilderkennungstechnologien für Sanitärbereiche durchgeführt, um festzustellen welche Techniken und Methoden bereits existieren und welche Forschung bereits zu automatisierter Verschmutzungserkennung in Sanitärbereichen durchgeführt wurde.

Anschließend wurde eine spezifische Recherche zu synthetischen Datengenerierungsmethoden für Bilderkennung durchgeführt, um herauszufinden, welche Methoden zur Generierung von synthetischen Datensätzen für die Bilderkennung existieren und wie diese in der Praxis angewendet werden.

2.1 Allgemeine Recherche

Dass ursprüngliche Praxisproblem, welches mir von meinem Unternehmen zugetragen wurde, betraf allgemein den Einsatz von Bilderkennungssystemen zur automatisierten Reinigung von Sanitärbereichen. Daher wurde zunächst eine umfassende allgemeine Literaturrecherche durchgeführt, um ein Verständnis für den aktuellen Stand der Technik und Forschung zu erlangen. Dabei ergaben sich folgende Fragestellungen, die im Rahmen der Recherche beantwortet werden sollten:

- Welche Bilderkennungsmethoden eignen sich um Verschmutzungen zu erkennen/klassifizieren?
- Welche Bilderkennungsmethoden lassen sich einfach an lokale Gegebenheiten anpassen?
- Gibt es Bilderkennungssysteme in ähnlichen Bereichen?
- Gibt es Datensätze für die Bilderkennung von Sanitärbereichen?

Da die Suchmaschine Google Scholar eine Vielzahl an wissenschaftlichen Datenbanken wie IEEE Xplore, SpringerLink, ScienceDirect, ResearchGate und viele mehr durchsucht, wurde diese als primäre Suchmaschine verwendet. Die Suchergebnisse wurden nach Relevanz sortiert, um die relevantesten Studien und Artikel anzuzeigen. Die Auswahl der Suchbegriffe erfolgte so, dass sie möglichst allgemein und umfassend waren, um eine breite Palette an Literatur abzudecken. Zur Filterung und Bewertung der gefundenen Literatur wurden folgende Kriterien angewendet:

- Aktualität der Veröffentlichung (nicht älter als 5 Jahre)
- Relevanz zum Thema (Titel, Abstract, Fazit)
- Seriösität der Quelle (Peer Review)
- Verfügbarkeit des Volltextes

Die folgenden Suchbegriffe führten dabei unter Anwendung der oben genannten Kriterien zu den relevantesten Ergebnissen, die in Tabelle 2.1 dargestellt sind.

Suchbegriff	Treffer	Relevante Literatur
„image processing“ AND „dirt detection“	225	4
„image processing“ AND (bathroom OR restroom OR toilet OR sanitary)	6.010	6
„image processing“ restroom cleaning	117	7
visual recognition for cleaning	16.900	1
„camera“ AND „dirt detection“	257	8
„camera“ AND „dirt detection“ AND „privacy“	42	4
„camera“ AND „cleaning“ AND „toilet“	6.900	8

Tabelle 2.1: Ergebnisse der allgemeinen Literaturrecherche

2.1.1 Stand der Technik

Die Ergebnisse der allgemeinen Literaturrecherche zeigen, dass vor allem im Bereich der Bodenreinigung (Saug- und Wischroboter) bereits viel Forschung zur Verschmutzungserkennung mittels Bilderkennung durchgeführt wurde. Dieser Abschnitt gibt einen Überblick über die wichtigsten Erkenntnisse und Forschungsergebnisse in diesem Bereich.

So zeigt die Arbeit von Yun et al. (2022), dass Deep-Learning-Modelle effektiv zur Erkennung von Verschmutzungen auf Böden, auch in realen Umgebungen, eingesetzt werden können. Die Forscher verwenden in dieser Arbeit YOLOv5 zur Erkennung und DDRNet (Depth Map

Denoising and Refinement Network) zur Segmentierung der Verschmutzungen. Beide Netzwerke sind Convolutional Neural Networks (CNNs) und wurden auf einem realen Datensatz trainiert. Dabei betont die Arbeit, dass bestehende Forschungen häufig synthetische Datensätze zur Bilderkennung verwenden, aufgrund der Schwierigkeit echte Datensätze zu sammeln und der Verfügbarkeit bestehender Datensätze.

Die Arbeit von Canedo et al. (2021) präsentiert ebenfalls ein auf YOLOv5 basierendes Modell zur Erkennung von Verschmutzungen auf Böden. In dieser Studie wurde das Modell auf einem synthetisch erstellten Datensatz trainiert, der eine Vielzahl von Proben unterschiedlicher Schmutzarten auf verschiedenen Bodenarten enthält. Die Ergebnisse der Studie zeigen, dass das trainierte Modell in der Lage ist, Verschmutzungen in realen Datensätzen, wie dem ACIN-Datensatz,[1] mit hoher Präzision zu erkennen. Dies unterstreicht die Effektivität des synthetischen Datensatzes als Trainingsgrundlage.

Bei der Suche nach Forschung zur Verschmutzungserkennung in Sanitärbereichen konnte nur eine relevante Studie gefunden werden, die sich mit der Erkennung von Verschmutzungen beschäftigt. Die Arbeit von Jayasinghe et al. (2019) zeigt, dass es möglich ist die Sauberkeit von Pissoirs anhand von Bildern zu klassifizieren. Dabei wurde ein CNN, basierend auf dem Inception-v3 Modell, zusammen mit einer Principal Component Analysis (PCA) zur Klassifizierung der Sauberkeit verwendet. Die vorgestellte Methode bietet mit einer Präzision von 91 % eine bessere Klassifizierung als die getesteten Modelle wie Inception-v3 oder ResNet50.

Im weiteren konnte nur ein Bericht zu einem Reinigungsroboter der Firma *Somatic* gefunden werden, der zur Reinigung öffentlicher Sanitärbereiche eingesetzt wird. Dieser Roboter hat keine Bilderkennung zur Verschmutzungserkennung, sondern arbeitet jediglich vortrainierte Routinen ab.[2] Ein solcher Roboter könnte jedoch von einem Bilderkennungssystem profitieren, um Verschmutzungen zu erkennen und gezielt zu reinigen. Dieses Beispiel zeigt die Relevanz des Themas und die Notwendigkeit für weitere Forschung in diesem Bereich.

Eine Recherche nach bestehenden öffentlichen Datensätzen für die Bilderkennung von Sanitärbereichen ergab keine relevanten Ergebnisse. Es konnten keine öffentlichen Datensätze für die Verschmutzungserkennung in Sanitärbereichen gefunden werden.

[1] *ACIN Dataset*. (n. d.). Roboflow. Verfügbar 9. März 2024 unter https://universe.roboflow.com/alena-gdt4w/acin-dataset/dataset/2

[2] Heater, B. (2020, 4. März). *This bathroom cleaning robot is trained in VR to clean up after you.* TechCrunch. Verfügbar 13. März 2024 unter https://techcrunch.com/2020/03/04/this-bathroom-cleaning-robot-is-trained-in-vr-to-clean-up-after-you/

2.1.2 Forschungslücke

Die allgemeine Literaturrecherche hat gezeigt, dass es bisher nur sehr wenig Forschung zur Verschmutzungserkennung in Sanitärbereichen mittels Bilderkennung gibt. Es konnten keine öffentlichen Datensätze für die Bilderkennung von Sanitärbereichen gefunden werden. Die Mehrheit der Forschungsarbeiten zur Verschmutzungserkennung mittels Bilderkennung konzentriert sich auf die Reinigung von Böden durch Saug- und Wischroboter. Die wenigen Studien, die sich mit der Verschmutzungserkennung in Sanitärbereichen befassen, zeigen jedoch, dass es möglich ist, die Sauberkeit von Pissoirs anhand von Bildern zu klassifizieren. Arbeiten zur Bodenreinigung weisen zudem darauf hin, dass synthetische Datensätze eine effektive Alternative zum Training mit realen Datensätzen darstellen.

Angesichts dieser Erkenntnisse und dem Aufkommen neuer Bildgenerierungstechnologien wie Stable Diffusion stellt sich die Frage, ob diese Technologie zur Erstellung eines synthetischen Datensatzes für die Bilderkennung von Sanitärbereichen genutzt werden kann und ob dies eine effiziente Methode zur Generierung eines solchen Datensatzes darstellt.

2.2 Spezifische Recherche

Vor diesem Hintergrund wurde eine spezifische Recherche zu Methoden der synthetischen Datengenerierung für die Bilderkennung durchgeführt. Besonderes Augenmerk wurde dabei auf die neuartige Technologie Stable Diffusion gelegt, die eine grundlegend neue Methode zur Generierung von synthetischen Bildern darstellt. Das Ziel dieser Recherche war es, zu untersuchen, ob es bereits Forschung zu dieser Technologie und ihrer Anwendung zur Generierung von synthetischen Datensätzen für die Bilderkennung gibt. Die zentralen Fragestellungen, die dabei beantwortet werden sollten, lauten:

- Welche Methoden zur Generierung von synthetischen Datensätzen für die Bilderkennung existieren?
- Kann Stable Diffusion zur Generierung von synthetischen Datensätzen für die Bilderkennung verwendet werden?
- Welche Methoden eignen sich für die Generierung von segmentierten Labeln?

Die folgenden Suchbegriffe führten dabei unter Anwendung der oben genannten Kriterien zu den relevantesten Ergebnissen, die in Tabelle 2.2 dargestellt sind.

Die spezifische Literaturrecherche brachte keine Forschung zu konkreten Anwendungen und Ergebnisse einer Bilderkennung mittels synthetischer Datensätze, die mit Stable Diffusion generiert wurden. Es konnte jedoch eine Arbeit von Shenoda und Kim (2023) gefunden

Suchbegriff	Treffer	Relevante Literatur
„dataset generation“ AND stable diffusion	1.110	4
„dataset generation“ AND „diffusion models“	417	8
„dataset generation“ AND (bathroom OR restroom OR toilet OR sanitary)	179	0
„dataset generation“ AND „restroom“	11	0
„dataset generation“ AND „diffusion models“ AND „restroom“	0	0

Tabelle 2.2: Ergebnisse der spezifischen Literaturrecherche

werden, die ein Framework zur Nutzung von Stable Diffusion, speziell zur Generierung von synthetischen Datensätzen mit segmentierten Labeln, vorstellt.

Das Framework „DiffuGen“ ist als Open Source Projekt verfügbar und bietet eine effektive Methode zur Generierung von synthetischen Datensätzen zur Bilderkennung und legt dar, welche Mechanismen und Methoden zur Generierung von synthetischen Bildern verwendet werden können. So verwendet dieses Framework die Textual Inversion Methode um Stable Diffusin neue Konzepte zu lehren. Der Cross-Attention-Mechanismus in Stable Diffusion wird genutzt um segmentierte Labeln aus der Aufmerksamkeit des Modell auf bestimmte Bildbereiche zu generieren.

Zusammenfassend lässt sich sagen, dass Stable Diffusion vielversprechende Möglichkeiten für die Generierung synthetischer Datensätze bietet. Im nächsten Kapitel werden die theoretischen Grundlagen dieser Technologien und Methoden genauer erläutert.

3 Grundlagen

Dieses Kapitel bietet einen umfassenden Überblick über die zentralen Konzepte und Technologien, die für das Verständnis dieser Arbeit unerlässlich sind. Der Fokus liegt auf der detaillierten Darstellung von Stable Diffusion und seinen Komponenten, die für die Generierung von Bildern aus textuellen Beschreibungen erforderlich sind. Zudem werden wesentliche Aspekte der Bilderkennung mit Convolutional Neural Networks (CNNs), einschließlich Transfer Learning und Leistungsmetriken, behandelt. Da grundlegende Informationen zu neuronalen Netzen und CNNs bereits ausführlich in der Fachliteratur beschrieben wurden, werden hier nur die für diese Arbeit relevanten Aspekte erläutert. Weitere grundlegende Informationen zu neuronalen Netzen und maschinellem Lernen finden sich beispielsweise den Bücher *Neuronale Netze kompakt* von Sonnet (2022), oder *Deep Learning* von Goodfellow et al. (2016). Eine umfassende Einführung in die Bilderkennung mittels CNNs bietet das Buch „Bilderkennung mit tiefen neuronalen Netzen“ von Paaß und Hecker (2020).

3.1 Stable Diffusion

Stable Diffusion ist ein latentes Diffusionsmodell (LDM), das in der Lage ist, mittels der stochastischen Diffusion hochauflösende Bilder aus textuellen Beschreibungen zu generieren (Ho et al., 2020). Der grundlegende Algorithmus wurde von der ComVis-Gruppe der LMU München in Zusammenarbeit mit der amerikanischen Firma Runway entwickelt (Rombach et al., 2022).[1] Die Entwicklung wurde von Stability AI unterstützt, welches maßgeblich das Budget und die Hardware-Ressourcen zur Verfügung stellte (Wiggers, 2022).

Veröffentlicht werden die Modelle unter der Open Source Lizenz *CreativeML OpenRAIL M*. Die ersten Versionen (1.1 - 1.4) wurden von der CompVis-Gruppe im August 2022 veröffentlicht,[2] die Version 1.5 folgte im Januar 2023 veröffentlicht von Runway.[3] Weitere Versionen (2.0, 2.1, XL, XL-turbo) wurden anschließend von Stability AI veröffentlicht.[4]

[1] *Revolutionizing image generation by AI: Turning text into images.* (2022, 1. September). Verfügbar 18. April 2024 unter https://www.lmu.de/en/newsroom/news-overview/news/revolutionizing-image-generation-by-ai-turning-text-into-images.html

[2] *CompVis (CompVis).* (n. d.). Verfügbar 27. Mai 2024 unter https://huggingface.co/CompVis

[3] *Runwayml (Runway).* (n. d.). Verfügbar 27. Mai 2024 unter https://huggingface.co/runwayml

[4] *Stabilityai (Stability AI).* (n. d.). Verfügbar 27. Mai 2024 unter https://huggingface.co/stabilityai

Trainiert wurde Stable Diffusion auf verschiedenen Teilmengen des öffentlichen Datensatzes *LAION-5B*.[5]

3.1.1 Diffusionsmodelle und ihre Funktionsweise

Diffusionsmodelle sind eine Klasse von generativen Modellen, die darauf abzielen, Daten durch ein stochastisches Verfahren zu erzeugen. Ein prominentes Beispiel ist das Denoising Diffusion Probabilistic Model (DDPM), das für die Bildgenerierung verwendet wird (Ho et al., 2020). Diese Modelle funktionieren, indem sie ein Bild schrittweise durch Hinzufügen und Entfernen von Rauschen erzeugen (Ho et al., 2020). Der Prozess umfasst zwei Hauptphasen:

- **Vorwärts-Diffusion**: In dieser Phase wird zu einem Bild iterativ Rauschen hinzugefügt, bis das Bild vollständig verrauscht ist. Dieser Prozess kann als eine Markov-Kette modelliert werden, bei der jedes Zwischenschrittbild eine bedingte Verteilung auf das vorherige Bild hat. Mathematisch kann dies als Übergang von $q(x_t|x_{t-1})$ beschrieben werden, wobei x_t den Zustand des Bildes zum Zeitpunkt t darstellt. Die Vorwärts-Diffusion wird im Training verwendet, um die bedingten Wahrscheinlichkeitsverteilungen zu lernen, die für die Erzeugung von Bildern notwendig sind (Ho et al., 2020).
- **Rückwärts-Diffusion**: Der Denoising-Prozess beginnt mit einem verrauschten Bild und entfernt iterativ das Rauschen, um das ursprüngliche Bild wiederherzustellen. Dieser Prozess wird durch ein neuronales Netzwerk gesteuert, das die bedingten Wahrscheinlichkeitsverteilungen gelernt hat, die für die Entfernung des Rauschens notwendig sind. Dies kann als Übergang von $p_\theta(x_{t-1}|x_t)$ beschrieben werden, wobei θ die Parameter des neuronalen Netzwerks sind (Ho et al., 2020).

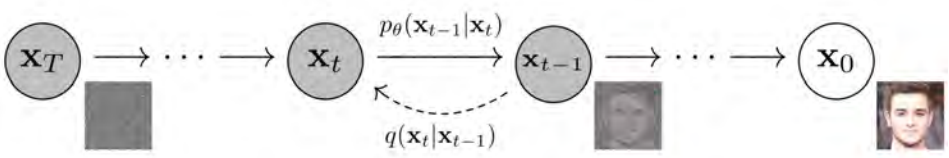

Abbildung 3.1: Überblick über den Diffusionsprozess. (Ho et al., 2020)

3.1.2 Latente Diffusionsmodelle

Latente Diffusionsmodelle (LDMs), wie Stable Diffusion, sind eine spezielle Klasse von Diffusionsmodellen, die im latenten Raum arbeiten. Anstatt direkt auf den hochdimensionalen

[5] *Runwayml/Stable-Diffusion-v1-5 · Hugging Face*. (n. d.). Verfügbar 18. April 2024 unter https://huggingface.co/runwayml/stable-diffusion-v1-5

Bilddaten zu operieren, transformieren LDMs die Bilddaten in einen komprimierten, abstrakten Raum, den sogenannten latenten Raum. In diesem niederdimensionalen Raum können die relevanten Merkmale der Bilddaten effizient extrahiert und verfeinert werden, was den Generierungsprozess vereinfacht und beschleunigt (Rombach et al., 2022).

Der Hauptunterschied zwischen traditionellen Diffusionsmodellen und latenten Diffusionsmodellen liegt also in der Datenrepräsentation und -verarbeitung. Während herkömmliche Diffusionsmodelle direkt im Bildraum arbeiten, nutzen LDMs in einer zusätzlichen Komprimierungsschicht einen Encoder (siehe Kapitel 3.1.3.1), um die Dimension der Daten zu reduzieren (Rombach et al., 2022).

Die beschriebene Funktionsweise von LDMs bietet mehrere Vorteile gegenüber traditionellen Diffusionsmodellen, die im Folgenden beschrieben werden:

- **Reduzierter Ressourcenbedarf**: Da die Berechnungen im latenten Raum erfolgen, der eine niedrigere Dimension hat als der ursprüngliche Bildraum, sind die erforderlichen Rechenressourcen deutlich geringer, sodass ein Training auch auf Konsumer-Hardware mit unter 10 GB VRAM möglich wird (Rombach et al., 2022).[6]
- **Effiziente Bildgenerierung**: Im latenten Raum können die relevanten Merkmale der Bilddaten effizienter verarbeitet und extrahiert werden, was zu einer schnelleren und präziseren Bildgenerierung führt (Rombach et al., 2022).
- **Skalierbarkeit:** LDMs können effizienter auf größere Datensätze und komplexere Modelle skaliert werden, da die Rechenressourcen effektiver genutzt werden und die Modellgröße reduziert wird (Rombach et al., 2022).

3.1.3 Hauptkomponenten von Stable Diffusion

Stable Diffusion besteht aus mehreren Komponenten, die zusammenarbeiten, um hochauflösende Bilder aus textuellen Beschreibungen zu generieren. Die wichtigsten Komponenten werden in diesem Abschnitt vorgestellt und erläutert.

3.1.3.1 Autoencoder

Ein Autoencoder ist eine spezielle Form eines neuronalen Netzes, das darauf trainiert ist, Eingabedaten in einer komprimierten, niederdimensionalen Form darzustellen und sie anschließend in ihre ursprüngliche Form zu rekonstruieren. In Stable Diffusion spielt der

[6] *Stable Diffusion launch announcement.* (2022, 10. August). Stability AI. Verfügbar 27. Mai 2024 unter https://stability.ai/news/stable-diffusion-announcement

Autoencoder eine entscheidende Rolle bei der Transformation und Rekonstruktion von Bilddaten im latenten Raum (Kramer, 1992; Rombach et al., 2022). Der Autoencoder besteht aus zwei Hauptteilen:

Der **Encoder** komprimiert die Eingabebilder in eine niederdimensionale, latente Repräsentation. Diese Komprimierung ermöglicht es, die wesentlichen Merkmale der Bilddaten in einem reduzierten Raum zu speichern, was die nachfolgenden Berechnungen effizienter macht (Kingma & Welling, 2022).

Der **Decoder** nimmt die latenten Repräsentationen und rekonstruiert daraus die hochauflösenden Bilder. Der Decoder stellt sicher, dass die Details und die Qualität der ursprünglichen Bilddaten bei der Rücktransformation erhalten bleiben (Kingma & Welling, 2022).

Der Autoencoder ist also erforderlich, damit Stable Diffusion effektiv im latenten Raum arbeiten kann und die beschriebenen Vorteile von LDMs nutzen kann (Rombach et al., 2022). Eine schematische Darstellung eines Autoencoders ist in Abbildung 3.2 dargestellt.

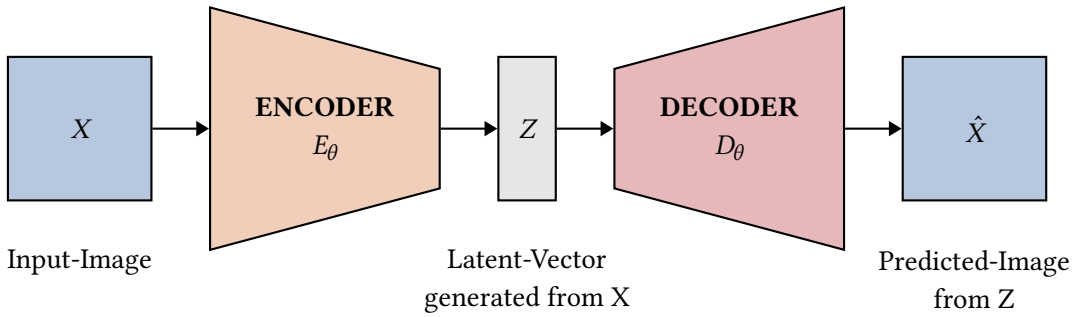

Abbildung 3.2: Schematische Darstellung eines Autoencoder. Eigene Darstellung basierend auf Mendelevitch et al., 2020

Variational Autoencoder (VAE)
In einigen Versionen von Stable Diffusion wird ein Variationaler Autoencoder (VAE) verwendet, um die latente Repräsentation der Bilddaten zu erzeugen (Rombach et al., 2022). Ein VAE ist eine spezielle Form eines Autoencoders, der zusätzlich zur Komprimierung und Rekonstruktion der Bilddaten auch eine latente Verteilung erzeugt. Diese latente Verteilung ermöglicht es dem Modell, neue, realistische Bilder zu generieren, indem es zufällige Stichproben aus der Verteilung zieht und diese in den Decoder einspeist (Kingma & Welling, 2019).

3.1.3.2 Text-Encoder

Der Text-Encoder transformiert die Texteingaben, auch *Prompts* genannt, in latente Repräsentationen. Ein häufig verwendetes Modell hierfür ist *CLIP* (Contrastive Language–Image

Pretraining),[7] das darauf trainiert ist, Text und Bild in einem gemeinsamen latenten Raum zu repräsentieren (Radford et al., 2021).

Der Prozess der Text-Encodierung umfasst mehrere Schritte:

- **Zerlegen in Tokens**: Die Texteingaben werden zunächst in einzelne Tokens zerlegt, was in der Regel Wörter, Wortteile oder Zeichenfolgen sind (Radford et al., 2021).
- **Embedding**: Diese Tokens werden in einen Vektorraum eingebettet, wobei CLIP Einbettungsvektoren für jeden Token erzeugt, die die semantischen Bedeutungen der Tokens in einem hochdimensionalen Raum repräsentieren. Ein typischer Text-Encoder, wie er in CLIP verwendet wird, basiert auf einem Transformer-Modell. Diese Modelle nutzen Self-Attention-Mechanismen,[8] um kontextuelle Informationen zu erfassen und kontextuelle Einbettungen für jedes Token zu erzeugen (Radford et al., 2021; Saharia et al., 2022).
- **Integration in den Diffusionsprozess**: Die erzeugten Vektor-Einbettungen werden in den Diffusionsprozess eingespeist, um die Bildgenerierung zu steuern und sicherzustellen, dass die generierten Bilder den Texteingaben entsprechen (Rombach et al., 2022; Saharia et al., 2022).

Stable Diffusion verwendet als Text-Encoder sowohl *CLIP* von OpenAI (Radford et al., 2021)[9] als auch die Open Source Weiterentwicklung *OpenCLIP* (Cherti et al., 2022).[10] Die Tabelle 3.1 gibt einen Überblick über die verschiedenen Versionen des Text-Encoders in Stable Diffusion.

SD Version	**Textencoder**
Stable Diffusion v1	CLIP[11]
Stable Diffusion v2	OpenCLIP[12]
Stable Diffusion XL (SDXL)	OpenCLIP und CLIP[13]

Tabelle 3.1: Textencoder in verschiedenen Versionen von Stable Diffusion

[7] *Openai/CLIP*. (2024, 27. Mai). Verfügbar 27. Mai 2024 unter https://github.com/openai/CLIP

[8] Self-Attention ist ein Mechanismus in neuronalen Netzen, der es ermöglicht, die Bedeutung jedes einzelnen Elements einer Sequenz im Kontext der gesamten Sequenz zu bestimmen, indem er die Beziehungen zwischen allen Elementen gleichzeitig berücksichtigt.
Golroudbari, A. A. (2023, 1. September). *Understanding Self-Attention - A Step-by-Step Guide |*. Verfügbar 17. Juni 2024 unter https://armanasq.github.io/nlp/self-attention/

[9] *Openai/CLIP*. (2024, 27. Mai). Verfügbar 27. Mai 2024 unter https://github.com/openai/CLIP

[10] Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., & Schmidt, L. (2021, Juli). *OpenCLIP* (Version v0.1). https://doi.org/10.5281/zenodo.5143773

3.1.3.3 Cross-Attention

Cross-Attention ist ein wesentlicher Mechanismus in neuronalen Netzen, der zur Kombination von Informationen aus zwei verschiedenen Eingabesequenzen verwendet wird, beispielsweise Text und Bild. In Stable Diffusion spielt Cross-Attention eine zentrale Rolle, indem es Textbeschreibungen in den Bildgenerierungsprozess integriert (Rombach et al., 2022). Dabei dienen die Text-Embeddings aus dem Text-Encoder als Abfragen (Queries) und die latenten Bildrepräsentationen als Schlüssel (Keys) und Werte (Values) für den Cross-Attention-Mechanismus (Vaswani et al., 2023).

