

Im Handumdrehen zum Renoir

ZEICHNEN MIT NEURONALEN
NETZWERKEN - DIE KUNST FÜR JEDEN

Ziel der vorliegenden Arbeit ist es, eine Farbskizze in ein Kunstwerk umzuwandeln. Dies soll dabei möglichst wenig Zeit in Anspruch nehmen und als Plugin in einer Bildverarbeitungssoftware (wie GIMP) verwendet werden. Für die Umwandlung wird neben der Farbskizze ein Bild von einem Kunstwerk als Eingabe verwendet, in dessen Stil das Ergebnisbild gemalt sein soll. Für jeden neuen Kunststil wird dabei ein eigenes neuronales Netzwerk trainiert.

DER JUNGFORSCHER



Marcel Ullrich (1998)

Georg Cantor Gymnasium,
Halle (Saale)

Eingang der Arbeit:

31.5.2018

Arbeit angenommen:

11.12.2018



Im Handumdrehen zum Renoir

ZEICHNEN MIT NEURONALEN NETZWERKEN
– DIE KUNST FÜR JEDEN

1. Einleitung

1.1 Motivation und Zielstellung

Das Ziel dieses Projektes ist es, mithilfe einer Software komplexe Kunstwerke für jeden möglich zu machen. Innerhalb kurzer Zeit sollen mit minimalem Aufwand individuelle Kunstwerke kreiert werden, die den Stil von Renoir, Monet, Munch oder VanGogh aufnehmen. Vorlagen sind grobe Farbskizzen, wie sie von Kleinkindern gezeichnet werden können.

Die entwickelte Software soll in Grafikprogrammen wie GIMP oder MS-Paint implementiert werden können. Die Bedienung soll dabei so leicht sein, dass ein Grundschulkind sie verwenden kann. Kunst soll nicht mehr von Übung, Talent und Utensilien, sondern nur noch von der eigenen Kreativität abhängen.

Um die Kreativität nicht durch lange Rechen- und Konvertierungsvorgänge zu hemmen, setzte ich mir als weiteres Ziel, den Prozess der Umwandlung in Echtzeit zu realisieren. Ein Anstoß für diese Idee war der neuronale Stiltransfer von Gatys [1], bei dem Fotos in Gemälde umgewandelt werden. Zuerst war meine Idee, einfach Zeichnungen statt der Fotos einzusetzen. Da dies jedoch keine annehmbaren Resultate lieferte, versuchte ich neuronale Netzwerke für meine Anwendung abzuändern. Eine der größten Herausforderungen war dabei, das Netzwerk so zu trainieren, dass es nicht nur eine Umwandlung zwischen gleichwertigen Bildern vornimmt, sondern Farbvielfalt erzeugt und Details hinzufügt.

1.2 Verwandte Arbeiten

Meine Arbeit wurde stark von der Bild-zu-Bild-Abbildung mit *Conditio-*

nal Adversarial Networks inspiriert [2]. Entscheidende Unterschiede sind die Anwendung im künstlerischen Bereich als App und Plugin, sowie die auf ein Eingabebild reduzierte Menge an Trainingsdaten. Der Code wurde in Python mit der *tensorflow*-Bibliothek geschrieben [5].

2. Methoden

2.1 Aufbau

Als Modell für das neuronale Netzwerk wurde ein *Conditional Generative Adversarial Network* verwendet [2]. Dieses Modell besteht aus zwei Teilen. Zum einen aus einem Generator und zum anderen aus einem Diskriminator, die sich untereinander Eingabedaten und Feedback liefern, um die jeweiligen Aufgaben zu lernen. Dabei versucht jedes der beiden Teilnetzwerke, besser als das andere zu werden.

2.1.1 Generator

Das Generatorkennetzwerk bekommt die Farbskizze als Eingabe und versucht ein Gemälde (Computerkunstwerk) zu generieren. Dies geschieht mit einem *Encoder-Decoder*-Netzwerk (siehe Abb. 1), welches über *Convolutional Neural Network* (CNN) implementiert wurde. Ein *Encoder-Decoder*-Netzwerk nutzt ein CNN, um ein Bild auf einzelne Werte abzubilden und ein *DeConvolutional*-Netzwerk (DNN), um diese Werte wieder auf ein Bild abzubilden. Ein CNN geht Pixelgruppen durch und erstellt dadurch höherdimensionale aber kleinere Matrizen. Die Aktivierungsfunktion ist die *Restricted-linear-unit*-Funktion, kurz ReLu. Das DNN kehrt den Prozess des CNN um. Da die grobe Struktur der Farbskizze in dem Computerkunstwerk erscheinen soll – schließlich basiert das Computerkunstwerk auf der Farbskizze – werden die Eingabewerte der Farbskizze gespiegelt, indem sie als zusätzliche Werte im DNN verwendet werden. Da-

mit das Generatorknetzwerk nicht immer das gleiche Computerkunstwerk erzeugt, wird zusätzlich noch ein Rauschen als Eingabe verwendet.

