Generative Modelle

Nur eine beeindruckende Spielerei ?

Janis Keuper

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Vorstellung: *www.imla.ai*

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

12 Professuren für Methoden und Anwendungen in ML und KI

Studiengang: Angewandte KI

- Start WS20

Inhalte und Ablauf des Bachelor-Studiengangs Angewandte Künstliche Intelligenz

7. Semester	Bachelorthesis		Anwendung der KI	Wahlpflichtfach 2	Seminar 2	
6. Semester	Ethik und IT-Recht	Computer Vision	KI-Systeme und Architekturen	Anwendung der KI	Wahlpflichtfach 1	Projekt 2
5. Semester	Unternehmenspraxis					
4. Semester	Autonome Systeme	Deep Learning	Data Engineering	Natural Language Processing	Seminar 1	
3. Semester	Programmierung mit Java	Software Engineering	Machine Learning 2	Datenbank-systeme	Methoden-kompetenz	Projekt 1
2. Semester	Betriebliche Organisation	Programmierung 2	Machine Learning 1	Statistik	Mathematik 2	
1. Semester	Programmierung 1	Einführung in die Künstliche Intelligenz	Visual Analytics	Mathematik 1		

EMI Elektrotechnik, Medizintechnik und Informatik

Studiengang wird aktuell eingerichtet. Stand: 27.02.2020

Agenda

Teil I : Einführung
– Spielereien und Beispiele

Teil II : Methoden
– Generative Modelle

Teil III: Anwendungen
- praktischer Nutzen generativer Modelle

Teil IV: Gefahren
- Deepfakes

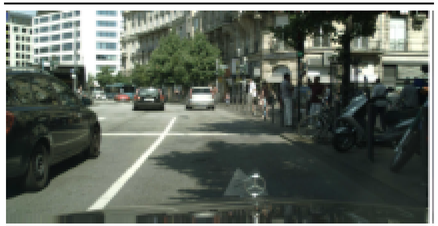
Vom Klassifikator zum Generator

$$X \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \cdots \times \mathbb{R}^{n_k} \longrightarrow \boxed{\text{ML}} \longrightarrow Y \in \mathbb{R}$$

Input **Output**

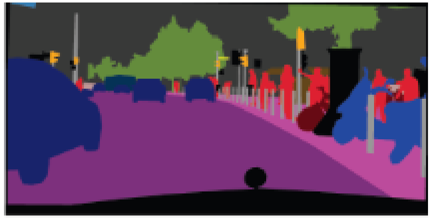
Vom Klassifikator zum Generator

$$X \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \cdots \times \mathbb{R}^{n_k} \longrightarrow \boxed{\text{DL}} \longrightarrow Y \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \cdots \times \mathbb{R}^{m_k}$$



CNN

CITYSCAPES DATASET



semantic segmentation:

Vom Klassifikator zum Generator

$$X \in \mathbb{R}^n \longrightarrow \boxed{\text{DL}} \longrightarrow Y \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \cdots \times \mathbb{R}^{m_k}$$

$\longrightarrow$

Generative Modelle

Beispiel: Bilder



Beispiel: Bilder

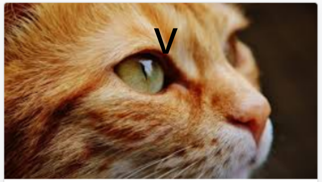
https://thisxdoesnotexist.com/

This Startup Does Not Exist

Using generative adversarial networks (GAN), we can learn how to create realistic-looking fake versions of almost anything, as shown by this collection of sites that have sprung up in the past month. Learn how it works.

This Person Does Not Exist

The site that started it all, with the name that says it all. Created using a style-based generative adversarial network (StyleGAN), this website had the tech community buzzing with excitement and intrigue and inspired many more sites.



This Cat Does Not Exist

These purr-fect GAN-made cats will freshen your feeline-gs and make you wish you could reach through your screen and cuddle them. Once in a while the cats have visual deformities due to imperfections in the model – beware, they can cause nightmares.

This Rental Does Not Exist

Why bother trying to look for the perfect home when you can create one instead? Just find a listing you like, buy some land, build it, and then enjoy the rest of your life.

Beispiel: Beethovens Unvollendete


SPIEGEL Kultur

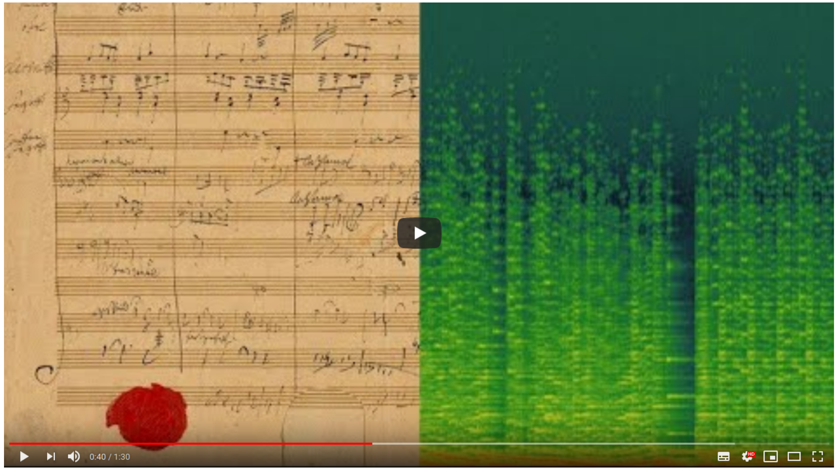
Menü ‹ Kultur › Musik › Ludwig van Beethoven › Ludwig van Beethoven: Künstliche Intelligenz soll Unvollendete vollenden

Zum 250. Geburtstag

Künstliche Intelligenz soll Beethovens Unvollendete vollenden

Wie die Werke vieler großer Komponisten blieb auch das von Ludwig van Beethoven unvollendet. Zum Jubiläumsjahr 2020 soll die 10. Sinfonie laut einem Bericht nun einen Abschluss finden - mit einem speziell trainierten Algorithmus.

07.12.2019, 16:28 Uhr



https://www.youtube.com/watch?v=sCJKFo7zZig

Agenda

Teil I : Einführung
– Spielereien und Beispiele

Teil II : Methoden
– Generative Modelle

Teil III: Anwendungen
- praktischer Nutzen generativer Modelle

Teil IV: Gefahren
- Deepfakes

Generative Adversarial Networks (GANs)

Generative Adversarial Nets

Ian J. Goodfellow*, Jean Pouget-Abadie†, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair‡, Aaron Courville, Yoshua Bengio§
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Abstract

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

1 Introduction

The promise of deep learning is to discover rich, hierarchical models [2] that represent probability distributions over the kinds of data encountered in artificial intelligence applications, such as natural images, audio waveforms containing speech, and symbols in natural language corpora. So far, the most striking successes in deep learning have involved discriminative models, usually those that map a high-dimensional, rich sensory input to a class label [14, 20]. These striking successes have primarily been based on the backpropagation and dropout algorithms, using piecewise linear units [17, 8, 9] which have a particularly well-behaved gradient . Deep *generative* models have had less of an impact, due to the difficulty of approximating many intractable probabilistic computations that arise in maximum likelihood estimation and related strategies, and due to difficulty of leveraging the benefits of piecewise linear units in the generative context. We propose a new generative model estimation procedure that sidesteps these difficulties. [1]

In the proposed *adversarial nets* framework, the generative model is pitted against an adversary: a discriminative model that learns to determine whether a sample is from the model distribution or the data distribution. The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistiguishable from the genuine articles.

[1] Original Paper

NIPS 2016 Tutorial: Generative Adversarial Networks

Ian Goodfellow

OpenAI, ian@openai.com

Abstract

This report summarizes the tutorial presented by the author at NIPS 2016 on *generative adversarial networks* (GANs). The tutorial describes: (1) Why generative modeling is a topic worth studying, (2) how generative models work, and how GANs compare to other generative models, (3) the details of how GANs work, (4) research frontiers in GANs, and (5) state-of-the-art image models that combine GANs with other methods. Finally, the tutorial contains three exercises for readers to complete, and the solutions to these exercises.

Introduction

This report[1] summarizes the content of the NIPS 2016 tutorial on *generative adversarial networks* (GANs) (Goodfellow *et al.*, 2014b). The tutorial was designed primarily to ensure that it answered most of the questions asked by audience members ahead of time, in order to make sure that the tutorial would be as useful as possible to the audience. This tutorial is not intended to be a comprehensive review of the field of GANs; many excellent papers are not described here, simply because they were not relevant to answering the most frequent questions, and because the tutorial was delivered as a two hour oral presentation and did not have unlimited time cover all subjects.

