

# Design experiments with Artificial Intelligence



**Designexperiments with  
Artificial Intelligence**

Bachelorarbeit  
Theoretischer Teil

Finn Jacobsen  
Wintersemester 22 / 23  
Fachhochschule Aachen  
Fachbereich Gestaltung

betreut von  
Prof. Eva Vitting

# Inhaltsverzeichnis

---

## 1

---

<b>Einleitung</b>	7
-------------------	---

---

Thematische Einleitung	8
------------------------	---

Pix2Pix	10
---------	----

## 2

---

<b>Experimente</b>	13
--------------------	----

---

2.1 Pflanzen	14
--------------	----

2.2 Artificial BeReal	22
-----------------------	----

2.3 Finngenerator	30
-------------------	----

2.4 UnBlur	38
------------	----

2.5 Pix2Food	46
--------------	----

2.6 ground2sky	54
----------------	----

2.7 Next Frame	62
----------------	----

2.8 square2face	74
-----------------	----

2.9 helvetica2didot	82
---------------------	----

2.10 regular2bold	92
-------------------	----

2.11 Zoom	102
-----------	-----

2.12 Bitmoji Face CycleGAN	114
----------------------------	-----

2.13 rotes und weißes Rechteck	120
--------------------------------	-----

2.14 shape2gradient	128
---------------------	-----

2.15 line2line	136
----------------	-----

2.16 shape2status	144
-------------------	-----

2.17 Wiederholende Form	152
-------------------------	-----

---

<b>3</b>	<b>Installation</b>	161
	3.1 Konzept	162
	3.2 Technik	168
	3.3 Tests	170
	3.4 Reflexion	172
<b>4</b>	<b>Doku</b>	175
	4.1 Konzept	176
	4.2 Programmiertes Layout	178
<b>5</b>	<b>Fazit</b>	181
<b>6</b>	<b>Anhang</b>	187
	6.1 Literaturverzeichnis	188
	6.2 Online Ressourcen	190
	6.3 Danksagung	196
	6.4 Erklärung zur Eigenständigkeit	200



# Einleitung

---

Im folgenden Kapitel gehe ich auf die Themenstellung und den Umfang der Bachelorarbeit ein.

# Thematische Einleitung

---

## Künstliche Intelligenz und Design

In den letzten Jahren und Monaten hat sich der Bereich von KI (Künstlicher Intelligenz) im Bereich Design sehr schnell entwickelt.

Die Entwicklung von Text-zu-Bild-Generatoren wie Dall-E und Midjourney, welche frei im Internet verfügbar sind, erregt derzeit sehr viel Aufsehen. Die Frage, ob solche Algorithmen Künstler\*innen, Illustrator\*innen oder Gestalter\*innen ersetzen können hat sich aufgetan.

Die Angst, dass die KI Gestalter\*innen und Künstler\*innen die Arbeitsplätze wegnimmt halte ich für nicht angebracht. Zum einen braucht es immer jemanden der die Maschine steuert. Dadurch entsteht eine neue Kunstform, das „Prompting“ wobei es um das Finden des richtigen Prompts geht, um ein gewünschtes Ergebnis zu erzielen. Zum anderen sind solche Werkzeuge nicht in der Lage zu reflektieren. Sie können unentwegt produzieren, aber nicht bewerten, wie gut ihnen die Lösung der Aufgabe auf einer inhaltlichen Ebene gelungen ist. Zur Bewertung der Ergebnisse ist immer noch ein Mensch nötig. Zusätzlich sind die Ergebnisse solcher Generatoren sehr kitschig, da sie oft keine neuen visuellen Ergebnisse zeigen, sondern den Durchschnitt der Trainingsdaten. Ein ganz anderes Problem ist die Frage nach dem Urheber, da die Trainingsdaten auch von Menschen erstellt wurden.

In meiner Bachelorarbeit möchte ich mich jedoch nicht mit den Problemen und Chancen von Stable-Diffusion und Midjourney befassen, sondern experimentell untersuchen, wie man noch auf anderem Wege KI nutzen kann, um sich gestalterisch und künstlerisch auszudrücken.

---

## Einordnung dieser Bachelorarbeit

In diesem Bachelorsemester habe ich mich sehr experimentel mit dem Thema KI und Design auseinander gesetzt. Das kennenlernen und den Umgang mit der neuen Technik steht bei mir im Vordergrund.

Ich habe mich dem Thema aus der Sicht eines Creative Technologist / Creative Coders angenommen. Die Technik auf kreative Weise zu nutzen ist der Schwerpunkt in meiner Bachelorarbeit. Dazu gehört insbesondere die Datenerstellung sowie das Sammeln, wofür ich in einigen Fällen Creative Coding Software wie z.B. Processing genutzt habe.

Die Experimente die ich vor allem mit dem Pix2Pix Modell gemacht habe, sind in dieser Dokumentation chronologisch und systematisch abgebildet. Ich gehe auf die Erstellung der Daten ein, auf den Trainingsprozess und zeige einige Ergebnisse und Reflektiere das Experiment in Bezug auf die Anwendung im Kommunikationsdesign.

Der Letzte Teil dieser Dokumentation befasst sich mit der Installation, welche ein Experiment weiter ausführt. Dies Installation wird bei der Diploma 2023 (03.02. – 04.02.2023) ausgestellt.

## Digitales Selbstbild

Die Aufgeführten Experimente befassen sich im weitesten Sinne mit den Themen Online Identität und digitales Selbstbild. Dabei wurde entweder der Datensatz und die Datensammlung an dem Thema orientiert oder durch den visuellen Output eine Stellung bezogen.

In meinem ersten Schritt habe ich viel mit Gesichtern gearbeitet, um das Thema in meinen Experimenten unterzubringen. Jedoch habe ich um den Output abwechslungsreicher zu gestalten mich auch von dieser Optik entfernt.

# Pix2Pix

---

## Image-to-Image Translation

Unter dem technischen Begriff „Image-to-Image Translation“ versteht man, dass aus einem Inputbild ein passendes Outputbild generiert wird.

Ein klassisches Anwendungsgebiet ist der so genannte „Style Transfer“. Dabei wird ein Bild (z.B. eine Fotografie) als Inputbild durch ein neuronales Netz so verändert dass es am Ende einen anderen Stil hat (z.B. eine Zeichnung). Es gibt „Paired“ und „Unpaired“ Image-to-Image Translation.

Bei Paired Image-to-Image Translation (z.B. pix2pix) wird das künstliche neuronale Netz auf Bildpaare trainiert. Dabei ist ein Bild dem anderen fest zu geordnet.

Bei Unpaired Image-to-Image Translation gibt man dem künstlichen neuronalen Netz jeweils einen Datensatz von der einen Bildsorte und einen Datensatz der anderen Bildsorte. Hierbei müssen die Daten nicht paarweise gegeben werden.

Unpaired Image-to-Image Translation hat den Vorteil, dass man die Trainingsdaten vorher nicht besonders aufbereiten muss. Auch ist es bei Problemen von Vorteil, bei denen man nicht die Möglichkeit hat die Bilder in passenden Paaren zu organisieren (z.B. Pferd zu Zebra). Es hat aber den Nachteil, dass man größere Datensätze benötigt, weshalb man mehr Rechenleistung und Zeit braucht, um das Modell zu trainieren.

Bei Paired Image-to-Image Translation bedarf es deutlich kleiner Datensätze, um das Modell erfolgreich zu trainieren. Durch die Zusammenstellung der Paare hat man einen größeren Einfluss darauf, was das künstliche neuronale Netz lernen soll.

## Pix2Pix

Pix2Pix ist ein Paired Image-to-Image Translation Modell. Es nutzt Conditional Adversarial Nets als Netzwerkstruktur. 2017 wurde es von Phillip Isola, Jun-Yan Zhu, Tinghui Zhou und Alexei A. Efros an der University of California, Berkeley veröffentlicht. Der Code für verschiedene Architekturen ist öffentlich zugänglich. Für meine Experimente habe ich den Code für PyTorch von Jun-Yan Zhu und Taesung Park verwendet. (Quelle: <https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix>)

### Grobe Funktionsweise

Pix2Pix wird auf Bildpaare trainiert. Es gibt das Bild „A“ und „B“; sie werden in einem Bild nebeneinander in die Maschine gegeben.

Der Trainingsprozess sieht wie folgt aus. Zufällig wird eine der zwei Verfahren ausgewählt:

- ↳ Die Maschine generiert mit dem Generator ein Bild (fake B). Der Discriminator versucht dieses als „real“ oder „fake“ zu labeln.
- ↳ Der Discriminator labelt das passende Paar (A und B) als „real“ oder „fake“.

Durch das ständige Wiederholen spielen sich die beiden Netzwerke (Generator und Discriminator) aus. Der Generator generiert immer bessere Bilder und der Discriminator wird immer besser im labeln der Bilder.

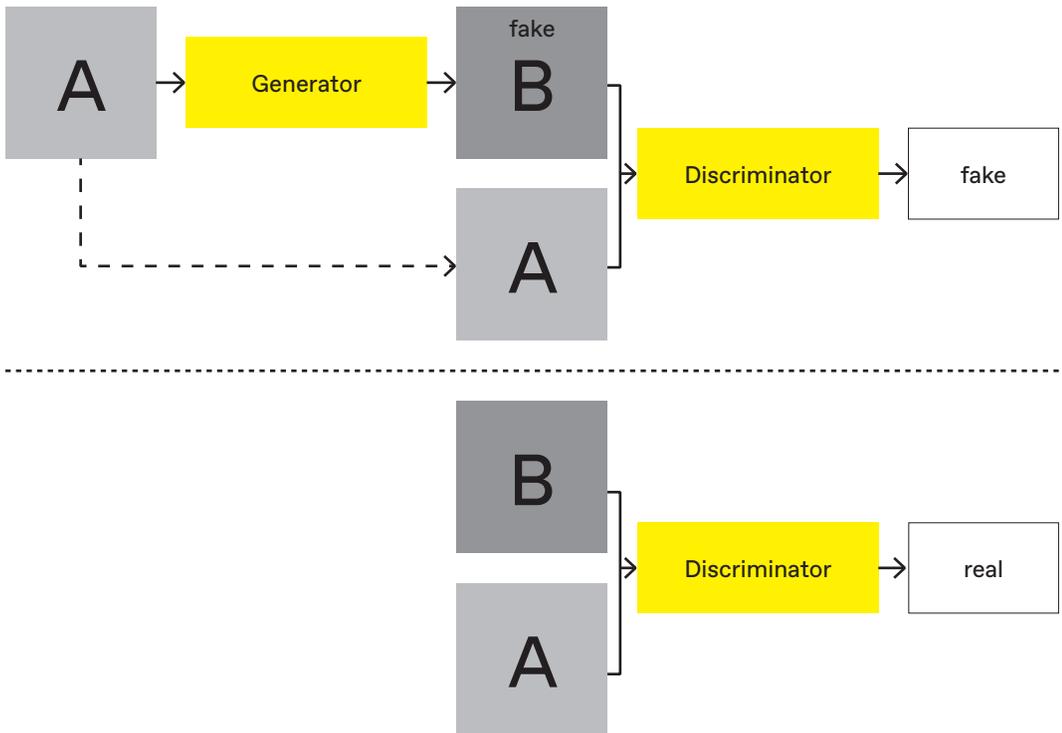
In der Abbildung rechts ist der Prozess systematisch dargestellt.

In der Testphase kann man nun Bilder eingeben, welche die Maschine nicht kennt und beobachten wie gut sie die Übersetzung schafft.

Trainingsdaten



Training Phase



2

# Experimente

---

Diese Kapitel stellt alle in diesem Semester entworfenen Experimente mit dem Pix2Pix Modell dar.

Zu Beginn ist das Experiment kurz beschrieben. Auf der folgenden Seite wird das Training des Modells abgebildet. Zuerst sieht man einen Ausschnitt aus den Trainingsdaten und es wird beschrieben, wie diese Daten aufgebaut sind.

Der Fortschritt des Trainingsprozess wird mittels der Checkpoints gezeigt. Diese generiert das Modell während des Trainingsprozesses. In der Tabelle bedeutet der Begriff „Epoch“ die Wiederholungen, die das Modell durchläuft.

Die darauf folgende Seite zeigt den Aufbau der Testdaten und die Ergebnisse, die durch das Experiment generiert wurden.

Auf der Letzten Seite jedes Experiments wird das Experiment in Bezug auf Kommunikationsdesign reflektiert.

PLANT

## Idee

Die Idee dieses Experiments ist es, das Modell auf Blätter zu trainieren und im zweiten Schritt dann mit Typografie das System zu testen. Dabei ist die Vorstellung, dass die Typografie durch das Modell eine Blattstruktur bekommt.

## Erstes Modell

Das auf den folgenden Seiten beschriebene Experiment ist mein erster Versuch, ein Pix-2Pix Modell zu trainieren. Durch die Hilfe von einem studierten Data Science Experten gelang es mir, dieses Modell auf meinem eigenen PC zu trainieren.

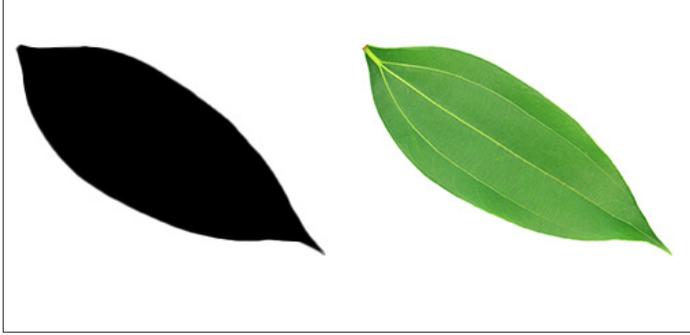
## Datensatz

Ich habe mich bei meinem erstes Modell bewusst für einen kleinen Datensatz von 33 Bildern entschieden. Damit wollte ich erforschen, wie der Algorithmus mit einer kleinen Datenmenge umgeht. Unter der Fragestellung, ob der Algorithmus überhaupt etwas bei so wenigen Daten lernt, führte ich das Experiment durch.

Außerdem hat die kleine Datenmenge den Vorteil, dass der Trainingsprozess sehr kurz ist. Vor allem zu Beginn meiner Arbeit war mir wichtig, dass ich den gesamten Trainingsprozess beobachten kann und bei eventuellen technischen Fehlern intervenieren kann. Mit diesem kleinen Datensatz konnte ich auch testen, ob meine Rechenleistung überhaupt für das Training eines Modells ausreicht.

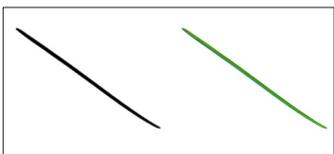
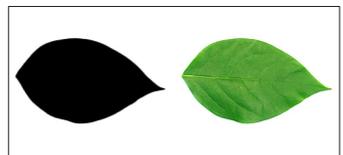
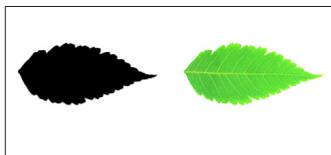
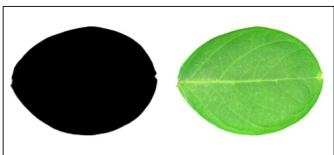
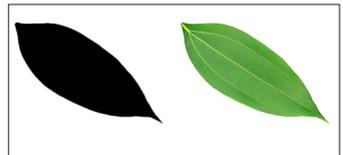
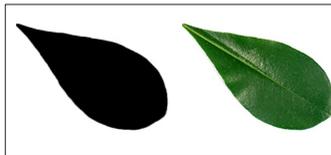
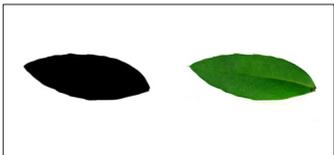
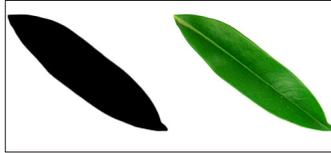
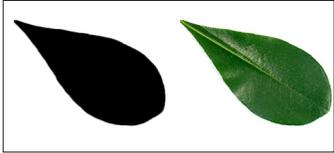
# Pflanzen pix2pix

Trainingsdaten (33 Bilder)



**A:**  
Zeigt die Silhouette des Blattes  
schwarz auf weiß

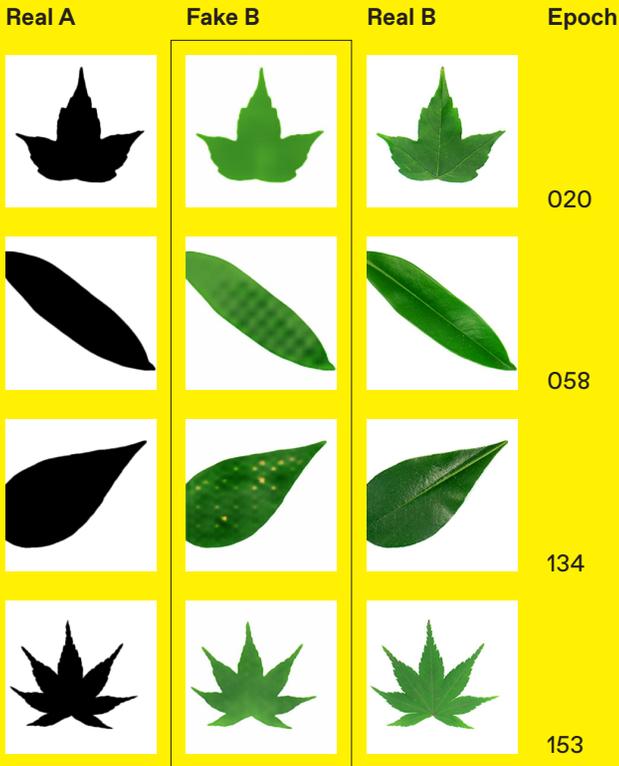
**B:**  
Zeigt das Originalbild aus der  
Datenbank



...

Trainingsprozess des Pix2Pix Modells

Epochs: 200

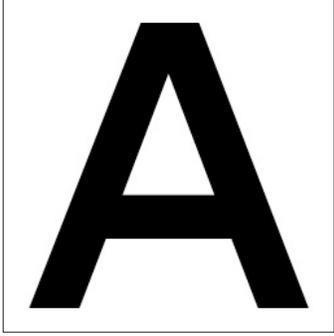


Epoch: 191



# Pflanzen pix2pix

Kreation der Testdaten



A:  
Schwarz auf weiße Typo.



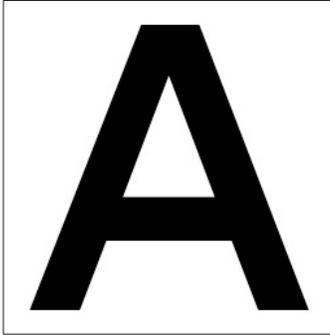
Pflanzen  
Pix2Pix Modell

Training auf  
vorheriger Seite  
←

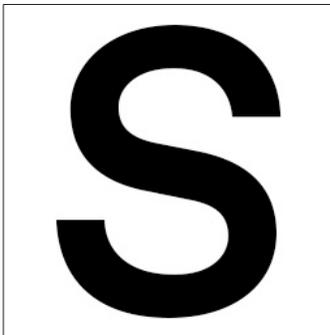
---

Ergebnisse

Input

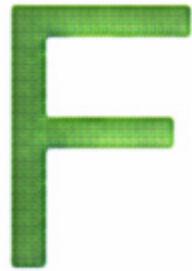


Output



# Pflanzen pix2pix

---

A large, green, textured letter 'L' with a slight gradient, centered in a white square box.A large, green, textured letter 'E' with a slight gradient, centered in a white square box.A large, green, textured letter 'A' with a slight gradient, centered in a white square box.A large, green, textured letter 'F' with a slight gradient, centered in a white square box.

---

## Reflexion

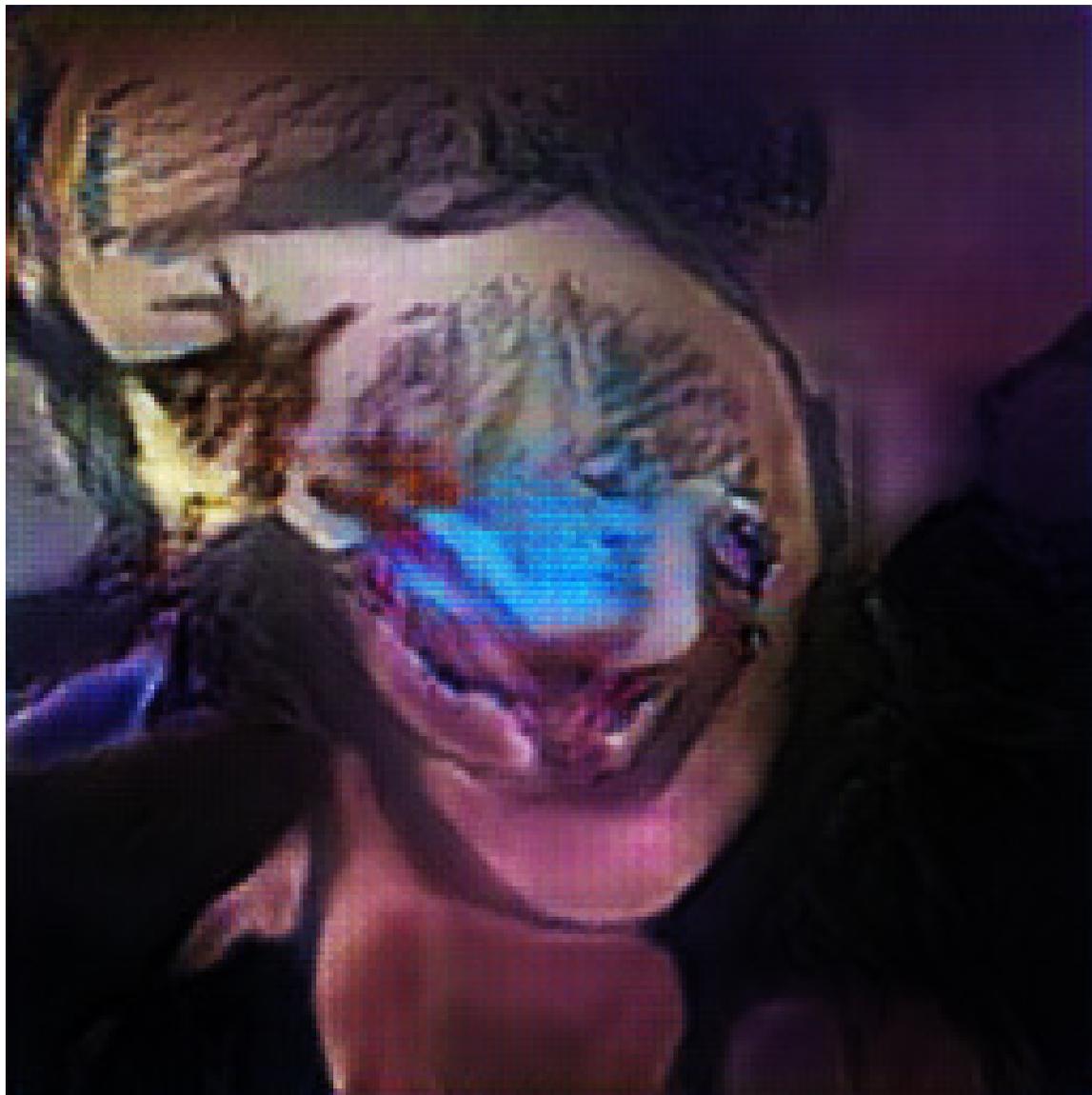
### Erlernte Muster

Dieses erste Experiment zeigt, wie die Maschine lernt und, dass der Algorithmus funktioniert. Man sieht relativ deutlich welche Muster die Maschine erkannt hat. An Stellen wo das Inputbild schwarz ist, wird das Outputbild Grün und an Stellen, wo das Inputbild weiß ist, bleibt auch das Outputbild weiß.

Eine echte Blattstruktur ist jedoch nicht zu erkennen. Der Algorithmus variiert jedoch den Grünton in der Form, wodurch eine andere Ästhetik generiert wird. Auch ist der Rand der Flächen etwas dunkler als der Mittelteil.

### Anwendung

Zunächst fungiert dieses Modell als technischer Test, man könnte es jedoch auch für Designaufgaben nutzen. Würde man die Daten noch erweitern und den Trainingsprozess optimieren, könnte man womöglich dieses Modell als Tool zum texturieren von Grafiken nutzen. Zudem wäre es nützlich, für das Erstellen von Assets z.B für ein Computerspiel. Man müsste nur verschiedene Blattformen generieren und das Modell würde sie dann je nach Blattform richtig texturieren.



## Artificial BeReal

### Idee

Da Pix2Pix mit Bildpaaren trainiert wird, hielt ich nach diesen in unserer digitalen Welt Ausschau. Gerade in dieser Zeit, gab es einen Hype um die Socialmedia-App „BeReal“. Diese App verspricht durch ihr Konzept eine große Authentizität. Man wird einmal am Tag, zu einem zufälligen Zeitpunkt, dazu aufgefordert ein Bild zu machen. Dafür hat man nach Aufforderung 2 Minuten Zeit. Dabei wird sowohl von der Vorderkamera als auch von der Rückkamera des Smartphones ein Bild gemacht. Mich interessierte was passiert, wenn ich dem Modell diese Bilder als paar gebe. Kann man dadurch ein künstliches BeReal erschaffen?

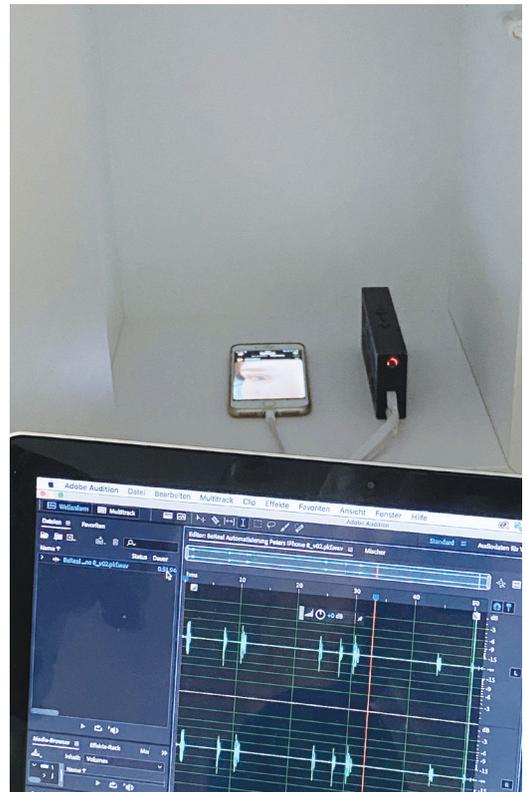
### Sammeln der Daten

Für das Sammeln der Daten nutzte ich den Discovery-Bereich der App. Hier sieht man nicht nur die Bilder die von Freund\*innen gepostet werden, sondern auch aller anderen, die sich dazu entschieden haben entschieden haben, ihre Bilder öffentlich und zugänglich öffentlich zu posten. Um nicht stundenlang damit beschäftigt zu sein die Bilder zu screenshoten, baute ich mir eine prototypische Maschine, die durch den Discovery-Bereich scrollt und Screenshots erzeugt. Dazu verwendete ich, die in iPhones integrierte Sprachsteuerung, selbst programmierte Shortcuts und einen Lautsprecher, mit dem ich meine aufgenommenen Sprachbefehle hintereinander abspielte.

### Datenschutz

Durch das screenshoten der Bilder in der App, bekommt der Nutzer / die Nutzerin eine Nachricht, dass ihr Bild gescreenshotet wurde. Aufgrund dieser Nachricht, wird einigenen Nutzer\*innen klar, dass ihre Daten für Jeden öffentlich zugänglich sind.

Um in dieser Bachelorarbeit den Datenschutz zu wahren, habe ich mich entschieden die echten Gesichter auf den folgenden Seiten unkenntlich zu machen.



# Artificial BeReal

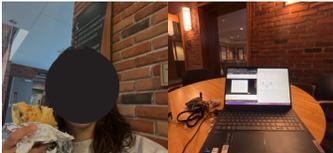
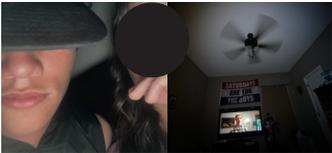
Trainingsdaten (1629 Bilder)



**A:**  
Foto mit der Frontkamera

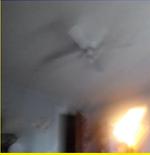
**B:**  
Foto mit der Rückkamera

Die Daten wurden Auto-  
matisiert gesammelt und  
stammen aus dem  
Discovery-Bereich



Trainingsprozess des Pix2Pix Modells

Epochs: 200

Real A	Fake B	Real B	Epoch
			003
			049
			088
			163

Achtung die Bezeichnung A und B sind hier verkehrt-herum.  
Das Modell wurde von B zu A trainiert. Die Benennung ist aufgrund von technischer Konventionen umgekehrt. Es wird also die Frontseite des BeReals auf Grundlage der Rückseite generiert.

