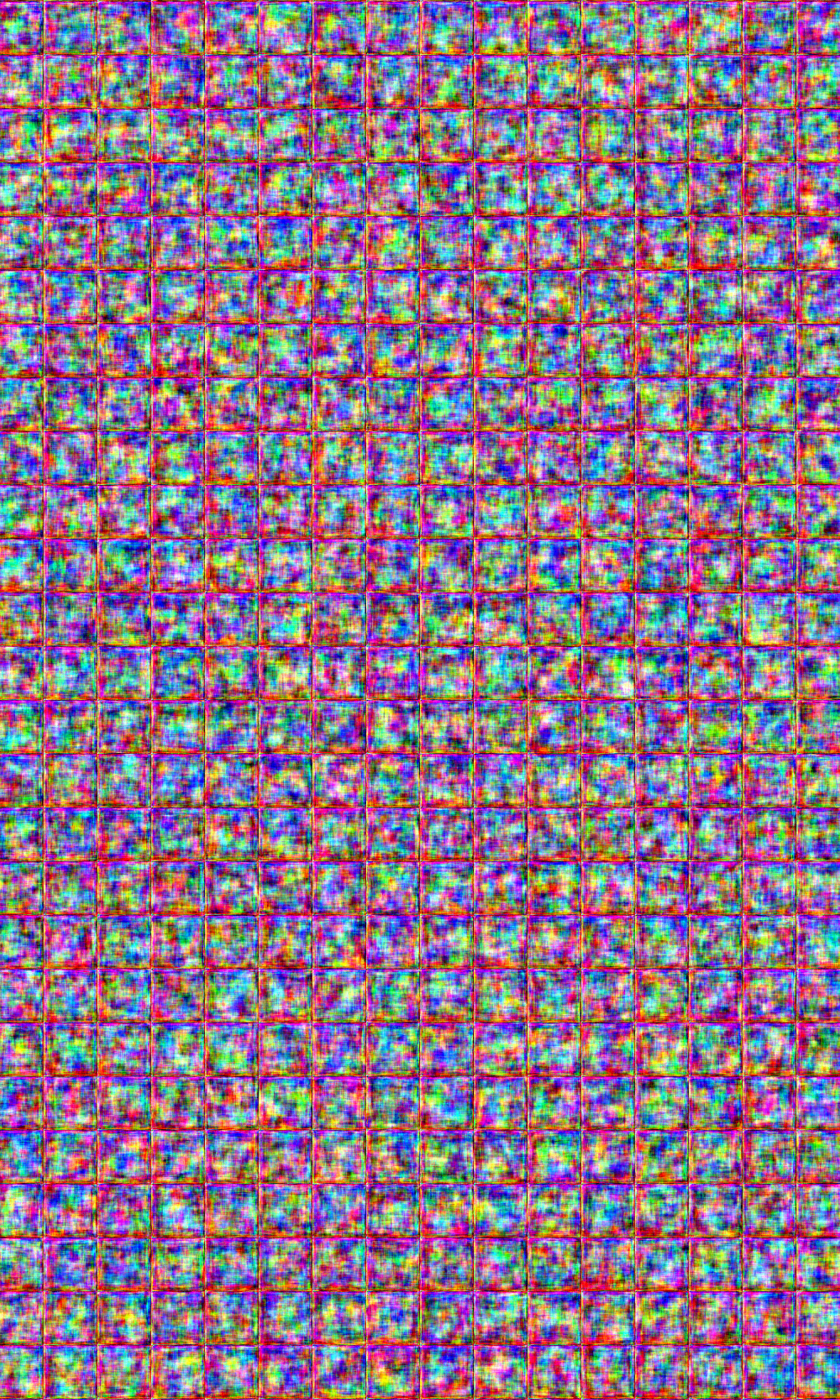
MACHINES DESCRIBING MAN, AS DESCRIBED BY MAN

Titus Ebbecke
7. Semester, 2019
Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt
Fakultät Gestaltung

Übersicht

Alle Bilder in diesem Buch wurden von einem selbst-
lernenden und selbsthandelnden Computer generiert.

Es ist sein Versuch zu verstehen, was Menschen sehen,
wenn sie mit Kunst andere Menschen beschreiben.



Die kognitiven Fähigkeiten des Menschen sind bei ihm so stark ausgeprägt, dass er bis heute nur einem Wesen die Eigenschaft „Intelligenz“ sicher zuschreibt: Sich selbst – trotz der gleichen Herkunft, der gleichen biologischen Grundprinzipien und den oft gleichen Organen, die wir uns mit Millionen von Tierarten teilen.

Schon Aristoteles reservierte die vierte epistemische Stufe „Wissen“ alleine für den Menschen. Unser kognitiver Vorsprung zu den nächst-intelligenteren Wesen ist so groß, dass wir noch heute im Alltag Schimpanse und Qualle gleichermaßen als Tier definieren, während wir uns mit der Bezeichnung Mensch abgrenzen und diesen in den Kontrast zur restlichen niederen Tierwelt stellen.

Die Evolutionstheorie formalisiert den gleichen Entstehungsprozess von Tier und Mensch und Physik und Chemie beschreiben akkurat die Umgebung, die sich diese Wesen schon immer geteilt haben. Beide Modelle können bis heute nicht eindeutig erklären, welche Eigenschaften diese Form von Intelligenz so einzigartig macht. Die Kompetenz, Kunst zu erstellen, scheint allerdings bis heute noch relativ deutlich dem intelligenten Wesen Mensch vorbehalten zu sein. Während Krähen und Affen Werkzeuge basteln können und Termiten unterirdische Städte mit Arbeitsteilung bauen, erstellt

„Creativity is a fundamental feature of human intelligence, and an inescapable challenge for AI“

Margaret Boden

keiner dieser Arten auf regelmäßiger und beabsichtigter Basis Kunst. Die Einzigartigkeit zwischen dem Erstellen von Kunst und die intellektuelle Sonderstellung des Menschen ist also eine relativ feste Parallele. Dass künstlerische Kreativität als eindeutiger Parameter von Intelligenz geeignet ist, ist zwar umstritten, doch die Korrelation zwischen Kunstherstellung und der Intelligenz ihres Urhebers ist deutlich erkennbar.

Künstliche Intelligenz & Deep Learning Algorithmen

Zuerst steht immer die Frage, was ist eine Künstliche Intelligenz? Was ist Intelligenz? Diese Frage soll hier unbeantwortet bleiben. Vorläufig gilt es als kaum möglich den Intelligenzbegriff eindeutig zu formulieren, ohne Gefahr zu laufen, falsche Implikationen zu integrieren. Ohne diese Definition wäre es fahrlässig in diesem Projekt von einer echten „künstlichen Intelligenz“ zu sprechen. Dennoch ist es gleichermaßen unfair die Art Computerprogramme, die hier verwendet werden, ausschließlich als „Machine Learning Algorithmus“ zu bezeichnen, wie es fragwürdig ist sie als „Künstliche Intelligenz“ zu betiteln. Dabei schließen sich beide Begriffe nicht aus.

Ein „Machine Learning“ Algorithmus, also eine mathematische Handlungsvorschrift mit maschinellem Lernen, ist essenziell für den Intelligenzbegriff. Letzterer wird schließlich primär mit der Fähigkeit zu Lernen definiert. Insofern hat das Verfahren, dass die hier abgebildeten Kunstwerke generiert hat, nichts mit konventionellen „dummen“ Computerprogrammen zu tun.

Das Programm besteht nicht aus einer Anweisung Farben zu wählen und sie an einen vordefinierten Ort zu platzieren. Stattdessen besteht die Operation aus einem sehr umfangreichen Lernverfahren. In diesem werden unter Einsatz von leistungsstarken Computern, mehrere tausende oder zehntausende Bilder analysiert und ihre einzelnen Aspekte in eine Art universelles Gedächtnis gespeichert.

Die Programme versuchen in vielen Millionen Berechnungen pro Sekunde herauszufinden, was die eingespeicherten Bilder (in diesem Fall) menschlich macht. Diese Lernphase dauert viele Stunden, Tage und sofern nicht die leistungsstärksten konventionellen Computer überhaupt genommen werden, oft Wochen oder Monate. Ähnlich wie bei einem menschlichen Agenten, hört diese Lernphase nie auf. Es gibt keinen Punkt an dem „fertig“ gelernt wurde, an dem der scheinbar einzelne, universelle Schlüssel zum Geheimnis des Urwerks gefunden wurde. Schließt man mit der Lernphase ab, weil die potenziellen Ergebnisse zufriedenstellend sind, so hat man kein fertiges Bild zu Hand. Stattdessen wird das Erlernte in einer Datei gespeichert. Erst diese wird benutzt, um beliebig viele Bilder zu generieren. Ähnlich einem Menschen, der differenziert, zwischen Lernprozess und reinem Kreationsprozess. Das Programm ist keine Anleitung zum Generieren von Bildern, sondern eine Aufforderung an den Computer alles auszutesten und zu bewerten, bis er eines Tages glaubt zu wissen, was richtig und was falsch ist.

Der Computer nutzt Komponenten wie Lernen, Kurz- und Langzeitgedächtnis und Probierprozesse ohne garantierte Erfolgschance. Ohne den Intelligenzbegriff genau definieren zu können, ist es schon dadurch möglich fundamentale Gemeinsamkeiten zwischen uns intelligenten Wesen und diesen Programmen zu erkennen.

Generative Adversarial Networks

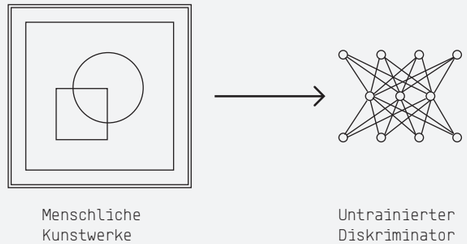
Generative Adversarial Networks sind Computeralgorithmen, welche meist dazu dienen möglichst realistische Bilder komplett synthetisch zu generieren. Sie werden mit tausenden oder zehntausenden echten Bildern trainiert, um nach ausreichend Training Bilder von Gesichtern, Tieren, Schlafzimmern oder ganze Landschaften zu generieren.

Das Erzeugen von fotorealistischen Resultaten in hoher Auflösung mit privat zugänglichen Computersystemen ist erst seit wenigen Monaten vor Veröffentlichung dieses Werkes möglich. GAN‘s wurden in ihrer Rohform erstmals 2013 beschrieben und 2014 konkretisiert.

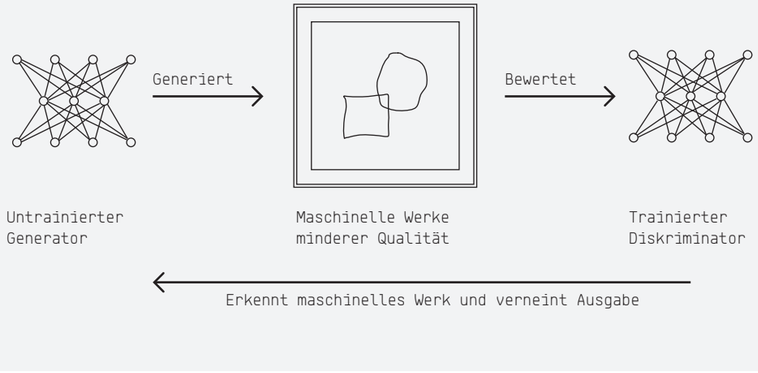
Neben der synthetischen Natur von GAN’s sind diese Arten von Machine Learning Algorithmen interessant, weil sie unbewacht lernen. Statt von einem menschlichen Überwacher abhängig zu sein, der die Resultate oft mehrere tausend Mal bewerten müsste, um dem Algorithmus beizubringen, was richtig und falsch ist, bestehen GAN’s aus zwei *künstlichen neuronalen Netzwerken* – KNN‘s. Während eines dieser Netzwerke, genannt „Generator“, beispielsweise Gemälde auf Zufallsbasis generiert, trainiert ein zweites KNN, genannt „Diskriminator“, zwischen den künstlich generierten, „falschen“ Daten des Generators und menschlich eingegebenen, „echten“ Daten zu unterscheiden. Hat dieser ausreichend gelernt, echte und falsche Bilder zu unterscheiden, so wird er zu Anfang erkennen, dass die Bilder des Generators nicht echt sind, sondern künstlich generiert. Er verneint die Ausgabe an den menschlichen Überwacher, sodass sein generativer Konkurrent erneut Bilder erstellt, welche minimal echter wirken als seine vorherigen. Wird dies iterativ wiederholt, so kann irgendwann der Diskriminator nicht mehr erkennen, ob die vom Generator erstellten Bilder echt sind. Erst hier braucht der Überwacher einzugreifen und die Bilder zu beurteilen. Je nach Qualität des Diskriminators ist dann selbst der Mensch nicht mehr in der Lage, zwischen echten Bildern und vom Generator erstellten zu unterscheiden.

Von den wenig öffentlich zugänglichen GAN‘s benötigen alle erhebliche Rechenleistung. Um fotorealistische Portraits in einer Auflösung von 1024^2 Pixeln zu generieren, brauchen konventionelle Heimrechner mehrere Wochen oder Monate, Laptops sogar Jahre. Die effizientesten GAN‘s können diese Resultate mit den stärksten heute zugänglichen Grafikkarten in einigen Tagen erreichen.

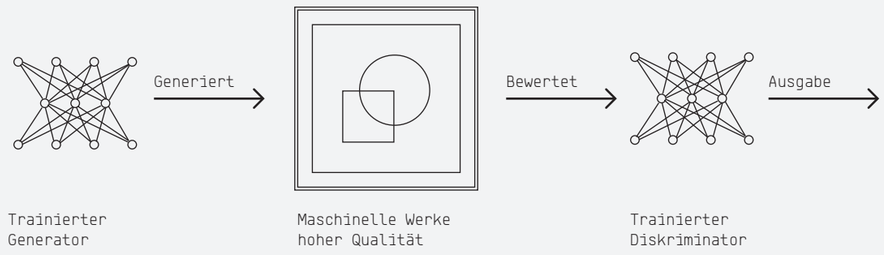
1



2



3

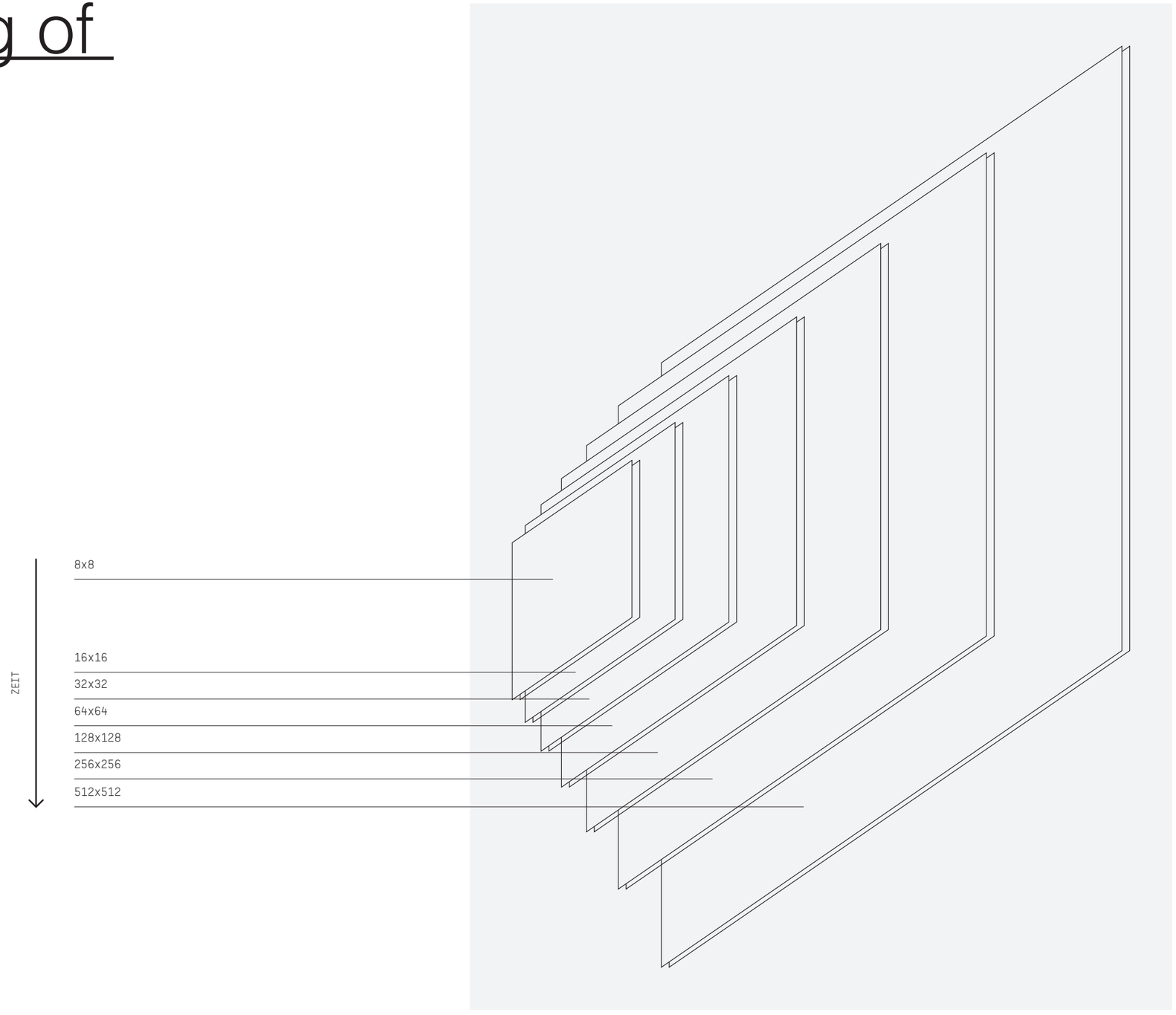


Progressive Growing of GAN‘s

Im Oktober 2017 stellten Tero Karras, Timo Aila, Samuli Laine und Jaakko Lehtinen ein hocheffizientes GAN vor. Dieses trainiert nicht linear, sondern progressiv. Die Trainingsbilder werden erst in unterschiedlich niedrige Auflösungen konvertiert. Durch den wesentlich geringeren Informationsgehalt von niedrigauflösenden Bildern, können die darin enthaltenden Informationen wesentlich schneller erlernt werden. Schrittweise wird die Auflösung erhöht und die neuronalen Netzwerke erlernen die mit der Pixelanzahl wachsenden Informationen. Dabei erinnern sie sich bei jedem Schritt an die erlernten Informationen aus der vorherigen, niedrigeren Auflösung und müssen das hochauflösende Bild nur zum Teil neu analysieren.

Dieser Algorithmus zählt zum Zeitpunkt der Veröffentlichung zu den effizientesten GAN’s und ist in der Lage nach einigen Tagen bis wenigen Wochen Training auf Grafikkarten mit Maximalleistung wie der NVIDIA V100, menschliche Gesichter in 1024^2 Pixeln zu generieren, die mit echten kaum noch zu unterscheiden sind. Dieser Fortschritt in der GAN Entwicklung ist bisher nur in einigen wenigen Algorithmen implementiert.

Der Zeitaufwand des Trainings ist um ein vielfaches geringer als bei linearem Training. Nur so ist es möglich mit einem realistischen Budget, Bilder in angemessener Auflösung zu generieren.



Training und Genres

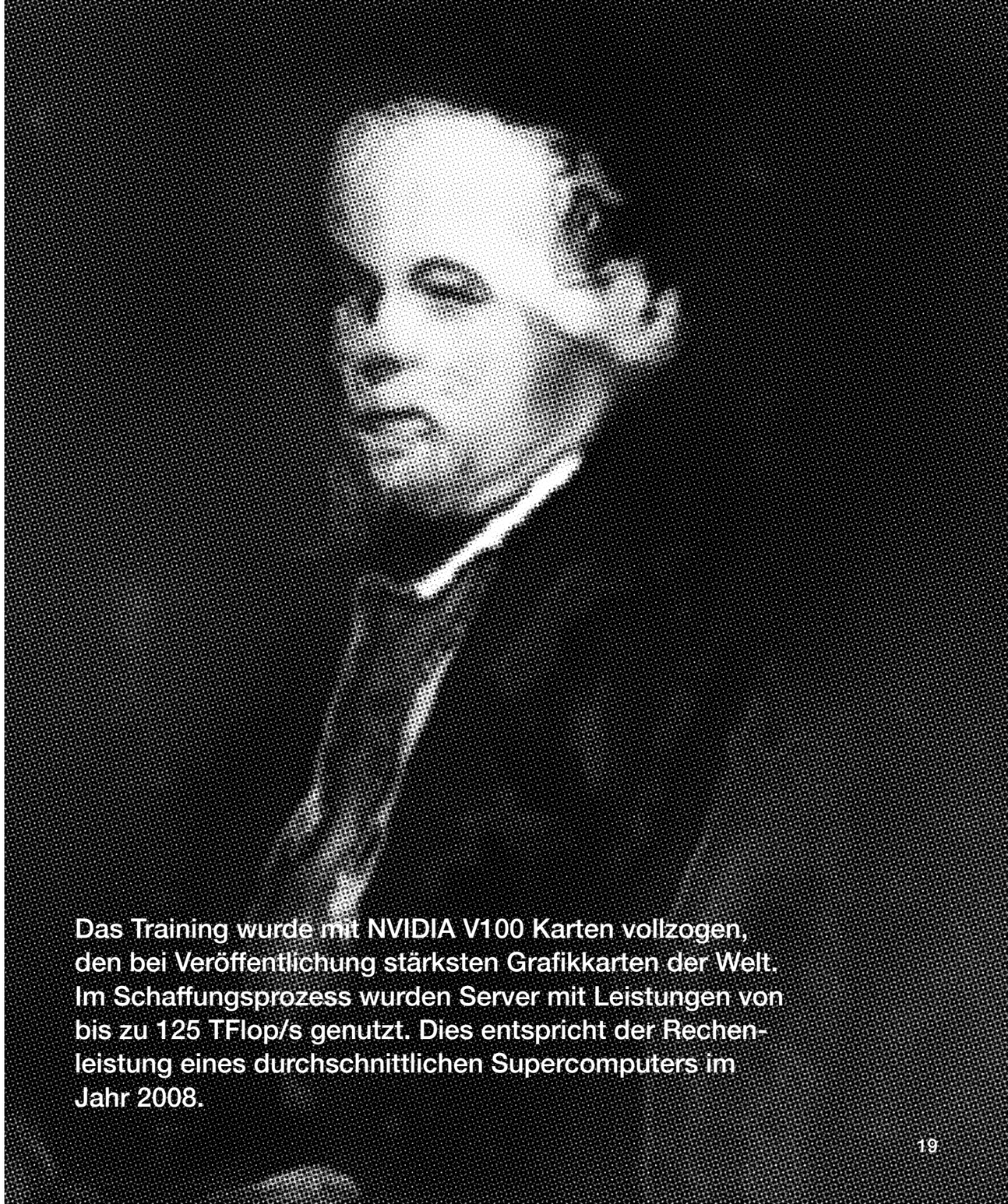
Dieses Projekt hat zwei Programme geschaffen, wobei eines dieser Programme auf Portraits und das andere auf abstrakte Werke trainiert ist. Beide wurden mit zweidimensionalen Kunstwerken der öffentlichen Domain über drei Tage trainiert.

Portrait

Hierbei wurden 14125 Portrait's und Selbst-Portraits aus allen verfügbaren Zeitepochen, von Abbildungen auf Vasen aus der Antike, bis zu kubistischen Darstellungen von Picasso genutzt. Die Herausforderung des Algorithmus ist es, Muster wie Gesichter zu erkennen, die gerade in stark abstrahierten Werken kaum erkennbar sind. Die Bilder wurden nicht beispielsweise in Epochen selektiert, um der Maschine das menschenähnliche Generieren zu erleichtern. Stattdessen war der Computer darauf angewiesen, einzelne Stile und Richtungen zu unterscheiden, um so in der Lage zu sein, visuell komplett unterschiedliche Werke zu erstellen. Das Computerprogramm arbeitet mit quadratischen Bildern. Um bei einem Zuschnitt zu garantieren, dass ein Gesicht im Bild ist, wurde eine Gesichtserkennungssoftware verwendet, um Gesichter im Bild zu identifizieren und das Bild anhand dieses Gesichtes anzuschneiden.