The tutorial describes: (1) Why generative modeling is a topic worth studying, (2) how generative models work, and how GANs compare to other generative models, (3) the details of how GANs work, (4) research frontiers in GANs, and (5) state-of-the-art image models that combine GANs with other methods. Finally, the tutorial contains three exercises for readers to complete, and the solutions to these exercises.

The slides for the tutorial are available in PDF and Keynote format at the following URLs:

`http://www.iangoodfellow.com/slides/2016-12-04-NIPS.pdf`

[2] NIPS Tutorial
(in depth)

UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS

Alec Radford & Luke Metz
indico Research
Boston, MA
{alec,luke}@indico.io

Soumith Chintala
Facebook AI Research
New York, NY
soumith@fb.com

ABSTRACT

In recent years, supervised learning with convolutional networks (CNNs) has seen huge adoption in computer vision applications. Comparatively, unsupervised learning with CNNs has received less attention. In this work we hope to help bridge the gap between the success of CNNs for supervised learning and unsupervised learning. We introduce a class of CNNs called deep convolutional generative adversarial networks (DCGANs), that have certain architectural constraints, and demonstrate that they are a strong candidate for unsupervised learning. Training on various image datasets, we show convincing evidence that our deep convolutional adversarial pair learns a hierarchy of representations from object parts to scenes in both the generator and discriminator. Additionally, we use the learned features for novel tasks - demonstrating their applicability as general image representations.

1 INTRODUCTION

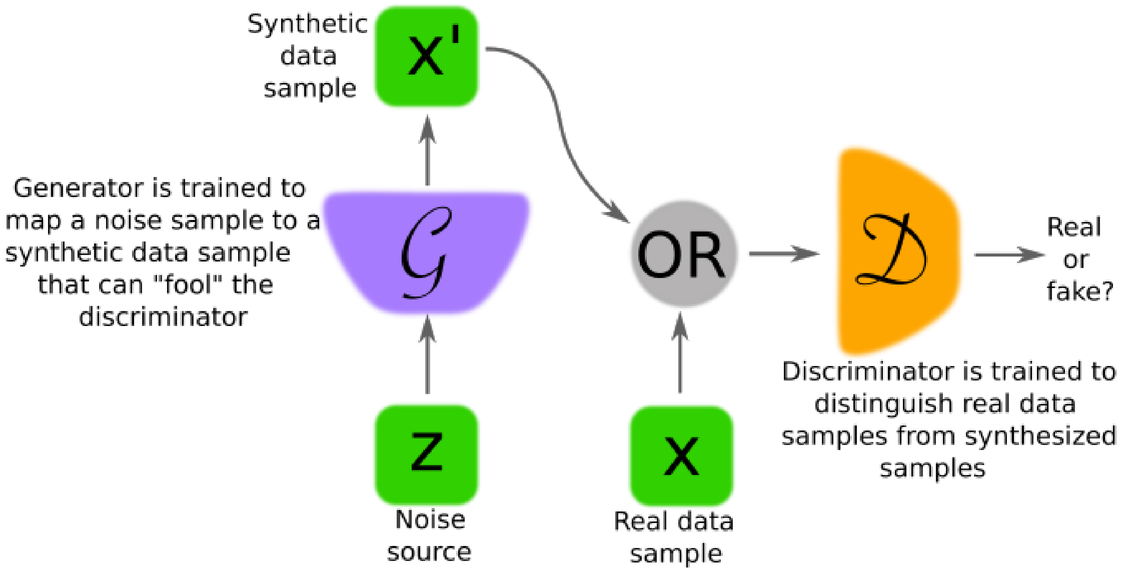
Learning reusable feature representations from large unlabeled datasets has been an area of active research. In the context of computer vision, one can leverage the practically unlimited amount of unlabeled images and videos to learn good intermediate representations, which can then be used on a variety of supervised learning tasks such as image classification. We propose that one way to build good image representations is by training Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), and later reusing parts of the generator and discriminator networks as feature extractors for supervised tasks. GANs provide an attractive alternative to maximum likelihood techniques. One can additionally argue that their learning process and the lack of a heuristic cost function (such as pixel-wise independent mean-square error) are attractive to representation learning. GANs have been known to be unstable to train, often resulting in generators that produce nonsensical outputs. There has been very limited published research in trying to understand and visualize what GANs learn, and the intermediate representations of multi-layer GANs.

[3] Selected popular architecture

Aufbau von GANs

- In a Nutshell -
Generative Adversarial Nets are:

- Groups of DNNs (at least two)
- Working against each other !
- Min parts:
  - Discriminator Network
  - Generator Network

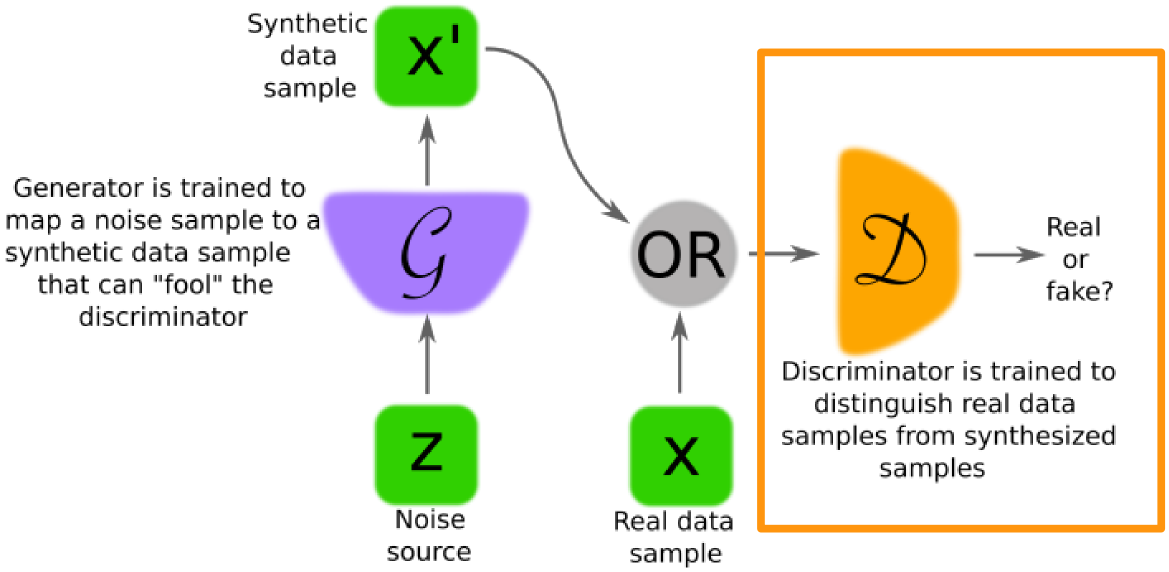


[Grafik: https://arxiv.org/pdf/1710.07035.pdf]

Aufbau von GANs

- In a Nutshell -
Generative Adversarial Nets are:

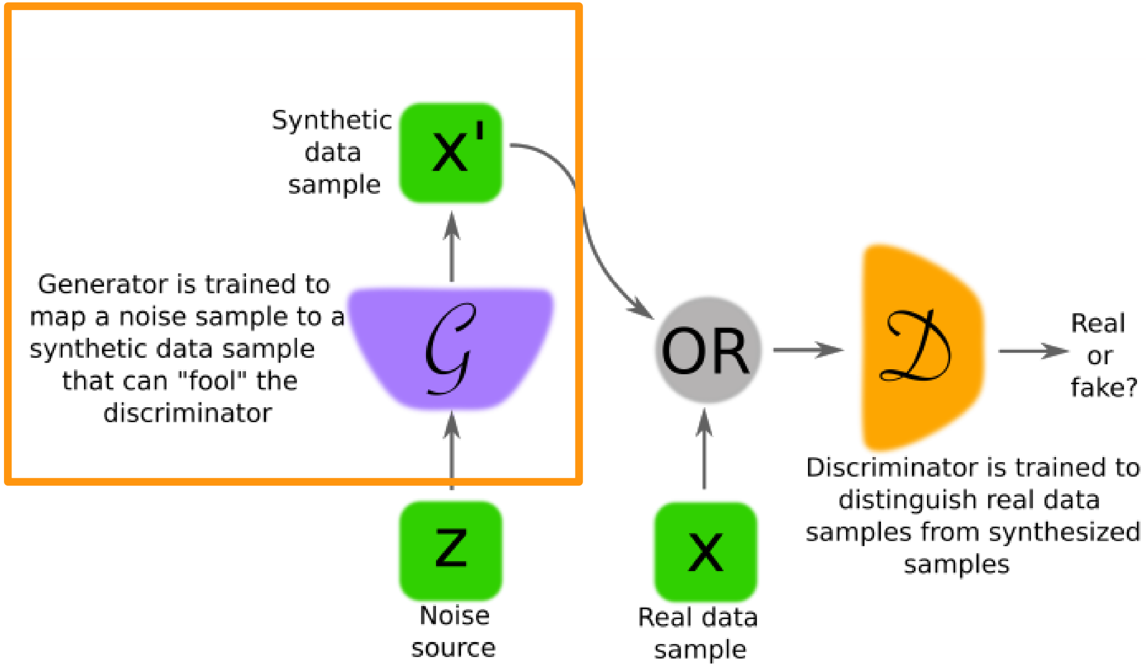
- Groups of DNNs (at least two)
- Working against each other !
- Min parts:
  - Discriminator Network
  - Generator Network



