Recent Advances in Machine Learning

Apr. 22th, 2022

Detecting Generated Images

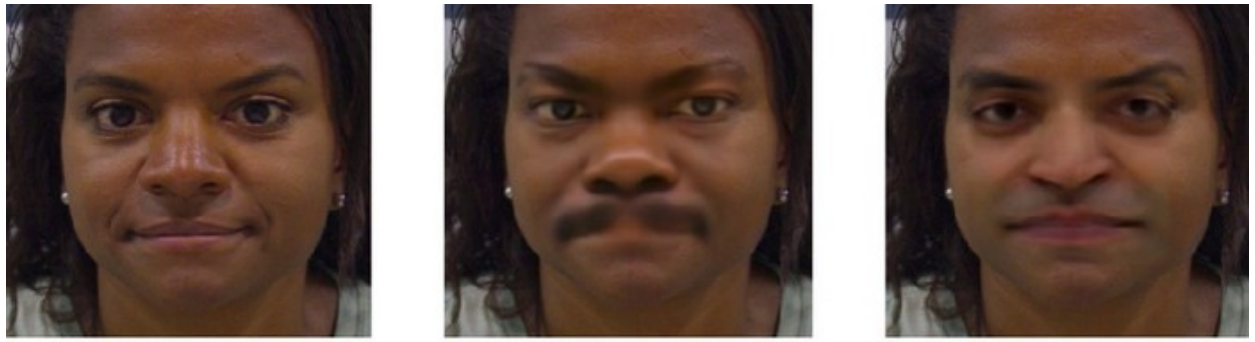
Margret Keuper
Professor for Visual Computing
University of Siegen

Image Generation with Neural Networks

- Can have various motivations
  - Solving inverse problems (see lecture by Prof. Möller)
  - Create artificial but **realistic** image content (aka “DeepFakes”)
  - Create Art
  - Augment training data for supervised image classification
  - Analysis of you classification network (deconvoutional Networks, DeepDreams)
- Can employ various neural network architectures
  - Encoder - Decoder Architectures (similar to AE for Reconstruction)
  - Variational Autoencoders
  - Adversarial Networks
  - Autoregressive Flow

Machine Learning

- With the recent success in image generation, questions on authenticity have become very important.
  - Deep Fake Detection



Face forensics dataset

Real or Generated?

DeepFake Detection

Outline

- **What is a DeepFake?**
- **Generating Data with Convolutional Neural Networks**
- **Generative Adversarial Networks**
  - Generating fake faces
- **DeepFake Detection**

What is a DeepFake?

“Deepfakes (a portmanteau of "deep learning" and "fake") are media that take a person in an existing image or video and replace them with someone else's likeness using artificial neural networks. They often combine and superimpose existing media onto source media using machine learning techniques known as autoencoders and generative adversarial networks (GANs).” - Wikipedia

Demo I

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

Demo II

http://www.whichfaceisreal.com/index.php

Click on the person who is real.