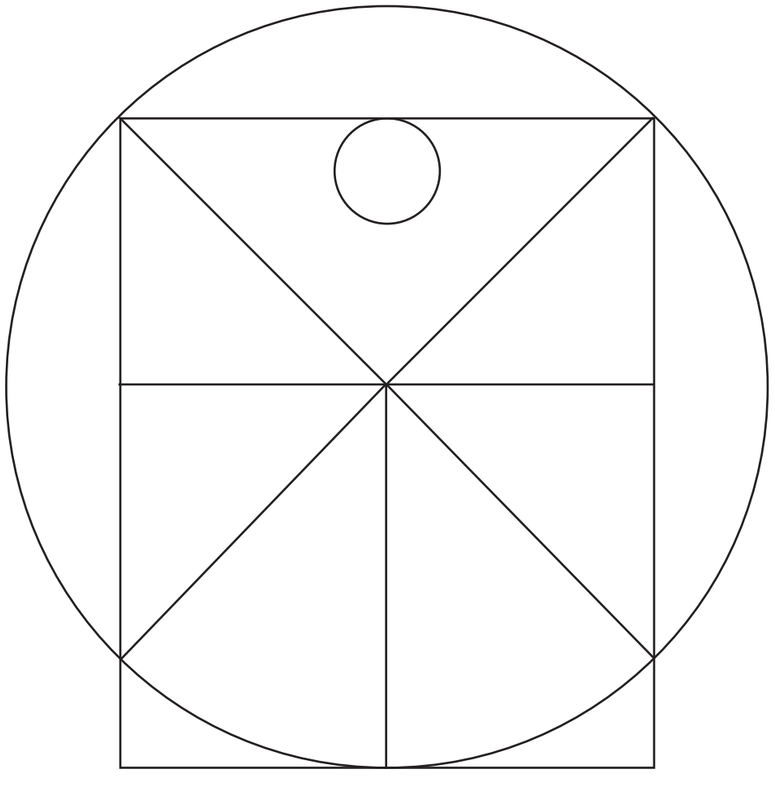
Abstrakt

Für dieses Training wurden 7360 Werke mit der Stileigenschaft „Abstrakt" ausgewählt. Die komplette visuelle Streuung, nicht nur durch künstlerische Interpretation des Motivs, sondern auch durch die Vielzahl an Motiven selber, ist eine sehr große Hürde im Lernprozess. Während Portraits zumindest inhaltlich eine Gleichheit aufweisen, muss der Algorithmus hier mit Darstellungen von Häusern, Menschen, Landschaften oder anderen Objekten arbeiten. Dabei ist ihm ein fotorealistischer Blick auf diese Dinge komplett fremd. Er erkennt das, was Menschen in diesen Dingen sehen – weit über das rein visuelle Sehen hinaus.



Das Training wurde mit NVIDIA V100 Karten vollzogen, den bei Veröffentlichung stärksten Grafikkarten der Welt. Im Schaffungsprozess wurden Server mit Leistungen von bis zu 125 TFlop/s genutzt. Dies entspricht der Rechenleistung eines durchschnittlichen Supercomputers im Jahr 2008.

Werke
Portrait



Die Maschine lernt über drei Tage menschliche Interpretationen von anderen Menschen. Weder Epoche, noch Stil wurden selektiert, um dem Algorithmus bei der Suche nach Mustern zu helfen. Er verarbeitet 14125 verschiedene Portraits und Selbst-Portraits. Fragile Konturzeichnungen, dramatische Barockgemälde und Pastellbilder des Impressionismus, müssen verarbeitet und unterschieden werden.

Während realistische Gemälde der Renaissance das Erkennen von Gesichtern erleichtern, muss der Algorithmus den Unterschied zu abstrakten Portraits erlernen. Ihm wird nicht die Aufgabe gegeben, herauszufinden wie ein Gesicht wahrhaftig aussieht, sondern wie Menschen arbeiten, wenn sie ihre eigene Vision des Gegenüber darstellen.

So beginnt der Algorithmus eine weites Spektrum aus Stilen und Darstellungstechniken zu erlernen. Er demonstriert ein gewissen Durchschnitt, auf seiner Suche nach dem Menschlichen in den Bildern. Das Programm ist kein Drucker, das Imitate erschafft, sondern das Ergebnis einer hoch frequentierten Suche nach den menschlichen Parallelen in dem Chaos aus Bildern etlicher Epochen, Stile und Maler.

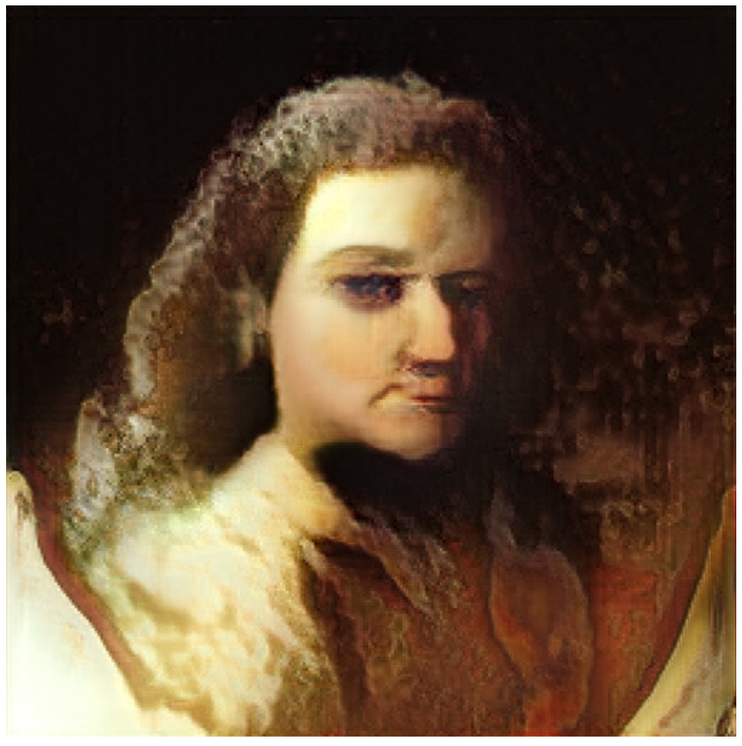
—

Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—

Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel



—

Seed 3
FP16 & FP32 bei 8062 Bildern
512² Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—

Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed 20
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

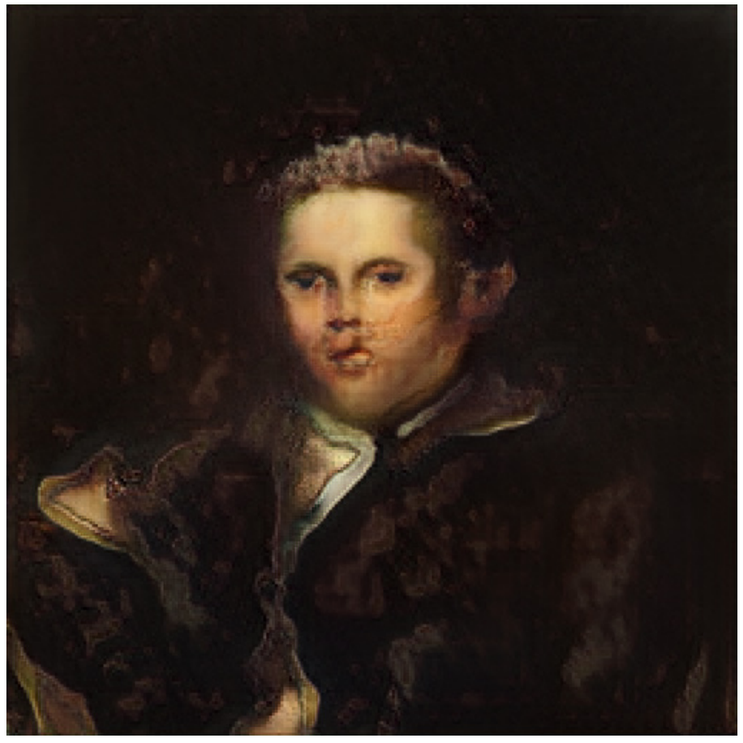
—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed 21
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel



—
Seed unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

„Die drei Geschwister". Man beachte die fast gleichen Gesichter, die zweimal auf einem weiblichen Körper und einmal auf einem männlichen verwendet wurden.

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—

Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

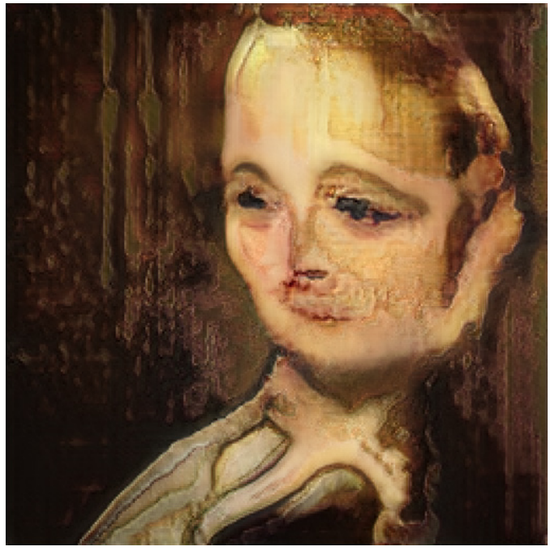
—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed 9
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed 13
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed 27
FP16 & FP32 bei 8062 Bildern
512^2 Pixel



—
Seed 22
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

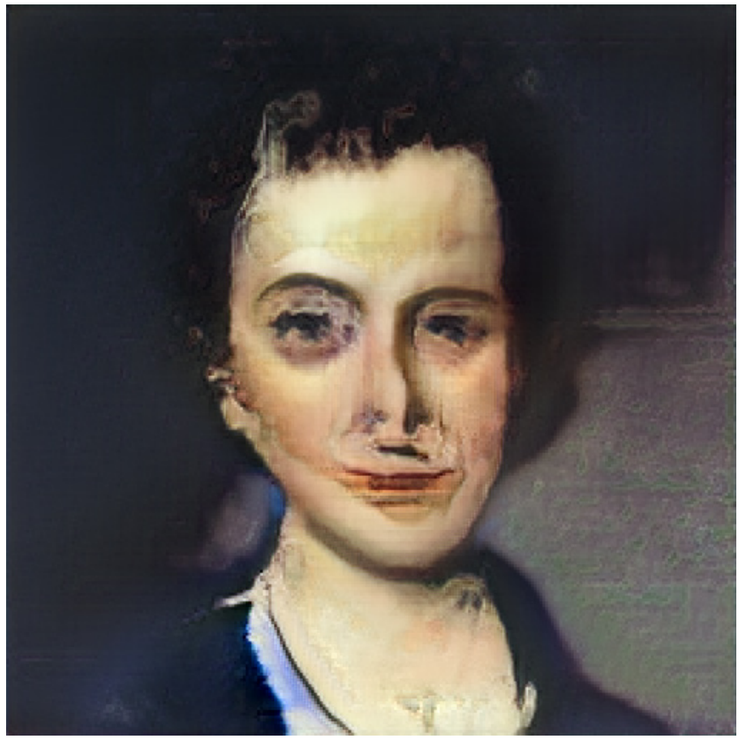
—

Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel



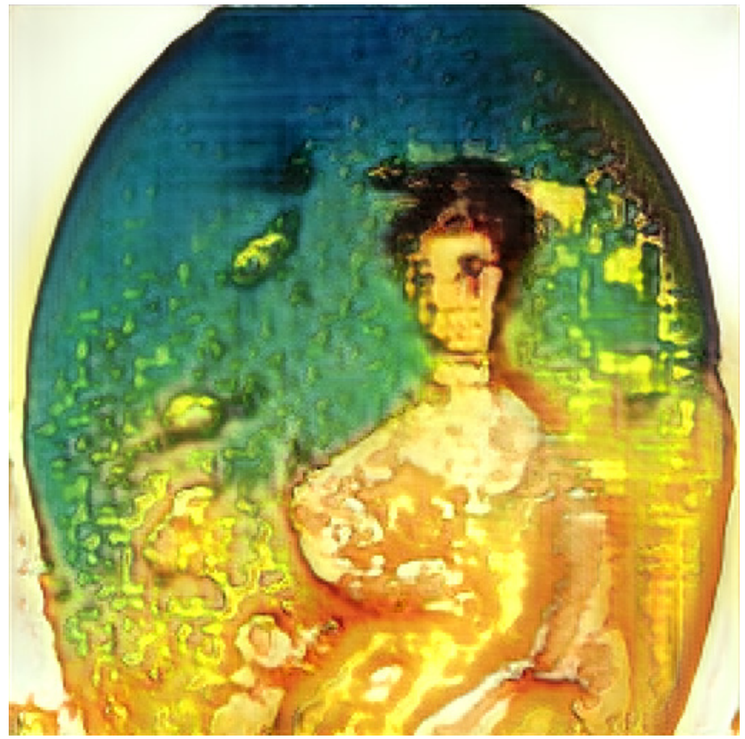
—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—

Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel



—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

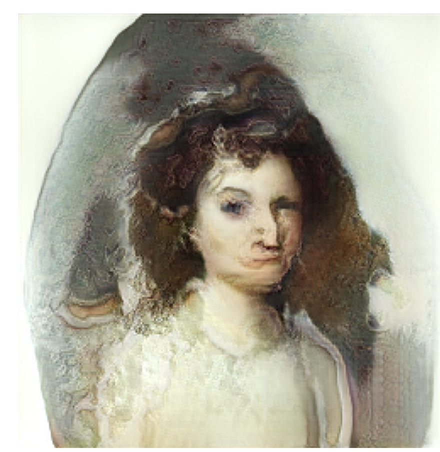
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

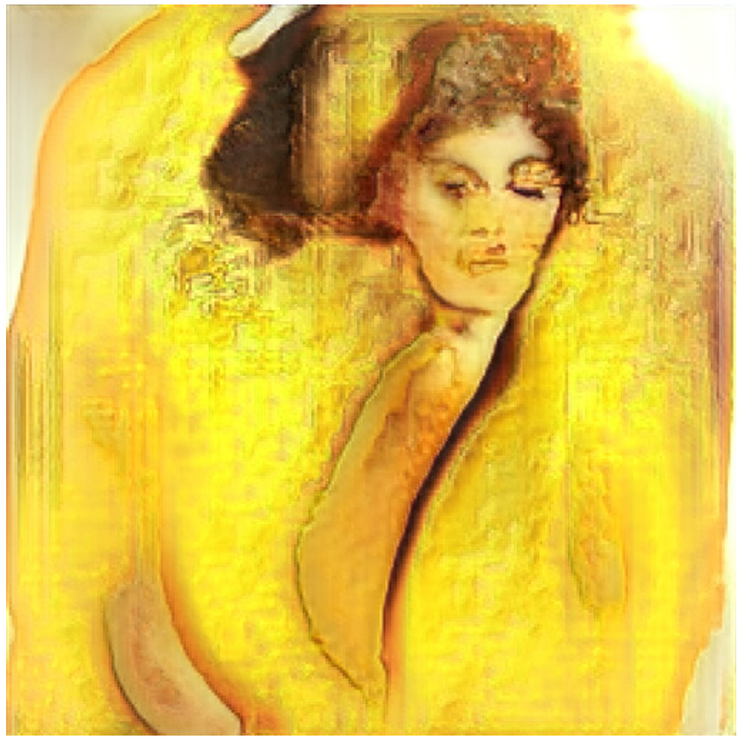
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel



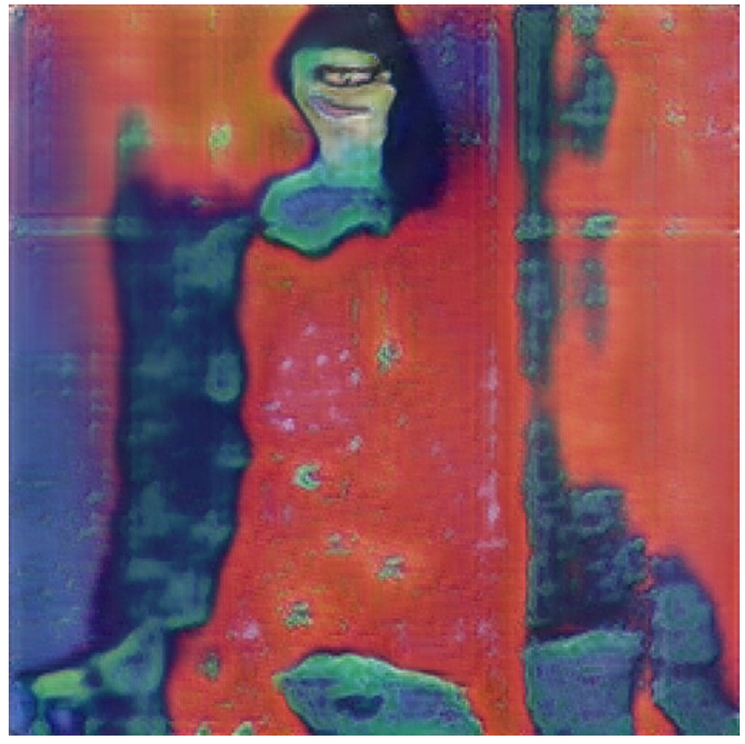








—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel



—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

Die Maschine verarbeitet sowohl Gemälde mit intensiven Farben, wie auch Zeichnungen oder Karikaturen. Wie hier zu sehen, ist sie genauso in der Lage minimalistische Konturzeichnungen zu erschaffen, wie sie gefüllte Barockgemälde mit vollem Chiaroscuro generieren kann.

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512² Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512² Pixel

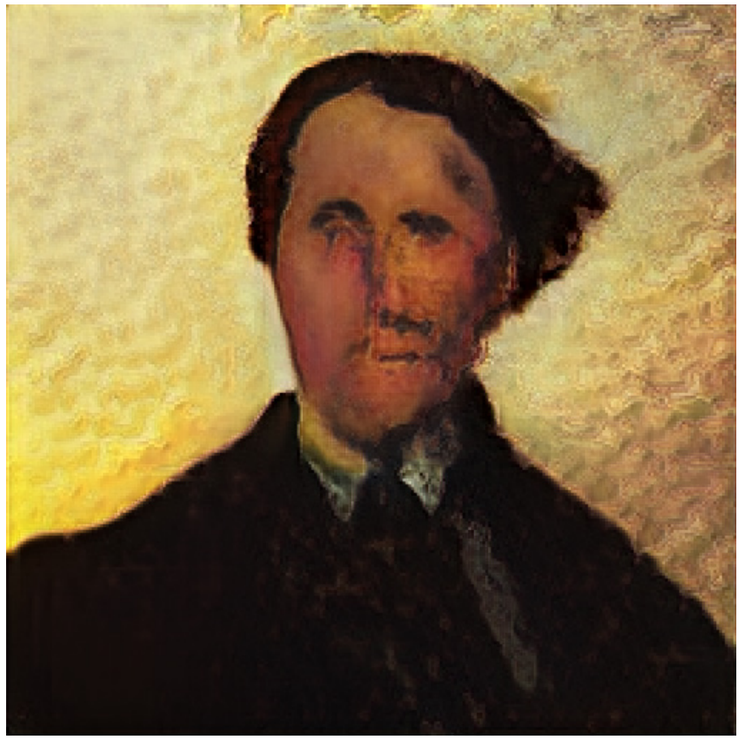
—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

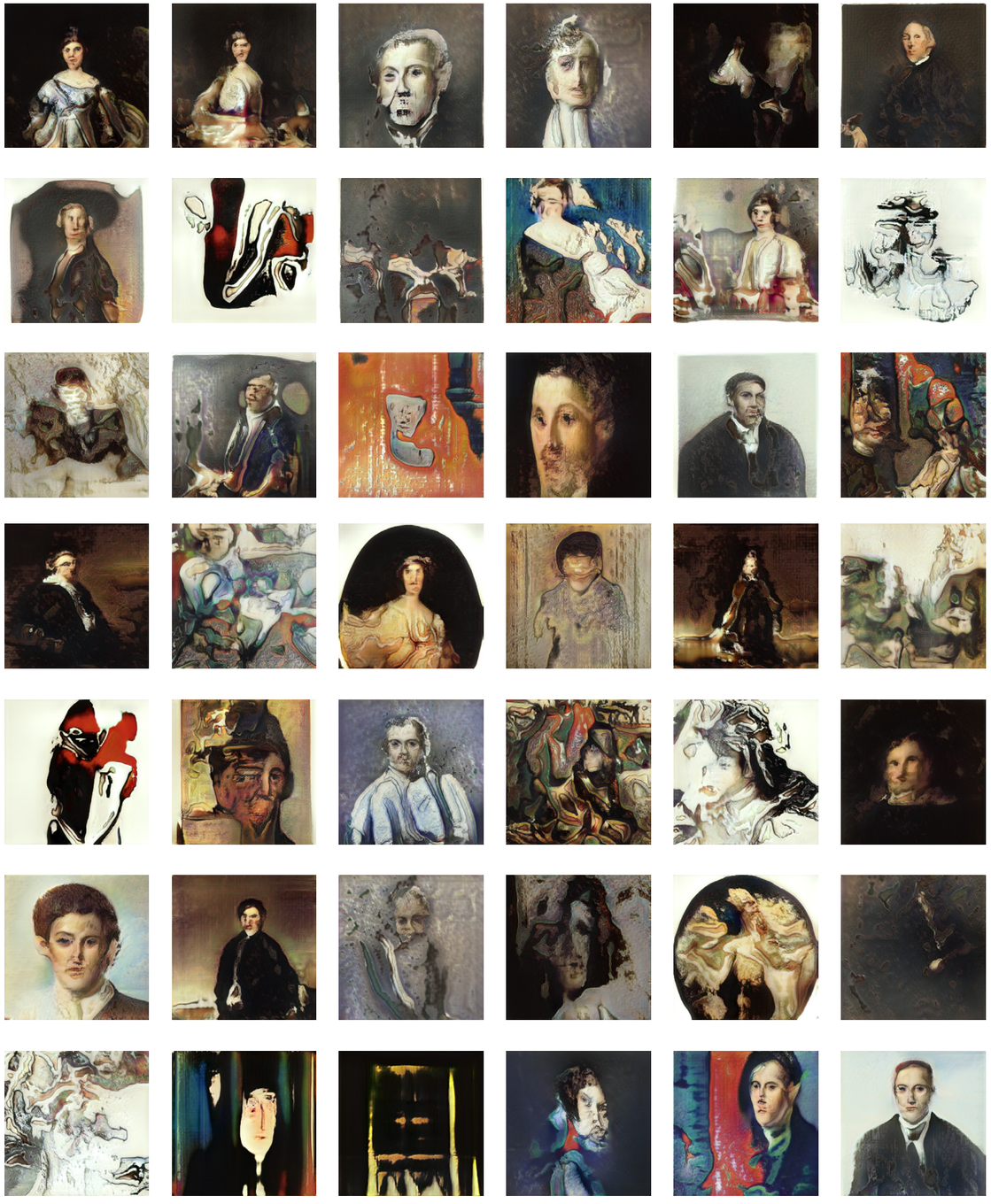


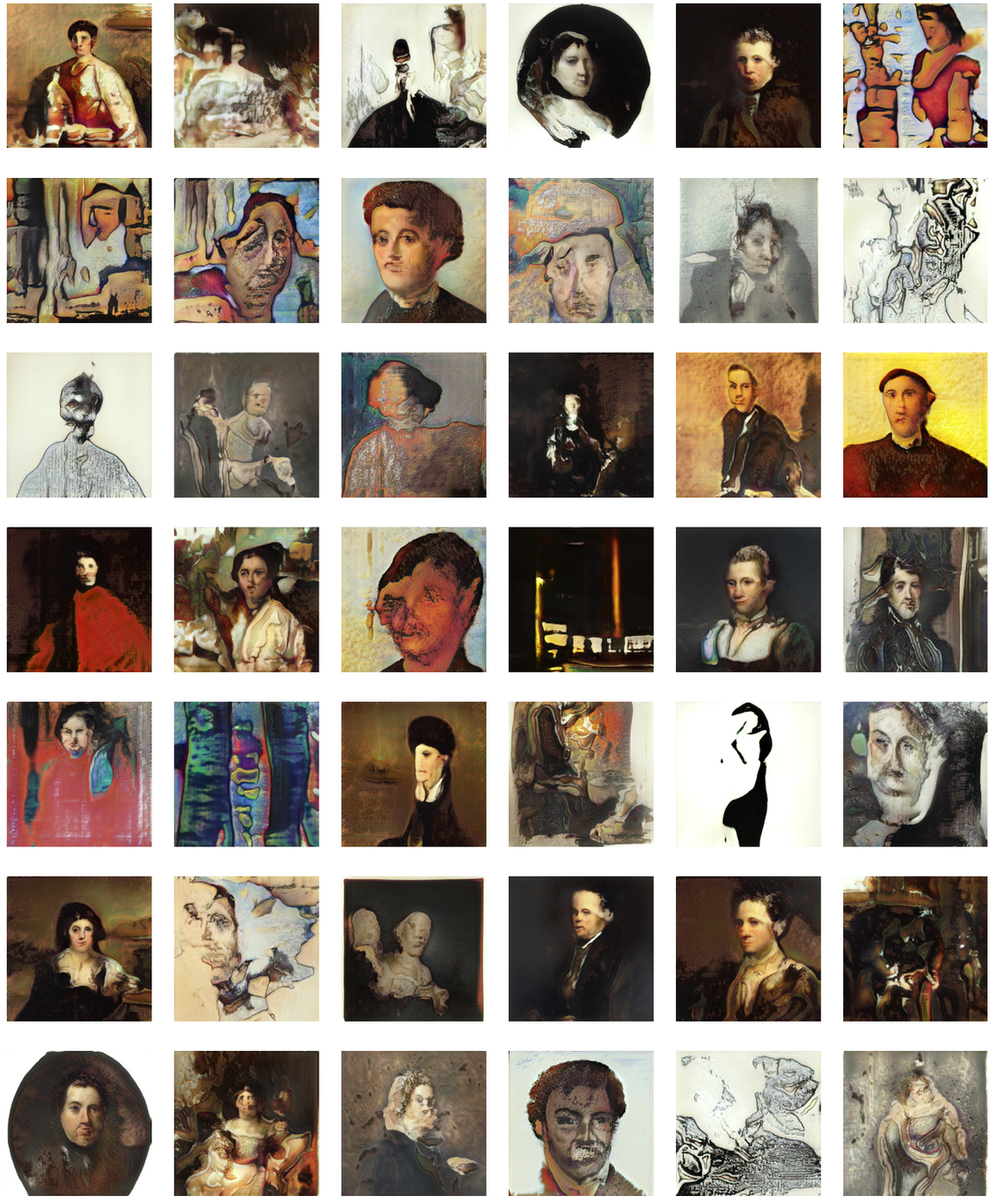
—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