Die Berechnung der Aufmerksamkeit erfolgt durch den Vergleich der Queries mit den Keys, wobei für jede Query die Ähnlichkeit zu allen Keys berechnet wird. Diese Ähnlichkeiten werden durch das Skalarprodukt der Query- und Key-Vektoren ermittelt und anschließend durch eine Softmax-Funktion normalisiert, um Aufmerksamkeitsgewichte zu erhalten. Diese Gewichte geben an, wie viel Gewicht jedem Key-Value-Paar in Bezug auf die jeweilige Query beigemessen wird. Die Values werden entsprechend dieser Gewichte gewichtet und summiert, um die endgültige Ausgabe der Cross-Attention-Schicht zu erzeugen. Die Berechnung der Cross-Attention wird durch die Gleichung 3.1 beschrieben (Vaswani et al., 2023).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.1}$$

Durch diesen Mechanismus kann das Modell wichtige Teile des Textes identifizieren und in die latenten Bildrepräsentationen integrieren. Dies ermöglicht es dem Modell, relevante Informationen aus den Textbeschreibungen zu extrahieren und in die Bildgenerierung einfließen zu lassen, was zu einer präzisen und detailreichen Umsetzung der textlichen Eingaben in visuelle Ausgaben führt (Rombach et al., 2022).

3.1.3.4 Sampler & Scheduler

Der Scheduler und der Sampler spielen eine entscheidende Rolle im Stable Diffusion Prozess, indem sie zusammenarbeiten, um qualitativ hochwertige Bilder zu generieren.
Der **Scheduler** steuert den Diffusionsprozess, indem er festlegt, wie und wann das Rauschen

[11] *Runwayml/Stable-Diffusion-v1-5 · Hugging Face*. (n. d.). Verfügbar 18. April 2024 unter https://huggingface.co/runwayml/stable-diffusion-v1-5

[12] *Stabilityai/Stable-Diffusion-2 · Hugging Face*. (n. d.). Verfügbar 27. Mai 2024 unter https://huggingface.co/stabilityai/stable-diffusion-2

[13] *Stabilityai/Stable-Diffusion-Xl-Base-1.0 · Hugging Face*. (n. d.). Verfügbar 27. Mai 2024 unter https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0

während der Bildgenerierung reduziert wird. Er bestimmt die Schrittweite und das Timing der Rauschreduktion, um eine stabile und konsistente Bildgenerierung zu gewährleisten (Song et al., 2022). Der Scheduler arbeitet eng mit dem UNet zusammen, indem er die Eingabedaten für jede Iteration vorbereitet und sicherstellt, dass das Modell in jedem Schritt eine optimierte Version des Bildes ausgibt (Rombach et al., 2022; Song et al., 2022).

Der **Sampler** hingegen entscheidet, wie die Samples aus dem latenten Raum des Modells gezogen werden. Er wählt die Strategien für das Sampling aus, die die Vielfalt und Qualität der generierten Bilder beeinflussen (Karras et al., 2022; Song et al., 2022). Während der Scheduler die Intervalle und Intensitäten der Rauschreduktion steuert, bestimmt der Sampler die spezifischen Schritte und Werte für die Generierung jedes Bildes. Samples, die durch den Sampler gezogen werden, werden in den Decoder eingespeist, um die generierten Bilder zu erzeugen. Der Sampler beeinflusst somit direkt die Qualität und Vielfalt der Ausgaben (Song et al., 2022).

Gemeinsam sorgen sie dafür, dass das Modell effizient arbeitet und qualitativ hochwertige, konsistente Bilder erzeugt.

3.1.3.5 UNet

Ursprünglich wurde das UNet von Ronneberger et al. (2015) für die medizinische Bildsegmentierung entwickelt und zeichnet sich durch seine charakteristische U-Form aus, die eine symmetrische Anordnung von Encoder- und Decoder-Schichten mit Cross-Connections umfasst (Rombach et al., 2022; Ronneberger et al., 2015).

In Stable Diffusion dient das UNet als das primäre neuronale Netz, das den Diffusionsprozess durchführt. Es transformiert verrauschte Eingabebilder schrittweise in detaillierte und hochauflösende Ausgaben, indem es eine tiefenfokussierte Architektur verwendet, die eine effektive Feature-Extraktion und -Rekonstruktion ermöglicht (Rombach et al., 2022; Ronneberger et al., 2015).

Aufbau und Besonderheit des UNet

Das UNet besteht aus einem symmetrischen Encoder-Decoder-Setup. Der Encoder enthält eine Reihe von Residual-Blöcken, bestehend aus Convolutional-Schichten und Aktivierungsfunktionen, zur Extraktion von Merkmalen. Diese Blöcke sind durch Pooling-Schichten verbunden, die die räumliche Auflösung der Eingabe schrittweise reduzieren. Der Decoder hingegen rekonstruiert das Bild durch schrittweise Erhöhung der Auflösung mittels transponierter Convolutional-Schichten. Die besondere U-Form des Netzwerks wird durch

Cross-Connections realisiert, die Merkmale von den Encoder-Schichten direkt zu den entsprechenden Decoder-Schichten weiterleiten. Diese Verbindungen ermöglichen die Wiederverwendung von Details und führen zu präziseren und detaillierteren Ausgaben (Ronneberger et al., 2015).

Die Abbildung A.1 im Anhang zeigt die Architektur des UNet nach Ronneberger et al. (2015). Eine schematische Darstellung des UNet mit Attention-Schichten, wie in Stable Diffusion, ist in Abbildung 3.3 dargestellt.

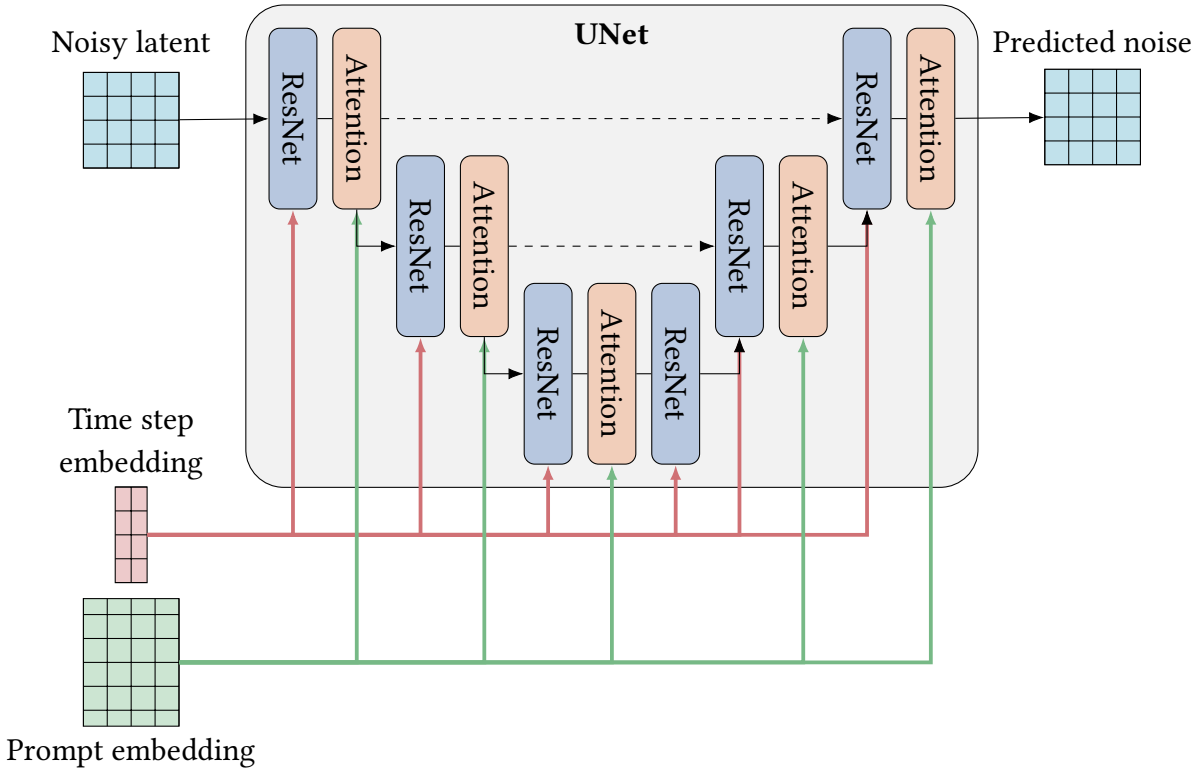

Abbildung 3.3: Schematische Darstellung des UNet in Stable Diffusion mit Attention-Schichten. Eigene Darstellung basierend auf Rombach et al., 2022 und Graph (2024).[14]

Funktionsweise

Während des Diffusionsprozesses wird ein verrauschtes Bild in den Encoder des UNet eingespeist. Der Encoder reduziert die räumliche Auflösung der Eingabe und komprimiert diese in eine latente Darstellung. Diese Darstellung wird dann durch den Decoder verarbeitet, um

[14] Graph, R. (2024, 22. Mai). *Diffusion Models: A Comprehensive High-Level Understanding*. Medium. Verfügbar 17. Juni 2024 unter https://medium.com/@researchgraph/diffusion-model-comprehensive-high-level-understanding-55d6ecad2cba

das Bild schrittweise zu rekonstruieren. Dabei wird das Rauschen iterativ reduziert, bis das ursprüngliche Bild wiederhergestellt ist. Dieser Prozess der iterativen Rauschreduktion wird durch den Scheduler gesteuert, der die Schrittweite und das Timing der Diffusionsschritte festlegt (Ho et al., 2020; Rombach et al., 2022).

Um den Generierungsprozess zu steuern, werden die vom UNet-Encoder extrahierten Merkmale mit den Texteingaben durch Cross-Attention-Schichten verknüpft und über die Cross-Connections an den Decoder weitergeleitet. Dies ermöglicht es dem Modell, relevante Informationen aus den Textbeschreibungen in die Bildgenerierung zu integrieren und sicherzustellen, dass die generierten Bilder den textuellen Eingaben entsprechen (Rombach et al., 2022).

Durch diese Architektur und Mechanismen trägt das UNet maßgeblich zur Fähigkeit von Stable Diffusion bei, qualitativ hochwertige und detaillierte Bilder aus textuellen Beschreibungen zu erzeugen.

3.1.4 Aufbau und Funktion von Stable Diffusion

Im Zusammenspiel mit den in Kapitel 3.1.3 beschriebenen Komponenten ermöglicht Stable Diffusion die Generierung hochauflösender Bilder aus textuellen Beschreibungen im Text-to-Image-Modus oder im Image-to-Image-Modus, bei dem zusätzlich zu den textuellen Beschreibungen auch Bilder als Eingabe verwendet werden (Rombach et al., 2022).

Der Prozess der Bildgenerierung umfasst mehrere Schritte, die im Folgenden erläutert werden. Abbildung 3.4 zeigt eine schematische Darstellung des Bildgenerierungsprozesses in Stable Diffusion.

Training: Vorwärtsdiffusion

Im Trainingprozess wird eine Vorwärtsdiffusion im latenten Raum durchgeführt, bei der einem Bild iterativ Rauschen hinzugefügt wird. Dieser Prozess wird modelliert durch eine Markov-Kette, bei der jedes Bild in eine zunehmend verrauschte Version überführt wird. Dadurch kann das Modell die bedingten Wahrscheinlichkeitsverteilungen der Bilder lernen, um so die Rauschreduktion im Generierungsprozess steuern zu können (Ho et al., 2020).

Text-Encoder: Text in latente Repräsentationen

Bei der Bildgenerierung mit Stable Diffusion wird zunächst die Texteingabe durch den Text-Encoder in latente Repräsentationen transformiert. Diese Text-Embeddings werden anschließend im UNet in den Cross-Attention Schichten zur Steuerung des Generierungsprozesses verwendet (Rombach et al., 2022).

Diffusion im latenten Raum: UNet und Scheduler

Die eigentliche Bildgenerierung erfolgt an dieser Stelle. Als Eingaben dienen ein zufällig verrauschtes Bild, die Text-Embeddings aus dem Text-Encoder sowie die Timestep-Embeddings des Schedulers. Im Image-to-Image-Modus wird ein zusätzliches Bild verwendet um daraus das verrauschte Eingabebild zu erzeugen (Rombach et al., 2022). Der Bildgenerierungsprozess kann beliebig viele Schritte durchlaufen, wobei in jedem Schritt dem Bild mehr Details hinzugefügt werden (Ho et al., 2020; Rombach et al., 2022).

Der Diffusionsprozess wird hierbei rückwärts durchlaufen. Im latenten Raum wird das verrauschte Bild iterativ durch das UNet verarbeitet. Dabei entfernt das UNet schrittweise das Rauschen, wobei dieser Prozess durch Cross-Attention-Mechanismen und den Scheduler gesteuert wird. Die Cross-Attention ermöglicht es dem Modell, die Text-Embeddings effektiv in die Bildgenerierung zu integrieren (Ho et al., 2020).

Jeder Schritt im Diffusionsprozess trägt zur Verfeinerung des Bildes bei, indem das Modell lernt, die relevanten Merkmale aus den Text-Embeddings mit den Bildmerkmalen zu kombinieren (Rombach et al., 2022).

Decodierung: Transformation in den Bildraum

Sobald das Bild im latenten Raum erstellt wurde, erfolgt die Decodierung, um ein hochauflösendes Bild zu erhalten. (Rombach et al., 2022).

Durch diese sorgfältig orchestrierten Schritte ist Stable Diffusion in der Lage, aus Textbeschreibungen realistische und qualitativ hochwertige Bilder zu generieren.

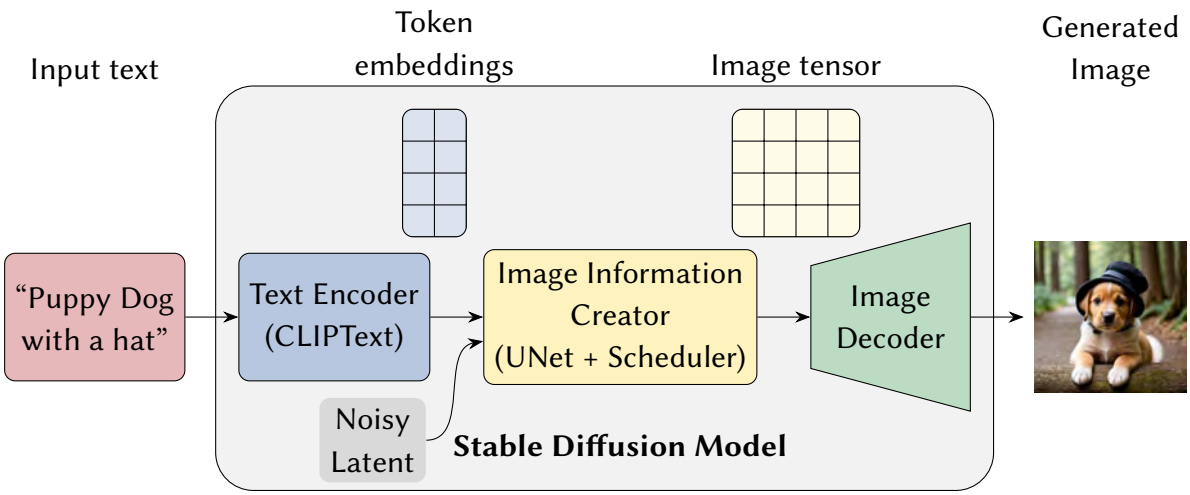

Abbildung 3.4: Überblick über den Stable Diffusion Generierungsprozess. Eigene Darstellung, angepasst von Shpuntov (2023).[15]

[15] Shpuntov, I. (2023, 5. Dezember). *Enhancing Video Creativity with AI: Stable Diffusion and AnimateDiff with*

3.1.5 Parameter und Einstellungen

Stable Diffusion bietet eine Vielzahl von Parametern und Einstellungen, die es ermöglichen, den Generierungsprozess zu steuern und zu optimieren. Diese Parameter beeinflussen die Qualität, Geschwindigkeit und Vielfalt der generierten Bilder und können je nach Anwendungsfall und Zielsetzung angepasst werden. Im Folgenden werden einige der wichtigsten Parameter und Einstellungen vorgestellt, die in Stable Diffusion verwendet werden und für diese Arbeit relevant sind. Leider werden diese Parameter in der Literatur wenig beschrieben, sodass die folgenden Informationen auf Erfahrungswerten und Empfehlungen von Anwendern basieren. In der Bibliographie finden sich zahlreiche Quellen, die sich mit der Anwendung und Optimierung von Stable Diffusion beschäftigen.

Seed
Der Seed ist ein Zufallswert, der den Startpunkt für die Generierung bestimmt und die Zufälligkeit der Ausgaben beeinflusst. Ein zufälliger Seed liefert stets unterschiedliche Ergebnisse, während ein fester Seed die Reproduzierbarkeit der Ergebnisse gewährleistet. So sollte der Seed festgelegt werden, um den Einfluss anderer Parameter auf die Generierung zu überprüfen und zu vergleichen.

Sampling-Steps
Die Sampling-Steps entsprechen der Anzahl der Denoising-Schritte, die während der Generierung durchgeführt werden. In der Regel gilt hier: Je höher die Anzahl der Schritte, desto mehr Details werden im generierten Bild sichtbar. Allerdings steigt auch die benötigte Rechenzeit. Abhängig vom Modell und Scheduler sind 20-45 Schritte ausreichend für gute Ergebnisse. Eine geringe Anzahl an Schritten ist hilfreich um schnell Parameter zu testen und die Auswirkungen auf die Generierung zu überprüfen. Sind gute Einstellungen gefunden, kann die Anzahl der Schritte erhöht werden, um detailliertere Ergebnisse zu erzielen.

CFG-Scale
Der Guidance-Scale bestimmt, wie stark der Text die Bildgenerierung beeinflusst. Ein höherer Wert bedeutet einen stärkeren Einfluss des Textes auf das generierte Bild. In der Regel führen Werte zwischen 4 und 8 zu guten Ergebnissen, abhängig von Model und Anwendung. Ein niedriger Wert (2-4) führt zu wenig textuellem Einfluss und gibt dem Modell mehr Freiheit bei der Generierung.

---

conditions... Medium. Verfügbar 13. Juni 2024 unter https://medium.com/@cultural.nickname/enhancing-video-creativity-with-ai-stable-diffusion-and-animatediff-with-conditions-1fb96592e3f1

Denoising-Strength
Dieser Parameter ist nur für die Image-toImage Generierung relevant. Der Denoising-Strength-Parameter bestimmt, wie stark das Eingabebild vor dem Denoisingprozess verrauscht wird. Ein niedriger Wert (0) fügt dem Bild kein Rauschen hinzu, sodass das gleiche Bild wieder ausgegeben wird. Ein hoher Wert (0,75) fügt viel Rauschen hinzu, sodass die Ausgabe stark vom Eingabebild abweicht, oder sogar ein völlig anderes Bild generiert wird. Abhängig von der Anwendung sind Werte zwischen 0,3 und 0,7 sinnvoll, um das Eingabebild als Grundlage für die Generierung zu nutzen und es zu variieren und zu manipulieren.

3.1.6 Fine-Tuning

Fine-Tuning ist der Prozess der Anpassung eines vortrainierten Modells, das auf einem umfangreichen Datensatz trainiert wurde, an eine spezifische Aufgabe. Dabei wird das Modell auf einem neuen, kleineren Datensatz weitertrainiert, um die Leistungsfähigkeit und Genauigkeit für spezifische Anwendungen zu verbessern. Fine-Tuning ist besonders nützlich, da es die Trainingszeit verkürzt und weniger Daten benötigt als das Training eines Modells von Grund auf (Sonnet, 2022).

3.1.6.1 Textual Inversion

Textual Inversion ist eine Fine-Tuning-Methode für Stable Diffusion, bei der nur die Gewichte des Text-Encoders angepasst werden (Gal et al., 2022). Der Trainingsprozess führt zur Einführung neuer Token, die spezifische Vektoren im Text-Encoder darstellen. Da nur diese spezifischen Vektoren angepasst werden, ist der Trainingsprozess schneller und ressourceneffizienter als ein vollständiges Neutraining des Modells. Die resultierende Datei ist nur wenige Kilobyte groß, da nur die neuen Embedding-Vektoren gespeichert werden müssen (Gal et al., 2022).

3.1.6.2 LoRA

Die Low-Rank Adaptation (LoRA) wurde ursprünglich für Large Language Models wie GPTs entwickelt (Hu et al., 2021) und wurde kürzlich auch für Stable Diffusion angepasst.[16] LoRA reduziert die Anzahl der zu trainierenden Parameter, indem es die großen Gewichtsmatrizen eines Modells in zwei kleinere, niedrigere Rangmatrizen zerlegt. Anstatt die gesamte große Gewichtsmatrix zu trainieren, werden nur diese kleineren Matrizen trainiert. Dies führt zu effizienteren Trainingsprozessen und ermöglicht es, Modelle schneller und mit weniger

[16] Ryu, S. (2024, 25. Mai). *Low-Rank Adaptation for Fast Text-to-Image Diffusion Fine-Tuning* (Version 0.0.1). Verfügbar 25. Mai 2024 unter https://github.com/cloneofsimo/lora

Rechenressourcen anzupassen. Dabei entstehen Modellgrößen von etwa 100 - 200 MB, was im Vergleich zu vollständigen Modellen eine erhebliche Reduzierung darstellt (Hu et al., 2021).

3.1.7 Frameworks für Stable Diffusion

Um Stable Diffusion effektiv nutzen zu können, stehen mehrere Frameworks zur Verfügung, die den Aufbau, das Training und die Anwendung der Modelle erleichtern. Folgend werden einige gängige Frameworks vorgestellt, die speziell für die Arbeit mit Stable Diffusion entwickelt wurden.

Diffusers

Das *Diffusers*-Framework von Hugging Face ist eine weitverbreitete Bibliothek, die speziell für die Arbeit mit Diffusionsmodellen entwickelt wurde. Es bietet eine einfach zu bedienende API für das Training und die Inferenz von Diffusionsmodellen und unterstützt eine Vielzahl von vortrainierten Modellen, die direkt eingesetzt werden können. Diffusers integriert sich nahtlos mit anderen beliebten Machine-Learning-Frameworks wie PyTorch und TensorFlow, was es Forschern und Entwicklern ermöglicht, schnell und effizient mit Diffusionsmodellen zu arbeiten.[17]

AUTOMATIC1111

Das *AUTOMATIC1111/stable-diffusion-webui*-Projekt bietet eine benutzerfreundliche Weboberfläche, die es ermöglicht, Stable Diffusion-Modelle einfach zu bedienen und damit zu experimentieren. Die Oberfläche ermöglicht die Text-to-Image, sowie die Image-to-Image Generierung und bietet eine umfangreiche Möglichkeit zur Anpassung der Parameter. Das Framework ist besonders nützlich für Benutzer, die eine visuelle und interaktive Methode zur Arbeit mit Diffusionsmodellen bevorzugen ohne tief in den Code eintauchen zu müssen. Für die programmatische Nutzung bietet AUTOMATIC1111 auch eine API, die es ermöglicht, die Funktionalitäten des Frameworks in eigene Anwendungen zu integrieren.[18]

Kohya_ss

bmaltais/kohya_ss ist ein weiteres Framework, das speziell für das Training und die Feinabstimmung von Stable Diffusion-Modellen entwickelt wurde und kann ebenfalls über eine Weboberfläche bedient werden. Es bietet umfassende Werkzeuge für die Anpassung von

[17] von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., Nair, D., Paul, S., Liu, S., Berman, W., Xu, Y., & Wolf, T. (2024, 30. Mai). *Diffusers: State-of-the-art Diffusion Models* (Version 0.12.1). Verfügbar 30. Mai 2024 unter https://github.com/huggingface/diffusers

[18] AUTOMATIC1111. (2022, August). *Stable Diffusion Web UI.* Verfügbar 30. Mai 2024 unter https://github.com/AUTOMATIC1111/stable-diffusion-webui

Hyperparametern, die Verwaltung von Trainingsdaten und die Optimierung der Modellarchitektur. Kohya_ss ist besonders nützlich für fortgeschrittene Anwender, die maßgeschneiderte Modelle entwickeln und detaillierte Kontrolle über den Trainingsprozess haben möchten.[19]

DiffuGen

Das Framework *mshenoda/diffugen*[20] basiert auf der DiffUsers-Libary und bietet einen flexiblen Ansatz zur Generierung von Datensätzen für die Bilderkennung. Mittels Prompt-Templating können Text-Prompts erstellt und zur Text-to-Image und Image-to-Image Generierung verwendet werden. Das Framework ermöglicht die Anpassung von Modellen durch Textual Inversion und die Erstellung von segmentierten Labeln durch die Nutzung von Cross-Attention Heatmaps[21] (Shenoda & Kim, 2023).

3.2 Bilderkennung und Bilderkennungsmethoden

Bilderkennung ist ein wesentlicher Bereich der Informatik, der sich mit der automatischen Analyse und Interpretation von digitalen Bildern durch Computer befasst. Dabei werden Techniken der Bildvorverarbeitung, Merkmalserkennung und Objekterkennung eingesetzt, um visuelle Daten zu analysieren und zu klassifizieren (Paaß & Hecker, 2020). Anwendungen finden sich in zahlreichen Bereichen wie der Medizin zur Diagnose von Krankheiten (Chen et al., 2022; Paaß & Hecker, 2020), in der Sicherheit zur Gesichtserkennung (Paaß & Hecker, 2020) und in der Automobilindustrie für autonome Fahrzeugsysteme (Dong & Cappuccio, 2023). Fortschritte in maschinellem Lernen und neuronalen Netzen haben die Genauigkeit und Effizienz der Bilderkennung erheblich verbessert. Diese Technologien eröffnen vielfältige neue Möglichkeiten zur Lösung komplexer visueller Aufgaben und erweitern kontinuierlich das Spektrum potenzieller Anwendungen (Paaß & Hecker, 2020).

3.2.1 Bilderkennung mittels neuronalen Netzen

Abseits der klassischen, algorithmischen Bilderkennungsmethoden haben sich Neuronale Netze, insbesondere Convolutional Neural Networks (CNNs), als äußerst effektive Werkzeuge für die Bilderkennung etabliert. CNNs sind aufgrund ihres Aufbaus mit Convolutional-, Pooling- und Fully-Connected-Schichten speziell für die Verarbeitung von Bilddaten optimiert.