Aufbau von GANs

- In a Nutshell -
Generative Adversarial Nets are:

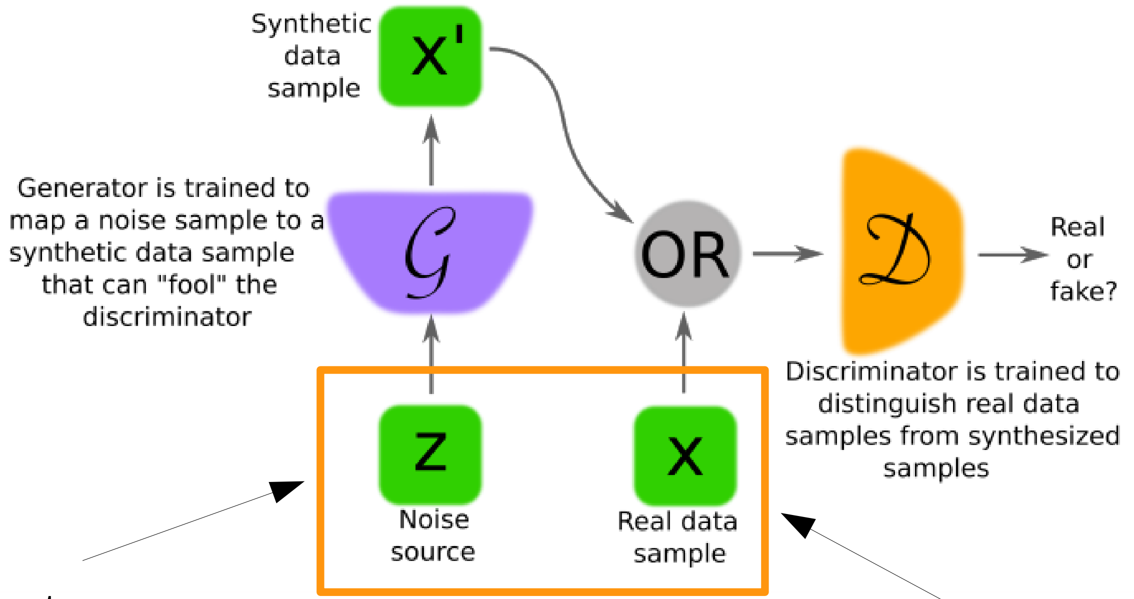
- Groups of DNNs (at least two)
- Working against each other !
- Min parts:
  - Discriminator Network
  - Generator Network



Aufbau von GANs

- In a Nutshell -
Generative Adversarial Nets are:

- Groups of DNNs (at least two)
- Working against each other !
- Min parts:
  - Discriminator Network
  - Generator Network



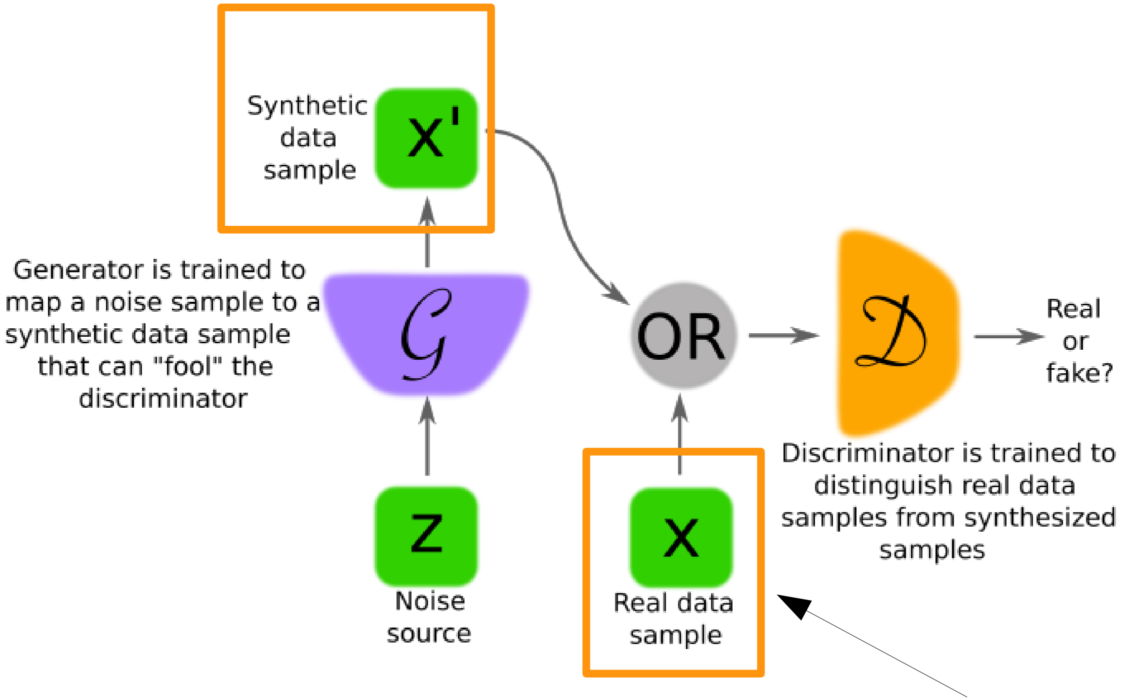
Low dim latent space
Random generated !

Real, un-labeled Data (e.g. Images)

Aufbau von GANs

- In a Nutshell -
Generative Adversarial Nets are:

- Groups of DNNs (at least two)
- Working against each other !
- Min parts:
  - Discriminator Network
  - Generator Network



Real, un-labeled Data (e.g. Images)

Generator Netze

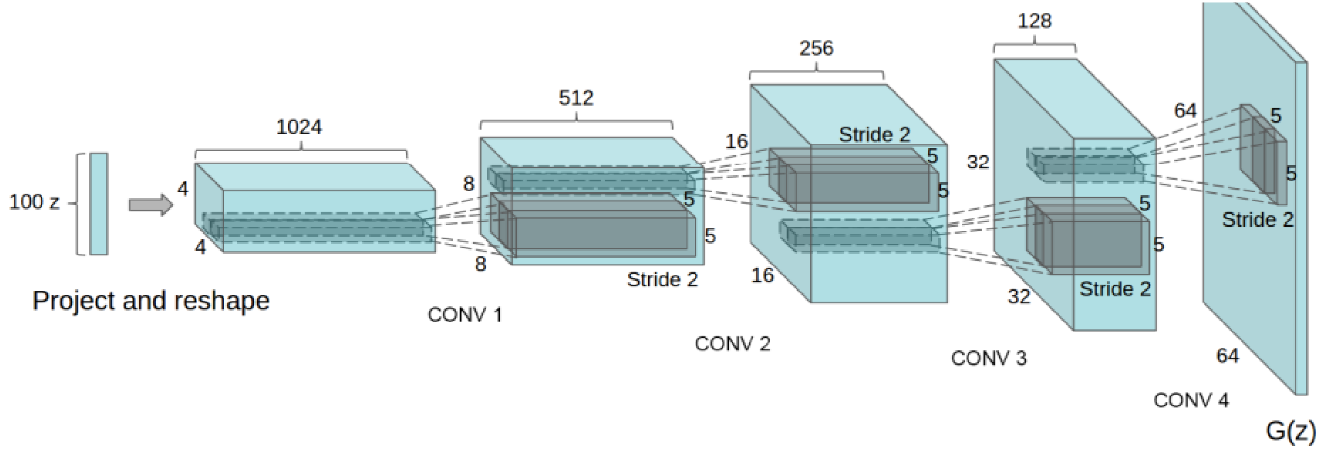
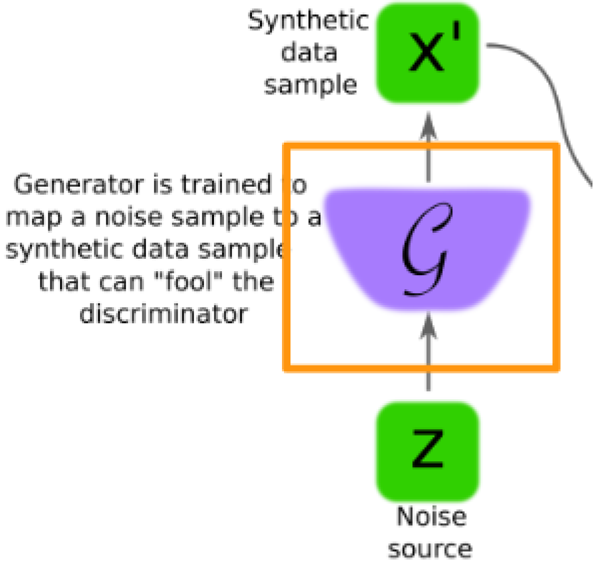
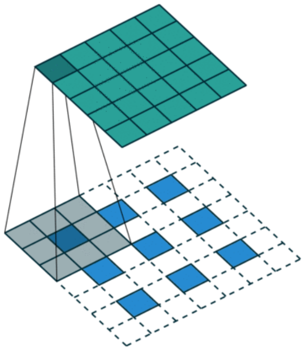


Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64×64 pixel image. Notably, no fully connected or pooling layers are used.