Epoch: 197

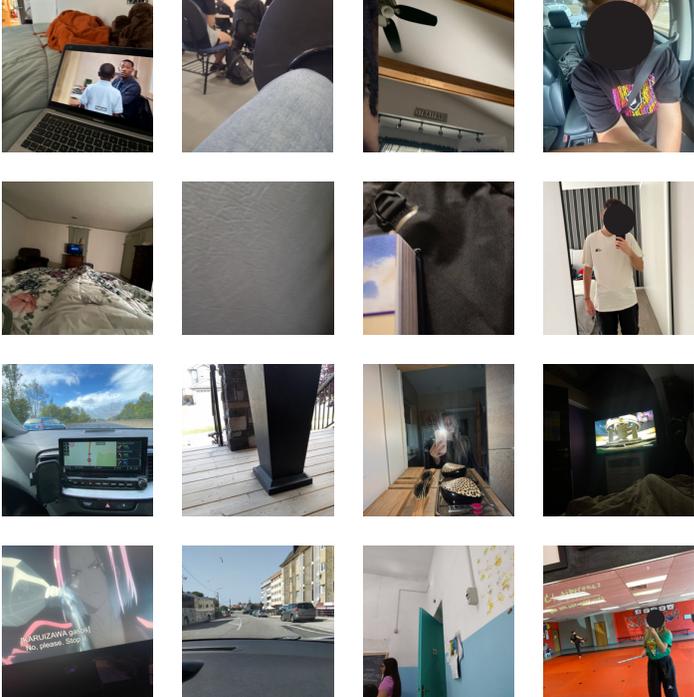


# Artificial BeReal

Kreation der Testdaten



**Input:**  
Foto von der Rückkamera,  
ebenfalls aus der App.



Artificial BeReal  
Pix2Pix Modell

Training auf  
vorheriger Seite  
←

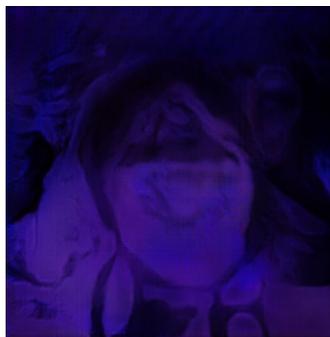
---

Ergebnisse

Input

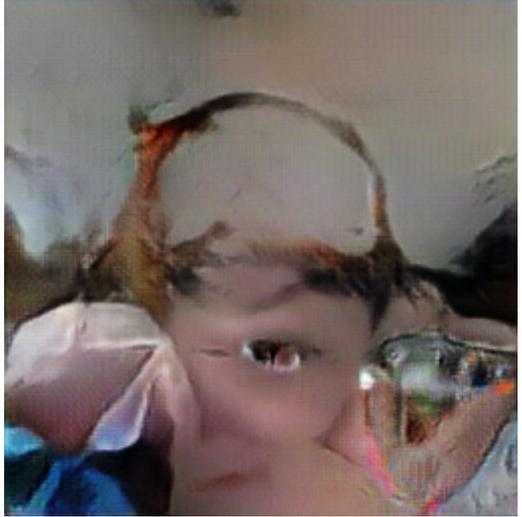
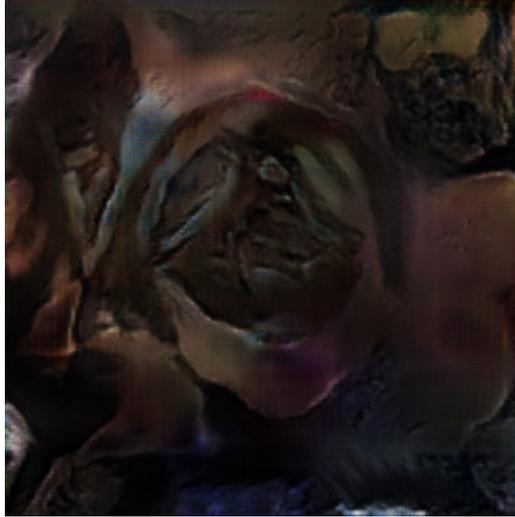


Output



# Artificial BeReal

---



**Erkenntnisse über das Modell**

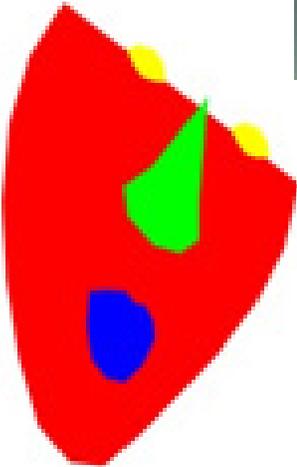
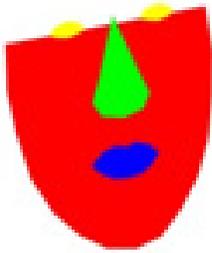
Bei diesem Experiment habe ich viel über die Funktionsweise des Modells gelernt. Die größere Datenmenge von ca. 1600 Bildern hat dazu geführt, dass der Trainingsprozess sehr lange gedauert hat (ca. 8 Stunden). Es war nicht zu erwarten, dass das künstliche neuronale Netz dazu in der Lage ist fotorealistische Bilder zu generieren. Jedoch ist es erstaunlich, wie das Modell die Farbgebung des Inputbildes übernimmt.

Außerdem ist das Modell in der Lage eine sehr abstrahierte Form eines Gesichtes zu zeichnen. Die meisten Outputbilder zeigen eine Kopfform in der Mitte des Bildes. Dies ergibt Sinn, da bei den meisten Bildern mit der Frontkamera ein Gesicht zu sehen ist. Die Maschine scheint den Durchschnitt aller Gesichter in der Mitte zu berechnen. Auf einigen Bildern kann man schon fast ein Auge oder einen Mund erkennen. Aufgrund ihrer Verzerrung sind die Bilder jedoch sehr unkenntlich.

**Anwendung**

Die Bilder wirken relativ verstörend, da die gesichtsähnlichen Formen sehr verzerrt sind. Es ist eher ein Gesichtsmatsch, bei dem ab und zu ein Auge oder ein Mund zum Vorschein kommt. Sie könnten sich aber als Inspiration für ein Monster aus einem Horror- und Sciene Fiction Film eignen. Die leicht erkennbare menschliche Natur, welche die Bilder zeichnen, erinnert an ein Alien oder an eine unentdeckte Lebensform.

Außerdem könnte das Modell auch genutzt werden, um ein Musikvideo mit passendem Thema zu entwerfen. Durch die Bildsprache wäre dafür hauptsächlich experimentelle Musik geeignet.



---

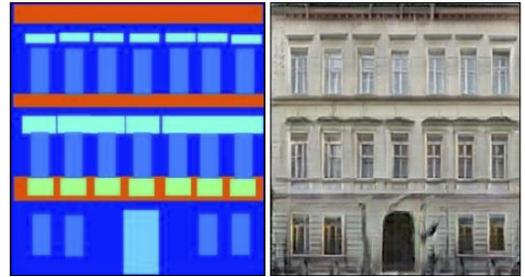
## Fingergenerator

### Idee

Bei diesem Experiment war die Idee, mein äußeres Erscheinungsbild in einem Modell zu speichern. Das Pix2Pix Modell wird vor allem dafür genutzt, um eine „Labelmap“ in ein fotografisches Bild umzuwandeln. Eine Labelmap ist ein Bild, in dem Bereiche in einer bestimmten Farbe semantisch zugeordnet sind. Dieses Verfahren wollte ich in diesem Experiment ausprobieren.

### Datenerstellung

Zur Erstellung der Facetracking Daten nutzte ich eine Library für P5.js, welche die verschiedenen Gesichtsmerkmale tracken kann. Durch diese Technik konnte ich meine Webcam auslesen und gleich eine passende Labelmap generieren. Ich kleidete mich extra auffällig, sodass ich auf dem Hintergrund klar zu erkennen bin und das Modell einen starken Kontrast zwischen dem verschiedenen Bereichen hat. Zudem legte ich meine Brille ab, damit die Facetracking Software mich besser erkennen konnte.



# Finngenerator

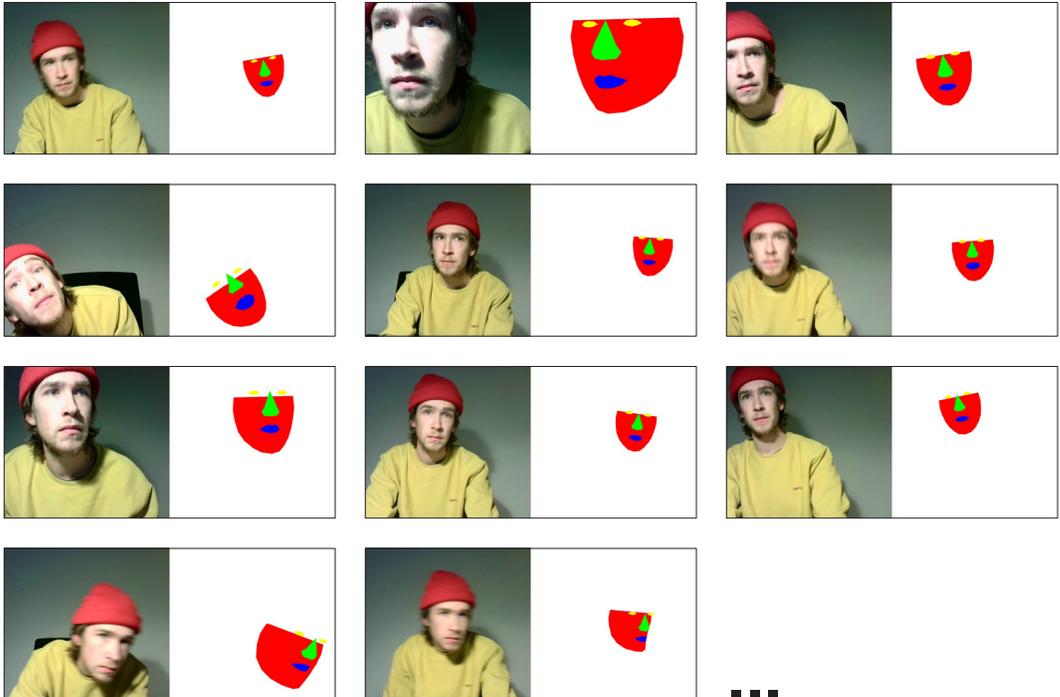
Trainingsdaten (1009 Bilder)



**A:**  
Zeigt das Webcam Bild

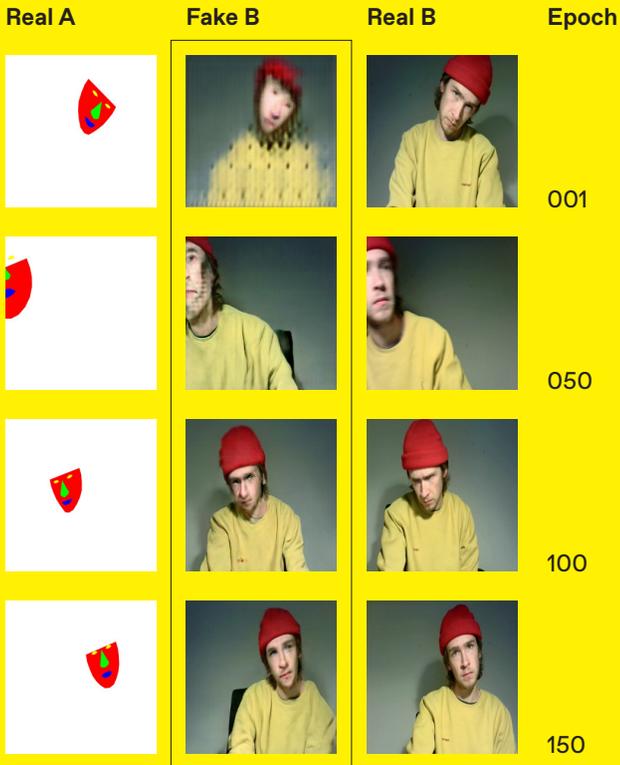
**B:**  
Zeigt die Labelmap die ver-  
schieden Gesichtsmerkma-  
len farblich markiert.

Die Trainingsdaten wurden mithilfe einer Facetracking Library für P5.js generiert.



Trainingsprozess des Pix2Pix Modells

Epochs: 200



Epoch: 200

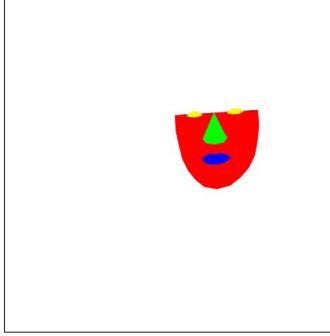


# Finngenerator

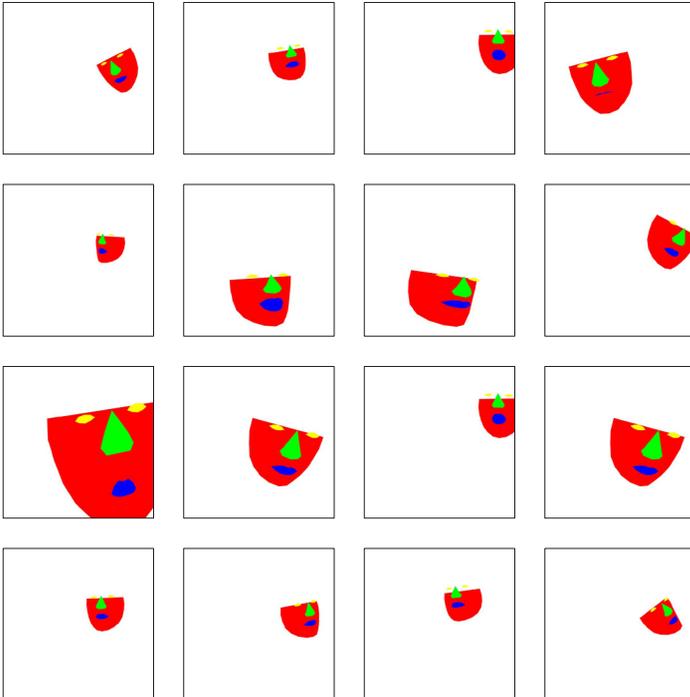
Kreation der Testdaten



**Input:**  
Webcam Aufnahme einer  
anderen Person.



**Verarbeiteter Input:**  
Generierte Labelmap zeigt die  
verschiedenen Gesichtsmerk-  
malen eingefärbt.

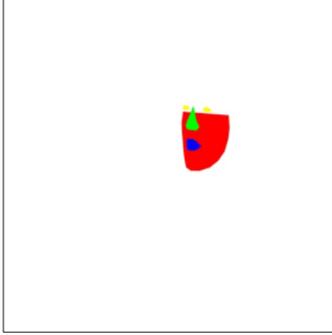


Finngenerator  
Pix2Pix Modell

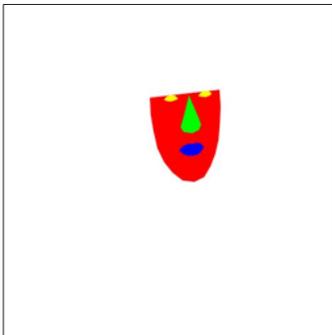
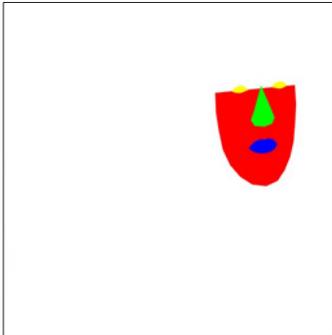
Training auf  
vorheriger Seite  
←

## Ergebnisse

Input



Output



Video zum  
Ergebnis →



# Finngenerator

---



**Erkenntnisse**

Dieses Experiment konnte erst durch eine Kombination aus zwei Machine-Learning Tools verwirklicht werden. Zum einen das Erstellen der Labelmap mit einem Facetracking Algorithmus und durch das Generieren der Bilder mit dem Pix2Pix Modell. Durch die Abstraktion in eine Labelmap, kann mich nun rein theoretisch jeder mit seinem Gesicht steuern. Man sieht jedoch sofort, dass es sich bei dem generierten Bild um einen Fake handelt. Dies ist auch von mir beabsichtigt, da ich die Fehler, die das System macht interessanter finde, als wenn alles richtig funktioniert. Außerdem finde ich spannend, dass sich mein Gesicht leicht verändert, wenn jemand anderes gescannt wird.

**Anwendung**

Eine Anwendung für dieses Modell könnte wiederum ein Musikvideo sein, zu welchem die erzielte Optik passt. Der leichte Glitcheffekt könnte auch als spannendes Element auf einem Plakat genutzt werden. Zudem auch im künstlerisch, kulturellen Kontext für einen Experimentalfilm.

Denkt man diese Technik weiter, gibt es jedoch auch beunruhigende Zukunftsszenarien. Sollte es irgendwann möglich sein, die Bilder fotorealistisch zu bekommen, könnte die Technik auch für das Erstellen von falschen Beweisvideos genutzt werden.

Heutzutage gibt es auch schon Technik die dazu in der Lage ist. Sogenannte Deepfakes gibt es schon seit einigen Jahren. Bei dieser Technik wird ebenfalls Deeplearning genutzt, um Gesichter fotorealistisch auszutauschen. Unter anderem haben sich schon Künstler dieser Methode angenommen. In der Pornografie wird diese Technik jedoch auch verwendet, um Material zu generieren. Dies wird meistens ohne die Zustimmung der abgebildeten Person gemacht, wodurch das Erstellen von solchem Material sehr verwerflich ist.

# UnBlur



---

## Unblur

### Idee

Bei dem Experiment UnBlur habe ich das Pix-2Pix Modell von unscharfen auf scharfe Bilder trainiert. Es ist dem Projekt Transhancement von Mario Klingemann<sup>2</sup> nachempfunden, bei dem ein ähnliches Modell trainiert wurde.

Ich wollte untersuchen was passiert, wenn ich das Modell nur auf Bilder von Gesichtern trainiere und als Testdaten anschließend Bilder eingabe, die kein Gesicht zeigen. Generiert das Modell daraufhin ein Gesicht in die Bilder?

### Datensatz

Die Bilddaten der Gesichter sind von der Bilddatenbank Unsplash. Sie bieten eine große Auswahl an hochauflösender Fotos, die frei für kommerzielle und nicht kommerzielle Zwecke genutzt werden dürfen. (Quelle: <https://unsplash.com/de>)

Die Unschärfe der 2150 Bilder wurde automatisiert mit Photoshop erstellt.

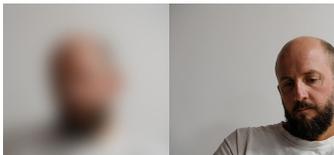
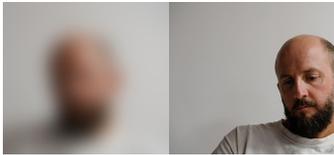
# Unblur

Trainingsdaten (2151 Bilder)



**A:**  
Unschärfes Bild eines Ge-  
sichts

**B:**  
Das passende scharfe Bild des  
Gesichts

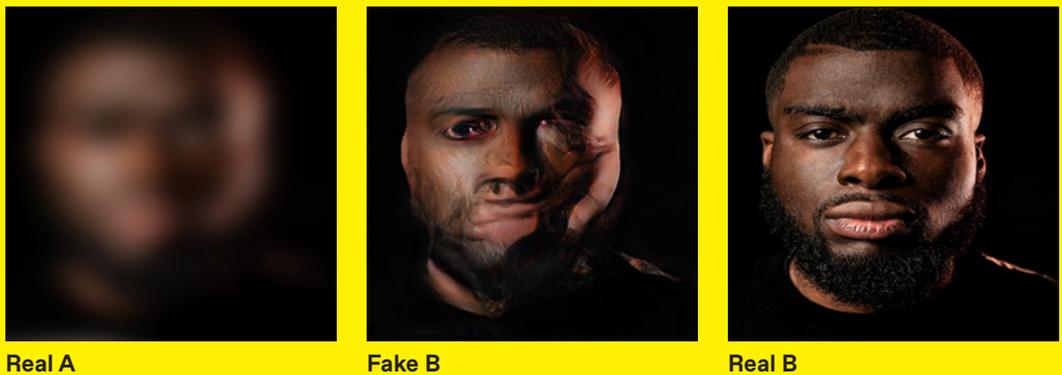


Trainingsprozess des Pix2Pix Modells

Epochs: 200

Real A	Fake B	Real B	Epoch
			001
			049
			085
			113

Epoch: 200



# Unblur

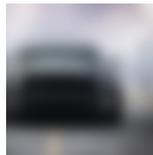
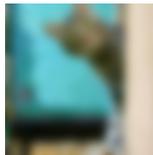
Kreation der Testdaten



**Vor Bearbeitung:** →  
Ein Bild welches kein Gesicht zeigt.



**Nach Bearbeitung:**  
Das bild wird unscharf gemacht.



Unblur  
Pix2Pix Modell

Training auf  
vorheriger Seite  
←

---

Ergebnisse

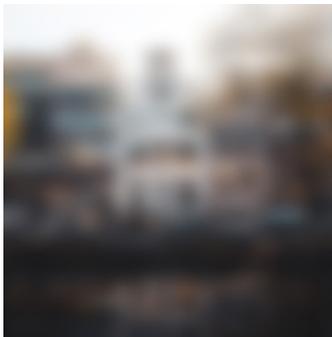
Real A



Fake B



Real B



# Unblur

---



Video zum  
← Ergebnis

---

## Reflexion

### Optik

Durch dieses Modell wird eine interessante Optik erzielt. Durch den Prozess des Unschärfen geht einiges an Bildinformation verloren, welche die Maschine dann wieder probiert zu erzeugen. Dadurch entsteht eine Textur auf den Bildern, die ein bisschen an gemalte Bilder erinnert.

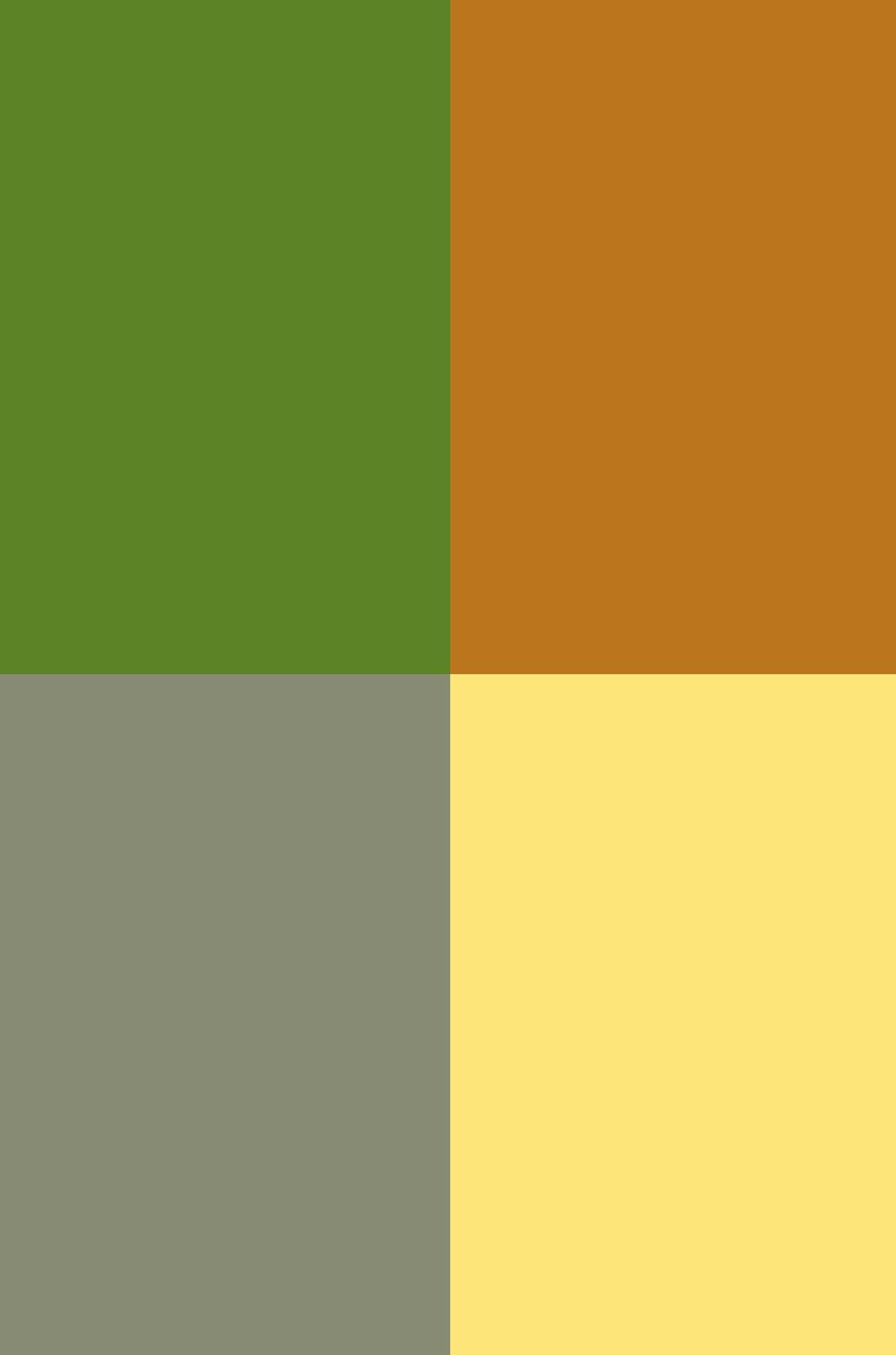
Ich hatte mir vor dem Testen die Frage gestellt, ob das Modell auf Grundlage der Trainingsdaten auch in Bildern, die kein Gesicht zeigen, ein Gesicht generiert. Das kann ich mittlerweile verneinen. Es ist höchstens ab und zu sehr schemenhaft ein Auge oder Mund zu sehen. Es wird aber keine gesamte Gesichtsform generiert.

Im weiteren habe ich probiert, was das Modell mit Typografie erstellt und habe zwei verschiedene Schriften getestet. Eine Groteskschrift und eine Antiqua. Auch hier bildet sich wieder die fließende Struktur, die an einen Pinselstrich erinnert.

### Anwendung

Diese Modell eignet sich in dieser Form nicht zum Schärfen von Bildern. Es verändert die Optik des Bildes doch sehr stark, und macht das Bild vielleicht auf den ersten Blick schärfer. Inhaltlich wird zudem viel vom Algorithmus zusätzlich generiert.

Ich sehe die Anwendung eher im künstlerischen Bereich, um eine bestimmte Optik zu erstellen. Außerdem kann das Modell unter dem Gesichtspunkt Corporate Design genutzt werden. In der Kombination mit Schrift entsteht eine andersartige Bildsprache.



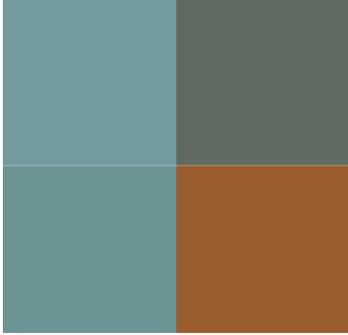
### Idee

Bei diesem Experiment wollte ich testen, was das Modell mit sehr groben Daten anfängt. Dazu habe ich in meinem Bekanntenkreis Fotos von Essen gesammelt. Mit einem einfachen Algorithmus habe ich dann aus den Bildern an bestimmten Positionen den Pixelwert berechnet und ein  $2 \times 2$  Raster mit diesen Werten gefüllt.

Die Idee war es, dass ich am Ende ein zufälliges Raster generiere und sich anhand dessen ein Essen generiert.

Dieses Experiment habe ich noch mit einem Raster von  $3 \times 3$  und  $5 \times 5$  wiederholt.

Trainingsdaten (164 Bilder)



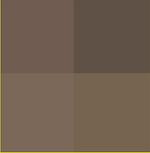
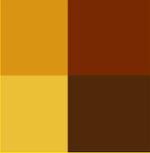
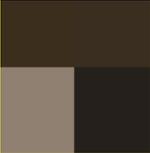
**A:**  
Grobe Rasterung der Bilder  
mit Farben aus dem Bild.