[19] bmaltais. (2024, 30. Mai). *Bmaltais/Kohya_ss*. Verfügbar 30. Mai 2024 unter https://github.com/bmaltais/kohya_ss

[20] Shenoda, M. (2024, 21. April). *Mshenoda/Diffugen*. Verfügbar 8. Juni 2024 unter https://github.com/mshenoda/diffugen

[21] Visualisierung der Aufmerksamkeit je Token der Cross-Attention-Layer des Unets.

Sie ermöglichen die Extraktion komplexer visueller Muster und die Klassifikation von Bildern mit hoher Genauigkeit und Effizienz (Paaß & Hecker, 2020; Simonyan & Zisserman, 2015).

Die genaue Funktionsweise von CNNs und deren Architekturen sind in zahlreichen Büchern und wissenschaftlichen Arbeiten ausführlich beschrieben und bieten einen umfassenden Einblick in die Techniken und Methoden der Bilderkennung mittels neuronaler Netzen. Einen guten Überblick bietet beispielsweise das Buch Paaß und Hecker (2020).

3.2.2 Datensätze und Annotationsmethoden

Für das Training von Bilderkennungsmodellen sind umfangreiche Datensätze und präzise Annotationsmethoden unerlässlich. Die Datensätze müssen vielfältige Bilder enthalten, um die Modelle auf unterschiedliche Szenarien vorzubereiten und die Generalisierungsfähigkeit zu verbessern, sodass sie auch in realen undungekannten Situationen robust arbeiten können. Genaue und umfassende Annotationen beziehungsweise Labels sind ebenso entscheidend, da sie die Lernbasis der Modelle bilden.

Einige umfangreiche Datensätze sind zum Beispiel *ImageNet* (Deng et al., 2009), *COCO (Common Objects in Context)* (Lin et al., 2015) und *Open Images* (Kuznetsova et al., 2020). Diese Datensätze enthalten Millionen von Bildern in tausenden Kategorien und bieten umfassende Annotationen. Viele vortrainierte Modelle, wie z.B. *ResNet50* oder *InceptionV3*, basieren auf diesen Datensätzen und können für spezifische Anwendungen weiter trainiert oder feinabgestimmt werden (Kapitel 3.2.3).

Die Annotationen in Bilderkennungsdatensätzen sind entscheidend für das Training und die Evaluierung von Modellen. Sie ermöglichen es den Modellen, Objekte in Bildern zu identifizieren und zu klassifizieren. Ein Datensatz kann auf verschiedene Arten annotiert werden, die auch kombiniert vorliegen können.

- **Labels**: Ein Bild wird einer oder mehreren Klassen zugeordnet, um die Objekte oder Szenen im Bild zu identifizieren. Dies ist die einfachste Form der Annotation, da sich das Label auf das gesamte Bild bezieht (Paaß & Hecker, 2020).
- **Bounding Boxes**: Bei dieser Methode wird ein rechteckiger Rahmen um ein Objekt im Bild gezeichnet, um es zu markieren und zu lokalisieren. Bounding Boxes sind weit verbreitet und einfach zu implementieren (Paaß & Hecker, 2020).
- **Segmentierungsmasken**: Jedes Pixel eines Objekts wird markiert, um die genaue Form und Position des Objekts zu bestimmen. Segmentierungsmasken sind besonders nützlich für detaillierte Bildanalysen, erfordern jedoch einen größeren Aufwand bei der Annotation. Intelligente Methoden können die Segmentierung automatisieren (Paaß & Hecker, 2020).

3.2.3 Trainingsmethoden

Da das Training von Bilderkennungsmodellen von Grund auf zeitaufwändig und rechenintensiv ist und umfangreiche Datensätze erfordert, nutzt man Trainingsmethoden wie Transfer Learning, um ein Bilderkennungsmodell für spezifische Aufgaben anzupassen. Dazu nutzt man bereits vortrainierte Modelle, die auf großen Datensätzen trainiert wurden und eine gute Generalisierungsfähigkeit besitzen. Solche Modelle haben bereits gelernt, visuelle Merkmale zu extrahieren, sodass sie auf dieser Basis für spezifische Aufgaben weitertrainiert werden können (Sonnet, 2022).

- **Fine-Tuning**: Beim Fine-Tuning wird ein vortrainiertes Modell auf einen neuen Datensatz angepasst, indem die Gewichte des Modells weitertrainiert werden. Dies ermöglicht eine flexible und effiziente Anpassung des Modells bei vergleichsweise geringem Trainingsaufwand.[22]
- **Feature Extraction**: Bei der Feature-Extraktion werden die Gewichte des vortrainierten Modells eingefroren und nur die obersten Schichten des Modells für die spezifische Aufgabe weitertrainiert. Dadurch bleibt die vortrainierte Merkmalsextraktion erhalten und die oberen Schichten werden auf die neue Klassifikationsaufgabe angepasst. Das macht diese Methode etwas weniger flexibel als das Fine-Tuning, reduziert jedoch weiter den Trainingsaufwand.[23]

3.2.4 Leistungsmetriken

Verschiedene Metriken und Evaluierungsmethoden werden verwendet, um die Leistung von Bilderkennungsmodellen zu bewerten. Dazu gehören:

- **Loss**: Eine Kennzahl, die die Differenz zwischen den vorhergesagten und den tatsächlichen Werten misst und während des Trainings minimiert wird. Der Loss-Wert gibt an, wie gut das Modell an die Trainingsdaten angepasst ist, und ist ein wichtiger Indikator für die Modellleistung (Goodfellow et al., 2016).
- **Accuracy**: Der Anteil der korrekt klassifizierten Bilder im Verhältnis zur Gesamtzahl der Bilder. Die Accuracy ist eine einfache und weit verbreitete Metrik zur Bewertung der Modellleistung. Sie gibt an wie viele der vorhergesagten Klassen korrekt sind (Goodfellow et al., 2016).

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.2}$$

[22] Chilamkurthy, S. (n. d.). *Transfer Learning for Computer Vision Tutorial — PyTorch Tutorials*. Verfügbar 5. Juni 2024 unter https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html

[23] Chilamkurthy, S. (n. d.). *Transfer Learning for Computer Vision Tutorial — PyTorch Tutorials*. Verfügbar 5. Juni 2024 unter https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html

- **Recall**: Der Anteil der korrekt positiv klassifizierten Bilder im Verhältnis zur Gesamtzahl der tatsächlich positiven Bilder. Der Recall gibt an, wie viele der tatsächlich positiven Bilder korrekt erkannt wurden. Der Recall ist besonders relevant für Anwendungen, bei denen die Identifikation aller positiven Fälle wichtig ist und falsch positive Ergebnisse akzeptabel sind, z.B. bei medizinischen Diagnosen (Goodfellow et al., 2016).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.3}$$

- **Precision**: Der Anteil der korrekt positiv klassifizierten Bilder im Verhältnis zur Gesamtzahl der positiv klassifizierten Bilder. Die Precision gibt an, wie viele der positiv klassifizierten Bilder tatsächlich korrekt sind. Die Precision ist besonders relevant für Anwendungen, bei denen falsch positive Ergebnisse vermieden werden müssen, z.B. bei der Erkennung von Spam-E-Mails (Goodfellow et al., 2016).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.4}$$

- **Konfusionsmatrix**: Eine Matrix, die die Anzahl der korrekten und falschen Klassifikationen für jede Klasse darstellt. Die Konfusionsmatrix ist ein nützliches Werkzeug zur Analyse der Modellleistung je Klasse.

Diese Metriken bieten die erforderlichen Informationen, um die Leistung von Bilderkennungsmodellen sowohl im Trainingsverlauf als auch im Testprozess zu bewerten und zu optimieren.

4 Methoden / Experimentaufbau

Um die Forschungsfrage präzise und umfassend beantworten zu können, wurde ein Experiment durchgeführt. Experimente bieten kontrollierte Bedingungen, die es ermöglichen, Variablen gezielt zu variieren und zu untersuchen. Dies erlaubt eine genaue und detaillierte Datensammlung, die für die Analyse unerlässlich ist.

Zudem gewährleisten Experimente durch eine präzise Dokumentation und Transparenz der Methodik die Reproduzierbarkeit und Validität der Ergebnisse. Dies ist entscheidend, um wissenschaftliche Standards zu erfüllen und eine Vergleichbarkeit mit anderen Studien sicherzustellen. Darüber hinaus tragen Experimente zur Objektivität der Forschung bei, da sie systematische und unvoreingenommene Daten erfassen. Experimente bieten auch eine hohe praktische Relevanz, da sie konkrete Anwendungen und Lösungen für reale Probleme liefern können.

Daher soll in einem Experiment untersucht werden, wie Stable Diffusion genutzt werden kann, um einen realistischen und umfassenden Datensatz für die Schulung von Bilderkennungsmodellen zur effektiven Bewertung der Sauberkeit von Sanitärbereichen zu erstellen. Es soll gezeigt werden, welche Schritte und Methoden notwendig sind, um einen solchen Datensatz zu generieren. Dazu werden verschiedene Techniken und Modelle verwendet, die im Folgenden detailliert beschrieben werden. Anschließend sollen die generierten Datensätze zum Training eines Bilderkennungsmodells verwendet werden, um daraus Erkenntnisse über die Effektivität und Genauigkeit der Sauberkeitserkennung zu gewinnen.

Dieses Kapitel beschreibt den methodischen Rahmen und den detaillierten Aufbau des Experiments. Angewandte Techniken und getroffene Entscheidungen sollen transparent dargelegt werden, um die Reproduzierbarkeit und Nachvollziehbarkeit der Ergebnisse zu gewährleisten.

4.1 Experimentbedingungen

Das Festlegen der Rahmenbedingungen für das Experiment ist entscheidend, um die Reproduzierbarkeit und Validität der Ergebnisse zu gewährleisten. Durch klare Definition der Variablen und Bedingungen können Verzerrungen minimiert und die Zuverlässigkeit der

Ergebnisse erhöht werden. Dies bildet die Grundlage für valide und belastbare Schlussfolgerungen in der wissenschaftlichen Forschung.

Im folgenden werden die experimentellen Rahmenbedingungen und die verwendeten Techniken detailliert beschrieben und die Gründe für ihre Auswahl erläutert. Dies umfasst die Auswahl des Objekts, der Klassen und des Stable Diffusion Modells. Durchgeführt wurde das Experiment auf einem leistungsstarken Konsumer-PC. Die Hardwarekonfiguration umfasste eine NVIDIA GeForce RTX 3060 Grafikkarte mit 12GB VRAM, einen AMD Ryzen 5600x Prozessor und 32 GB RAM.

4.1.1 Auswahl des Objekts

Die Entscheidung, das Experiment auf die Generierung eines einzigen Objekts zu beschränken, beruht auf mehreren wesentlichen Gründen. Zunächst ermöglicht diese Einschränkung eine bessere Kontrolle der Variablen, die die Genauigkeit und Präzision der Sauberkeitserkennung beeinflussen könnten. Durch die Konzentration auf ein einzelnes Objekt können wir den Trainingsprozess des Stable Diffusion Modells und die anschließende Feinabstimmung des Bilderkennungsmodells vereinfachen und optimieren.

Die Konzentration auf ein einzelnes Objekt reduziert die Komplexität des Trainingsprozesses für das Stable Diffusion Modell sowie die nachfolgende Feinabstimmung des Bilderkennungsmodells. Durch die Beschränkung auf Urinale können spezifische Eigenschaften und Verschmutzungsarten detailliert analysiert und klassifiziert werden, was bei einer breiteren Objektauswahl schwieriger zu erreichen wäre. Zudem minimiert diese Beschränkung systematische Fehlerquellen, die bei der Analyse mehrerer unterschiedlicher Objekte entstehen könnten. Dadurch wird gewährleistet, dass die Auswertung der Bilderkennung klar auf die Sauberkeit fokussiert bleibt, ohne dass ein Bias durch verschiedene Objekte entsteht.

Die Auswahl stützt sich auch auf die Forschungsarbeit von Jayasinghe et al. (2019), die sich speziell mit der Klassifizierung der Sauberkeit von Pissoirs befasst. Die Studie liefert wertvolle Einblicke in die spezifischen Herausforderungen und Lösungsansätze, die auch für diese Arbeit relevant sind. Durch die Anlehnung an etablierte Forschungsarbeiten wird sichergestellt, dass die experimentellen Methoden auf einer soliden wissenschaftlichen Grundlage stehen und die Ergebnisse mit vorhandenen Erkenntnissen vergleichbar sind.

4.1.2 Auswahl der Klassen

Die Festlegung der Verschmutzungsklassen, die das Bilderkennungsmodell klassifizieren soll, ist ein wesentlicher Bestandteil des Forschungsdesigns. In Anlehnung an die Arbeit von Jayasinghe et al. (2019) wurden die Klassen *sauber*, *mittel* und *dreckig* (clean, average, dirty)

gewählt. Diese Klassifizierung stellt einen guten Kompromiss zwischen einer differenzierten Bewertung der Sauberkeit und einer angemessenen Einfachheit für maschinelle Trainings- und Generierungsprozesse dar.

Die Einteilung in drei Klassen reduziert die Komplexität der Datensatzgenerierung, indem nur zwischen diesen drei Stufen unterschieden werden muss, anstatt eine feinere Abstufung vorzunehmen. Diese Vereinfachung ist vorteilhaft, da sie weniger Varianz innerhalb der Daten erfordert, was die Datensatzgenerierung vereinfacht. Gleichzeitig bietet die Klasseneinteilung eine ausreichende Differenzierung, um die verschiedenen Zustände des Reinigungsbedarfs gezielt zu erkennen.

4.1.3 Auswahl des Stable Diffusion Modells

Da Stable Diffusion in einer Vielzahl von Modellen und Versionen verfügbar ist, musste eine Auswahl getroffen werden, die den Anforderungen des Experiments entspricht.

4.1.3.1 Auswahl des Basismodells

Zunächst wurde das Basismodell von Stable Diffusion ausgewählt. Die verfügbaren Optionen umfassten die Versionen 1.5, 2.0 und Stable Diffusion XL (SDXL). Viele spezialisierte Modelle (Abschnitt 4.1.3.2), die auf Plattformen wie *Hugging Face*[1] und *Civitai*[2] verfügbar sind, basieren auf Stable Diffusion 1.5 oder SDXL, wie beispielsweise die bekannten Modelle *Realistic Vision*[3], *Dreamshaper*[4] oder *Cyberrealistic*[5].

Daher beschränkte sich die Auswahl auf die Versionen 1.5 und SDXL. SDXL bietet eine höhere Auflösung (1024 x 1024) und damit eine bessere Bildqualität. Dadurch und aufgrund der Grundstruktur von SDXL (Podell et al., 2023) benötigt es aber auch mehr Rechenleistung, was den Einsatz auf Konsumer-Hardware wie der RTX 3060, insbesondere für das Training

[1] huggingface.co

[2] civitai.com

[3] **SD1.5:** *Realistic Vision V6.0 B1 - V5.1 Hyper (VAE) | Stable Diffusion Checkpoint | Civitai.* (2024, 31. Mai). Verfügbar 7. Juni 2024 unter https://civitai.com/models/4201/realistic-vision-v60-b1
SDXL: *RealVisXL V4.0 - V4.0 Lightning (BakedVAE) | Stable Diffusion Checkpoint | Civitai.* (2024, 25. Mai). Verfügbar 7. Juni 2024 unter https://civitai.com/models/139562/realvisxl-v40

[4] **SD1.5:** *DreamShaper - 8 | Stable Diffusion Checkpoint | Civitai.* (2024, 9. Mai). Verfügbar 7. Juni 2024 unter https://civitai.com/models/4384/dreamshaper
SDXL: *DreamShaper XL - v2.1 Turbo DPM++ SDE | Stable Diffusion Checkpoint | Civitai.* (2024, 5. Mai). Verfügbar 7. Juni 2024 unter https://civitai.com/models/112902/dreamshaper-xl

[5] **SD1.5:** *CyberRealistic - v5.0 | Stable Diffusion Checkpoint | Civitai.* (2024, 30. Mai). Verfügbar 7. Juni 2024 unter https://civitai.com/models/15003/cyberrealistic
SDXL: *CyberRealistic XL - v1.1 (VAE) | Stable Diffusion Checkpoint | Civitai.* (2024, 23. März). Verfügbar 7. Juni 2024 unter https://civitai.com/models/312530/cyberrealistic-xl

von Stable Diffusion Modellen, erschwert. Daher fiel die Wahl auf das Stable Diffusion Basismodell 1.5, das aufgrund der großen Anzahl verfügbarer spezialisierter Modelle und der effizienten Performance auf Konsumer-Hardware als geeignet erschien.

4.1.3.2 Auswahl des spezialisierten Modells

Spezialisierte Modelle bieten die Möglichkeit, spezifische Anpassungen und Erweiterungen vorzunehmen, um die Generierung von Bildern für bestimmte Anwendungsfälle zu ermöglichen. Diese Modelle sind auf bestimmte Konzepte oder Anforderungen spezialisiert und bieten eine höhere Genauigkeit und Relevanz für die Bildgenerierung in spezifischen Bereichen.

Die Wahl des spezialisierten Modells fiel auf *Realistic Vision 5.1*.[6] Dieses Modell ist besonders für die Generierung photorealistischer Bilder bekannt, was eine entscheidende Eigenschaft für die Authentizität des zu erstellenden Datensatzes ist. Bei der Auswahl wurden verschiedene populäre Modelle auf der Plattform CivitAI verglichen, darunter Dreamshaper, Cyberrealistic und Realistic Vision. Realistic Vision 5.1 wurde aufgrund seiner hohen Bewertungen und der umfangreichen Downloadzahlen ausgewählt. Es ist bereits auch eine neuere Version (6.0) verfügbar, die sich allerdings noch in der Beta-Phase befindet und somit für den produktiven Einsatz nicht geeignet ist.

Die Entscheidung wurde durch die Tatsache gestützt, dass das Framework *DiffuGen* von Shenoda und Kim (2023) eine ältere Version dieses Modells, Realistic Vision 4.0, zur Datensatzgenerierung verwendet hat und damit gute und realistische Ergebnisse erzielt hat. Dies zeigt, dass Realistic Vision Modelle für die Generierung von realistischen Bildern geeignet sind, was für die Erstellung eines authentischen Datensatzes von entscheidender Bedeutung ist.

4.2 Datensatzgenerierung

4.2.1 Stable Diffusion Fine-Tuning

Da mit den vortrainierten Stable Diffusion Modellen (Stable Diffusion 1.5 und Realistic Vision 5.1) keine realistischen Urinale und Verschmutzungen erzeugt werden konnten und auch keine geeigneten spezialisierten Modelle zur Verfügung standen, war es notwendig, ein eigenes Modell als Erweiterung zu Realistic Vision 5.1 zu trainieren. Dafür wurde mit Hilfe des

[6] *SG161222/Realistic_Vision_V5.1_noVAE · Hugging Face*. (2023, 4. Dezember). Verfügbar 15. Mai 2024 unter https://huggingface.co/SG161222/Realistic_Vision_V5.1_noVAE

Kohya_ss Frameworks ein Fine-Tuning-Prozess durchgeführt, um mittels der LoRA-Technik das Basismodell an die spezifischen Anforderungen anzupassen.

An dieser Stelle wurde sich entschieden, separate Anpassungen für das Objekt (Urinal) und den Zustand (Verschmutzung) zu trainieren. Dies vereinfacht zunächst den Trainingsprozess, da die Modelle nur *ein* neues Konzept lernen müssen und auf die jeweiligen Merkmale fokussiert werden können. Es ermöglicht zudem, die Modelle flexibel anzupassen, zu verbessern, zu erweitern und mit verschiedenen Gewichtungen miteinander zu kombinieren, sodass Änderungen an einem Modell nicht zwangsläufig Anpassungen am anderen Modell erfordern. Die Abbildung 4.1 zeigt, dass das untrainierte Realistic Vision 5.1 Modell nicht in der Lage ist, Urinale zu generieren. Mit den trainierten LoRA-Modellen ist es hingegen möglich, Urinale und Verschmutzungen zu generieren.

Abbildung 4.1: Vergleich zwischen einem untrainierten Modell (links) und spezialisierten LoRA-Modellen (rechts) für die Generierung von Urinalen und Verschmutzungen. Eigene Darstellung

4.2.1.1 Bildersuche

Um Stable Diffusion trainieren zu können, ist es notwendig, entsprechende Trainingsbilder zu sammeln, die die gewünschten Konzepte (Objekte oder Style/Zustände) abbilden. Da getrennte Modelle für das Objekt und den Zustand trainiert werden, wurden spezifische Bilder für Urinale und Verschmutzungen gesucht. Folgende Kriterien wurden bei der Bildersuche berücksichtigt:

- **Diversität**: Bilder sollten eine Vielzahl von Urinalen und Verschmutzungen in verschiedenen Umgebungen, Perspektiven und Lichtverhältnissen zeigen, um die Robustheit des Modells zu gewährleisten.

- **Relevanz**: Die Bilder sollten spezifisch für die Anwendung in Sanitärbereichen sein, um eine hohe Genauigkeit und Relevanz der generierten Bilder zu gewährleisten.
- **Qualität**: Die Bildqualität sollte hoch genug sein, um eine effektive Generierung realistischer Bilder zu ermöglichen.
- **Lizenzfreiheit**: Die Bilder sollten lizenzfrei und frei verfügbar sein, um rechtliche Probleme zu vermeiden.
- **Relevanz für das Objekt**: Die Bilder sollten nur einen Typ von Urinalen zeigen, um das Modell auf diese spezifische Ausführung des Objekts zu trainieren. In diesem Fall wurde sich auf rundliche, ovale Urinale konzentriert.

Die Bildersuche erfolgte hauptsächlich über Google und andere lizenzfreie Bilddatenbanken.[7] Dabei wurden verschiedene Suchbegriffe verwendet, um eine breite Palette von Bildern zu erhalten, die die oben genannten Kriterien erfüllen. Die ausgewählten Bilder wurden dann manuell überprüft und für die Verwendung im Trainingsprozess ausgewählt.

4.2.1.2 Datensatzvorbereitung

Für das LoRA-Training müssen die gesammelten Trainingsbilder entsprechend aufbereitet werden. Dabei ist es wichtig, dass die Bilder mit einer Beschreibung versehen werden, auf die sich das Modell während des Trainings bezieht, ähnlich dem Prompt bei der Bildgenerierung. Zur Unterstützung wurde die Option *WD14 Captioning* in Kohya_ss für die Erstellung der Beschreibung verwendet. Konkret wurde das Image-to-Text Modell *SmilingWolf/wd-v1-4-convnextv2-tagger-v2*[8] verwendet, um automatisch Beschreibungen für die Bilder zu generieren. Die Beschreibungen werden als Textdateien zusammen mit den Bildern in einem Ordner abgelegt, damit sie für das Training zu verwendet werden können. Diese Beschreibungen wurden anschließend manuell überprüft und gegebenenfalls angepasst, um sicherzustellen, dass sie das Bild korrekt beschreiben und die spezifischen Merkmale und Details des Bildes erfassen. Die Software *Booru Dataset Tag Manager*[9] wurde verwendet, um die Beschreibungen zu bearbeiten und zu organisieren.

Da Kohya_ss die Bilder während des Trainings automatisch skaliert, war es nicht notwendig, die Bilder vorher auf ein bestimmtes Format zu skalieren.[10]

[7] Pexels.com, Pixabay.com, Unsplash.com, Flickr.com

[8] *SmilingWolf/Wd-v1-4-Convnextv2-Tagger-v2 · Hugging Face*. (n. d.). Verfügbar 11. Juni 2024 unter https://huggingface.co/SmilingWolf/wd-v1-4-convnextv2-tagger-v2

[9] starik222. (2024, 11. Juni). *Starik222/BooruDatasetTagManager*. Verfügbar 11. Juni 2024 unter https://github.com/starik222/BooruDatasetTagManager

[10] *LoRA Training Parameters · Bmaltais/Kohya_ss Wiki*. (n. d.). Verfügbar 11. Juni 2024 unter https://github.com/bmaltais/kohya_ss/wiki/LoRA-training-parameters

4.2.1.3 Stable Diffusion Training

Schließlich wurde der Trainingsprozess für die LoRA-Modelle durchgeführt. Dabei entstanden die LoRAs „urinal-LoRA“, welches auf die Generierung von Urinalen spezialisiert ist, und „dirty-LoRA“, welches auf die Generierung von Verschmutzungen spezialisiert ist. Diese basieren beide auf dem Basismodell Stable Diffusion 1.5. Der Vorteil der LoRA-Technik besteht darin, dass die Anpassungen auf jedes Modell angewendet werden können, das auf demselben Basismodell basiert.

Die Abbildung 4.1 auf Seite 29 zeigt die Einbindung der trainierten LoRA-Modelle in die Text-Prompts mittels des Ausdrucks `‹lora:urinal_LoRA_v2-000007:0.8›` bzw. `‹lora:dirtyStyle_LoRA_v2-000008:0.4›`. Die Epoche des Modells und die Gewichtung, die den Einfluss des Modells auf die Bildgenerierung bestimmt, können dabei ausgewählt werden. Die erstellten Modell-Checkpoints werden dazu im entsprechenden Modellordner abgelegt und können dann für die Bildgenerierung verwendet werden.

4.2.2 Framework zur Datensatzgenerierung

Für die automatisierte Bild- und Datensatzgenerierung wurde ein eigenes pythonbasiertes Framework entwickelt, das auf verschiedenen Komponenten basiert. Ein Jupyter Notebook mit einfacher UI ermöglicht die Bedienung der Python-Skripte mit benutzerdefinierten Parametern ohne Commandozeilenkenntnisse. Die Abbildung A.2 zeigt ein UML-Diagramm des Frameworks, das die verschiedenen Komponenten und ihre Interaktionen veranschaulicht.

4.2.2.1 Prompt Templating

Um einen diversen Datensatz zu erstellen, ist es essentiell, eine Vielzahl unterschiedlicher Prompts zu generieren, die verschiedene Aspekte des Bildes variieren, wie Objekttyp, Material, Hintergrund und Perspektive. Prompt-Templates können genutzt werden, um eine ausführliche Liste von Prompts zu erstellen. Diese Templates enthalten Platzhalter für verschiedene Aspekte wie Objekttyp, Material, Hintergrund und Perspektive. Diese Platzhalter werden dann mit definierten Begriffen gefüllt, um eine Vielzahl von Prompts zu erzeugen.