2.1.2 Diskriminator

Das Diskriminatornetzwerk bekommt zwei Bilder als Eingabe: Die Ausgabe des Generators und die entsprechende Farbskizze als Eingabe und soll entscheiden, ob die Ausgabe ein Computerkunstwerk ist und zur Farbskizze passt. Um dies zu lernen, bekommt der Diskriminator beim Training das originale Gemälde und eine dazu passend angefertigte Maske als Eingabe, für die er „wahr“ liefern soll. Diese Maske wird für ein Originalbild per Hand erstellt. Der Diskriminator ist ein CNN, welches kleinere Quadrate abgleicht und die Ausgabe zu einem Gesamturteil kombiniert. Der verwendete Aufbau wird auch als *Markovian Generative Adversarial Network* bezeichnet [3]. Dies dient dazu, Ausgaben des Generators von den Originalen zu unterscheiden und so den Generator zu zwingen, immer bessere Computerkunstwerke aus Skizzen zu erstellen.

2.2 Implementierung und Umsetzung

2.2.1 Idee

Die Netzwerke werden am Anfang ein paar Stunden trainiert, indem das Kunstwerk und die manuell erstellten Masken sowohl skaliert als auch in Ausschnitten als Eingaben verwendet werden. Die Testdaten haben gezeigt, dass es wichtig ist, auch einzelne Abschnitte zu verwenden, da ansonsten *Overfitting* auftritt. Das Netzwerk kann demnach nur das Originalbild replizieren, lässt sich aber nicht auf andere Bilder übertragen, da es das Zielbild „auswendig“ lernt.

Wenn man einzelne Abschnitte verwendet, lernt es die einzelnen Details und Muster, auf die eine Farbfläche in einer speziellen Umgebung abgebildet werden soll. Ein gutes Beispiel ist das Bild von Renoir (siehe Abb. 2a). Es lernt, dass Grün Gras entspricht, aber auch, dass wenn Grün auf Weiß am Horizont trifft, Büsche gebildet werden sollen.

Zudem kann das Netzwerk am oberen Rand des blauen Bereiches Spiegelun-

gen im Wasser erzeugen, welche in vielen Fällen erstaunlich genau der Realität entsprechen.

Einen Trainingsdurchlauf sieht man in Abb. 2. Mit dem positiven Training (siehe Abb. 2a) wird der Diskriminator anhand von Ausschnitten aus dem Originalbild darauf trainiert dieses als „korrekt“ zu akzeptieren. Bei dem negativen Training (siehe Abb. 2b) erzeugt der Generator eine Ausgabe, auf die der Diskriminator trainiert wird, diese als falsch zu erkennen, um dem Generator beizubringen, Computerkunstwerke zu erstellen, welche dem Original vom Stil her möglichst ähnlich sind.

Schon zwei Stunden reichen aus, um in vergleichsweise hoher Qualität die Kunstwerke zu replizieren und neue erschaffen zu können. Dazu kommt, dass das Netzwerk nur einmal angelernet werden muss und danach immer wieder angewandt werden kann. Wenn es einmal einen Künstler gelernt hat, wozu ein Bild ausreicht, können die trainierten Gewichte des Netzwerkes in eine circa 200 Megabyte große Datei gespeichert werden. Nach dem Laden dieser Datei können die trainierten Gewichte auf

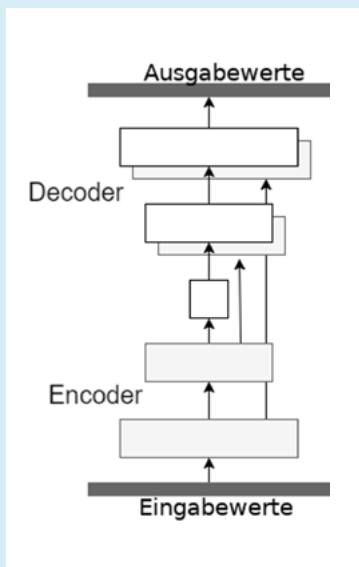
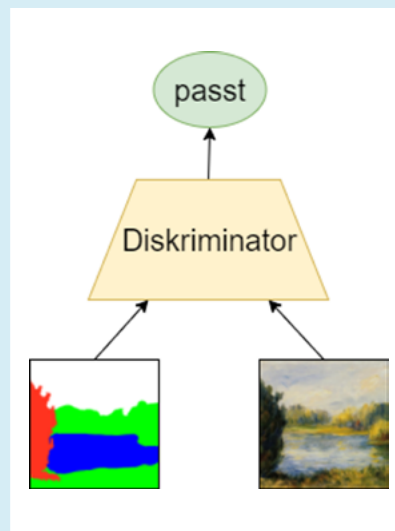
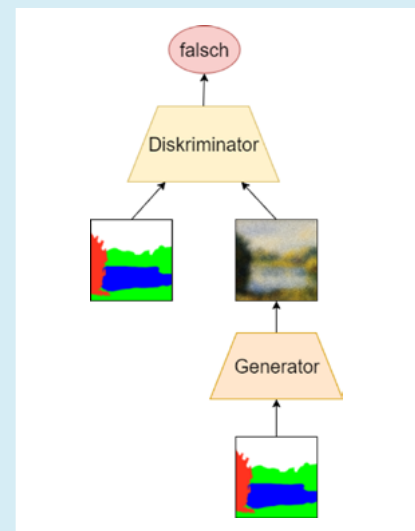


