

Master-Teamprojekt

Titel der Arbeit

Titel of the Thesis

Max Mustermann

Mat.Nr.: 420815

Betreuer:

Prof. Dr. rer. nat. E.-G. Haffner

Datum:

14. März 2022



---

# Eidesstattliche Erklärung

---

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig verfasst habe und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht. Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

---

Ort, Datum

---

Unterschrift



---

## Danksagung

---

Ich bedanke mich bei all denjenigen, die mich während der Anfertigung dieser Arbeit unterstützt und motiviert haben. Ein ganz besonderer Dank gilt meinem Betreuer Prof. Dr. rer. nat. E.-G. Haffner. Ebenfalls möchte ich der Projektgruppe eTRainer der Hochschule Trier, sowie meinen Korrekturlesenden danken.



---

# Abstract

---

This is a summary of all the important points and achievements of this work.

# Zusammenfassung

---

Hierbei handelt es sich um eine Zusammenfassung aller wichtige Punkte und Errungenschaften dieser Arbeit.





# Abkürzungsverzeichnis

<b>ARS</b>	Audience Response System
<b>CGI</b>	Common Gateway Interface
<b>CSS</b>	Cascading Style Sheets
<b>HTML</b>	Hypertext Markup Language
<b>ML</b>	Machine Learning



# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
<b>2. Ein Kapitel</b>	<b>3</b>
2.1. Ein Abschnitt . . . . .	3
2.1.1. Ein Unterabschnitt . . . . .	3
2.2. Pix2Pix . . . . .	3
2.2.1. Pix2Pix-Kernkonzepte . . . . .	4
<b>3. Hauptteil - 1</b>	<b>7</b>
<b>4. Hauptteil - 2</b>	<b>9</b>
<b>5. Hauptteil - 3</b>	<b>11</b>
<b>6. Evaluation</b>	<b>13</b>
<b>7. Fazit und Ausblick</b>	<b>15</b>
7.1. Fazit . . . . .	15
7.2. Ausblick . . . . .	15
<b>A. Anhang - Code</b>	<b>19</b>
<b>B. Anhang - Dokumentationen</b>	<b>21</b>
<b>Verzeichnisse</b>	<b>23</b>
Literaturverzeichnis . . . . .	23
Abbildungsverzeichnis . . . . .	25
Tabellenverzeichnis . . . . .	27
Code-Auszugs-Verzeichnis . . . . .	29
Glossar . . . . .	31



## 1

## Einleitung

Hier wird in die Arbeit eingeleitet.



## 2

## Ein Kapitel

## 2.1. Ein Abschnitt

### 2.1.1. Ein Unterabschnitt

## 2.2. Pix2Pix

Pix2Pix, initiiert von Isola et al., hat sich als zentrales Framework für Bild-zu-Bild-Übersetzungen auf der Basis von bedingten generativen adversariellen Netzwerken (cGANs) etabliert. Es ermöglicht die Erstellung einer abstrakten Abbildung von einem Eingangsbild zu einem korrespondierenden Ausgangsbild und bewältigt dabei eine vielfältige Palette an Bildübersetzungsaufgaben, wie die Transformation von Skizzen in realistische Bilder oder die Konvertierung von Tages- zu Nachtaufnahmen.

Pix2Pix fungiert hier als Generative Adversarial Network (GAN), spezialisiert auf diverse Formen der Bildübersetzung. Darunter fallen die Umwandlung von Schwarz-Weiß-Fotos in Farbbilder, die Transformation von Skizzen in realistische Bilder, und relevant für diese Arbeit, die Konvertierung von Satellitenbildern in kartographische Darstellungen, ähnlich den Visualisierungen von Google Maps.

Die Architektur von Pix2Pix besteht aus einem Generator und einem Diskriminator. Der Generator, der eine U-Net-Architektur verwendet, besteht aus einem Encoder und einem Decoder. Der Encoder komprimiert das Eingangsbild schrittweise zu einer niedrigdimensionalen Repräsentation, während der Decoder diese dazu nutzt, das Ausgangsbild zu rekonstruieren. Skip-Verbindungen zwischen Encoder und Decoder helfen dabei, sowohl globale als auch lokale Informationen im generierten Bild zu bewahren.