**B:**  
Das Originalbild, auf dessen  
Grundlage Bild A generiert  
wurde.

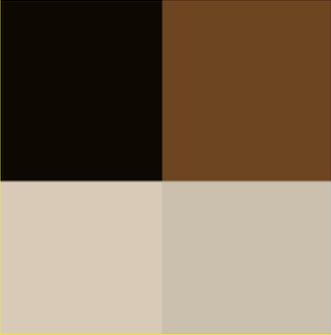


Trainingsprozess des Pix2Pix Modells

Epochs: 200

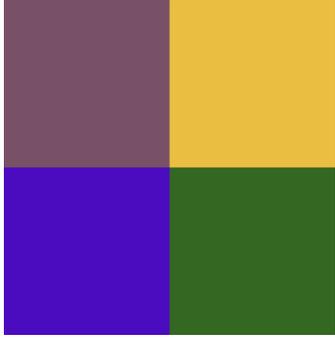
Real A	Fake B	Real B	Epoch
			004
			044
			110
			165

Epoch: 197

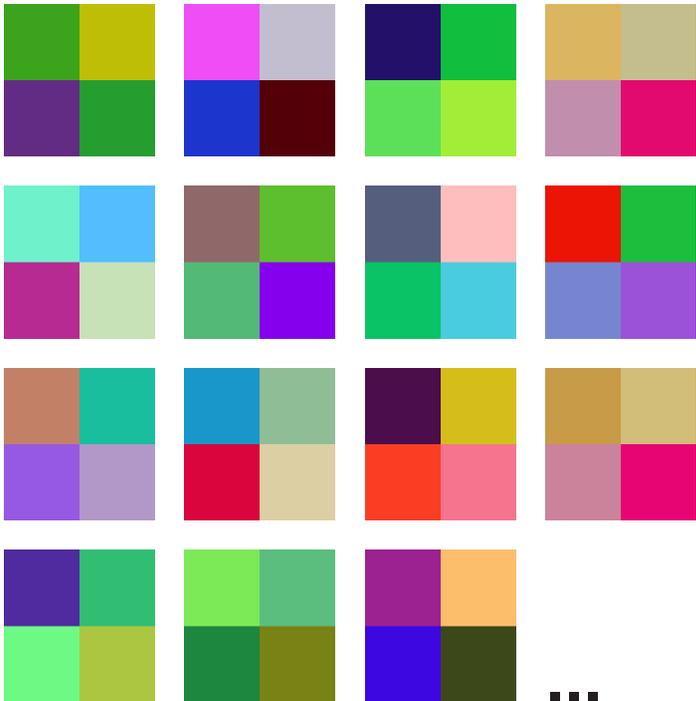
		
Real A	Fake B	Real B

# pix2food

Kreation der Testdaten



**A:**  
Zufällig generiertes Raster aus  
Farben.



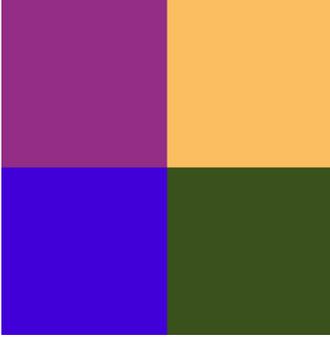
pix2food  
Pix2Pix Modell

Training auf  
vorheriger Seite  
←

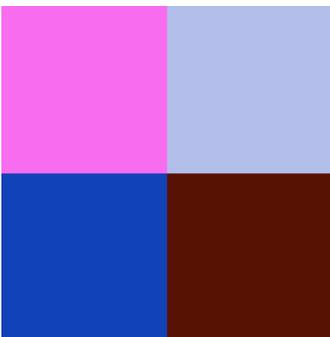
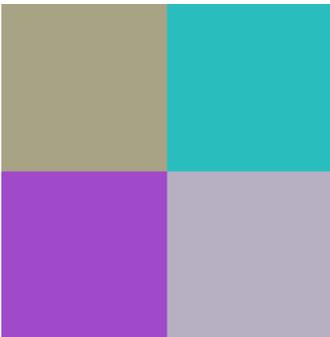
## Ergebnisse

---

Real A



Fake B



Video zum  
Ergebnis →





---

## Reflexion

### Assoziationen

Schon während des Trainingsprozess wurde klar, dass der Algorithmus aus den Daten nicht in der Lage ist Essen zu generieren. Bei einigen Beispielen könnte man in der Lagen sein, ein Essen zu assoziieren. Oft erinnert die Optik aber eher an Erbrochenes. Dies hängt von den Daten ab, die der Algorithmus bekommen hat. Da diese schon brauntönig waren, hat das Modell diese Farbgebung natürlich übernommen. Des Weiteren ist die Auswahl der Pixelwerte für die Rasterung etwas wahllos gewählt, da oft die Farbe der Tischdecke oder des Tellers aufgenommen wird. Infolgedessen hat das Modell keine Chance die Daten vernünftig zu interpretieren.

Einige Tester\*innen hatten tatsächlich die Assoziation von Essen beim betrachten der Bilder. Vor allem bei dem zufällig generierten Essen wurden aber eher Gebirge (Naturaufnahmen), Gedärme und Erbrochenes erkannt. Spannend daran ist, dass es hauptsächlich natürlich und organisch anmutende Formen sind.

### Optimierung

Man kann dieses Experiment in vielerlei Hinsicht optimieren. Zum einen ist da die Auswahl der Trainingsdaten. Hätte man ästhetischere Bilder genommen, wären die Ergebnisse wahrscheinlich auch appetitlicher geworden.

Die Auswahl der Farben, die das Raster bilden ist auch zufällig. Hätte man nur Farben aus dem Essen genommen, hätte dies vielleicht auch zu einer Verbesserung des Endergebnis geführt. Man hätte auch eine subjektive Farberstellung anstreben können, indem man die Leute bittet, die Farben zu zeigen, die sie mit dem Geschmack des Essens assoziieren. Dadurch hätte man auch spannendere Ergebnisse erzeugen können.



---

ground2sky

### Idee

Bei diesem Experiment ging es vor allem um die Erstellung der Trainingsdaten und den Erfahrungsraum, den man dem Modell gibt. Ich zeichnete eine Wanderung ähnlich wie beim BeReal Experiment auf. Alle 10 Sekunden machte ich mit der Front- und Rückkamera des Smartphones ein Bild. Die Rückseite war dabei immer auf den Boden ausgerichtet und die Frontkamera auf den Himmel.

Ziel war es dem Modell einen Zusammenhang von Boden und Himmel beizubringen. Das Setting von einem Wald wurde gewählt, um ein natürlich und fraktales Ergebnis zu generieren.

# ground2sky

Trainingsdaten (769 Bilder)



**A:**  
Foto der Rückkamera  
(Boden).

**B:**  
Foto der Frontkamera  
(Himmel).



Trainingsprozess des Pix2Pix Modells

Epochs: 200

Real A	Fake B	Real B	Epoch
			001
			055
			100
			154

Epoch: 194

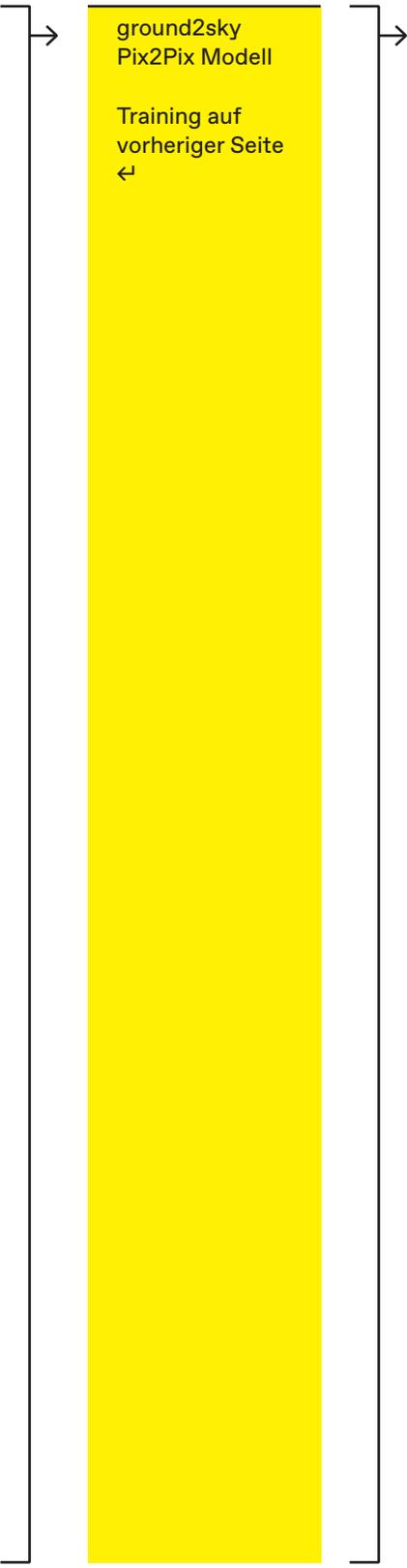


# ground2sky

Kreation der Testdaten



**Input:**  
Aufnahme des Bodens in der Stadt



---

## Ergebnisse

Input

Output



Video zum  
Ergebnis →



# ground2sky

---



---

## Reflexion

### **Erfahrungsraum**

Bei diesem Experiment hat das Modell nur einen bestimmten Erfahrungsraum bekommen. Da es nur mit Naturbildern trainiert wurde, wird sich der Output auch immer daran orientieren. Dadurch entsteht ein starker Kontrast zwischen dem Input und dem Output.

Testet man das Modell nun mit Bodendaten aus der Stadt bei schlechtem Wetter, generiert sich auf der Rückseite dennoch ein blauer Himmel mit Blätterdach.

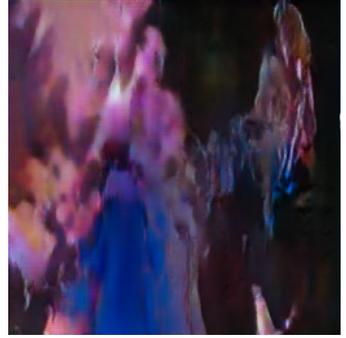
Hieran sieht man wieder, wie abhängig die Ergebnisse von den Daten sind. Ziel ist es nun den Erfahrungsraum des Modells so zu wählen, dass es in eine bestimmte optische Richtung geht, man aber dennoch eine große Varianz der Ergebnisse erzielt.

### **Optimierung**

Mit der Optimierung der Trainingsdaten, könnte man mit diesem Modell noch spannendere Ergebnisse erzielen. Ich hatte als Zieloptik eine fraktale Form. Man könnte das Modell mit dieser Methode auch auf Städte trainieren.

### **Anwendung**

Dieses Modell könnte wieder für einen künstlerischen Zweck genutzt werden. Mit diesem Modell könnte man zum Beispiel eine Aktion zum Begrünen der Städte anregen.



---

## NextFrame

### Wie geht es weiter?

Bei dem Experiment „Next Frame“ geht es darum, der Maschine eine Abfolge von Bildern beizubringen und sie dann aufzufordern anhand dieser weitere zu generieren.

Dazu wird dem Modell ein Bild aus einer Filmszene mit dem jeweiligen nächsten Bild gezeigt.

Das Modell wurde aus verschiedenen Videos trainiert. Zuerst mit einer Tanzszene aus dem Film „Call me by Your Name“ von Luca Guadagnino, da in dieser viel Bewegung passiert und sie ohne Schnitt ist. Sie ist außerdem optisch ansprechend und bunt wodurch spannende Ergebnisse erwartbar waren.

Anschließend wurde das Modell auf ein Video von einem Fluss trainiert. Die Erwartung dabei war, dass die Bewegung des Wassers endlos fortgeführt wird.

Das Testen war iterativ aufgebaut. Das heißt, dass der Frame welcher von dem Modell generiert wurde wiederum als Input verwendet wird. Dadurch entfernt sich der Output immer mehr von dem Original und fängt an eine eigene Form zu gestalten, die mit dem Ursprungsbild keine Ähnlichkeit mehr hat.

# NextFrame

Trainingsdaten (1047 Bilder)



**A:**  
Ein Bild aus der Szene.



**B:**  
Das darauf folgende Bild.

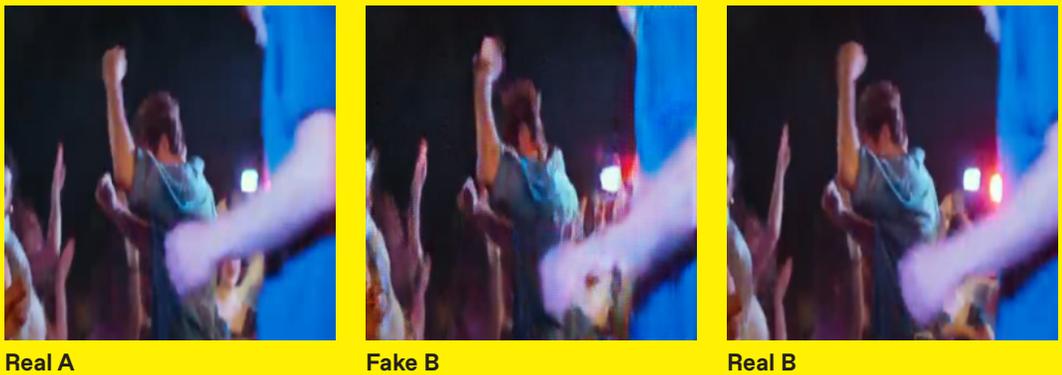


Trainingsprozess des Pix2Pix Modells

Epochs: 200

Real A	Fake B	Real B	Epoch
			001
			050
			100
			150

Epoch: 200



# NextFrame

Kreation der Testdaten

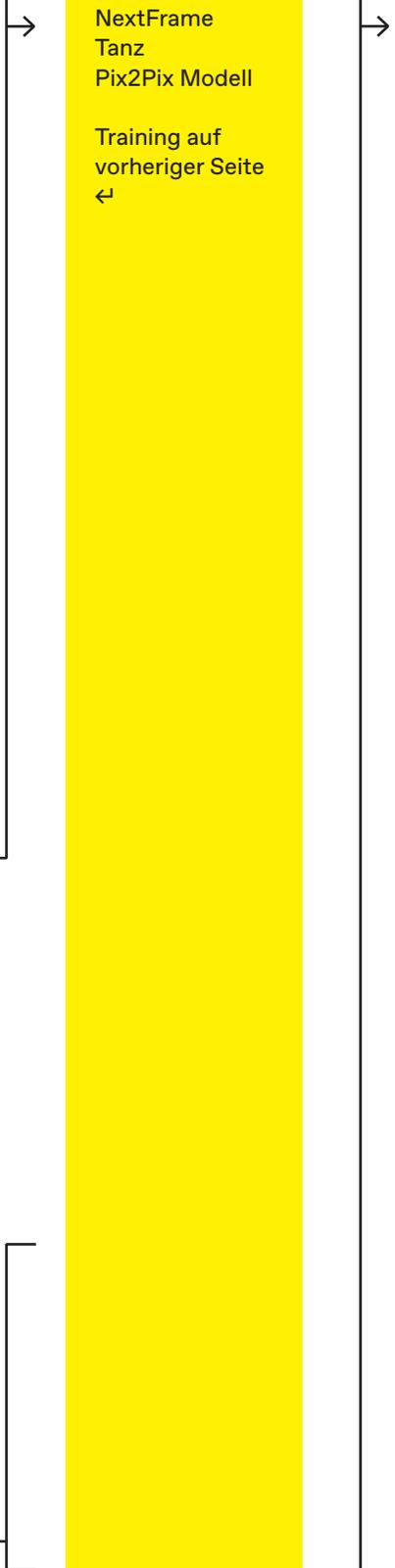


**A:**  
Das erste Bild aus der Szene.  
Dieses Bild wird bei jeder Iteration mit dem neu generierten Bild ausgetauscht.



NextFrame  
Tanz  
Pix2Pix Modell

Training auf  
vorheriger Seite  
←



Ergebnisse



Video zum  
Ergebnis →



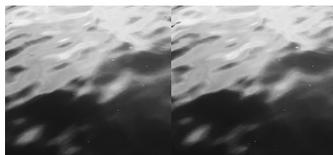
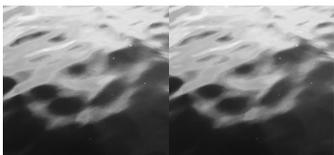
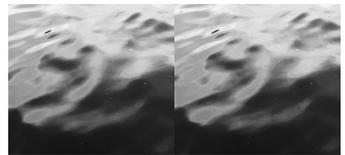
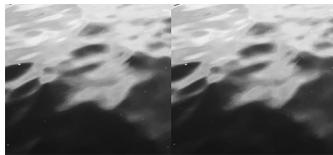
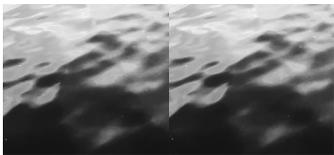
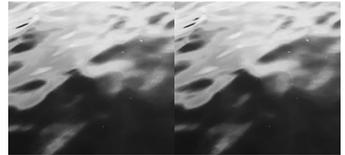
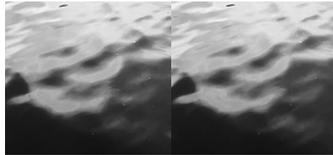
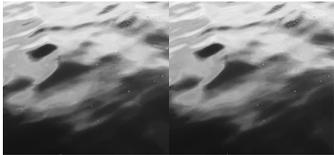
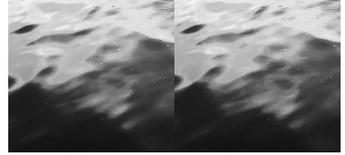
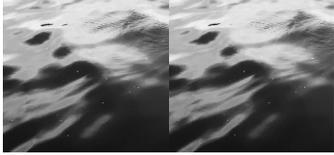
# NextFrame

Trainingsdaten (361 Bilder)



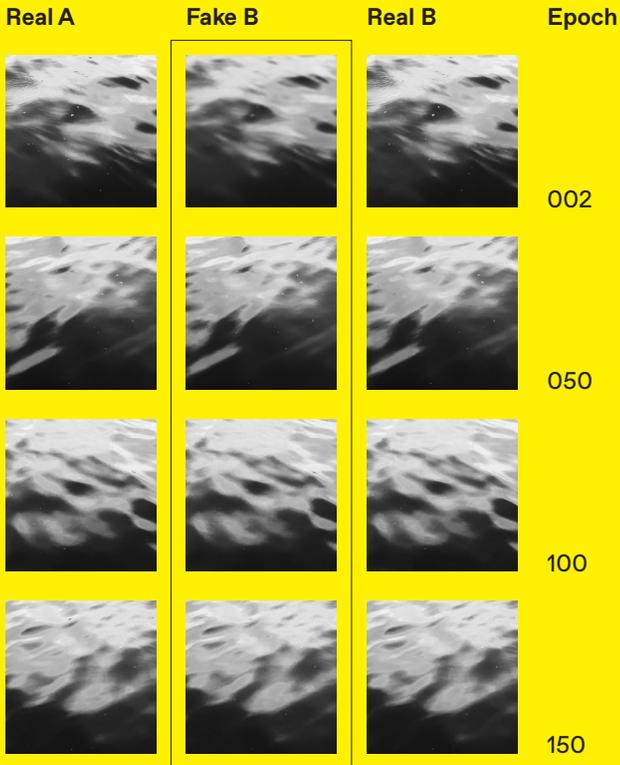
**A:**  
Ein Bild aus dem Video

**B:**  
Das nächste Bild aus dem Video



Trainingsprozess des Pix2Pix Modells

Epochs: 200



Epoch: 200

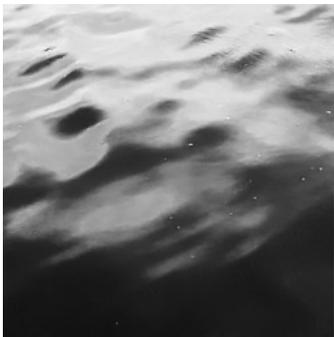


# NextFrame

Kreation der Testdaten

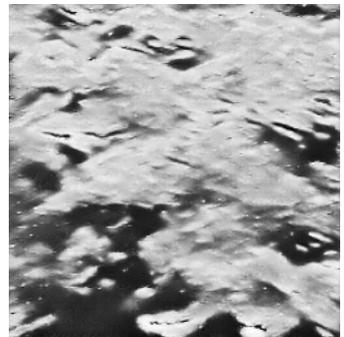
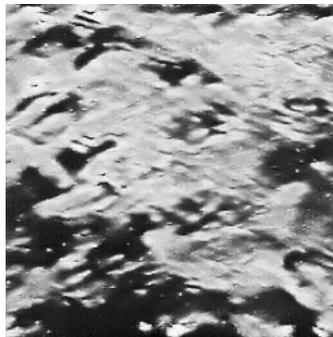
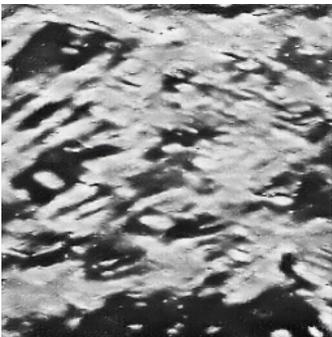
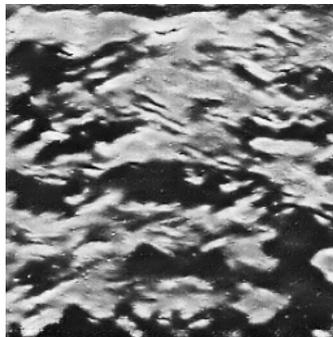
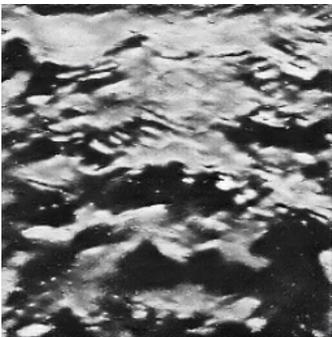
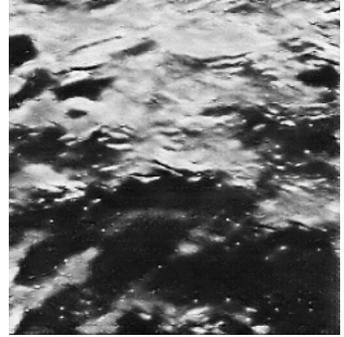


**A:**  
Das Ausgangsbild der Animation. Es wird mit jeder Iteration ausgetauscht.



## Ergebnisse

---

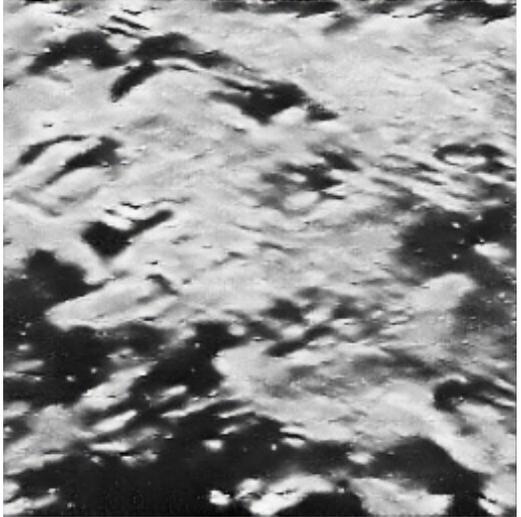
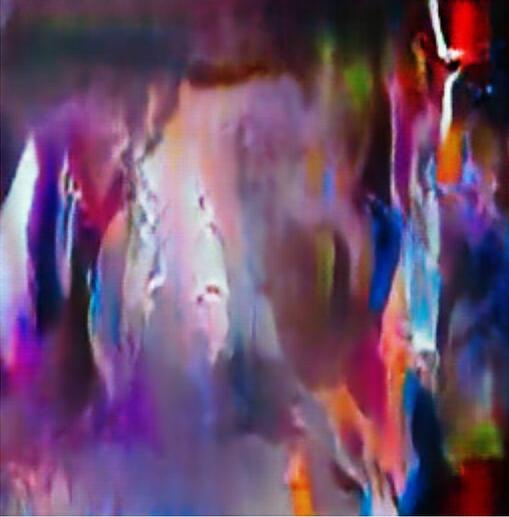


Video zum  
Ergebnis →



# NextFrame

---



---

## Reflexion

### Visueller Output

Die Outputs die das Modell generiert, haben eine sehr abstrakte Anmutung.

### Tanzszene

Die Tanzszene verliert ihren Informationsgehalt schon sehr schnell. Durch das Wiederholen des Prozesses, das Outputbild als Inputbild wiederzugeben zerfließt das Bild. Es bilden sich abstrakte Formen aus Farben, die entstehen und wieder verschwinden. Die Farben aus der Szene, welche sehr bunt gestaltet ist, bleiben aber beibehalten. Auch erinnert die Bewegung der Formen an einen Tanz.

### Wasser

Bei dem Wasser-Input bleibt die inhaltliche Information noch sehr lange erhalten. Man kann die generierten Bilder noch sehr deutlich als Wasser identifizieren. Zuerst werden die künstlichen Wellen etwas kleiner und stärker im Kontrast. Danach erscheint das komplette Bild gräulich überbelichtet und die Wasserstruktur geht verloren.

### Anwendung

Je nach Trainingsdaten kann diese Methode verwendet werden, um ein Video um einen kurzen Moment zu verlängern. Dies kann in dem Fall praktisch sein, wenn man sich beim Schnitt eines Filmes die Einstellung doch etwas länger halten möchte. Dazu sollte die Einstellung allerdings eine monotone, Wiederkehrende Bewegung darstellen.

Das Modell könnte auch als optischer Effekt verwendet werden. In einem Film könnte dieser Effekt einen psychedelischen Rausch kommunizieren oder die Realität zerfließen lassen. Das visuelle Ergebnis kann auch als Visuals für Partys oder Konzerte gebraucht werden.



---

square2face

### **Identitätssuche**

Dieses Experiment beschäftigt sich mit der Identitätssuche in der digitalen Welt. Es geht um ein Quadrat, welches sich über die Fläche bewegt und je nach Positionierung eine Identität erzeugt. Zusätzlich entstehen Töne.

### **Technik**

Für dieses Experiment wurde wieder eine Kombination an KI Werkzeugen genutzt. Zum einen wie in jedem der Experimente das Pix-2Pix Modell und zusätzlich noch die OpenCV Library für Processing. Diese Library ist auch in der Lage Gesichter zu erkennen. Ich nutzte sie hier, um die Trainingsdaten automatisiert zu erstellen. Außerdem verwendete ich das Tool Wekinator zum generieren der Sounds.

### **Datensatz**

Für die Trainingsdaten benutzte ich den gleichen Datensatz wie für das UnBlur-Experiment. Dieser enthielt 2152 Bilder von Gesichtern. Von diesen Gesichtern erkannte die Gesichtserkennung von OpenCV 891 Bilder.

### **Sound**

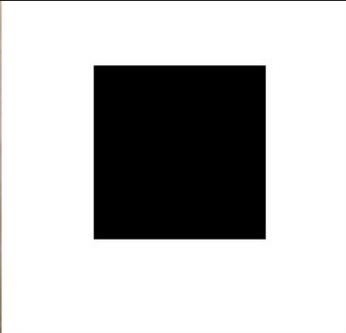
Zum generieren des Sounds diente ich das Tool Wekinator von Rebecca Fiebrink. Dieses ermöglicht, Machine Learning für kreative Zwecke zu verwenden. Es eignet sich sehr gut für herkömmliche Creative Coding Software wie Processing. Das generieren von Sounds ist hiermit auch sehr einfach, da es schon vorprogrammierte Beispieldateien gibt.

### **Kommunikation**

Durch die vorhergegangenen Experimente war zu erwarten, dass das generieren eines Gesichtes eher verstörend wirkt. Dies nutzte ich in diesem Experiment, um diesen Effekt mit dem Sound zu verstärken.

# square2face

Trainingsdaten (890 Bilder)



**A:**  
Ein Bild eines Gesichtes

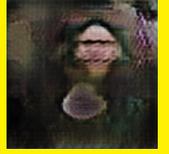
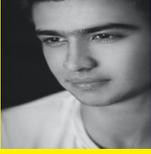
**B:**  
Ein schwarzes Quadrat wird  
an der Position des Gesichtes  
platziert.



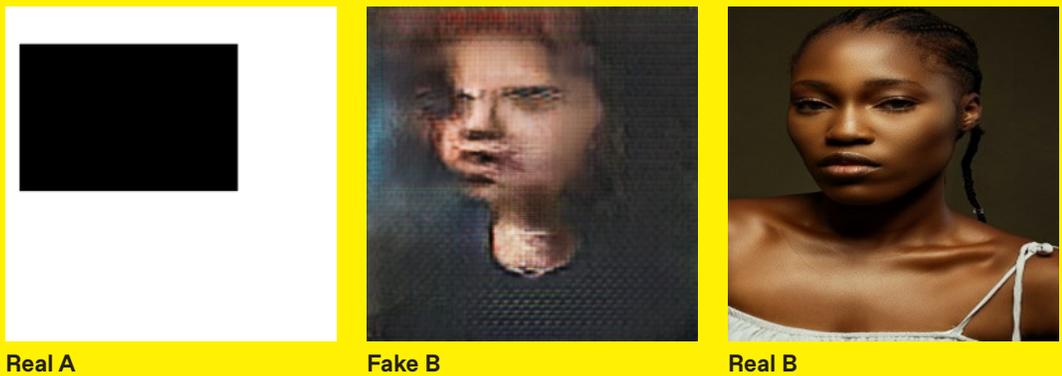
...