Abb. 1: Darstellung für Generatorarchitektur



a)

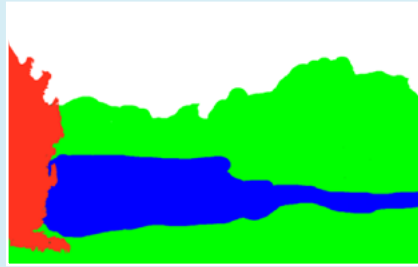
Abb. 2: a) Positives Training des Diskriminators mit einem Originalbild von Renoir, b) Anwendung und negatives Training



b)



a)



b)



c)

Abb. 3: a) Originalbild von Renoir, b) manuell erstellte Maske zum Bild, c) Überlagerung der Maske



neue Bilder angewandt oder weiter trainiert werden. Dafür benötigt die Anwendung zwischen einer und zehn Millisekunden.

2.2.2 Umsetzung

Da die Netzwerke eine sehr komplizierte Struktur haben, die jedoch in einzelne, schon bekannte Module aufgeteilt werden können, wurde die Bibliothek *Tensorflow* [5] verwendet. Diese beinhaltet bereits höhere Netzwerke wie CNN, und man kann sich eigene Netzwerkarchitekturen aufbauen.

Eine besondere Herausforderung war es, ein Modell zu trainieren, welches beliebige Computerkunstwerke im Stil eines Künstlers zeichnen kann, wenn nur ein einziges Beispiel zur Verfügung steht. Zum Vergleich sei angemerkt, dass schon Mengen mit unter tausend Trainingsdaten als wenige Eingaben bezeichnet werden.

Um den starken Mangel an Eingabedaten auszugleichen, wurden das Beispielbild (siehe Abb. 3a) und die dazu passende Beispielmaste (siehe Abb. 3b und Abb. 3c) in einer besonders hohen Auflösung erstellt und in Teilbilder zerlegt. Die Korrespondenz zwischen Maske und Originalbild sieht man in Abb. 3c, wo die Maske auf das Bild gelegt wurde. Dies geschieht mit großen Bereichen des Bildes (siehe Abb. 4a), aber auch mit kleineren Ausschnitten, welche dann skaliert werden (siehe Abb. 4b).

Zudem werden Ausschnitte von je 256×256 Pixel aus dem Bild gewählt, um die Details zu lernen und *Overfitting* zu verhindern, also das Auswendiglernen des Generators zu vermeiden. Damit das Modell Übergänge zwischen Bereichen lernt und keinen Rand mitten im Bild zeichnet, werden noch weitere Ausschnitte verwendet, welche die anderen überlappen, indem die Bereiche um die halbe Länge verschoben werden.

Es wurden im Programm acht Encoder- und Decoder-Netzwerkschichten verwendet, welche das Bild von anfangs 256×256 Pixel mit drei Farbkanälen auf einen 512-dimensionalen Vektor reduziert. Dies wird durch stückweise Halbierung der Seitenlängen erreicht. Der Decoder kehrt den Prozess um und erhält zusätzlich die Ausgaben des gleichgroßen Encoders als Eingabe, um die Struktur des Bildes zu bewahren.

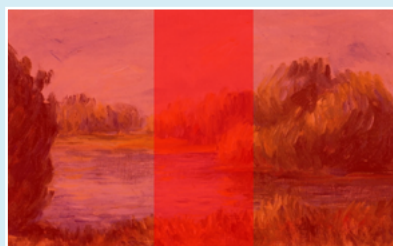
3. Ergebnisse

3.1 Computerkunstwerke nach unterschiedlicher Trainingsdauer

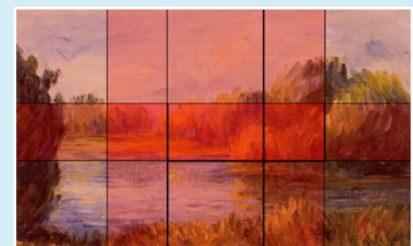
In Abb. 5 sieht man, wie das Netzwerk mit zunehmender Trainingsdauer lernt, die Computerkunstwerke zu erstellen. Zuerst kann es die Farben richtig übersetzen und man sieht schon den Anfang von Licht und Schatten. Danach kommen immer mehr Texturen hinzu, wie beispielsweise Pinselstriche, Reflexionen und Wellen.

Dabei sollte beachtet werden, dass, egal wie lange trainiert wird, die Umwandlung immer nur wenige Millisekunden benötigt.

Je länger das Netzwerk trainiert wird, desto genauer und detailreicher werden



a)



b)

Abb. 4: Zum Trainieren wird das Originalbild in a) große Teile und b) kleinere Ausschnitte unterteilt.