Als Beispiel für ein Prompt-Template zeigt der Codeblock A.1 im Anhang das Template, welches im Experiment zur Erstellung der Prompt-Liste verwendet wurde.

4.2.2.2 Automatisierte Bildgenerierung

Die beliebte AUTOMATIC1111 Web-UI für Stable Diffusion (siehe Abschnitt 3.1.7) bietet eine API, um mit Stable Diffusion zu interagieren. Diese API wurde genutzt, um die Bildgenerierung zu automatisieren. Großer Vorteil bei dieser Methode ist, dass die gleichen Einstellungen und Modelle wie in der WebUI verwendet werden können, was es einfach macht Einstellungen zu testen und dann für die Automatisierung zu übernehmen.

Die Bildgenerierung erfolgt in mehreren Schritten. Zunächst werden die generierten Prompts verwendet, um mit Hilfe der Text-To-Image Bildgenerierung (saubere) Urinale zu generieren. Dabei wird für jede Generierung ein zufälliger Seed verwendet. Anschließend werden diese generierten Bilder verwendet, um mit Hilfe der Image-To-Image Bildgenerierung, dem selben Seed und dem gleichen Prompt die Verschmutzungen hinzufügen. Der Prompt wird dabei leicht angepasst und die Gewichtung für das trainierte dirty-LoRA für Verschmutzungen wird angepasst.

So entstehen pro Prompt drei Bilder im Format 512 x 512 Pixel, jeweils eines für die Klassen sauber, mittel, dreckig. Die Bilder werden für die Weiterverarbeitung entsprechend abgespeichert.

4.2.2.3 Segmentierung

Optional kann eine Segmentierung durchgeführt werden, um die generierten Objekte vom Hintergrund zu trennen. Dies kann die Bilderkennung erleichtern, da die Klassifizierung der Verschmutzungsgrade unabhänging vom Hintergrund wird.

Zur Segmentierung wird das Plugin *Segmentate Anything (SAM)*[11] für die AUTOMATIC1111 Web-UI verwendet. Die Segmentierung kann ebenfalls über die AUTOMATIC1111 API gesteuert werden. Das Plugin nutzt dabei *GroundingDino*[12] um die Objekte zu erkennen.

4.2.2.4 Erstellung des Datensatzes

Abschießend können die generierten Bilder in Trainings-, Validierungs- und Testsets aufgeteilt und in einer geeigneten Ordnerstruktur abgelegt werden, um sie für das Training des Bilderkennungsmodells zu verwenden.

[11] *Facebookresearch/Segment-Anything*. (2024, 17. Juni). Verfügbar 17. Juni 2024 unter https://github.com/facebookresearch/segment-anything

[12] *IDEA-Research/GroundingDINO*. (2024, 17. Juni). Verfügbar 17. Juni 2024 unter https://github.com/IDEA-Research/GroundingDINO

Es sollen mehrere Datensätze generiert werden, um verschiedene Aspekte der Bilderkennung zu untersuchen. Dazu werden verschiedene Parameter und Einstellungen variiert, um die Auswirkungen auf die Generierung und die Leistung des Bilderkennungsmodells zu untersuchen.

4.3 Training des Bilderkennungsmodells

Da das Hauptaugenmerk des Experiments auf der Datensatzgenerierung mit Stable Diffusion liegt und das Bilderkennungsmodell nur zur Evaluierung der generierten Daten verwendet wird, wurden die Trainingsparameter und -methoden festgelegt, um den Einfluss verschiedener Parameter in der Bildgenerierung untersuchen zu können.

4.3.1 Modellauswahl

Die Auswahl des geeigneten CNN-Modells ist entscheidend für die Leistung des Bilderkennungssystems. Nach sorgfältiger Überlegung und Evaluierung verschiedener Modelle wie ResNet50 und InceptionV3 fiel die Entscheidung auf InceptionV3. Dieses Modell wurde aufgrund seiner allgemein hohen Genauigkeit und Effizienz ausgewählt. Die Arbeit von Devi et al. (2023) zeigt, dass InceptionV3 andere vortrainierte Modelle wie ResNet50, SqueezeNet und DenseNet-121 in Genauigkeit und Geschwindigkeit übertrifft. Dazu wurden diese vortrainierten Modelle auf dem PASCAL-VOC12 Datensatz getestet und evaluiert. In der Arbeit von Jayasinghe et al. (2019) wurde InceptionV3 erfolgreich für die Klassifizierung der Sauberkeit von Pissoirs eingesetzt. Daher wurde InceptionV3 als Grundlage für das Bilderkennungsmodell gewählt.

4.3.2 Transfer Learning

Wie in Kapitel 3.2.3 beschrieben, gibt es verschiedene Ansätze um ein vortrainiertes Bilderkennungsmodell anzupassen. Für die Anpassung des vortrainierten InceptionV3 Modells wurde *Fine-Tuning* als Transfer-Learning-Technik verwendet, da diese Methode eine bessere anpassungsfähigkeit ermöglicht.

4.3.3 Dataset Augmentation

Dataset Augmentation hilft, die Vielfalt der Trainingsdaten zu erhöhen und das Modell robuster gegen Überanpassung zu machen. Die Auswahl dieser Methoden basiert auf ihrer

Fähigkeit, verschiedene Variationen der Bilder zu erzeugen, die in realen Szenarien auftreten können. Die Augmentationstechniken sollen vor allem Variationen einbringen, die in der Datensatzgenerierung nicht oder nicht ausreichend umzusetzen waren. Die Tabelle 4.3.3 zeigt die verwendeten Augmentationstechniken und für welche Zwecke sie eingesetzt wurden.

Methode	**Zweck**	**Parameter**
Resize	Bildgröße skalieren, falls nötig	([512,512])
ToTensor	Konvertierung in Tensor	-
Normalize	Normalisierung der Pixelwerte	([mean, std])
RandomHorizontalFlip	Perspektiven variieren	(p=0.5)
RandomRotation	Rotation / Orientierung variieren	(degrees=30)
GaussianBlur	simuliere unfokussierte Bilder	(kernel_size=3)

Tabelle 4.1: CNN-Trainigsdatensatz Augmentation Methoden

4.3.4 Trainingsprozess

Für den Trainingsprozess wurde unter Verwendung der Frameworks PyTorch und Torchvision eine Trainingspipeline entwickelt, die es ermöglicht, das Modell mit den generierten Datensätzen zu trainieren. Die Tabelle 4.2 zeigt die verwendeten Trainingsparameter, die für das Training des Bilderkennungsmodells festgelegt wurden.

Parameter	**Wert**
Batch-Size	16
maximale Anzahl an Epochen	100
Learning Rate	0,0015
Learning Rate Scheduler	ReduceLROnPlateau (mode='min', factor=0.1, patience=3)
Loss Funktion	CrossEntropyLoss
Optimizer	Adam
Early Stopping	(patience=6)

Tabelle 4.2: Trainingsparameter für das Bilderkennungsmodell

Die Trainingspipeline überwacht kontinuierlich den Fortschritt des Trainings, indem sie Leistungsmetriken, wie den Trainings- und Validierungsverlust (Loss) sowie die Genauigkeit (Accuracy) sowohl im Training als auch im Test mit Testdaten aufzeichnet und auswertet. Der Learning Rate Scheduler passt die Lernrate anhand dieser Metriken dynamisch an, um sicherzustellen, dass das Modell effizient konvergiert. Early Stopping wird eingesetzt, um das Training zu beenden, wenn keine Verbesserung der Validierungsgenauigkeit mehr festgestellt wird. Dies spart Zeit und Rechenressourcen und minimiert das Risiko einer Überanpassung.

Zusätzlich wird *TensorBoard* verwendet, um den Trainingsfortschritt visuell darzustellen. TensorBoard ermöglicht es, die Leistungsmetriken in Echtzeit zu überwachen und Trends im Training zu erkennen. Dies umfasst Diagramme der Lernrate, der Verlustkurve und der Genauigkeit, die alle helfen, das Verhalten des Modells während des Trainings zu verstehen und zu optimieren.

4.3.5 Validierung und Evaluierung

Die Validierung des Modells erfolgt nach Abschluss des Trainings mit einem weiteren Teildatensatz der generierten Bilder, die nicht für das Training und die Tests verwendet wurden. Diese Validierung ermöglicht es, die Leistung des Modells auf neuen, unbekannten Daten zu testen und sicherzustellen, dass es nicht nur die Trainingsdaten gut gelernt hat. Zu diesem Zweck werden Metriken wie die Accuracy und der Loss ausgewertet, um die Genauigkeit und Zuverlässigkeit des Modells zu bewerten. Eine Konfusionsmatrix hilft dabei die Klassifikationsergebnisse je Klasse zu visualisieren und zu verstehen.

Zusätzlich erfolgt eine Evaluierung des Modells mit realen Bildern, um seine Leistungsfähigkeit in praktischen Szenarien zu testen und seine Generalisierungsfähigkeit zu beurteilen. Diese Evaluierung ist entscheidend, um festzustellen, ob ein synthetisch generierter Datensatz geeignet ist, ein Modell für reale Anwendungen zu trainieren. Auch hier werden die Metriken Accuracy und Loss, sowie eine Konfusionsmatrix verwendet, um die Leistung des Modells zu bewerten.

4.4 Aufgetretene Probleme und Anpassungen

Während der Durchführung des Experiments traten verschiedene Probleme auf, die eine Anpassung der Methoden und Vorgehensweisen erforderlich machten. Diese Probleme und Anpassungen werden im Folgenden näher erläutert.

Insebesondere wurde in der ursprünglichen Planung beabsichtigt, das Framework *DiffuGen* (Shenoda & Kim, 2023) für die Datensatzgenerierung zu verwenden. Dieses Framework bot vielversprechende Funktionen zur Generierung von Datensätzen mittels Stable Diffusion, wie die Anpassung von Modellen durch Textual Inversion, die Generierung von Prompts mittels Prompt-Templating und die Erstellung von segmentierten Labeln für die Klassifizierung von Bildern.

Jedoch traten bei der Implementierung und Nutzung des Frameworks DiffuGen mehrere Herausforderungen auf, die die geplante Automatisierung des Datensatzgenerierungsprozesses behinderten.

4.4.1 Textual Inversion

Da das Framework DiffuGen vorsieht die Stable Diffusion Modelle mittels Textual Inversion anzupassen, um spezifische Konzepte zu erlernen, wurde zunächst versucht diese Methode für das Training der Modelle für Urinale und Verschmutzungen zu verwenden. Jedoch stellte sich im Trainingsprozess schnell herraus, dass die Ergebnisse der Bildgenerierung nicht den Erwartungen entsprachen. Der Verlauf der Trainingsparameter wie Vector Strength und Vector Magnitude waren zwar gut, jedoch lieferte das Modell keine zufriedenstellenden Ergebnisse. Es entstanden defekte Bilder, die nicht den gewünschten Konzepten entsprachen. Außerdem kam es zu Trainingszeiten von bis zu 15 Stunden.

Die Ursache für die schlechten Ergebnisse liegt vermutlich darin, dass Textual Inversion *nur* den Text Encoder trainiert, um die Konzepte zu erlernen. Dies funktioniert gut für Konzepte, die dem Modell bereits bekannt sind, wie Personen oder Tiere. Für völlig neue Konzepte wie Urinale und Verschmutzungen ist Textual Inversion scheinbar nicht geeignet, da das Modell diese Konzepte nicht kennt und es schwer ist, sie auf diese Weise völlig neu zu lernen. Da ein untrainiertes Modell (Realistic Vision oder Stable Diffusion 1.5) nicht in der Lage ist Urinale zu generieren, liegt die Vermutung nahe, dass das Modell keine Urinale kennt und es schwer ist auf diese Weise das Konzept zu erlernen.

Lösung: Low-Rank Adaptation (LoRA)

Um das Modell an die neuen Konzepte von Urinalen und Verschmutzungen anzupassen, wurde erfolgreich die LoRA Methode verwendet. Durch die Möglichkeit mit LoRA die Gewichte des Modells zu trainieren, können gänzlich neue Konzepte erlernt werden, die im ursprünglichen Modell nicht enthalten waren.

Durch die Verwendung von LoRA konnten die Trainingszeiten erheblich verkürzt werden und betrugen nur etwa drei Stunden pro Training. Darüber hinaus ermöglicht LoRA das Zusammenführen der LoRA Modelle mit Stable Diffusion Modellen, wodurch die neuen Gewichte direkt in das Modell integriert werden können. Dies führt zur Erstellung eines neuen, eigenständigen Modells, das die neuen Konzepte enthält, ohne dass eine separate LoRA-Datei benötigt wird.

4.4.2 Probleme bei der Nutzung von DiffuGen

Die Verwendung von LoRA anstelle von Textual Inversion führte dazu, dass eine Anpassung des Frameworks DiffuGen notwendig wurde. Dabei traten jedoch weitere Schwierigkeiten auf, die die Verwendung von DiffuGen erschwerten.

Veraltete Diffusers Library:
Das Framework DiffuGen verwendet für die Bildgenerierung die Version 0.16.1 der Diffusers Library, die inzwischen veraltet ist. Da die LoRA-Methode erst in neueren Versionen der Diffusers Library verfügbar ist, muss die Diffusers Library aktualisiert werden, um LoRA verwenden zu können. Dies ist jedoch nicht ohne weiteres möglich, da die neueren Versionen der Diffusers Library Inkompatibilitäten und Fehler verursachen, die eine aufwändige Anpassung des Codes erfordern.

Custom Model Support:
Die Diffusers Library ermöglicht das einfache Laden und Verwenden von Stable Diffusion Modelle auf Huggingface. Die Verwendung von selbst trainierten Modellen, die lokal gespeichert sind, erfordert jedoch das Anlegen einer eigenen Generation-Pipeline, was den Prozess deutlich verkompliziert.

Lösung: Eigenes Framework zur Datensatzgenerierung

Aufgrund der beschriebenen Schwierigkeiten bei der Implementierung von DiffuGen wurde entschieden, dass es sinnvoller ist ein eigenes Framework zu entwickeln, um die Datensatzgenerierung zu automatisieren. Prinzipiell funktioniert das entwickelte Framework ähnlich wie DiffuGen, mit Prompt-Templates, Text-to-Image und Image-to-Image Generierung und Segmentierung.

Daher wurde das in Kapitel 4.2.2 beschriebene Framework entwickelt, um die Datensatzgenerierung zu automatisieren. Der große Vorteil des eigenen Frameworks ist, dass die AUTOMATIC1111 API zur Bildgenerierung nutzt, die auch für die Interaktion mit Stable Diffusion Modellen verwendet wird. Dies ermöglicht es, Einstellungen, Prompts und Modelle einfach zu testen und dann für die Automatisierung zu übernehmen. So wird sichergestellt, dass die automatisierte Datensatzgenerierung die gleichen Ergebnisse wie die manuelle Generierung liefert.

5 Ergebnisse

In diesem Kapitel werden die Ergebnisse des durchgeführten Experiments vorgestellt. Zunächst werden die Ergebnisse des Datensatzgenerierungsprozesses präsentiert und die generierten Datensätze bewertet. Anschließend werden die Resultate des Bilderkennungstrainings beschrieben und die Leistung der trainierten Modelle analysiert.

Zur Wahrung der Übersichtlichkeit zeigt dieses Kapitel nur die wichtigsten Ergebnisse sowie exemplarische Auszüge aus den Datensätzen und den entstandenen Bildern des Experiments. Die vollständigen Ergebnisse, Datensätze, trainierten Modelle, Konfigurationsdateien und weitere Informationen befinden sich in den beigefügten Dateien zu dieser Arbeit.

5.1 Training von Stable Diffusion

Wie in Kapitel 4 beschrieben, wurde das Training von Stable Diffusion in zwei Schritten durchgeführt. Zunächst wurde ein LoRA Modell für die Generierung des Objektes Urinal trainiert. Dieses Modell wird im Folgenden als *urinal-LoRA* bezeichnet. Anschließend wurde ein LoRA Modell für die Generierung von Verschmutzungen trainiert, welches im Folgenden als *dirty-LoRA* bezeichnet wird. Beide Modelle wurden auf dem Basismodell Stable Diffusion 1.5 trainiert und später sowohl mit dem Stable Diffusion 1.5 als auch mit dem Realistic Vision 5.1 Modell kombiniert.

5.1.1 urinal-LoRA

Der Datensatz für das urinal-LoRA-Training bestand aus 17 Bildern von Urinalen des gleichen Typs, die in verschiedenen Umgebungen und Perspektiven aufgenommen wurden. Abbildung A.3 zeigt Beispielbilder aus dem Datensatz auf dem das urinal-LoRA Modell trainiert wurde.

Trainiert wurden zehn Epochen mit einer Batchsize von zwei. Je Epoche wurden 150 Iterationen je Bild durchgeführt, sodass im gesamten Training 12750 Iterationen durchgeführt wurden.

$$17\ \textit{Bilder} \times 150\ \textit{Iterationen/Bild} \times 10\ \textit{Epochen} \div \textit{Batchsize}\ 2 = 12750\ \textit{Iterationen}$$

Nach jeder Epoche wurde das Modell gespeichert und ein Beispielbild generiert, um den Fortschritt des Trainings visuell überprüfen zu können. Zur Erstellung des Beispielbildes während des Trainingsprozesses wurde der Sampler *Euler a*, 20 Sampling-Schritte und das Basismodell Stable Diffusion 1.5 verwendet. Als Prompt wurde folgender Ausdruck verwendet: `white urinal, wall mounted, bathroom, front view, tiles, white tile wall, tile floor`. Abbildung A.5 zeigt die generierten Beispielbilder je Epoche.

Kohya_ss erlaubt es eine Vielzahl von Einstellungen für das Training von LoRA Modellen zu konfigurieren. Tabelle 5.1 zeigt die wichtigsten Trainingseinstellungen für das urinal-LoRA Training. In den beigefügten Dateien zu dieser Arbeit finden sich die vollständigen Trainingseinstellungen in der Datei `training-config_urinalLora_v2.json`. Abbildung A.7 zeigt den Verlauf des Trainingsverlusts über die Epochen. Der Trainingsverlust sinkt kontinuierlich und erreicht nach zehn Epochen einen Wert von etwa 0.05. Die Trainingszeit betrug 3 Stunden und 15 Minuten.

Parameter	**Wert**
Batchsize	2
Iterations pro Bild	150
Anzahl der Epochen	10
Learning Rate	0.0002
Unet Learning Rate	0.0001
Text-Encoder Learning Rate	0.00005
vortrainiertes Modell	Stable Diffusion v1.5

Tabelle 5.1: Trainingseinstellungen für urinal-LoRA

5.1.2 dirty-LoRA

Der Datensatz für das dirty-LoRA Training bestand aus acht Bildern von verschiedenen Verschmutzung im Sanitärbereich, besonders Toiletten und Urinalen. Die Bilder wurden in verschiedenen Umgebungen und Perspektiven aufgenommen. Abbildung A.4 zeigt die Bilder des Datensatzes auf dem das dirty-LoRA Modell trainiert wurde.

Trainiert wurden zehn Epochen mit einer Batchsize von zwei. Je Epoche wurden 200 Iterationen je Bild durchgeführt, sodass im gesamten Training 8000 Iterationen durchgeführt wurden.

$$8\ \textit{Bilder} \times 200\ \textit{Iterationen/Bild} \times 10\ \textit{Epochen} \div \textit{Batchsize}\ 2 = 8000\ \textit{Iterationen}$$

Nach jeder Epoche wurde das Modell gespeichert und ein Beispielbild generiert, um den Fortschritt des Trainings visuell überprüfen zu können. Zur Erstellung des Beispielbildes während

des Trainingsprozesses wurde der Sampler *Euler a*, 20 Sampling-Schritte und das Basismodell Stable Diffusion 1.5 verwendet. Als Prompt wurde folgender Ausdruck verwendet: `white urinal, wall mounted, bathroom, tile wall, tiles, dirty, stains, grimy`. Abbildung A.6 zeigt die generierten Beispielbilder.

Kohya_ss erlaubt es eine Vielzahl von Einstellungen für das Training von LoRA-Modellen zu konfigurieren. Tabelle 5.2 zeigt wichtigsten Trainingseinstellungen für das dirty-LoRA Training. In den beigefügten Dateien zu dieser Arbeit finden sich die vollständigen Trainingseinstellungen in der Datei `training-config_dirtyLora_v2.json`. Abbildung A.8 zeigt den Verlauf des Trainingsverlusts über die Epochen. Der Trainingsverlust sinkt kontinuierlich und erreicht nach zehn Epochen einen Wert von etwa 0.06. Die Trainingszeit betrug zwei Stunden.

Parameter	**Wert**
Batchsize	2
Iterations pro Bild	200
Anzahl der Epochen	10
Learning Rate	0.0001
Unet Learning Rate	0.0001
Text-Encoder Learning Rate	0.0001
vortrainiertes Modell	Stable Diffusion v1.5

Tabelle 5.2: Trainingseinstellungen für dirty-LoRA

5.2 Parametervergleich der Bildgenerierung

Um die Qualität der zu generierenden Bilder zu optimieren, wurden verschiedene Parameter der Bildgenerierung variiert und in X/Y/Z-Plots gegenübergestellt. Dadurch können die Auswirkungen der Parameter auf das Generierungsergebnis visuell beurteilt und die besten Parameter für die Generierung von Bildern identifiziert werden.

5.2.1 Vergleich der urinal-LoRA Epochen und Gewichtung mit verschiedenen Modellen

Nach dem Training der LoRA-Modelle galt es zunächst die beste Epoche und das beste Gewicht zu identifizieren. Dazu wurden die verschiedenen Epochen des urinal-LoRAs mit verschiedenen Gewichtungen gegenübergestellt und gleichzeitig auf verschiedene Modelle

angewendet. Abbildung A.9 und Abbildung A.10 zeigen die Ergebnisse des Parametervergleichs für das urinal-LoRA Training mit dem Stable Diffusion 1.5 und Realistic Vision 5.1 Modell.[1]

Als Ergebnis des Parametervergleichs wurde beim Stable Diffusion 1.5 Modell die Epoche neun mit einer Gewichtung von 0.8 als beste Kombination identifiziert. Beim Realistic Vision 5.1 Modell wurde die Epoche sieben mit einer Gewichtung von 0.8 als beste Kombination identifiziert. Mit Hilfe der Option *Merge LoRA* in Kohya_ss wurde aus diesen Kombinationen jeweils ein eigenständiges Modell erstellt, das zur weiteren Verwendung in der Bildgenerierung verwendet wurde. Durch die Zusammenführung des LoRA Modells mit dem Stable Diffusion Modell kann dieses Modell flexibel für die Bildgenerierung eingesetzt werden, auch in Kombination mit anderen LoRA Modellen, wie dem dirty-LoRA. Die Bennenung dieser Modelle basiert auf dem verwendetem LoRA Modell, der gewählten Epoche, dem Stable Diffusion Modell und der Gewichtung des LoRA Modells zum Stable Diffusion Modell.

Zusammengeführte Modelle:

- **Realistic Vision 5.1:** `urinalLoRA_v2-ep7_RealVis-80.safetensors`
- **Stable Diffusion 1.5:** `urinalLoRA_v2-ep9_SD15-80.safetensors`

5.2.2 Vergleich von Sampler, Scheduler, CFG-Scale

Der Sampler, Scheduler und CFG-Scale sind wichtige Parameter, die die Bildgenerierung maßgeblich beeinflussen. Um die besten Parameter für die Bildgenerierung zu identifizieren, wurden verschiedene Kombinationen dieser Parameter getestet und gegenübergestellt.

Zunächst wurde eine Auswahl von Samplern und Schedulern getestet und gegenübergestellt. Abbildung A.11 zeigt verschiedene Sampler- und Scheduler-Kombinationen bei einem CFG-Scale von sieben, 20 Sampling-Schritten und dem *urinalLora_v2-ep7_RealVis-80* Modell. Da viele Kombinationen bei einem CFG-Scale von sieben defekte Bilder generierten, wurde der CFG-Scale testweise auf fünf reduziert. Abbildung A.12 zeigt, dass dadurch weniger defekte Bilder generiert wurden. Auffällig ist an dieser Stelle, dass der Scheduler *Karras* in allen Kombinationen bessere Ergebnisse lieferte und weniger defekte Bilder generierte. Daher wurden unter Verwendung des Schedulers *Karras* die Kombination der Sampler und verschiedenen CFG-Scales getestet. Abbildung A.13 zeigt, dass die Sampler *DPM++ 2M* und *DPM++ SDE* bei allen CFG-Scales die stabilsten Ergebnisse lieferten. Zu erkennen ist, dass ein höherer CFG-Scale bessere Ergebnisse liefert und die Prompts besser umgesetzt werden.

[1] Die vollständigen Ergebnisse und hochauflösende Bilder des Parametervergleichs finden sich in den beigefügten Dateien zu dieser Arbeit.

Für die weitere Bildgenerierung wurde daher der Sampler *DPM++ 2M*, der Scheduler *Karras* und ein CFG-Scale von sieben festgelegt.

5.2.3 Vergleich der dirty-LoRA-Epochen und Gewichtung

Nachdem das grundlegende Modell für die Urinal-Generierung festgelegt und die optimalen Parameter für die Bildgenerierung identifiziert wurden, wurde das dirty-LoRA Modell in der Bildgenerierung eingesetzt und getestet. Dazu mussten zunächst die beste Epoche und die optimale Gewichtung identifiziert werden.

Abbildung A.14 zeigt die Ergebnisse des Parametervergleichs für das dirty-LoRA-Training mit dem *urinalLoRA_v2-ep7_RealVis-80* Modell.

Es ist auffällig, dass die Gewichtung des dirty-LoRA Modells einen Einfluss auf die Form des Urinals hat und ab einer Gewichtung von 0.6 bis 0.8 zu stark deformierten Urinalen führt. Als Ergebnis des Parametervergleichs wurde die Epoche acht mit einer Gewichtung von 0.4 als beste Kombination ausgewählt.

5.3 Ergebnisse der Datensatzgenerierung

Mit den trainierten Modellen und den identifizierten Parametern konnten nun die Datensätze für das Bilderkennungstraining generiert werden. In diesem Abschnitt werden die Ergebnisse der Datensatzgenerierung vorgestellt und bewertet. Zusammenfassend zeigt Tabelle 5.3 die Parameter, die für die Bildgenerierung aller Datensätze verwendet wurden.