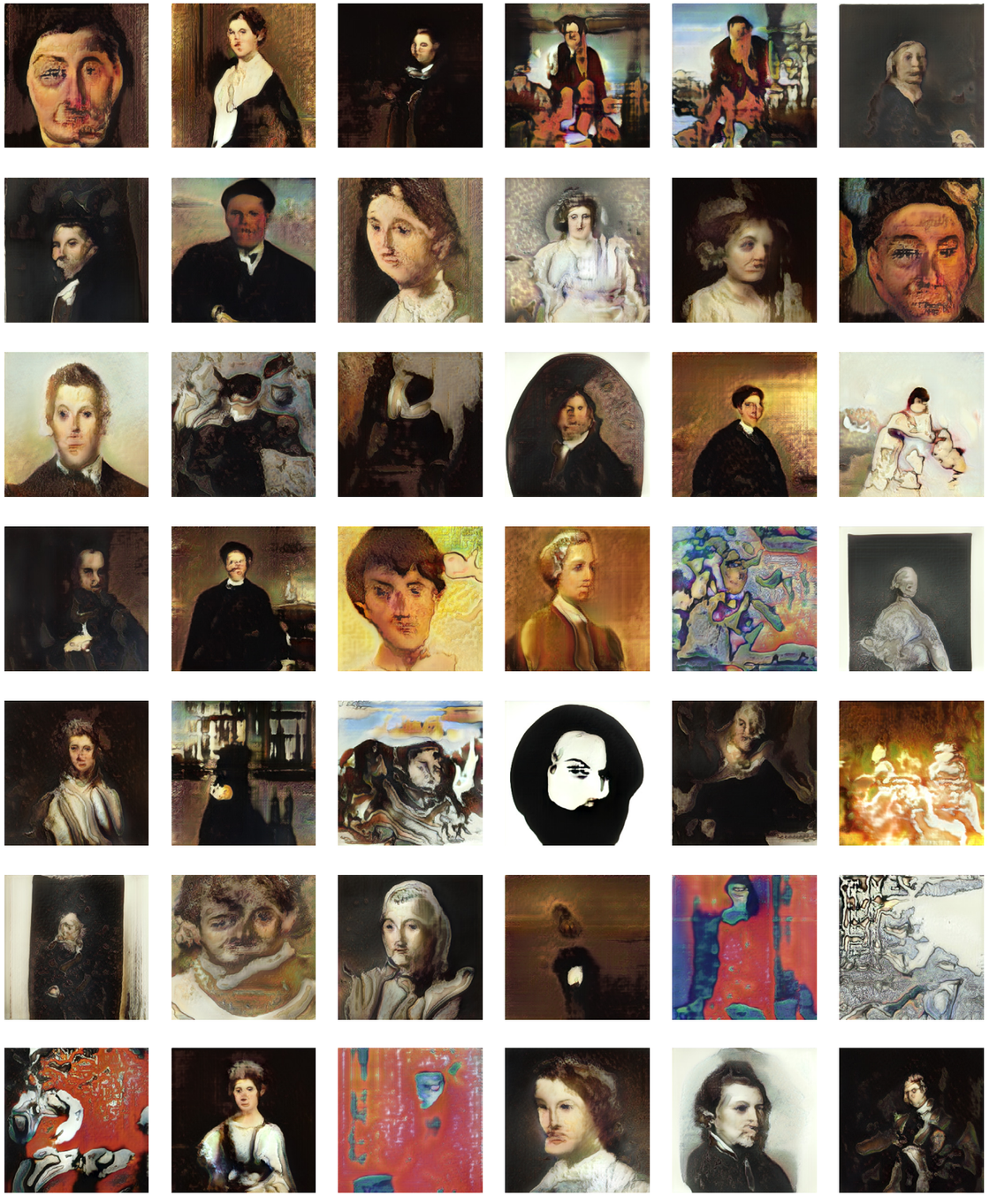
—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

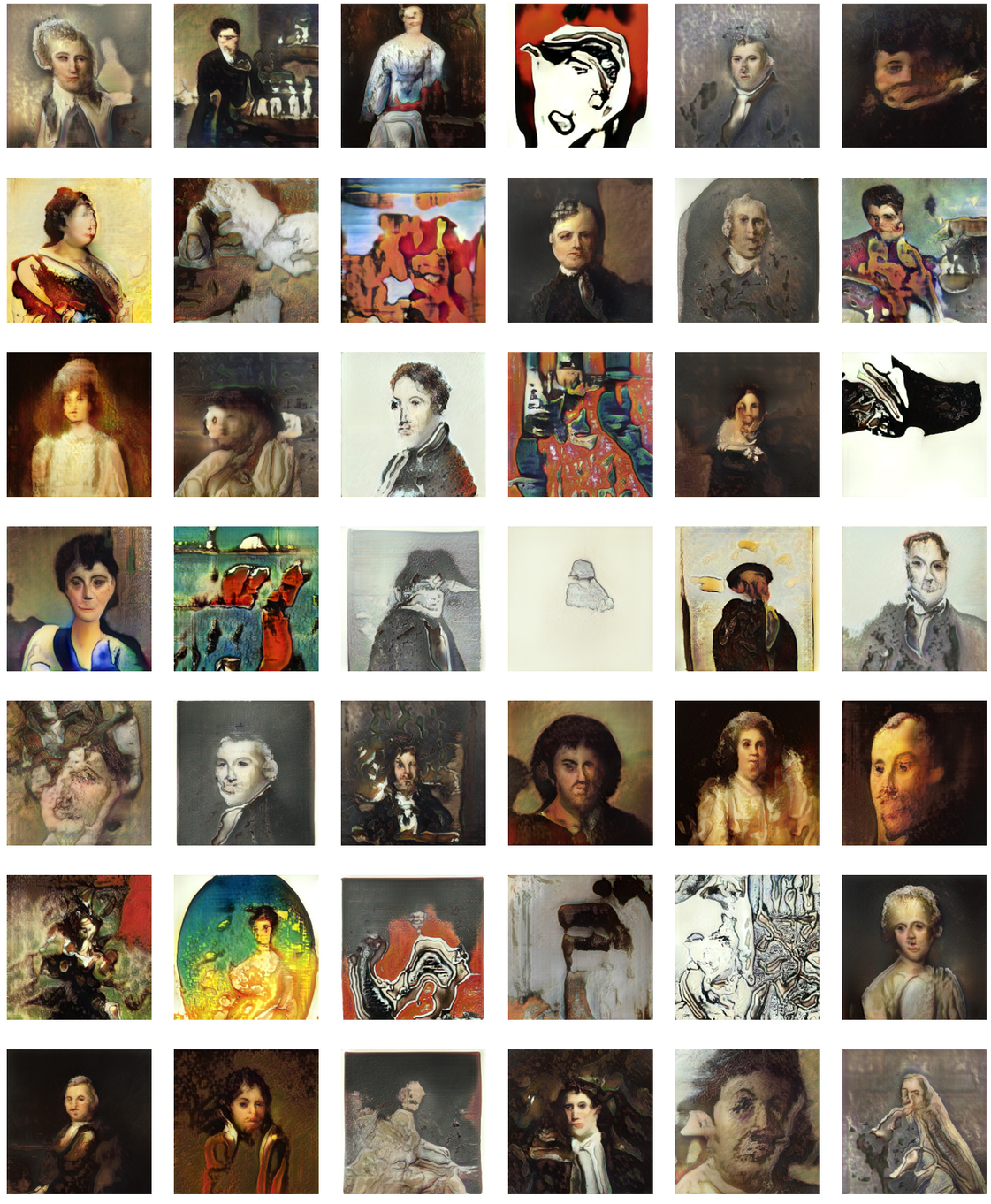
—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

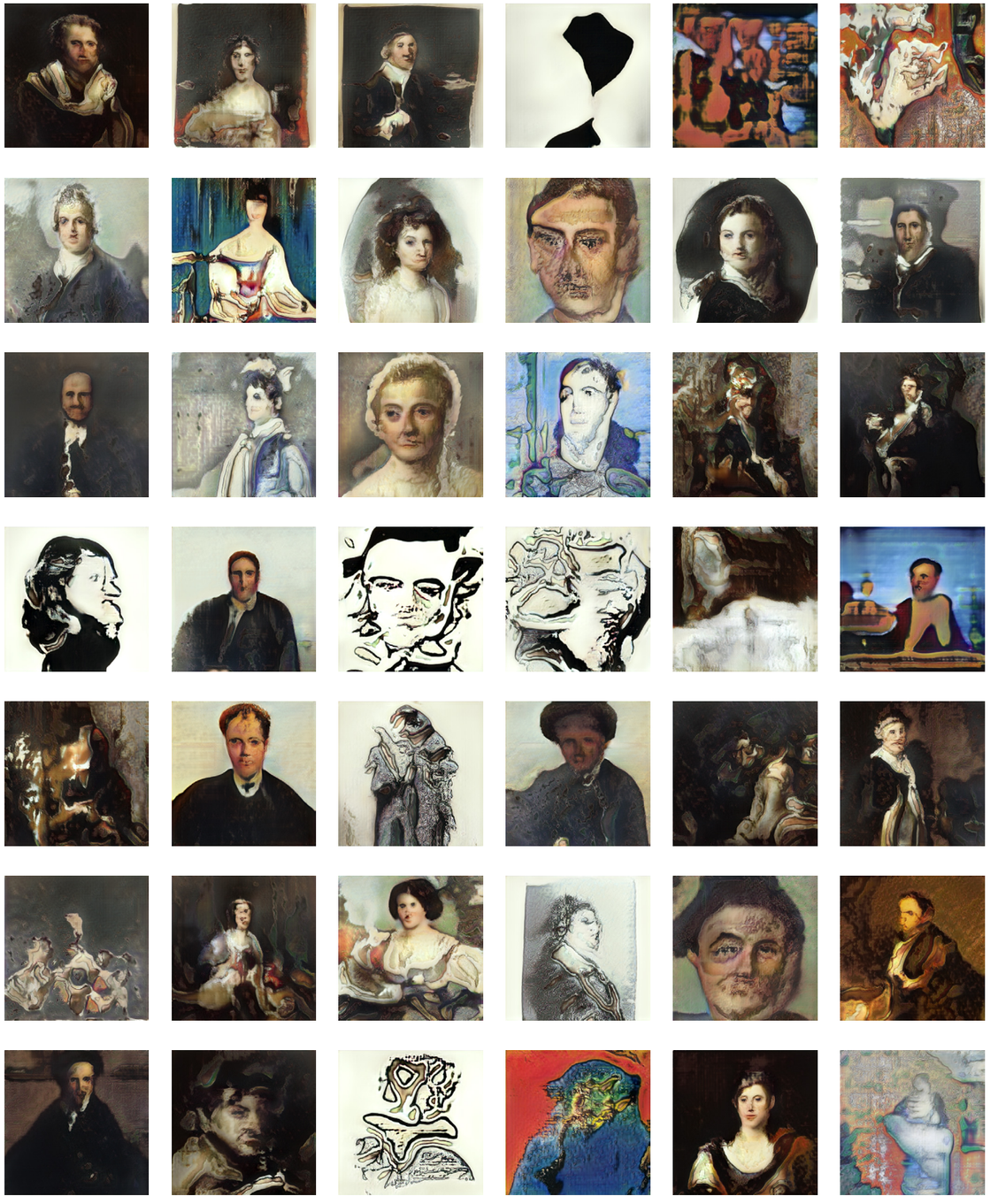
—
Seed Unbekannt
FP16 & FP32 bei 8062 Bildern
512^2 Pixel

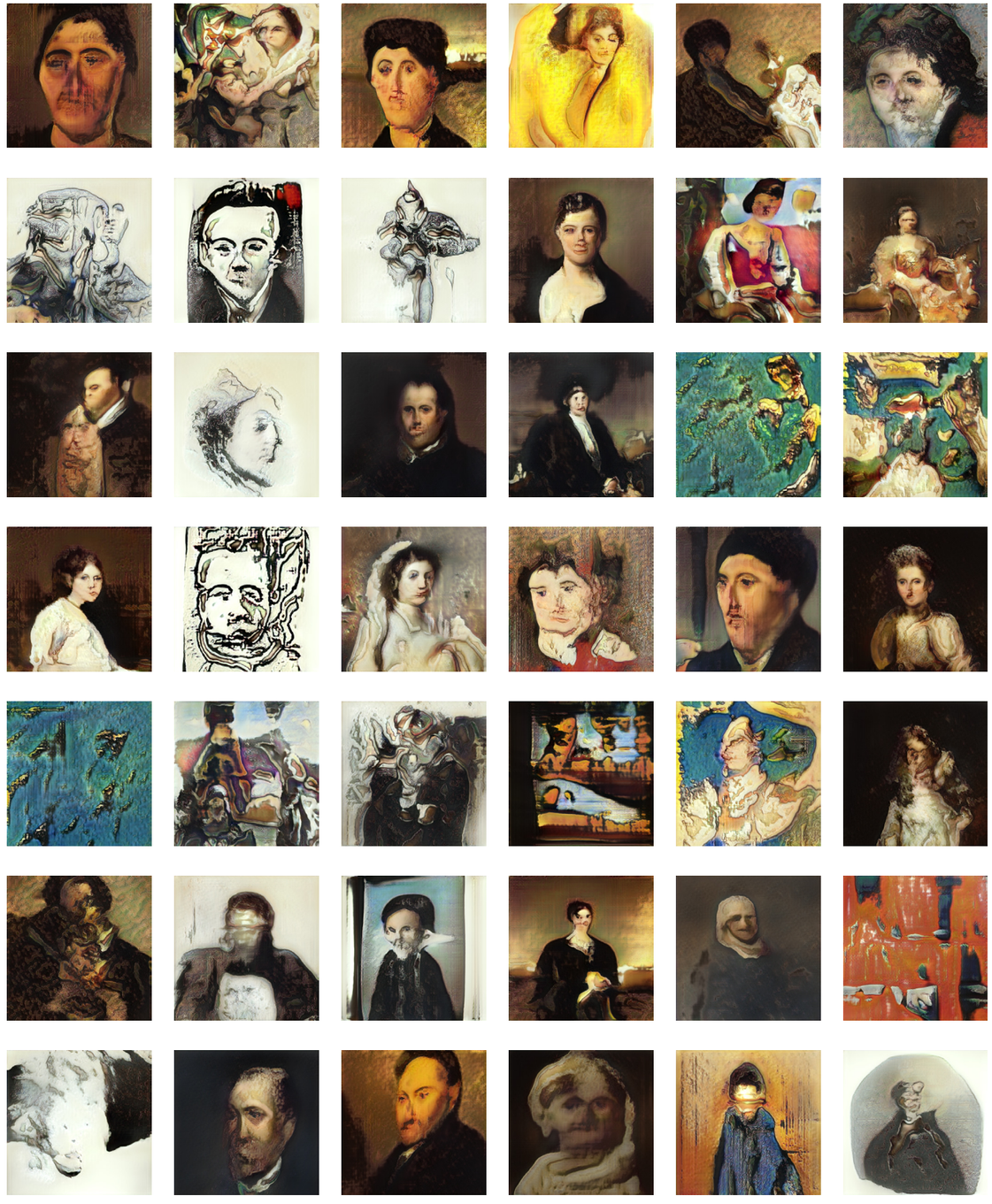












Machines describing man, as described by machines

Google‘s AutoML Vision ermöglicht es, beliebige Bilder von einer Software auf KI Basis mit unterschiedlichen Eigenschaften zu bewerten. Dieser Algorithmus kann den Inhalt von Bildern erkennen, beschreiben und in unterschiedlicher Weise kategorisieren. Um einen Einblick zu geben, was eine Maschine von maschinengenerierter Kunst hält, wurden verschiedene Werke von dieser externen KI analysiert und bewertet.

Labels

Portrait	95%
Self-portrait	95%
Art	88%
Head	88%
Painting	85%
Illustration	79%
Visual Arts	79%
Jaw	74%
Drawing	72%
Modern Art	64%
Sketch	54%
Artist	53%

Faces

Joy	**Very unlikely**
Sorrow	**Very unlikely**
Anger	**Very unlikely**
Surprise	**Very unlikely**
Exposed	**Very unlikely**
Blurred	**Very unlikely**
Headwear	**Very unlikely**

Roll: -2° Tilt: 2° Pan: 10°

Confidence 72%

Labels

Head	89%
Illustration	85%
Art	79%
Portrait	71%
Self-portrait	69%
Drawing	62%
Jaw	57%
T-shirt	54%

Faces

Joy	**Very unlikely**
Sorrow	**Very unlikely**
Anger	**Very unlikely**
Surprise	**Very unlikely**
Exposed	**Very unlikely**
Blurred	**Very unlikely**
Headwear	**Likely**

Roll: -4° Tilt: 2° Pan: -9°

Confidence 64%

Labels

Hair	98%
Portrait	96%
Lady	95%
Hairstyle	84%
Chin	83%
Self-portrait	81%
Painting	70%
Long Hair	65%
Art	50%

Faces

Joy	**Very unlikely**
Sorrow	**Very unlikely**
Anger	**Very unlikely**
Surprise	**Very unlikely**
Exposed	**Very unlikely**
Blurred	**Very unlikely**
Headwear	**Very unlikely**

Roll: 0° Tilt: -6° Pan: 38°

Confidence 100%

Labels

Portrait	97%
Painting	95%
Self-portrait	94%
Lady	92%
Art	78%
Human	76%
Visual Arts	59%

Faces

Joy	**Very unlikely**
Sorrow	**Very unlikely**
Anger	**Very unlikely**
Surprise	**Unlikely**
Exposed	**Very unlikely**
Blurred	**Very unlikely**
Headwear	**Very unlikely**

Roll: 2° Tilt: 10° Pan: 12°

Confidence 100%

Labels

Self-portrait	95%
Portrait	94%
Painting	89%
Art	86%
Drawing	82%
Figure Drawing	79%
Human	77%
Sketch	76%
Watercolor Paint	74%
Jaw	74%
Illustration	70%
Visual Arts	69%

Logos

misako — **Portugiesischer Mode-Accessoire Hersteller**

Safe Search

Violence — **Possible**

Labels

Portrait	96%
Painting	96%
Lady	92%
Art	88%
Self-portrait	86%
Visual Arts	76%
Illustration	50%

Faces

Joy	**Very unlikely**
Sorrow	**Very unlikely**
Anger	**Very unlikely**
Surprise	**Very unlikely**
Exposed	**Very unlikely**
Blurred	**Very unlikely**
Headwear	**Very unlikely**

Roll: -3° Tilt: -5° Pan: -10°

Confidence 100%

Labels

Lady	96%
Portrait	93%
Painting	87%
Self-portrait	81%
Art	78%

Faces

Joy	**Very unlikely**
Sorrow	**Very unlikely**
Anger	**Very unlikely**
Surprise	**Very Unlikely**
Exposed	**Very unlikely**
Blurred	**Very unlikely**
Headwear	**Likely**

Roll: -2° Tilt: 4° Pan: 20°

Confidence 99%

Labels

Portrait	98%
Lady	95%
Painting	92%
Self-portrait	90%
Art	82%
Artist	65%
Visual Arts	59%
Jaw	57%
Smile	54%

Faces

Joy	**Very unlikely**
Sorrow	**Very unlikely**
Anger	**Very unlikely**
Surprise	**Very unlikely**
Exposed	**Very unlikely**
Blurred	**Very unlikely**
Headwear	**Very unlikely**

Roll: -2° Tilt: -6° Pan: 8°

Confidence 100%

Labels

Lady	96%
Portrait	96%
Painting	95%
Self-portrait	83%
Art	76%
Visual Arts	55%
Artist	53%
Modern Art	51%

Faces

Joy	**Very unlikely**
Sorrow	**Very unlikely**
Anger	**Very unlikely**
Surprise	**Very unlikely**
Exposed	**Very unlikely**
Blurred	**Very unlikely**
Headwear	**Very unlikely**

Roll: 0° Tilt: 3° Pan: -24°

Confidence 100%

Labels

Portrait	96%
Self-portrait	92%
Gentleman	87%
Painting	80%
Art	78%
Jaw	67%
Artist	61%

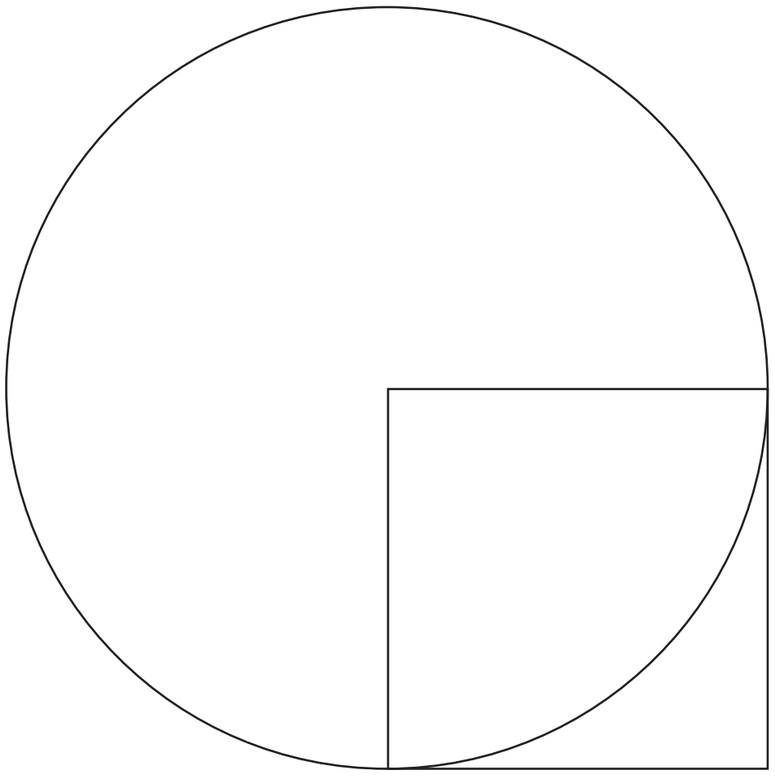
Faces

Joy	**Very unlikely**
Sorrow	**Very unlikely**
Anger	**Very unlikely**
Surprise	**Very unlikely**
Exposed	**Very unlikely**
Blurred	**Very unlikely**
Headwear	**Very unlikely**

Roll: -2° Tilt: -14° Pan: -4°

Confidence 97%

Werke
Abstrakt



Die Maschine lernt anhand von 7360 Werken mit der Stilbezeichnung „Abstrakt“. Das Datenset besteht aus Bildern beinahe aller Zeitepochen. Japanische Zen Schriften aus der Edo Periode, totale Simplifizierungen im Neo-Minimalismus oder Kandinsky‘s *Kompositionen*. Werke auf Papier, Ton oder Aluminium werden verglichen, im verzweifelten Versuch einen gemeinsamen Nenner zu finden und eine Einschätzung zu schaffen, was der Mensch mit abstrakter Kunst versucht zu kommunizieren.

Die Maschine blickt nicht auf Objekte, die sie mit ihrer technischen Perfektion rekreieren könnte, wie ein Fotoapparat, der keinerlei eigene Interpretationsentscheidungen trifft, wenn er ein Objekt abbildet. Sie erschafft stattdessen den Blick, den tausende Künstler zu vereinen scheint, in ihrem Ansatz die Konstante in diesen variablen Werken zu finden.

Die Maschine findet auf ihrer Suche nach Mustern und Gemeinsamkeiten das, was menschliches Betrachten von Dingen, von der optischen Realität unterscheidet. Sie scheitert in ihrem Versuch die tausenden unterschiedlichen Objekte zu erkennen und siegt in ihrem Streben nach dem Finden des gemeinsamen Nenners der abstrakten Sicht- und Darstellungsweise der Menschen.