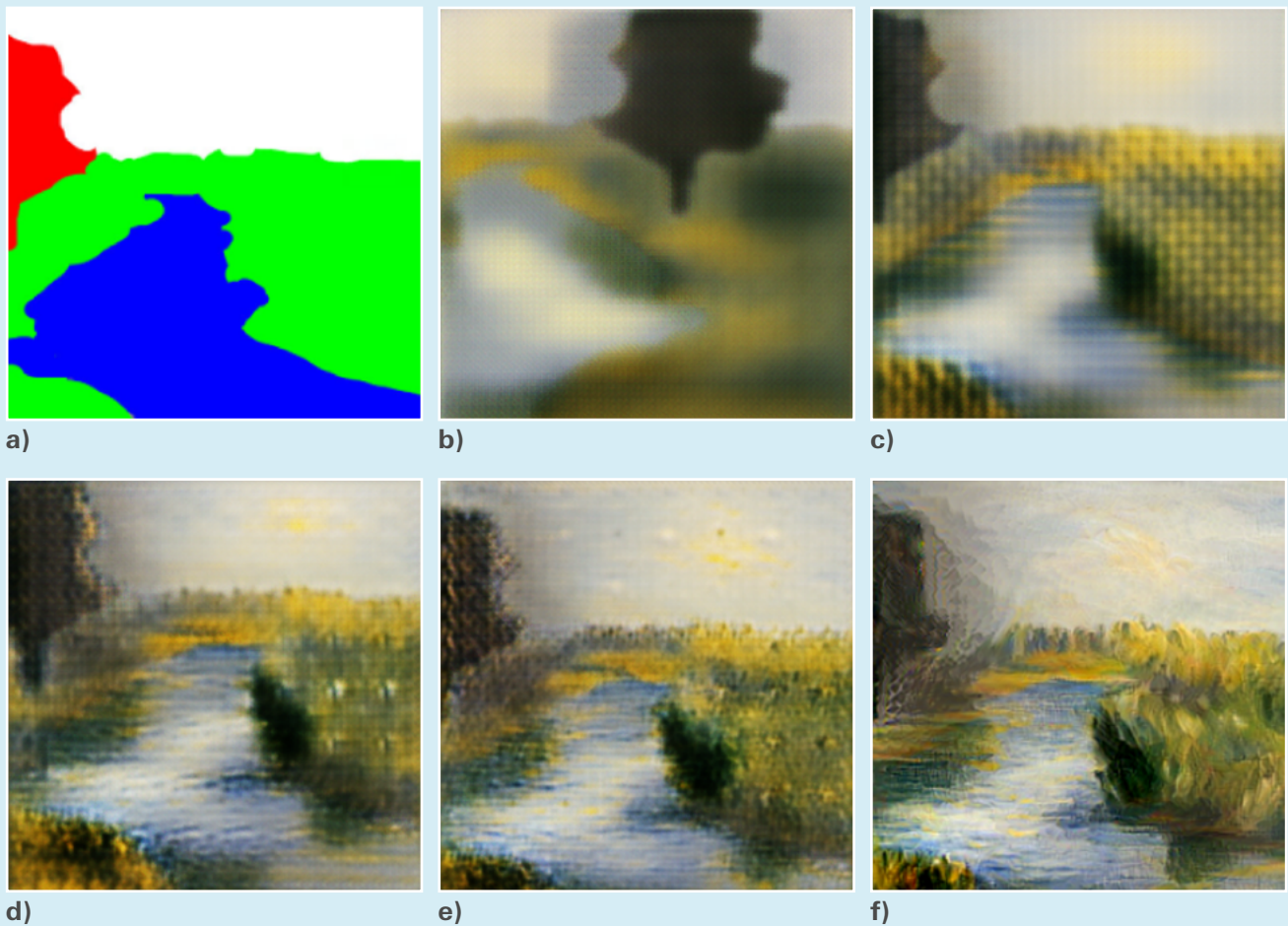


Abb. 5: Berechnung eines Computerkunstwerkes im Stil von Renoir a) vorgegebene Farbskizze, errechnetes Bild nach b) 30 s, c) 1 min, d) 10 min e) 20 min, f) 1 h Trainingsdauer

die Resultate. Man sollte jedoch nicht zu lange trainieren, da es sonst passieren kann, dass das Netzwerk das Originalbild auswendig lernt.

3.2 Anwendung mit verschiedenen Stilen

Bevor die Trainingsdaten für einen bestimmten Stil erstellt werden können,

müssen von Originalbildern Masken erstellt werden. [Abb. 6](#) und [Abb. 7](#) zeigen zwei Beispiele.

[Abb. 8](#) und [Abb. 9](#) zeigen jeweils die selbsterstellte Farbskizze und das berechnete Computerkunstwerk im entsprechenden Stil. Die Skizzen wurden im eigenen Editor erstellt, können aber auch mit Gimp erstellt werden.