Parameter	**Wert**
Anzahl der Bilder	576 pro Klasse
Anzahl der Klassen	3 (sauber, mittel, dreckig)
Seed	-1 (random seed)
CFG-Scale	7
Diffusion Steps	35
Bildgröße	512 x 512 Pixel
Sampler	DPM++ 2M
Scheduler	Karras
img2img Denoising Strength	0.5

Tabelle 5.3: Parameter der Bild- und Datensatzgenerierung

5.3.1 Prompt-Konfiguration

Für die Datensatzerstellung wurde das Prompt-Template `prompt_config_v11.json` verwendet (siehe Codeblock A.1), welches zur Generierung einer ausführlichen Prompt-Liste verwendet wurde.[2] Die Tabelle 5.4 zeigt für jede Klasse einen Beispielprompt aus dieser Promptliste, der für die Datensatzgenerierung verwendet wurde. Varriert wurden der Ausdruck für die Verschmutzung und die Gewichtung des dirty-LoRA Modells.

Klasse	Promptbeispiel
sauber	`urinal, a white urinal in a bathroom, white wall, tile floor, front view, wall mounted, soft light, (clean)`
mittel	`urinal, a white urinal in a bathroom, white wall, tile floor, front view, wall mounted, soft light, (stains), ‹lora:dirtyStyle_LoRA_v2-000008:0.3›`
dreckig	`urinal, a white urinal in a bathroom, white wall, tile floor, front view, wall mounted, soft light, (dirty, stains), ‹lora:dirtyStyle_LoRA_v2-000008:0.4›`

Tabelle 5.4: Prompt-Beispiele je Klasse für die Datensatzgenerierung

Als negativ Prompt wurde folgender Ausdruck verwendt, um unerwünschte Merkmale in den generierten Bildern zu vermeiden:

```
(semi-realistic,cgi,3d,render,sketch,cartoon,drawing,anime),text,cropped,out of
frame,cut off,(worst quality,low quality),jpeg
artifacts,duplicate,(deformed),blurry,bad proportions,faucet,UnrealisticDream
```

5.3.2 Parameter der Datensatzgenerierung

Der Parametervergleich (Kapitel 5.2) ergab, dass das Modell *urinalLoRA_v2-ep7_RealVis-80* basierend auf dem Realistic Vision 5.1 Modell visuell die besseren Ergebnisse liefert, als das Modell *urinalLoRA_v2-ep9_SD15-80* basierend auf Stable Diffusion 1.5. Um die Leistung der Modelle auch quantitativ bewerten zu können, wurden beide Modelle für die Datensatzgenerierung verwendet.

Das dirty-LoRA Modell wurde in der Datensatzgenerierung eingesetzt, um aus den Bildern der sauberen Urinale die mittel dreckigen und dreckigen Urinale zu generieren. Um den Einfluss der Gewichtung des dirty-LoRA Modells auf die Leistung des Bilderkennungsmodells

[2] Die vollständigen Prompt-Config-Dateien und Prompt-Liste finden sich in den beigefügten Dateien zu dieser Arbeit.

zu testen, wurden die Gewichtungen in der Datensatzgenerierung varriert. Daher wurden für die Klassen *mittel* und *dreckig* die Gewichtungen 0.3 und 0.4 als auch 0.4 und 0.5 verwendet.

Desweitern ist aufgefallen, dass durch das Hinzufügen der Verschmutzung in die generierten Bilder, es zu farblichen Abweichungen in den Bildern kam. Um einen möglichen Einfluss auf die Bilderkennung zu testen, wurden die generierten Bilder zusätzlich segmentiert und für jeden Datensatz ein weiterer mit Segmentierung erstellt.

Tabelle 5.5 gibt einen Überblick über die generierten Datensätze und die variierten Parameter.

Datensatz	**Modell**	**dirty-LoRA Gewichtung (sauber/mittel/dreckig)**	**Segmentation**
v12	urinalLoRA_v2-ep7_RealVis-80	0 / 0.3 / 0.4	Nein
v12_seg	urinalLoRA_v2-ep7_RealVis-80	0 / 0.3 / 0.4	Ja
v13	urinalLoRA_v2-ep9_SD15-80	0 / 0.3 / 0.4	Nein
v13_seg	urinalLoRA_v2-ep9_SD15-80	0 / 0.3 / 0.4	Ja
v14	dirtyLoRA_v2-ep7_RealVis-80	0 / 0.4 / 0.5	Nein
v14_seg	dirtyLoRA_v2-ep7_RealVis-80	0 / 0.4 / 0.5	Ja
v15	dirtyLoRA_v2-ep9_SD15-80	0 / 0.4 / 0.5	Nein
v15_seg	dirtyLoRA_v2-ep9_SD15-80	0 / 0.4 / 0.5	Ja

Tabelle 5.5: Generierte Datensätze und variierte Parameter

Zur Datensatzgenerierung wurde das selbst entwickelte Framework genutzt. Pro Datensatz betrug die Generierungszeit für insgesamt 1728 Bilder (576 pro Klasse) etwa 5 Stunden, zusätzlich 1,5 Stunden für die Segmentierung.

5.3.3 Überblick über die generierten Datensätze

Um die generierten Bilder zur Bilderkennung verwenden zu können, wurden die generierten Bilder in Trainings-, Test- und Validierungsdaten aufgeteilt. Dazu wurden die generierten Bilder der Klassen sauber, mittel und dreckig in einem Verhältnis von 70 % Trainingsdaten, 20 % Testdaten und 10 % Validierungsdaten aufgeteilt. Dadurch entstanden für jeden Datensatz 403 Trainingsbilder, 115 Testbilder und 58 Validierungsbilder pro Klasse. Also insgesamt 1209 Trainingsbilder, 345 Testbilder und 174 Validierungsbilder pro Datensatz.

Einen Einblick in die generierten Bilder der Datensätze geben die Abbildungen A.15, A.16, A.17, A.18, A.19, A.20, A.21 und A.22. Die Abbildungen zeigen Beispiele für die Klassen sauber, mittel und dreckig aus den generierten Datensätzen. Die vollständigen Datensätze und Bilder finden sich in den beigefügten Dateien zu dieser Arbeit.

5.3.4 Bewertung der Qualität und Diversität des Datensatzes

Generell bieten die generierten Datensätze eine realistische Darstellung von Urinalen, die in verschiedenen Umgebungen und Perspektiven abgebildet sind. Die Verschmutzungen sind deutlich zu erkennen und zwischen den Klassen sauber, mittel und dreckig gut unterscheidbar. Allerdings wirken die eingefügten Verschmutzungen weniger realistisch als der Rest des Bildes und erscheinen eher wie ein Overlay auf den Urinalen. Zudem führt das Hinzufügen der Verschmutzungen zu einer Veränderung der allgemeinen Weißtöne der Urinale und des Hintergrundes. Dies könnte die Bilderkennung beeinflussen und sollte bei der Bewertung der Modellleistung berücksichtigt werden.

Auffällig ist auch, dass in den Datensätzen v13, v13_seg, v15 und v15_seg, welche mit dem Modell *urinalLoRA_v2-ep9_SD15-80* generiert wurden, weniger Verschmutzungen in allen Klassen zu erkennen sind.

Vereinzelt sind in den Datensätzen auch unrealistische Urinale zu finden, die teils starke Artefakte aufweisen, verformt oder überblendet sind. Beispiele für solche Bildfehler in den Datensätzen sind in Abbildung A.23 zu sehen. Diese Bilder sind jedoch in der Minderheit und sollten keinen signifikanten Einfluss auf das Training des Bilderkennungsmodells haben.

5.4 Ergebnisse des Bilderkennungstrainings

5.4.1 Trainingsverlauf

Die generierten Datensätze wurden benutzt, um jeweils ein InceptionV3 Bilderkennungsmodell zu trainieren. Die Trainingsparameter und -einstellungen wurden für alle Modelle gleich gehalten. Diese Konsistenz ermöglicht einen fairen Vergleich der Leistung der verschiedenen Datensätze.

Um den Trainingsverlauf bewerten zu können, wurden während des Trainings diverse Metriken aufgezeichnet, sowohl für das Trainings- als auch für das Testset. Die Metriken umfassen den Loss, die Accuracy, den Recall und die Precision. Im weiteren Verlauf dieser Arbeit wird neben dem Loss die Accuracy als primäre Metrik zur Bewertung der Modellleistung verwendet und analysiert, da sie eine umfassende und leicht verständliche Metrik zur Bewertung der Gesamtleistung des Modells darstellt. Die Abbildungen 5.1 und 5.2 zeigen die Test-Accuracy und den Test-Loss der verschiedenen Datensätze im Verlauf des Trainings. In beiden Abbildungen ist der typische Verlauf von Loss und Accuracy zu erkennen, der zeigt, dass die Modelle im Laufe der Epochen eine Verbesserung der Metriken aufweisen, die sich nach anfänglichen Schwankungen stabilisiert. So nimmt der Loss kontinuierlich ab und die

Accuracy kontinuierlich zu und konvergiert nach einigen Epochen auf einen Wert, bis keine Verbesserung mehr erzielt wird und das Training beendet wird.

Die Trainingspipeline speichert nach jeder Epoche das Modell mit der besten Test-Accuracy. Diese Modelle wurden für die weitere Analyse und Validierung verwendet. Die Tabellen A.1 und A.2 enthalten die Werte der Metriken für die beste Epoche der Trainings- und Testdaten.

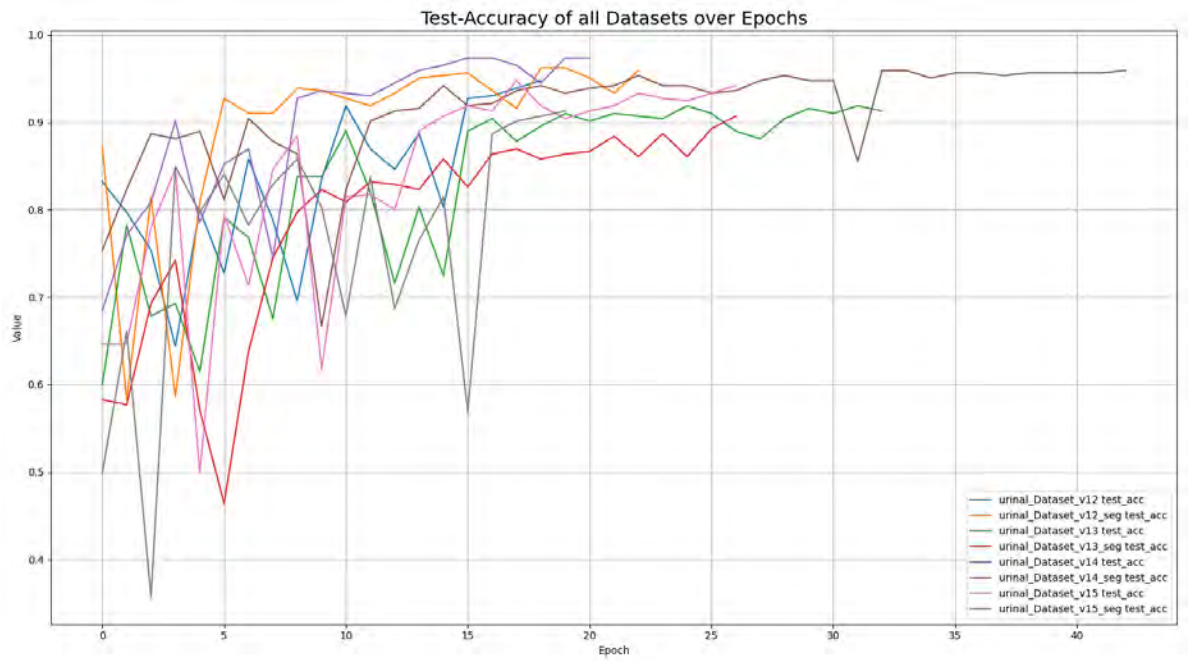

Abbildung 5.1: Test-Accuracy der Datensätze im Training auf dem InceptionV3 Modell

5.4.2 Modellvalidierung und Evaluation

Zur Validierung der Leistung der trainierten Modelle wurden diese sowohl mit einem Validierungsdatensatz als auch zur Evaluation mit realen Bildern getestet. Der Validierungsdatensatz besteht aus 174 generierten Bildern (58 pro Klasse), die nicht für das Training oder Testen der Modelle verwendet wurden.
Der Evaluationsdatensatz mit realen Bildern umfasst verschiedene Aufnahmen von Urinalen und enthält die Klassen sauber, mittel und dreckig, insgesamt elf Bilder (6 sauber, 4 mittel, 1 dreckig). Aufgrund der geringen Anzahl an realen Bildern und der ungleichen Verteilung der Klassen wurde die Evaluation mit realen Bildern nur als zusätzliche Überprüfung der Modellleistung durchgeführt. Sie bietet jedoch wertvolle Einblicke in die Generalisierungsfähigkeit der Modelle. Aufgrund der begrenzten Zeit und Ressourcen war es nicht möglich,

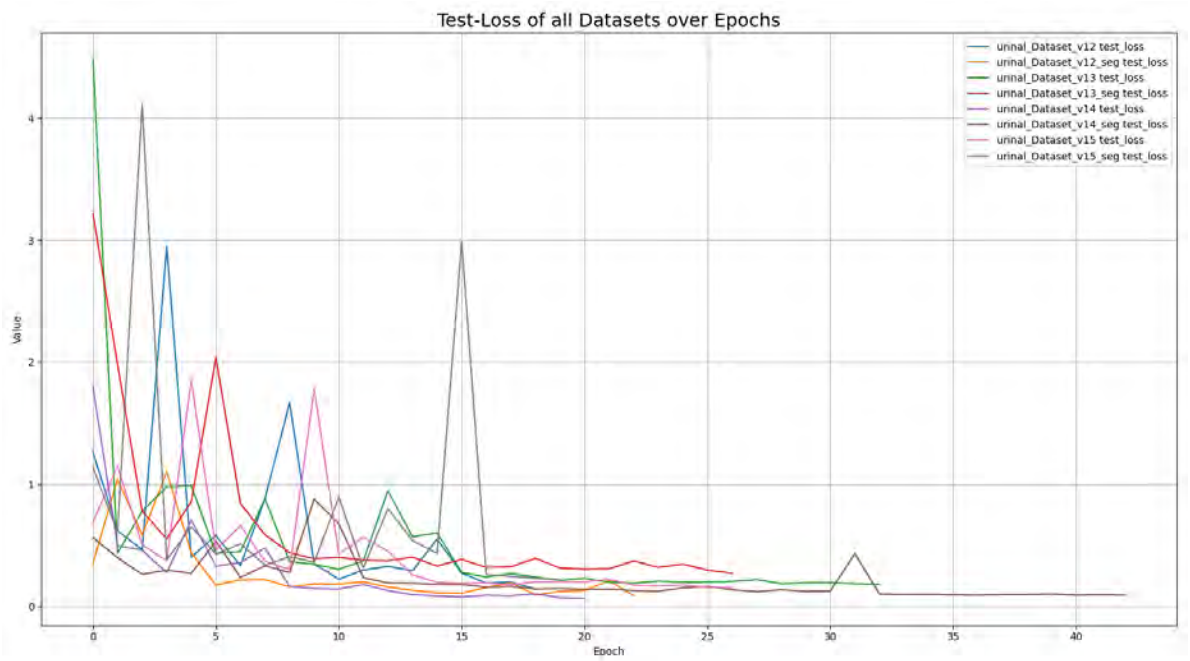

Abbildung 5.2: Test-Loss der Datensätze im Training auf dem InceptionV3 Modell

eine größere Anzahl an realen Bildern zu sammeln und zu verwenden. Die Abbildung A.24 zeigt beispielhaft acht der elf realen Bilder, die für die Modellevaluation verwendet wurden.

Die Abbildungen 5.3 und 5.4 veranschaulichen die Accuracy und den Loss der verschiedenen Datensätze im Vergleich. Dabei zeigen die Datensätze v14_seg und v14, basierend auf den Validierungsmetriken, die besten Ergebnisse. Die Datensätze v12_seg und v12 erzielten die zweitbesten Ergebnisse. Diese Datensätze wurden alle mit dem Modell *urinalLoRA_v2-ep7_RealVis-80* generiert, welches auf dem Realistic Vision 5.1 Modell basiert. Die Modelle, die mit dem Modell *urinalLoRA_v2-ep9_SD15-80* generiert wurden, erzielten schlechtere Ergebnisse. Dennoch haben alle Modelle eine Accuracy von über 90 % erzielt, was auf eine gute Leistung der Modelle hinweist. Die Werte der Metriken der Modellvalidierung mit dem generierten Validierungsdatensatz sind in der Tabelle A.3 aufgeführt.

Im Gegensatz dazu erzielten die Modelle in der Evaluation mit realen Bildern schlechtere Ergebnisse. Es ist kein Zusammenhang zwischen den Ergebnissen und den variierten Parametern der Bildgenerierung zu erkennen. Die Accuracy der Datensätze liegt zwischen 50 % und 77 %, wobei der Datensatz v14_seg, welcher die beste Validierungs-Accuracy erzielte, die schlechteste Evaluation-Accuracy erreichte. Die Datensätze v15_seg und v13 erzielten die besten Evaluation-Ergebnisse. Da teils sehr hohe Loss-Werte in der Evaluation auftreten, wurden diese Werte in der Abbildung 5.4 ausgelassen, da sie die Übersichtlichkeit der Grafik beeinträchtigen. Die Werte der Metriken der Modellevaluation mit realen Bildern sind in der

Tabelle A.4 aufgeführt.

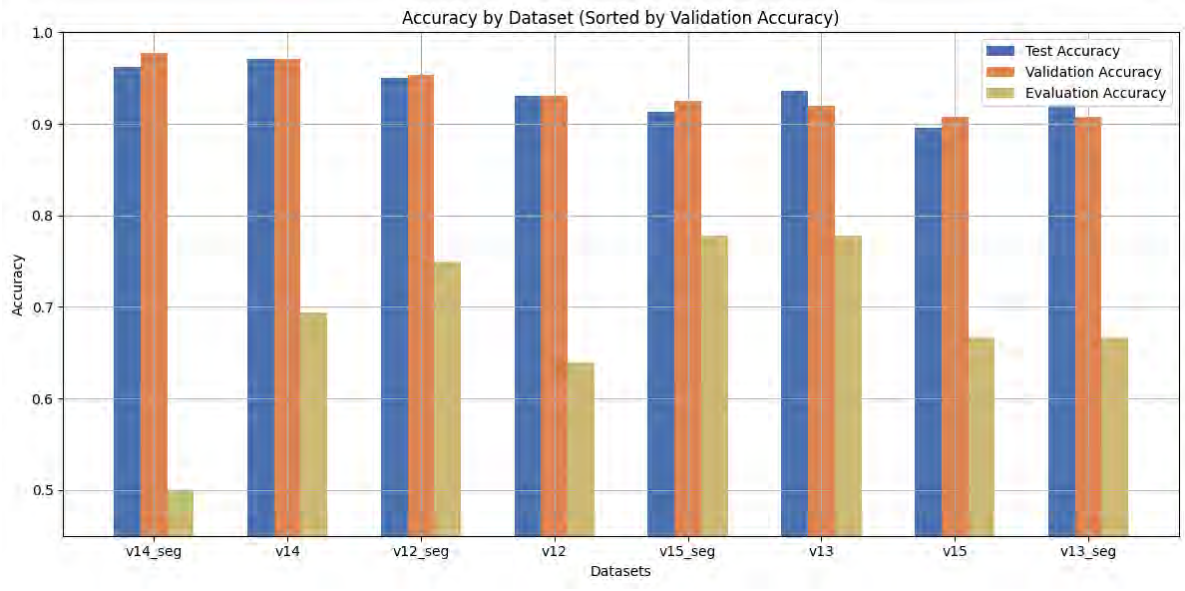

Abbildung 5.3: Accuracy der besten Modell-Epochen je Datensatz

Konfusionsmatrix

Um die Leistung der Modelle in den einzelnen Klassen (sauber, mittel, dreckig) zu bewerten, wurde für jedes Modell eine Konfusionsmatrix, basierend auf dem jeweiligen Validierungsdatensatz, erstellt.

Die Konfusionsmatrizen der besten vier Modelle (v14_seg, v14, v12_seg, v12) sind in den Abbildungen 5.5, 5.6, 5.7 und 5.8 dargestellt. Diese Grafiken verdeutlichen, dass die Modelle insgesamt eine gute Leistung in allen Klassen zeigen, wobei die Mehrheit der Bilder korrekt klassifiziert wurde. Besonders hervorzuheben ist, dass die Klasse *sauber* in allen Modellen vollständig korrekt klassifiziert wurde. Mit bis zu fünf Fehlklassifikationen von insgesamt 58 Bildern weist die Klasse *mittel* die meisten Fehler bei diesen Modellen auf. Das Modell v14_seg erzielte hierbei die besten Ergebnisse und wies nur zwei Fehler in den Klassen *mittel* und *dreckig* auf.

Die Konfusionsmatrizen der Modelle v13, v13_seg, v15 und v15_seg, dargestellt in den Abbildungen A.25, A.26, A.27 und A.28, zeigen eine geringere Leistung im Vergleich zu den besten Modellen. Auch bei diesen Modellen wurde die Klasse *sauber* am besten klassifiziert, mit maximal einem Fehler von 58 Bildern. Die Klassen *mittel* und *dreckig* weisen jedoch mit bis zu 14 Fehlklassifikationen deutlich mehr Fehler auf als die besten Modelle.

Die Konfusionsmatrizen für den Evaluationsdatensatz sind in den beigefügten Dateien zu dieser Arbeit zu finden. Da die Anzahl der echten Bilder gering und die Klassenverteilung

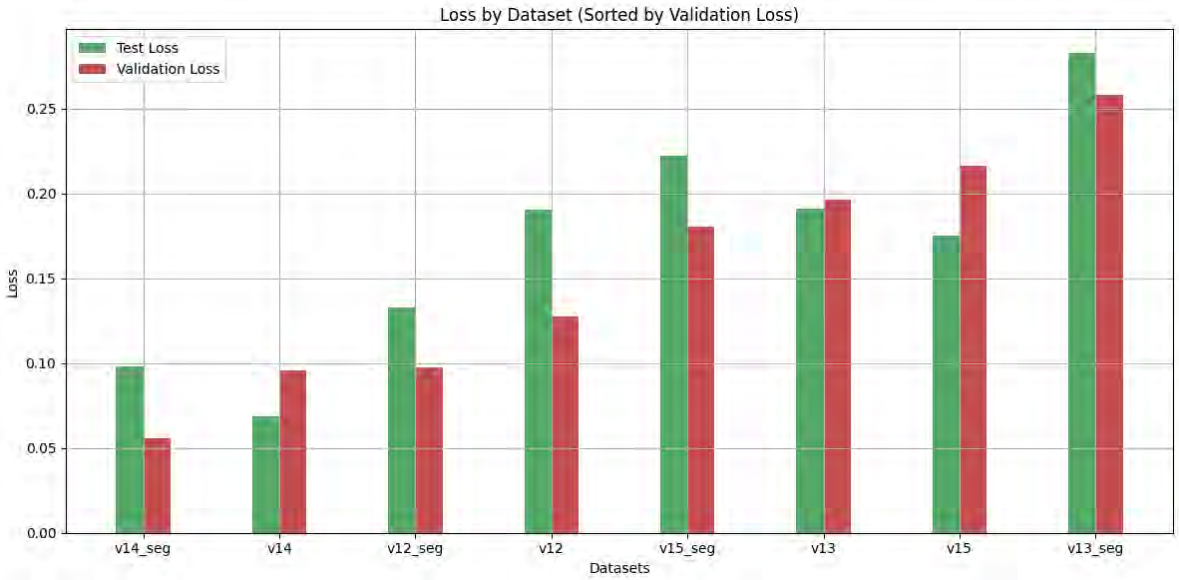

Abbildung 5.4: Loss der besten Modell-Epochen je Datensatz

ungleich ist, sind die Ergebnisse der Konfusionsmatrizen für die Evaluation mit echten Bildern nicht repräsentativ und können nicht direkt mit den Konfusionsmatrizen der Validierungsdatensätze verglichen werden.