Transposed convolutions

Aufbau von GANs: Up-Convolution

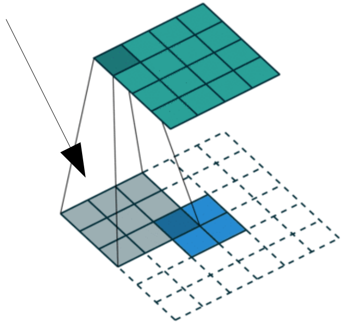


Up-Convolution

Not really a "de-convolution"! (like sometimes proposed)

→ "Transposed Convolution"

Zero padding



[Animationen: https://github.com/vdumoulin/conv_arithmetic]

Training GANs

The cost of training is evaluated using a value function, $V(\mathcal{G}, \mathcal{D})$ that depends on both the generator and the discriminator. The training involves solving:

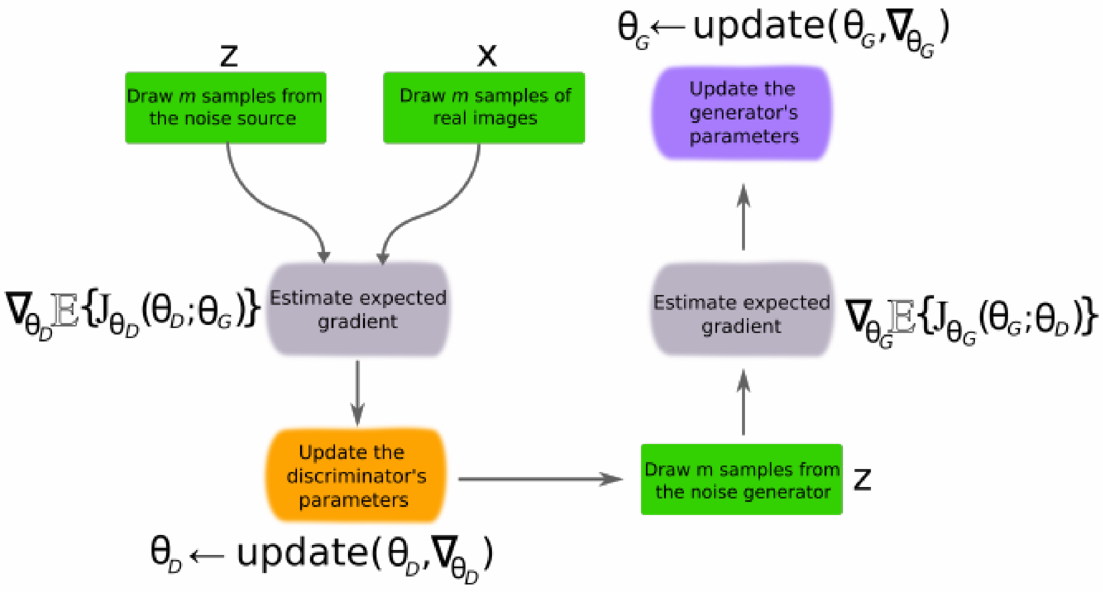
$$\max_{\mathcal{D}} \min_{\mathcal{G}} V(\mathcal{G}, \mathcal{D})$$

where

$$V(\mathcal{G}, \mathcal{D}) = \mathbb{E}_{p_{data}(\mathbf{x})} \log \mathcal{D}(\mathbf{x}) + \mathbb{E}_{p_g(\mathbf{x})} \log(1 - \mathcal{D}(\mathbf{x}))$$

During training, the parameters of one model are updated, while the parameters of the other are fixed.

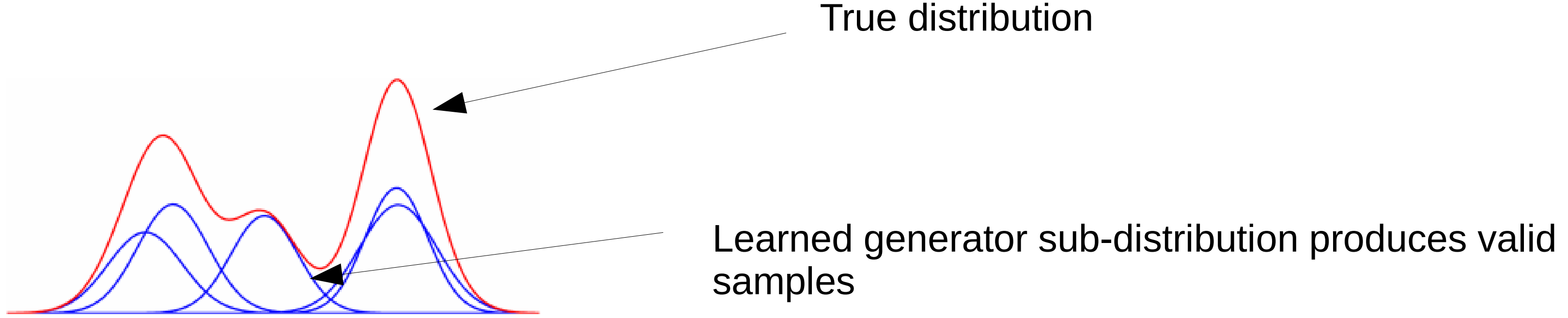
Iterative Training Process via Back Propagation (SGD):



Probleme:

GANs are very hard to train (unstable)

- Generator learns just a small sub-distribution and produces good examples from that
- Rest of the “true” distribution collapses

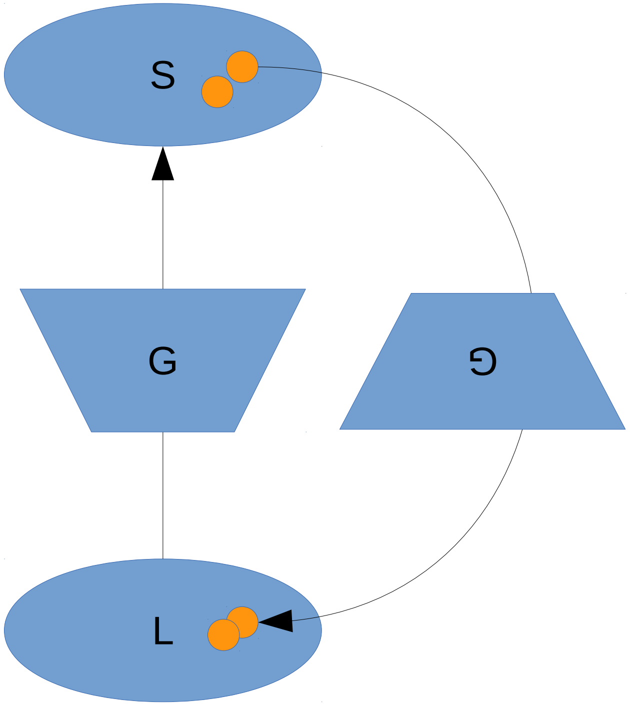


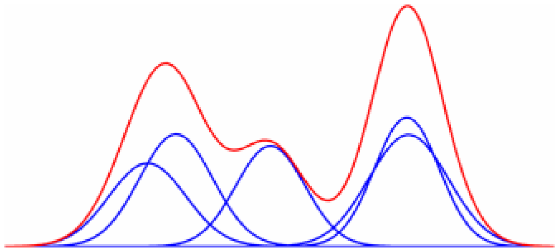
Latent space magic

Even though the latent space is fill with random samples during training, it actually inherits the structure of the generated samples.

Remember: DNNs are somewhat smooth: so neighborhood in latent space should also result in similarity in the sample space.