—
Seed Unbekannt
FP16 & FP32 bei 8211 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8224 Bildern
512^2 Pixel

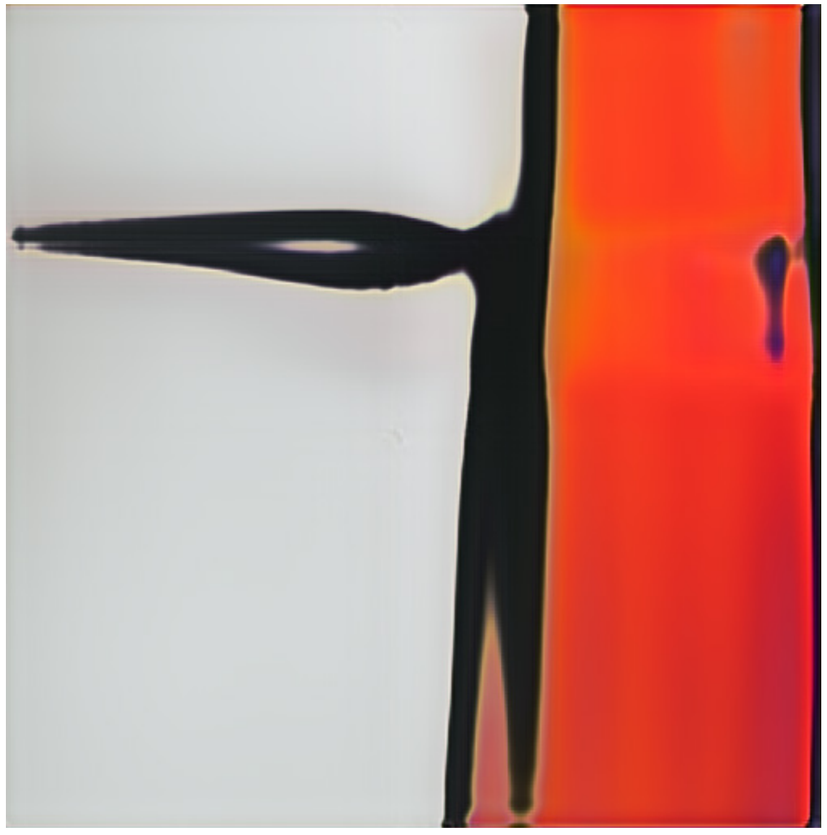
—
Seed Unbekannt
FP16 & FP32 bei 8326 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8326 Bildern
512^2 Pixel

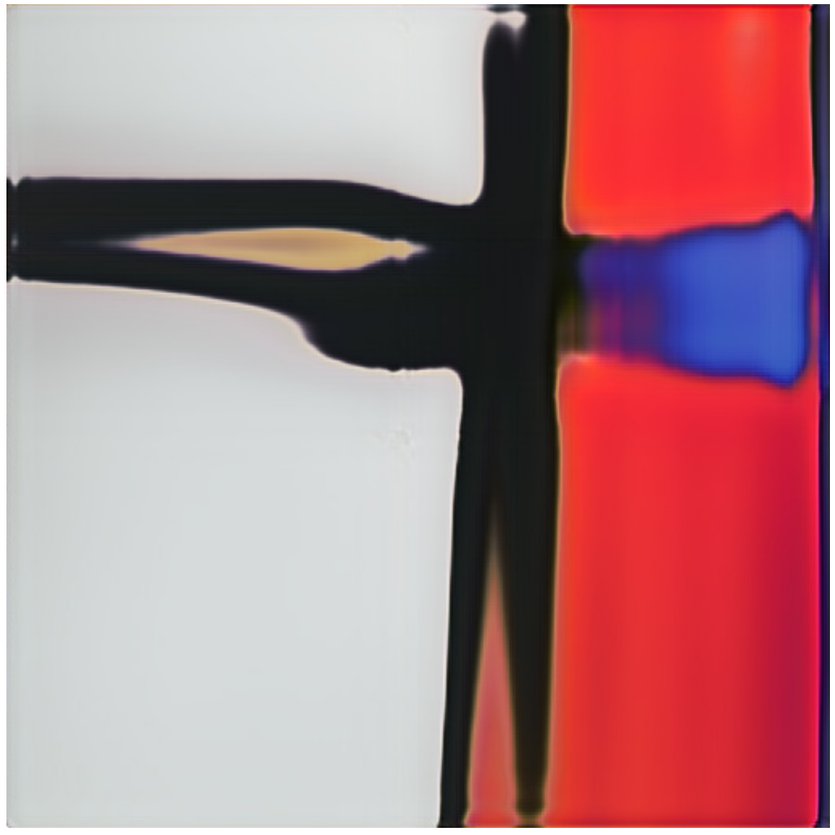
—
Seed Unbekannt
FP16 & FP32 bei 8326 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8326 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8305 Bildern
512^2 Pixel



—
Seed Unbekannt
FP16 & FP32 bei 8306 Bildern
512^2 Pixel

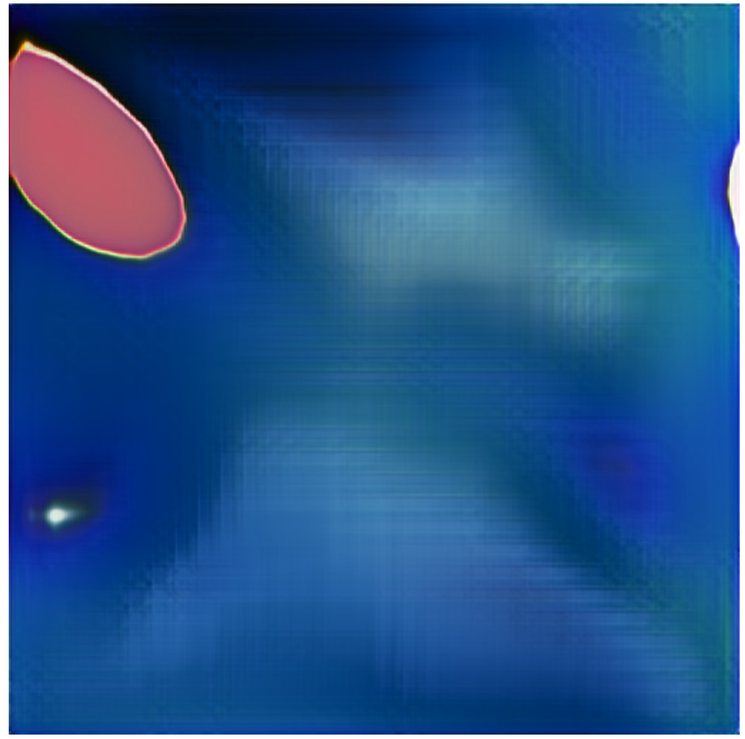


—
Seed Unbekannt
FP16 & FP32 bei 8307 Bildern
512^2 Pixel

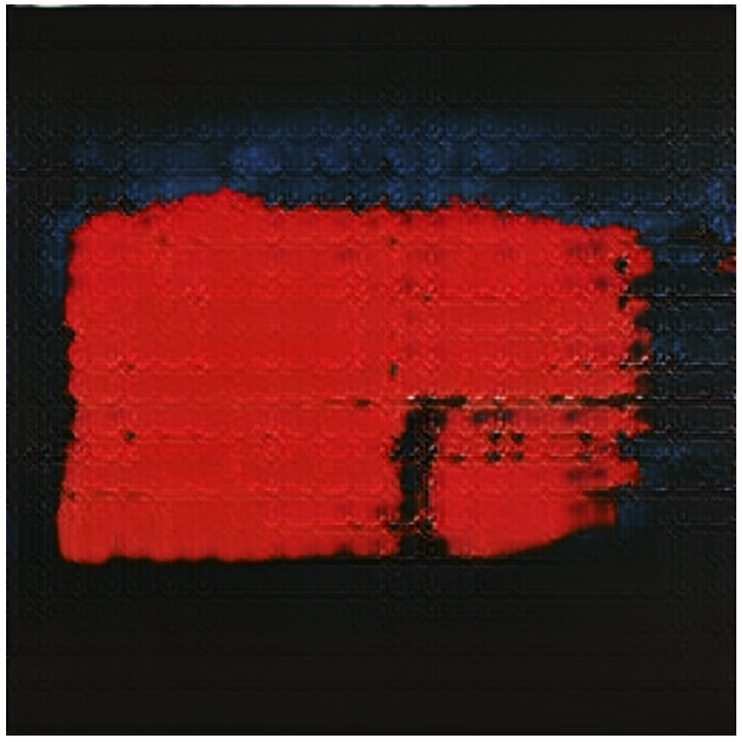
—

Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel

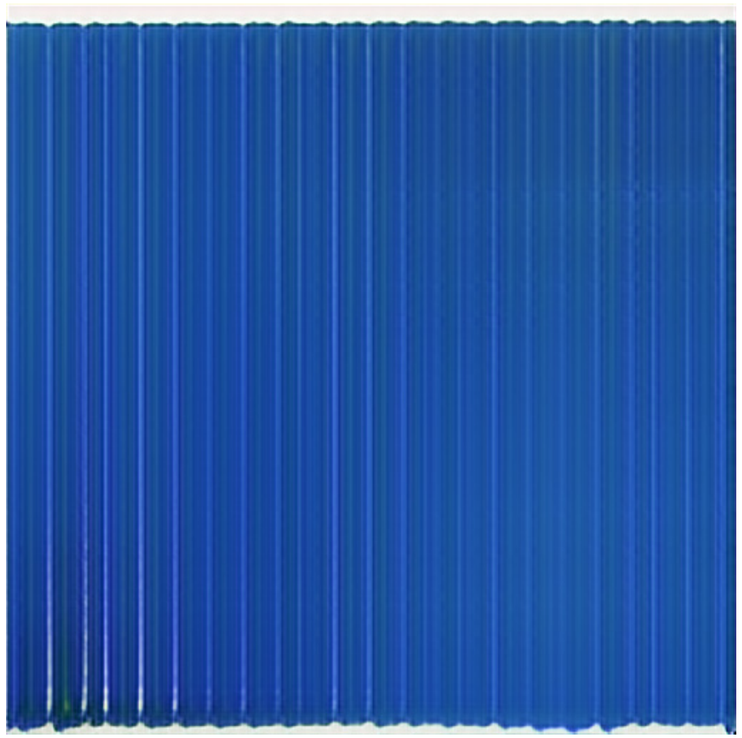
—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel



—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel



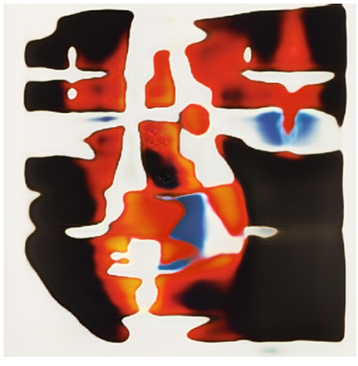
—
Seed Unbekannt
FP16 & FP32 bei 8326 Bildern
512^2 Pixel

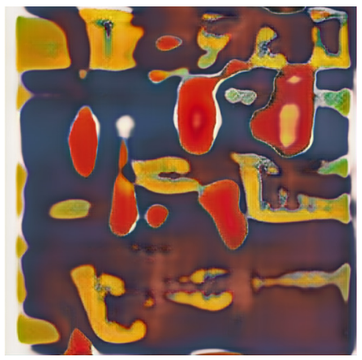


—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel

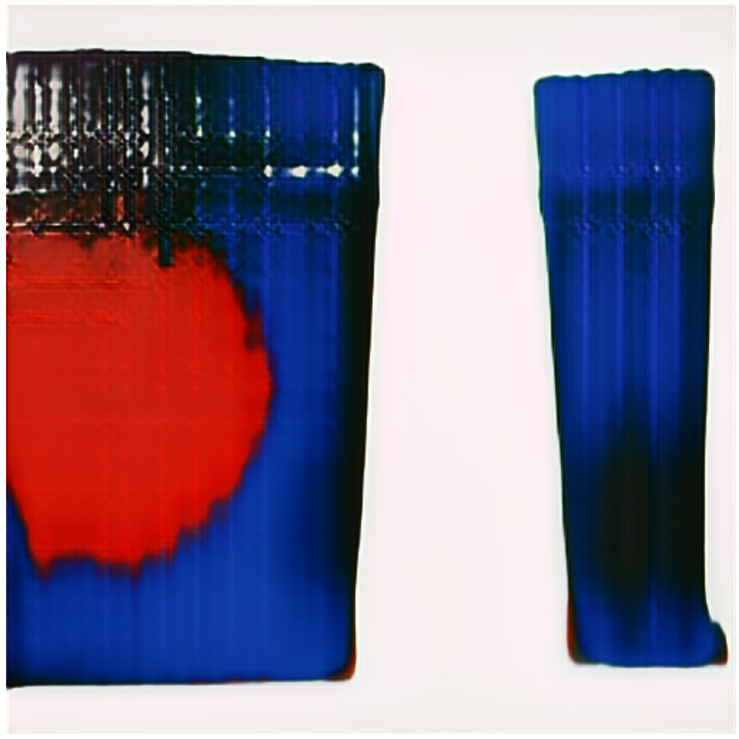




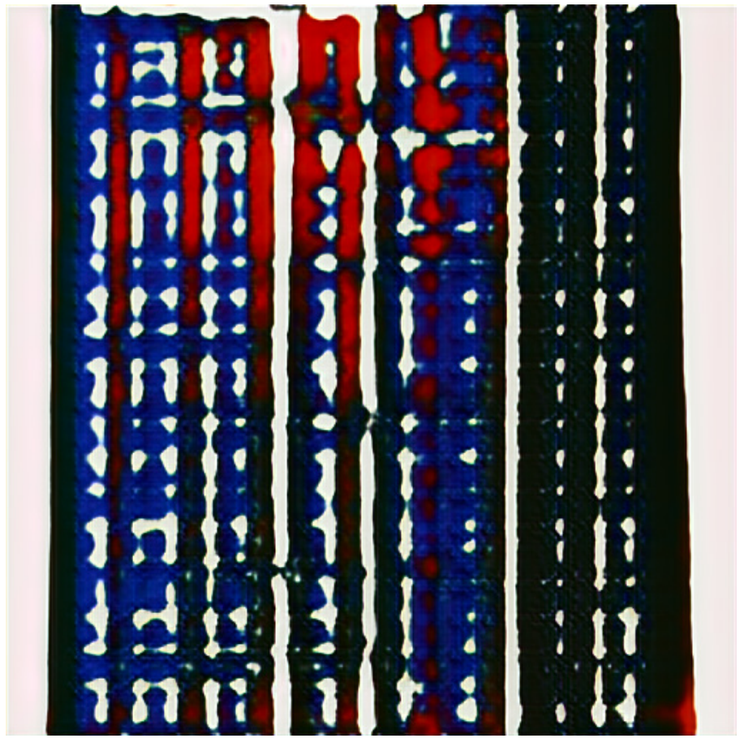



—
Seed Unbekannt
FP16 & FP32 bei 8209, 8211, 8214, 8239, 8224 & 8254 Bildern
512^2 Pixel

Schrittweise Entwicklung einer Lerneinheit. Die Resultate veranschaulichen den Lernprozess. Sie sind an unterschiedlichen Zeitpunkten des gleichen Seed's entstanden und demonstrieren die explorative Natur des Generators.

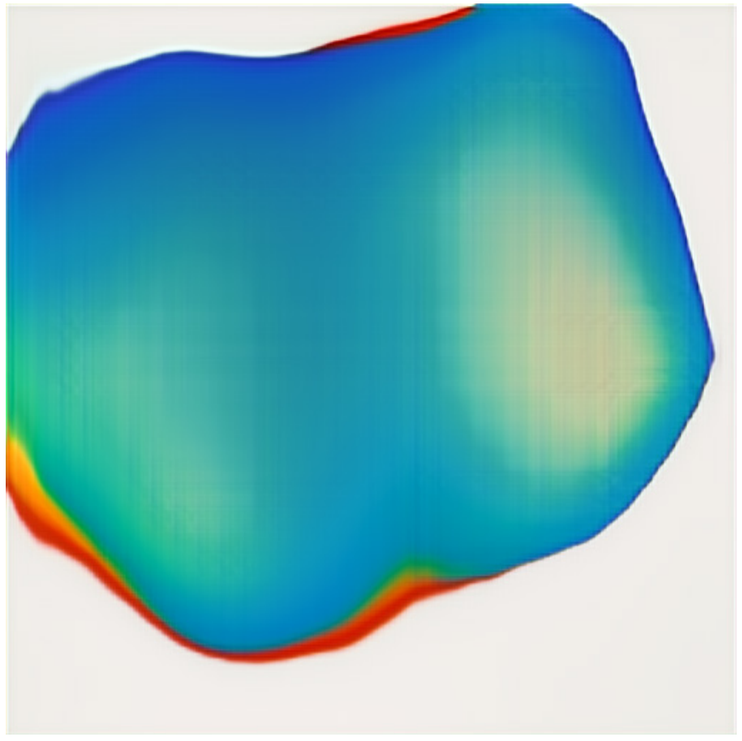


—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel

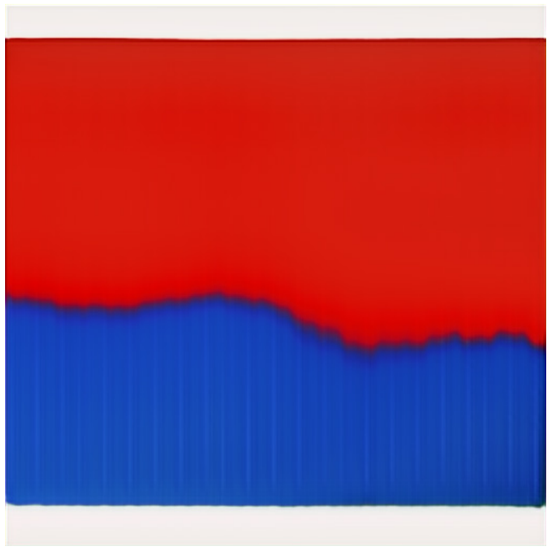


—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel

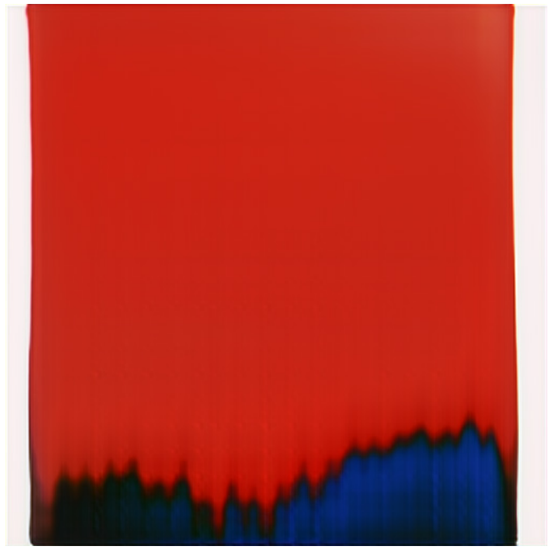
—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel



—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel



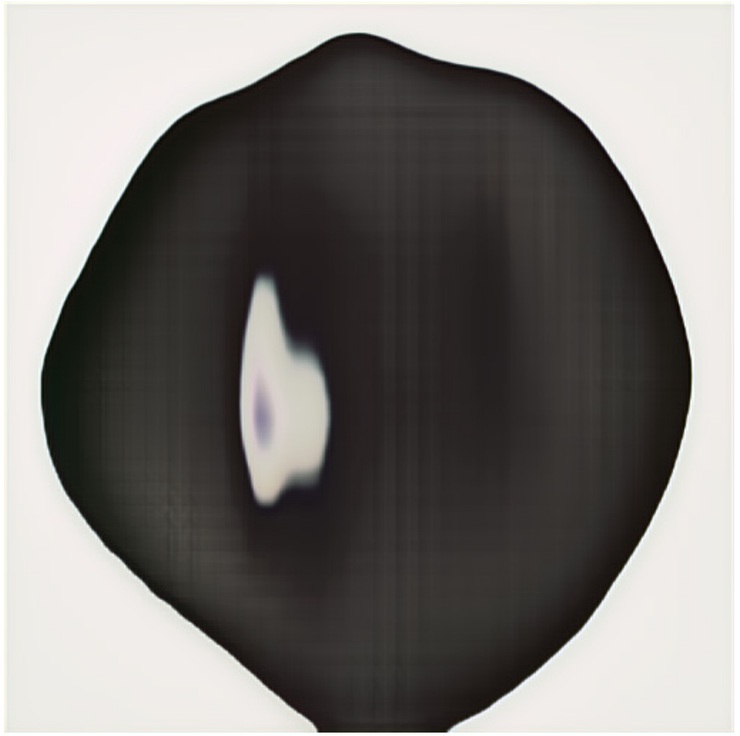
—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel
—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel



—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel
—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel

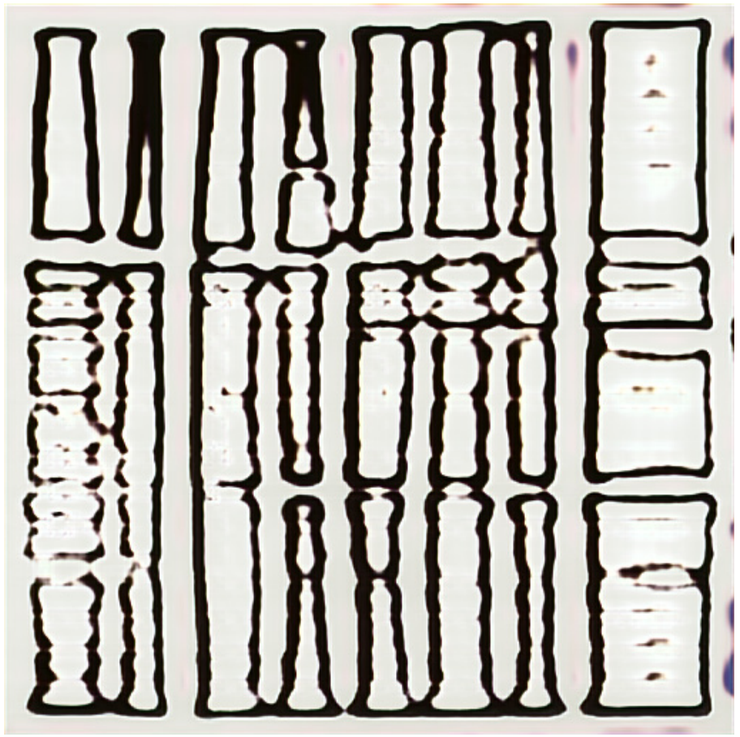
—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel
—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel
—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel

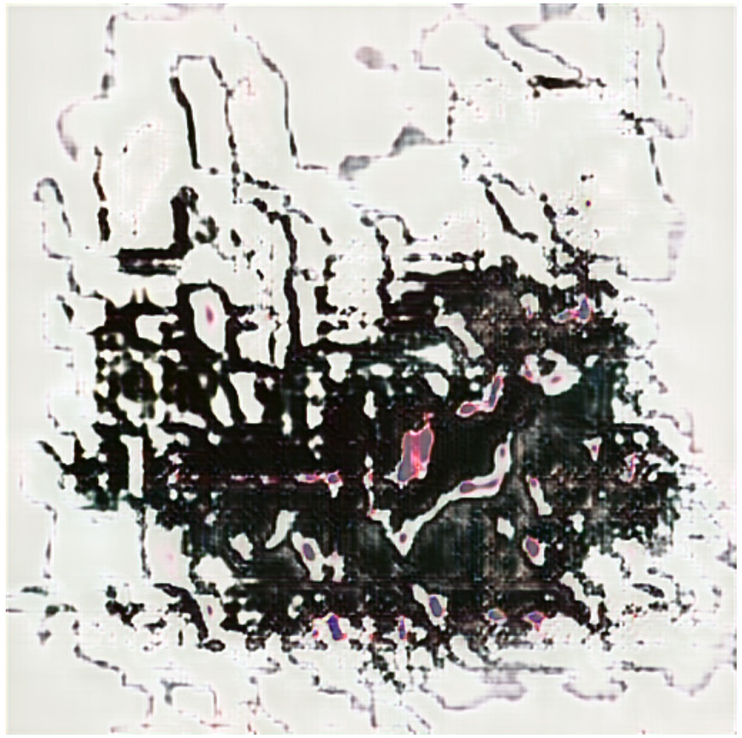


—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel



Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel



Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel

—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel

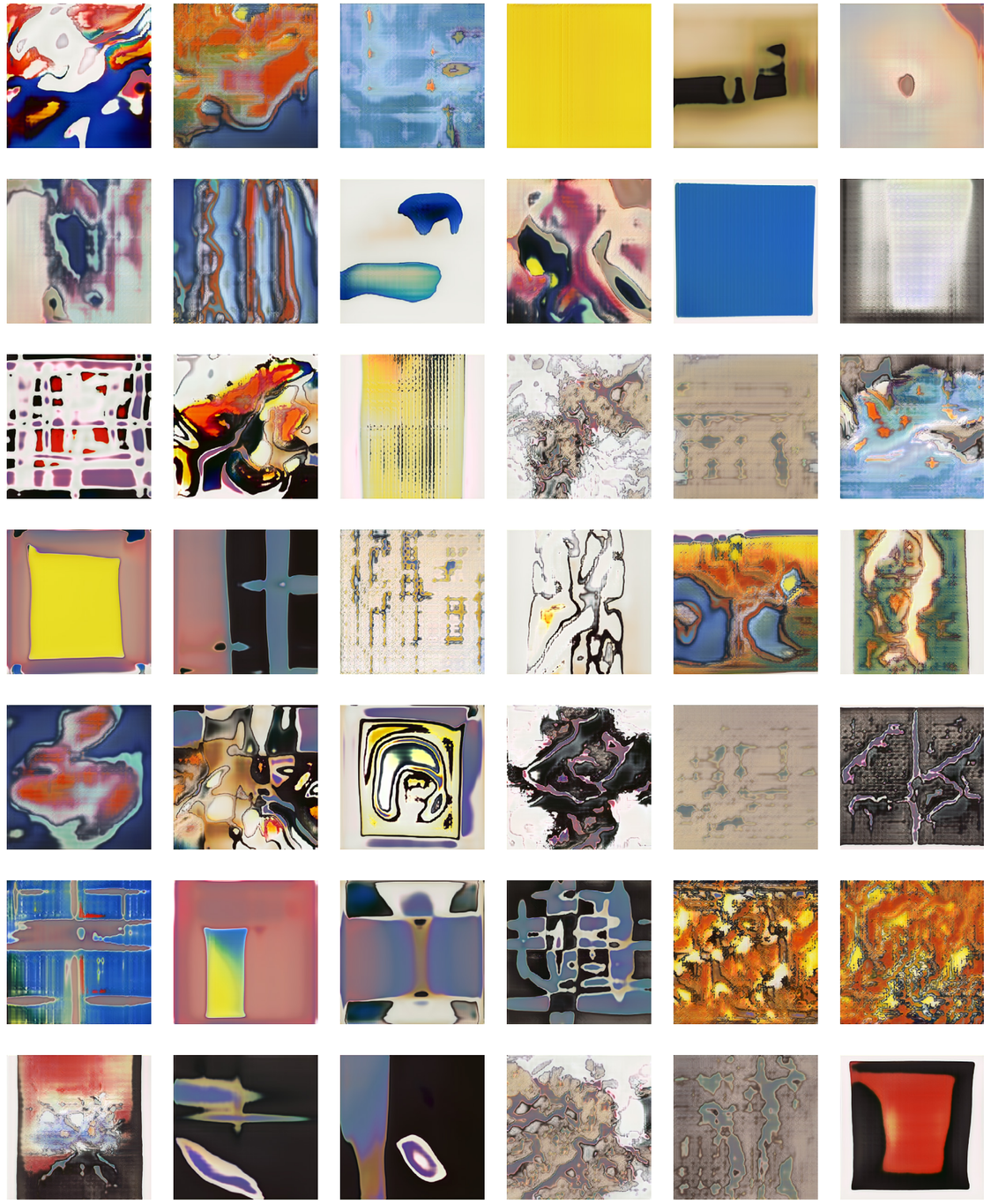
—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel

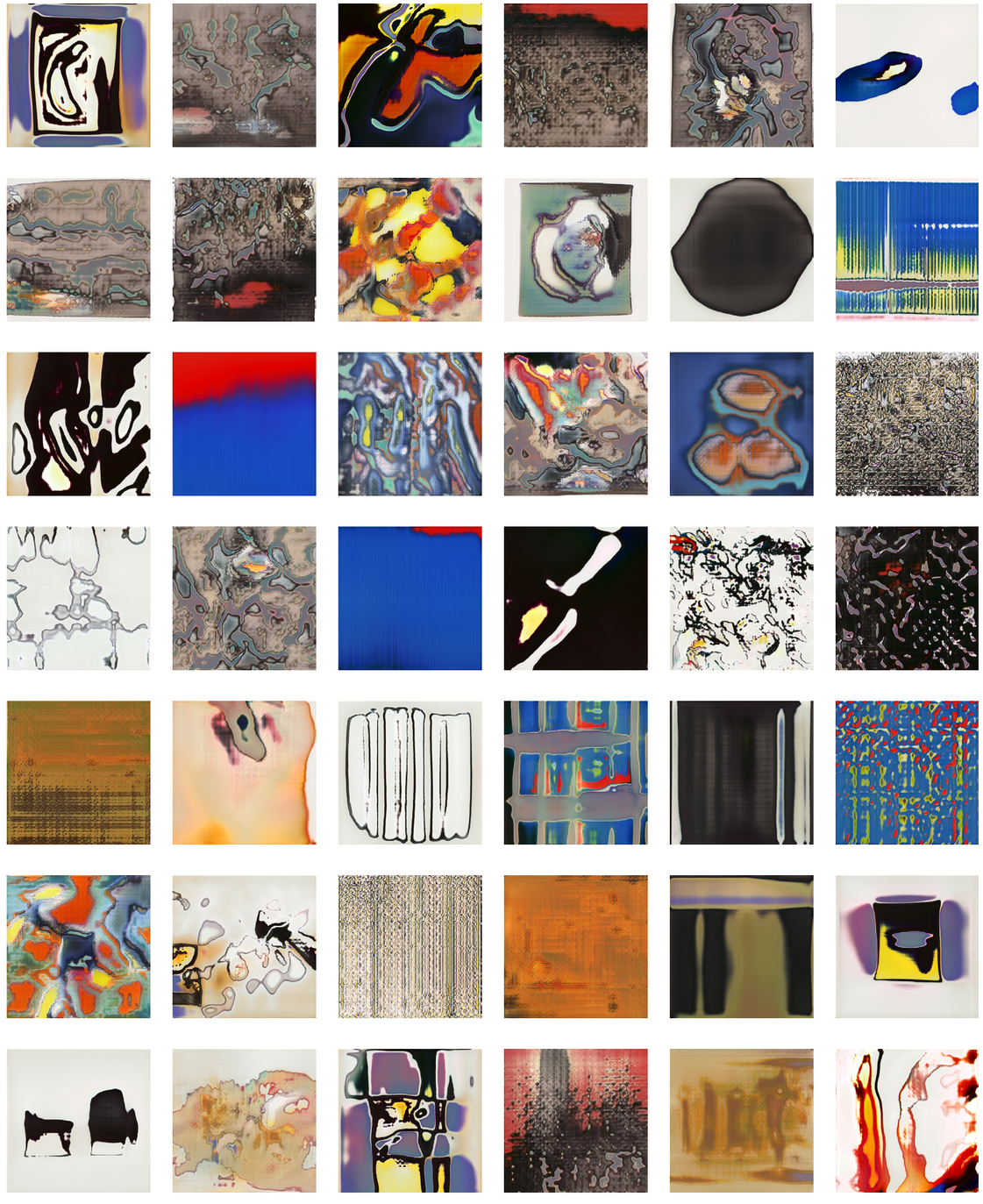
—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel

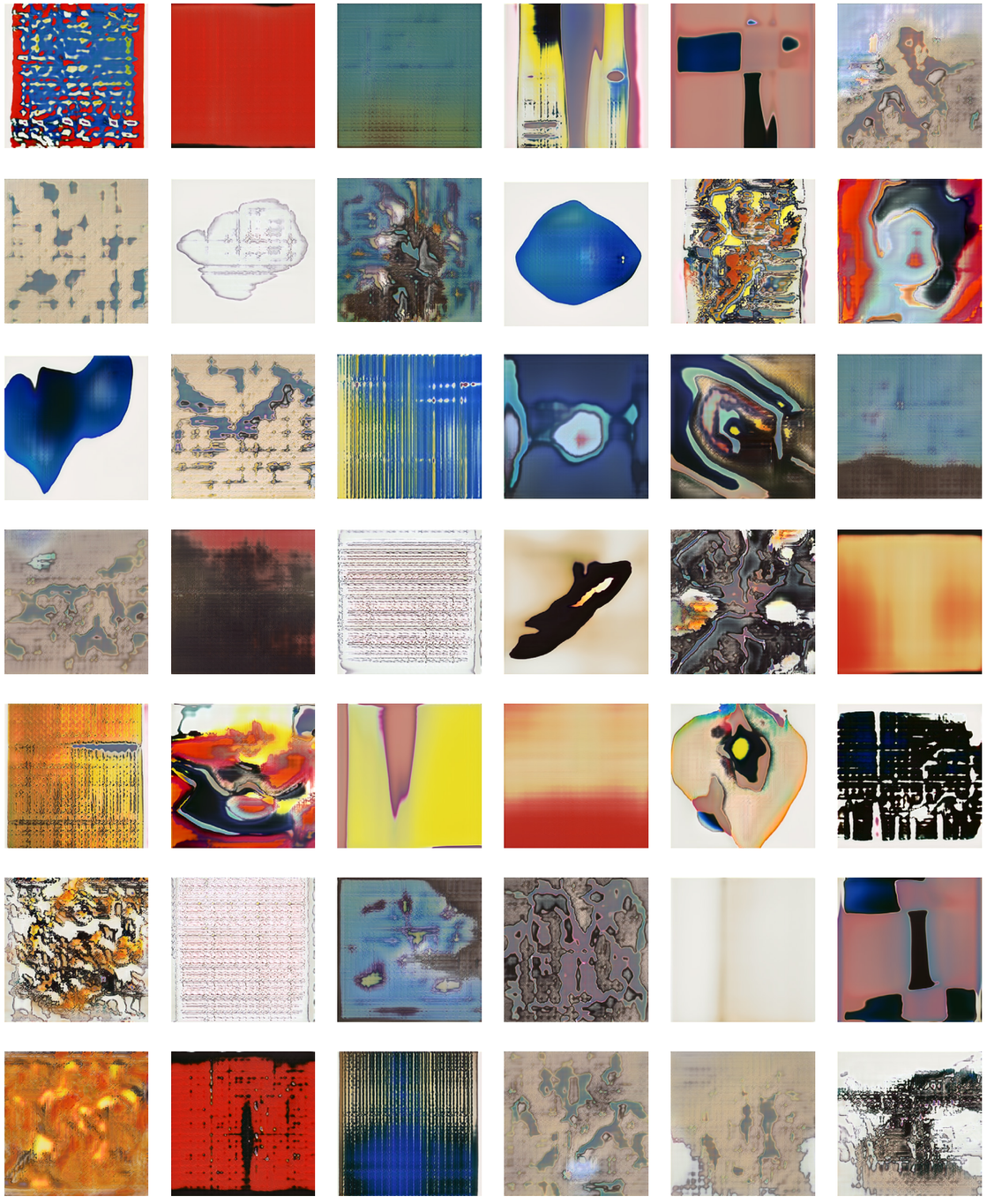
—
Seed Unbekannt
FP16 & FP32 bei 8400 Bildern
512^2 Pixel

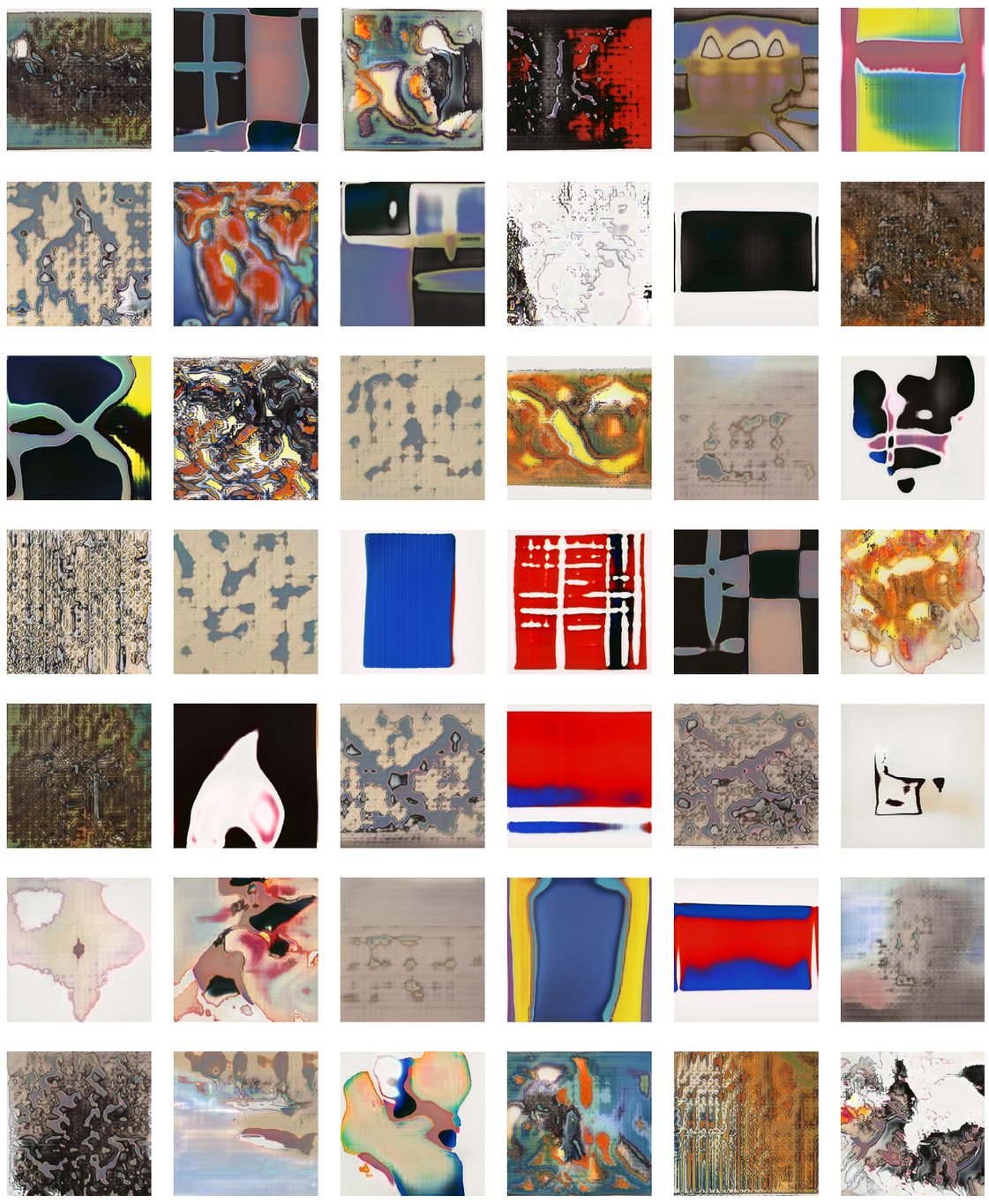
—

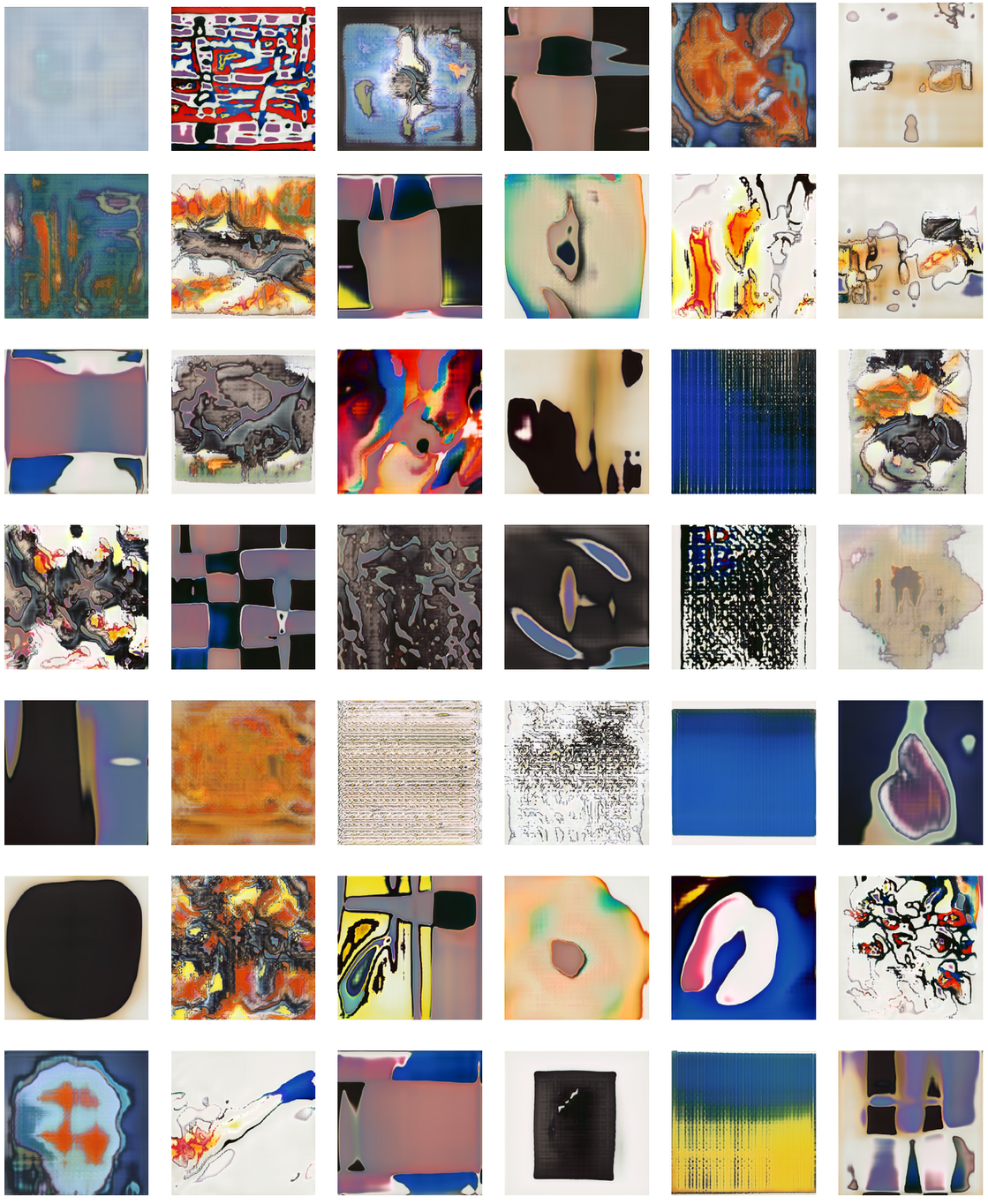
Folgende Seiten
Seed v. l. n. r. „649“ bis „0“
FP16 & FP32 bei 8400 Bildern
512^2 Pixel