Der Diskriminator nimmt die Form eines PatchGAN-Modells an und bewertet Patches des Bildes, indem er die Wahrscheinlichkeit für die Echtheit jedes Patches ausgibt. Dies ermöglicht die Anwendung des Diskriminators auf Bilder unterschiedlicher Größen. Im Zuge des adversariellen Trainingsprozesses passen sowohl der Generator als auch der Diskriminator ihre Fähigkeiten fortlaufend an. Während der Generator lernt, immer realistischere Übersetzungen zu erzeugen, wird der Diskriminator stetig besser darin, zwischen echten und generierten Bildern zu unterscheiden.

### 2.2.1. Pix2Pix-Kernkonzepte

#### Generator

Die Bildverarbeitung hat in den letzten Jahren durch den Einsatz tiefer neuronaler Netzwerke erhebliche Fortschritte gemacht. Im Mittelpunkt vieler dieser Fortschritte steht die U-Net-Architektur, die speziell für die Bildsegmentierung entwickelt wurde. Diese Architektur zeichnet sich durch ihre angeklügelte Kombination aus Encoder- und Decoder-Strukturen sowie durch den Einsatz von Skip-Verbindungen aus.

Bei der Encoder-Decoder-Struktur handelt es sich um einen Ansatz, bei dem das Eingangsbild zunächst durch den Encoder schrittweise reduziert wird. Dieser Prozess dient dazu, wesentliche Merkmale des Bildes zu erfassen. Anschließend wird das Bild durch den Decoder wiederhergestellt, indem die zuvor extrahierten Merkmale verwendet werden. Während dieser Prozesse besteht jedoch das Risiko des Informationsverlustes, insbesondere in den tieferen Schichten des Netzwerks. Um dieses Problem zu adressieren, führt die U-Net-Architektur Skip-Verbindungen ein. Diese direkten Verbindungen zwischen korrespondierenden Schichten des Encoders und Decoders sorgen dafür, dass Detailinformationen nicht verloren gehen. Genauer gesagt, ermöglichen diese Verbindungen den direkten Informationsfluss zwischen jeweils äquivalenten Schichten, wodurch die Rekonstruktion des Bildes im Decoder mit einer höheren Genauigkeit erfolgt.

Die Bedeutung von Skip-Verbindungen zeigt sich insbesondere in Anwendungen wie der Bild-zu-Bild-Übersetzung. Hier muss oft ein Bild mit niedriger Auflösung in ein Bild mit hoher Auflösung überführt werden, ohne dass Details verloren gehen. Die U-Net-Architektur, die angereichert mit diesen Verbindungen ist, ermöglicht daher eine feinere Rekonstruktion, die sowohl globale als auch lokale Informationen berücksichtigt.

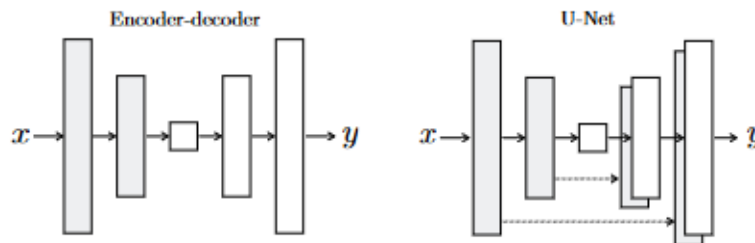
Somit kann die U-Net-Architektur durch ihre Kombination aus Encoder-Decoder-Struktur und Skip-Verbindungen ein effektives Werkzeug für die Bildsegmentierung darstellen. Ihre Fähigkeit, sowohl globale Muster als auch feine Details zu berücksichtigen, macht sie zu einer bevorzugten Wahl für viele Bildverarbeitungsaufgaben.