After a GAN is trained, we can manually add labels to the samples and track label distributions in latent space (and have fun with it :-))





Beispiel: Bildmanipulation → “Attribute transfer learning“

Semi Few-Shot Attribute Translation

Ricard Durall[1,2,3] Franz-Josef Pfreundt[1] Janis Keuper[1,4]
[1]Fraunhofer ITWM, Germany
[2]IWR, University of Heidelberg, Germany
[3]Fraunhofer Center Machine Learning, Germany
[4]Institute for Machine Learning and Analytics, Offenburg University, Germany

***Abstract*—Recent studies have shown remarkable success in image-to-image translation for attribute transfer applications. However, most of existing approaches are based on deep learning and require an abundant amount of labeled data to produce good results, therefore limiting their applicability. In the same vein, recent advances in meta-learning have led to successful implementations with limited available data, allowing so-called *few-shot learning*.**

In this paper, we address this limitation of supervised methods, by proposing a novel approach based on GANs. These are trained in a meta-training manner, which allows them to perform image-to-image translations using just a few labeled samples from a new target class. This work empirically demonstrates the potential of training a GAN for few shot image-to-image translation on hair color attribute synthesis tasks, opening the door to further research on generative transfer learning.

I. Introduction

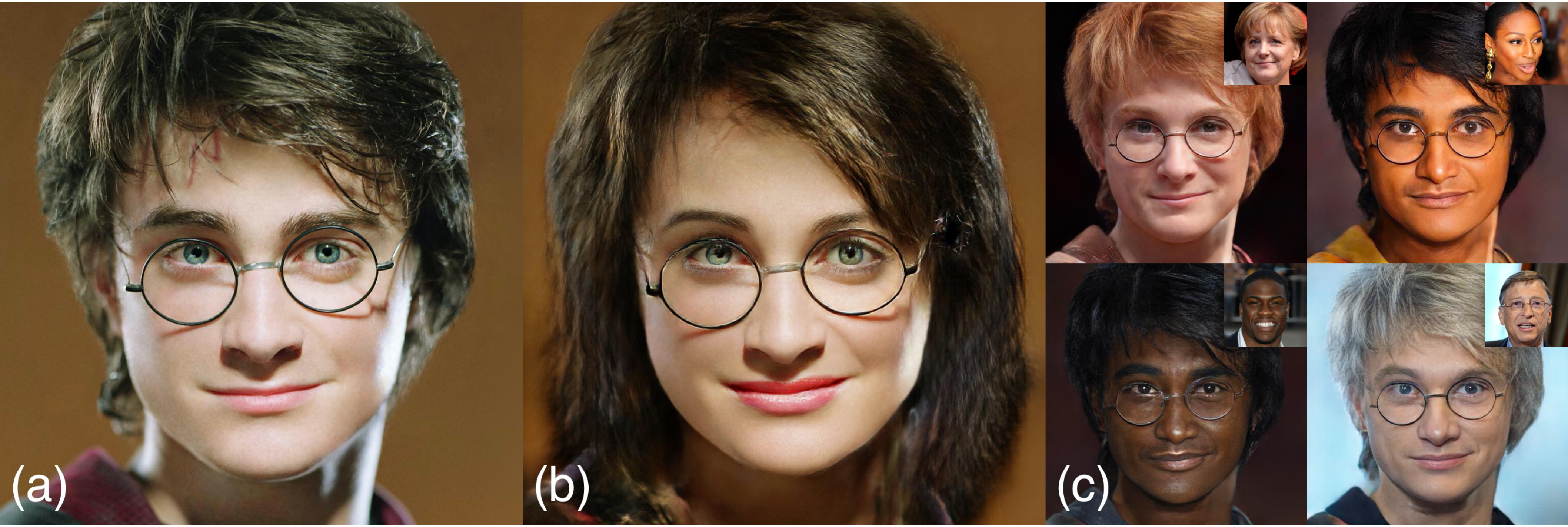
Deep learning models have achieved state-of-the-art performance in large varity of tasks, from image synthesis [1], [2], [3], text style transfer [4], video generation [5], [6] to image-to-image translation [7], [8], [9], [10], [11]. The latter task, image-to-image translation, is a computer vision problem that aims at translating images from one domain to another, including colorization [12], super-resolution [13], style transfer [14], [15], [9], [16], inpainting [7], [17], [18], [8], [19] and attribute transfer [20], [21], [22], [10]. The strong performance, however, heavily relies on training a network with abundant algorithm includes metric-based methods [28], [23], [26], [27], model-based methods [29], [30] and optimization-based methods [24], [25], [31], [32], [33]. These models have allowed learning tasks to perform well on novel data sampled from the same distribution as the training data. These meta-learning algorithms have seen direct applications in supervised and reinforcement learning. Additionally, due to their general applicability, recent works based on meta-learning have successfully been utilized for image generation [34], [35], [36], [37].

The objective of attribute transfer is to synthesize realistic appearing images for a pre-defined target domain. For instance, given an image with a particular attribute "blond hair" (original domain), change it to "black hair" (target domain). We refer to a domain as a set of images sharing the same attributes. Such attributes are meaningful semantic feature inherent in an image such as "smiling" or "face with eyeglasses". After the introduction of generative adversarial networks (GANs) [38], transfer domain algorithms have experienced significant improvements achieving state-of-the-art results in style transfer [14], [15], [9], [16] and in attribute transfer [21], [22], [10]. However, GANs require several orders of magnitude more data points than humans in order to generate comprehensible images successfully from a given class of images [37]. This impairs the ability of GANs to generate novelty. Additionally, in many cases, if the data is abundant enough to successfully train a GAN, there is little purpose to generating more of this data.

(a) Brown hair attribute transform.

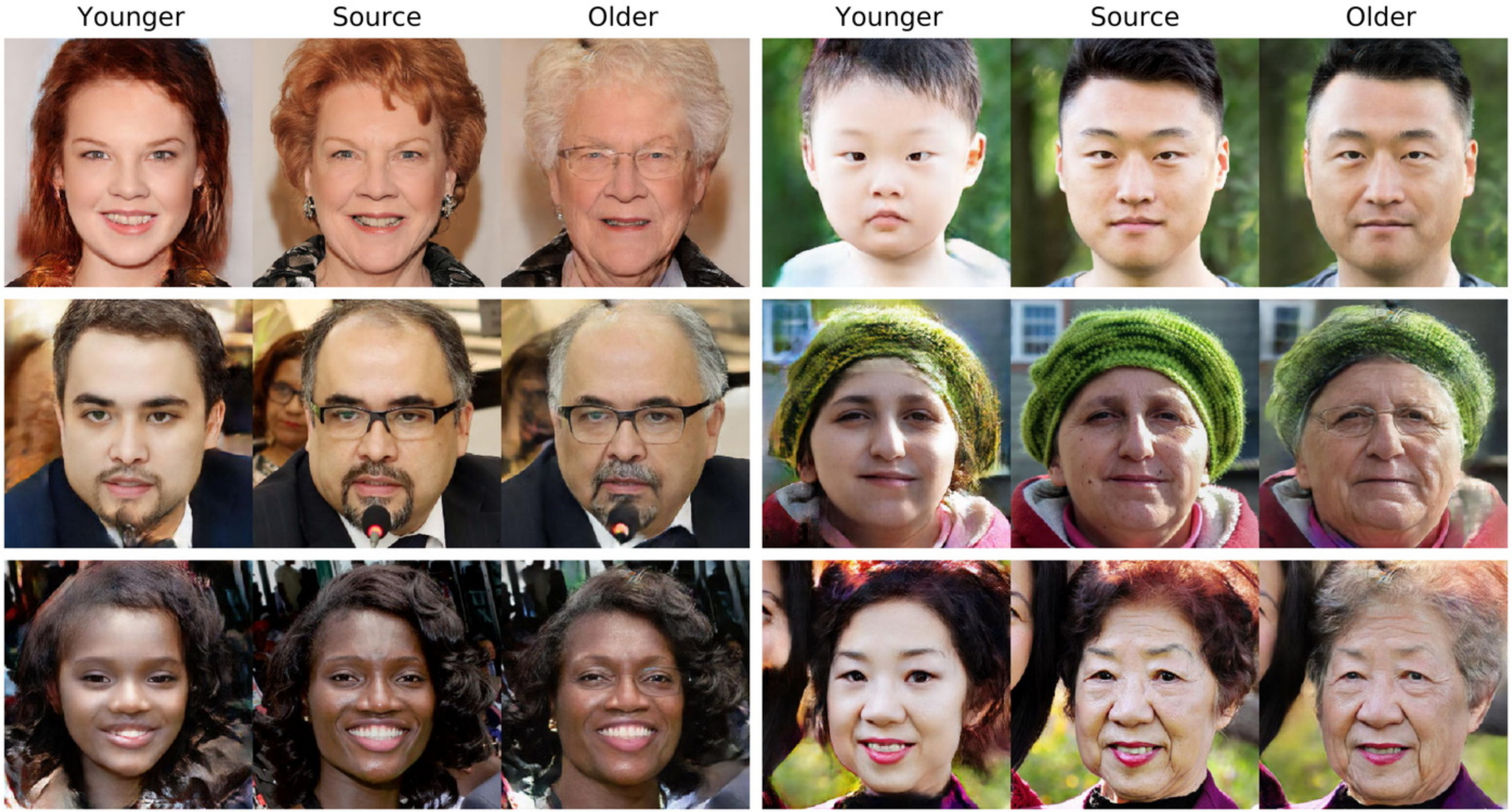
Best student paper ICVNZ 2019

Beispiel: Bildmanipulation → “Attribute transfer learning“



[images from: https://arxiv.org/abs/2003.03581]