3.3 Vergleich mit anderen Methoden

Ähnlich gute Resultate wie das hier vorgestellte Programm erzielt auch Neural Doodler [4] ([siehe Abb. 10](#)), wobei dieses Projekt keine Künstler lernen kann, sondern jedes Mal das Netzwerk für das aktuelle Beispiel neu lernen lassen muss. Daher ergeben sich signifikant

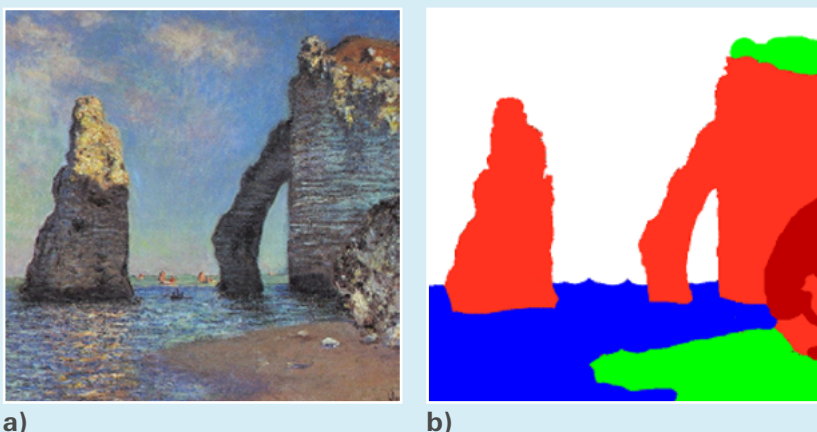


Abb. 6: a) Originalbild Étretat von Monet [8] und b) Maske zum Bild



a)



b)

Abb. 7: a) Sunflowers F456 Van Gogh [8] und b) Maske zum Bild



a)



b)

Abb. 8: a) erstellte Farbskizze
b) errechnetes Computerkunstwerk im Stil von Monet Étretat



a)



b)

Abb. 9: a) erstellte Farbskizze
b) errechnetes Computerkunstwerk im Stil von Van Gogh
Sunflowers F456



längere Laufzeiten von über vier Stunden pro Bild auf einem modernen Computer. Wenn das Netzwerk bei einem Bild besonders schlecht lernt, kann man mit Neural Doodler zusätzliche Eingabedaten zum Lernen erschaffen, da dieser Vorgang nur einmal ausgeführt werden muss und danach mit dem schnellen Programm gearbeitet werden kann.

Eine weitere Methode ist der *Neural Style Transfer* [1], welcher auf Objekterkennung und Texturerersetzung beruht. Diese Technik kann man mit geringem Qualitätsverlust in Echtzeit ausführen [3], jedoch kann sie nicht auf Farbskizzen angewandt werden, wie der Vergleich in Abb. 11 zeigt. In Abb. 11b sieht man, dass ein Ansatz besteht, das Gras mit detailreicherer Textur zu füllen, jedoch scheitert das Programm an der einheitlichen Fläche. Das Programm [7] schafft es schon besser das Gras und Wasser zu ersetzen (siehe Abb. 11c), jedoch fehlen kontextsensitive Details wie Reflexionen und Licht und Schatten. Diese Details schafft das hier entwickelte Programm, wie Abb. 11d zeigt.

4. Anwendungen

Es ist möglich, das entwickelte Programm auf einem Server laufen zu lassen, weil nur wenige Millisekunden benötigt werden, um die Farbskizze in ein Kunstwerk umzuwandeln. Über eine Webseite [11] kann man auf das Programm zugreifen und selbst Computerkunstwerke erstellen, ohne das Programm und die Datei mit den trainierten Gewichten des Netzwerkes für die unterschiedlichen Künstler zu haben (siehe [Abb. 12](#)).

Da die Umwandlung so schnell funktioniert, war es möglich, die Idee von *Neural Doodler* umzusetzen und ein Plugin für das frei verfügbare, pixelbasierte Grafikprogramm GIMP zu schreiben, mit dem man in Echtzeit eigene Skizzen umwandeln und das erzeugte Computer-

kunstwerk mit allen Mitteln der Bildbearbeitung bearbeiten kann. Dies hat den Vorteil, dass man noch mit anderen Werkzeugen das Computerkunstwerk direkt weiterbearbeiten kann. Die Einbindung erfolgt über einen Filter, sodass man die Umwandlung auf gewohntem Weg, wie beispielsweise mit einem Unschärfefilter, einfach vornehmen kann.

Eine weitere Anwendung ist eine im Rahmen dieser Arbeit entwickelte App, mit der man mit einem Touchpad die Skizze leicht zeichnen kann. Über ein Menü kann man dann mit einem Klick das Bild konvertieren (siehe Abb. 13). Hier besteht der Vorteil, dass man direkt mit Fingern und Stiften malen kann und so näher am originalen Entstehungsprozess eines Kunstwerks ist.

5. Ausblick

Obwohl der Algorithmus bereits recht schnell ist, lässt sich auch hier noch etwas verbessern. So wird dieser weiterentwickelt, um noch mehr Details zu erzeugen und noch besser mit hohen Auflösungen zurecht zu kommen. Dies kann durch längeres Training oder durch die Kombination mit anderen Methoden wie Supersampling erreicht werden. Zudem wird versucht, die gelernten Daten des Netzwerkes näher zu analysieren, um mehr über den Umwandlungsprozess zu erfahren. Beispielsweise kann so herausgefunden werden, wie das Netzwerk Reflexionen und Licht und Schatten in das Bild hinzufügen kann. Eine weitere Verbesserungsmöglichkeit ist es, die Künstlerbibliothek auf mehr Künstler und Kunststile zu erweitern. Und schließlich ist geplant, die App allgemein verfügbar zu machen.