Trainingsprozess des Pix2Pix Modells

Epochs: 200

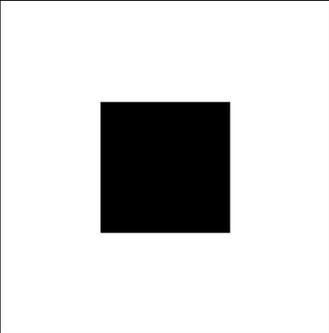
Real A	Fake B	Real B	Epoch
			002
			058
			096
			137

Epoch: 197

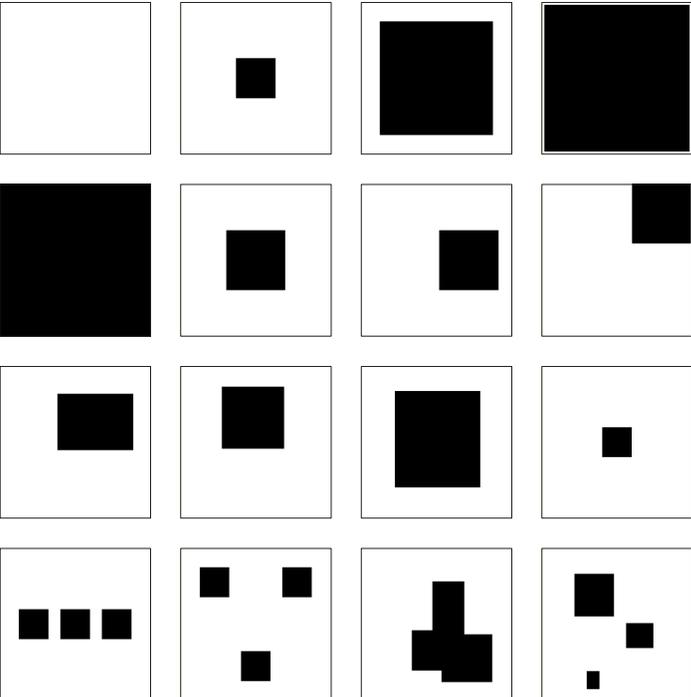


# square2face

Kreation der Testdaten



**Input:**  
Animation eines Quadrates,  
welches sich über die Fläche  
bewegt.



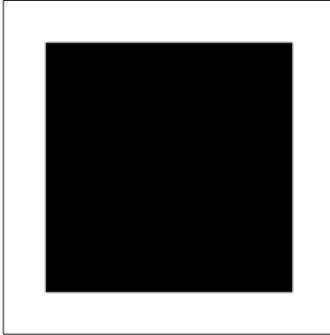
square2face  
Pix2Pix Modell

Training auf  
vorheriger Seite  
←

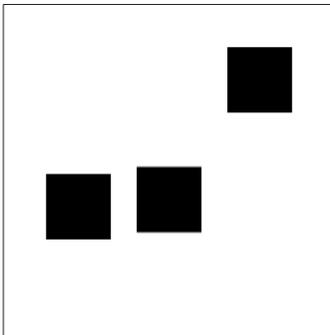
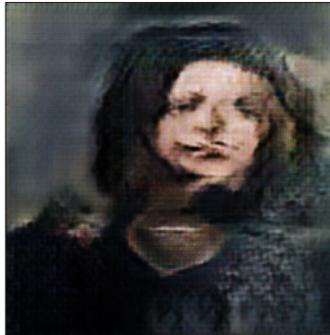
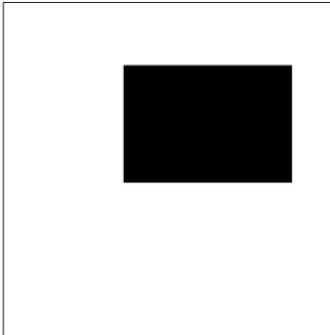
## Ergebnisse

---

Input



Output

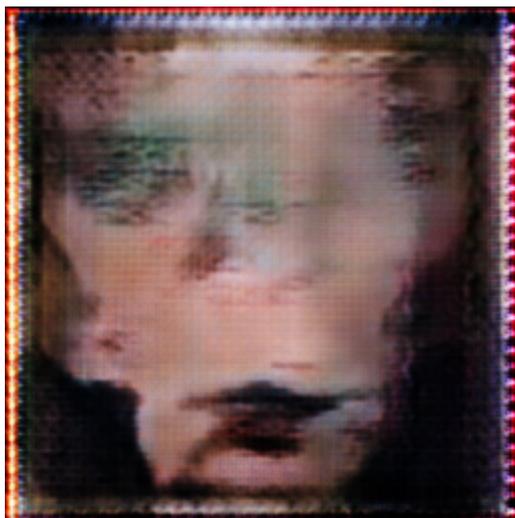
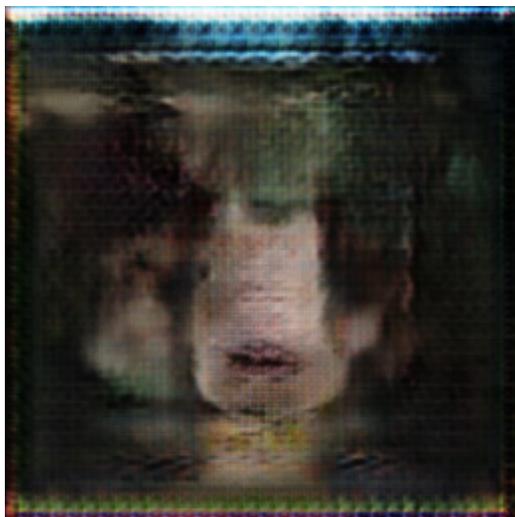


Video zum  
Ergebnis →



# square2face

---



---

## Reflexion

### Anmutung

Das Ergebnis dieses Experiments wirkt wie erwartet abschreckend. Durch die Kombination mit dem Ton ist es schon fast unangenehm. Es bilden sich wieder bis zur Unkenntlichkeit verzerrte Gesichter. Das schwarze Quadrat bewegt sich in einer suchenden Bewegung über die Fläche, jedoch ohne etwas zu finden.

### Optimierung

Mit einem größeren Datensatz könnten die Ergebnisse deutlich schärfer gelingen. Diese Unschärfe macht für mich bei der Aussage, die ich treffen möchte jedoch auch Sinn.

### Anwendung

Das aus dem Experiment erfolgte Ergebnis kann auch im künstlerischem Bereich angewendet werden. In optimierter Form könnte es auch als künstlerische Installation verwendet werden. Vielleicht auch mit live Userinput.

HeIvetica to Didot

---

helvetica2didot

### Schrift

Bei der Beschäftigung mit KI für die Nutzung in Design, kommt man um den Bereich Typografie nicht herum.

Dieses Experiment ist ein erster Versuch von mir Tyledesign mit dem Pix2Pix Modell zu verbinden. Da sich Pix2Pix unter anderem auch für Styletransfers (das Übertragen eines Bildstils auf ein anderes) eignet, wollte ich dies auch mit Schrift versuchen.

Für den Versuch wurden zwei absolute Klassiker der Schriftgestaltung verwendet: Helvetica und Didot. Die Idee war es dem Modell jede Glyphe, die in beiden Schriften vorhanden war, paarweise zu zeigen. Daraus sollte das Modell nun Muster erlernen und aus der Groteskschrift Helvetica die klassizistische Antiqua Didot gestalten.

### Animation

In einem weiteren Schritt wurde getestet, wie das Modell nun mit animierten Zeichen umgeht. Was passiert, wenn die Glyphe nicht wie erwartet genau in der Mitte der Fläche steht, sondern sich von rechts nach links bewegt.

# helvetica2didot

Trainingsdaten



ú ú

**A:**  
Eine Glyphen in der Schrift  
Helvetica gesetzt

**B:**  
Die gleiche Glyphen in der  
Schrift Didot gesetzt

( (

g g

1 I

> >

M M

! !

ž ž

r r

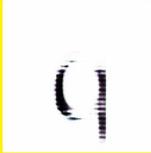
~ ~

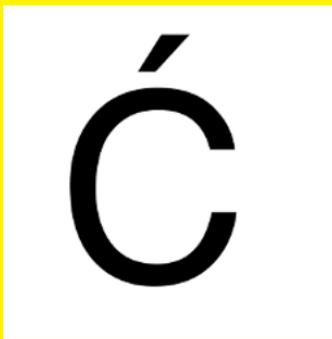
d' d'

! !

E E

Trainingsprozess des  
Pix2Pix Modells

Real A	Fake B	Real B	Epoch
			003
			041
			102
			155



Real A



Fake B



Real B

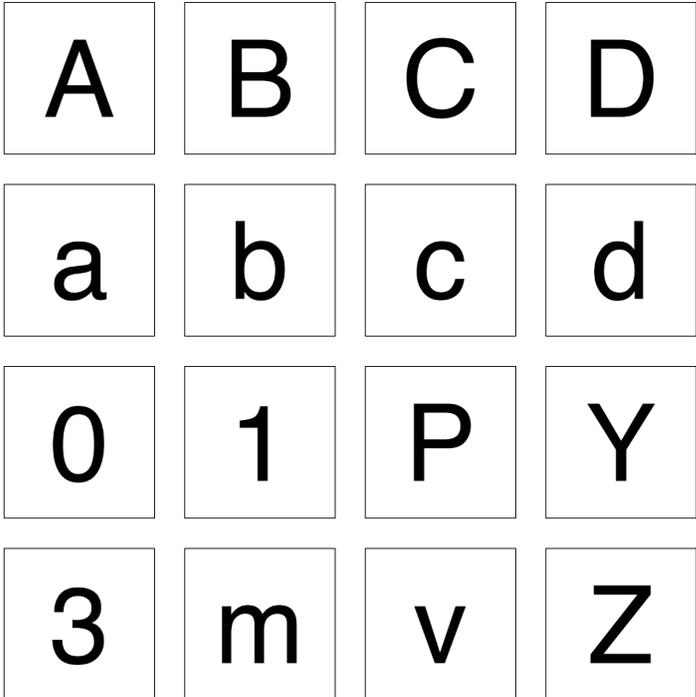
# helvetica2didot

Kreation der Testdaten



**Input:**

Das gesamte Alphabet  
gesetzt in Helvetica



helvetica2didot  
Pix2Pix Modell

Training auf  
vorheriger Seite  
←

---

Ergebnis



O



G



M



f



e



m



u



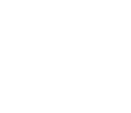
x



o



g



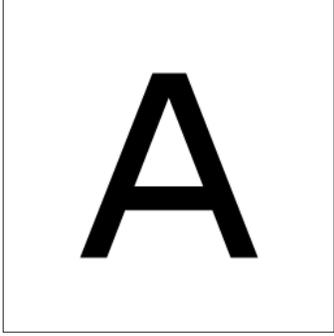
m



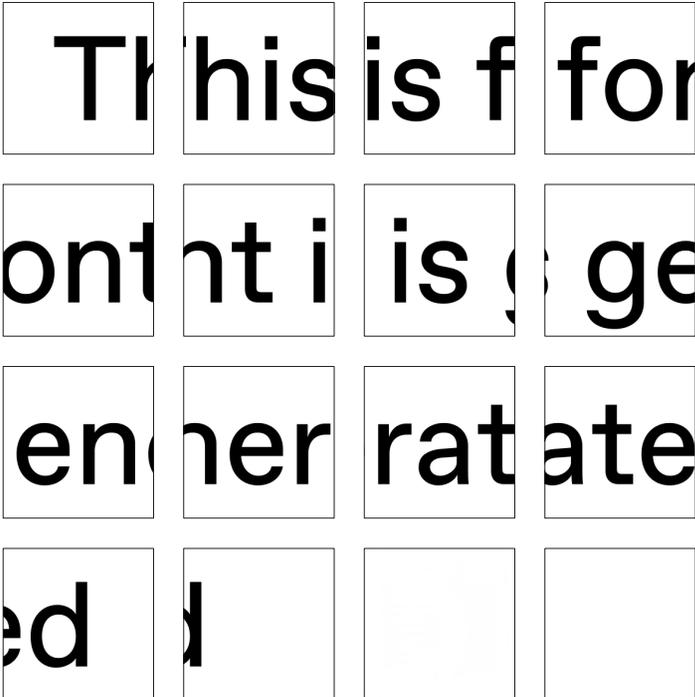
e

# helvetica2didot

Kreation der Testdaten



**Input:**  
Die Frames einer Textanimation.



helvetica2didot  
Pix2Pix Modell

Training auf  
Seite 84  
←

## Ergebnisse

Input

A

Output

A

his

his

ge

ge

Video zum  
Ergebnis →



# helvetica2didot

---

# Helvetica to Didot

Pos et ea etur alitatur re plibealia doluptatint vent  
quassit magnis ea corit, quostot atesedis quide vid  
quo veres rernam quis event, cuplati dolora volupta  
lestiis suntiis lisiilia siteniensis cuplatatur reiciis ciatur,  
et, ut eatur, nonseria voluptiet abo. Ut officiiis si  
doluptaquis esti volumque ne conecat emperum et  
dolupfibus, nos autaectal.

Milioriatem. Neque aut fugit pratet quassim faciis mo  
et quassin ctaque eatempo rerrore stiumquis explibus  
maximi, quos quodit voloresti consedis dolorem aut  
omnimporae voluptam sed qualibe atiosse cuplibcius  
simin nessimi nloria aut aut liquibusam quodi qui  
optassi rende nossi qui blaborepudae volorbsam  
quam quaerror aspicias aut experem comnis etusacali  
digendi taspindis rem esedi ditiat latepos simagnis  
repudam nobit rati dolorpos veliti nis aliqui reped  
et, adipisimus explige nissequ iberiore ipsae que

**Neue Schrift**

Das Ergebnis dieses Experiments ist eine neue Schriftart. Besonders interessant ist, dass diese Schrift, obwohl sie von einer Maschine gemacht wurde, eine sehr organische Formsprache hat. Sie wirkt eher haptisch als digital.

Die Qualität der Lesbarkeit lässt sehr stark nach. Manche Serifen sowie einige Haarlängen verschwinden, während sich manche Serifen verbinden und den Buchstaben schließen.

Das Pix2Pix Modell arbeitet ausschließlich mit Pixeln. Also musste die Schrift erst gerastert werden und nachdem sie durch das künstliche neuronale Netz geschickt wurde, wieder in Vektorgrafiken umgewandelt werden. Dadurch hat sie Einiges an Bildinformation verloren.

**Anwendung**

Dieses Modell kann als Inspiration für Schriftgestalter\*innen dienen. Dadurch kann man angeregt werden, die formale Schriftgestaltung aufzubrechen und experimentellere Buchstaben zu gestalten. Durch das Training verschiedener Schriftarten kann man zu ganz unterschiedlichen Ergebnissen kommen.

Interessant wäre auch das Umwandeln von Scriptschriften zu Grotesk.

Durch das Mischen von Schriften, kann man mit diesem Modell auch eine große Anzahl von Displayschriften gestalten. Die Auswahl der Schriften, auf die das Modell trainiert wird, ist hierbei der wichtigste Faktor.

**how  
bold  
is  
to**



---

regular2bold

### Zu Bold

Auf Grundlage des Experimentes zuvor (Helvetica2Didot) wurde dieses Modell von Regular auf Bold trainiert. Dazu wurde eine wahllose Auswahl an 6 Schriften getroffen. Aus dieser Schrift wurden die Glyphen einmal in Regular und einmal in Bold, für das Pix2Pix Modell optimiert und nebeneinander gesetzt.

Durch diesen Vergleich sollte das Modell lernen, wie sich die Schriften verändern, wenn sie Bold werden.

Im zweiten Schritt schaltete ich das System wieder in schleife, womit Buchstabe bis zu Unkenntlichkeit „bolder“ wurde.

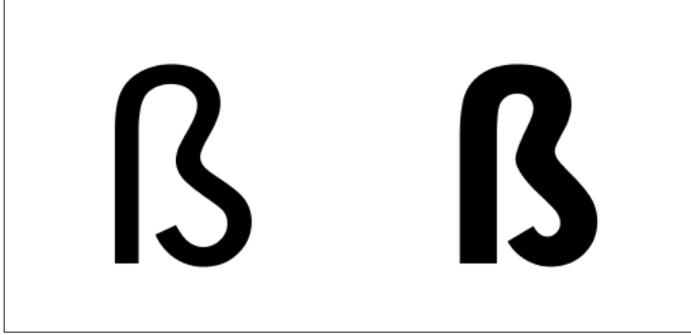
Im Gegensatz zu den anderen Experimenten läuft dieses Experiment durch 300 Epochs.

### Zufall

Bei den vorigen Experimenten ist aufgefallen, dass das Pix2Pix Modell eine Zufallsvariable beinhaltet. Denn auch wenn man das gleiche Inputbild eingibt, entstehen immer minimale Unterschiede. Besonders gut kann man das bei den Video Ergebnissen sehen. Dabei entsteht ein Flakern. Um diese Unsicherheit des Modells zu zeigen, generierte ich in einem weiteren Schritt drei mal den gleichen Buchstaben. Diese färbte ich dann cyan, magenta und gelb ein und multiplizierte die Farben ineinander. Dadurch wird der Buchstabe an den Flächen, die bei allen drei eingefärbt waren schwarz und der Rest mischte sich je nach Deckung.

# regular2bold

Trainingsdaten (1478 Bilder)

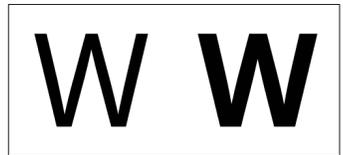
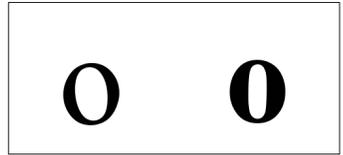
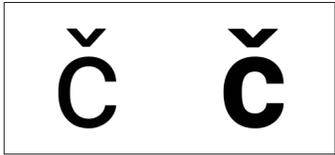


**A:**  
Die Glyphen in Regular  
gesetzt.

**B:**  
Die gleiche Glyphe in Bold  
gesetzt.

Die verwendeten Schriften  
sind:

- ↳ Arial
- ↳ Comic Sans
- ↳ IBM Plex
- ↳ Lab Grotesk
- ↳ Roboto
- ↳ Times New Roman



...

Trainingsprozess des Pix2Pix Modells

Epochs: 300

Real A	Fake B	Real B	Epoch
			001
			051
			106
			155

Epoch: 298

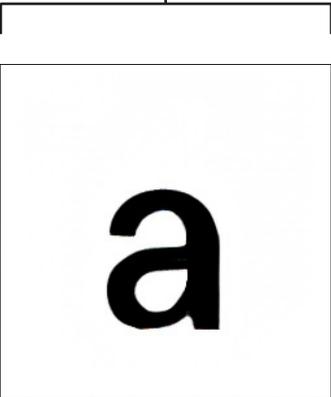


# regular2bold

Kreation der Testdaten

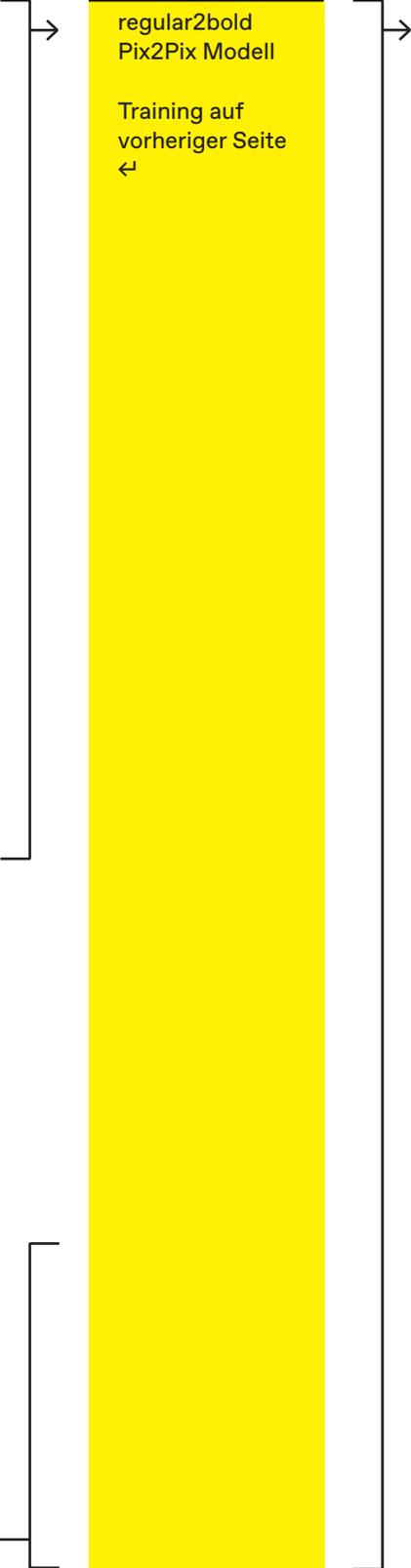


**Input:**  
Der Buchstabe in Regular  
gesetzt.

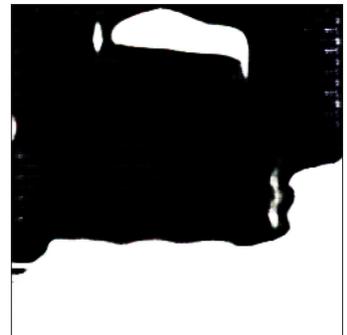
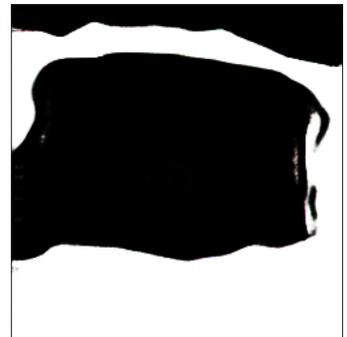
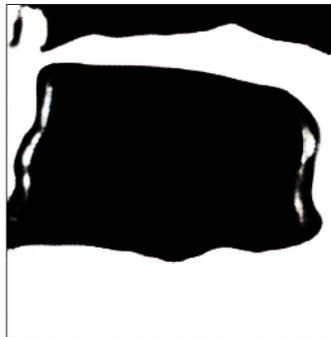
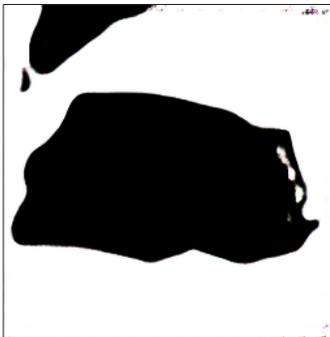


regular2bold  
Pix2Pix Modell

Training auf  
vorheriger Seite  
←



## Ergebnisse



Video zum  
Ergebnis →

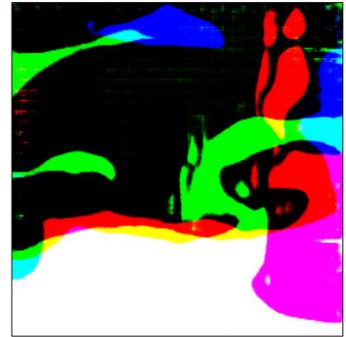
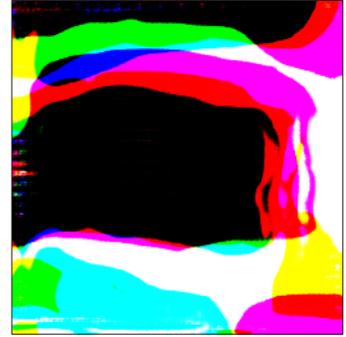
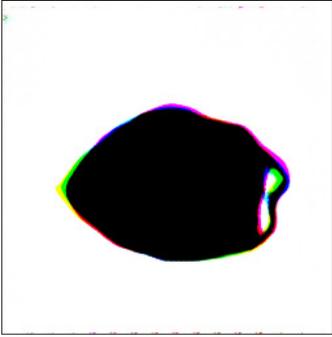


# regular2bold

---



Alphabet  
Grid  
Video zum  
← Ergebnis



bolder than  
bold  
Video zum  
Ergebnis →



Lost in Data  
Video zum  
Ergebnis →



# regular2bold

---



Type Tool  
Video zum  
← Ergebnis

---

Reflexion

## Anwendung

Das trainierte Modell kann in der ersten und zweiten Schleife noch dafür verwendet werden, eine Schrift die nicht in Bold vorliegt in diesen Schnitt umzuwandeln. Natürlich funktioniert dieser Effekt nicht einwandfrei.

Die Schrift die mittels der Überlagerung produziert wurde, kann dabei schon als eigene Displayschrift verwendet werden. In dem Beispielvideo „Lost in Data“ sieht man eine mögliche Verwendung der Schrift für Kommunikationsdesign. Die Aussage wird durch die Form der Schrift bestärkt.

Um die Schrift auch in statischer Form nutzen zu können, programmierte ich noch ein Tool in Processing. In diesem kann man etwas schreiben und den Verzerrungsgrad der Schrift mit einem Schieberegler einstellen. Mit diesem Tool kann man auch Überschriften für Plakate, Publikationen etc. machen.

Durch das eingeben verschiedener Inputschriften kann man auch verschiedene Schriften in kurzer Zeit generieren.



---

## Zoom

### Idee

Die Idee bei diesem Experiment ist es, das Modell auf eine programmierte Bewegung zu trainieren. Optisch ansprechende Ergebnisse erhoffte ich mir mit einer Zoom Bewegung. Dazu trainierte ich die Maschine zum einen auf Bildern von Städten und auf Bildern von natürlichen Berglandschaften.

Das trainierte Modell wurden dann wieder in Schleife gesetzt, wodurch die Maschine wiederum das generierte Bild als Input bekam. Dadurch generiert der Algorithmus immer neue Bilder, die die Bewegung von einem Zoom abbilden.

Mich interessierte besonders, wie lange die Maschine ein Bild anzeigt, welches mit dem ersten Inputbild noch etwas gemein hat und ab wann es nur noch Rauschen ausgibt.

### Datensatz

Den Datensatz habe ich automatisiert von Unsplash heruntergeladen. (Quelle: <https://unsplash.com/de>) Durch einen Trick ist es in Firefox möglich sich, alle angezeigten Medien auf der Website herunter zu laden. Dadurch musste ich nicht jedes Bild einzeln runterladen und hatte die Möglichkeit das Modell auf verschiedenen Daten zu trainieren.

# Stadt Zoom

Trainingsdaten (694 Bilder)



**A:**  
Das Bild einer Stadt

**B:**  
Das Bild einer Stadt leicht  
vergrößert

Durch den Wert der Vergrößerung kann man die spätere Animationsgeschwindigkeit einstellen.



Trainingsprozess des Pix2Pix Modells

Epochs: 200

Real A	Fake B	Real B	Epoch
			001
			058
			090
			150

Epoch: 199

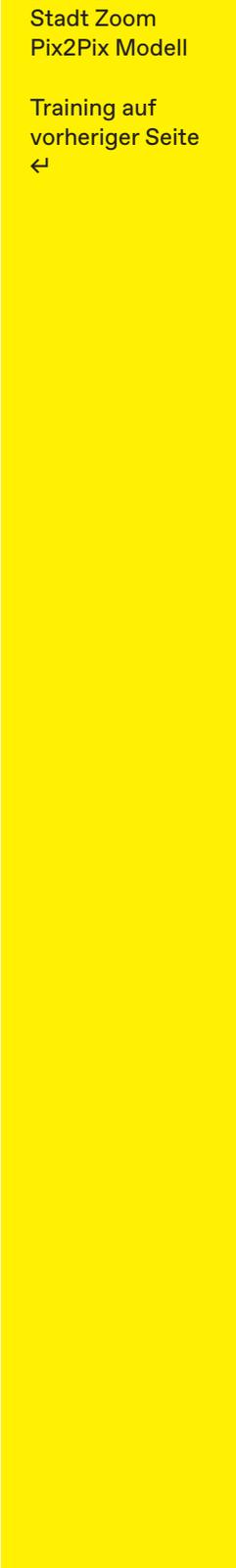


# Stadt Zoom

Kreation der Testdaten



**Input:**  
Das Bild einer Cityskyline



Stadt Zoom  
Pix2Pix Modell

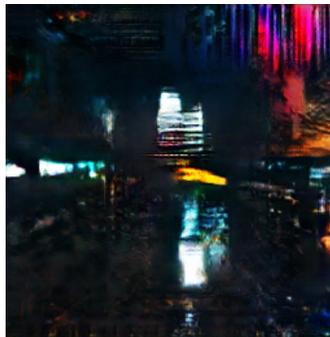
Training auf  
vorheriger Seite  
←



---

## Ergebnisse

### Outputs



Video zum  
Ergebnis →



# Berge Zoom

Trainingsdaten (694 Bilder)



**A:**  
Das Bild einer Berglandschaft



**B:**  
Das Bild einer Berglandschaft  
leicht vergrößert

Durch den Wert der Vergrößerung kann man die spätere Animationsgeschwindigkeit einstellen.