In Abbildung 2.1 ist die typische U-Net-Architektur dargestellt. Die linke Seite des "U" repräsentiert den Encoder-Teil, der das Eingangsbild schrittweise reduziert und wesentliche Merkmale extrahiert. Die rechte Seite repräsentiert den Decoder-Teil, der das Bild mithilfe der extrahierten Merkmale rekonstruiert. Die horizontalen Linien repräsentieren die Skip-Verbindungen, die sicherstellen, dass Detailinformationen zwischen den korrespondierenden Schichten des Encoder und Decoders direkt übertragen werden.

In der Pix2Pix Technologie dient diese U-Net-Architektur als Generator. Er ist das zentrale Element, das für die Bild-zu-Bild-Übersetzung verantwortlich ist. Die Wahl der U-Net-Struktur für den Generator liegt in ihrer Fähigkeit, feinere Details und Kontextinformationen aus dem Eingangsbild beizubehalten, was für die Bild-zu-Bild-Übersetzung von entscheidender Bedeutung ist. Die Encoder-



Decoder-Struktur des U-Net ermöglicht es dem Generator, den globalen Kontext des Bildes zu erfassen, während die Skip-Verbindungen sicherstellen, dass auch lokale Details im resultierenden Bild berücksichtigt werden.



**Abbildung 2.1.:** Schematische Darstellung der U-Net-Architektur. Die Architektur besteht aus einem Encoder-Teil (links), einem Decoder-Teil (rechts) und Skip-Verbindungen zwischen korrespondierenden Schichten.

## Diskriminator

Im adversariellen Lernprozess spielen Generatoren und Diskriminatoren eine entscheidende Rolle. Während der Generator versucht, Daten zu erzeugen, die von echten Daten kaum zu unterscheiden sind, evaluiert der Diskriminator die vom Generator erzeugten Daten und versucht, zwischen echten und gefälschten Daten zu unterscheiden.

Im Kontext von Generative Adversarial Networks (GANs), insbesondere im speziellen Fall des Pix2Pix GANs, spielt der PatchGAN-Diskriminator eine besonders wichtige Rolle. Der zentrale Unterschied dieses Diskriminators zu traditionellen Diskriminatoren liegt in der Art und Weise, wie er Bilder bewertet. Statt das gesamte Bild zu beurteilen, zerlegt der PatchGAN-Diskriminator das Bild in mehrere kleinere Bildabschnitte oder Patches und bewertet jeden Patch einzeln auf seine Echtheit.

Ein solches Vorgehen hat den klaren Vorteil, dass feinere Strukturen und Details in den Bildern erkannt und beurteilt werden können. Durch diese segmentierte Beurteilung kann der Diskriminator besser einschätzen, ob die Struktur und Beschaffenheit eines bestimmten Bildteils realistisch ist. Dies ist besonders nützlich, da kleinere Unstimmigkeiten in den Bildern, die ein traditioneller Diskriminator möglicherweise übersieht, vom PatchGAN erfasst werden können.

Ein weiterer Vorteil des PatchGAN-Diskriminators ist seine Skalierbarkeit. Da er auf festen Patchgrößen basiert, kann er flexibel auf Bilder unterschiedlicher Größen angewendet werden, ohne dass das zugrunde liegende Modell geändert werden muss. Dies führt nicht nur zu einer schnelleren Bildverarbeitung, sondern ermöglicht auch eine effiziente Ausführung auf großen Bildern. Darüber hinaus reduziert es potenzielle Kackelartefakte, die bei traditionellen Diskriminatoren auftreten können.

Eine Metrik, die oft verwendet wird, um die Leistung des Diskriminators zu beurteilen, ist der FCN-Score. Dieser bewertet die Qualität der vom Generator erzeugten

Bilder. Ein hoher FCN-Score zeigt, dass der Diskriminator erfolgreich echte von gefälschten Bildern unterscheiden kann.

Der PatchGAN-Diskriminator kann wenn er effektiv eingesetzt wird, zu besseren und realistischeren Bildern im adversariellen Lernprozess beitragen. Seine Fähigkeit, lokale Bildinformationen zu bewerten, ermöglicht es auch subtile Unterschiede in den Bildern zu erkennen, was zu einer verbesserten Qualität der generierten Bilder führt.