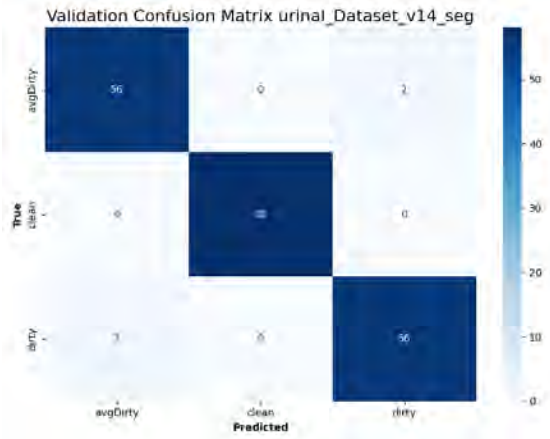

Abbildung 5.5: Validierungs-Konfusionsmatrix für Datensatz v14_seg

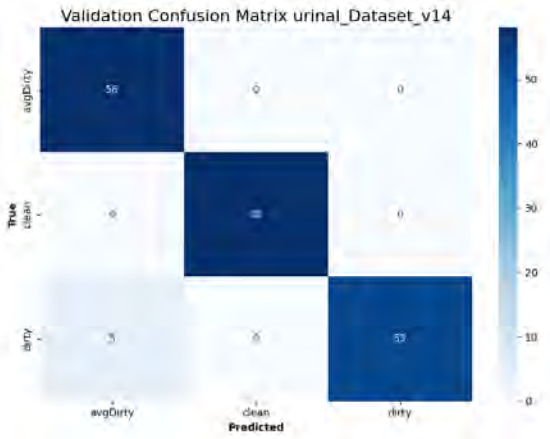

Abbildung 5.6: Validierungs-Konfusionsmatrix für Datensatz v14

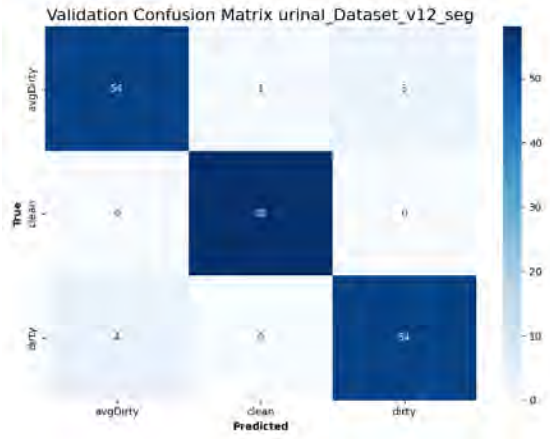

Abbildung 5.7: Validierungs-Konfusionsmatrix für Datensatz v12_seg

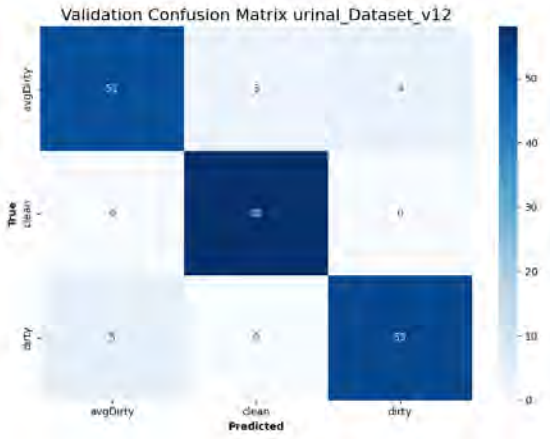

Abbildung 5.8: Validierungs-Konfusionsmatrix für Datensatz v12

6 Diskussion/Evaluation

6.1 Zusammenfassung der Ergebnisse

Die Ergebnisse dieser Arbeit zeigen, dass die Anpassung von Stable Diffusion mittels LoRA insbesondere für die Generierung von Objekten gut geeignet ist. Die Methode zur Hinzufügung von Verschmutzungen durch LoRA funktioniert grundsätzlich, jedoch besteht hier Verbesserungsbedarf bei den Trainingsdaten. Da im Rahmen dieser Bachelorarbeit keine realistischen Verschmutzungsbilder zur Verfügung standen oder erstellt werden konnten, enthielt der Trainingsdatensatz für das dirty-LoRA Modell zwar Bilder von Verschmutzungen, jedoch waren diese nicht ausreichend realistisch im Hinblick auf typische Verschmutzungen in Sanitärbereichen. Zudem bestand der Datensatz überwiegend aus Bildern von Toiletten anstatt von Urinalen, was zur Folge hatte, dass das dirty-LoRA Modell auch Einfluss auf die Form der generierten Urinale hatte. Mit realistischeren Trainingsdaten könnte die Methode weiter verbessert werden, um realistischere und vielseitigere Verschmutzungsmuster zu erzeugen.

Hinsichtlich der Datensatzerstellung konnte festgestellt werden, dass die Generierung von Datensätzen mithilfe der entwickelten Methoden auf Basis der *AUTOMATIC1111 API* automatisiert und in einer akzeptablen Zeit erfolgen kann. Die allgemeine Qualität der erzeugten Datensätze wurde als gut bewertet. Dies zeigt das Potenzial der Methode zur effizienten Erstellung großer und vielfältiger Datensätze.

Beim Training der Bilderkennungsmodelle zeigte sich, dass diese in der Lage sind, die Validierungsdaten recht zuverlässig zu klassifizieren. Die besten Ergebnisse wurden mit den Datensätzen v14_seg, v14, v12_seg und v12 erzielt, die mit Realistic Vision 5.1 erstellt wurden. Im Vergleich dazu schnitten die Datensätze v13, v13_seg, v15 und v15_seg schlechter ab, die mit Stable Diffusion 1.5 erstellt wurden. Dies deutet darauf hin, dass Realistic Vision 5.1 besser geeignet ist als Stable Diffusion 1.5, um Trainingsdatensätze für die Bilderkennung zu erzeugen. Die Validierungsgenauigkeit lag durchweg über 90 %, was darauf hindeutet, dass die Modelle gut trainiert wurden und in der Lage sind, die Validierungsdaten zuverlässig zu klassifizieren.

Ein wesentlicher Befund ist jedoch, dass die trainierten Modelle nicht robust genug sind, um auf echten Bildern zu generalisieren. Die Evaluierung der Modelle mit echten Bildern zeigte

eine schlechte Performance von 50 % bis 77 % Accuracy, ohne erkennbare Zusammenhänge zwischen den Modellen und den Ergebnissen.

6.1.1 Interpretation der Ergebnisse

Die Ergebnisse zeigen, dass die Bilderkennungsmodelle, die auf einem segmentierten Datensatz trainiert wurden (v14_seg, v12_seg), bessere Leistungen erzielten als die Modelle, die auf einem nicht segmentierten Datensatz trainiert wurden (v14, v12). Insbesondere die Tatsache, dass der Datensatz zur Validierung nicht segmentiert war, deutet darauf hin, dass die Modelle, die ohne Hintergrund trainiert wurden, robuster sind und besser generalisieren können. In der Evaluation mit realen Bildern lässt sich dies jedoch nicht bestätigen.

Die hohe Validierungsgenauigkeit zeigt, dass es möglich ist, mit Stable Diffusion zuverlässig konstante Datensätze zu generieren, die für das Training von Bilderkennungsmodellen verwendet werden können, ohne dass es zu großen Abweichungen in den generierten Klassen kommt.

Die fehlende Generalisierung für echte Bilder macht jedoch deutlich, dass die entwickelte Methodik noch nicht ausgereift ist und weiterer Verbesserungen bedarf. Die schlechte Performance der Modelle bei der Evaluierung mit echten Bildern deutet darauf hin, dass die generierten Verschmutzungen nicht realistisch genug sind, um eine zuverlässige Generalisierung zu ermöglichen. Zu beachten ist aber auch, dass der Evaluierungsdatensatz zu klein und ungleich über die Klassen verteilt war, um zuverlässige Schlussfolgerungen zu ziehen. Die Erstellung eines größeren und ausgewogeneren Evaluierungsdatensatzes war aufgrund der begrenzten Ressourcen im Rahmen dieser Arbeit nicht möglich, würde jedoch die Aussagekraft der Ergebnisse erhöhen.

Dies unterstreicht die Notwendigkeit weiterer Verbesserungen sowohl bei der Erstellung der Trainingsdatensätze für die LoRA-Modelle als auch bei der Modellanpassung, um eine bessere Leistung in realen Anwendungsszenarien zu gewährleisten.

6.1.2 Vergleich mit bestehenden Studien

Die von Jayasinghe et al. (2019) vorgestellte Methode zur Klassifizierung von Urinalen in die Klassen *sauber*, *mittel* und *dreckig* erreichte eine Genauigkeit von 90 %. Im Vergleich dazu erzielten die Modelle in dieser Arbeit eine Genauigkeit von 50 % bis 77 % bei der Evaluation mit echten Bildern. Es ist anzumerken, dass die Modelle in dieser Arbeit auf generierten Bildern trainiert wurden, während die Arbeit von Jayasinghe et al. (2019) eine Principal Component Analysis (PCA) in Kombination mit einem Convolutional Neural Network (CNN) verwendete, um die Klassifizierung realer Bilder durchzuführen.

Eine weitere relevante Studie ist die Arbeit von Canedo et al. (2021), die ein YOLOv5 Modell mit einem synthetischen Datensatz zur Erkennung von Bodenverschmutzung trainiert haben. Die Ergebnisse dieser Studie zeigen, dass die Modelle auf realen Bildern gut generalisieren und eine hohe Präzision erzielen, wobei die mittlere Durchschnittspräzision (mAP) als Metrik verwendet wurde. So wurde eine Durchschnittspräzision von 0.874 für das YOLOv5 Modell erreicht. Im Vergleich dazu erreichten die Modelle dieser Arbeit eine geringere Genauigkeit bei der Evaluation mit echten Bildern.

Zusammenfassend lässt sich sagen, dass die vorgestellten Modelle zwar gute Ergebnisse bei generierten Bildern erzielen, jedoch bei der Evaluation mit echten Bildern noch Verbesserungspotential besteht. Die Diskrepanzen zu den Ergebnissen aus der Literatur unterstreichen die Notwendigkeit einer weitergehenden Optimierung der Modelle und der verwendeten Datensätze.

6.2 Bewertung der Methodik

Die Bildgenerierung mittels der *Automatic1111 API* in Verbindung mit LoRAs erwies sich als eine effektive Methode zur Interaktion mit Stable Diffusion. Diese Herangehensweise ermöglicht eine effiziente und automatisierte Erstellung von Datensätzen und bietet den Vorteil einer benutzerfreundlichen und einfachen Nutzung, ohne dass tiefgehende programmatische Kenntnisse erforderlich sind. Das Experiment war jedoch stark fokussiert, um die Machbarkeit der Methode festzustellen, was zu einer Beschränkung auf ein sehr spezifisches Szenario mit nur einem Objekt (Urinal) führte und somit die Generalisierbarkeit der Ergebnisse einschränken könnte. Die Auswahl der Datensatzparameter, einschließlich Modell, Segmentation und LoRA, gab Aufschluss über die Auswirkungen dieser Parameter auf die Qualität der generierten Datensätze. Es wurde festgestellt, dass die Gewichtung der LoRA je nach Modell optimiert werden könnte, um die Qualität weiter zu verbessern.

Das Training der Bilderkennungsmodelle war notwendig, um die generierten Datensätze zu bewerten. Dabei wurde jedoch keine Optimierung der Trainingsmethoden oder Hyperparameter vorgenommen. Die Auswahl der Modelle und Hyperparameter erfolgte recht willkürlich, da keine systematische Optimierung durchgeführt wurde, was die Aussagekraft der Ergebnisse einschränken könnte.

Die Evaluierung der Modelle mit realen Bildern war notwendig, um die Generalisierbarkeit der Modelle zu überprüfen. Es stellte sich heraus, dass der Evaluierungsdatensatz zu klein und ungleichmäßig über die Klassen verteilt war, um verlässliche Aussagen treffen zu können. Ebenso benötigt das dirty-LoRA Modell realistischere Trainingsdaten, um realistischere Verschmutzungsmuster zu generieren und damit die Leistung der Bilderkennungsmodelle auf

realen Bildern zu verbessern. Die Erstellung solcher Datensätze erfordert jedoch erheblichen Aufwand und Ressourcen, die im Rahmen dieser Arbeit nicht zur Verfügung standen.

Es sollte außerdem geprüft werden, ob der Aufwand der Datensatzerstellung für das Training der Bildgenerierung im Verhältnis zum Nutzen steht oder ob es nicht effizienter ist, direkt entsprechende Trainingsdaten für die Bilderkennung zu erstellen.

6.3 Praktische Implikationen und Empfehlungen

6.3.1 Empfehlungen für zukünftige Forschung

Für zukünftige Forschungen gibt es verschiedene Ansatzpunkte zur Verbesserung der Datensatzgenerierung und Bilderkennung. Eine Erweiterung der Datensatzgenerierung auf andere Objekte und Zustände in Sanitärbereichen sowie variierende Lichtverhältnisse und weitere Perspektiven könnte die Robustheit und Generalisierbarkeit der Modelle signifikant erhöhen. Wie bereits erwähnt, ist die Erstellung realistischer Trainingsdaten für das dirty-LoRA Modell ein entscheidender Faktor für die Verbesserung der Leistung der Bilderkennungsmodelle auf realen Bildern. Ebenso sollte die Erweiterung des Evaluierungsdatensatzes auf eine größere Anzahl von realen Bildern in Betracht gezogen werden, um eine präzisere Bewertung der Generalisierbarkeit der Modelle zu ermöglichen.

Die Ausweitung der Anwendbarkeit der Modelle auf weitere Objekte, wie Waschbecken und Toiletten, würde ebenfalls zur Verbesserung der Modelle beitragen. Zudem könnte die Erkennung unterschiedlicher Verschmutzungsgrade, wie feste und flüssige Verschmutzungen, oder weitere Verschmutzungsstufen (z.B. Klassen 1-5), die Anwendbarkeit der Modelle erweitern. Dafür ist es notwendig, die Trainingsdaten entsprechend zu erweitern und zu verfeinern, um eine detaillierte Abstufung der Verschmutzungen in der Datensatzgenerierung zu ermöglichen.

Ein weiterer wichtiger Aspekt ist die kürzlich angekündigte Version Stable Diffusion 3, die auf einem neuartigen Transformer-Modell (Esser et al., 2024) basiert und signifikante Verbesserungen in der Qualität und Realitätsnähe der generierten Bilder verspricht.[1] Diese Fortschritte könnten zukünftige Forschungsarbeiten und Anwendungen erheblich bereichern und die Generalisierbarkeit der Modelle auf reale Szenarien weiter verbessern. Die in dieser Arbeit entwickelten Methoden zur Datensatzerstellung und -nutzung könnten somit unmittelbar von den Verbesserungen in Stable Diffusion 3 profitieren.[2]

[1] *Stable Diffusion 3 Medium.* (n. d.). Stability AI. Verfügbar 20. Juni 2024 unter https://stability.ai/news/stable-diffusion-3-medium

[2] Zu erwähnen sei, dass Stable Diffusion 3 zum Zeitpunkt der Erstellung dieser Arbeit unter der „Stability AI Non-Commercial Research Community License“ veröffentlicht wurde die keine kommerzielle Nutzung

6.3.2 Anwendung der Ergebnisse in der Praxis

Werden die Modellleistungen durch die genannten Verbesserungen optimiert, könnten die entwickelten Modelle vielfältige Anwendungsmöglichkeiten im Facility Management bieten. Die Automatisierung von Reinigungsprozessen könnte durch die entwickelten Modelle erheblich verbessert werden, indem Verschmutzungen in Echtzeit erkannt, gemeldet und beseitigt werden. Dies ermöglicht eine effiziente und zeitnahe Reinigung, die die Sauberkeit in öffentlichen Sanitärbereichen sicherstellt. Zudem kann die Qualitätskontrolle durch die objektive und zuverlässige Bewertung der Reinigungsqualität und Sauberkeit optimiert werden. Darüber hinaus könnten robotergestützte Reinigungssysteme von einem leistungsfähigen Bilderkennungssystem profitieren, um Verschmutzungen selbstständig zu erkennen und zu entfernen, was die Effizienz und Qualität der Reinigung weiter steigern und die Arbeitsbelastung des Reinigungspersonals reduzieren würde.

Denkbar wäre auch, dass diese Methode der synthetischen Datensatzerstellung sich einfach auf spezifische örtliche Gegebenheiten anpassen lässt, wie unterschiedliche Sanitäranlagen oder auch Bordtoiletten in Zügen oder Flugzeugen. Möglicherweise könnte die Methode auch eingesetzt werden um Bilderkennungssysteme für andere Anwendungen zu trainieren, wie der Qualitätskontrolle in der Produktion, um Defekte und Verschmutzungen zu erkennen und zu klassifizieren.

erlaubt und daher nicht der Open Source Natur der anderen Modelle entspricht.

7 Fazit

Die vorliegende Arbeit beschäftigte sich mit der Frage, wie die Technologie der Stable Diffusion genutzt werden kann, um einen realistischen und umfassenden Datensatz für die Schulung von Bilderkennungsmodellen zur effektiven Bewertung der Sauberkeit von Sanitärbereichen zu erstellen. Im Rahmen dieser Untersuchung wurden verschiedene Aspekte der Datensatzerstellung und -nutzung analysiert, wobei der Fokus auf der Anwendung von Stable Diffusion und CNNs lag.

Zunächst konnte gezeigt werden, dass die Anpassung von Stable Diffusion erfolgreich für die spezifische Anwendung der Objekt- und Verschmutzungsgenerierung eingesetzt werden kann. Die automatisierte Datensatzerstellung mittels der AUTOMATIC1111 API hat sich als effizient erwiesen und ermöglicht die Generierung großer Mengen an Trainingsdaten in kurzer Zeit. Dies ist ein wesentlicher Vorteil, da die manuelle Datensatzerstellung oft zeitaufwändig und kostenintensiv ist.

Bei der Validierung der Bilderkennungsmodelle zeigte sich, dass diese bei den generierten Datensätzen eine gute Leistung erbrachten, jedoch Schwierigkeiten bei der Generalisierung auf reale Szenarien hatten. Trotz einer durchweg hohen Validierungsgenauigkeit von über 90 % bei den generierten Datensätzen konnten die Modelle nicht die gleiche Genauigkeit auf reale Daten übertragen. Dies deutet darauf hin, dass die generierten Datensätze nicht ausreichend realistische Verschmutzungsbilder enthalten.

Ein weiterer wichtiger Befund war, dass Realistic Vision 5.1 sich aufgrund der photorealistischen Fähigkeiten besser für die Datensatzerstellung eignete als das Stable Diffusion 1.5 Basismodell. Darüber hinaus wurde festgestellt, dass Bilderkennungsmodelle, die auf segmentierten Datensätzen trainiert wurden, bessere Ergebnisse erzielten als solche, die auf nicht segmentierten Datensätzen trainiert wurden. Dies unterstreicht die Bedeutung von qualitativ hochwertigen und gut strukturierten Trainingsdaten für die Leistungsfähigkeit der Bilderkennungsmodelle.

Trotz der erzielten Fortschritte wurden auch Herausforderungen und Verbesserungspotenziale identifiziert. Die generierten Verschmutzungsmuster waren nicht realistisch genug, um eine zuverlässige Generalisierung zu gewährleisten. Daher ist es notwendig, die Generierung der Verschmutzungsbilder durch einen größeren und realistischeren Trainingsdatensatz zu verbessern. Darüber hinaus sollte in zukünftigen Studien versucht werden, einen größeren

und ausgewogeneren Evaluierungsdatensatz zu verwenden, um die die Aussagekraft der Modellevaluierung zu erhöhen.

Als Beitrag zur Forschung demonstriert diese Arbeit die Machbarkeit und Effizienz der Verwendung von Stable Diffusion Modellen zur automatisierten Generierung von Trainingsdatensätzen für die Bilderkennung. Die entwickelte Methode zur automatisierten Datensatzerstellung ermöglicht eine effiziente und benutzerfreundliche Generierung von Trainingsdaten. Das entwickelte Framework wird öffentlich zugänglich gemacht, um anderen Forschern und Anwendern die Möglichkeit zu geben, die Stable Diffusion Technologie für ihre eigenen Anwendungen zu nutzen.

Zusammenfassend bietet diese Arbeit einen fundierten Ansatz zur Anwendung von Stable Diffusion zur Datensatzgenerierung für die Bilderkennung. Gleichzeitig wurden Forschungsbedarf und Verbesserungspotenziale identifiziert, die als Basis für weiterführende Studien dienen können. Die Ergebnisse dieser Arbeit zeigen, dass die Technologie der Stable Diffusion vielversprechend ist und das Potenzial hat, die Bilderkennung und Datensatzerstellung in verschiedenen Anwendungsbereichen zu revolutionieren. Es ist zu erwarten, dass zukünftige Entwicklungen vor allem mit Blick auf Stable Diffusion 3 die Qualität und Leistungsfähigkeit der vorgestellten Methodik weiter verbessert.

Literatur

Bergfeld, M. (2022). UNI Europa Literature Review: Labour Shortages and Turnover in Industrial Cleaning, Long-term Care and Private Security. *RETAIN Project (VS/2019/0292)*. Verfügbar 7. März 2024 unter https://www.academia.edu/84989885/UNI_Europa_Literature_Review_Labour_Shortages_and_Turnover_in_Industrial_Cleaning_Long_term_Care_and_Private_Security

Canedo, D., Fonseca, P., Georgieva, P., & Neves, A. J. R. (2021). A Deep Learning-Based Dirt Detection Computer Vision System for Floor-Cleaning Robots with Improved Data Collection. *Technologies*, *9*(4), 94. https://doi.org/10.3390/technologies9040094

Chaudhari, P. R., & Bagde, A. (2021). Smart Washroom Cleaning System Using Hub Technology. *2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT)*, 550–552. https://doi.org/10.1109/ICEECCOT52851.2021.9707953

Chen, W., Yao, M., Zhu, Z., Sun, Y., & Han, X. (2022). The application research of AI image recognition and processing technology in the early diagnosis of the COVID-19. *BMC Medical Imaging*, *22*(1), 1–10. https://doi.org/10.1186/s12880-022-00753-1

Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., & Jitsev, J. (2022, 14. Dezember). *Reproducible Scaling Laws for Contrastive Language-Image Learning*. arXiv: 2212.07143 [cs]. https://doi.org/10.48550/arXiv.2212.07143

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li & Li Fei-Fei. (2009). ImageNet: A Large-Scale Hierarchical Image Database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Devi, S., Albraikan, A., Al-Wesabi, F., Nour, M., Ashour, A., & Hilal, A. (2023). Intelligent Deep Convolutional Neural Network Based Object Detection Model for Visually Challenged People. *Computer Systems Science and Engineering*, *46*(3), 3191–3207. https://doi.org/10.32604/csse.2023.036980

Dong, X., & Cappuccio, M. L. (2023, 15. November). *Applications of Computer Vision in Autonomous Vehicles: Methods, Challenges and Future Directions*. arXiv: 2311.09093 [cs]. Verfügbar 30. Mai 2024 unter http://arxiv.org/abs/2311.09093

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Lacey, K., Goodwin, A., Marek, Y., & Rombach, R. (2024, 5. März). *Scaling Rectified Flow Transformers for High-Resolution*

Image Synthesis. arXiv: 2403.03206 [cs]. Verfügbar 20. Juni 2024 unter http://arxiv.org/abs/2403.03206

Eurofound & Mandl, I. (2021). *The Digital Age – Implications of Automation, Digitisation and Platforms for Work and Employment*. Publications Office of the European Union. https://doi.org/doi/10.2806/288

Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., & Cohen-Or, D. (2022, 2. August). *An Image Is Worth One Word: Personalizing Text-to-Image Generation Using Textual Inversion*. arXiv: 2208.01618 [cs]. https://doi.org/10.48550/arXiv.2208.01618

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. The MIT Press. http://www.deeplearningbook.org

Ho, J., Jain, A., & Abbeel, P. (2020, 16. Dezember). *Denoising Diffusion Probabilistic Models*. arXiv: 2006.11239 [cs, stat]. https://doi.org/10.48550/arXiv.2006.11239

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021, 16. Oktober). *LoRA: Low-Rank Adaptation of Large Language Models*. arXiv: 2106.09685 [cs]. https://doi.org/10.48550/arXiv.2106.09685

Jayasinghe, L., Wijerathne, N., Yuen, C., & Zhang, M. (2019). Feature Learning and Analysis for Cleanliness Classification in Restrooms. *IEEE Access*, *7*, 14871–14882. https://doi.org/10.1109/ACCESS.2019.2894006

Karras, T., Aittala, M., Aila, T., & Laine, S. (2022, 11. Oktober). *Elucidating the Design Space of Diffusion-Based Generative Models*. arXiv: 2206.00364 [cs, stat]. https://doi.org/10.48550/arXiv.2206.00364

Kingma, D. P., & Welling, M. (2019). An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, *12*(4), 307–392. https://doi.org/10.1561/2200000056

Kingma, D. P., & Welling, M. (2022, 10. Dezember). *Auto-Encoding Variational Bayes*. arXiv: 1312.6114 [cs, stat]. https://doi.org/10.48550/arXiv.1312.6114

Kramer, M. (1992). Autoassociative neural networks. *Computers & Chemical Engineering*, *16*(4), 313–328. https://doi.org/10.1016/0098-1354(92)80051-A

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., & Ferrari, V. (2020). The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale. *International Journal of Computer Vision*, *128*(7), 1956–1981. https://doi.org/10.1007/s11263-020-01316-z

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2015, 20. Februar). *Microsoft COCO: Common Objects in Context*. arXiv: 1405.0312 [cs]. Verfügbar 30. Mai 2024 unter http://arxiv.org/abs/1405.0312

Mendelevitch, O., Lesh, M. D., Mendelevitch, O., & Lesh, M. D. (2020, 9. September). Beyond Differential Privacy: Synthetic Micro-Data Generation with Deep Generative Neural Networks. In *Security and Privacy From a Legal, Ethical, and Technical Perspective*. IntechOpen. https://doi.org/10.5772/intechopen.92255

Mezil, A. (2023, 5. Mai). *Cleaning Companies and Data Analytics: Improving Service Quality*. Hellamaid. Verfügbar 4. Juni 2024 unter https://hellamaid.ca/technology/cleaning-companies-and-data-analytics-improving-service-quality/

Paaß, G., & Hecker, D. (2020). Bilderkennung mit tiefen neuronalen Netzen. In G. Paaß & D. Hecker (Hrsg.), *Künstliche Intelligenz: Was steckt hinter der Technologie der Zukunft?* (S. 119–166). Springer Fachmedien. https://doi.org/10.1007/978-3-658-30211-5_5

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., & Rombach, R. (2023, 4. Juli). *SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis*. arXiv: 2307.01952 [cs]. https://doi.org/10.48550/arXiv.2307.01952

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021, 26. Februar). *Learning Transferable Visual Models From Natural Language Supervision*. arXiv: 2103.00020 [cs]. https://doi.org/10.48550/arXiv.2103.00020

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022, 13. April). *High-Resolution Image Synthesis with Latent Diffusion Models*. arXiv: 2112.10752 [cs]. https://doi.org/10.48550/arXiv.2112.10752

Ronneberger, O., Fischer, P., & Brox, T. (2015, 18. Mai). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv: 1505.04597 [cs]. https://doi.org/10.48550/arXiv.1505.04597

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D. J., & Norouzi, M. (2022, 23. Mai). *Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding*. arXiv: 2205.11487 [cs]. https://doi.org/10.48550/arXiv.2205.11487

See-To, E. W. K., Wang, X., Lee, K.-Y., Wong, M.-L., & Dai, H.-N. (2023). Deep-Learning-Driven Proactive Maintenance Management of IoT-Empowered Smart Toilet. *IEEE Internet of Things Journal*, *10*(3), 2417–2429. https://doi.org/10.1109/JIOT.2022.3211889

Shenoda, M., & Kim, E. (2023, 1. September). *DiffuGen: Adaptable Approach for Generating Labeled Image Datasets Using Stable Diffusion Models*. arXiv: 2309.00248 [cs]. https://doi.org/10.48550/arXiv.2309.00248

Simonyan, K., & Zisserman, A. (2015, 10. April). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv: 1409.1556 [cs]. https://doi.org/10.48550/arXiv.1409.1556

Song, J., Meng, C., & Ermon, S. (2022, 5. Oktober). *Denoising Diffusion Implicit Models*. arXiv: 2010.02502 [cs]. https://doi.org/10.48550/arXiv.2010.02502

Sonnet, D. (2022). *Neuronale Netze kompakt: Vom Perceptron zum Deep Learning*. Springer Fachmedien. https://doi.org/10.1007/978-3-658-29081-8

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023, 1. August). *Attention Is All You Need*. arXiv: 1706.03762 [cs]. https://doi.org/10.48550/arXiv.1706.03762

Wiggers, K. (2022, 17. Oktober). *Stability AI, the startup behind Stable Diffusion, raises $101M*. TechCrunch. Verfügbar 27. Mai 2024 unter https://techcrunch.com/2022/10/17/stability-ai-the-startup-behind-stable-diffusion-raises-101m/

Yun, Y., Hou, L., Feng, Z., Jin, W., Liu, Y., Wang, H., He, R., Guo, W., Han, B., Qin, B., & Li, J. (2022). A Deep-Learning-based System for Indoor Active Cleaning. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7803–7808. https://doi.org/10.1109/IROS47612.2022.9982137

Weiterführende Literatur

Beginners Guide to How Stable Diffusion Works (Illustrative) - Excurl. (n. d.). Verfügbar 13. Juni 2024 unter https://www.excurl.com/2023/02/ultimate-dummies-illustration-guide-to-how-stable-diffusion-works.html

Blain, L. (2023, 18. August). *Toilet-cleaning robot makes us feel better about all this AI business*. New Atlas. Verfügbar 7. März 2024 unter https://newatlas.com/robotics/toilet-cleaning-robot-somatic/

Create high-quality images with Stable Diffusion models and deploy them cost-efficiently with Amazon SageMaker. (2023, 26. Mai). Snap-Tech News. Verfügbar 22. Juni 2024 unter https://snap-tech.com/technology/create-high-quality-images-with-stable-diffusion-models-and-deploy-them-cost-efficiently-with-amazon-sagemaker/

Diffusers. (n. d.). Verfügbar 22. Juni 2024 unter https://huggingface.co/docs/diffusers/en/index

Everything You Need To Know About Stable Diffusion. (n. d.). Verfügbar 22. Juni 2024 unter https://www.hyperstack.cloud/blog/case-study/everything-you-need-to-know-about-stable-diffusion

Feng, C., Zhong, Y., Jie, Z., Xie, W., & Ma, L. (2024, 19. Februar). *InstaGen: Enhancing Object Detection by Training on Synthetic Dataset*. arXiv: 2402.05937 [cs]. https://doi.org/10.48550/arXiv.2402.05937

Graph, R. (2024, 22. Mai). *Diffusion Models: A Comprehensive High-Level Understanding*. Medium. Verfügbar 17. Juni 2024 unter https://medium.com/@researchgraph/diffusion-model-comprehensive-high-level-understanding-55d6ecad2cba

LoRA vs Dreambooth vs Textual Inversion vs Hypernetworks - YouTube. (n. d.). Verfügbar 22. Juni 2024 unter https://www.youtube.com/watch?v=dVjMiJsuR5o

martinellison. (2023, 1. August). *Hardware Requirements for SDXL on A1111?* r/StableDiffusion. Verfügbar 22. Juni 2024 unter www.reddit.com/r/StableDiffusion/comments/15ez4ow/hardware_requirements_for_sdxl_on_a1111/

The ML developers guide to Schedulers in Stable Diffusion. (2023, 26. Mai). Automagically by Segmind. Verfügbar 22. Juni 2024 unter https://blog.segmind.com/what-are-schedulers-in-stable-diffusion/

NAMA-ADMIN. (n. d.). *Beginners Guide to How Stable Diffusion Works (Illustrative)*. excurl.