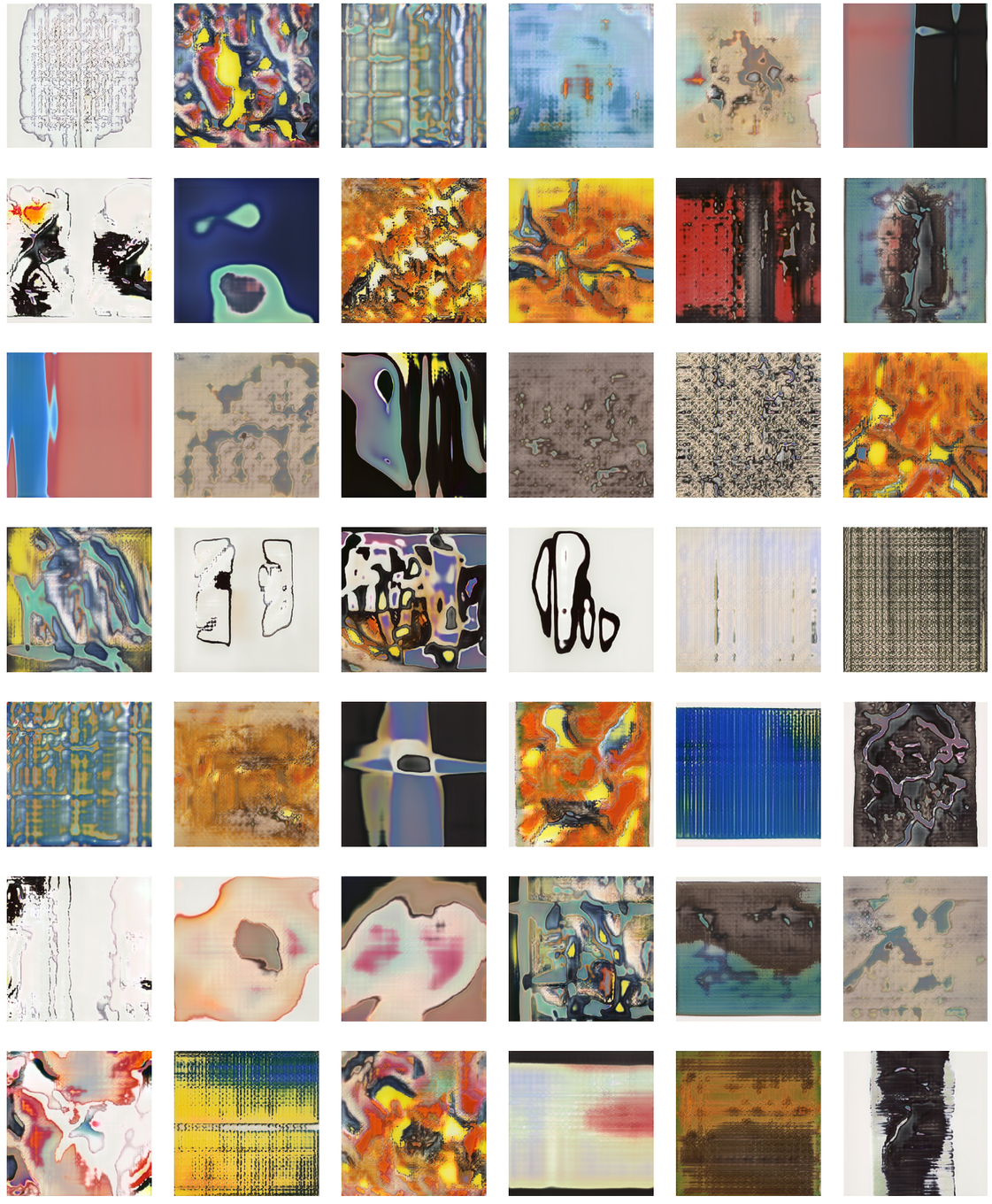


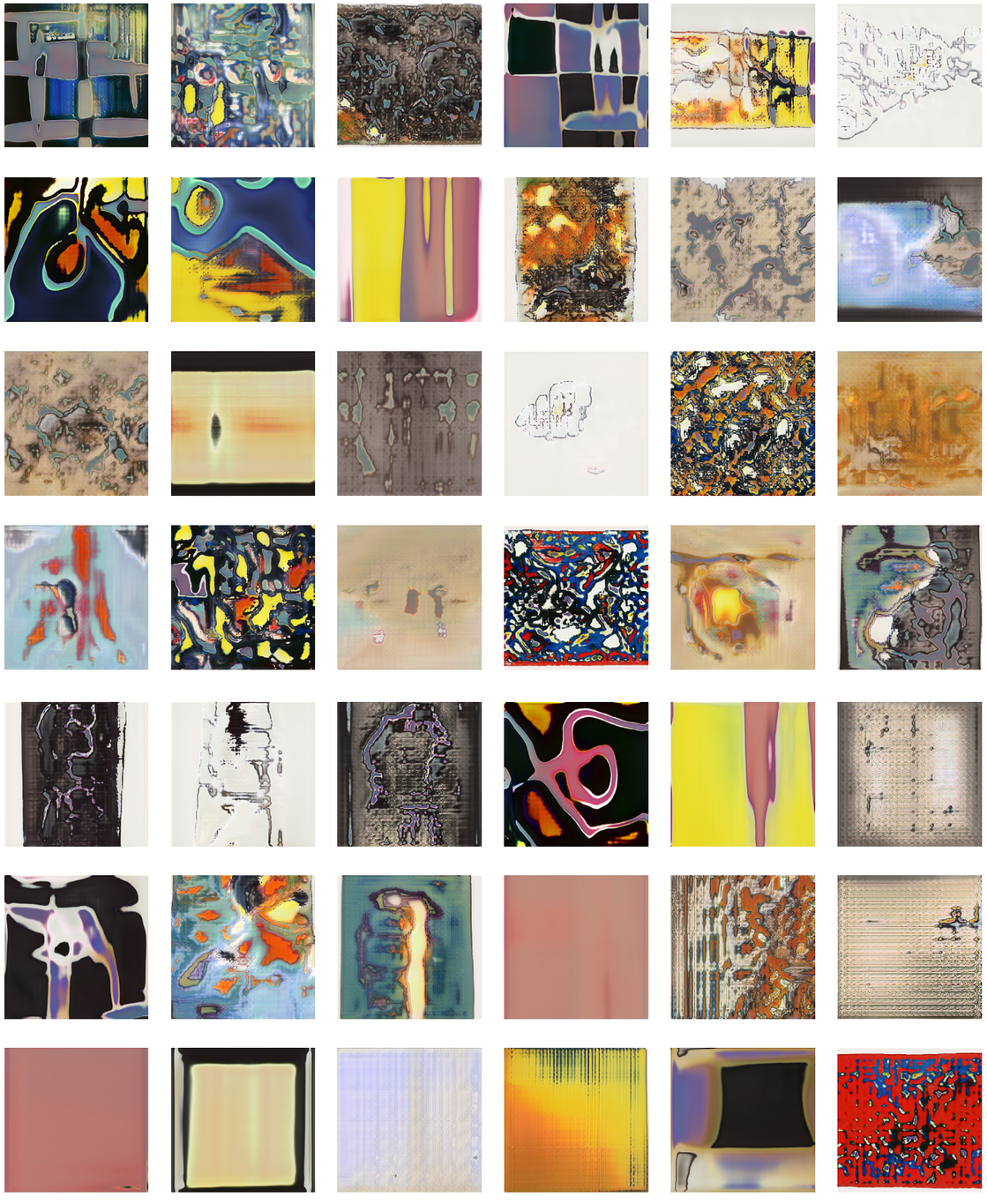


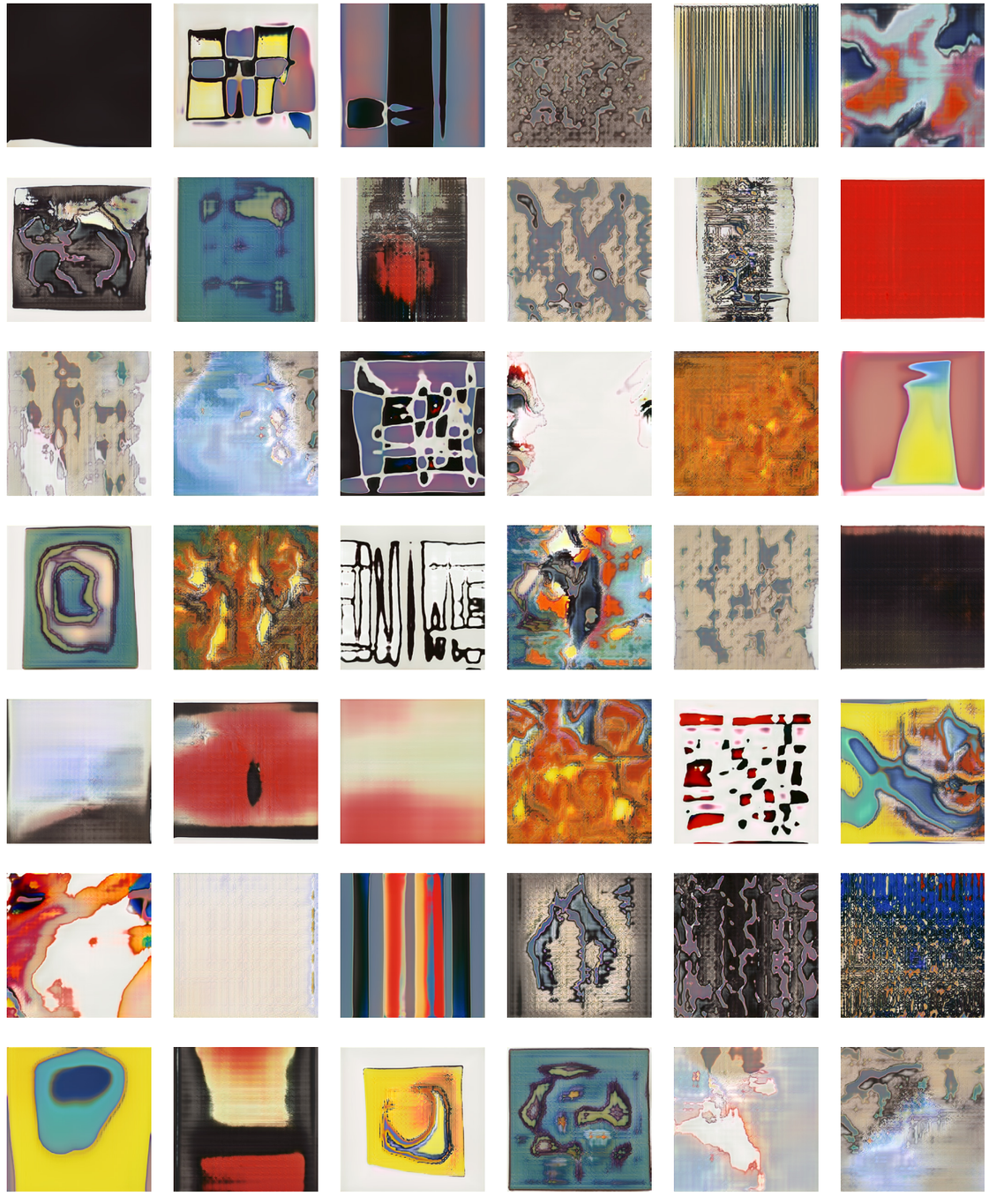


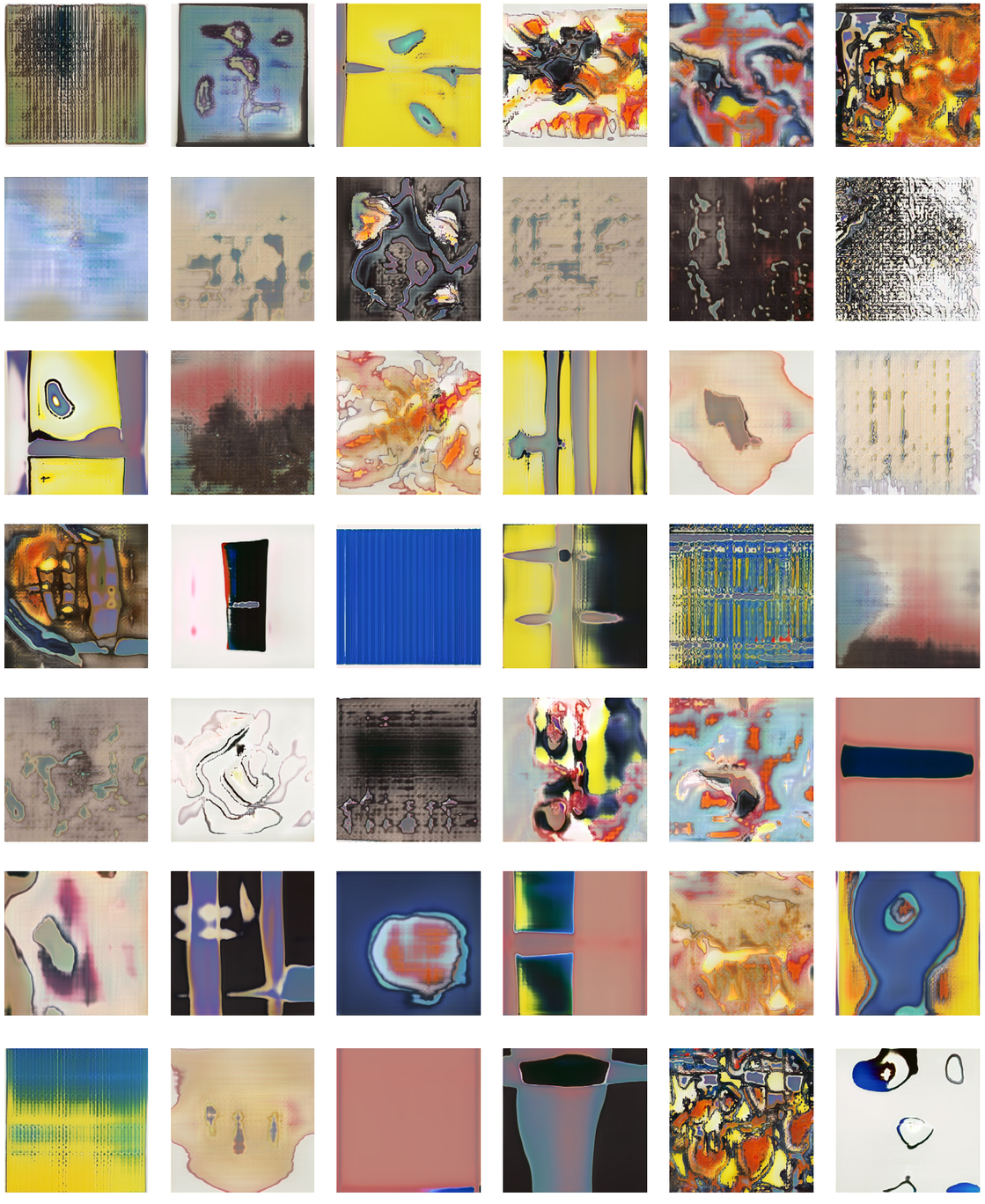


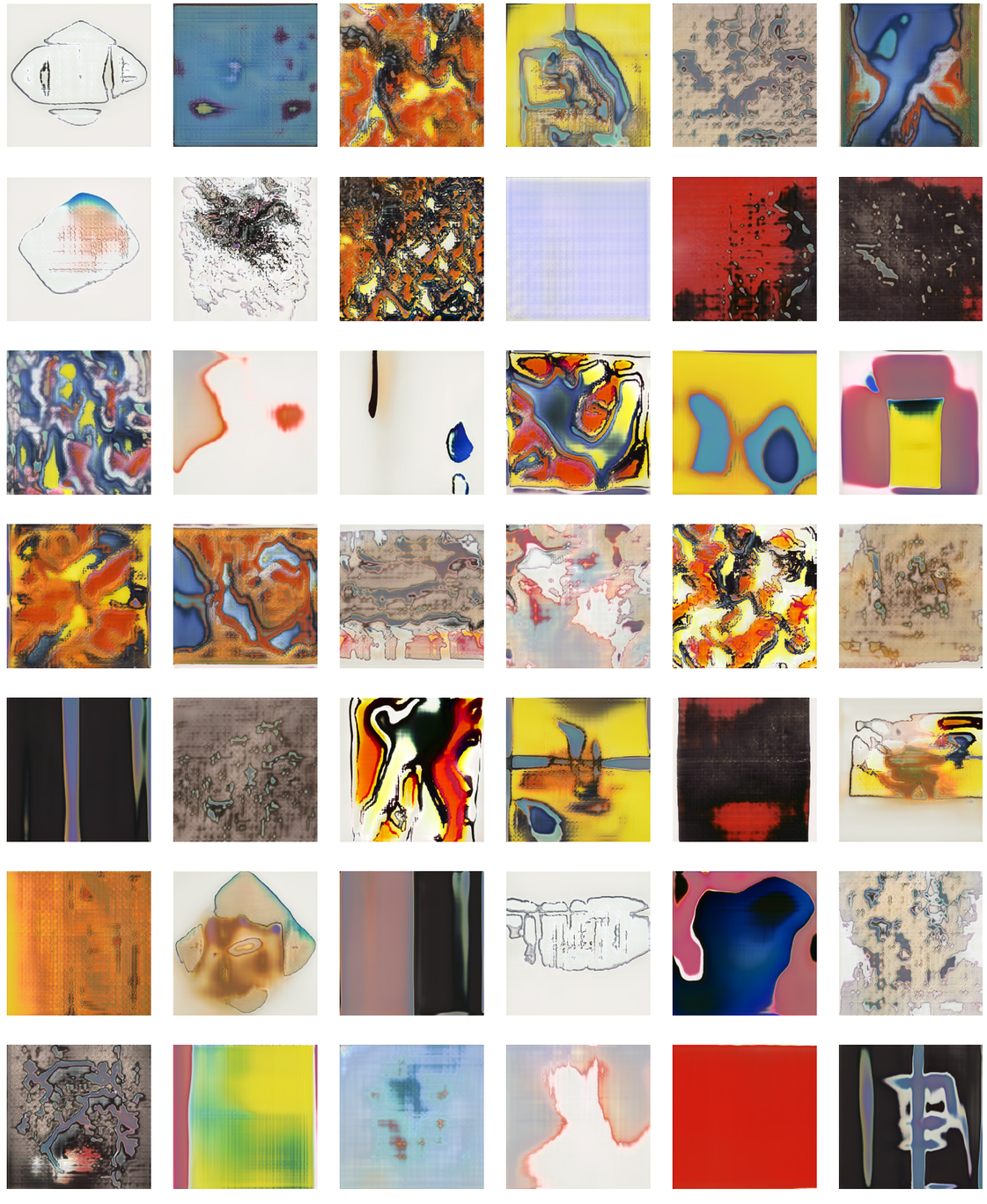












Protokoll eines Traums

Während der Lernprozess des Menschen's nur eingeschränkt oder stark abstrahiert zu visualisieren ist, ermöglichen uns Computer ihren Lernprozess zu überwachen. Die nachfolgenden Textpassagen sind Logbucheinträge des Computers auf seiner Suche nach Mustern. Man erkennt wie das GAN die Auflösung der Bilder erweitert und der Algorithmus mit dem exponentiellen Wachstum des Informationsumgangs immer länger braucht, um Muster und visuelle Prinzipien zu entdecken.

Abstrakt

```
Initializing TensorFlow...
Running train.train_progressive_gan()...
Streaming data using dataset.TFRecordDataset...
Dataset shape = [3, 512, 512]
Dynamic range = [0, 255]
Label size    = 0
Constructing networks...
```

G	Params	OutputShape	WeightShape
latents_in	-	(?, 512)	-
labels_in	-	(?, 0)	-
lod	-	()	-
4x4/PixelNorm	-	(?, 512)	-
4x4/Dense	4194816	(?, 512, 4, 4)	(512, 8192)
4x4/Conv	2359808	(?, 512, 4, 4)	(3, 3, 512, 512)
ToRGB_lod7	1539	(?, 3, 4, 4)	(1, 1, 512, 3)
8x8/Conv0_up	2359808	(?, 512, 8, 8)	(3, 3, 512, 512)
8x8/Conv1	2359808	(?, 512, 8, 8)	(3, 3, 512, 512)
ToRGB_lod6	1539	(?, 3, 8, 8)	(1, 1, 512, 3)
Upscale2D	-	(?, 3, 8, 8)	-
Grow_lod6	-	(?, 3, 8, 8)	-
16x16/Conv0_up	2359808	(?, 512, 16, 16)	(3, 3, 512, 512)
16x16/Conv1	2359808	(?, 512, 16, 16)	(3, 3, 512, 512)
ToRGB_lod5	1539	(?, 3, 16, 16)	(1, 1, 512, 3)
Upscale2D_1	-	(?, 3, 16, 16)	-
Grow_lod5	-	(?, 3, 16, 16)	-
32x32/Conv0_up	2359808	(?, 512, 32, 32)	(3, 3, 512, 512)
32x32/Conv1	2359808	(?, 512, 32, 32)	(3, 3, 512, 512)
ToRGB_lod4	1539	(?, 3, 32, 32)	(1, 1, 512, 3)
Upscale2D_2	-	(?, 3, 32, 32)	-
Grow_lod4	-	(?, 3, 32, 32)	-
64x64/Conv0_up	1179904	(?, 256, 64, 64)	(3, 3, 256, 512)
64x64/Conv1	590080	(?, 256, 64, 64)	(3, 3, 256, 256)
ToRGB_lod3	771	(?, 3, 64, 64)	(1, 1, 256, 3)
Upscale2D_3	-	(?, 3, 64, 64)	-
Grow_lod3	-	(?, 3, 64, 64)	-
128x128/Conv0_up	295040	(?, 128, 128, 128)	(3, 3, 128, 256)
128x128/Conv1	147584	(?, 128, 128, 128)	(3, 3, 128, 128)
ToRGB_lod2	387	(?, 3, 128, 128)	(1, 1, 128, 3)
Upscale2D_4	-	(?, 3, 128, 128)	-
Grow_lod2	-	(?, 3, 128, 128)	-
256x256/Conv0_up	73792	(?, 64, 256, 256)	(3, 3, 64, 128)
256x256/Conv1	36928	(?, 64, 256, 256)	(3, 3, 64, 64)
ToRGB_lod1	195	(?, 3, 256, 256)	(1, 1, 64, 3)
Upscale2D_5	-	(?, 3, 256, 256)	-
Grow_lod1	-	(?, 3, 256, 256)	-
512x512/Conv0_up	18464	(?, 32, 512, 512)	(3, 3, 32, 64)
512x512/Conv1	9248	(?, 32, 512, 512)	(3, 3, 32, 32)
ToRGB_lod0	99	(?, 3, 512, 512)	(1, 1, 32, 3)
Upscale2D_6	-	(?, 3, 512, 512)	-
Grow_lod0	-	(?, 3, 512, 512)	-
images_out	-	(?, 3, 512, 512)	-
---	---	---	---
Total	23072120		

D	Params	OutputShape	WeightShape
images_in	-	(?, 3, 512, 512)	-
lod	-	()	-
FromRGB_lod0	128	(?, 32, 512, 512)	(1, 1, 3, 32)
512x512/Conv0	9248	(?, 32, 512, 512)	(3, 3, 32, 32)
512x512/Conv1_down	18496	(?, 64, 256, 256)	(3, 3, 32, 64)
Downscale2D	-	(?, 3, 256, 256)	-
FromRGB_lod1	256	(?, 64, 256, 256)	(1, 1, 3, 64)
Grow_lod0	-	(?, 64, 256, 256)	-
256x256/Conv0	36928	(?, 64, 256, 256)	(3, 3, 64, 64)
256x256/Conv1_down	73856	(?, 128, 128, 128)	(3, 3, 64, 128)
Downscale2D_1	-	(?, 3, 128, 128)	-
FromRGB_lod2	512	(?, 128, 128, 128)	(1, 1, 3, 128)
Grow_lod1	-	(?, 128, 128, 128)	-
128x128/Conv0	147584	(?, 128, 128, 128)	(3, 3, 128, 128)
128x128/Conv1_down	295168	(?, 256, 64, 64)	(3, 3, 128, 256)
Downscale2D_2	-	(?, 3, 64, 64)	-
FromRGB_lod3	1024	(?, 256, 64, 64)	(1, 1, 3, 256)
Grow_lod2	-	(?, 256, 64, 64)	-
64x64/Conv0	590080	(?, 256, 64, 64)	(3, 3, 256, 256)
64x64/Conv1_down	1180160	(?, 512, 32, 32)	(3, 3, 256, 512)
Downscale2D_3	-	(?, 3, 32, 32)	-
FromRGB_lod4	2048	(?, 512, 32, 32)	(1, 1, 3, 512)
Grow_lod3	-	(?, 512, 32, 32)	-
32x32/Conv0	2359808	(?, 512, 32, 32)	(3, 3, 512, 512)
32x32/Conv1_down	2359808	(?, 512, 16, 16)	(3, 3, 512, 512)
Downscale2D_4	-	(?, 3, 16, 16)	-
FromRGB_lod5	2048	(?, 512, 16, 16)	(1, 1, 3, 512)
Grow_lod4	-	(?, 512, 16, 16)	-
16x16/Conv0	2359808	(?, 512, 16, 16)	(3, 3, 512, 512)
16x16/Conv1_down	2359808	(?, 512, 8, 8)	(3, 3, 512, 512)
Downscale2D_5	-	(?, 3, 8, 8)	-
FromRGB_lod6	2048	(?, 512, 8, 8)	(1, 1, 3, 512)
Grow_lod5	-	(?, 512, 8, 8)	-
8x8/Conv0	2359808	(?, 512, 8, 8)	(3, 3, 512, 512)
8x8/Conv1_down	2359808	(?, 512, 4, 4)	(3, 3, 512, 512)
Downscale2D_6	-	(?, 3, 4, 4)	-
FromRGB_lod7	2048	(?, 512, 4, 4)	(1, 1, 3, 512)
Grow_lod6	-	(?, 512, 4, 4)	-
4x4/MinibatchStddev	-	(?, 1, 4, 4)	-
4x4/Conv	2364416	(?, 512, 4, 4)	(3, 3, 513, 512)
4x4/Dense0	4194816	(?, 512)	(8192, 512)
4x4/Dense1	513	(?, 1)	(512, 1)
scores_out	-	(?, 1)	-
labels_out	-	(?, 0)	-
---	---	---	---
Total	23080225		

```
Building TensorFlow graph...
Setting up snapshot image grid...
Setting up result dir...
Saving results to results/012-pgan-from_images-preset-v2-1gpu-fp16
Training...
tick 1     kimg 160.3    lod 7.00  minibatch 128  time 1m 54s       sec/tick 113.5   sec/kimg 0.71    maintenance 48.2
```

tick 2	kimg 320.5	lod 7.00	minibatch 128	time 3m 51s	sec/tick 107.0	sec/kimg 0.67	maintenance 10.4
tick 3	kimg 480.8	lod 7.00	minibatch 128	time 5m 38s	sec/tick 106.7	sec/kimg 0.67	maintenance 0.3
tick 4	kimg 621.1	lod 6.97	minibatch 128	time 7m 25s	sec/tick 106.8	sec/kimg 0.76	maintenance 0.3
tick 5	kimg 761.3	lod 6.73	minibatch 128	time 10m 04s	sec/tick 158.7	sec/kimg 1.13	maintenance 0.4
tick 6	kimg 901.6	lod 6.50	minibatch 128	time 12m 43s	sec/tick 158.9	sec/kimg 1.13	maintenance 0.3
tick 7	kimg 1041.9	lod 6.26	minibatch 128	time 15m 23s	sec/tick 158.7	sec/kimg 1.13	maintenance 0.5
tick 8	kimg 1182.2	lod 6.03	minibatch 128	time 18m 02s	sec/tick 158.8	sec/kimg 1.13	maintenance 0.3
tick 9	kimg 1322.5	lod 6.00	minibatch 128	time 20m 33s	sec/tick 150.7	sec/kimg 1.07	maintenance 0.5
tick 10	kimg 1462.8	lod 6.00	minibatch 128	time 23m 03s	sec/tick 149.2	sec/kimg 1.06	maintenance 0.4
tick 11	kimg 1603.1	lod 6.00	minibatch 128	time 25m 35s	sec/tick 149.5	sec/kimg 1.07	maintenance 2.6
tick 12	kimg 1743.4	lod 6.00	minibatch 128	time 28m 05s	sec/tick 150.0	sec/kimg 1.07	maintenance 0.4
tick 13	kimg 1863.7	lod 5.89	minibatch 128	time 31m 34s	sec/tick 208.2	sec/kimg 1.73	maintenance 0.5
tick 14	kimg 1984.0	lod 5.69	minibatch 128	time 36m 00s	sec/tick 265.8	sec/kimg 2.21	maintenance 0.5
tick 15	kimg 2104.3	lod 5.49	minibatch 128	time 40m 25s	sec/tick 264.7	sec/kimg 2.20	maintenance 0.3
tick 16	kimg 2224.6	lod 5.29	minibatch 128	time 44m 51s	sec/tick 265.3	sec/kimg 2.20	maintenance 0.4
tick 17	kimg 2345.0	lod 5.09	minibatch 128	time 49m 16s	sec/tick 265.0	sec/kimg 2.20	maintenance 0.4
tick 18	kimg 2465.3	lod 5.00	minibatch 128	time 53m 37s	sec/tick 260.6	sec/kimg 2.17	maintenance 0.3
tick 19	kimg 2585.6	lod 5.00	minibatch 128	time 57m 54s	sec/tick 256.9	sec/kimg 2.13	maintenance 0.4
tick 20	kimg 2705.9	lod 5.00	minibatch 128	time 1h 02m 12s	sec/tick 256.9	sec/kimg 2.13	maintenance 0.4
tick 21	kimg 2826.2	lod 5.00	minibatch 128	time 1h 06m 29s	sec/tick 256.9	sec/kimg 2.14	maintenance 0.9
tick 22	kimg 2946.6	lod 5.00	minibatch 128	time 1h 10m 47s	sec/tick 256.8	sec/kimg 2.13	maintenance 0.4
tick 23	kimg 3046.7	lod 4.92	minibatch 64	time 1h 17m 57s	sec/tick 430.3	sec/kimg 4.30	maintenance 0.3
tick 24	kimg 3146.8	lod 4.76	minibatch 64	time 1h 29m 17s	sec/tick 678.9	sec/kimg 6.78	maintenance 0.4
tick 25	kimg 3246.8	lod 4.59	minibatch 64	time 1h 40m 36s	sec/tick 679.4	sec/kimg 6.79	maintenance 0.3
tick 26	kimg 3346.9	lod 4.42	minibatch 64	time 1h 51m 56s	sec/tick 679.6	sec/kimg 6.79	maintenance 0.4
tick 27	kimg 3447.0	lod 4.26	minibatch 64	time 2h 03m 16s	sec/tick 679.6	sec/kimg 6.79	maintenance 0.3
tick 28	kimg 3547.1	lod 4.09	minibatch 64	time 2h 14m 36s	sec/tick 679.9	sec/kimg 6.79	maintenance 0.3
tick 29	kimg 3647.2	lod 4.00	minibatch 64	time 2h 25m 49s	sec/tick 672.8	sec/kimg 6.72	maintenance 0.3
tick 30	kimg 3747.3	lod 4.00	minibatch 64	time 2h 36m 54s	sec/tick 663.8	sec/kimg 6.63	maintenance 0.3
tick 31	kimg 3847.4	lod 4.00	minibatch 64	time 2h 47m 58s	sec/tick 663.5	sec/kimg 6.63	maintenance 0.9
tick 32	kimg 3947.5	lod 4.00	minibatch 64	time 2h 59m 02s	sec/tick 663.9	sec/kimg 6.63	maintenance 0.3
tick 33	kimg 4047.6	lod 4.00	minibatch 64	time 3h 10m 06s	sec/tick 663.8	sec/kimg 6.63	maintenance 0.4
tick 34	kimg 4147.7	lod 4.00	minibatch 64	time 3h 21m 11s	sec/tick 663.9	sec/kimg 6.63	maintenance 0.3
tick 35	kimg 4227.7	lod 3.95	minibatch 32	time 3h 33m 47s	sec/tick 756.5	sec/kimg 9.46	maintenance 0.4
tick 36	kimg 4307.7	lod 3.82	minibatch 32	time 3h 53m 38s	sec/tick 1190.1	sec/kimg 14.88	maintenance 0.4
tick 37	kimg 4387.7	lod 3.69	minibatch 32	time 4h 13m 29s	sec/tick 1190.3	sec/kimg 14.88	maintenance 0.3
tick 38	kimg 4467.7	lod 3.55	minibatch 32	time 4h 33m 19s	sec/tick 1190.5	sec/kimg 14.88	maintenance 0.4
tick 39	kimg 4547.7	lod 3.42	minibatch 32	time 4h 53m 10s	sec/tick 1190.1	sec/kimg 14.88	maintenance 0.4
tick 40	kimg 4627.7	lod 3.29	minibatch 32	time 5h 13m 01s	sec/tick 1190.5	sec/kimg 14.88	maintenance 0.4
tick 41	kimg 4707.7	lod 3.15	minibatch 32	time 5h 32m 52s	sec/tick 1190.4	sec/kimg 14.88	maintenance 0.8
tick 42	kimg 4787.7	lod 3.02	minibatch 32	time 5h 52m 43s	sec/tick 1191.1	sec/kimg 14.89	maintenance 0.4
tick 43	kimg 4867.7	lod 3.00	minibatch 32	time 6h 12m 05s	sec/tick 1161.1	sec/kimg 14.51	maintenance 0.4
tick 44	kimg 4947.7	lod 3.00	minibatch 32	time 6h 31m 21s	sec/tick 1155.8	sec/kimg 14.45	maintenance 0.3
tick 45	kimg 5027.7	lod 3.00	minibatch 32	time 6h 50m 36s	sec/tick 1155.0	sec/kimg 14.44	maintenance 0.3
tick 46	kimg 5107.7	lod 3.00	minibatch 32	time 7h 09m 51s	sec/tick 1154.0	sec/kimg 14.42	maintenance 0.3
tick 47	kimg 5187.7	lod 3.00	minibatch 32	time 7h 29m 06s	sec/tick 1154.6	sec/kimg 14.43	maintenance 0.4
tick 48	kimg 5267.7	lod 3.00	minibatch 32	time 7h 48m 21s	sec/tick 1154.6	sec/kimg 14.43	maintenance 0.3
tick 49	kimg 5347.7	lod 3.00	minibatch 32	time 8h 07m 36s	sec/tick 1154.6	sec/kimg 14.43	maintenance 0.4
tick 50	kimg 5407.7	lod 2.99	minibatch 16	time 8h 23m 59s	sec/tick 983.4	sec/kimg 16.38	maintenance 0.4
tick 51	kimg 5467.8	lod 2.89	minibatch 16	time 8h 54m 13s	sec/tick 1813.1	sec/kimg 30.20	maintenance 1.0
tick 52	kimg 5527.8	lod 2.79	minibatch 16	time 9h 24m 32s	sec/tick 1818.0	sec/kimg 30.28	maintenance 0.4
tick 53	kimg 5587.8	lod 2.69	minibatch 16	time 9h 54m 50s	sec/tick 1817.7	sec/kimg 30.28	maintenance 0.4
tick 54	kimg 5647.9	lod 2.59	minibatch 16	time 10h 25m 13s	sec/tick 1823.0	sec/kimg 30.37	maintenance 0.4
tick 55	kimg 5707.9	lod 2.49	minibatch 16	time 10h 55m 38s	sec/tick 1824.3	sec/kimg 30.39	maintenance 0.3
tick 56	kimg 5767.9	lod 2.39	minibatch 16	time 11h 26m 01s	sec/tick 1822.7	sec/kimg 30.36	maintenance 0.4
tick 57	kimg 5828.0	lod 2.29	minibatch 16	time 11h 56m 28s	sec/tick 1826.0	sec/kimg 30.42	maintenance 0.4
tick 58	kimg 5888.0	lod 2.19	minibatch 16	time 12h 26m 51s	sec/tick 1822.8	sec/kimg 30.36	maintenance 0.4
tick 59	kimg 5948.0	lod 2.09	minibatch 16	time 12h 57m 05s	sec/tick 1813.9	sec/kimg 30.22	maintenance 0.4
tick 60	kimg 6008.1	lod 2.00	minibatch 16	time 13h 27m 11s	sec/tick 1805.9	sec/kimg 30.08	maintenance 0.3
tick 61	kimg 6068.1	lod 2.00	minibatch 16	time 13h 56m 25s	sec/tick 1752.7	sec/kimg 29.20	maintenance 1.0
tick 62	kimg 6128.1	lod 2.00	minibatch 16	time 14h 25m 36s	sec/tick 1750.5	sec/kimg 29.16	maintenance 0.4
tick 63	kimg 6188.2	lod 2.00	minibatch 16	time 14h 54m 48s	sec/tick 1752.2	sec/kimg 29.19	maintenance 0.4
tick 64	kimg 6248.2	lod 2.00	minibatch 16	time 15h 24m 02s	sec/tick 1753.1	sec/kimg 29.20	maintenance 0.4
tick 65	kimg 6308.2	lod 2.00	minibatch 16	time 15h 53m 15s	sec/tick 1752.9	sec/kimg 29.20	maintenance 0.4
tick 66	kimg 6368.3	lod 2.00	minibatch 16	time 16h 22m 29s	sec/tick 1753.4	sec/kimg 29.21	maintenance 0.4