# 3

## Hauptteil - 1

Hier wird der Inhalt der Arbeit präsentiert.



# 4

## Hauptteil - 2

Hier wird der Inhalt der Arbeit präsentiert.



# 5

## Hauptteil - 3

Hier wird der Inhalt der Arbeit präsentiert.





# 6

## Evaluation

Hier wird die Arbeit evaluiert.



# 7

## Fazit und Ausblick

Hier wird ein Fazit und ein Ausblick gegeben.

### 7.1. Fazit

Fazit.

### 7.2. Ausblick

Ausblick.



# Generative Adversial Network

Generative Adversial Networks, kurz GANs, sind eine aufstrebende Technologie im Bereich des maschinellen Lernens und der künstlichen Intelligenz. Inspiriert von Ian Goodfellow und seinen Kollegen im Jahr 2014, bieten GANs eine effiziente Möglichkeit, tiefe Repräsentationen von Daten zu erlernen, ohne dass große Mengen an annotierten Trainingsdaten benötigt werden. Dies wird durch die Verwendung von Backpropagation und den Wettbewerb zwischen zwei neuronalen Netzen - dem Generator und dem Diskriminator - erreicht. Daraus ergeben sich zahlreiche neue Ansätze zur Generierung realistischer Inhalte. Die Anwendungen reichen von der Bildgenerierung bis hin zur Superauflösung und Textgenerierung.

## Verfahren

Der Generator und der Diskriminator sind die Hauptkomponenten eines GAN. Die beiden neuronalen Netze werden gleichzeitig trainiert und konkurrieren miteinander, wobei der Generator versucht, den Diskriminator zu täuschen, indem er synthetische Inhalte erzeugt. Um die Glaubwürdigkeit des Generators zu erhöhen, so dass der Diskriminator nicht mehr zwischen den Eingaben unterscheiden kann, wird das gesamte Netz trainiert. Die Netze werden in der Regel als Mehrschichtnetze implementiert, die aus Faltungsschichten und vollständig verbundenen Schichten bestehen.

## Generator

Der Generator dient zur Erzeugung künstlicher Daten wie Bilder und Texte. Der Generator ist nicht mit dem realen Datensatz verbunden und lernt daher nur durch die Interaktion mit dem Diskriminator. Wenn der Diskriminator nur noch 50% der Eingaben richtig vorhersagt, gilt der Generator als optimal.

## Diskriminator

Die Unterscheidung zwischen echten und unechten Eingaben ist Aufgabe des Diskriminators. Der Diskriminator kann sowohl künstliche als auch reale Daten verwenden. Wenn der Diskriminator nicht mehr richtig unterscheiden kann, wird er als konvergierend bezeichnet. Andernfalls wird er als optimal bezeichnet, wenn seine Klassifizierungsgenauigkeit maximiert ist. Im Falle eines optimalen Diskriminators wird das Training des Diskriminators gestoppt und der Generator trainiert alleine weiter, um die Genauigkeit des Diskriminators wieder zu verbessern.

## Training

Durch das Finden von Parametern für beide Netze wird das Training durchgeführt. Ziel ist die Optimierung beider Netze durch Anwendung von Backpropagation zur Verbesserung der Parameter. Das Training wird oft als schwierig und

instabil beschrieben, da es einerseits schwierig erscheint, beide Modelle konvergieren zu lassen. Auf der anderen Seite kann der Generator sehr ähnliche Muster für verschiedene Eingaben erzeugen, und der Diskriminatorverlust kann schnell gegen Null konvergieren, so dass es keinen zuverlässigen Weg für die Gradientenaktualisierung zum Generator gibt. Zur Lösung dieser Probleme wurden verschiedene Ansätze vorgeschlagen, wie z.B. die Verwendung heuristischer Verlustfunktionen. Eine weitere Möglichkeit, die von Sonderby et al. vorgeschlagen wurde, besteht darin, den Datensatz vor der Verwendung zu verrauschen.