Trainingsprozess des Pix2Pix Modells

Epochs: 200



Epoch: 199



# Berge Zoom

Kreation der Testdaten



**Input:**  
Das Bild einer Cityskyline

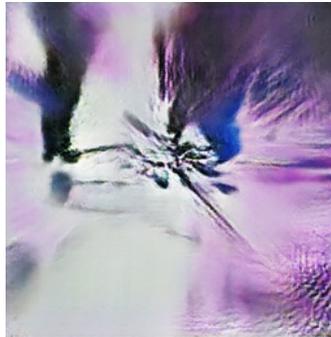
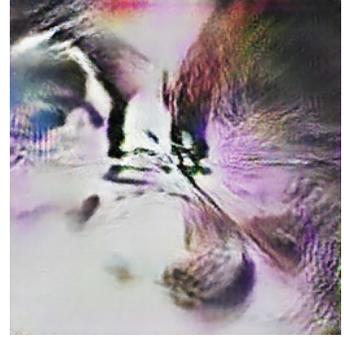
Berge Zoom  
Pix2Pix Modell

Training auf  
vorheriger Seite  
←



## Ergebnisse

### Outputs

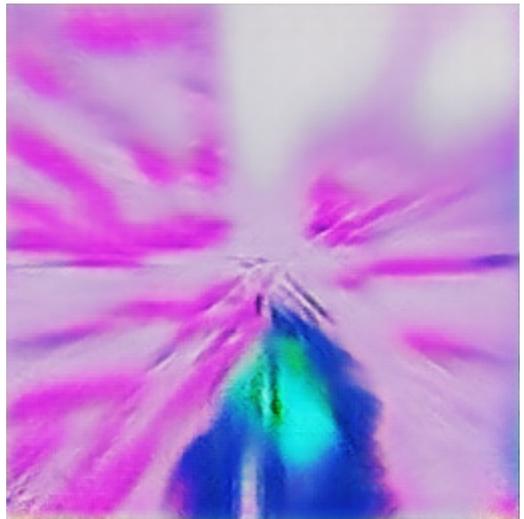
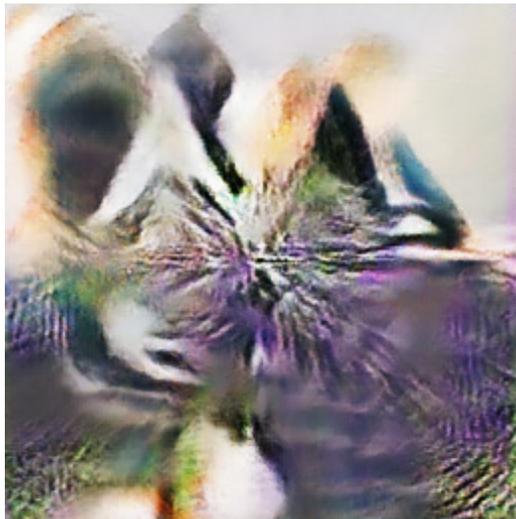
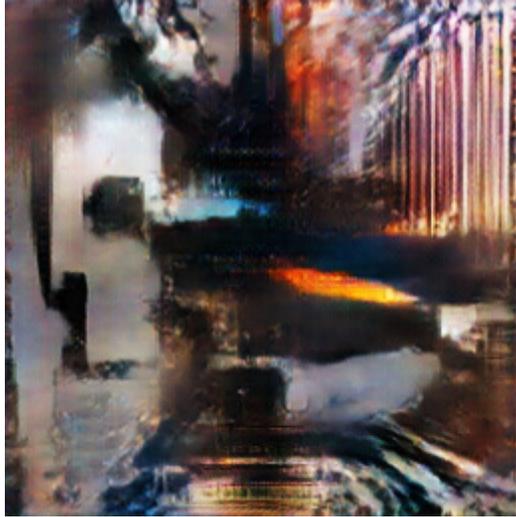


Video zum  
Ergebnis →



# zoom Pix2Pix

---



---

## Reflexion

### Optik

Bei dem Ergebnis, welches auf die Stadtdatei trainiert wurde, ist zu beobachten, dass das Modell die Optik einer Stadt ganz gut gelernt hat. Auch nach vielen Schleifen, die das Bild durch das Modell geht, ist immer noch der Eindruck einer Stadt zu erschließen. Man kann auch später noch einige Hochhäuser, Straßen und Lichter erkennen. Außerdem ist die Zoombewegung auch deutlich zu erkennen. Da die Trainingsdaten vor allem Bilder von Städten bei Nacht zeigen bleibt diese Optik lange erhalten. Die generierten Bilder wirken dennoch kaputt, und sehen der Cyberpunk-Ästhetik ähnlich.

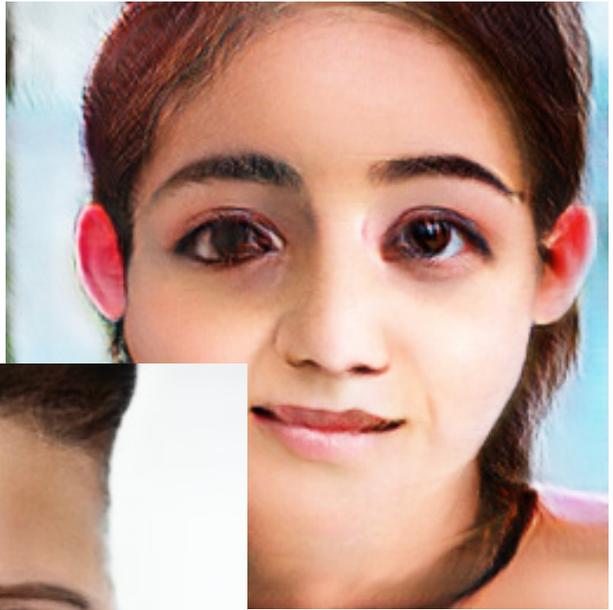
Bei den Berglandschaften schafft es das Modell nicht wirklich neue Landschaften zu gestalten. Die Animation läuft schnell in einen Violetttönen über, welcher nur mit leichten Störungen versehen ist.

Hierbei habe ich mich entschieden, ein Bild von mir selber als Inputbild zu geben. Erhofft hatte ich mir, dass das Bild immer näher an mein Gesicht zoomt und daraus ein Gebirge formt. Dies ist leider nicht der Ausgang dieses optischen Experimentes geworden. Mein Gesicht löst sich lediglich in einer Zoombewegung auf. Daran sieht man wieder, dass man nie vorhersagen kann, wie das Modell ein Bild interpretiert. Erst durch testen des Modells kann man darauf schließen, was das Modell mit dem Bild anstellt.

### Anwendung

Diese Methode kann wiederum für einen optischen Effekt genutzt werden. Durch den Zoom erinnert es an die Stargate Szene aus dem Film 2001: Odyssee im Weltraum.

Durch Optimierung der Daten oder durch versuchen verschiedener Bewegungen, könnte man diese Methode noch weiter erforschen und mit Sicherheit weitere interessante optische Ergebnisse erzeugen.



## Technik

Für dieses Experiment habe ich ein anderes Modell verwendet, als für die ganzen anderen Experimente. Den Code den ich bisher verwendet habe, unterstützt auch unpaired Image-to-Image Translation. In diesem Fall wird das ganze mit CycleGAN gemacht. Die Architektur dieses Modells ist komplexer als die von Pix2Pix. Es basiert grundlegend auf Cycle-Consistent. Eine gute Analogie dazu ist das Übersetzen von Sprachen. Übersetzt man einen Satz von Englisch zu Deutsch, und wieder zurück von Deutsch zu Englisch, kommt im Optimalfall der gleiche Satz bei raus. Das Gleiche macht auch die Maschine in diesem Modell. Es übersetzt ein Bild in dem Stil von einem anderen und übersetzt es dann zurück. Idealerweise kommt dann das gleiche Bild wie das Inputbild raus. Durch das Testen dieser Übersetzungsarbeit trainiert nun das Modell.

## Avatare

In diesem Fall stellte ich mir die Frage, wie wir unsere Identität online ausdrücken. Sofort kamen mir Avatare in den Sinn. Diese gestalten wir immer nach unseren Vorlieben. Dabei kann es sein, dass wir unseren Avatar nach unserem echten Erscheinungsbild gestalten, es kann aber auch sein, dass wir uns von unserem eigenen Körper entfernen und ein ganz anderes Aussehen wählen. Ein häufiges Beispiel ist das Ändern des Geschlechts. Diese Avatare haben dann nichts mehr mit der Person zu tun, die dahinter steckt.

Mit meinem Modell wollte ich dieses Vorgehen aufbrechen. Ziel war es, einen automatisierten Avatar-Generator zu trainieren, welcher sich ausschließlich auf das echte Äußere bezieht.

## Datenbank

Da ich bei diesem Modell nicht mehr darauf angewiesen war, gepaarte Bilder zu finden, konnte ich eine freie Datenbank von Bildern mit Gesichtern wählen. Als Gegenseite suchte ich mir eine Sammlung von Avataren. Für dieses Experiment wählte ich die Avatare von Snapchat aus, die so genannten Bitmojis.

Die Datenbank der Gesichter umfasst 7219 Bilder. Die der Bitmojis 4084 Bilder. Es ist somit auch die größte Datenmenge, die ich für ein Experiment genutzt habe. Dadurch dauerte der Trainingsprozess jedoch auch 10 Tage.

# Bitmoji2Face

Trainingsdaten

7219 Bilder



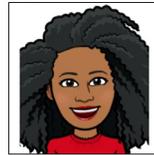
**A:**  
Fotografien von Gesichtern



4084 Bilder

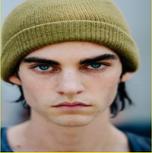


**B:**  
BitMoji Avatare von Snapchat



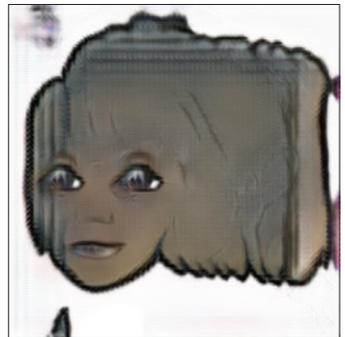
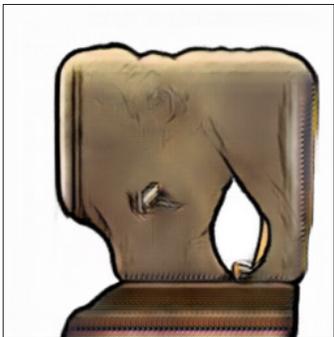
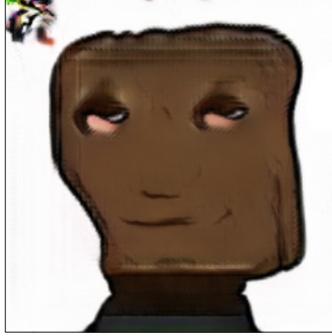
## Trainingsprozess des CycleGAN Modells

Epochs: 200

Real A	Real B	Fake A	Fake B	Epoch
				001
				050
				100
				150
				150
				150
				150

# Bitmoji2Face

---



**Overfitted**

In der Datasience gibt es den Begriff Overfitted. Dieser bedeutet, dass man dem Modell zu viele Trainingsdaten gibt und das Ergebnis dadurch unbrauchbar macht. Nach meiner Empfindung ist dies hier passiert. Ich habe mich zwei relativ großer Datenbanken bedient. Nach den 10 Tagen Trainingszeit waren die Ergebnisse jedoch schlechter, als zur Mitte des Experiments.

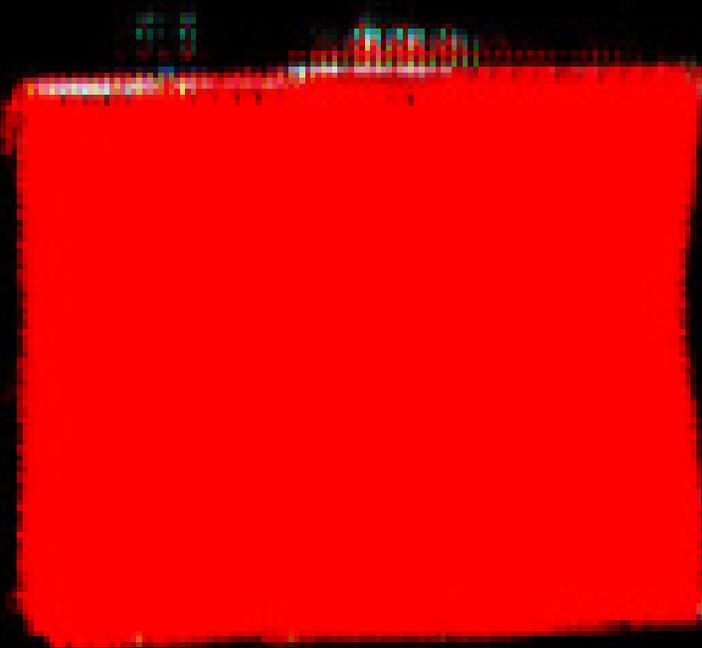
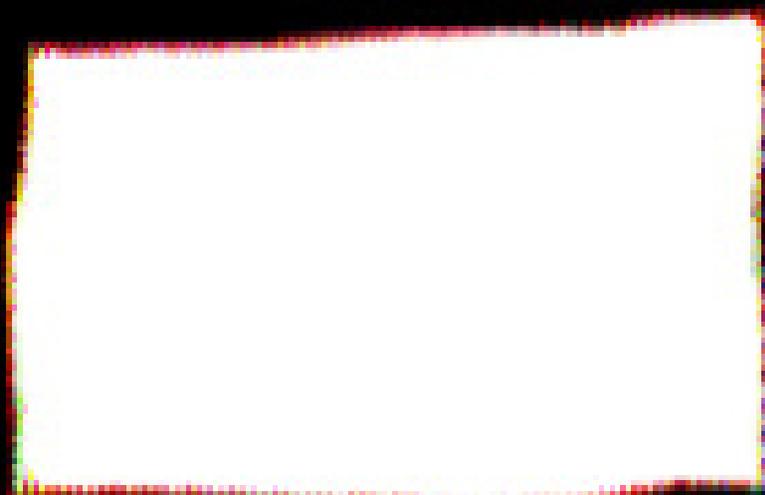
Das Modell hat es nicht geschafft brauchbare Ergebnisse zu erzielen. Dies kann an verschiedenen Faktoren liegen. Zum einen können die Anzahl als auch die Qualität der Trainingsdaten dazu geführt haben. Zudem kann es auch sein, dass das Modell gar nicht dazu in der Lage ist, so eine Übersetzungsleistung zu erzeugen. Die beiden Bildstile unterscheiden sich doch sehr stark.

Da das Modell so lange zum trainieren gebraucht hat und die Ergebnisse nicht wirklich überzeugend waren, habe ich mich entschieden lieber wieder weiter mit dem Pix2Pix Modell zu arbeiten als dieses Modell neu zu erlernen.

**Anwendung**

Die Optik der Bilder ist sehr verstörend. Man sieht verzerrte Fratzen mit ausradierten Augen oder Gesichtsmerkmale an Stellen wo sie gar nicht hingehören. Diese Optik ist in der Werbung für die breite Maße wohl nicht vertretbar. Auch glückt der Versuch aus einem echten Gesicht einen Avatar zu gestalten kaum. Es werden eher gruselige graue Farbflächen mit einem Auge an bestimmten Stellen generiert.

Im kulturellen Bereich könnte man solche Grafiken vielleicht noch verwenden. Sie werfen zumindest ein sehr negatives Licht auf das behandelte Thema: Online Identität.



---

rotes und weißes Rechteck

## Programmierte Regeln

Den Datensatz für dieses Modell generierte ich selber. Dazu benutzte ich Processing. Ich wollte testen, wie gut die Maschine die optischen Regeln, die ich in Processing programmiert hatte, lernen kann.

Dazu überlegte ich mir ein einfaches optisches Regelwerk:

- ↳ Ein schraffiertes Rechteck ist rot
- ↳ Ein leeres Rechteck ist weiß
- ↳ Alles andere ist schwarz.

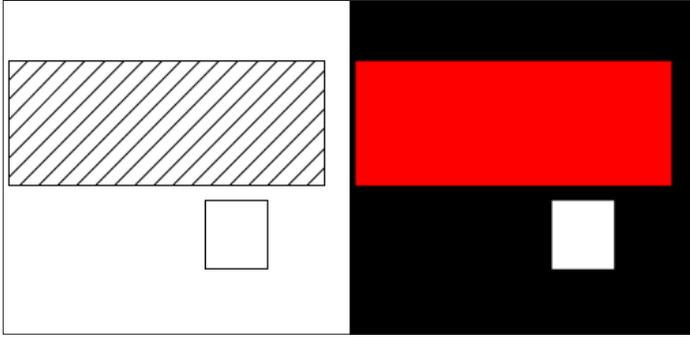
Dieses Regelwerk konnte ich schnell und einfach in Processing in den richtigen Pixelmaßen und für das Modell optimiert in Bilder übersetzen. Außerdem gelang es mir automatisiert eine gewünschte Anzahl an Trainingsdaten in Sekunden zu erzeugen. Dadurch hatte ich auch die Möglichkeit, die Daten nach meinem Geschmack schnell zu ändern.

## User Input

Beim Testen der Daten wollte ich versuchen von mir gezeichneten Input zu geben. Dadurch konnte ich mit etwas haptischen, etwas zeichnen und dann vom Modell interpretieren lassen.

# rotes und weißes Rechteck

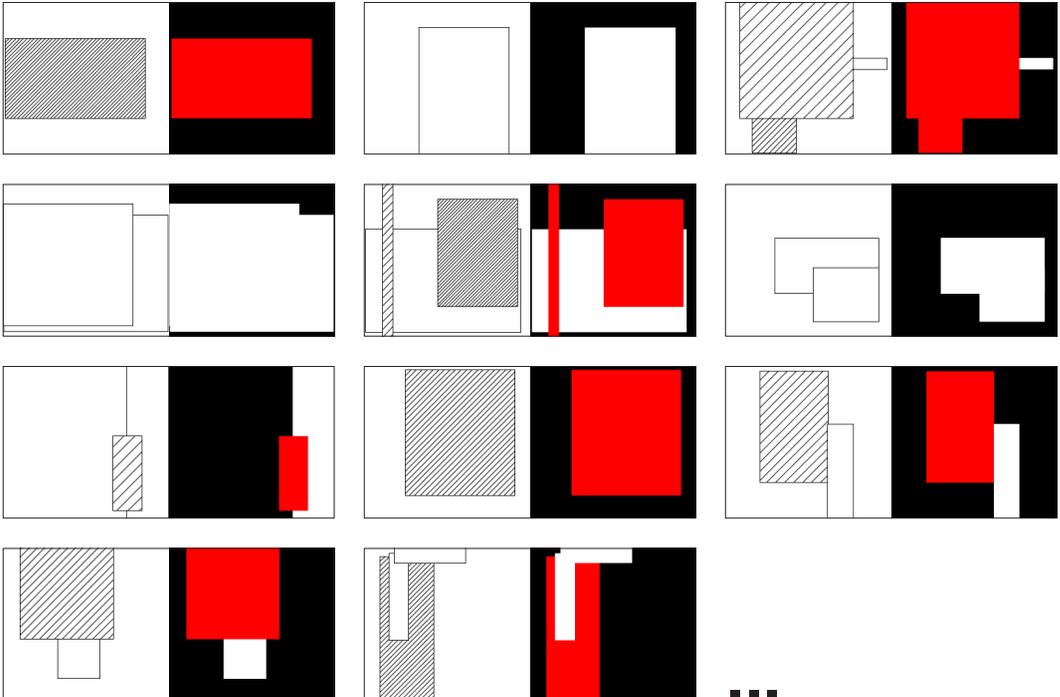
Trainingsdaten (1000 Bilder)



Die Erstellung der Trainingsdaten erfolgt automatisiert mit Processing

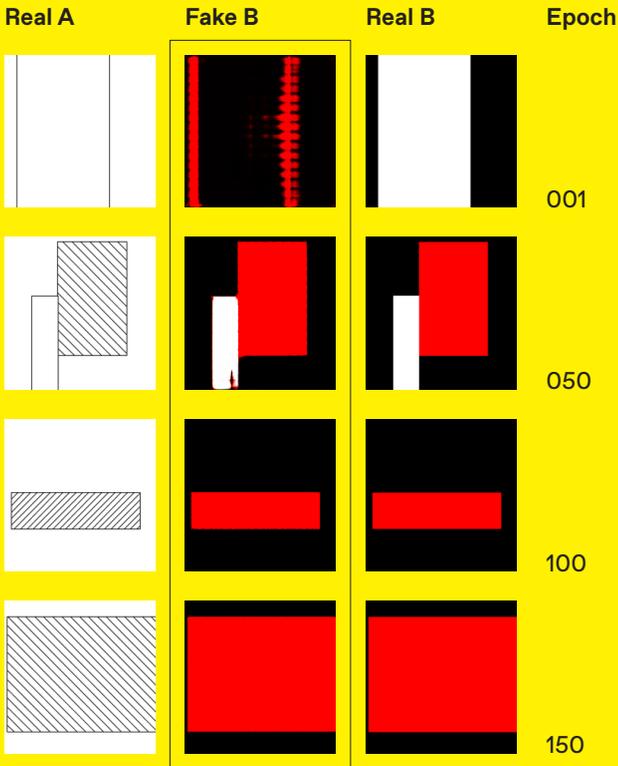
**A:**  
Mit Processing generierte A Seite. Schraffiertes Rechteck und weißes Rechteck.

**B:**  
Mit Processing generierte B Seite. Schraffiertes Rechteck wird rot, weißes Rechteck bleibt weiß. Hintergrund wird schwarz.

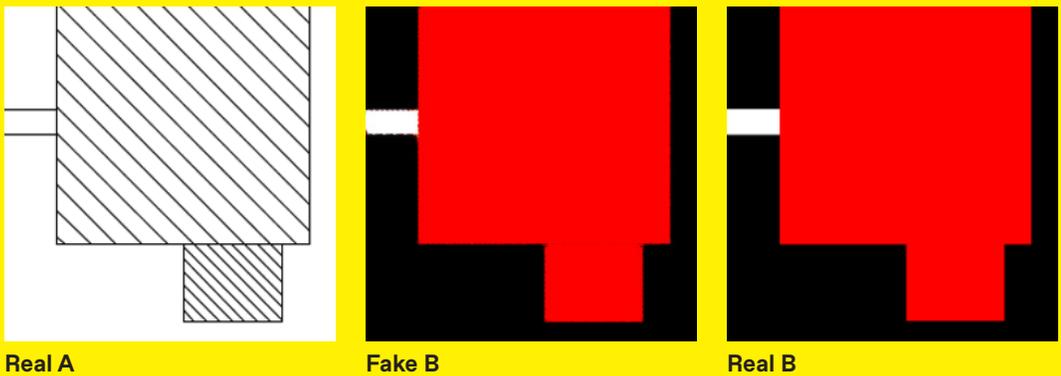


Trainingsprozess des Pix2Pix Modells

Epochs: 200

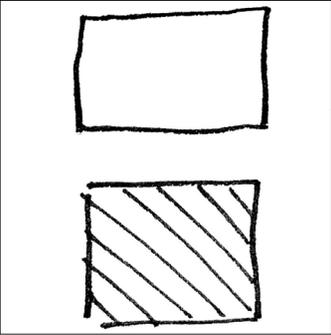


Epoch: 200

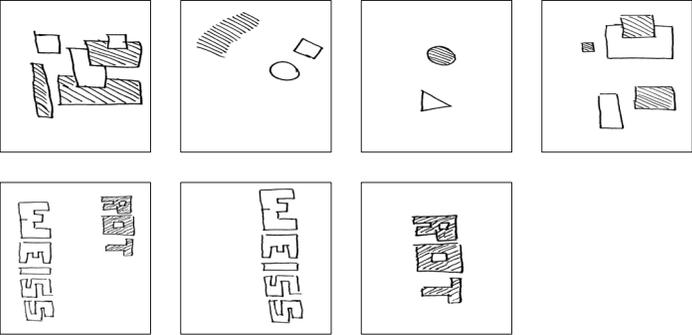


# rotes und weißes Rechteck

Kreation der Testdaten



A:  
Gezeichnetes Inputbild

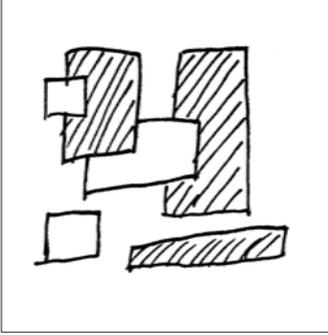


rotes und weißes  
Rechteck  
Pix2Pix Modell

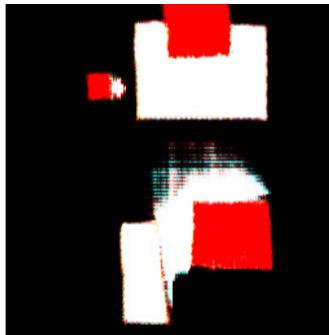
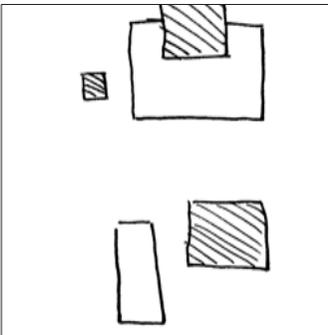
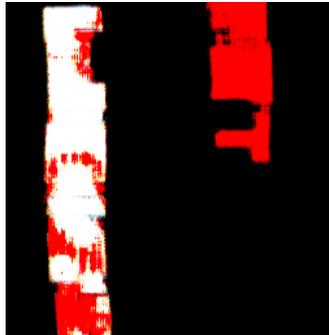
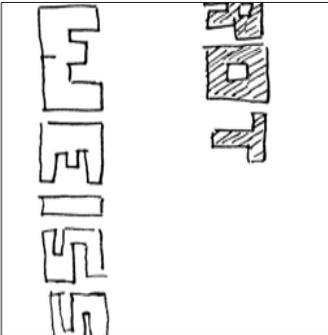
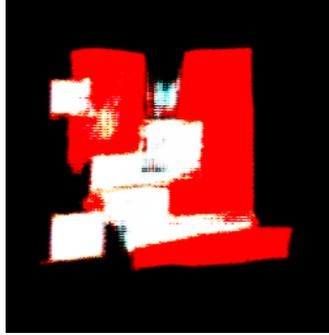
Training auf  
vorheriger Seite  
←

## Ergebnisse

Input

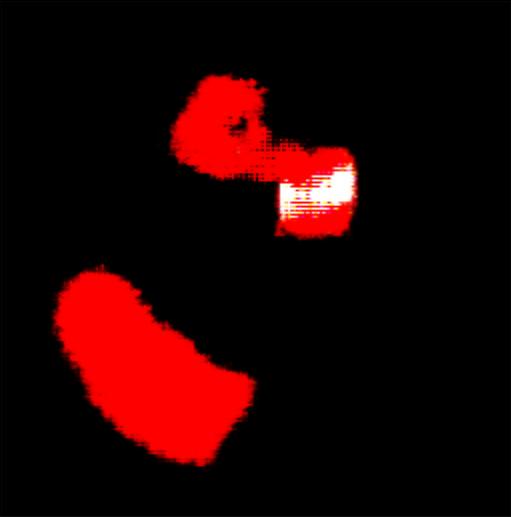
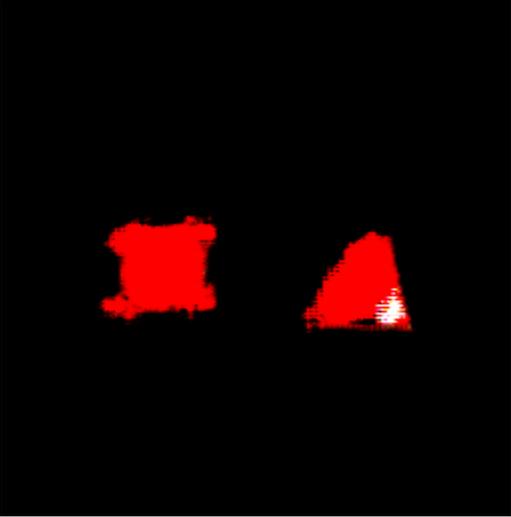


Output



# rotes und weißes Rechteck

---



---

## Reflexion

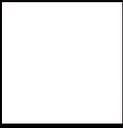
### Regeln festlegen

Bei diesem Modell ist erstaunlich, wie gut die Übersetzung funktioniert. Das Modell erkennt eigentlich immer die Rechtecke und färbt sie dem entsprechend richtig ein. Gibt man dem Modell Daten, die es nicht erwartet, wie zum Beispiel Schrift oder andere Formen, ist sich das Modell unsicher. Auch bei besonders vielen oder überlagernden Flächen wird das Modell etwas verunsichert.

Hier sehe ich einen großen Vorteil der Technik. Man kann sich ein Regelwerk an optischen Regeln überlegen, diese dem Modell beibringen, und dann mit jeglichem Input Testen.

### Anwendung

Dieses Konzept hat viel Anwendungspotenzial für Gestalter\*innen. Zum einen kann es dazu verwendet werden z.B. Zeichnungen in ihrer Farbigkeit automatisiert zu ändern. Es ist grundlegend in der Lage optische Codierungen zu erlernen. Man kann dadurch auch die Texturen einer Gestaltung schnell ändern.

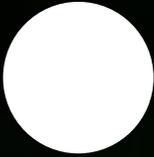


**Shape**

**2**



**Gradient**



---

shape2gradient

### Idee

Durch das vorherige Experiment überlegte ich, welchen Userinput der Nutzer / die Nutzerin noch geben könnte. Hier probierte ich aus, wie das Modell auf Grundformen (Rechteck, Dreieck, Kreis) reagiert, und ob es diese auch erlernen kann. Ich wollte, dass das Modell die Formen erkennt, und anhand dieser einen Verlauf erstellt.

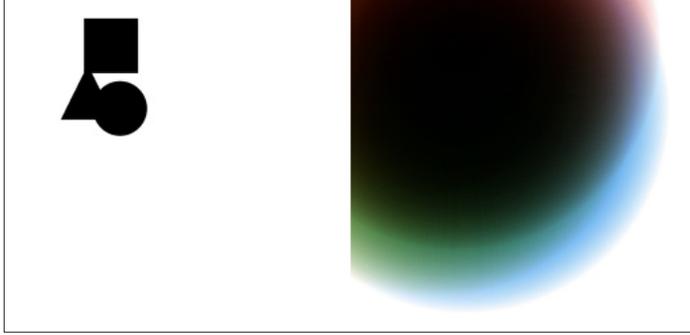
Dazu programmierte ich die Regeln der Übersetzung wieder in Processing, wodurch ich viele Trainingsdaten generieren konnte.

Außerdem wollte ich testen was passiert, wenn ich von den erwartbaren Inputdaten abweiche und Dinge zeige, die das Modell nicht erwartet.

Im weiteren versuchte ich das Modell mit Bewegtbild testen.