Beispiel: Bildmanipulation → “Attribute transfer learning“



Agenda

Teil I : Einführung
– Spielereien und Beispiele

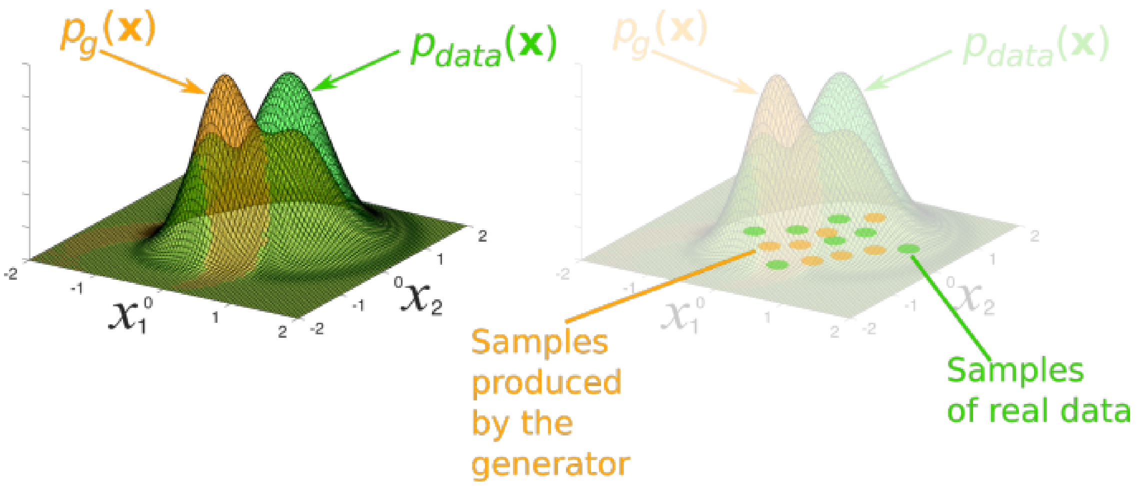
Teil II : Methoden
– Generative Modelle

Teil III: Anwendungen
- praktischer Nutzen generativer Modelle

Teil IV: Gefahren
- Deepfakes

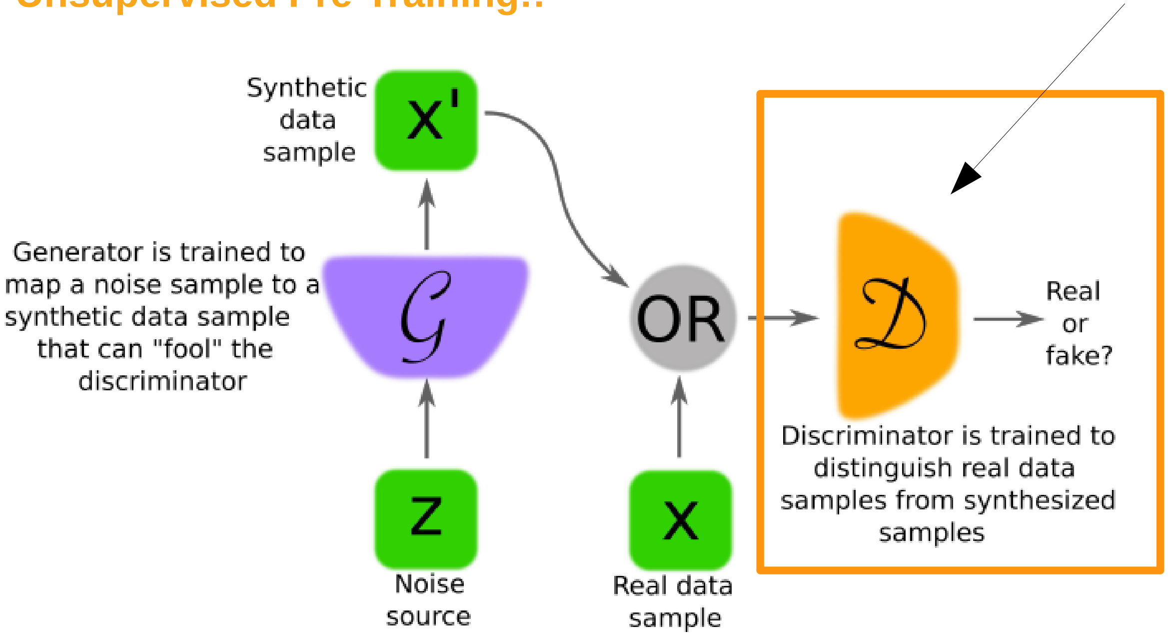
ML Theorie

A GAN is basically a distribution approximator



Anwendung

- **Generating synthetic training data (auto data augmentation)**
- **Unsupervised Pre-Training!:**



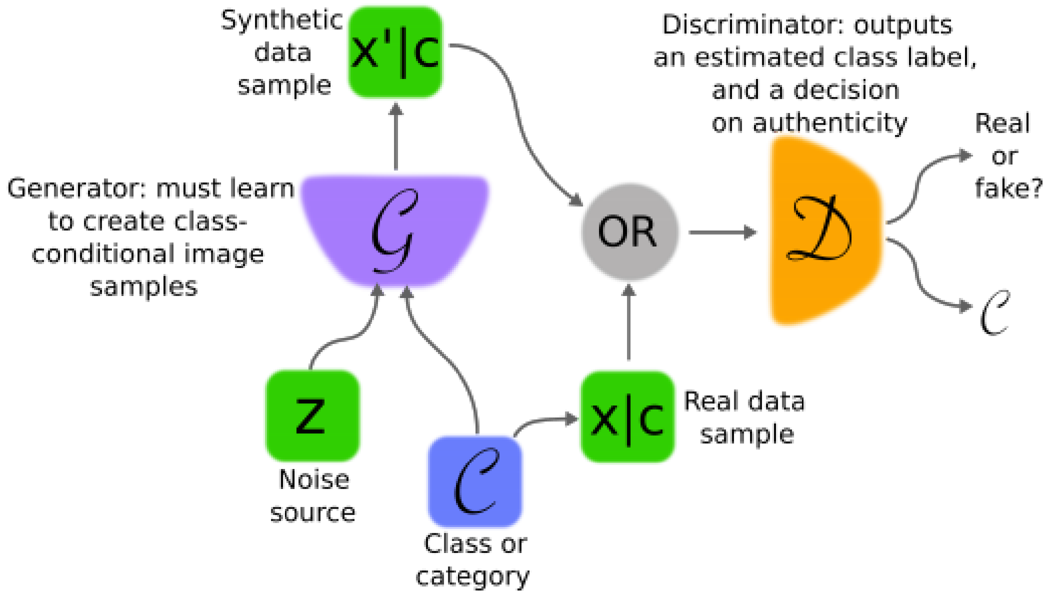
Assume this is a CNN:

Training the Discriminator will train Problem specific conv weights **without labels** !

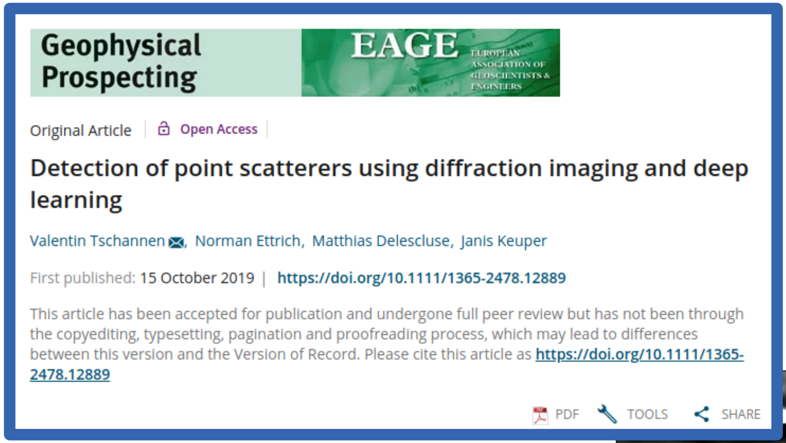
Can then be used for Classification tasks with little annotated data...