## Anwendung

GAN wurde ursprünglich für unüberwachtes maschinelles Lernen entwickelt. Die Architektur liefert jedoch ebenso gute Ergebnisse beim halbüberwachten Lernen und beim Reinforcement Learning. Aus diesem Grund wird sie in einer Vielzahl von Bereichen wie dem Gesundheitswesen, dem Maschinenbau und dem Bankwesen eingesetzt. Beispielsweise wird GAN in der Medizin zur Erkennung und Behandlung chronischer Krankheiten eingesetzt. Aber auch die Identifikation von 3D-Objekten und die Generierung von realen Bildern und Texten ist durch den Einsatz von GANs möglich.

## Limitationen

Die Tatsache, dass ein Generative Adversarial Network in der Lage ist, Inhalte zu erzeugen, die nahezu identisch mit realen Inhalten sind, kann in der realen Welt zu Problemen führen, insbesondere bei der menschlichen Bildsynthese. Diese Bilder können von Betrügern verwendet werden, um falsche Profile in sozialen Medien zu erstellen. Auch dies kann durch den Einsatz von GANs verhindert werden, indem einzigartige und pragmatische Bilder von nicht existierenden Personen erzeugt werden.



## Anhang - Code

Hier sehen Sie den gesamten Quellcode!





# B

## Anhang - Dokumentationen

Hier sehen Sie die gesamten Dokumentationen zu den erstellten Programmen.



# Literaturverzeichnis

- [Haf18] HAFNER, E.G.: *Lineare Algebra für Dummies*. Wiley, 2018 (Für Dummies). <https://books.google.de/books?id=XfWADwAAQBAJ>. – ISBN 9783527819430



# Abbildungsverzeichnis

- 2.1. Schematische Darstellung der U-Net-Architektur. Die Architektur besteht aus einem Encoder-Teil (links), einem Decoder-Teil (rechts) und Skip-Verbindungen zwischen korrespondierenden Schichten. . . 5



# Tabellenverzeichnis





# Code-Auszugs-Verzeichnis



# Glossar

- **C++:**  
Hardwarenahe, objektorientierte Programmiersprache.
- **HTML:**  
Hypertext Markup Language - textbasierte Auszeichnungssprache zur Strukturierung elektronischer Dokumente.
- **HTTP:**  
Hypertext Transfer Protocol - Protokoll zur Übertragung von Daten auf der Anwendungssicht über ein Rechnernetz.
- **iARS:**  
innovative Audio Response System - System mit zwei Applikationen (iARS-master-App; iARS-student-App), dass sich zum Einsetzen von e-TR-ainer-Inhalten in Vorlesungen eignet.
- **ISO:**  
Internationale Vereinigung von Normungsorganisationen.
- **JavaScript:**  
Skriptsprache zu Auswertung von Benutzerinteraktionen.
- **Konstruktor:**  
Beim Erzeugen einer Objektinstanz aufgerufene Methode zum Initialisieren von Eigenschaften.
- **MySQL:**  
Relationales Datenbankverwaltungssystem.
- **OLAT:**  
Online Learning and Training - Lernplattform für verschiedene Formen von webbasiertem Lernen.
- **OOP:**  
Objektorientierte Programmierung - Programmierparadigma, nach dem sich die Architektur eine Software an realen Objekten orientiert.
- **Open Source:**  
Software, die öffentlich von Dritten eingesehen, geändert und genutzt werden kann.
- **PHP:**  
Skriptsprache zur Erstellung von Webanwendungen.
- **Python:**  
Skript- und Programmiersprache, die unter Anderem objektorientiertes Programmieren ermöglicht.

- **Shell:**  
Shell oder auch Unix-Shell - traditionelle Benutzerschnittstelle von Unix-Betriebssystemen
- **Spyder:**  
Entwicklungsumgebung für wissenschaftliche Programmierung in der Programmiersprache Python.
- **SymPy:**  
Python-Bibliothek für symbolische Mathematik.