# shape2gradient

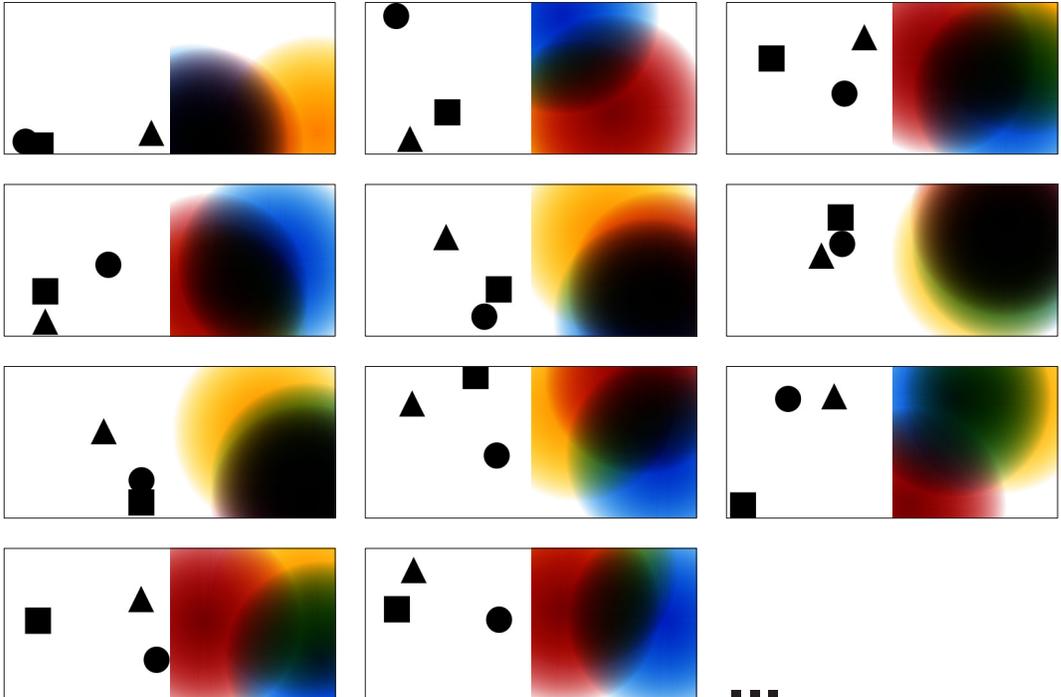
Trainingsdaten (1500 Bilder)



Die Erstellung der Trainingsdaten erfolgt automatisiert mit Processing

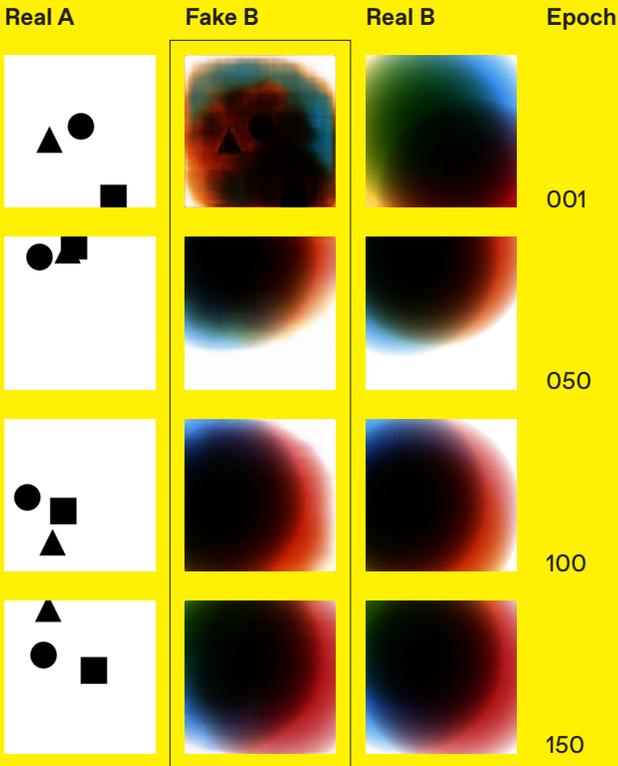
**A:**  
Es werden automatisch mit Processing drei Grundformen (Quadrat, Dreieck, Kreis) zufällig auf der Zeichenfläche platziert.

**B:**  
Jeder Form wird eine Farbe zugeordnet und ein radialer Verlauf von deckend zu transparent generiert. Der Mittelpunkt dieses Verlaufes ist an der gleichen Position wie die Form auf Bild A.

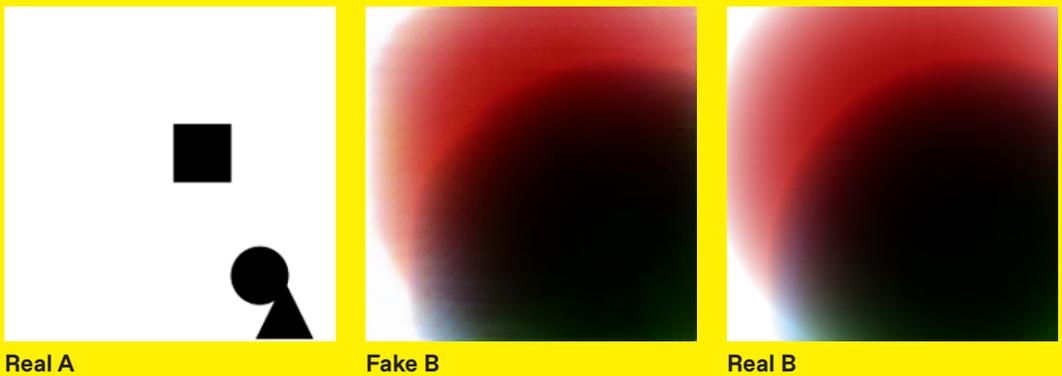


Trainingsprozess des Pix2Pix Modells

Epochs: 200

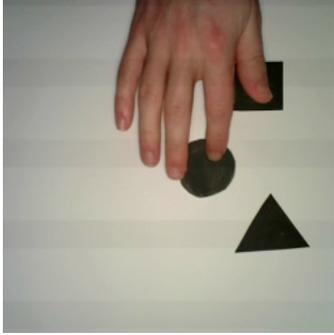


Epoch: 200



# shape2gradient

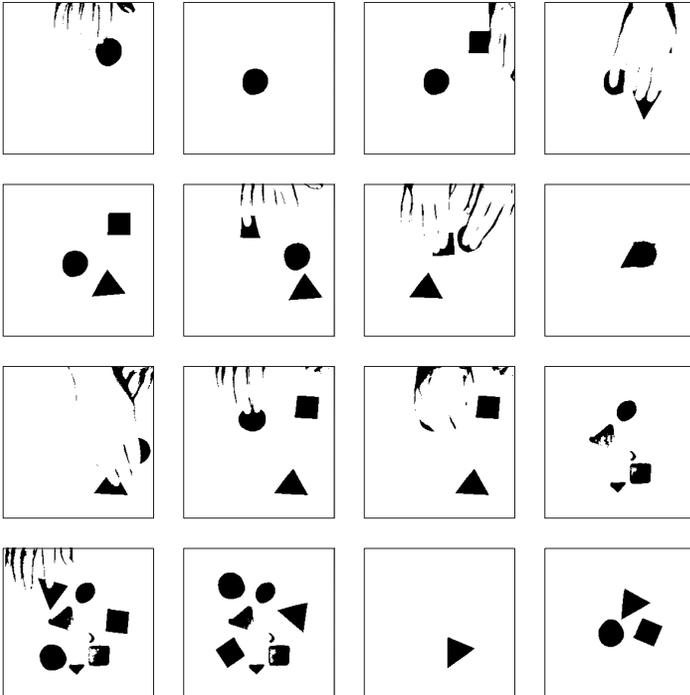
Kreation der Testdaten



**Vor Bearbeitung:**  Unbearbeitetes Inputbild aus der Webcam.



**Nach Bearbeitung:** Um einen grafischen Input, der den Trainingsdaten möglichst ähnlich ist zu generieren, wurde das Webcam Bild in ein Schwarz-Weiß Bild umgewandelt. Bei einem bestimmten Schwellenwert wird unterschieden zwischen absolutem Weiß und Schwarz.

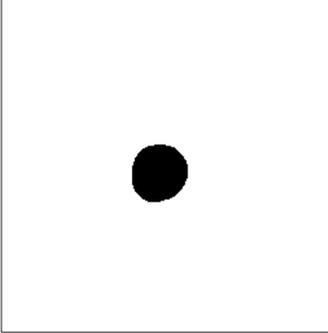


shape2gradient  
Pix2Pix Modell

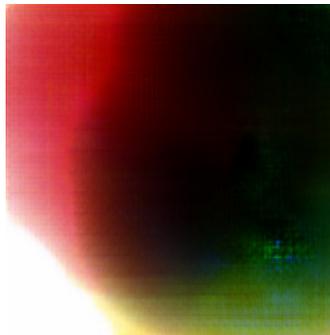
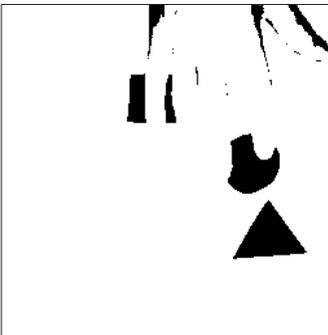
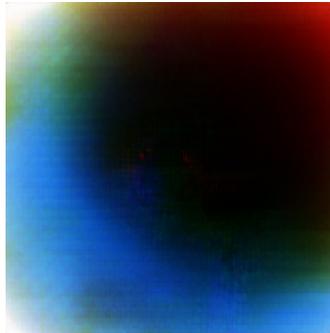
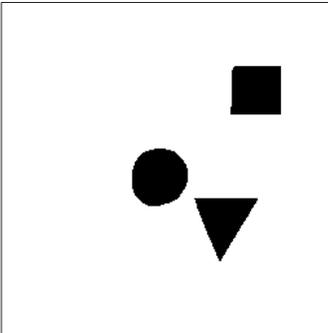
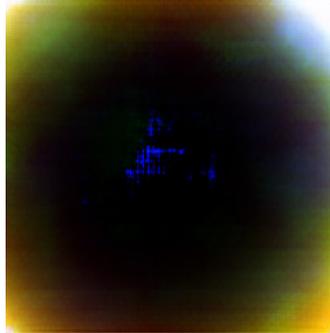
Training auf  
vorheriger Seite  
←

## Ergebnisse

Input



Output



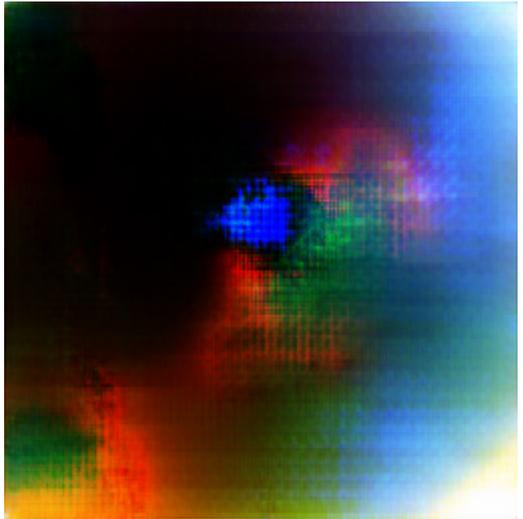
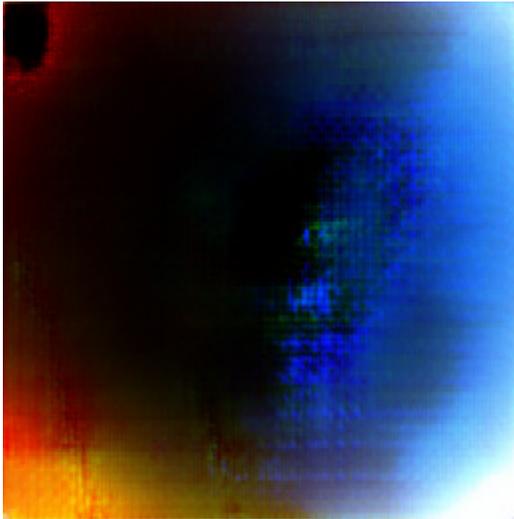
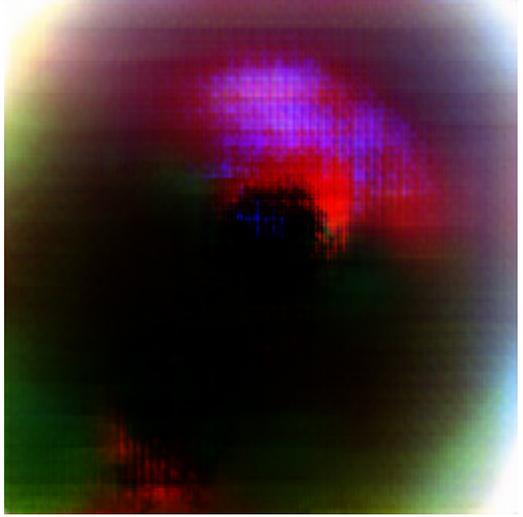
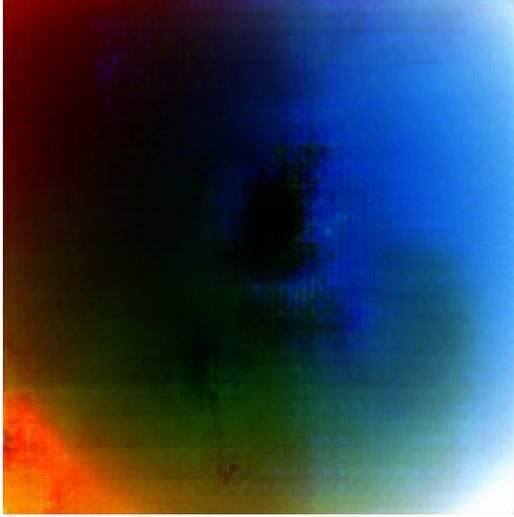
Video zum  
Ergebnis →



# shape2gradient

---

Results



---

## Reflexion

### Technik

Dieses Experiment zeigt, dass die Technik sehr gut funktioniert. Das Modell ist fast immer in der Lage, die Formen zu erkennen. Dazu ist jedoch zu sagen, dass der Output in Form eines Verlaufes nicht sehr genau ist, da man den Mittelpunkt nicht genau ermitteln kann.

Interessant sind auch die Ergebnisse auf der linken Seite, welche den Output zeigen wenn mehr als die 3 Zeichen in die Kamera gehalten werden, oder meine Hand noch im Bild ist.

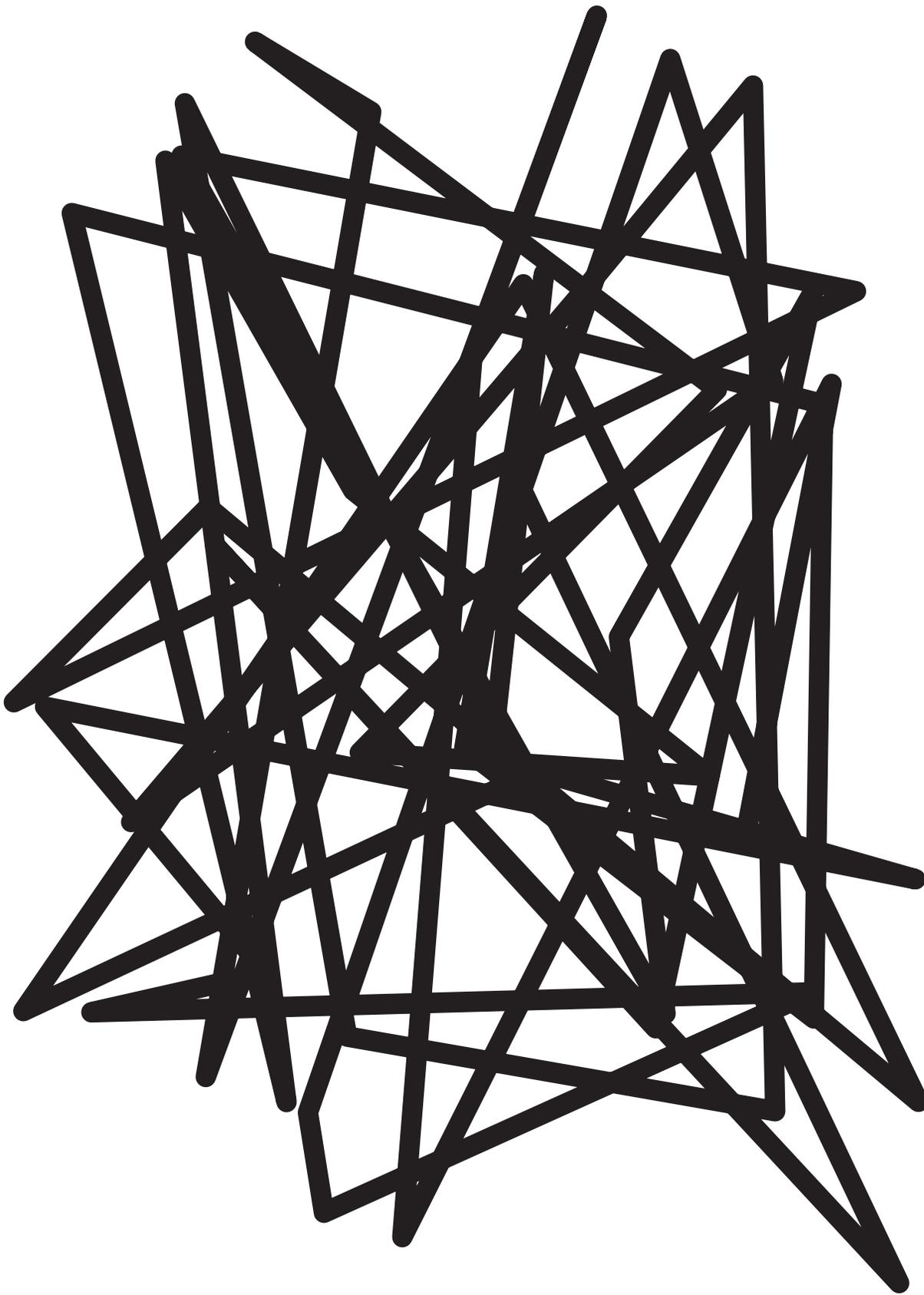
### Optik

Die Optik, vor allem bei den nicht erwartbaren Inputdaten, hat einen leichten Glitchcharakter. Man erkennt Muster aus vorherigen Experimenten wieder, an Stellen wo sich das Modell nicht sicher ist. Diese sich wiederholenden Störungen geben dem Bild eine besondere Struktur.

### Anwendung

Ich sehe in diesem Modell deutliche Vorteile in der Anwendung mit Userinput. Bei diesem Experiment habe ich auch das erste Mal den Prozess des Entstehens der Inputdaten mit aufgezeichnet (die Hand, die Formen verschiebt) und habe dadurch spannendere Ergebnisse erzielt. Dieses Experiment hat mir auch gezeigt, welche Interaktion mit der Maschine Spaß macht.

Die Anwendung für solch ein Modell könnte im künstlerischen Bereich liegen oder in der visuellen Suche für Gestalter\*innen, für eine neue Optik.



---

line2line

## Linien Erkennung

Dies ist ein weiterer Test dafür, welche Regeln sich gut für den Algorithmus eignen und in diesem Fall, welche auch nicht.

### Regelwerk

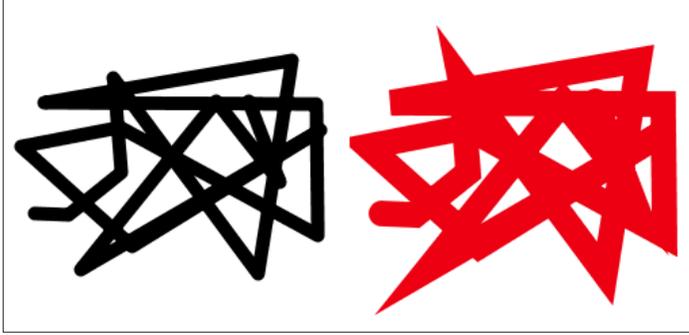
In diesem Experiment habe ich mit folgenden etwas komplexeren Regeln gearbeitet:

- ↳ Eine Linie wird automatisiert gezeichnet.
- ↳ Die Strichstärke der Linie ist abhängig von der Länge (länger = dicker)
- ↳ Ist die Linie abgerundet, ist sie grün.
- ↳ Ist die Linie Eckig, ist sie rot.
- ↳ Die Helligkeit der Linie ist auch abhängig von der Länge (länger = heller).

Auch hier habe ich wieder mit gemalten Userinput gearbeitet.

# line2line

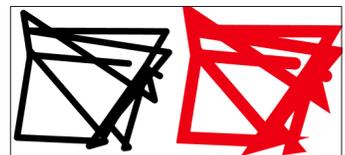
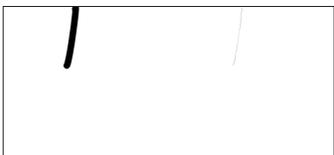
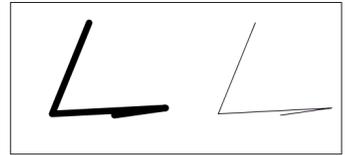
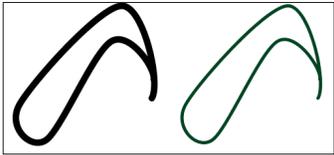
Trainingsdaten (1500 Bilder)



Die Trainingsdaten wurden automatisiert mit Processing generiert.

**A:**  
Eine zufällige Linie wird über die Zeichenfläche gezeichnet.

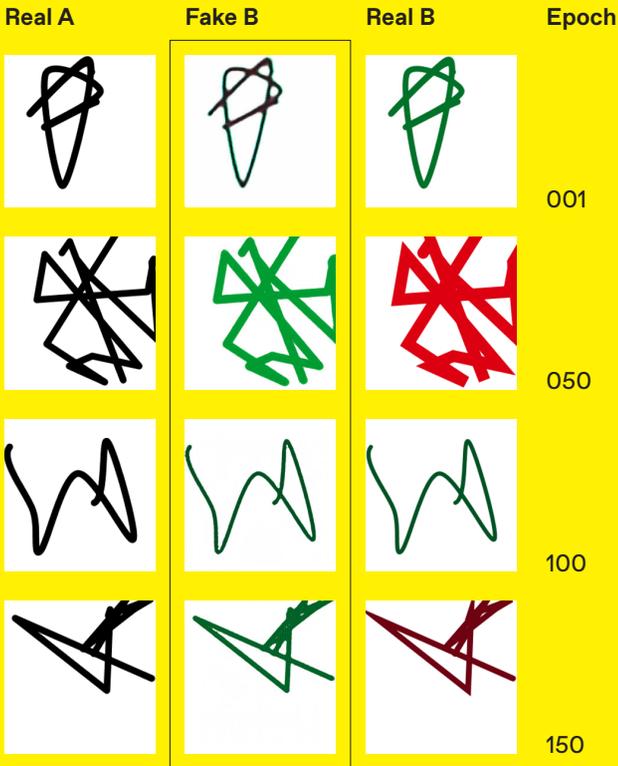
**B:**  
Eine nach dem Regelwerk passende Linie wird auf die Zeichenfläche gezeichnet.



...

Trainingsprozess des Pix2Pix Modells

Epochs: 200

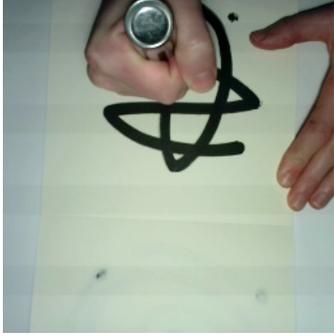


Epoch: 200

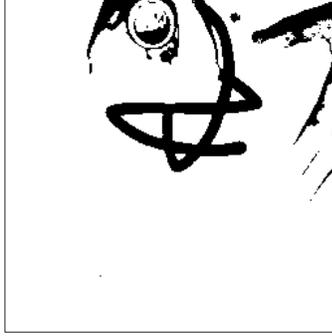


# line2line

Kreation der Testdaten



**Vor Bearbeitung:**  
Unbearbeitetes Inputbild aus der Webcam.



**Nach bearbeitung:**  
In absolute schwarze und weiße Pixel umgewandelte Bild



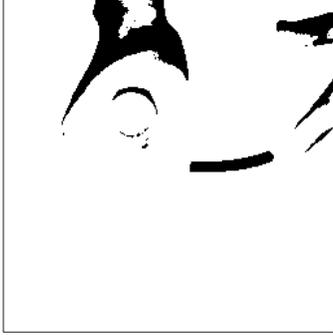
line2line  
Pix2Pix Modell

Training auf  
vorheriger Seite  
←

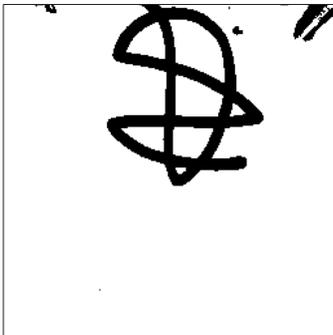
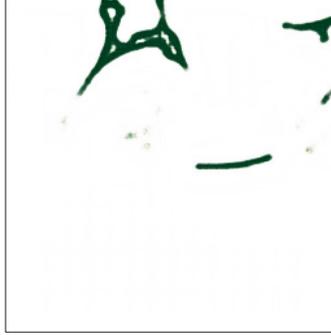
## Ergebnisse

---

Input



Output



Video zum  
Ergebnis →



# line2line

---



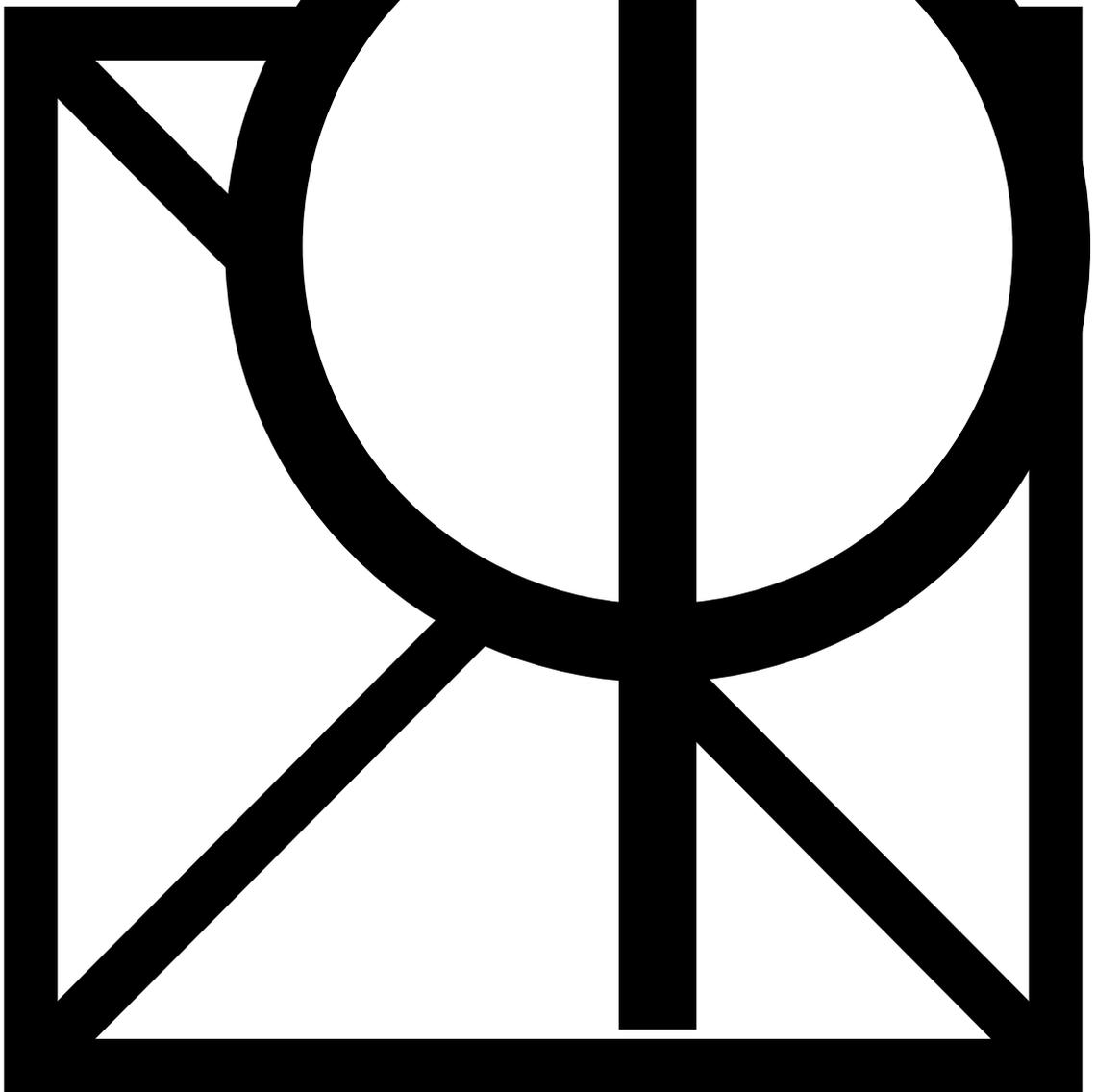
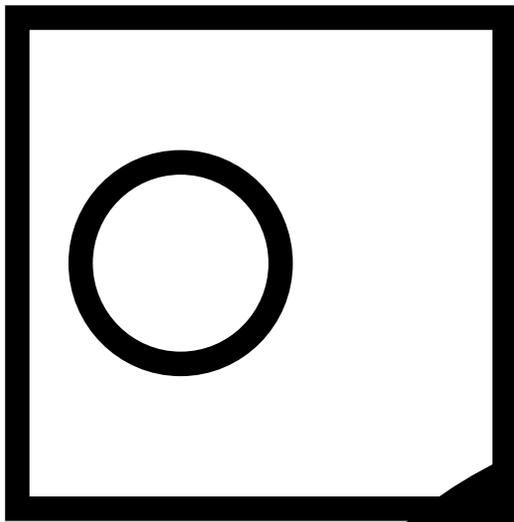
---

## Reflexion

### **Regelwerk funktioniert nicht**

Bei diesem Experiment habe ich gelernt, dass es Regeln gibt, die besser funktionieren als andere. Die Regel mit der Strichstärke funktioniert. Die Regel mit den geraden und abgerundeten Ecken hingegen nicht. Hierbei erkennt der Algorithmus immer eine abgerundete Linie und färbt sie deswegen immer grün ein.