Anwendung

- **Learning class distributions**



Anwendung Beispiel



Geophysical Prospecting

EAGE EUROPEAN ASSOCIATION OF GEOSCIENTISTS & ENGINEERS

Original Article | Open Access |

Detection of point scatterers using diffraction imaging and deep learning

Valentin Tschannen, Norman Ettrich, Matthias Delescluse, Janis Keuper

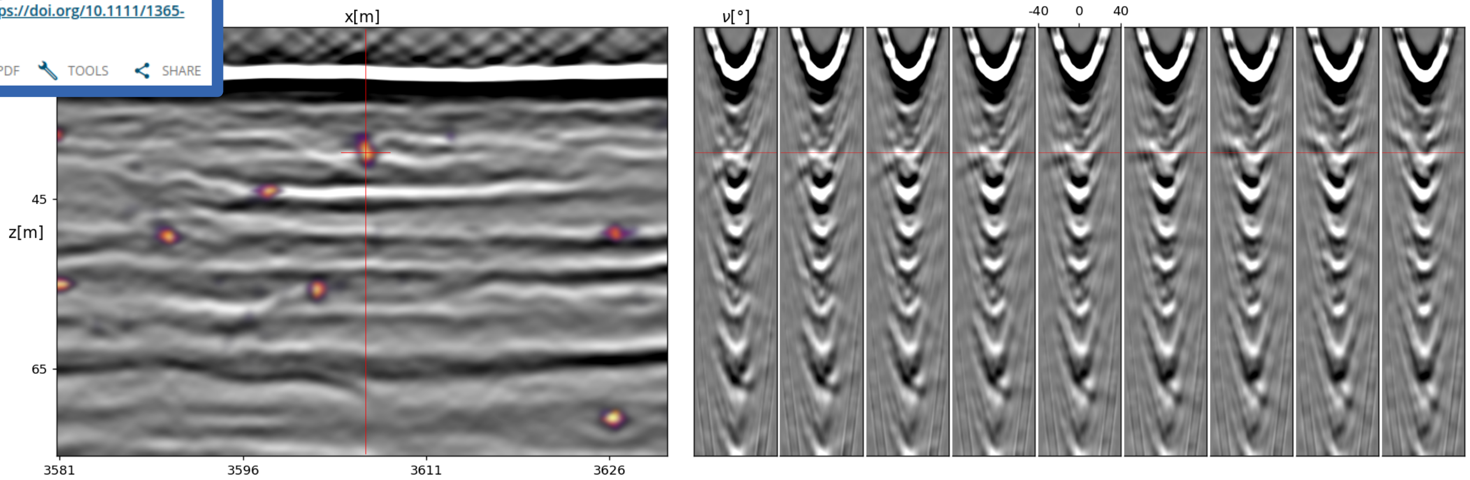
First published: 15 October 2019 | https://doi.org/10.1111/1365-2478.12889

This article has been accepted for publication and undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process, which may lead to differences between this version and the Version of Record. Please cite this article as https://doi.org/10.1111/1365-2478.12889

PDF TOOLS SHARE

Anwendung: Objektdetektion in 4D Seismischen Daten

- Manuelle Annotation sehr schwierig
- Pre-Training mit GAN
- Transfer learning von Simulationsdaten



Agenda

Teil I : Einführung
– Spielereien und Beispiele

Teil II : Methoden
– Generative Modelle

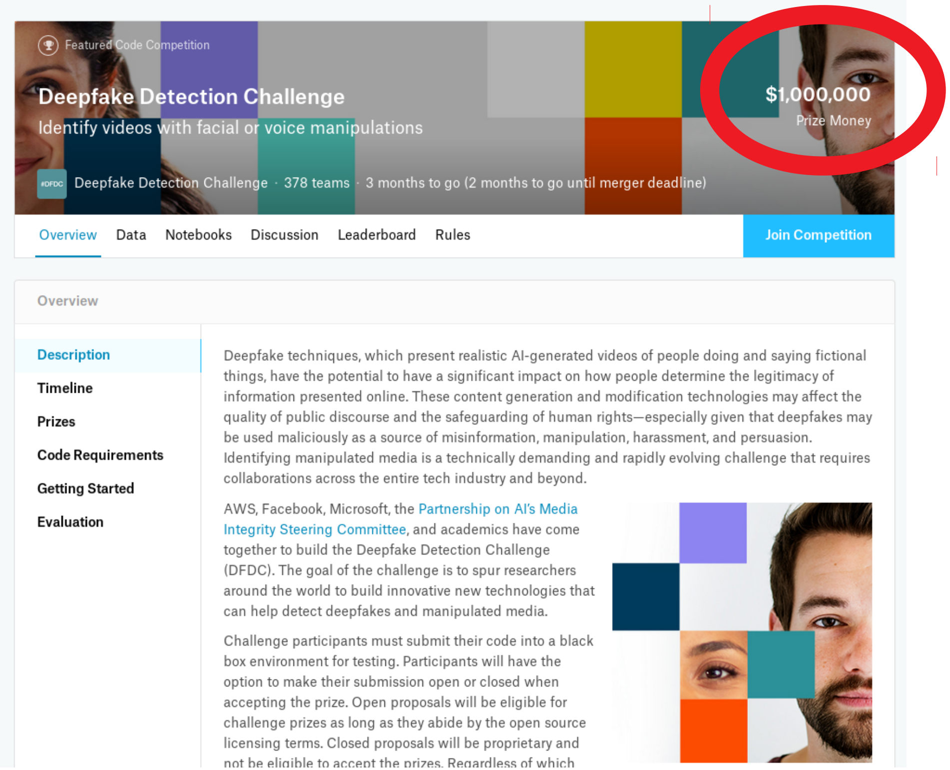
Teil III: Anwendungen
- praktischer Nutzen generativer Modelle

Teil IV: Gefahren
- Deepfakes

Deepfakes

https://www.youtube.com/watch?v=cQ54GDm1eL0

Deepfakes



https://www.kaggle.com/c/deepfake-detection-challenge/overview

Deepfakes

Übliche Ansätze:

- Nutze CNNs um fakes zu klassifizieren
  - Woher kommen die Trainingsdaten?
  - Supervised ML deckt nur bekannte Manipulationen ab

- Alternative: Finde Inhärente Fehler

Unmasking DeepFakes with simple Features

Ricard Durall[1,2,3] Margret Keuper[4] Franz-Josef Pfreundt[1] Janis Keuper[1,5]
[1]Fraunhofer ITWM, Germany
[2]IWR, University of Heidelberg, Germany
[3]Fraunhofer Center Machine Learning, Germany
[4]Data and Web Science Group, University Mannheim, Germany
[5]Institute for Machine Learning and Analytics, Offenburg University, Germany

***Abstract*—Deep generative models have recently achieved impressive results for many real-world applications, successfully generating high-resolution and diverse samples from complex data sets. Due to this improvement, fake digital contents have proliferated growing concern and spreading distrust in image content, leading to an urgent need for automated ways to detect these AI-generated fake images.**

Despite the fact that many face editing algorithms seem to produce realistic human faces, upon closer examination, they do exhibit artifacts in certain domains which are often hidden to the naked eye. In this work, we present a simple way to detect such fake face images - so-called *DeepFakes*. Our method is based on a classical frequency domain analysis followed by a basic classifier. Compared to previous systems, which need to be fed with large amounts of labeled data, our approach showed very good results using only a few annotated training samples and even achieved good accuracies in fully unsupervised scenarios. For the evaluation on high resolution face images, we combined several public data sets of real and fake faces into a new benchmark: *Faces-HQ*. Given such high-resolution images, our approach reaches a perfect classification accuracy of 100% when it is trained on as little as 20 annotated samples. In a second experiment, in the evaluation of the medium-resolution images of the *CelebA* data set, our method achieves 100% accuracy supervised and 96% in an unsupervised setting. Finally, evaluating a low-resolution video sequences of the *FaceForensics++* data set, our method achieves 91% accuracy detecting manipulated videos.

Source Code: https://github.com/cc-hpc-itwm/DeepFakeDetection

***Index Terms*—GAN images, DeepFake, Image forensic, Forgery detection**

I. INTRODUCTION

Over the last years, the increasing sophistication of smartphones and the growth of social networks have led to a gigantic amount of new digital object contents. This tremendous use of digital images has been followed by a rise of techniques to alter image contents. Until recently, such techniques were beyond the reach of most users since they were dull and time-consuming and they required a high domain expertise on computer vision. Nevertheless, thanks to the recent advances of machine learning and the accessibility to large-volume

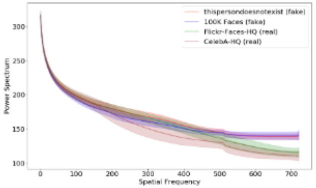


Fig. 1: 1D power spectrum statistics from each sub-data set from Faces-HQ. The higher the frequency, the bigger is the difference between real or fake data.