Demo III

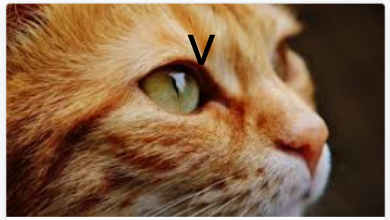
https://thisxdoesnotexist.com/

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

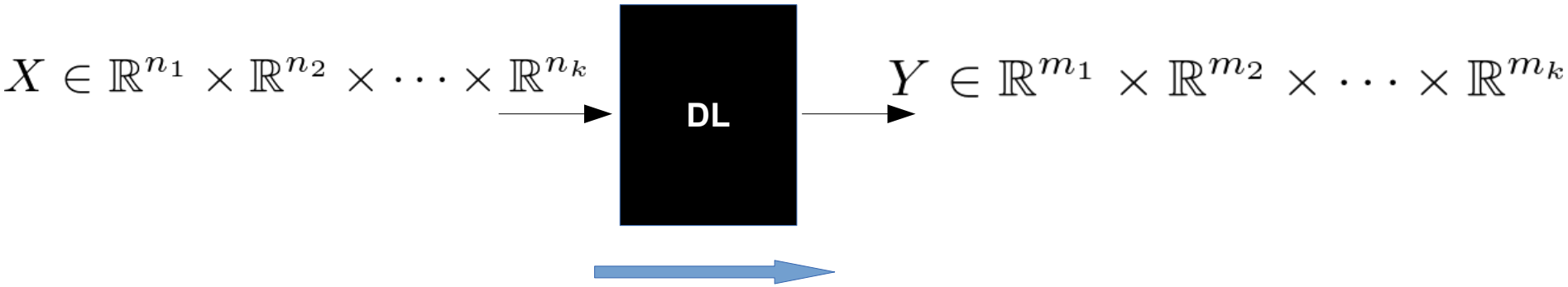
These purr-fect GAN-made cats will freshen your feeline-gs and make you wish you could reach through your screen and cuddle them. Once in a while the cats have visual deformities due to imperfections in the model – beware, they can cause nightmares.

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.