tick 67	kimg 6428.3	lod 2.00	minibatch 16	time 16h 51m 47s	sec/tick 1757.7	sec/kimg 29.28	maintenance 0.4
tick 68	kimg 6488.3	lod 2.00	minibatch 16	time 17h 20m 58s	sec/tick 1751.0	sec/kimg 29.17	maintenance 0.4
tick 69	kimg 6548.4	lod 2.00	minibatch 16	time 17h 50m 15s	sec/tick 1756.3	sec/kimg 29.26	maintenance 0.4
tick 70	kimg 6600.1	lod 2.00	minibatch 8	time 18h 15m 34s	sec/tick 1518.3	sec/kimg 29.34	maintenance 0.4
tick 71	kimg 6640.1	lod 1.93	minibatch 8	time 18h 55m 40s	sec/tick 2405.3	sec/kimg 60.13	maintenance 1.0
tick 72	kimg 6680.1	lod 1.87	minibatch 8	time 19h 35m 44s	sec/tick 2403.3	sec/kimg 60.08	maintenance 0.6
tick 73	kimg 6720.1	lod 1.80	minibatch 8	time 20h 15m 48s	sec/tick 2403.4	sec/kimg 60.09	maintenance 0.5
tick 74	kimg 6760.1	lod 1.73	minibatch 8	time 20h 55m 51s	sec/tick 2402.1	sec/kimg 60.05	maintenance 0.5
tick 75	kimg 6800.1	lod 1.67	minibatch 8	time 21h 35m 55s	sec/tick 2403.6	sec/kimg 60.09	maintenance 0.5
tick 76	kimg 6840.1	lod 1.60	minibatch 8	time 22h 15m 58s	sec/tick 2403.0	sec/kimg 60.07	maintenance 0.5
tick 77	kimg 6880.1	lod 1.53	minibatch 8	time 22h 56m 01s	sec/tick 2402.2	sec/kimg 60.06	maintenance 0.5
tick 78	kimg 6920.1	lod 1.47	minibatch 8	time 23h 36m 04s	sec/tick 2402.2	sec/kimg 60.05	maintenance 0.5
tick 79	kimg 6960.1	lod 1.40	minibatch 8	time 1d 00h 16m	sec/tick 2402.5	sec/kimg 60.06	maintenance 0.6
tick 80	kimg 7000.1	lod 1.33	minibatch 8	time 1d 00h 56m	sec/tick 2402.7	sec/kimg 60.07	maintenance 0.6
tick 81	kimg 7040.1	lod 1.27	minibatch 8	time 1d 01h 36m	sec/tick 2403.5	sec/kimg 60.09	maintenance 1.1
tick 82	kimg 7080.1	lod 1.20	minibatch 8	time 1d 02h 16m	sec/tick 2402.2	sec/kimg 60.06	maintenance 0.6
tick 83	kimg 7120.1	lod 1.13	minibatch 8	time 1d 02h 56m	sec/tick 2403.6	sec/kimg 60.09	maintenance 0.6
tick 84	kimg 7160.1	lod 1.07	minibatch 8	time 1d 03h 36m	sec/tick 2402.5	sec/kimg 60.06	maintenance 0.6
tick 85	kimg 7200.1	lod 1.00	minibatch 8	time 1d 04h 16m	sec/tick 2404.4	sec/kimg 60.11	maintenance 0.6
tick 86	kimg 7240.1	lod 1.00	minibatch 8	time 1d 04h 54m	sec/tick 2307.0	sec/kimg 57.68	maintenance 0.7
tick 87	kimg 7280.1	lod 1.00	minibatch 8	time 1d 05h 33m	sec/tick 2318.9	sec/kimg 57.97	maintenance 0.6
tick 88	kimg 7320.1	lod 1.00	minibatch 8	time 1d 06h 12m	sec/tick 2308.5	sec/kimg 57.71	maintenance 0.6
tick 89	kimg 7360.1	lod 1.00	minibatch 8	time 1d 06h 50m	sec/tick 2308.1	sec/kimg 57.70	maintenance 0.6
tick 90	kimg 7400.1	lod 1.00	minibatch 8	time 1d 07h 29m	sec/tick 2307.7	sec/kimg 57.69	maintenance 0.6
tick 91	kimg 7440.1	lod 1.00	minibatch 8	time 1d 08h 07m	sec/tick 2307.2	sec/kimg 57.68	maintenance 1.2
tick 92	kimg 7480.1	lod 1.00	minibatch 8	time 1d 08h 46m	sec/tick 2309.7	sec/kimg 57.74	maintenance 0.6
tick 93	kimg 7520.1	lod 1.00	minibatch 8	time 1d 09h 24m	sec/tick 2315.3	sec/kimg 57.88	maintenance 0.6
tick 94	kimg 7560.1	lod 1.00	minibatch 8	time 1d 10h 03m	sec/tick 2327.0	sec/kimg 58.18	maintenance 0.6
tick 95	kimg 7600.1	lod 1.00	minibatch 8	time 1d 10h 42m	sec/tick 2350.9	sec/kimg 58.77	maintenance 0.7
tick 96	kimg 7640.1	lod 1.00	minibatch 8	time 1d 11h 21m	sec/tick 2336.1	sec/kimg 58.40	maintenance 0.7
tick 97	kimg 7680.1	lod 1.00	minibatch 8	time 1d 12h 00m	sec/tick 2311.0	sec/kimg 57.78	maintenance 0.6
tick 98	kimg 7720.1	lod 1.00	minibatch 8	time 1d 12h 38m	sec/tick 2307.9	sec/kimg 57.70	maintenance 0.6
tick 99	kimg 7760.1	lod 1.00	minibatch 8	time 1d 13h 17m	sec/tick 2308.6	sec/kimg 57.71	maintenance 0.6
tick 100	kimg 7800.0	lod 1.00	minibatch 4	time 1d 13h 55m	sec/tick 2314.0	sec/kimg 57.92	maintenance 0.6
tick 101	kimg 7820.0	lod 0.97	minibatch 4	time 1d 14h 36m	sec/tick 2434.9	sec/kimg 121.74	maintenance 1.1
tick 102	kimg 7840.0	lod 0.93	minibatch 4	time 1d 15h 16m	sec/tick 2437.3	sec/kimg 121.87	maintenance 0.8
tick 103	kimg 7860.0	lod 0.90	minibatch 4	time 1d 15h 57m	sec/tick 2438.8	sec/kimg 121.94	maintenance 0.8
tick 104	kimg 7880.0	lod 0.87	minibatch 4	time 1d 16h 38m	sec/tick 2438.7	sec/kimg 121.94	maintenance 0.9
tick 105	kimg 7900.0	lod 0.83	minibatch 4	time 1d 17h 18m	sec/tick 2437.3	sec/kimg 121.86	maintenance 0.8
tick 106	kimg 7920.0	lod 0.80	minibatch 4	time 1d 17h 59m	sec/tick 2438.4	sec/kimg 121.92	maintenance 0.8
tick 107	kimg 7940.0	lod 0.77	minibatch 4	time 1d 18h 40m	sec/tick 2439.1	sec/kimg 121.95	maintenance 0.8
tick 108	kimg 7960.0	lod 0.73	minibatch 4	time 1d 19h 20m	sec/tick 2441.8	sec/kimg 122.09	maintenance 0.8
tick 109	kimg 7980.0	lod 0.70	minibatch 4	time 1d 20h 02m	sec/tick 2471.8	sec/kimg 123.59	maintenance 1.1
tick 110	kimg 8000.0	lod 0.67	minibatch 4	time 1d 20h 43m	sec/tick 2473.7	sec/kimg 123.68	maintenance 1.0
tick 111	kimg 8020.0	lod 0.63	minibatch 4	time 1d 21h 24m	sec/tick 2474.8	sec/kimg 123.74	maintenance 1.4
tick 112	kimg 8040.0	lod 0.60	minibatch 4	time 1d 22h 05m	sec/tick 2473.3	sec/kimg 123.67	maintenance 0.9
tick 113	kimg 8060.0	lod 0.57	minibatch 4	time 1d 22h 47m	sec/tick 2473.0	sec/kimg 123.65	maintenance 0.9
tick 114	kimg 8080.0	lod 0.53	minibatch 4	time 1d 23h 28m	sec/tick 2471.8	sec/kimg 123.59	maintenance 0.9
tick 115	kimg 8100.0	lod 0.50	minibatch 4	time 2d 00h 09m	sec/tick 2471.3	sec/kimg 123.57	maintenance 0.8
tick 116	kimg 8120.0	lod 0.47	minibatch 4	time 2d 00h 50m	sec/tick 2471.6	sec/kimg 123.58	maintenance 0.9
tick 117	kimg 8140.0	lod 0.43	minibatch 4	time 2d 01h 31m	sec/tick 2471.3	sec/kimg 123.56	maintenance 0.9
tick 118	kimg 8160.0	lod 0.40	minibatch 4	time 2d 02h 13m	sec/tick 2471.8	sec/kimg 123.59	maintenance 0.9
tick 119	kimg 8180.0	lod 0.37	minibatch 4	time 2d 02h 53m	sec/tick 2453.1	sec/kimg 122.66	maintenance 0.9
tick 120	kimg 8200.0	lod 0.33	minibatch 4	time 2d 03h 34m	sec/tick 2443.5	sec/kimg 122.17	maintenance 0.8
tick 121	kimg 8220.0	lod 0.30	minibatch 4	time 2d 04h 15m	sec/tick 2443.4	sec/kimg 122.17	maintenance 1.4
tick 122	kimg 8240.0	lod 0.27	minibatch 4	time 2d 04h 56m	sec/tick 2440.8	sec/kimg 122.04	maintenance 0.8
tick 123	kimg 8260.0	lod 0.23	minibatch 4	time 2d 05h 36m	sec/tick 2439.5	sec/kimg 121.98	maintenance 0.9
tick 124	kimg 8280.0	lod 0.20	minibatch 4	time 2d 06h 17m	sec/tick 2439.8	sec/kimg 121.99	maintenance 0.9
tick 125	kimg 8300.0	lod 0.17	minibatch 4	time 2d 06h 58m	sec/tick 2441.9	sec/kimg 122.10	maintenance 0.9
tick 126	kimg 8320.0	lod 0.13	minibatch 4	time 2d 07h 38m	sec/tick 2442.5	sec/kimg 122.12	maintenance 0.9
tick 127	kimg 8340.0	lod 0.10	minibatch 4	time 2d 08h 19m	sec/tick 2421.0	sec/kimg 121.05	maintenance 0.8
tick 128	kimg 8360.0	lod 0.07	minibatch 4	time 2d 08h 59m	sec/tick 2419.6	sec/kimg 120.98	maintenance 0.6
tick 129	kimg 8380.0	lod 0.03	minibatch 4	time 2d 09h 39m	sec/tick 2411.8	sec/kimg 120.59	maintenance 0.6
tick 130	kimg 8400.0	lod 0.00	minibatch 4	time 2d 10h 20m	sec/tick 2408.9	sec/kimg 120.44	maintenance 0.6
tick 131	kimg 8420.0	lod 0.00	minibatch 4	time 2d 10h 58m	sec/tick 2279.0	sec/kimg 113.95	maintenance 1.1

```
tick 132   kimg 8440.0   lod 0.00  minibatch 4    time 2d 11h 36m   sec/tick 2278.8  sec/kimg 113.94  maintenance 0.6
tick 133   kimg 8460.0   lod 0.00  minibatch 4    time 2d 12h 14m   sec/tick 2278.4  sec/kimg 113.92  maintenance 0.6
tick 134   kimg 8480.0   lod 0.00  minibatch 4    time 2d 12h 51m   sec/tick 2277.8  sec/kimg 113.89  maintenance 0.6
tick 135   kimg 8500.0   lod 0.00  minibatch 4    time 2d 13h 29m   sec/tick 2279.1  sec/kimg 113.95  maintenance 0.6
tick 136   kimg 8520.0   lod 0.00  minibatch 4    time 2d 14h 07m   sec/tick 2278.0  sec/kimg 113.90  maintenance 0.6
tick 137   kimg 8540.0   lod 0.00  minibatch 4    time 2d 14h 45m   sec/tick 2276.4  sec/kimg 113.82  maintenance 0.6
tick 138   kimg 8560.0   lod 0.00  minibatch 4    time 2d 15h 23m   sec/tick 2278.8  sec/kimg 113.94  maintenance 0.6
tick 139   kimg 8580.0   lod 0.00  minibatch 4    time 2d 16h 01m   sec/tick 2279.0  sec/kimg 113.95  maintenance 0.6
tick 140   kimg 8600.0   lod 0.00  minibatch 4    time 2d 16h 39m   sec/tick 2277.5  sec/kimg 113.87  maintenance 0.5
tick 141   kimg 8620.0   lod 0.00  minibatch 4    time 2d 17h 17m   sec/tick 2280.9  sec/kimg 114.04  maintenance 1.1
tick 142   kimg 8640.0   lod 0.00  minibatch 4    time 2d 17h 56m   sec/tick 2300.4  sec/kimg 115.02  maintenance 0.6
tick 143   kimg 8660.0   lod 0.00  minibatch 4    time 2d 18h 34m   sec/tick 2314.6  sec/kimg 115.73  maintenance 0.7
tick 144   kimg 8680.0   lod 0.00  minibatch 4    time 2d 19h 13m   sec/tick 2303.6  sec/kimg 115.18  maintenance 0.7
tick 145   kimg 8700.0   lod 0.00  minibatch 4    time 2d 19h 51m   sec/tick 2298.6  sec/kimg 114.93  maintenance 0.6
Traceback (most recent call last):
  File .train.py", line 285, in <module>
    tfutil.call_func_by_name(**config.train)
  File ./home/paperspace/prGAN_04_190118/tfutil.py", line 236, in call_func_by_name
    return import_obj(func)(*args, **kwargs)
  File ./home/paperspace/prGAN_04_190118/train.py", line 229, in train_progressive_gan
    tfutil.run([D_train_op, Gs_update_op], {lod_in: sched.lod, lrate_in: sched.D_lrate, minibatch_in: sched.minibatch})
  File ./home/paperspace/prGAN_04_190118/tfutil.py", line 21, in run
    return tf.get_default_session().run(*args, **kwargs)
  File ./home/paperspace/anaconda3/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 900, in run
    run_metadata_ptr)
  File ./home/paperspace/anaconda3/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1135, in _run
    feed_dict_tensor, options, run_metadata)
  File ./home/paperspace/anaconda3/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1316, in _do_run
    run_metadata)
  File ./home/paperspace/anaconda3/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1322, in _do_call
    return fn(*args)
  File ./home/paperspace/anaconda3/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1307, in _run_fn
    options, feed_dict, fetch_list, target_list, run_metadata)
  File ./home/paperspace/anaconda3/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1409, in _call_tf_session-
run
    run_metadata)
KeyboardInterrupt
```

Portrait

```
Initializing TensorFlow...
Running train.train_progressive_gan()...
Streaming data using dataset.TFRecordDataset...
Dataset shape = [3, 512, 512]
Dynamic range = [0, 255]
Label size    = 0
Constructing networks...
```

G	Params	OutputShape	WeightShape
latents_in	-	(?, 512)	-
labels_in	-	(?, 0)	-
lod	-	()	-
4x4/PixelNorm	-	(?, 512)	-
4x4/Dense	4194816	(?, 512, 4, 4)	(512, 8192)
4x4/Conv	2359808	(?, 512, 4, 4)	(3, 3, 512, 512)
ToRGB_lod7	1539	(?, 3, 4, 4)	(1, 1, 512, 3)
8x8/Conv0_up	2359808	(?, 512, 8, 8)	(3, 3, 512, 512)
8x8/Conv1	2359808	(?, 512, 8, 8)	(3, 3, 512, 512)
ToRGB_lod6	1539	(?, 3, 8, 8)	(1, 1, 512, 3)
Upscale2D	-	(?, 3, 8, 8)	-
Grow_lod6	-	(?, 3, 8, 8)	-
16x16/Conv0_up	2359808	(?, 512, 16, 16)	(3, 3, 512, 512)
16x16/Conv1	2359808	(?, 512, 16, 16)	(3, 3, 512, 512)
ToRGB_lod5	1539	(?, 3, 16, 16)	(1, 1, 512, 3)
Upscale2D_1	-	(?, 3, 16, 16)	-
Grow_lod5	-	(?, 3, 16, 16)	-
32x32/Conv0_up	2359808	(?, 512, 32, 32)	(3, 3, 512, 512)
32x32/Conv1	2359808	(?, 512, 32, 32)	(3, 3, 512, 512)
ToRGB_lod4	1539	(?, 3, 32, 32)	(1, 1, 512, 3)
Upscale2D_2	-	(?, 3, 32, 32)	-
Grow_lod4	-	(?, 3, 32, 32)	-
64x64/Conv0_up	1179904	(?, 256, 64, 64)	(3, 3, 256, 512)
64x64/Conv1	590080	(?, 256, 64, 64)	(3, 3, 256, 256)
ToRGB_lod3	771	(?, 3, 64, 64)	(1, 1, 256, 3)
Upscale2D_3	-	(?, 3, 64, 64)	-
Grow_lod3	-	(?, 3, 64, 64)	-
128x128/Conv0_up	295040	(?, 128, 128, 128)	(3, 3, 128, 256)
128x128/Conv1	147584	(?, 128, 128, 128)	(3, 3, 128, 128)
ToRGB_lod2	387	(?, 3, 128, 128)	(1, 1, 128, 3)
Upscale2D_4	-	(?, 3, 128, 128)	-
Grow_lod2	-	(?, 3, 128, 128)	-
256x256/Conv0_up	73792	(?, 64, 256, 256)	(3, 3, 64, 128)
256x256/Conv1	36928	(?, 64, 256, 256)	(3, 3, 64, 64)
ToRGB_lod1	195	(?, 3, 256, 256)	(1, 1, 64, 3)
Upscale2D_5	-	(?, 3, 256, 256)	-
Grow_lod1	-	(?, 3, 256, 256)	-
512x512/Conv0_up	18464	(?, 32, 512, 512)	(3, 3, 32, 64)
512x512/Conv1	9248	(?, 32, 512, 512)	(3, 3, 32, 32)
ToRGB_lod0	99	(?, 3, 512, 512)	(1, 1, 32, 3)
Upscale2D_6	-	(?, 3, 512, 512)	-
Grow_lod0	-	(?, 3, 512, 512)	-
images_out	-	(?, 3, 512, 512)	-
---	---	---	---
Total	23072120		