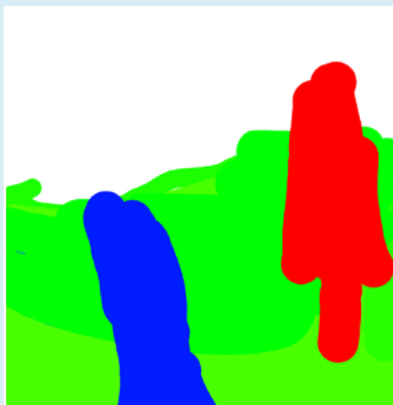


a)



b)

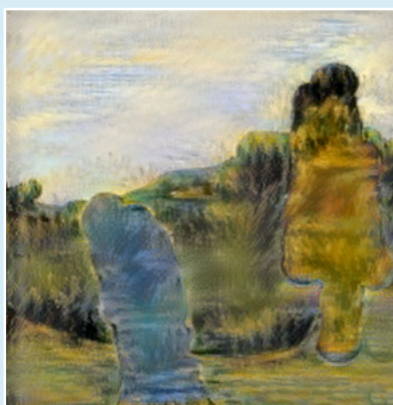
Abb. 10: Vergleich der Computerkunstwerke erstellt mit a) dem hier vorgestellten Programm b) dem Neural Doodler [4]



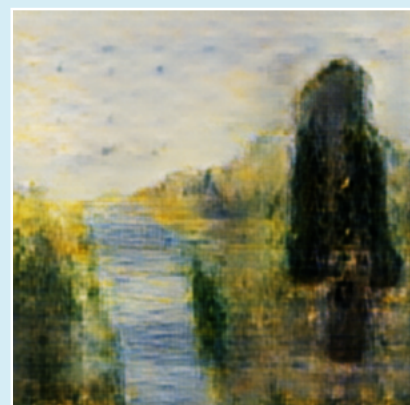
a)



b)



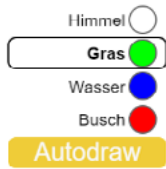
c)



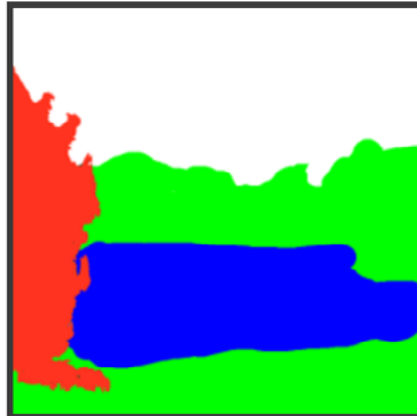
d)

Abb. 11: Vergleich verschiedener Programme mit *Neural Style Transfer*, die Farbskizzen in Computerkunstwerke umwandeln. a) vorgegebene Farbskizze b) Umwandlung mit [6] c) Umwandlung mit [7] d) Umwandlung mit dem hier entwickelten Programm

FARBEN



EINGABE



zurueck

loesche

Convert

AUSGABE



speichern

Monet

FARBEN



EINGABE



Convert

AUSGABE



Abb. 12: Interaktive Webseite zum Erstellen von Computerkunstwerken [11]



Literaturverzeichnis

- [1] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. CoRR, abs/1508.06576, 2015.
- [2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. CoRR, abs/1611.07004, 2016.
- [3] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. CoRR, abs/1604.04382, 2016.
- [4] Alex J. Champandard. Semantic style transfer and turning two-bit doodles into fine art-works. CoRR, abs/1603.01768, 2016.
- [5] Martin Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org, 2015.
- [6] Prisma. <https://prisma-ai.com/> (Letzter Aufruf: 2.10.2019)
- [7] deepart.io – become a digital artist. <https://deepart.io/> (Letzter Aufruf: 2.10.2019)
- [8] The Museum of Modern Art. <https://www.moma.org/> (Letzter Aufruf: 2.10.2019)
- [9] TwoInchBrush | Bob Ross Database. <https://www.twoinchbrush.com/> (Letzter Aufruf: 2.10.2019)
- [10] Kunstkopie. <https://www.kunstkopie.de/> (Letzter Aufruf: 2.10.2019)
- [11] Zeichnen mit Neuronalen Netzwerken. <http://neuralpainter.lima-city.de> (Letzter Aufruf: 2.10.2019)