Wieder mal habe ich gesehen, dass man nicht vorhersehen kann, welche Ergebnisse das Modell zeigt. Es ist jedoch auch sehr interessant zu sehen, was nicht funktioniert, um dies in den nächsten Versuchen zu ändern.



---

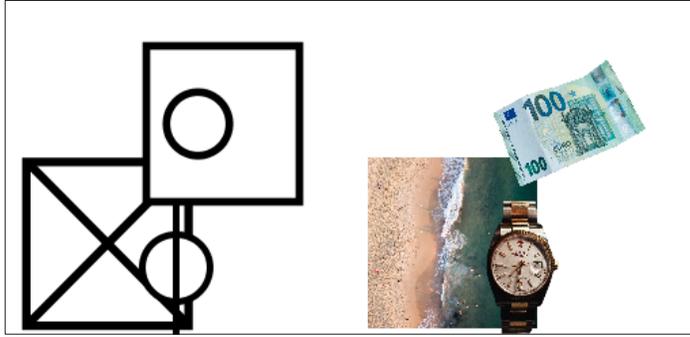
shape2status

## **Zeichen zu Bild**

Bei diesem Experiment habe ich versucht, Bilder mit einem Zeichen zu verknüpfen. Ich habe mich dabei an typischen Statussymbolen, die auf sozialen Medien sehr vertreten sind orientiert.

# shape2status

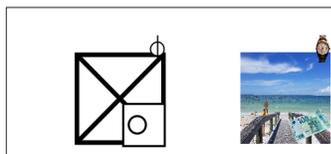
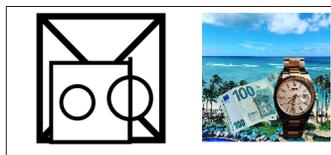
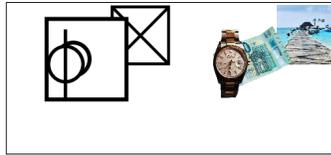
Trainingsdaten (1500 Bilder)



**A:**  
Für jede Fotografie wird ein Zeichen auf der Zeichenfläche platziert.

**B:**  
Die Bilder werden an der gleichen Position platziert.

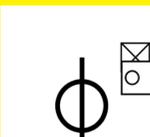
Die Trainingsdaten wurden automatisiert mit Processing generiert.



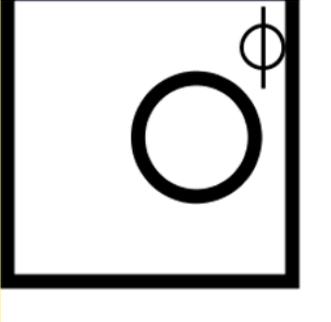
...

Trainingsprozess des Pix2Pix Modells

Epochs: 200

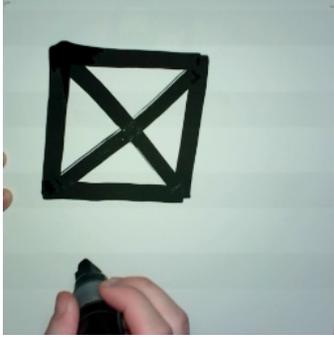
Real A	Fake B	Real B	Epoch
			001
			050
			100
			150

Epoch: 200

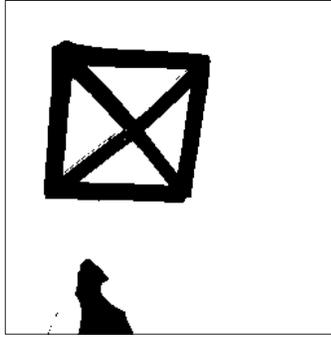
		
Real A	Fake B	Real B

# shape2status

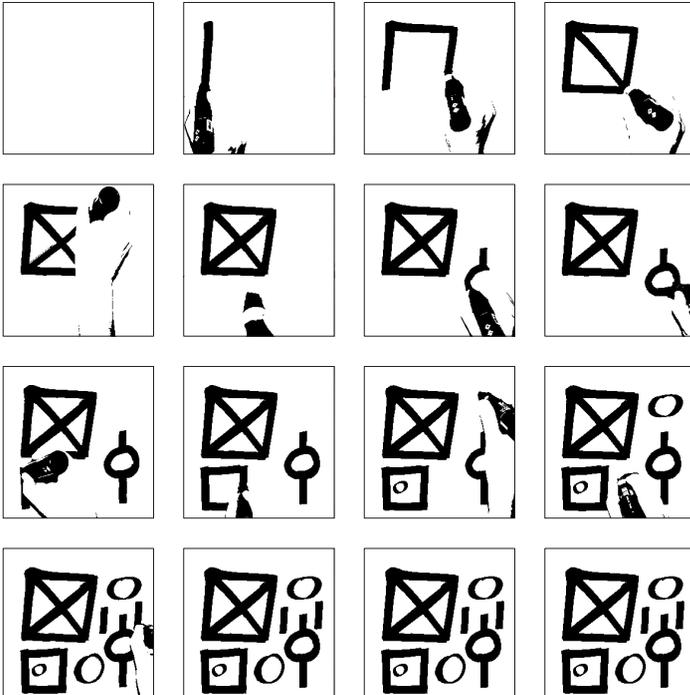
Kreation der Testdaten



**Vor Bearbeitung:**  
Unbearbeitetes Inputbild aus der Webcam.



**Nach bearbeitung:**  
In absolute schwarze und weiße Pixel umgewandelte Bild



shape2status  
Pix2Pix Modell

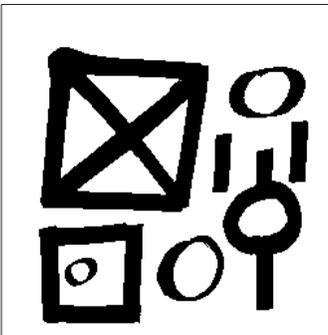
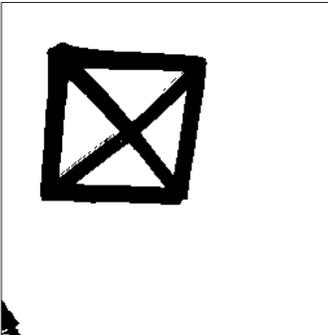
Training auf  
vorheriger Seite  
←

## Ergebnisse

Input



Output



Video zum  
Ergebnis →



# shape2status

---



---

Reflexion

## **Zeichnerischer Input**

Bei diesem Experiment habe ich den Input wieder zeichnerisch selber erstellt. Interessant hierbei ist, dass der Output stilistisch keinerlei Ähnlichkeit zu dem Input hat.

## **Anwendung**

Dieses Modell könnte dafür optimiert werden, um z.B. Layoutskizzen besser zu beurteilen. Wenn man die Codierung vorher mit dem Zeichner abstimmt, könnte man die Skizzen automatisiert und in Echtzeit in einem realistischen Stil anzeigen lassen. Sobald man schon weiß welche Bilder man verwendet, kann man diese auch schon in das Training des Modells integrieren und sie auch zeichnerisch platzieren.



# Wiederholende Form

---

## Wiederholende Form

### Mit der Maschine zeichnen

Das in diesem Experiment trainierte Modell, ist ein Vorläufer des für die Installation trainierten Modells.

Die Idee bei diesem Modell war es, das Inputbild mittels künstlicher Intelligenz zu erweitern. Ähnlich wie bei den letzten Experimenten nutzte ich wieder die Technik des Pix2Pix Modells, um ein Regelwerk auf weitere Inputdaten anzuwenden. Um die Ursprungsform entstehen der Form angepasste Konturen. Das heißt wenn ich einen Kreis in der Mitte habe, bilden sich weitere Kreise um diesen herum. Dadurch entsteht eine sendende und an Schallwellen erinnernder Form.

Für die Grundformen Kreis, Rechteck und Dreieck, ist es sehr einfach so einen Effekt zu programmieren. Sobald die Form jedoch komplexer ist, wird ein komplexer Algorithmus benötigt, um diese Outline zu generieren. Deswegen bot es sich hier an, das Pix2Pix Modell zu verwenden.

Ich trainierte die Maschine nicht nur auf das Erweitern der Form, sondern auch auf das Einfärben der verschiedenen Grundformen. Ich habe mich dabei an einer Zuordnung orientiert, die an das Bauhaus erinnert.  
(Quadrat - Rot, Dreieck - Gelb, Kreis - Blau)

# Wiederholende Form

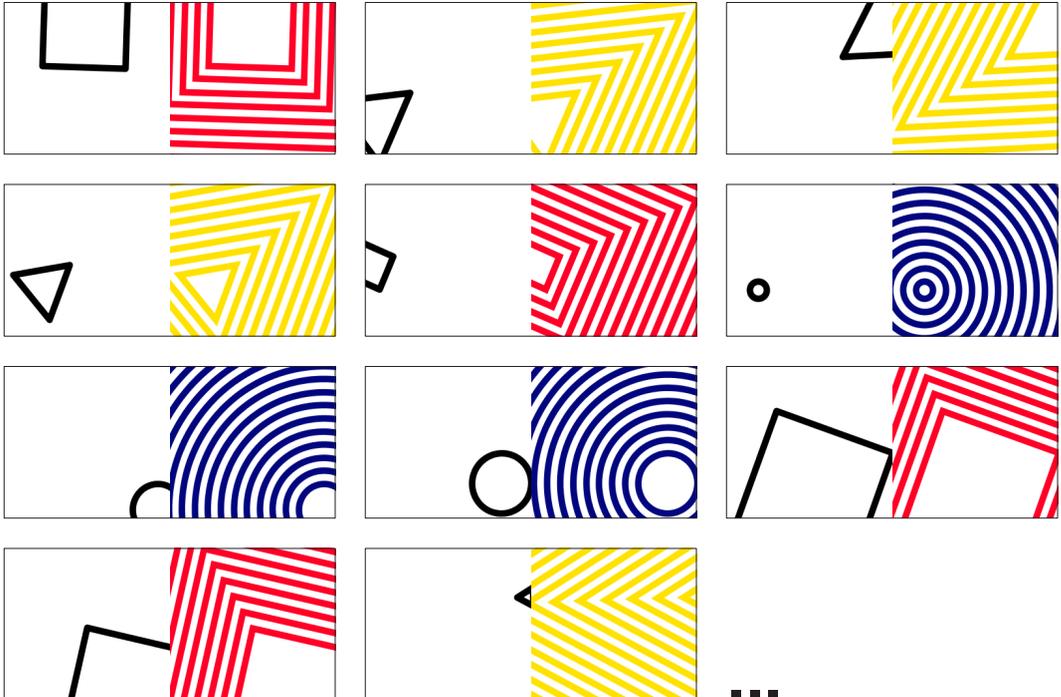
Trainingsdaten (1500 Bilder)



Die Trainingsdaten wurden automatisiert mit Processing generiert.

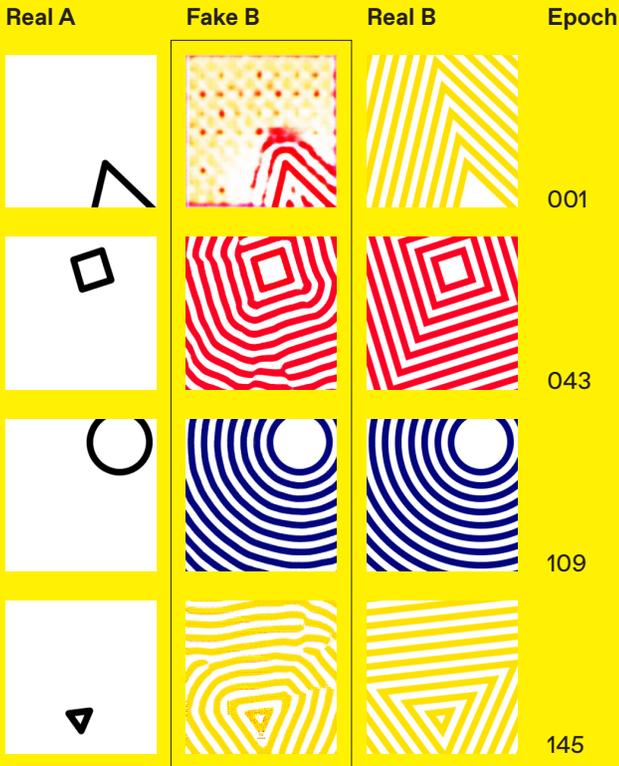
**A:**  
Auf der Zeichenfläche wird zufällig eine der drei Grundformen (Quadrat, Dreieck, Kreis) platziert.

**B:**  
Um die Grundformen wird eine sich immer wiederholende Kontur generiert.



Trainingsprozess des Pix2Pix Modells

Epochs: 200



Epoch: 199



# Wiederholende Form

Kreation der Testdaten



**A:**  
Die Webcam wird auf ein Blatt Papier gerichtet auf dem mit einem dicken schwarzen Stift gezeichnet wird.



**B:**  
Durch ein Programm wird nur der schwarze Strich auf der weißen Fläche angezeigt.



Wiederholende Form  
Pix2Pix Modell

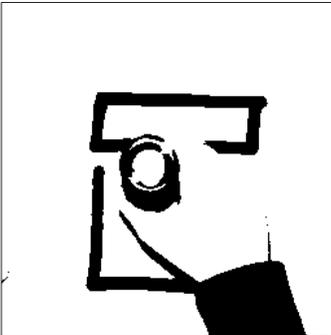
Training auf  
vorheriger Seite  
←

## Ergebnisse

Input



Output

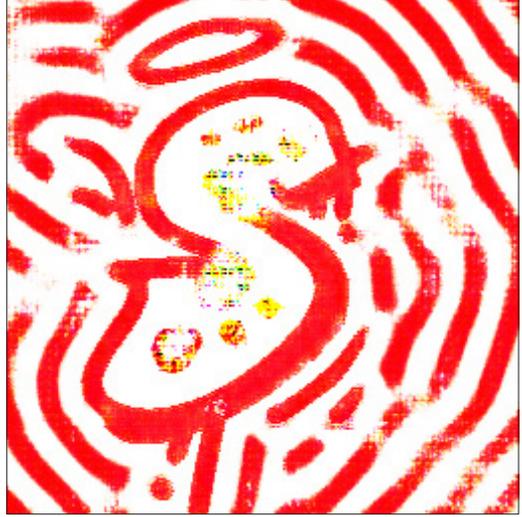


Video zum  
Ergebnis →



# Wiederholende Form

---



Video zum  
← Ergebnis

---

## Reflexion

### Live Daten

Bei diesem Experiment nutzte ich das erste Mal live Daten der Webcam. Aufgrund der hohen Rechenleistung, war aber nur möglich alle 10 Sekunden ein Bild zu generieren. Dies wollte ich für die Ausstellung noch etwas optimieren.

Außerdem ließ ich andere mein Modell anwenden, um zu sehen, was sie zeichnen und zu sehen wie sie mit dem Modell interagieren. Dabei wurde klar, dass nicht nur zeichnerische Inputdaten interessant sind. Auch etwas in die Kamera zu halten kann für spannende Ergebnisse sorgen.

Erstaunlich war, wie gut das Modell die Aufgabe erlernt hatte. In den meisten Fällen und auch bei komplexeren Formen, imitierte das Modell den vorgegebenen Stil sehr gut. Die Umrandungen sind zwar nicht perfekt symmetrisch und zerfließen an einigen Stellen. Jedoch ist der gesamte Eindruck sehr ähnlich zu dem, was man erwartet. Die vom System gemachten Fehler geben dem gesamten Bild eine natürliche Anmutung.

3

# Installation

---

Teil meiner Bachelorarbeit ist die Gestaltung einer Installation.

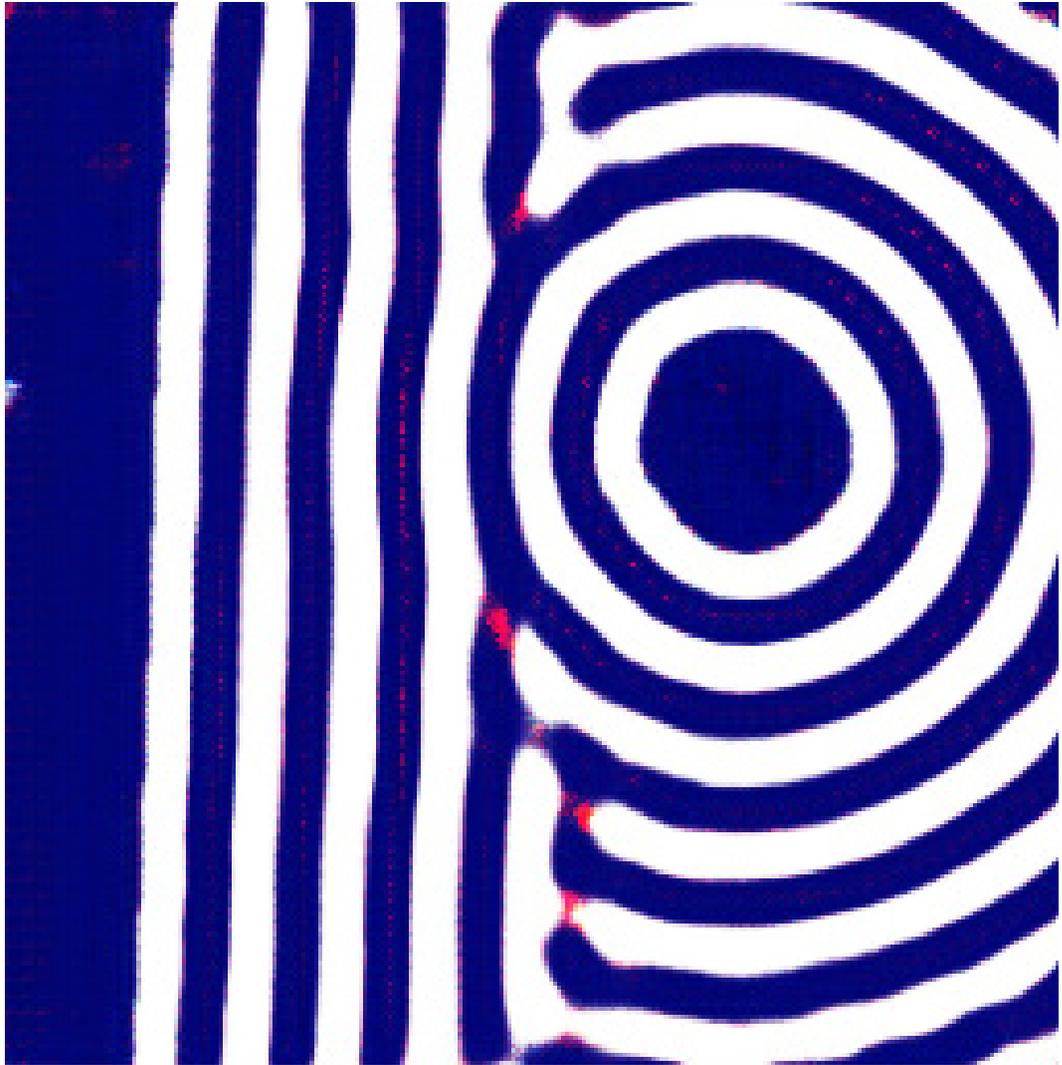
Die technische und gestalterische Ausarbeitung fußt auf den Ergebnissen aus den vorher abgebildeten Experimenten.

Die Installation ist ein prototypische Mensch-Maschinen-Interaktion. Der Computer reagiert mittels eines trainierten Pix2Pix Modells auf Live-Input.

Auf den folgenden Seiten ist die Idee, das Konzept, die Ausarbeitung der Technik, sowie eine Reflexion über die Installation aufgeführt.

# Konzept

---



## Idee

Wie kann der Mensch mit dem Computer interagieren? Die Entwicklung von Machine Learning Algorithmen lenkt ein neues Licht auf diese Frage. Durch künstliche neuronale Netze ist die Maschine in der Lage, viele verschiedene Inputs zu verarbeiten, Muster zu erkennen und anhand dieser etwas zu tun. Dadurch kann fast alles als Controller für den Computer genutzt werden.

Jetzt ist die Technik auch für Gestalter\*innen zugänglich. Durch Projekte wie z.B. Google Teachable Machine oder Wekinator ist das Erstellen von Machine Learning Modellen sehr einfach geworden.

Um diese Interaktion mit der Maschine geht es auch in meiner Installation, die als Teil meiner Bachelorarbeit auf der Diploma 2023 ausgestellt wird.

Ich will testen, wie ein Machine Learning Algorithmus zusammen mit menschlichem Input arbeiten kann. Dabei soll die Maschine als Co-Kreator mit dem Menschen zusammen etwas erschaffen.

## Umsetzung

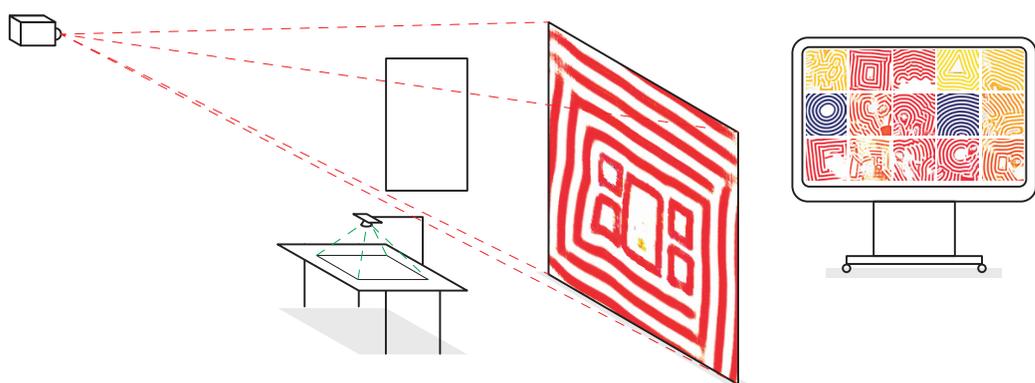
Besonders interessant finde ich eine Hands-On Lösung, bei dem die Nutzer\*innen mit der Maschine spielen können. Es soll ein Wechselspiel der Maschine mit den Nutzer\*innen entstehen.

Außerdem möchte ich die Daten sammeln und zu einer Collage zusammenstellen, welche über den Ausstellungszeitraum wächst.

Um dies zu gestalten, nutzte ich meine Erkenntnisse aus den zuvor erstellten Experimenten mit dem Pix2Pix Modell. Ich habe mich entschieden, mit dem zuletzt trainierten Modell „Wiederholende Form“ weiter zu arbeiten. Es ist in der Lage, visuellen Nutzerinput zu interpretieren und zu erweitern. Außerdem führt es zu spannenden Ergebnissen, die je nach Input eine große Varianz abbilden.

# Konzept

---



## Installations Aufbau

Der Installationsaufbau sieht wie folgt aus.

In der Mitte des Raumes steht eine Rückprojektionsleinwand. Diese teilt den Raum in zwei Bereiche. Der eine Bereich ist als der sendende Bereich zu verstehen und der andere als der empfangende Bereich.

Im sendenden Bereich ist ein Tisch aufgestellt, an dem eine Kamera befestigt ist. Diese filmt ein auf dem Tisch befestigtes Blatt Papier (DIN A2). Das aufgenommene Bild wird verarbeitet und als Inputbild an das trainierte Pix2Pix Modell geschickt. Dieses erweitert nun das Bild und projiziert es auf die Leinwand.

Auf der anderen Seite der Leinwand befindet sich der empfangende Bereich. Hier können die Besucher der Ausstellung stehen und das Ergebnis betrachten. Nebenbei steht noch ein großer Bildschirm, auf dem schon erstellte Output Bilder angezeigt werden. Wie die Bilder genau gespeichert werden beschreibe ich unter Punkt „Speicherung“.

Es gibt eine klare räumliche Trennung von der Person, die das Ergebnis produziert und den Betrachter\*innen auf der anderen Seite.

Auf der sendenden Seite hängt noch ein Plakat, mit dem Hinweis, dass alle Handlungen aufgezeichnet werden.

## Aussage

Die Installation soll die Beziehung von Sender und Empfänger von Sozialen Medien in eine räumliche Dimension übertragen. In der Online-Welt ist man örtlich von einander getrennt. Die Kommunikation erfolgt ebenfalls über eine Maschine. Vor allem bei bekannten Online Identitäten gibt es einen Sender und viele Empfänger.

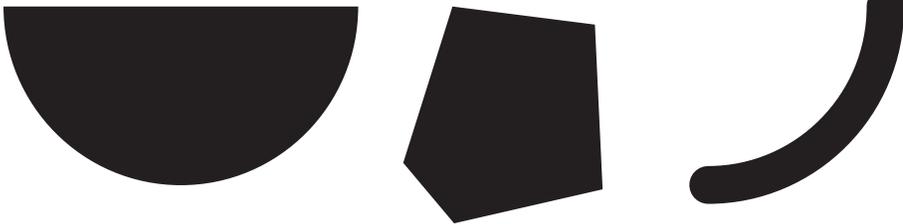
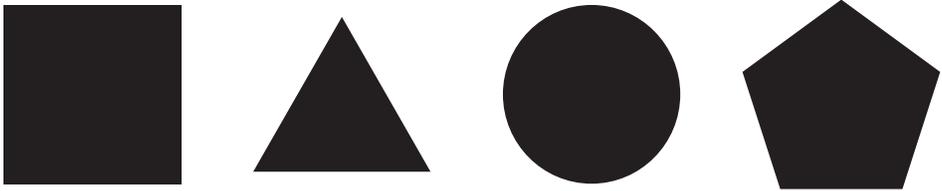
Das Plakat, mit dem Hinweis, dass alle Handlungen aufgezeichnet werden, soll eine leichte bedrohende Wirkung haben. Zum Einen soll es davon abhalten, dass anzügliche Dinge gezeichnet werden, zum Anderen soll es den User / die Userin auf das Thema Datenschutz aufmerksam machen. Es orientiert sich wieder an unserer Digitalen Identität, die überall aufgezeichnet und immer gespeichert wird.

**Achtung**  
Deine Handlungen werden  
aufgezeichnet!



# Konzept

---



## Mensch - Maschine Interaktion

In der Installation interagiert der Mensch mit der Maschine, indem er etwas in die Kamera hält oder zeichnet. Die Maschine interpretiert das Bild durch ein trainiertes Pix2Pix Modell und bildet das Ergebnis auf einer Projektionsfläche ab. Daraufhin wird der Mensch wieder aktiv und kann den Input variieren.

Den Besucher\*innen der Ausstellung stehen mehrere Möglichkeiten zur Verfügung, ein visuelles Bild zu gestalten. Es liegt eine Auswahl an schwarzen geometrischen Formen aus Holz aus. Diese kann der Nutzer / die Nutzerin auf die Zeichenfläche legen. Durch Kombinieren der Formen kann man viele verschiedene Ergebnisse erzeugen. Eine Auswahl der Formen sieht man links. Hierdurch wird der Nutzer / die Nutzerin dazu animiert, mit der Maschine zu interagieren. Die Hürde ist dabei so niedrig wie möglich.

Als weitere Möglichkeit kann der Nutzer / die Nutzerin auch mit einem dicken schwarzen Stift zeichnen. Dadurch kann man eigene Formen gestalten und sehen, wie die Maschine auf diese reagiert.

Als letzte Möglichkeit kann der Nutzer / die Nutzerin noch alle Dinge, die er bei sich hat, in die Kamera zeigen. Das kann alles sein: von der eigenen Hand über den Schlüssel aus der Hosentasche oder eine Postkarte, die man auf der Ausstellung gesammelt hat. Durch diese Methodik kommen die unterschiedlichsten Ergebnisse zustande. Hierdurch gibt der Nutzer / die Nutzerin aber auch am meisten von sich preis.

Diese verschiedenen Möglichkeiten werden dem Nutzer / der Nutzerin durch eine auf dem Tisch liegende Infokarte gezeigt.



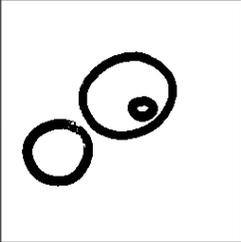
# Technik

---



Webcam Bild

In Schwarz-Weiß umwandeln



S-W Bild

Vortrainiertes Pix2Pix Model



Speichern und zur Übersicht hinzufügen



Bild wird auf dem Bildschirm abgebildet

Bild wird auf dem Beamer abgebildet

### Pix2Pix Modell

Das Pix2Pix Modell ist die Grundlage meiner technischen Lösung. Durch eine Kombination mit einer Streamlit-App werden die Daten automatisiert aktualisiert.