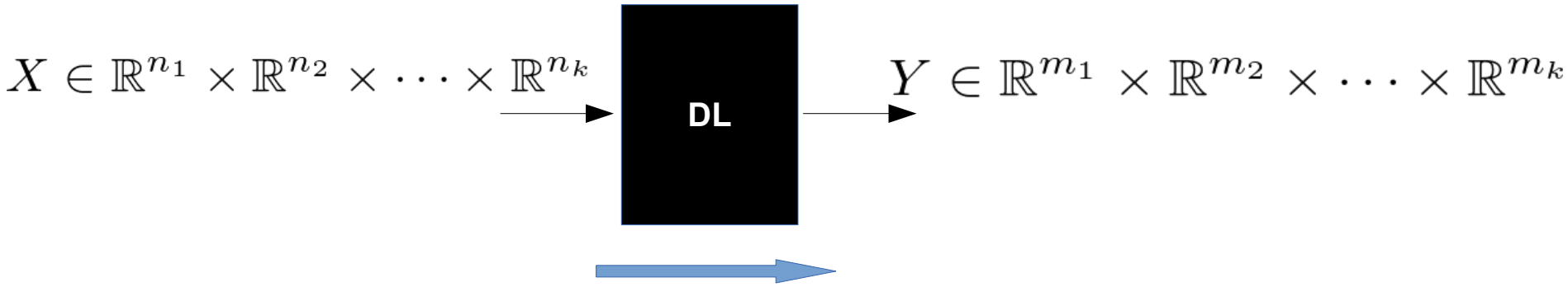
Motivation

Recall DL mapping capacities

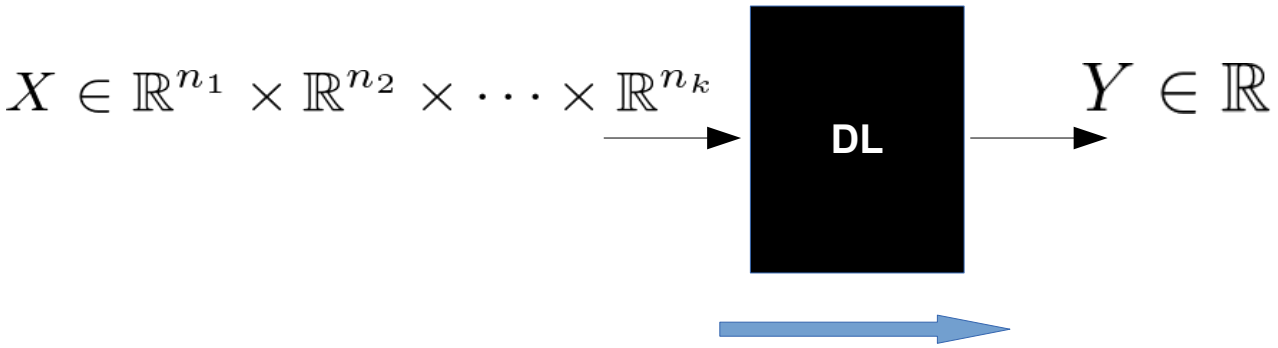


Motivation

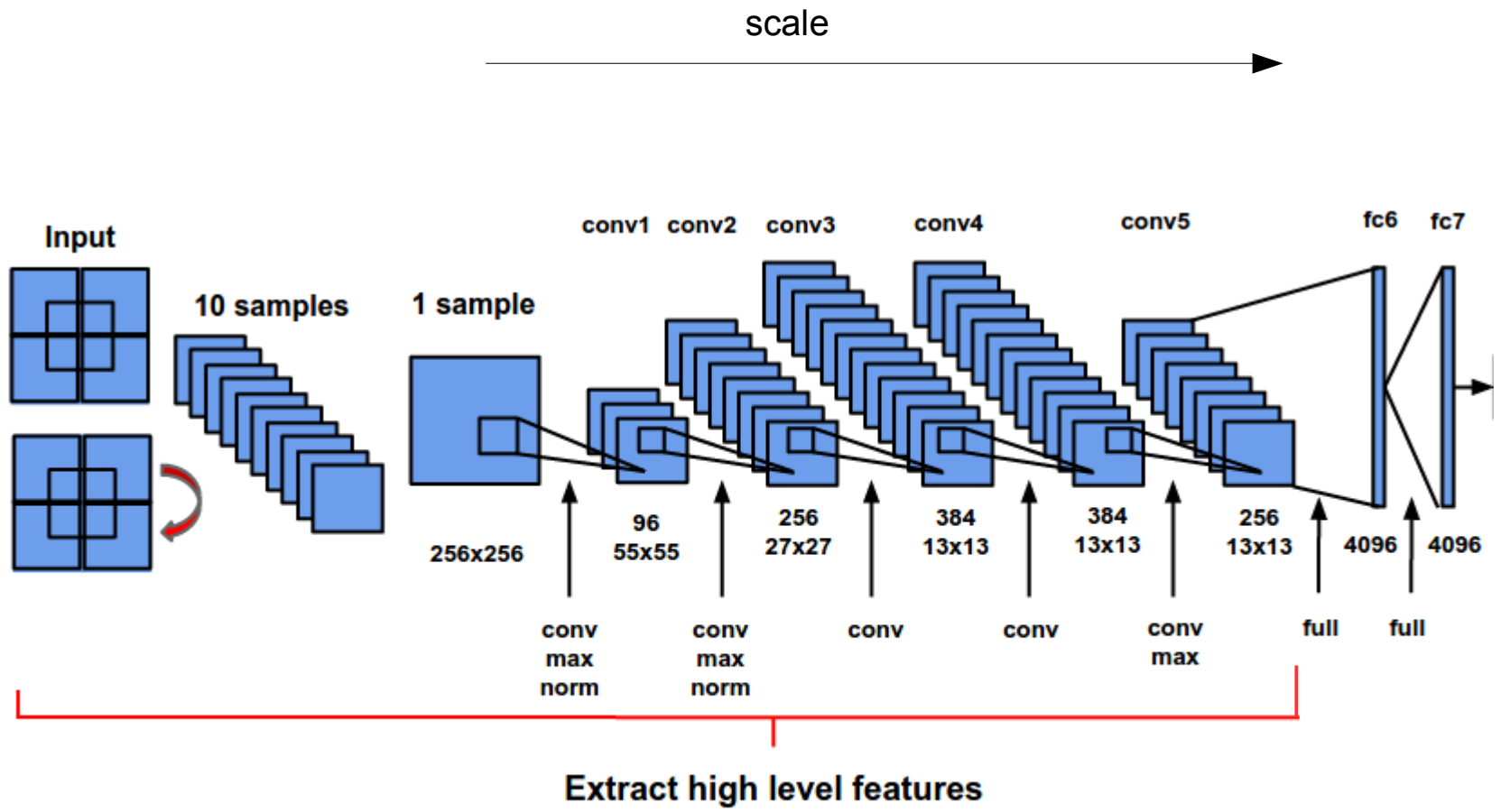
Recall DL mapping capacities – so far only abstract theory and some application examples