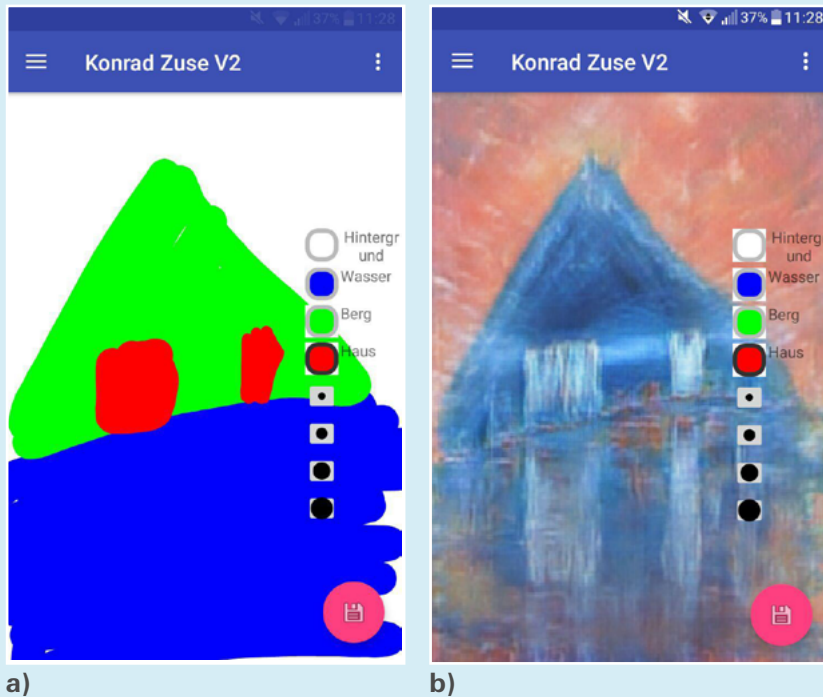


Abb. 13: App zum Erstellen von Computerkunstwerken
 a) die mit Fingern und Stiften erstellte Maske
 b) das Computerkunstwerk im Stil Konrad Zuse V2

Publiziere auch Du hier!

FORSCHUNGSARBEITEN VON
SCHÜLER/INNE/N UND STUDENT/INN/EN

In der Jungen Wissenschaft werden Forschungsarbeiten von SchülerInnen, die selbstständig, z.B. in einer Schule oder einem Schülerforschungszentrum, durchgeführt wurden, veröffentlicht. Die Arbeiten können auf Deutsch oder Englisch geschrieben sein.

Wer kann einreichen?

SchülerInnen, AbiturientInnen und Studierende ohne Abschluss, die nicht älter als 23 Jahre sind.

Was musst Du beim Einreichen beachten?

Lies die [Richtlinien für Beiträge](#). Sie enthalten Hinweise, wie Deine Arbeit aufgebaut sein soll, wie lang sie sein darf, wie die Bilder einzureichen sind und welche weiteren Informationen wir benötigen. Solltest Du Fragen haben, dann wende Dich gern schon vor dem Einreichen an die Chefredakteurin Sabine Walter.

Lade die [Erstveröffentlichungserklärung](#) herunter, drucke und fülle sie aus und unterschreibe sie.

Dann sende Deine Arbeit und die Erstveröffentlichungserklärung per Post an:

Chefredaktion Junge Wissenschaft

Dr.-Ing. Sabine Walter
Paul-Ducros-Straße 7
30952 Ronnenberg
Tel: 05109 / 561508
Mail: sabine.walter@verlag-jungewissenschaft.de

Wie geht es nach dem Einreichen weiter?

Die Chefredakteurin sucht einen geeigneten Fachgutachter, der die inhaltliche Richtigkeit der eingereichten Arbeit überprüft und eine Empfehlung ausspricht, ob sie veröffentlicht werden kann (Peer-Review-Verfahren). Das Gutachten wird den Euch, den AutorInnen zugeschickt und Du erhältst gegebenenfalls die Möglichkeit, Hinweise des Fachgutachters einzuarbeiten.

Die Erfahrung zeigt, dass Arbeiten, die z.B. im Rahmen eines Wettbewerbs wie **Jugend forscht** die Endrunde erreicht haben, die besten Chancen haben, dieses Peer-Review-Verfahren zu bestehen.

Schließlich kommt die Arbeit in die Redaktion, wird für das Layout vorbereitet und als Open-Access-Beitrag veröffentlicht.

Was ist Dein Benefit?

Deine Forschungsarbeit ist nun in einer Gutachterzeitschrift (Peer-Review-Journal) veröffentlicht worden, d.h. Du kannst die Veröffentlichung in Deine wissenschaftliche Literaturliste aufnehmen. Deine Arbeit erhält als Open-Access-Veröffentlichung einen DOI (Data Object Identifier) und kann von entsprechenden Suchmaschinen (z.B. BASE) gefunden werden.

Die Junge Wissenschaft wird zusätzlich in wissenschaftlichen Datenbanken gelistet, d.h. Deine Arbeit kann von Experten gefunden und sogar zitiert werden. Die Junge Wissenschaft wird Dich durch den Gesamtprozess des Erstellens einer wissenschaftlichen Arbeit begleiten – als gute Vorbereitung auf das, was Du im Studium benötigst.

Richtlinien für Beiträge

FÜR DIE MEISTEN AUTOR/INN/EN IST DIES DIE ERSTE WISSENSCHAFTLICHE VERÖFFENTLICHUNG. DIE EINHALTUNG DER FOLGENDEN RICHTLINIEN HILFT ALLEN – DEN AUTOR/INNEN/EN UND DEM REDAKTIONSTEAM

Die Junge Wissenschaft veröffentlicht Originalbeiträge junger AutorInnen bis zum Alter von 23 Jahren.