Das Abbilden und Speichern der Bilder wird mit Hilfe eines Processing Codes verwirklicht.

Auf Grund der hohen technischen Anforderungen, ist es zurzeit nur möglich, ein Bild alle vier Sekunden zu generieren.

Das Bild wird von einer montierten Webcam aufgezeichnet. Danach wird es in ein Schwarz-Weiß-Bild umgewandelt. Dieser Schritt ist nötig, damit das Pix2Pix Modell das Bild besser interpretieren kann. Durch das Pix2Pix Modell generiert sich ein neues Bild. Dieses wird nun auf der Rückprojektionsfläche abgebildet. Gleichzeitig wird es auf dem Computer gespeichert und zu der Übersicht auf dem Bildschirm hinzugefügt.

### Speicherung

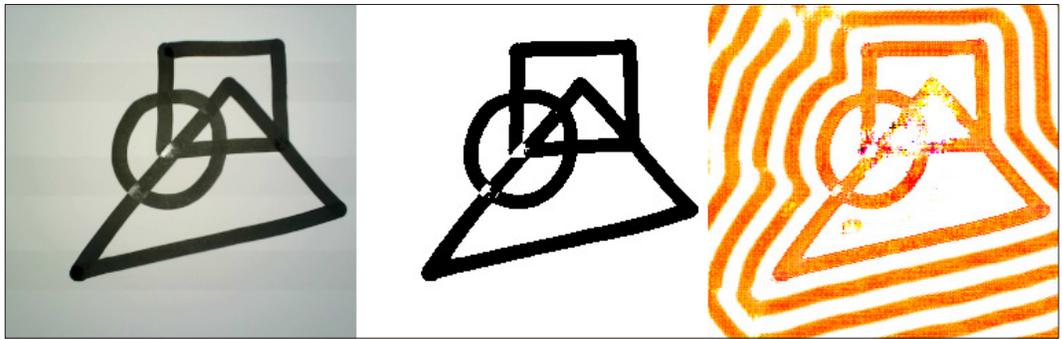
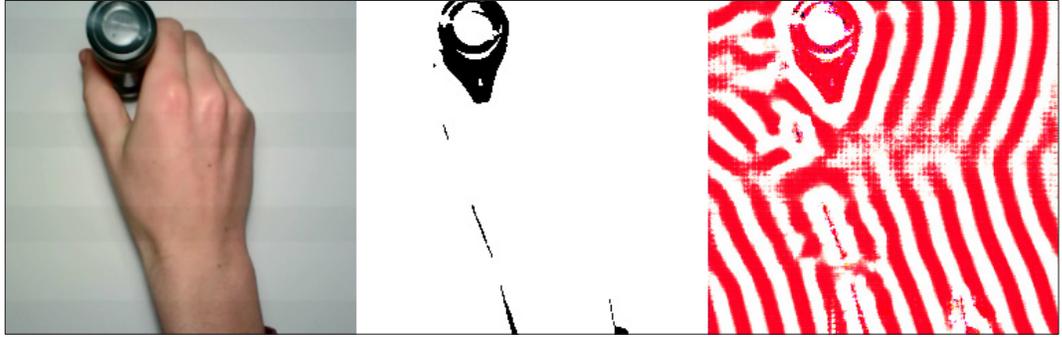
Die mit dem Modell generierten Bilder werden automatisch gespeichert. Auf Grundlage dieser bildet sich die Übersicht der generierten Ergebnisse. Diese ist in sechs Spalten und drei Zeilen unterteilt. In jeder Zelle wird ein Video mit niedriger Bildwiederholrate abgespielt. Die Anzahl der gespeicherten Bilder verteilt sich gleichmäßig über alle Zellen. Somit sind die Videos am Anfang der Ausstellung kürzer und am Ende länger. Zum Schluss ist ein Video aus allen Bildern der Ausstellung entstanden.

Die als letztes generierten Bilder sind demnach unten rechts zu sehen. Die Videos laufen vier mal durch, bevor neue Bilder geladen werden. Da das Laden der Bilder etwas Zeit in Anspruch nimmt, wird die Animation dadurch nicht ständig unterbrochen. Dennoch werden die zuletzt gezeichneten Bilder schnell bei den Resultaten angezeigt.



# Tests

---



Screencapture von  
der ersten Live-Input  
← Streamlit App

## Testphase

Ein weiterer Teil meines Entwurfsprozesses war es die Modelle zu testen. Hierbei interessierte mich, welche Art der Interaktion funktioniert und welche nicht.

Der Bau eines ersten Prototypen war hierbei ein wichtiger Teil. Dieser funktionierte leider noch nicht mit Live-Input, wodurch der Nutzer / die Nutzerin immer etwas warten musste, bis das Ergebnis angezeigt wurde.

Das Ergebnis dieser frühen Testphase war es, dass die Interaktion mit einem Stift sehr viel Spaß gemacht hat. Man wird jedoch auch etwas abgeschreckt, wenn man nicht weiß, was man zeichnen soll.

In einer zweiten Testphase funktionierte der Live-Input und durch die vorgelegten Formen waren die Testpersonen auch sofort animiert, diese in die Kamera zu halten. Dabei versuchten einige Testpersonen auch andere Gegenstände in die Kamera zu halten, dies führte auch zu interessanten Ergebnissen. Dadurch erweiterte ich mein Konzept noch um die Möglichkeit, eigene Dinge in die Kamera zu halten.

Auf der linken Seite sieht man eine Auswahl an Bildern aus den erstellten Testvideos.

## Technik

Um die Technik zu testen, ließ ich das Programm über mehrere Nächte laufen. Dadurch konnte ich sicherstellen, dass es stabil über die gesamte Ausstellungsdauer läuft.

Ein Ergebnis aus diesem Test war, dass das Programm, welches die Übersicht zeigt, nur ca. 4000 Bilder laden kann. Ich optimierte den Code so, dass es nur die letzten 4000 Bilder lädt.

Ansonsten funktionierte die Technik erwartungsgemäß stabil.

# Reflexion

---

## Performance

Ein schwieriger Teil beim Gestalten der Installation war es, zu überlegen, wie der Mensch mit der Installation interagieren könnte. Wie kann man die Personen dazu animieren, mit der Maschine in Kontakt zu treten?

Man muss den gesamten dreidimensionalen Raum bedenken und dazu noch die zeitliche Dimension. Zudem ist es nicht sehr einfach, einen Installationsaufbau schnell zu testen.

Zu Beginn machte ich mir noch etwas Sorgen, weil die generierten Bilder mit einer Auflösung von 256×256 px nicht besonders scharf sind. Wenn sie jedoch groß von einem Beamer projiziert werden, wird die niedrige Auflösung zum gewollten Stilelement. Es verstärkt den experimentellen und prototypischen Charakter der Installation.

Die Installation sollte unter dem Thema Online Identität und digitales Selbstbild stehen. Durch den Aufbau und der Tatsache, dass die Nutzer\*innen wissen, dass alles aufgezeichnet wird erhoffe, ich mir einen reflektierenden Gedanken der Besucher zu diesem Thema.

4

# Dokumentation

---

Die Dokumentation der Bachelorarbeit ist ein großer Teil dieses Projekts.

Im Folgenden gehe ich auf die Konzeption und technische Umsetzung dieser Dokumentation ein.



## Experimente

In der Dokumentation sind alle Experimente gesammelt, die ich über das Semester durchgeführt habe. Dadurch hat man einen Überblick über die im Semester trainierten Modelle.

Die Experimente sind systematisch und chronologisch aufgeführt. Es ist nicht möglich, alle Trainingsdaten und Ergebnisse abzubilden. Daher zeige ich lediglich einen zufälligen Auszug aus den Trainingsdaten und eine von mir getroffene Auswahl an Resultaten. Die Resultate wurden hauptsächlich anhand ihrer optischen Qualität gewählt.

Außerdem wird das Experiment immer von einem erklärenden Einleitungstext und einer abschließenden Reflexion umklammert.

## Installation

Der Teil über die Installation soll mein Konzept und Entwurf beschreiben. Die Installation soll bei der Diploma 2023 ausgestellt werden.

Die Erklärung des Konzepts wird mit Illustrationen und Grafiken unterstützt. Dies hilft zur räumlichen Vorstellung des Aufbaus und der Technik.

# Programmiertes Layout

---

```
#includepath "~/Documents/;%USERPROFILE%Documents";
#include "basiljs/bundle/basil.js";

var checkpoints = 'D:/Codes/pytorch-CycleGAN-and-pix2pix/check-
points/zoomMountainsPix2Pix/web/images/';
var epochs = [1, 50, 100, 150, 200];

function epochName(path, j){
  return path + 'epoch' + b.nf(epochs[j], 3);
}

function setup(){

}

function draw(){
  b.clear(b.page());

  b.units(b.MM);
  b.canvasMode(b.PAGE);

  //Gelb
  b.fill(0, 0, 100, 0);
  b.noStroke();
  b.rect(0, 23.533, 163, 212);

  //Linie
  b.noFill();
  b.stroke(0);
  b.line(0, 23.533, 163, 23.533);

  //kleiner text
  b.fill(0);
  b.noStroke();
  b.textFont('Favorit', 'Regular');
  b.textSize(9);
  var t1 = b.text('Trainingsprozess des Pix2Pix Models', 16, 23.533,
68, 4);
  applyGeneralTextStuff (t1);
  var t2 = b.text('Epochs: 200', 88, 23.533, 32, 4);
  applyGeneralTextStuff (t2);

  //Tabelle
  b.textFont('Favorit', 'Bold');

  var rA = b.text('Real A', 16, 39.533, 20, 4);
  applyGeneralTextStuff (rA);
  ...
}
```

**basil.js**

Um die vielen Experimente nicht per Hand layouts zu müssen, arbeitete ich mit basil.js.

basil.js ist eine JavaScript Library, die Befehle und Funktionen aus Processing in InDesign importiert. Dadurch ist es möglich, sehr komfortabel Skripts für InDesign zu schreiben.

Für die Experimente nutzte ich die Technik, um die Trainingsdaten zufällig auszuwählen und zu platzieren. Die Checkpoints wurden auch automatisiert platziert, genauso wie die Testdaten.

Diese programmierte Automatisierung ersparte mir besonders zum Ende der Bachelorarbeit viel Zeit. Es war jedoch immer abzuwägen, ob sich der Programmieraufwand rentiert oder es doch effizienter ist die Seiten manuell zu layouts. Der Vorteil der Programmierung ist, dass man bei Änderungen nur kleine Bereiche im Code ändern muss und sich alles neu generiert. Man muss sich jedoch vorher auch viele Gedanken um z.B. die Dateibenennung machen.

5

## **Design und KI**

Ich habe mich in diesem Semester sehr tief mit künstlicher Intelligenz im Designbereich auseinandergesetzt. In meinem Fazit gehe ich auf die von mir gestalteten Experimente ein, sowie auf das Thema im Allgemeinen. Ich schildere inwiefern ich die neue Entwicklung als positiv und negativ einordne.

Außerdem reflektiere ich meine Arbeit in Hinblick auf Optimierung, Technik und zukünftigen Einsatzmöglichkeiten für mich.

# Fazit

---

## KI als Helfer

Einen großen Vorteil von KI sehe ich in der Unterstützung von Gestalter\*innen in ausführenden Arbeiten. Zum Beispiel beim selektieren in Photoshop, so genanntes Inpainting bei Videos und dem Flattern von Texten. Das alles sind Aufgaben, die den Designprozess oft verlangsamen und verzögern. Mit KI kann man diese Aufgaben nun automatisiert lösen, wodurch Gestalter\*innen mehr Zeit für kreative Arbeit haben.

Wenn man die Befehle für die KI sogar mit Texteingabe lösen könnte, bräuchte man auch weniger technisches Verständnis für diese Aufgaben. Man könnte der Maschine einfach sagen: „Selektiere die Katze und setze sie auf blauen Hintergrund.“ Dadurch verschiebt sich der Aufgabenbereich von Gestalter\*innen sehr. Man müsste seine visuellen Gedanken in Worte übersetzen, die für die Maschine verständlich sind.

## Kreative KI

Dieses Bachelorprojekt hat mich in Hinsicht auf Kreativität und KI deutlich desillusioniert. Zu Beginn war ich noch fasziniert, wie gut Dall-E, Midjourney und Stable Diffusion „kreative“ Bilder generieren können. Schnell habe ich aber gemerkt, dass die wahre Kreativität nicht bei der Maschine liegt, sondern bei der Person, welche die Prompts eingibt. Genau so bei anderen generativen Modellen. Mit dem Optimieren eigener Modelle habe ich gelernt, wie abhängig das Ergebnis von den Daten ist, mit denen man das Modell trainiert. Genauso ist es nicht möglich, dass die Maschine von sich aus etwas Neues gestaltet. Die modernen Machine Learning Algorithmen sind so aufgebaut, dass sie Muster erkennen und reproduzieren können. Generative Modelle probieren also immer die Muster zu reproduzieren, die sie erlernt haben. Dadurch sind die Ergebnisse den Trainingsdaten aber sehr ähnlich und nicht als neu oder kreativ zu betiteln.

Durch Iteration bekommt man ein Ergebnis, das mehr von der Maschine gestaltet wurden. Dieses ist auch sehr stark von den Trainingsdaten abhängig sowie von der Systematik, die sich der Entwickler / die Entwicklerin überlegt hat.

## Neue Techniken

Ich habe in diesem Semester viele neue Techniken kennengelernt. Vor allem in Hinsicht auf meinen späteren Berufswunsch als Creative Technologist hat mich das sehr weitergebracht.

### Machine Learning

Ich habe gelernt, wie man ein vorprogrammiertes Machine Learning Netzwerk selber trainiert und testet. Hierbei habe ich Unterstützung von zwei Experten (Anik Jacobsen und Nils Freyer) aus der Data Science erhalten.

### Python

In diesem Semester habe ich das erste Mal mit der Programmiersprache Python gearbeitet. Da die meisten Machine Learning Algorithmen mit Python geschrieben sind war dies nötig.

### Facetracking

Für einige Experimente benötigte ich Facetracking Technik. Daher habe ich einen kleinen Exkurs in dieses Thema gemacht. Ich habe gelernt, wie man verschiedene Facetracking Technologien verwendet. Unter anderem habe ich mit OpenCV und clmTrackr gearbeitet.

### basil.js

Zur automatisierten Gestaltung der Dokumentation verwendete ich basil.js. Dieses Tool hat mir neue Möglichkeiten aufgezeigt, generativ mit InDesign zu arbeiten.

### Wekinator

Mit diesem Programm kann man ein künstliches neuronales Netz ohne Coding trainieren. Der OSC Input ermöglicht Einbindung von Creative Coding Software. In Zukunft werde ich dieses Tool noch öfter benutzen.

### ml5.js

Diese Library macht das Arbeiten mit Machine Learning Algorithmen in P5.js möglich. Es erweitert die Processing ähnliche Umgebung um ein sehr nützliches Feature.

### Processing

Ich habe in diesem Semester viele Codes mit Processing geschrieben. Zum einen dienten sie der Aufbereitung der Datensätze. Zum anderen auch zum Anzeigen und Weiterverarbeiten der Ergebnisse.

# Fazit

---

## Optimierung

Es gibt einige Punkte, die ich an meinem Entwurfsprozess für meine Bachelorarbeit anders gemacht hätte, wenn ich sie jetzt noch einmal beginnen würde.

Leider ist die Testphase, sowie die Planung der Installation etwas zu kurz gekommen. Dafür würde ich mir beim nächsten Mal mehr Zeit einplanen.

Zu Beginn habe ich mich auf das Thema Online Identität und Digitales Selbstbild festgelegt. Dies habe ich nach einiger Zeit bereut, da ich auch gerne etwas zu weiteren Themen gestaltet hätte. Da der Bachelor sehr experimentell ist, war unklar, in welche Richtung die Ergebnisse vorliegen werden.

Dieses Semester habe ich mich sehr stark mit der Technik aus einander gesetzt. Zukünftige Forschungsschwerpunkte sollten auch die konzeptionellen Aspekte stärker berücksichtigen.

Eine weitere Erkenntnis ist für mich ein besseres Zeitmanagement.

## Wie geht es weiter

In Zukunft wird Machine Learning wohl einen großen Einfluss auf die Designbranche haben. Deswegen werde ich mich auch weiter in diesem Bereich bilden und aktuelle Innovationen in diesem Bereich verfolgen.

Ob ich das in dieser Arbeit benutzte Pix2Pix Modell oder andere generative Modelle nutzen werde, ist fragwürdig. Ich sehe die Chancen von Machine Learning eher in der Mensch-Maschine-Interaktion und weniger in der eigenständigen Generierung von Bildern mit GANs und Ähnlichen. Dazu ist mir die Interpretationsfähigkeit zu fragwürdig.

Ich werde die Generierung von Bildern mit klassischen generativen Gestaltungsmethoden ausführen. Hierbei hat man eine größere Kontrolle über den erzeugten Output. Hierbei ist die Kombination mit Machine Learning natürlich auch möglich.

## **Der Bachelor**

Insgesamt hat mich die Beschäftigung mit einem generativen Machine Learning Modell inhaltlich vorangebracht. Ich habe viele neue Techniken erlernt und viele Erkenntnisse zum Thema KI gesammelt. Mein Plan war es immer schon, mit diesem Modernen Verfahren zu experimentieren.

Das Fehlen eines Teams hat mir hierbei Schwierigkeiten bereitet. Ich arbeite sehr gerne im Team, da man durch Aufgabenteilung viel mehr erreichen kann. Zudem fehlte mir die Reflektion durch Teammitglieder.

6



# Literaturverzeichnis

---

- 
- Albers, Marion, und Ingo Wolfgang Sarlet. Personality and data protection rights on the Internet: Brazilian and German approaches. Cham, Switzerland: Springer, [2022]. Print.
- Alter, Nora M. u. a. Hito Steyerl - I will survive: films and installations. 1. Auflage. Leipzig, Düsseldorf, Paris: Spector Books, Kunstsammlung Nordrhein-Westfalen, Centre Pompidou, 2020. Print.
- Doringe, Bogomir and Felderer, Brigitte. Faceless: Re-inventing Privacy Through Subversive Media Strategies, Berlin, Boston: De Gruyter, 2018.
- Engenhardt, Marc, und Sebastian Löwe. Design und künstliche Intelligenz: Theoretische und praktische Grundlagen der Gestaltung mit maschinell lernenden Systemen. Basel: Birkhäuser, [2022]. Print.
- KÄDE, Lisa, 2021. Kreative Maschinen und Urheberrecht : Die Machine Learning-Werkschöpfungskette vom Training über Modellschutz bis zu Computational Creativity. Baden -baden. ISBN 9783748912453
- Kaplan, Jerry. Künstliche Intelligenz. 1. Auflage. Frechen: mitp, 2017. Print.
- Levin, Golan, and Tega Brain. Code as creative medium: a handbook for computational art and design. MIT Press, 2021.
- López de Mántaras, Ramón. „Artificial Intelligence and the Arts: Toward Computational Creativity.“ In The Next Step. Exponential Life. Madrid: BBVA, 2016.
- Matthias, Michael. Echtzeit: die Kunst, intuitiv zu denken. Mainz: verlag hermann schmidt, 2021. Print.
- Miller, Arthur I. The artist in the machine: the world of AI-powered creativity. Mit Press, 2019.
- Pereira, Francisco Câmara. Creativity and artificial intelligence: a conceptual blending approach. Vol. 4. Walter de Gruyter, 2007.

# Online Ressourcen

---

Codes, Librarys & Papers

## **clmTracker References**

<https://www.auduno.com/clmtrackr/docs/reference.html>

## **Code für CycleGAN und Pix2Pix (PyTorch)**

<https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix>

## **Code für Pix2Pix**

<https://github.com/phillipi/pix2pix>

## **Code für Pix2Pix (tensorflow)**

<https://github.com/affinelayer/pix2pix-tensorflow>

## **Code für das Training von DC-GAN**

<https://github.com/ml5js/training-dcgan>

## **CycleGAN Paper**

<https://junyanz.github.io/CycleGAN/>

## **Library für Facetracking in JavaScript**

<https://github.com/auduno/clmtrackr>

## **Library für Machine Learning**

<https://github.com/yining1023/Machine-Learning-for-Physical-Computing>

## **ML4 Infoseite über Pix2Pix**

<https://ml4a.github.io/guides/Pix2Pix/>

## **ML5 Library für P5.js**

<https://ml5js.org/about>

## **Pix2Pix Paper**

<https://phillipi.github.io/pix2pix/>

## **Pix2Pix Paper**

<https://phillipi.github.io/pix2pix/>

## **Sammlung an vortrainierten Modellen**

<https://huggingface.co/>

## **StyleGAN**

<https://github.com/NVLabs/stylegan>

---

## Kurse und Tutorials

### **Kurs zu Machine Learning mit Physikal Computing**

<https://github.com/yining1023/Machine-Learning-for-Physical-Computing>

### **ml5.js Kurs von Daniel Shiffman**

[https://www.youtube.com/playlist?list=PLRqwx-V7Uu6YPSwT06y\\_AEYtqlwbeam3y](https://www.youtube.com/playlist?list=PLRqwx-V7Uu6YPSwT06y_AEYtqlwbeam3y)

### **Tutorial zum Trainieren von Pix2Pix**

<https://medium.com/@dongphilyoo/how-to-train-pix2pix-model-and-generating-on-the-web-with-ml5-js-87d879fb4224>

### **Videotutorial zum Trainieren von Pix2Pix**

[https://www.youtube.com/watch?v=TVCZLb1qe\\_0](https://www.youtube.com/watch?v=TVCZLb1qe_0)

### **Vorlesung über GANs**

<https://ml4a.github.io/classes/itp-F18/O6/#>

### **Wekinator Kurs**

<https://www.kadenze.com/courses/machine-learning-for-musicians-and-artists/info>

# Online Ressourcen

---

Vorträge, Konferenzen & Events

**DAI Conference**

<https://engenhartstudio.wixsite.com/daiconference2020>

**Encoding Cultures: Anna Ridler | The Possibilities of Machine Learning in a Creative Practise**

<https://zkm.de/de/media/video/encoding-cultures-anna-ridler-the-possibilities-of-machine-learning-in-a-creative-practise>

**Encoding Cultures: Mario Klingemann | Schöpfungswerkzeuge**

<https://zkm.de/de/media/video/encoding-cultures-mario-klingemann-schoepfungswerkzeuge>

**Event zu Thema Machine Learning in Design**

<https://www.correlationsforum.de/>

**Keynote von Rebecca Fiebrink über ML und Kunst**

<https://vimeo.com/287094397>

---

## Tools & Spielzeuge

### **Abraham AI**

<https://abraham.ai/>

### **AI Music Generator**

<https://soundraw.io/>

### **Artbreeder Collage**

<https://www.artbreeder.com/beta/collage>

### **Dall-E 2**

<https://openai.com/dall-e-2/>

### **Image-to-Image Demo**

<https://affinelayer.com/pixsrv/>

### **Stable Diffusion 2.0**

<https://stability.ai/blog/stable-diffusion-v2-release>

### **Teachable Machine**

<https://teachablemachine.withgoogle.com/>

### **Wekinator**

<http://www.wekinator.org/>

### **X Degrees of Separation**

[https://artsexperiments.withgoogle.com/xdegrees/8gHu5Z5RF4BsNg/BgHD\\_Fxb-V\\_K3A](https://artsexperiments.withgoogle.com/xdegrees/8gHu5Z5RF4BsNg/BgHD_Fxb-V_K3A)

# Online Ressourcen

---

## Artikel & Blogs

### **AI art is everywhere right now. Even experts don't know what it will mean**

<https://theconversation.com/ai-art-is-everywhere-right-now-even-experts-dont-know-what-it-will-mean-189800>

### **AI can now create any image in seconds, bringing wonder and danger**

<https://www.washingtonpost.com/technology/interactive/2022/artificial-intelligence-images-dall-e/>

### **AI Data Laundering: How Academic and Nonprofit Researchers Shield Tech Companies from Accountability**

<https://waxy.org/2022/09/ai-data-laundering-how-academic-and-nonprofit-researchers-shield-tech-companies-from-accountability/>

### **AI image generation is advancing at astronomical speeds. Can we still tell if a picture is fake?**

<https://theconversation.com/ai-image-generation-is-advancing-at-astronomical-speeds-can-we-still-tell-if-a-picture-is-fake-191674>

### **Artikelsammlung zu Autonomous Artists**

<https://medium.com/@genekogan#5918/artist-in-the-cloud-8384824a75c7>

### **DALL-E 2 vs \$10 Fiverr Commissions**

<https://simonberens.me/blog/dalle-2-vs-10-fiverr-commission>

### **Drowning in AI generated Garbage**

<https://ploum.net/2022-12-05-drowning-in-ai-generated-garbage.html>

### **EMANUEL GOLLOB, doing nothing with artificial intelligence**

<https://www.clotmag.com/interviews/emanuel-gollob-doing-nothing-with-artificial-intelligence>

### **Interview: Why Mastering Language Is So Difficult for AI**

<https://undark.org/2022/10/07/interview-why-mastering-language-is-so-difficult-for-ai/>

### **Kann künstliche Intelligenz kreativ sein?**

<https://www.change-magazin.de/de/kann-kuenstliche-intelligenz-kreativ-sein>

### **Linus Lee Is Living With AI**

<https://every.to/superorganizers/linus-lee-is-living-with-ai>

### **Mario Klingemann MEMORIES OF PASSERSBY I**

<https://medium.com/dipchain/mario-klingemann-memories-of-passersby-i-c73f72675743#:~:text=Memories%20of%20Passersby%20I%20is,faces%20created%20by%20a%20machine.>

---

## Weiter Ressourcen

### **Google Imagine Video**

<https://imagen.research.google/video/>

### **Have I Been Trained**

<https://haveibeen trained.com/>

### **KITeGG**

<https://gestaltung.ai/#/>

### **Make A Video (Meta AI)**

<https://makeavideo.studio/>

### **Neural Zoo**

<https://neuralzoo.com/>

### **Sammlung an Informationen zum Thema Facetracking**

<https://github.com/shiffman/Face-It>

### **The Follower**

<https://driesdepoorter.be/thefollower/>

### **Trending in the Multiverse**

<https://trendinginthemultiverse.com/>

# Abbildungsverzeichnis

---

## **Abb.1**

Fotos aus der Datenbank standardleaves

Quelle: <https://flavia.sourceforge.net/>

## **Abb.2**

Screenshot Maschine Prototyp

## **Abb.3**

BeReals von fremden Menschen

Quelle: BeReal Discovery Bereich

## **Abb.4**

BeReals von fremden Menschen

Quelle: BeReal Discovery Bereich

## **Abb.5**

Beispiel Labelmap von einer Fassade aus dem Pix2Pix Paper

Quelle: [https://phillipi.github.io/pix2pix/images/teaser\\_v3.png](https://phillipi.github.io/pix2pix/images/teaser_v3.png)

## **Abb.6**

Bilder von Gesichtern

Quelle: <https://unsplash.com/de/s/fotos/face>

## **Abb.7**

Bilder von unsplash.com

Quelle: <https://unsplash.com/de>

**Abb.8**

Szene Standbild aus dem Film „Call me by Your name“ von Luca Guadagnino

Quelle: <https://www.youtube.com/watch?v=Lj72RYJOzNs>

**Abb.9**

Bilder von unsplash.com

Quelle: <https://unsplash.com/de>

**Abb.10**

Bilder von unsplash.com

Quelle: <https://unsplash.com/de>

**Abb.11**

Bilder von unsplash.com

Quelle: <https://unsplash.com/de>

**Abb.12**

Fotos von Gesichtern

Quelle: <https://www.kaggle.com/datasets/ashwingupta3012/human-faces>

**Abb.13**

Bitmoji Avatare

Quelle: <https://www.kaggle.com/datasets/mostafamozafari/bitmoji-faces>

# Danksagung

---

## **Danke**

Ich danke allen, die mich bei der Bearbeitung dieser Bachelorarbeit unterstützt haben.

Ein besonderer Dank geht an Prof. Eva Vitting für die Betreuung dieses experimentellen Projektes.

Außerdem danke ich Anik Jacobsen und Nils Freyer für die technische Unterstützung. Ohne sie hätte ich das Projekt nicht umsetzen können.

Zusätzlich danke ich noch meiner Freundin, meiner Mutter und meinem Vater für das Korrekturlesen des wissenschaftlichen Teils.

Ich möchte außerdem noch allen Nutzer\*innen danken, die auf der Diploma mit meiner Installation interagieren und zu interessanten Ergebnissen beitragen.

# Erklärung zur Eigenständigkeit

---

## **Eidesstattliche Erklärung für die Abschlussarbeit**

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Abschlussarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Die Stellen der Arbeit, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind durch Angaben der Herkunft kenntlich gemacht. Dies gilt auch für Zeichnungen, Skizzen, Grafiken, Schemata, bildliche Darstellungen sowie für Quellen aus dem Internet.

Finn Jacobsen



**Designexperiments with  
Artificial Intelligence**

Bachelorarbeit  
Theoretischer Teil

Finn Jacobsen  
Wintersemester 22 / 23  
Fachhochschule Aachen  
Fachbereich Gestaltung

betreut von  
Prof. Eva Vitting