CNNs for Classification:



Recall: Convolutional Neural Networks



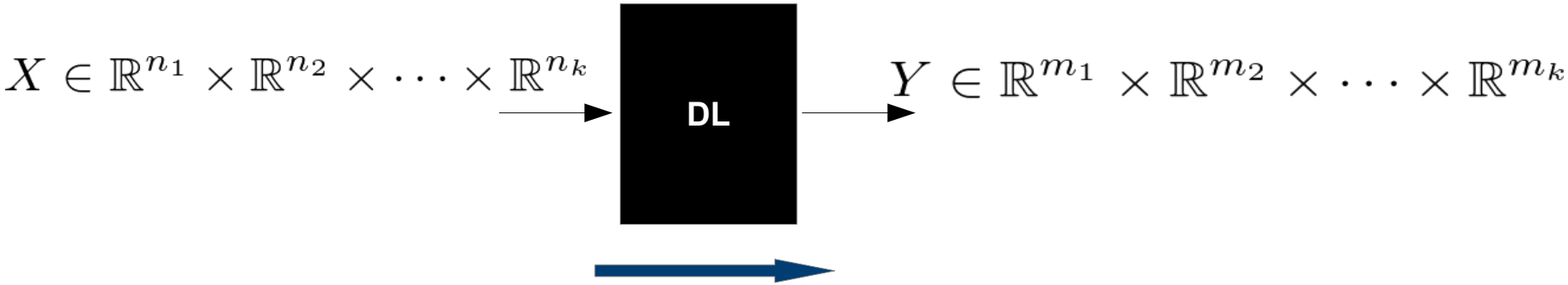
AlexNet

- Can have various motivations
  - Solving inverse problems (see lecture by Prof. Möller)
  - Create artificial but **realistic** image content (aka “DeepFakes”)
  - Create Art
  - Augment training data for supervised image classification
  - Analysis of you classification network (deconvoutional Networks, DeepDreams)
- Can employ various neural network architectures
  - Encoder - Decoder Architectures (similar to AE for Reconstruction)
  - Variational Autoencoders
  - Adversarial Networks
  - Autoregressive Flow

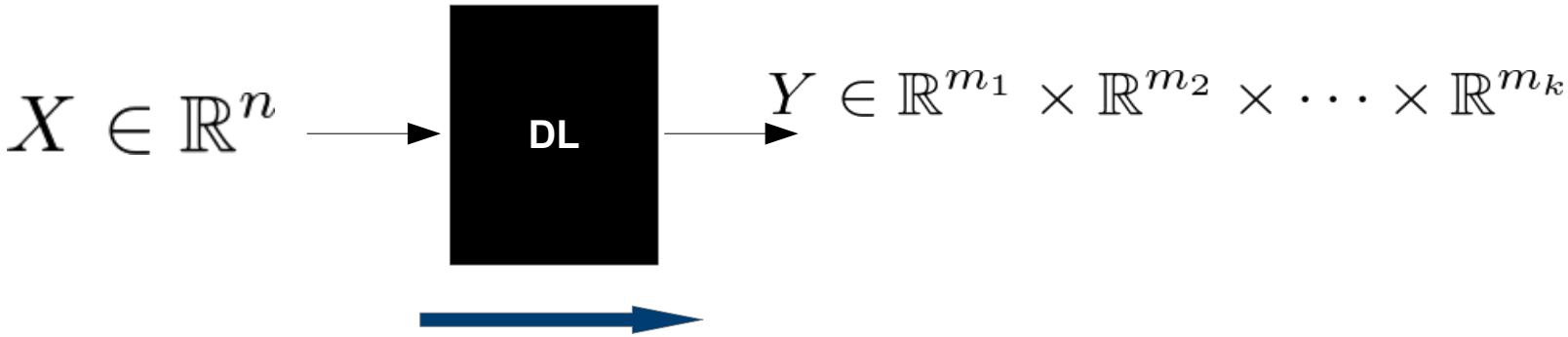
- Can have various motivations
  - Solving inverse problems (see lecture by Prof. Möller)
  - Create artificial but **realistic** image content (aka “DeepFakes”)
  - Create Art
  - Augment training data for supervised image classification
  - Analysis of you classification network (deconvoutional Networks, DeepDreams)
- Can employ various neural network architectures
  - **Encoder - Decoder Architectures (similar to AE for Reconstruction)**
  - Variational Autoencoders
  - **Adversarial Networks**
  - Autoregressive Flow