- Die Beiträge können auf Deutsch oder Englisch verfasst sein und sollten nicht länger als 15 Seiten mit je 35 Zeilen sein. Hierbei sind Bilder, Grafiken und Tabellen mitgezählt. Anhänge werden nicht veröffentlicht. Deckblatt und Inhaltsverzeichnis zählen nicht mit.
- Formulieren Sie eine eingängige Überschrift, um bei der Leserschaft Interesse für Ihre Arbeit zu wecken, sowie eine wissenschaftliche Überschrift.
- Formulieren Sie eine kurze, leicht verständliche Zusammenfassung (maximal 400 Zeichen).
- Die Beiträge sollen in der üblichen Form gegliedert sein, d. h. Einleitung, Erläuterungen zur Durchführung der Arbeit sowie evtl. Überwindung von Schwierigkeiten, Ergebnisse, Schlussfolgerungen, Diskussion, Liste der zitierten Literatur. In der Einleitung sollte die Idee zu der Arbeit beschrieben und die Aufgabenstellung definiert werden. Außerdem sollte sie eine kurze Darstellung schon bekannter, ähnlicher Lösungsversuche enthalten (Stand der Literatur). Am Schluss des Beitrages kann ein Dank an Förderer der Arbeit, z. B. Lehrer und Sponsoren, mit vollständigem Namen angefügt werden. Für die Leser kann ein Glossar mit den wichtigsten Fachausdrücken hilfreich sein.
- Bitte reichen Sie alle Bilder, Grafiken und Tabellen nummeriert und zusätzlich als eigene Dateien ein. Bitte geben Sie bei nicht selbst erstellten Bildern, Tabellen, Zeichnungen, Grafiken etc. die genauen und korrekten Quellenangaben an (siehe auch [Erstveröffentlichungserklärung](#)). Senden Sie Ihre Bilder als Originaldateien oder mit einer Auflösung von mindestens 300 dpi bei einer Größe von 10 · 15 cm! Bei Grafiken, die mit Excel erstellt wurden, reichen Sie bitte ebenfalls die Originaldatei mit ein.
- Vermeiden Sie aufwendige und lange Zahlentabellen.
- Formelzeichen nach DIN, ggf. IUPAC oder IUPAP verwenden. Gleichungen sind stets als Größengleichungen zu schreiben.
- Die Literaturliste steht am Ende der Arbeit. Alle Stellen erhalten eine Nummer und werden in eckigen Klammern zitiert (Beispiel: Wie in [12] dargestellt ...). Fußnoten sieht das Layout nicht vor.
- Reichen Sie Ihren Beitrag sowohl in ausgedruckter Form als auch als PDF

ein. Für die weitere Bearbeitung und die Umsetzung in das Layout der Jungen Wissenschaft ist ein Word-Dokument mit möglichst wenig Formatierung erforderlich. (Sollte dies Schwierigkeiten bereiten, setzen Sie sich bitte mit uns in Verbindung, damit wir gemeinsam eine Lösung finden können.)

- Senden Sie mit dem Beitrag die [Erstveröffentlichungserklärung](#) ein. Diese beinhaltet im Wesentlichen, dass der Beitrag von dem/der angegebenen AutorIn stammt, keine Rechte Dritter verletzt werden und noch nicht an anderer Stelle veröffentlicht wurde (außer im Zusammenhang mit **Jugend forscht** oder einem vergleichbaren Wettbewerb). Ebenfalls ist zu versichern, dass alle von Ihnen verwendeten Bilder, Tabellen, Zeichnungen, Grafiken etc. von Ihnen veröffentlicht werden dürfen, also keine Rechte Dritter durch die Verwendung und Veröffentlichung verletzt werden. Entsprechendes [Formular](#) ist von der Homepage www.junge-wissenschaft.ptb.de herunterzuladen, auszudrucken, auszufüllen und dem gedruckten Beitrag unterschrieben beizulegen.
- Schließlich sind die genauen Anschriften der AutorInnen mit Telefonnummer und E-Mail-Adresse sowie Geburtsdaten und Fotografien (Auflösung 300 dpi bei einer Bildgröße von mindestens 10 · 15 cm) erforderlich.
- Neulingen im Publizieren werden als Vorbilder andere Publikationen, z. B. hier in der Jungen Wissenschaft, empfohlen.

Impressum

[JUNGE]
wissenschaft



Junge Wissenschaft

c/o Physikalisch-Technische
Bundesanstalt (PTB)
www.junge-wissenschaft.ptb.de

Redaktion

Dr. Sabine Walter, Chefredaktion
Junge Wissenschaft
Paul-Ducros-Str. 7
30952 Ronnenberg
E-Mail: sabine.walter@verlag-jungewissenschaft.de
Tel.: 05109 / 561 508

Verlag

Dr. Dr. Jens Simon,
Pressesprecher der PTB
Bundesallee 100
38116 Braunschweig
E-Mail: jens.simon@ptb.de
Tel.: 0531 / 592 3006
(Sekretariat der PTB-Pressestelle)

Design & Satz

Sabine Siems
Agentur „proviele werbung“
E-Mail: info@proviele-werbung.de
Tel.: 05307 / 939 3350