Verfügbar 22. Juni 2024 unter https://www.excurl.com/2023/02/ultimate-dummies-illustration-guide-to-how-stable-diffusion-works.html

Nguyen, Q., Vu, T., Tran, A., & Nguyen, K. (2023). Dataset Diffusion: Diffusion-based Synthetic Data Generation for Pixel-Level Semantic Segmentation. *Advances in Neural Information Processing Systems*, *36*, 76872–76892. Verfügbar 14. März 2024 unter https://proceedings.neurips.cc/paper_files/paper/2023/hash/f2957e48240c1d90e62b303574871b47-Abstract-Conference.html

Realistic Vision V6.0 B1 - V5.1 Hyper (VAE) | Stable Diffusion Checkpoint | Civitai. (2024, 31. Mai). Verfügbar 7. Juni 2024 unter https://civitai.com/models/4201/realistic-vision-v60-b1

RealVisXL V4.0 - V4.0 Lightning (BakedVAE) | Stable Diffusion Checkpoint | Civitai. (2024, 25. Mai). Verfügbar 7. Juni 2024 unter https://civitai.com/models/139562/realvisxl-v40

SDXL VRAM System Requirements - Recommended GPU, CPU, and RAM for Stable Diffusion to Run Locally. (n. d.). Verfügbar 22. Juni 2024 unter https://gofind.ai/stable-diffusion/sdxl-system-requirements

Shpuntov, I. (2023, 5. Dezember). *Enhancing Video Creativity with AI: Stable Diffusion and AnimateDiff with conditions…* Medium. Verfügbar 13. Juni 2024 unter https://medium.com/@cultural.nickname/enhancing-video-creativity-with-ai-stable-diffusion-and-animatediff-with-conditions-1fb96592e3f1

Wu, W., Zhao, Y., Chen, H., Gu, Y., Zhao, R., He, Y., Zhou, H., Shou, M. Z., & Shen, C. (2023). DatasetDM: Synthesizing Data with Perception Annotations Using Diffusion Models. *Advances in Neural Information Processing Systems*, *36*, 54683–54695. Verfügbar 23. März 2024 unter https://proceedings.neurips.cc/paper_files/paper/2023/hash/ab6e7ad2354f350b451b5a8e14d04f51-Abstract-Conference.html

Anhang

Grundlagen

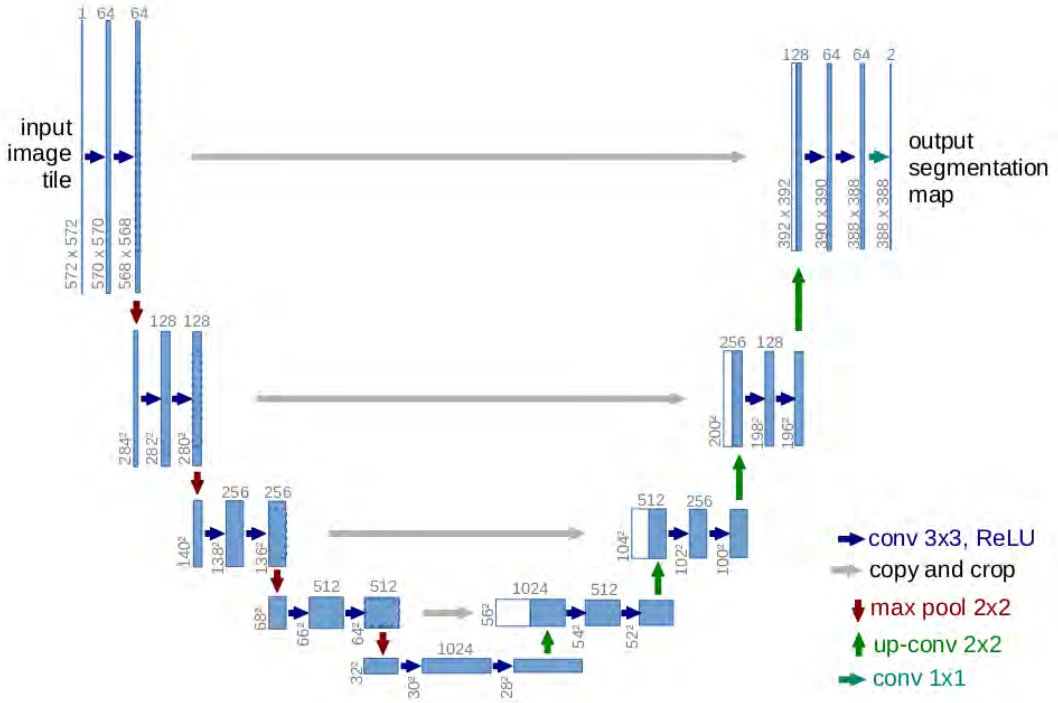

Abbildung A.1: UNet Architektur nach Ronneberger et al. (2015)

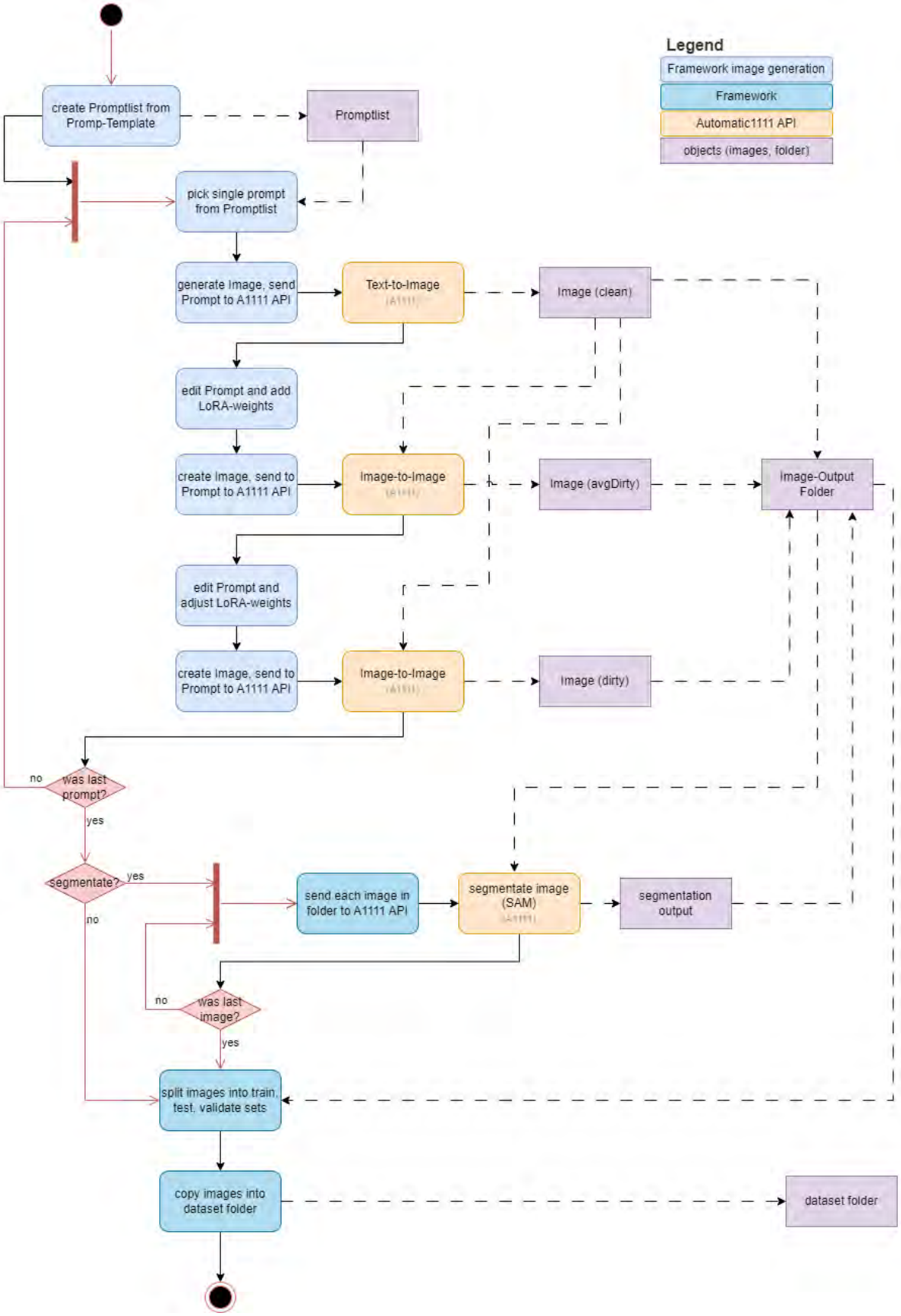

Abbildung A.2: UML Diagramm des Frameworks zur Datensatzgenerierung

Ergebnisse

Stable Diffusion Training

Abbildung A.3: urinal-LoRA Beispielbilder des Trainingsdatensatzes (10 von 17 Bildern)

Abbildung A.4: dirty-LoRA Bilder des Trainingsdatensatzes

Abbildung A.5: urinal-LoRA generierte Beispielbilder während des Trainings

Abbildung A.6: dirty-LoRA generierte Beispielbilder während des Trainings

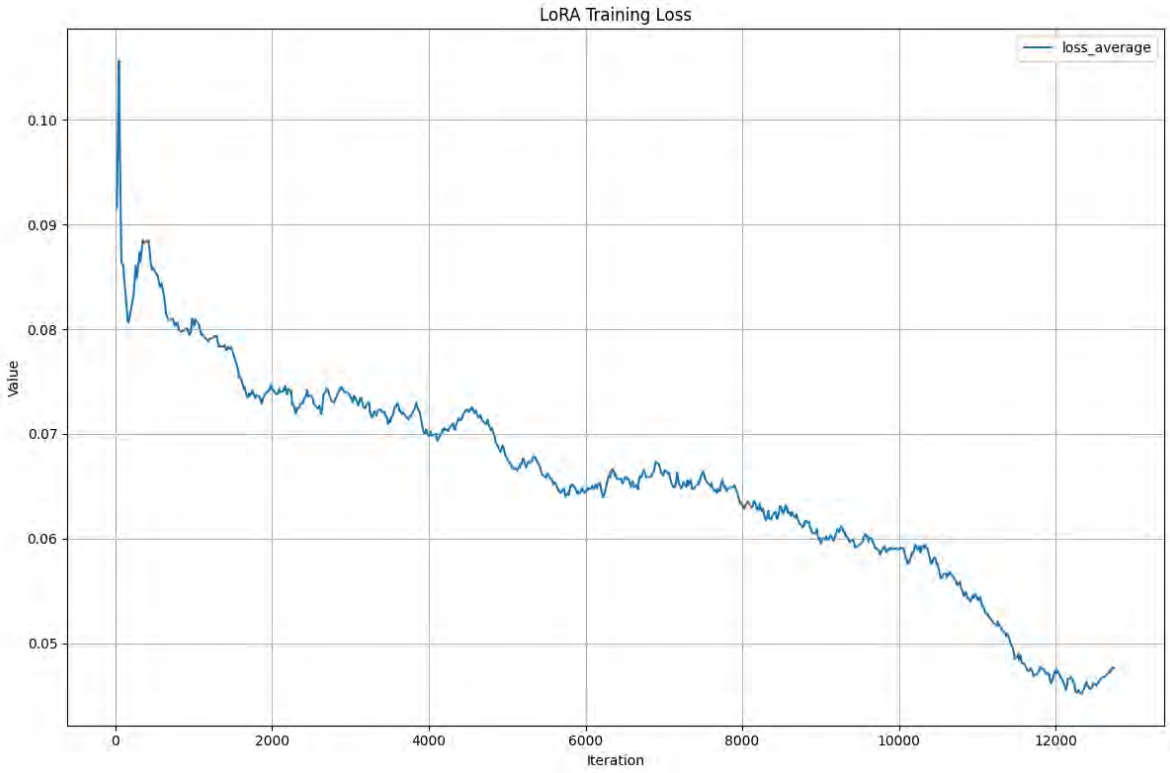

Abbildung A.7: Verlauf des Trainings-Loss für urinal-LoRA

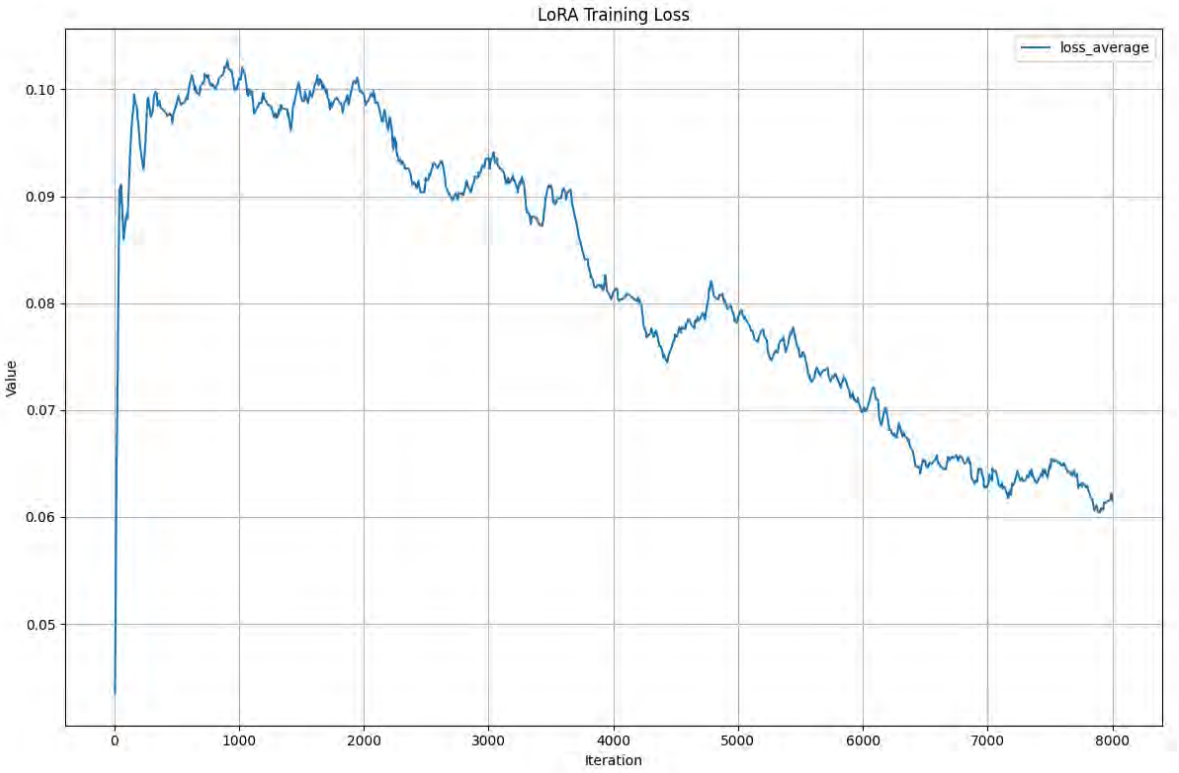

Abbildung A.8: Verlauf des Trainings-Loss für dirty-LoRA

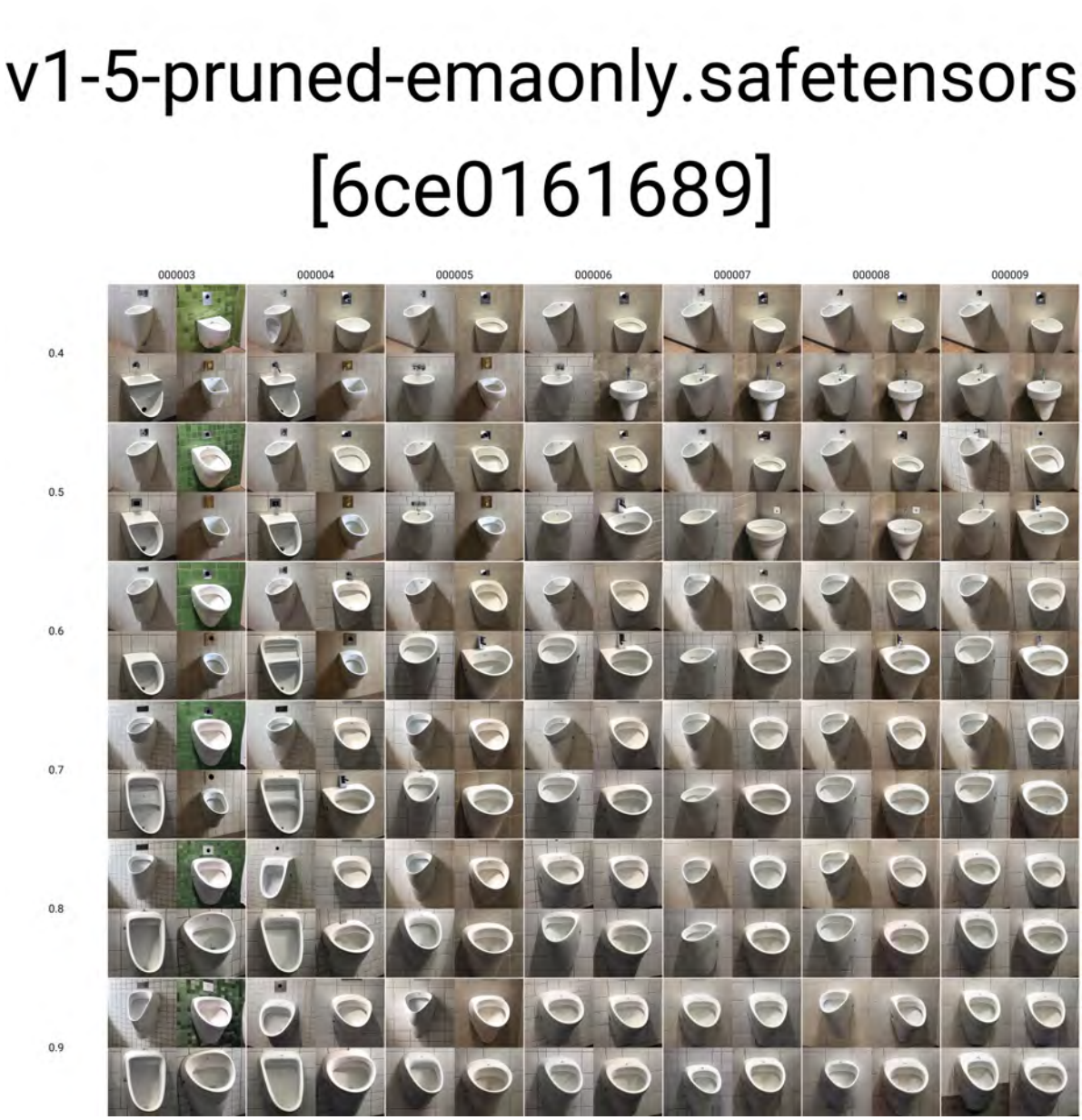

Abbildung A.9: Vergleich urinal-LoRA Epoche und Gewicht auf dem Stable Diffusion 1.5 Modell

Abbildung A.10: Vergleich urinal-LoRA Epoche und Gewicht auf dem Realistic Vision 5.1 Modell

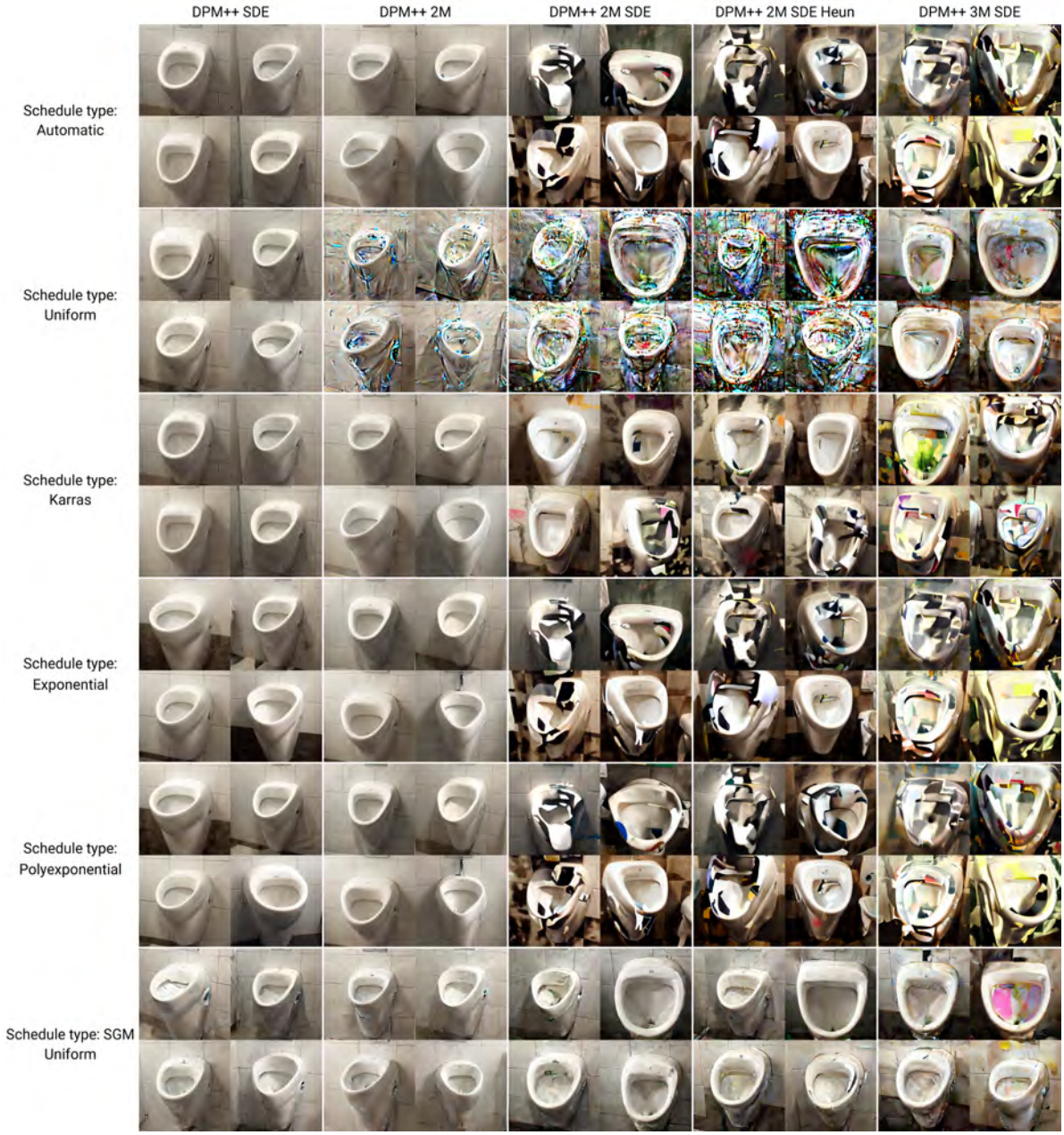

Abbildung A.11: Vergleich Sampler und Scheduler bei CFG-Scale 7 und urinalLora_v2-ep7_RealVis-80 Modell