Motivation

Let's see this first



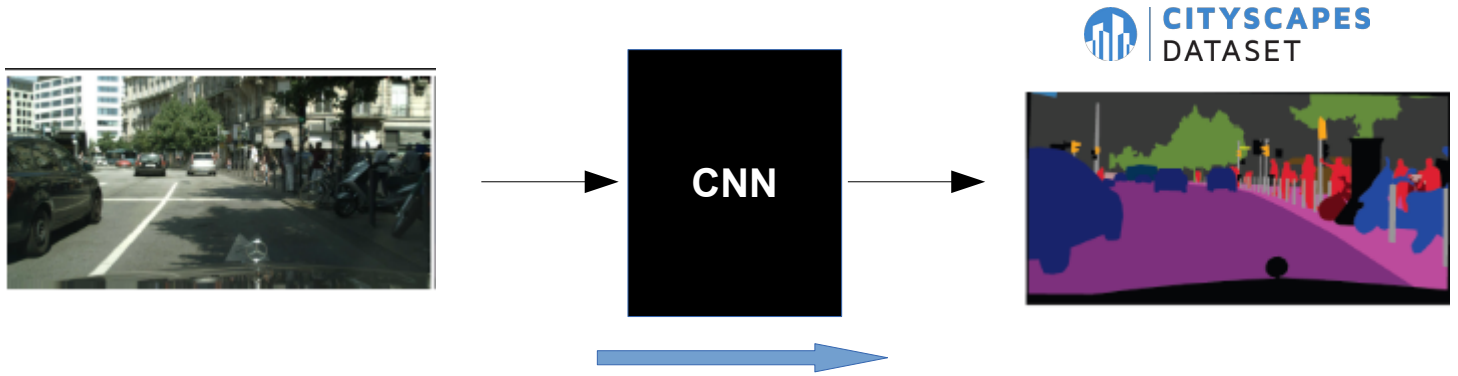
Then this



Generator Networks

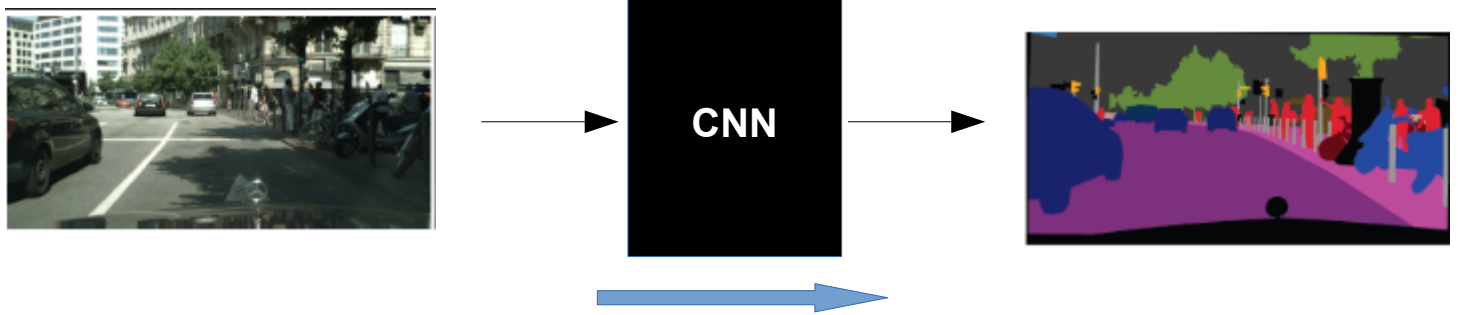
Increasing Output Scale in Networks

Let's look at the ***semantic segmentation*** example:

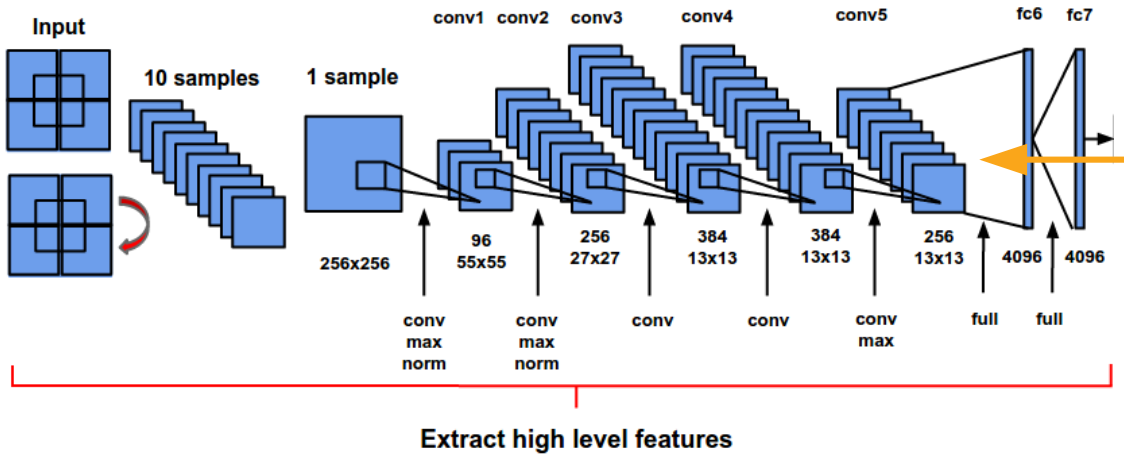


Increasing Output Scale in Networks

Let's look at the ***semantic segmentation*** example:



AlexNet

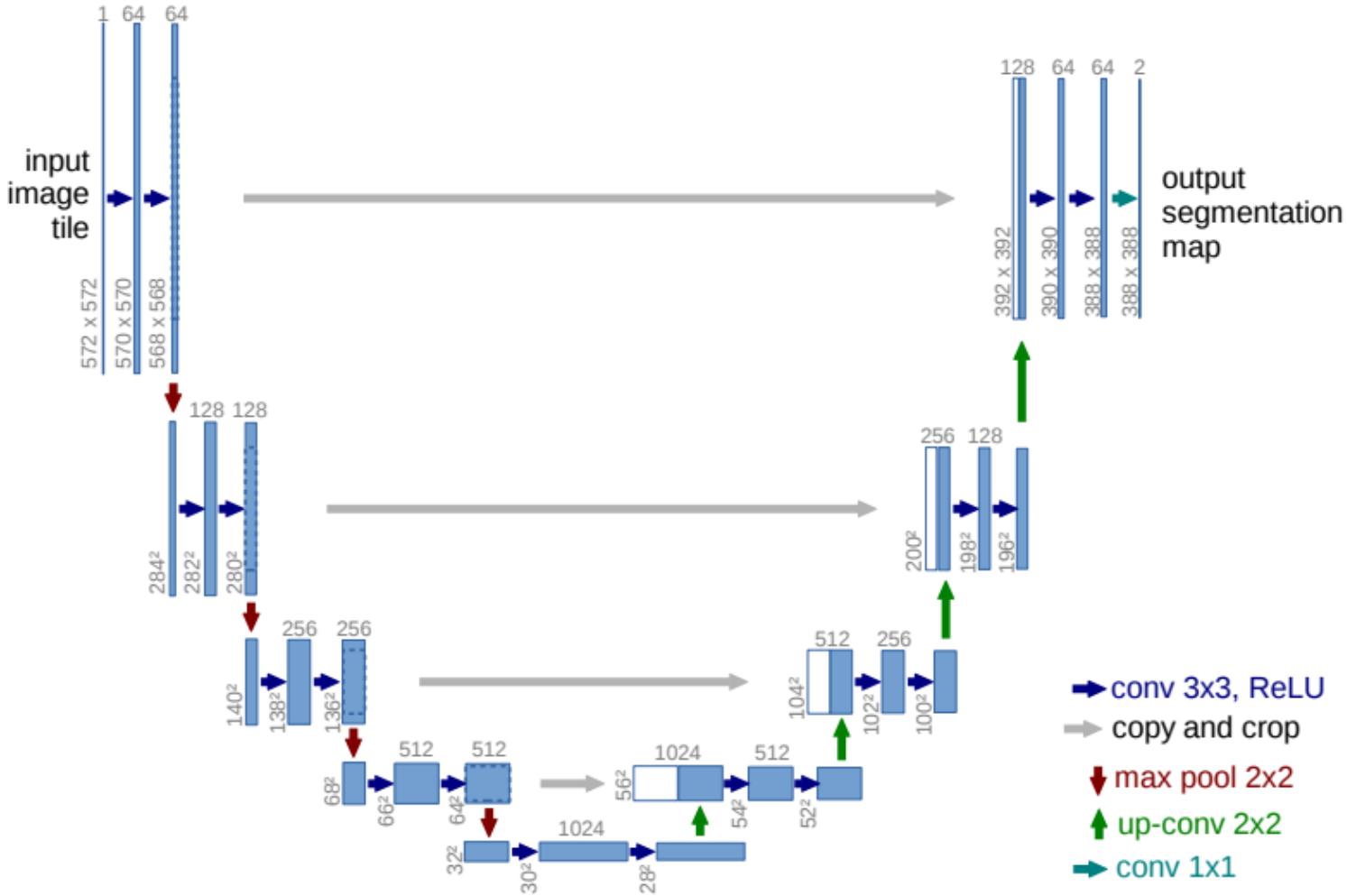


Very low spacial resolution

- **Convolution / stride**
- **Pooling**

How to make this work?