In particular, deep generative models have lately been extensively used to produce artificial images with realistic appearance. Theses models are based on deep neural networks which are able to approximate the true data distribution of a given training set. Hence, one can sample from the learned distribution and add variations. Two of the most commonly used and efficient approaches are Variational Autoencoders (VAE) [16] and Generative Adversarial Networks (GAN) [11]. Especially GAN approaches have lately been pushing the limits of state-of-the-art results, improving the resolution and quality of images produced [4], [14], [15]. As a result, deep generative models are opening the door to a new vein of AI-based fake image generation leading to a fast dissemination of high quality tampered image content. While significant developments have been made for image forgery detection, it still remains a hard task since most current methods rely on deep learning approaches, which require large amounts of labeled training data.

In this paper, we address the problem of detecting these artificial image contents, more specifically, fake faces. In order to determine the nature of these pictures, we introduce a new

https://arxiv.org/pdf/1911.00686.pdf

Deepfakes

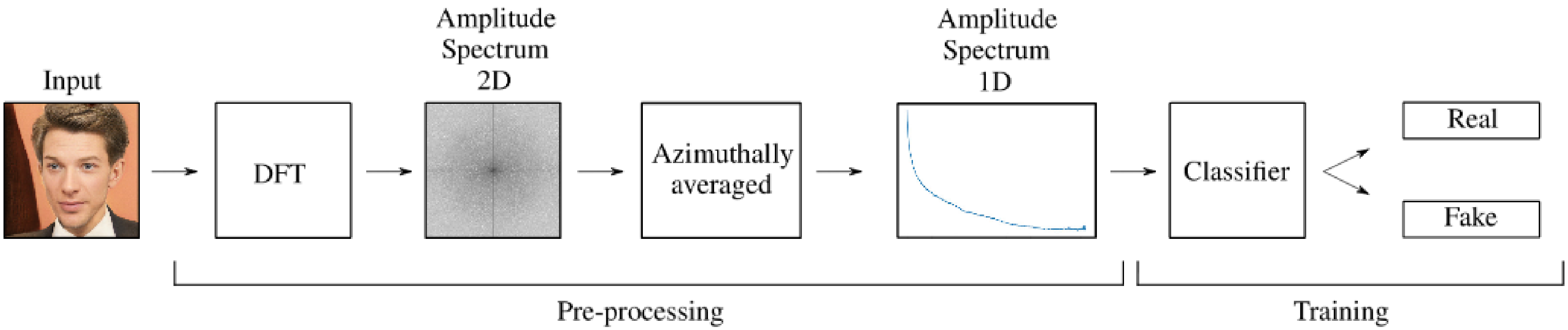


Fig. 2: Overview of the processing pipeline of our approach. It contains two main blocks, a feature extraction block using DFT and a training block, where a classifier uses the new transformed features to determine whether the face is real or not. Notice that input images are transformed to grey-scale before DFT.

Deepfakes

# samples	80% (train) - 20% (test)		
	SVM	Logistic Reg.	K-Means
2000	100%	100%	96%

TABLE VI: Test accuracy using SVM, logistic regression and k-means classifier.

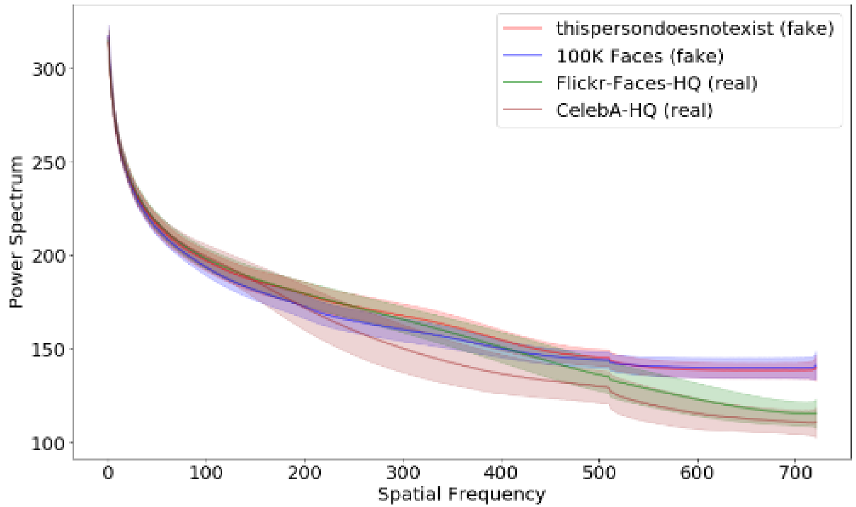


Fig. 1: 1D power spectrum statistics from each sub-data set from Faces-HQ. The higher the frequency, the bigger is the difference between real or fake data.

Fixing Deepfakes

CVPR 2020 Paper

Pre-print: https://arxiv.org/pdf/2003.01826.pdf

Watch your Up-Convolution: CNN Based Generative Deep Neural Networks are Failing to Reproduce Spectral Distributions

Ricard Durall[1,3], Margret Keuper[2], Janis Keuper[1,4]

[1]Competence Center High Performance Computing, Fraunhofer ITWM, Kaiserslautern, Germany
[2]Data- and Webscience Group, University Mannheim, Germany
[3]IWR, University of Heidelberg, Germany
[4]Institute for Machine Learning and Analytics, Offenburg University, Germany

Abstract

Generative convolutional deep neural networks, e.g. popular GAN architectures, are relying on convolution based up-sampling methods to produce non-scalar outputs like images or video sequences. In this paper, we show that common up-sampling methods, i.e. known as up-convolution or transposed convolution, are causing the inability of such models to reproduce spectral distributions of natural training data correctly. This effect is independent of the underlying architecture and we show that it can be used to easily detect generated data like deepfakes with up to 100% accuracy on public benchmarks. To overcome this drawback of current generative models, we propose to add a novel spectral regularization term to the training optimization objective. We show that this approach not only allows to train spectral consistent GANs that are avoiding high frequency errors. Also, we show that a correct approximation of the frequency spectrum has positive effects on the training stability and output quality of generative networks.

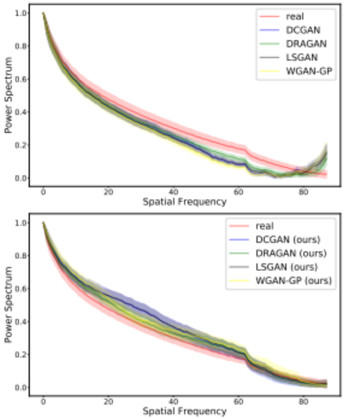
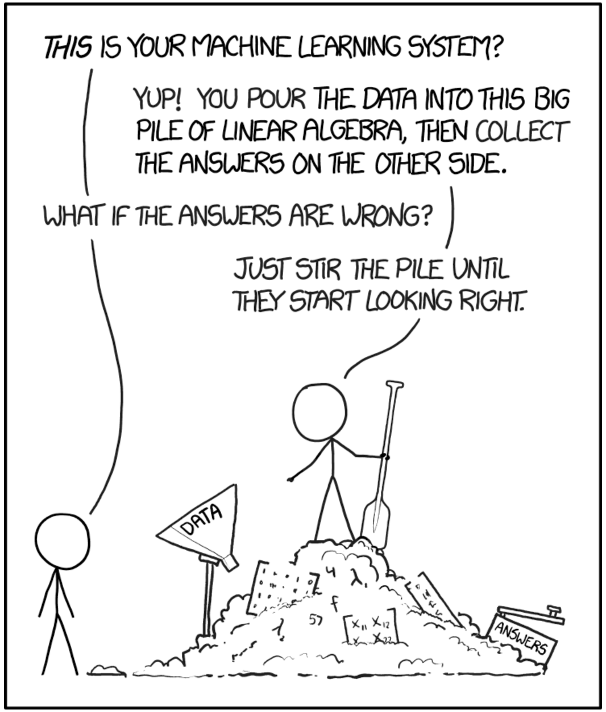


Figure 1: Common up-convolution methods are inducing heavy spectral distortions into generated images. The **top** figure shows the statistics (mean and variance) after azimuthal integration over the power-spectrum (see Section

1. Introduction

Generative convolutional deep neural networks have re-



Diskussion

Kontakt: keuper@imla.ai

Überschrift