D	Params	OutputShape	WeightShape
images_in	-	(?, 3, 512, 512)	-
lod	-	()	-
FromRGB_lod0	128	(?, 32, 512, 512)	(1, 1, 3, 32)
512x512/Conv0	9248	(?, 32, 512, 512)	(3, 3, 32, 32)
512x512/Conv1_down	18496	(?, 64, 256, 256)	(3, 3, 32, 64)
Downscale2D	-	(?, 3, 256, 256)	-
FromRGB_lod1	256	(?, 64, 256, 256)	(1, 1, 3, 64)
Grow_lod0	-	(?, 64, 256, 256)	-
256x256/Conv0	36928	(?, 64, 256, 256)	(3, 3, 64, 64)
256x256/Conv1_down	73856	(?, 128, 128, 128)	(3, 3, 64, 128)
Downscale2D_1	-	(?, 3, 128, 128)	-
FromRGB_lod2	512	(?, 128, 128, 128)	(1, 1, 3, 128)
Grow_lod1	-	(?, 128, 128, 128)	-
128x128/Conv0	147584	(?, 128, 128, 128)	(3, 3, 128, 128)
128x128/Conv1_down	295168	(?, 256, 64, 64)	(3, 3, 128, 256)
Downscale2D_2	-	(?, 3, 64, 64)	-
FromRGB_lod3	1024	(?, 256, 64, 64)	(1, 1, 3, 256)
Grow_lod2	-	(?, 256, 64, 64)	-
64x64/Conv0	590080	(?, 256, 64, 64)	(3, 3, 256, 256)
64x64/Conv1_down	1180160	(?, 512, 32, 32)	(3, 3, 256, 512)
Downscale2D_3	-	(?, 3, 32, 32)	-
FromRGB_lod4	2048	(?, 512, 32, 32)	(1, 1, 3, 512)
Grow_lod3	-	(?, 512, 32, 32)	-
32x32/Conv0	2359808	(?, 512, 32, 32)	(3, 3, 512, 512)
32x32/Conv1_down	2359808	(?, 512, 16, 16)	(3, 3, 512, 512)
Downscale2D_4	-	(?, 3, 16, 16)	-
FromRGB_lod5	2048	(?, 512, 16, 16)	(1, 1, 3, 512)
Grow_lod4	-	(?, 512, 16, 16)	-
16x16/Conv0	2359808	(?, 512, 16, 16)	(3, 3, 512, 512)
16x16/Conv1_down	2359808	(?, 512, 8, 8)	(3, 3, 512, 512)
Downscale2D_5	-	(?, 3, 8, 8)	-
FromRGB_lod6	2048	(?, 512, 8, 8)	(1, 1, 3, 512)
Grow_lod5	-	(?, 512, 8, 8)	-
8x8/Conv0	2359808	(?, 512, 8, 8)	(3, 3, 512, 512)
8x8/Conv1_down	2359808	(?, 512, 4, 4)	(3, 3, 512, 512)
Downscale2D_6	-	(?, 3, 4, 4)	-
FromRGB_lod7	2048	(?, 512, 4, 4)	(1, 1, 3, 512)
Grow_lod6	-	(?, 512, 4, 4)	-
4x4/MinibatchStddev	-	(?, 1, 4, 4)	-
4x4/Conv	2364416	(?, 512, 4, 4)	(3, 3, 513, 512)
4x4/Dense0	4194816	(?, 512)	(8192, 512)
4x4/Dense1	513	(?, 1)	(512, 1)
scores_out	-	(?, 1)	-
labels_out	-	(?, 0)	-
---	---	---	---
Total	23080225		

```
Building TensorFlow graph...
Setting up snapshot image grid...
Setting up result dir...
Saving results to results/004-pgan-from_images-preset-v2-1gpu-fp16
Training...
tick 1     kimg 160.3    lod 7.00  minibatch 128  time 1m 52s      sec/tick 112.2   sec/kimg 0.70    maintenance 44.5
tick 2     kimg 320.5    lod 7.00  minibatch 128  time 3m 49s      sec/tick 105.0   sec/kimg 0.66    maintenance 12.0
tick 3     kimg 480.8    lod 7.00  minibatch 128  time 5m 35s      sec/tick 104.9   sec/kimg 0.65    maintenance 0.4
```

```
tick 4     kimg 621.1    lod 6.97  minibatch 128  time 7m 20s      sec/tick 105.3   sec/kimg 0.75   maintenance 0.4
tick 5     kimg 761.3    lod 6.73  minibatch 128  time 9m 58s      sec/tick 157.5   sec/kimg 1.12   maintenance 0.4
tick 6     kimg 901.6    lod 6.50  minibatch 128  time 12m 37s     sec/tick 158.4   sec/kimg 1.13   maintenance 0.4
tick 7     kimg 1041.9   lod 6.26  minibatch 128  time 15m 15s     sec/tick 157.5   sec/kimg 1.12   maintenance 0.4
tick 8     kimg 1182.2   lod 6.03  minibatch 128  time 17m 53s     sec/tick 157.6   sec/kimg 1.12   maintenance 0.4
tick 9     kimg 1322.5   lod 6.00  minibatch 128  time 20m 23s     sec/tick 149.3   sec/kimg 1.06   maintenance 0.6
tick 10    kimg 1462.8   lod 6.00  minibatch 128  time 22m 52s     sec/tick 148.7   sec/kimg 1.06   maintenance 0.4
tick 11    kimg 1603.1   lod 6.00  minibatch 128  time 25m 21s     sec/tick 146.9   sec/kimg 1.05   maintenance 2.7
tick 12    kimg 1743.4   lod 6.00  minibatch 128  time 27m 45s     sec/tick 142.9   sec/kimg 1.02   maintenance 0.3
tick 13    kimg 1863.7   lod 5.89  minibatch 128  time 31m 12s     sec/tick 207.2   sec/kimg 1.72   maintenance 0.4
tick 14    kimg 1984.0   lod 5.69  minibatch 128  time 35m 38s     sec/tick 265.2   sec/kimg 2.20   maintenance 0.4
tick 15    kimg 2104.3   lod 5.49  minibatch 128  time 40m 03s     sec/tick 264.6   sec/kimg 2.20   maintenance 0.4
tick 16    kimg 2224.6   lod 5.29  minibatch 128  time 44m 28s     sec/tick 264.9   sec/kimg 2.20   maintenance 0.4
tick 17    kimg 2345.0   lod 5.09  minibatch 128  time 48m 53s     sec/tick 264.5   sec/kimg 2.20   maintenance 0.3
tick 18    kimg 2465.3   lod 5.00  minibatch 128  time 53m 13s     sec/tick 260.1   sec/kimg 2.16   maintenance 0.4
tick 19    kimg 2585.6   lod 5.00  minibatch 128  time 57m 30s     sec/tick 256.7   sec/kimg 2.13   maintenance 0.4
tick 20    kimg 2705.9   lod 5.00  minibatch 128  time 1h 01m 49s  sec/tick 258.1   sec/kimg 2.14   maintenance 0.4
tick 21    kimg 2826.2   lod 5.00  minibatch 128  time 1h 06m 08s  sec/tick 258.7   sec/kimg 2.15   maintenance 0.9
tick 22    kimg 2946.6   lod 5.00  minibatch 128  time 1h 10m 26s  sec/tick 256.8   sec/kimg 2.13   maintenance 0.4
tick 23    kimg 3046.7   lod 4.92  minibatch 64   time 1h 17m 39s  sec/tick 433.0   sec/kimg 4.33   maintenance 0.4
tick 24    kimg 3146.8   lod 4.76  minibatch 64   time 1h 29m 06s  sec/tick 686.7   sec/kimg 6.86   maintenance 0.4
tick 25    kimg 3246.8   lod 4.59  minibatch 64   time 1h 40m 34s  sec/tick 687.5   sec/kimg 6.87   maintenance 0.3
tick 26    kimg 3346.9   lod 4.42  minibatch 64   time 1h 52m 02s  sec/tick 688.1   sec/kimg 6.87   maintenance 0.3
tick 27    kimg 3447.0   lod 4.26  minibatch 64   time 2h 03m 32s  sec/tick 689.3   sec/kimg 6.89   maintenance 0.3
tick 28    kimg 3547.1   lod 4.09  minibatch 64   time 2h 15m 02s  sec/tick 689.5   sec/kimg 6.89   maintenance 0.3
tick 29    kimg 3647.2   lod 4.00  minibatch 64   time 2h 26m 24s  sec/tick 681.8   sec/kimg 6.81   maintenance 0.3
tick 30    kimg 3747.3   lod 4.00  minibatch 64   time 2h 37m 36s  sec/tick 671.4   sec/kimg 6.71   maintenance 0.3
tick 31    kimg 3847.4   lod 4.00  minibatch 64   time 2h 48m 49s  sec/tick 671.9   sec/kimg 6.71   maintenance 0.9
tick 32    kimg 3947.5   lod 4.00  minibatch 64   time 3h 00m 01s  sec/tick 671.7   sec/kimg 6.71   maintenance 0.4
tick 33    kimg 4047.6   lod 4.00  minibatch 64   time 3h 11m 12s  sec/tick 671.3   sec/kimg 6.71   maintenance 0.4
tick 34    kimg 4147.7   lod 4.00  minibatch 64   time 3h 22m 24s  sec/tick 671.4   sec/kimg 6.71   maintenance 0.3
tick 35    kimg 4227.7   lod 3.95  minibatch 32   time 3h 35m 09s  sec/tick 764.4   sec/kimg 9.56   maintenance 0.3
tick 36    kimg 4307.7   lod 3.82  minibatch 32   time 3h 55m 10s  sec/tick 1200.5  sec/kimg 15.01  maintenance 0.4
tick 37    kimg 4387.7   lod 3.69  minibatch 32   time 4h 15m 12s  sec/tick 1201.4  sec/kimg 15.02  maintenance 0.4
tick 38    kimg 4467.7   lod 3.55  minibatch 32   time 4h 35m 09s  sec/tick 1196.7  sec/kimg 14.96  maintenance 0.4
tick 39    kimg 4547.7   lod 3.42  minibatch 32   time 4h 55m 05s  sec/tick 1195.7  sec/kimg 14.95  maintenance 0.3
tick 40    kimg 4627.7   lod 3.29  minibatch 32   time 5h 15m 00s  sec/tick 1195.2  sec/kimg 14.94  maintenance 0.4
tick 41    kimg 4707.7   lod 3.15  minibatch 32   time 5h 34m 56s  sec/tick 1195.0  sec/kimg 14.94  maintenance 0.8
tick 42    kimg 4787.7   lod 3.02  minibatch 32   time 5h 54m 52s  sec/tick 1195.6  sec/kimg 14.94  maintenance 0.4
tick 43    kimg 4867.7   lod 3.00  minibatch 32   time 6h 14m 17s  sec/tick 1164.7  sec/kimg 14.56  maintenance 0.4
tick 44    kimg 4947.7   lod 3.00  minibatch 32   time 6h 33m 38s  sec/tick 1160.2  sec/kimg 14.50  maintenance 0.4
tick 45    kimg 5027.7   lod 3.00  minibatch 32   time 6h 52m 59s  sec/tick 1160.9  sec/kimg 14.51  maintenance 0.3
tick 46    kimg 5107.7   lod 3.00  minibatch 32   time 7h 12m 20s  sec/tick 1160.4  sec/kimg 14.51  maintenance 0.3
tick 47    kimg 5187.7   lod 3.00  minibatch 32   time 7h 31m 41s  sec/tick 1160.8  sec/kimg 14.51  maintenance 0.3
tick 48    kimg 5267.7   lod 3.00  minibatch 32   time 7h 51m 09s  sec/tick 1168.3  sec/kimg 14.60  maintenance 0.4
tick 49    kimg 5347.7   lod 3.00  minibatch 32   time 8h 10m 41s  sec/tick 1171.3  sec/kimg 14.64  maintenance 0.3
tick 50    kimg 5407.7   lod 2.99  minibatch 16   time 8h 27m 19s  sec/tick 997.2   sec/kimg 16.61  maintenance 0.4
tick 51    kimg 5467.8   lod 2.89  minibatch 16   time 8h 57m 53s  sec/tick 1833.4  sec/kimg 30.54  maintenance 1.0
tick 52    kimg 5527.8   lod 2.79  minibatch 16   time 9h 28m 27s  sec/tick 1834.0  sec/kimg 30.55  maintenance 0.4
tick 53    kimg 5587.8   lod 2.69  minibatch 16   time 9h 59m 01s  sec/tick 1833.2  sec/kimg 30.54  maintenance 0.4
tick 54    kimg 5647.9   lod 2.59  minibatch 16   time 10h 29m 35s sec/tick 1833.6  sec/kimg 30.54  maintenance 0.4
tick 55    kimg 5707.9   lod 2.49  minibatch 16   time 11h 00m 06s sec/tick 1831.1  sec/kimg 30.50  maintenance 0.3
tick 56    kimg 5767.9   lod 2.39  minibatch 16   time 11h 30m 25s sec/tick 1818.0  sec/kimg 30.28  maintenance 0.4
tick 57    kimg 5828.0   lod 2.29  minibatch 16   time 12h 00m 43s sec/tick 1817.7  sec/kimg 30.28  maintenance 0.4
tick 58    kimg 5888.0   lod 2.19  minibatch 16   time 12h 30m 58s sec/tick 1814.9  sec/kimg 30.23  maintenance 0.4
tick 59    kimg 5948.0   lod 2.09  minibatch 16   time 13h 01m 13s sec/tick 1814.0  sec/kimg 30.22  maintenance 0.4
tick 60    kimg 6008.1   lod 2.00  minibatch 16   time 13h 31m 18s sec/tick 1805.0  sec/kimg 30.07  maintenance 0.4
tick 61    kimg 6068.1   lod 2.00  minibatch 16   time 14h 00m 37s sec/tick 1758.4  sec/kimg 29.29  maintenance 1.0
tick 62    kimg 6128.1   lod 2.00  minibatch 16   time 14h 29m 54s sec/tick 1756.6  sec/kimg 29.26  maintenance 0.4
tick 63    kimg 6188.2   lod 2.00  minibatch 16   time 14h 59m 15s sec/tick 1759.8  sec/kimg 29.31  maintenance 0.4
tick 64    kimg 6248.2   lod 2.00  minibatch 16   time 15h 28m 31s sec/tick 1756.4  sec/kimg 29.26  maintenance 0.4
tick 65    kimg 6308.2   lod 2.00  minibatch 16   time 15h 57m 46s sec/tick 1754.4  sec/kimg 29.22  maintenance 0.4
tick 66    kimg 6368.3   lod 2.00  minibatch 16   time 16h 27m 04s sec/tick 1757.1  sec/kimg 29.27  maintenance 0.4
tick 67    kimg 6428.3   lod 2.00  minibatch 16   time 16h 56m 19s sec/tick 1755.1  sec/kimg 29.24  maintenance 0.4
tick 68    kimg 6488.3   lod 2.00  minibatch 16   time 17h 25m 33s sec/tick 1753.7  sec/kimg 29.21  maintenance 0.4
tick 69    kimg 6548.4   lod 2.00  minibatch 16   time 17h 54m 46s sec/tick 1752.6  sec/kimg 29.19  maintenance 0.4
tick 70    kimg 6600.1   lod 2.00  minibatch 8    time 18h 20m 04s sec/tick 1517.7  sec/kimg 29.33  maintenance 0.4
tick 71    kimg 6640.1   lod 1.93  minibatch 8    time 18h 59m 51s sec/tick 2385.6  sec/kimg 59.64  maintenance 1.0
tick 72    kimg 6680.1   lod 1.87  minibatch 8    time 19h 39m 42s sec/tick 2390.4  sec/kimg 59.76  maintenance 0.6
```



```
tick 73    kimg 6720.1   lod 1.80  minibatch 8    time 20h 19m 34s sec/tick 2391.3  sec/kimg 59.78  maintenance 0.5
tick 74    kimg 6760.1   lod 1.73  minibatch 8    time 20h 59m 25s sec/tick 2390.4  sec/kimg 59.76  maintenance 0.5
tick 75    kimg 6800.1   lod 1.67  minibatch 8    time 21h 39m 16s sec/tick 2391.4  sec/kimg 59.78  maintenance 0.5
tick 76    kimg 6840.1   lod 1.60  minibatch 8    time 22h 19m 09s sec/tick 2392.6  sec/kimg 59.82  maintenance 0.4
tick 77    kimg 6880.1   lod 1.53  minibatch 8    time 22h 59m 05s sec/tick 2395.1  sec/kimg 59.88  maintenance 0.6
tick 78    kimg 6920.1   lod 1.47  minibatch 8    time 23h 39m 19s sec/tick 2413.7  sec/kimg 60.34  maintenance 0.5
tick 79    kimg 6960.1   lod 1.40  minibatch 8    time 1d 00h 19m  sec/tick 2395.3  sec/kimg 59.88  maintenance 0.6
tick 80    kimg 7000.1   lod 1.33  minibatch 8    time 1d 00h 59m  sec/tick 2393.4  sec/kimg 59.83  maintenance 0.6
tick 81    kimg 7040.1   lod 1.27  minibatch 8    time 1d 01h 39m  sec/tick 2392.4  sec/kimg 59.81  maintenance 1.1
tick 82    kimg 7080.1   lod 1.20  minibatch 8    time 1d 02h 18m  sec/tick 2393.1  sec/kimg 59.83  maintenance 0.6
tick 83    kimg 7120.1   lod 1.13  minibatch 8    time 1d 02h 58m  sec/tick 2391.8  sec/kimg 59.79  maintenance 0.6
tick 84    kimg 7160.1   lod 1.07  minibatch 8    time 1d 03h 38m  sec/tick 2391.5  sec/kimg 59.79  maintenance 0.6
tick 85    kimg 7200.1   lod 1.00  minibatch 8    time 1d 04h 18m  sec/tick 2390.8  sec/kimg 59.77  maintenance 0.6
tick 86    kimg 7240.1   lod 1.00  minibatch 8    time 1d 04h 56m  sec/tick 2295.7  sec/kimg 57.39  maintenance 0.7
tick 87    kimg 7280.1   lod 1.00  minibatch 8    time 1d 05h 35m  sec/tick 2295.4  sec/kimg 57.38  maintenance 0.7
tick 88    kimg 7320.1   lod 1.00  minibatch 8    time 1d 06h 13m  sec/tick 2296.5  sec/kimg 57.41  maintenance 0.6
tick 89    kimg 7360.1   lod 1.00  minibatch 8    time 1d 06h 51m  sec/tick 2296.3  sec/kimg 57.41  maintenance 0.6
tick 90    kimg 7400.1   lod 1.00  minibatch 8    time 1d 07h 29m  sec/tick 2298.0  sec/kimg 57.45  maintenance 0.6
tick 91    kimg 7440.1   lod 1.00  minibatch 8    time 1d 08h 08m  sec/tick 2299.2  sec/kimg 57.48  maintenance 1.2
tick 92    kimg 7480.1   lod 1.00  minibatch 8    time 1d 08h 46m  sec/tick 2296.9  sec/kimg 57.42  maintenance 0.6
tick 93    kimg 7520.1   lod 1.00  minibatch 8    time 1d 09h 24m  sec/tick 2297.6  sec/kimg 57.44  maintenance 0.6
tick 94    kimg 7560.1   lod 1.00  minibatch 8    time 1d 10h 03m  sec/tick 2298.0  sec/kimg 57.45  maintenance 0.7
tick 95    kimg 7600.1   lod 1.00  minibatch 8    time 1d 10h 41m  sec/tick 2296.4  sec/kimg 57.41  maintenance 0.6
tick 96    kimg 7640.1   lod 1.00  minibatch 8    time 1d 11h 19m  sec/tick 2297.1  sec/kimg 57.43  maintenance 0.6
tick 97    kimg 7680.1   lod 1.00  minibatch 8    time 1d 11h 58m  sec/tick 2297.0  sec/kimg 57.43  maintenance 0.7
tick 98    kimg 7720.1   lod 1.00  minibatch 8    time 1d 12h 36m  sec/tick 2296.6  sec/kimg 57.42  maintenance 0.6
tick 99    kimg 7760.1   lod 1.00  minibatch 8    time 1d 13h 14m  sec/tick 2296.4  sec/kimg 57.41  maintenance 0.6
tick 100   kimg 7800.0   lod 1.00  minibatch 4    time 1d 13h 53m  sec/tick 2300.1  sec/kimg 57.57  maintenance 0.6
tick 101   kimg 7820.0   lod 0.97  minibatch 4    time 1d 14h 33m  sec/tick 2428.1  sec/kimg 121.40 maintenance 1.3
tick 102   kimg 7840.0   lod 0.93  minibatch 4    time 1d 15h 14m  sec/tick 2431.8  sec/kimg 121.59 maintenance 0.8
tick 103   kimg 7860.0   lod 0.90  minibatch 4    time 1d 15h 54m  sec/tick 2428.1  sec/kimg 121.40 maintenance 0.9
tick 104   kimg 7880.0   lod 0.87  minibatch 4    time 1d 16h 35m  sec/tick 2428.5  sec/kimg 121.42 maintenance 0.9
tick 105   kimg 7900.0   lod 0.83  minibatch 4    time 1d 17h 15m  sec/tick 2428.6  sec/kimg 121.43 maintenance 0.8
tick 106   kimg 7920.0   lod 0.80  minibatch 4    time 1d 17h 56m  sec/tick 2429.4  sec/kimg 121.47 maintenance 0.9
tick 107   kimg 7940.0   lod 0.77  minibatch 4    time 1d 18h 36m  sec/tick 2429.1  sec/kimg 121.45 maintenance 0.9
tick 108   kimg 7960.0   lod 0.73  minibatch 4    time 1d 19h 16m  sec/tick 2421.2  sec/kimg 121.06 maintenance 1.0
tick 109   kimg 7980.0   lod 0.70  minibatch 4    time 1d 19h 56m  sec/tick 2404.0  sec/kimg 120.20 maintenance 0.5
tick 110   kimg 8000.0   lod 0.67  minibatch 4    time 1d 20h 36m  sec/tick 2402.7  sec/kimg 120.14 maintenance 0.5
tick 111   kimg 8020.0   lod 0.63  minibatch 4    time 1d 21h 16m  sec/tick 2398.5  sec/kimg 119.93 maintenance 1.0
tick 112   kimg 8040.0   lod 0.60  minibatch 4    time 1d 21h 56m  sec/tick 2398.9  sec/kimg 119.95 maintenance 0.5
tick 113   kimg 8060.0   lod 0.57  minibatch 4    time 1d 22h 36m  sec/tick 2398.4  sec/kimg 119.92 maintenance 0.5
Traceback (most recent call last):
  File .train.py", line 285, in <module>
    tfutil.call_func_by_name(**config.train)
  File ./home/paperspace/progressive_growing_of_gans/tfutil.py", line 236, in call_func_by_name
    return import_obj(func)(*args, **kwargs)
  File ./home/paperspace/progressive_growing_of_gans/train.py", line 229, in train_progressive_gan
    tfutil.run([D_train_op, Gs_update_op], {lod_in: sched.lod, lrate_in: sched.D_lrate, minibatch_in: sched.minibatch})
  File ./home/paperspace/progressive_growing_of_gans/tfutil.py", line 21, in run
    return tf.get_default_session().run(*args, **kwargs)
  File ./home/paperspace/anaconda3/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 929, in run
    run_metadata_ptr)
  File ./home/paperspace/anaconda3/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1152, in _run
    feed_dict_tensor, options, run_metadata)
  File ./home/paperspace/anaconda3/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1328, in _do_run
    run_metadata)
  File ./home/paperspace/anaconda3/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1334, in _do_call
    return fn(*args)
  File ./home/paperspace/anaconda3/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1319, in _run_fn
    options, feed_dict, fetch_list, target_list, run_metadata)
  File ./home/paperspace/anaconda3/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1407, in _call_tf_session-
run
    run_metadata)
KeyboardInterrupt
```

Literaturverzeichnis

Karras, T., Aila, T., Laine, S., Lehtinen, J. (NVIDIA, Aalto University). Progressive Growing of GANs for Improved Quality, Stability, and Variation. arXiv:1710.10196 [cs.NE]. Zugriff am 26.11.2018. Verfügbar unter https://arxiv.org/pdf/1710.10196.pdf

Boden, M. (School of Cognitive and Computing Sciences, University of Sussex). Creativity and artificial intelligence. Zugriff am 26.11.2018. Verfügbar unter https://ac.els-cdn.com/S0004370298000551/1-s2.0-S0004370298000551-main.pdf?_tid=aaa7e7c5-2b98-42a7-8ee9-f04f7e77d0c6&acdnat=1543324440_e0a0946dac8ac7c2c0f09e8286ed7ee7

Smith, R. (The Stanford Encyclopedia of Philosophy (Winter 2018 Edition)). Aristotle's Logic. Zugriff am 26.11.2018. Verfügbar unter https://plato.stanford.edu/archives/win2018/entries/aristotle-logic

Valladas, H., Tisnérat-Laborde, N., Cachier, H., Arnold, M., De Quirós, F., Cabrera-Valdés, V., Moure-Romanillo, A. (2001). Radiocarbon AMS Dates for Paleolithic Cave Paintings. Radiocarbon, 43(2B), 977-986. doi:10.1017/S0033822200041643. Zugriff am 26.11.2018. Verfügbar unter https://www.cambridge.org/core/services/aop-cambridge-core/content/view/7FCEAC790DD3F2AC28F3B5629C96CC8A/S0033822200041643a.pdf/radiocarbon_ams_dates_for_paleolithic_cave_paintings.pdf

Hy, I., Day, A. (1981). Advances in Intrinsic Motivation and Aesthetics. New York: Plenum Press.

Elgammal, A., Liu, B., Elhoseiny, M., Mazzone, M. (The Art & AI Laboratory - Rutgers University, Department of Computer Science - Rutgers University, Facebook AI Research, Department of Art History - College of Charleston). CAN: Creative Adversarial Networks Generating "Art" by learning About Styles and Deviating from Style Norms. Zugriff am 26.11.2018. Verfügbar unter https://arxiv.org/pdf/1706.07068.pdf

Barrat, R. (GitHub). art-DCGAN - Modified implementation of DCGAN focused on generative art. Includes pre-trained models for landscapes, nude-portraits, and others. Zugriff am 26.11.2018. Verfügbar unter https://github.com/robbiebarrat/art-DCGAN

Fautrel, P., Caselles-Dupré, H., Verniere, G. (Obvius Art). Obvious. Zugriff am 19.11.2018. Verfügbar unter http://obvious-art.com/index.html

DiPaola, S., Gabora, L. (Springer Science+Business Media). Incorporating characteristics of human creativity into an evolutionary art algorithm. Zugriff am 28.11.2018. Verfügbar unter https://link.springer.com/content/pdf/10.1007%2Fs10710-008-9074-x.pdf

Vincent, J. (The Verge). How three french students used borrowed code to put the first AI portrait in Christie's. Zugriff am 28.11.2018. Verfügbar unter https://www.theverge.com/2018/10/23/18013190/ai-art-portrait-auction-christies-belamy-obvious-robbie-barrat-gans

ABBRobotics (YouTube). ABB Robotics - Robotic Artist at Long Distance Art event. Zugriff am 27.11.2018. Verfügbar unter https://www.youtube.com/watch?v=U8Iy0p7toeg

Nugent, C. (Time). The Painter Behind These Artworks Is an AI Program. Do They Still Count as Art? Zugriff am 28.11.2018. Verfügbar unter http://time.com/5357221/obvious-artificial-intelligence-art/

Chintala, S. (GitHub). dcgan.torch - A torch implementation of http://arxiv.org/abs/1511.06434. Zugriff am 28.11.2018. Verfügbar unter https://github.com/soumith/dcgan.torch

Gommel, M., Haitz, M., Zappe, J. (robotlab). Der Bibelschreiber. Zugriff am 28.11.2018. Verfügbar unter http://www.robotlab.de/bios/bible.htm

Gommel, M., Haitz, M., Zappe, J. (robotlab). Der Bibelschreiber. Zugriff am 28.11.2018. Verfügbar unter http://www.robotlab.de/thebig/picture.htm

—

Besonderer Dank an die Autoren des progressiven GAN's / Special thanks to the authors of the progressive GAN

Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen
arXiv:1710.10196 [cs.NE]