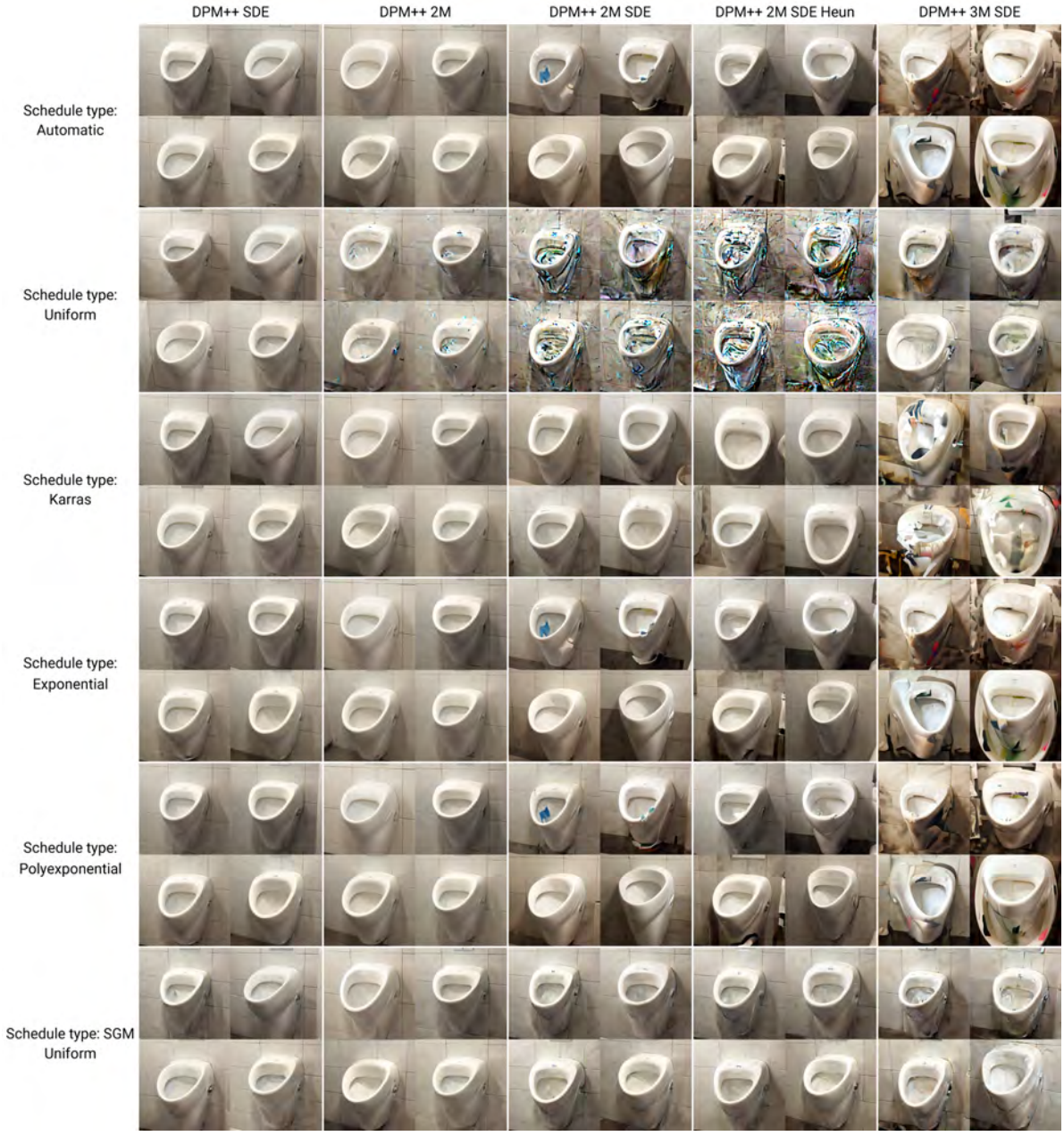

Abbildung A.12: Vergleich Sampler und Scheduler bei CFG-Scale 5 und urinalLora_v2-ep7_RealVis-80 Modell

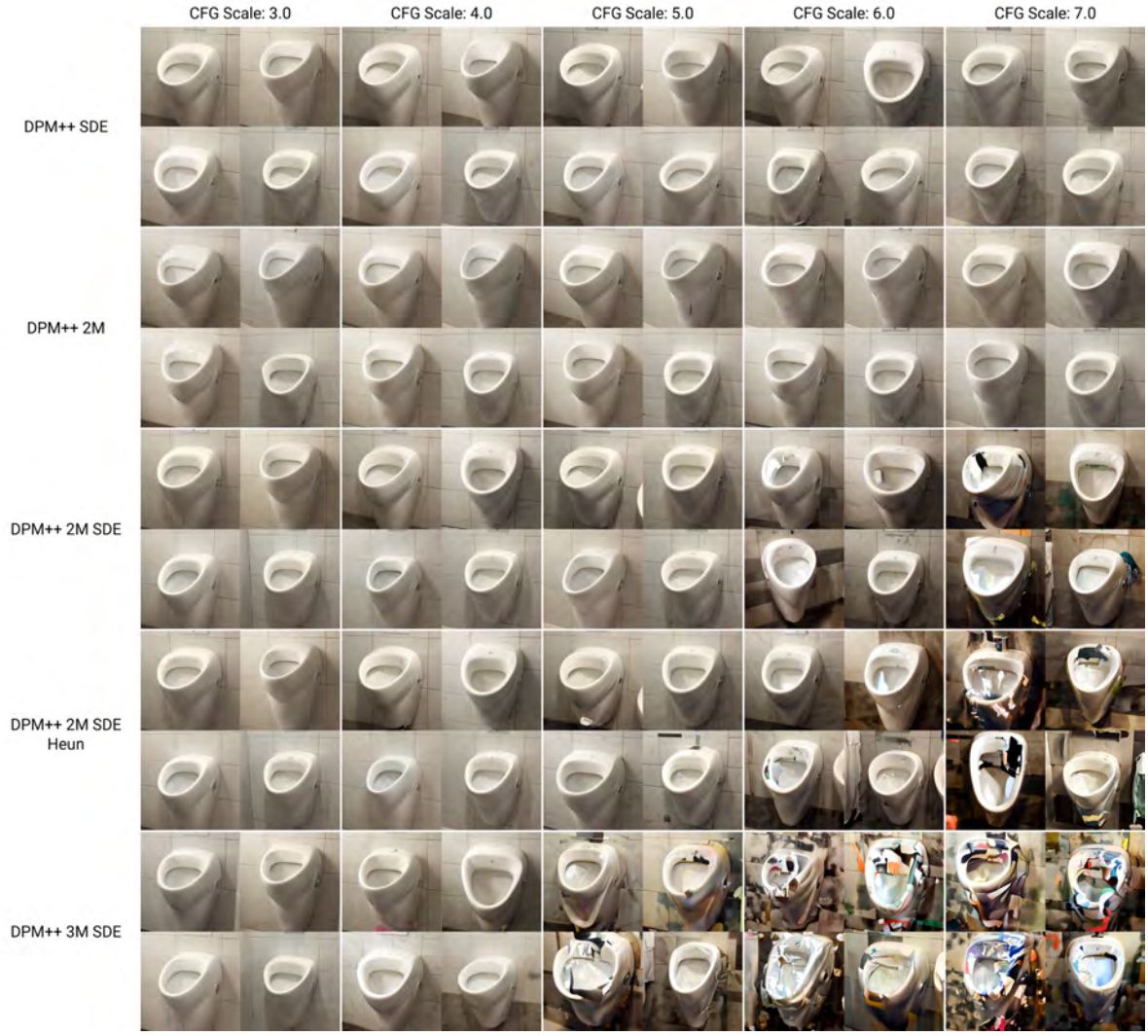

Abbildung A.13: Vergleich CFG-Scale und Sampler bei Scheduler Karras und urinalLora_v2-ep7_RealVis-80 Modell

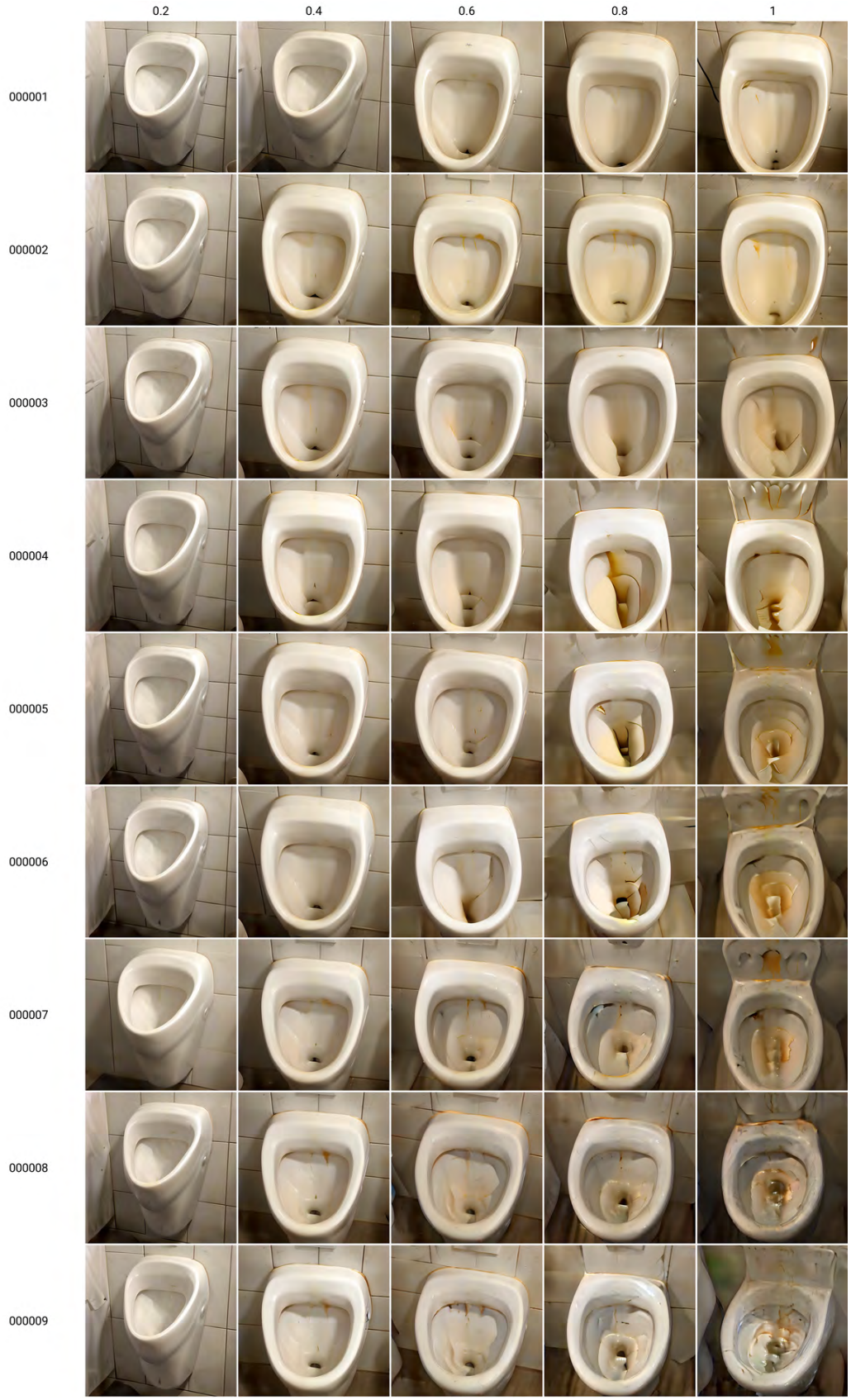

Abbildung A.14: Vergleich dirty-LoRA Epoche und Gewicht auf dem urinalLoRA_v2-ep7_RealVis-80 Modell

Datensatzgenerierung

Codeblock A.1: prompt_config_v11.json, verwendet zur Datensatzgenerierung

```
{
    "name": "urinalDataset_v11",
    "txt2img": {
        "object_names": ["urinal"],
        "object_materials": ["white porcelain", "white"],
        "backgrounds": ["white tile wall, tile floor",  "white wall, tile floor", "white tile wall", "white wall"],
        "perspectives": ["front view", "top view", "from side"],
        "viewpoints": ["close-up"],
        "seeds_per_prompt": 6,
        "prompts": [
            "{object_name}, a {object_material} {object_name} in a bathroom, {background}, {perspective}, wall mounted",
            "{object_name}, a {object_material} {object_name} in a bathroom, {background}, {perspective}, wall mounted, shadow cast",
            "{object_name}, a {object_material} {object_name} in a bathroom, {background}, {perspective}, wall mounted, soft light",
            "{object_name}, a {object_material} {object_name} in a bathroom, {background}, {perspective}, wall mounted, reflection"
        ]
    }
}
```

Abbildung A.15: Beispielbilder des generierten Datensatzes v12

Abbildung A.16: Beispielbilder des generierten Datensatzes v12_seg

Abbildung A.17: Beispielbilder des generierten Datensatzes v13

Abbildung A.18: Beispielbilder des generierten Datensatzes v13_seg

Abbildung A.19: Beispielbilder des generierten Datensatzes v14

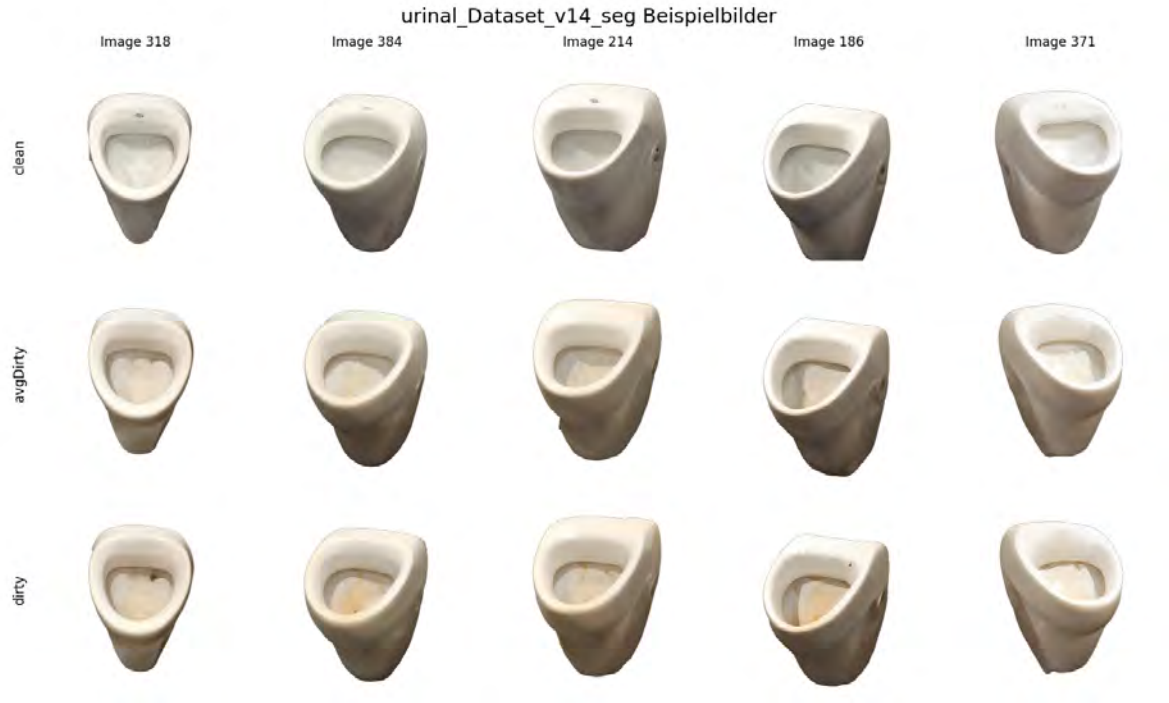

Abbildung A.20: Beispielbilder des generierten Datensatzes v14_seg

Abbildung A.21: Beispielbilder des generierten Datensatzes v15

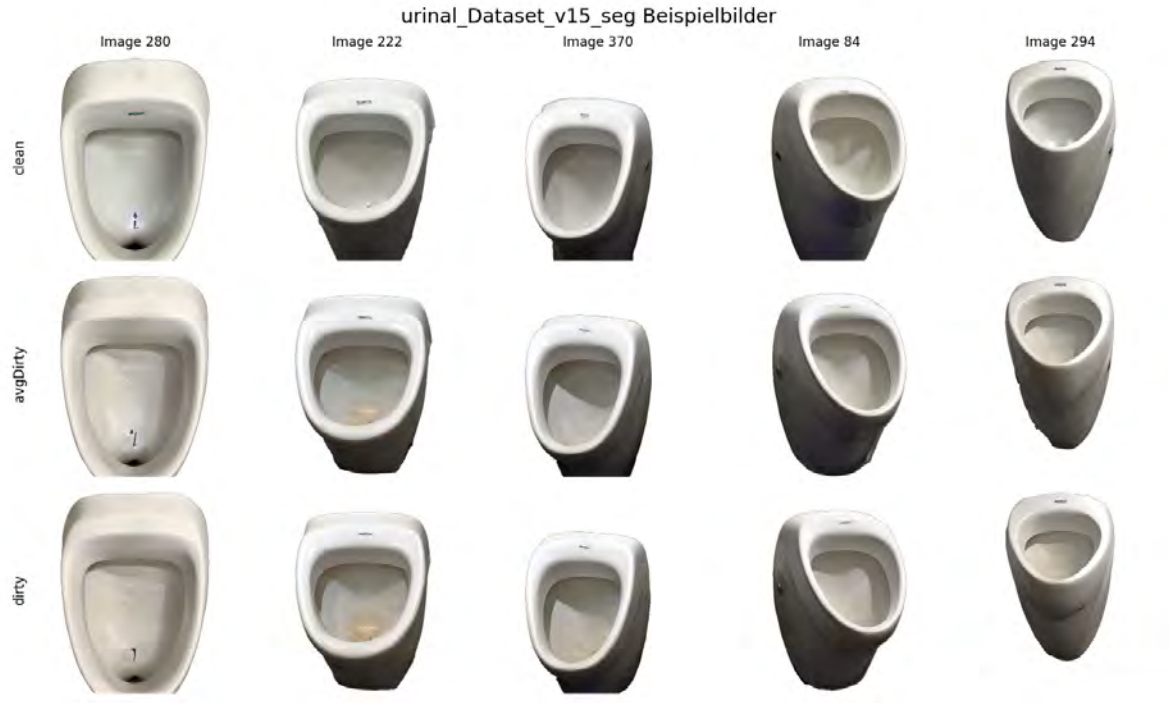

Abbildung A.22: Beispielbilder des generierten Datensatzes v15_seg

Abbildung A.23: Beispielbilder für Bildfehler in den generierten Datensätzen

Abbildung A.24: Beispielbilder des Evaluierungsdatensatzes mit realen Bildern

Dataset	Loss	Accuracy	Recall	Precision	Best Epoch
v12	0.1976	0.9520	0.9520	0.9520	18
v12_seg	0.1372	0.9669	0.9669	0.9668	22
v13	0.1770	0.9562	0.9562	0.9563	32
v13_seg	0.2759	0.9297	0.9297	0.9295	26
v14	0.1485	0.9628	0.9628	0.9629	20
v14_seg	0.0794	0.9801	0.9801	0.9802	42
v15	0.1878	0.9487	0.9487	0.9490	26
v15_seg	0.3290	0.9049	0.9049	0.9060	19

Tabelle A.1: Bilderkennung Trainings-Leistungsmetriken der besten Epoche

Dataset	Loss	Accuracy	Recall	Precision	Best Epoch
v12	0.1425	0.9478	0.9478	0.9475	18
v12_seg	0.0873	0.9594	0.9594	0.9592	22
v13	0.1810	0.9130	0.9130	0.9161	32
v13_seg	0.2717	0.9072	0.9072	0.9072	26
v14	0.0670	0.9739	0.9739	0.9739	20
v14_seg	0.0916	0.9594	0.9594	0.9594	42
v15	0.1546	0.9420	0.9420	0.9424	26
v15_seg	0.2203	0.9130	0.9130	0.9131	19

Tabelle A.2: Bilderkennung Test-Leistungsmetriken der besten Epoche

Dataset	Loss	Accuracy	Recall	Precision
v12	0.1272	0.9310	0.9310	0.9305
v12_seg	0.0973	0.9540	0.9540	0.9538
v13	0.1964	0.9080	0.9080	0.9091
v13_seg	0.2579	0.9080	0.9080	0.9091
v14	0.0956	0.9713	0.9713	0.9735
v14_seg	0.0556	0.9770	0.9770	0.9770
v15	0.2161	0.9195	0.9195	0.9352
v15_seg	0.1801	0.9253	0.9253	0.9317

Tabelle A.3: Bildererkennung Validierungs-Leistungsmetriken

Dataset	Loss	Accuracy	Recall	Precision
v12	6.6142	0.6389	0.6389	0.5000
v12_seg	6.0023	0.7500	0.7500	0.7500
v13	3.6577	0.6667	0.6667	0.5500
v13_seg	5.5489	0.6667	0.6667	0.5833
v14	5.1860	0.6944	0.6944	0.5556
v14_seg	23.9548	0.5000	0.5000	0.2667
v15	3.0968	0.7778	0.7778	0.6667
v15_seg	5.8559	0.7778	0.7778	0.7500

Tabelle A.4: Bilderkennung Evaluations-Leistungsmetriken (reale Bilder)

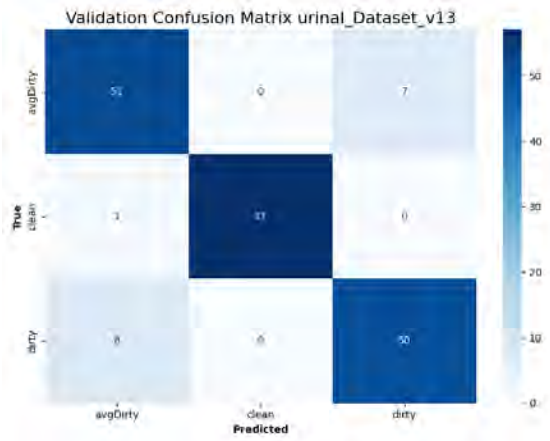

Abbildung A.25: Validierungs-Konfusionsmatrix für Datensatz v13

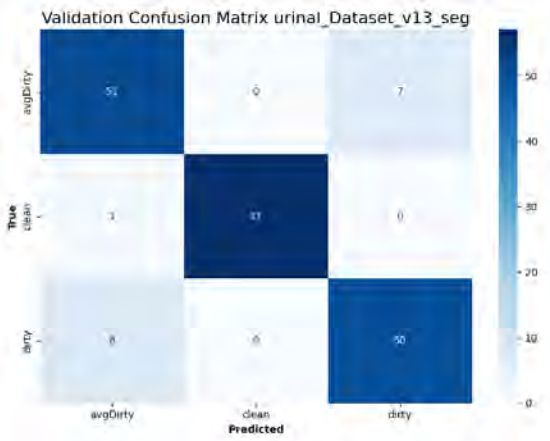

Abbildung A.26: Validierungs-Konfusionsmatrix für Datensatz v13_seg

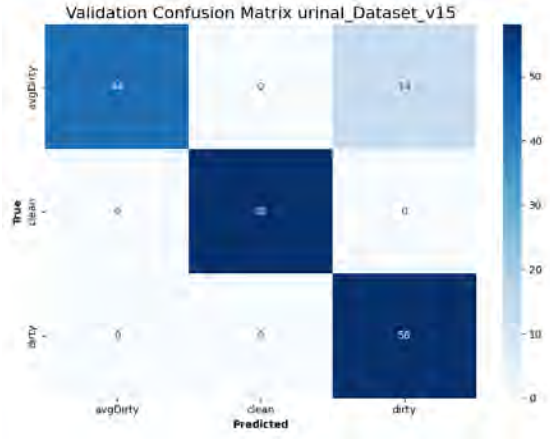

Abbildung A.27: Validierungs-Konfusionsmatrix für Datensatz v15

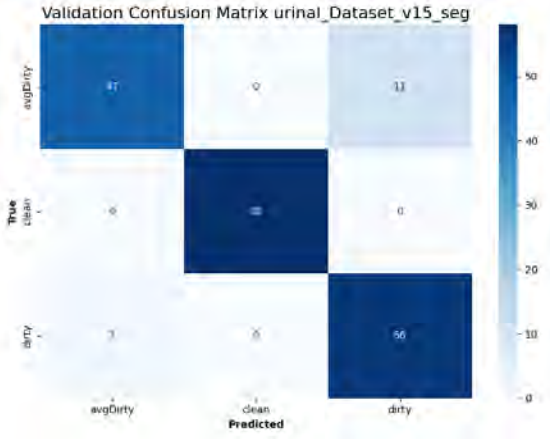

Abbildung A.28: Validierungs-Konfusionsmatrix für Datensatz v15_seg

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit mit dem Titel

Nutzung von Stable Diffusion zur Datensatzgenerierung für die Bilderkennung von Verschmutzungen in Sanitärbereichen

selbstständig und nur mit den angegebenen Hilfsmitteln verfasst habe.

Ich erkläre weiterhin, dass ich bei der Erstellung dieser Arbeit keine unzulässige Hilfe in Anspruch genommen habe und die Arbeit frei von Plagiaten ist.

Folgende KI-Tools wurden zur Unterstützung bei der Recherche, Datenanalyse, Programmierung und Textformulierung verwendet:

- ChatGPT (OpenAI) – zur Literaturrecherche, Datenanalyse und Formulierung von Texten
- GitHub Copilot – zur Programmierunterstützung

Die Verwendung dieser Tools erfolgte unter strikter Einhaltung wissenschaftlicher Standards und Prinzipien. Alle von den Tools generierten Inhalte wurden von mir sorgfältig geprüft, überarbeitet und gegebenenfalls korrigiert. Alle zitierten und paraphrasierten Inhalte sind entsprechend gekennzeichnet.

Ort, Datum Unterschrift