Increasing Output Scale in Networks



U-Net: Convolutional Networks for Biomedical Image Segmentation

Olaf Ronneberger, Philipp Fischer, and Thomas Brox

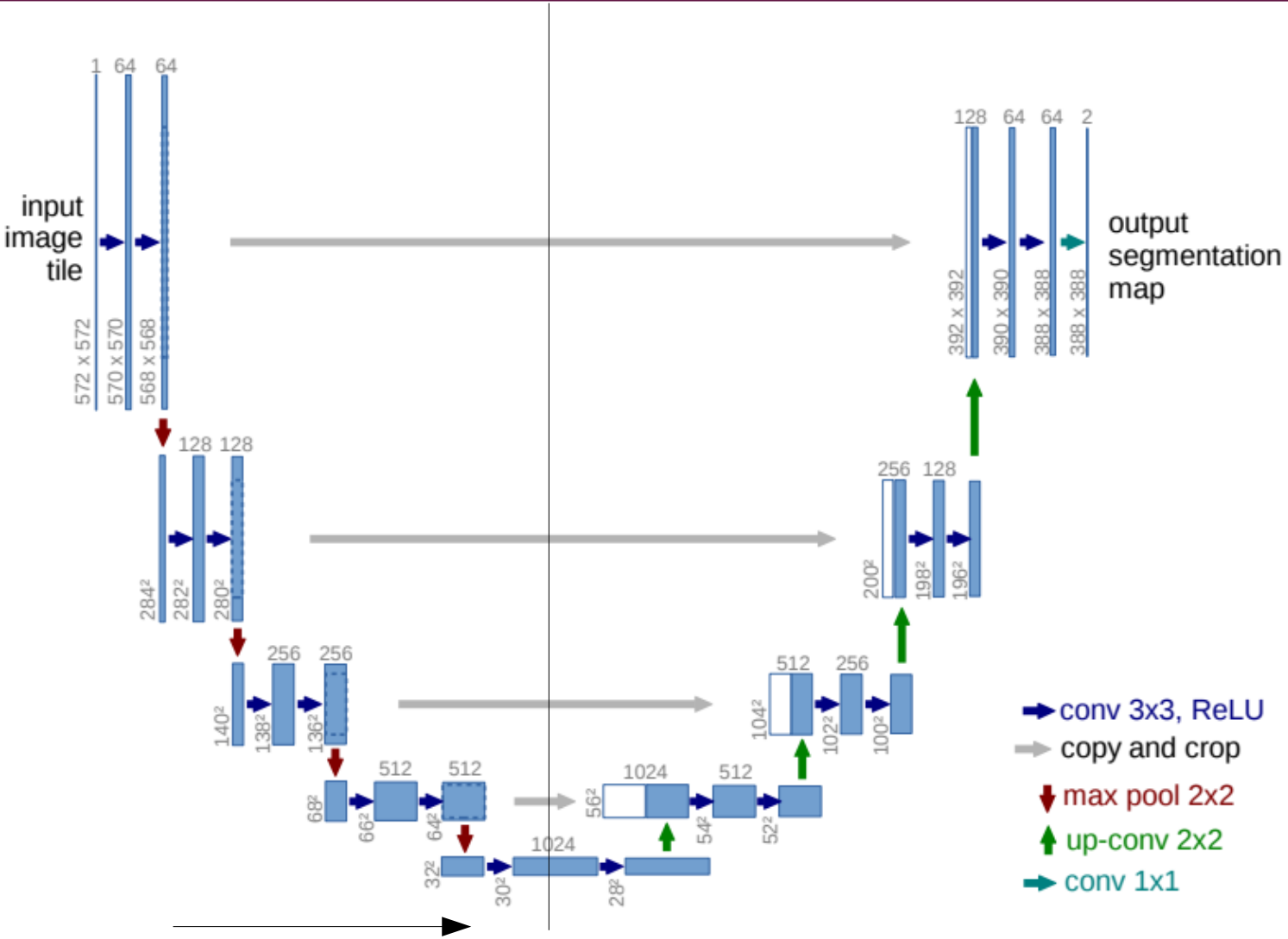
Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany
ronneber@informatik.uni-freiburg.de,
WWW home page: http://lmb.informatik.uni-freiburg.de/

Abstract. There is large consent that successful training of deep networks requires many thousand annotated training samples. In this paper, we present a network and training strategy that relies on the strong use of data augmentation to use the available annotated samples more efficiently. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. We show that such a network can be trained end-to-end from very few images and outperforms the prior best method (a sliding-window convolutional network) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks. Using the same network trained on transmitted light microscopy images (phase contrast and DIC) we won the ISBI cell tracking challenge 2015 in these categories by a large margin. Moreover, the network is fast. Segmentation of a 512x512 image takes less than a second on a recent GPU. The full implementation (based on Caffe) and the trained networks are available at http://lmb.informatik.uni-freiburg.de/people/ronneber/u-net.

1 Introduction

In the last two years, deep convolutional networks have outperformed the state of the art in many visual recognition tasks, e.g. [7,3]. While convolutional networks have already existed for a long time [8], their success was limited due to the size of the available training sets and the size of the considered networks. The breakthrough by Krizhevsky et al. [7] was due to supervised training of a large network with 8 layers and millions of parameters on the ImageNet dataset with

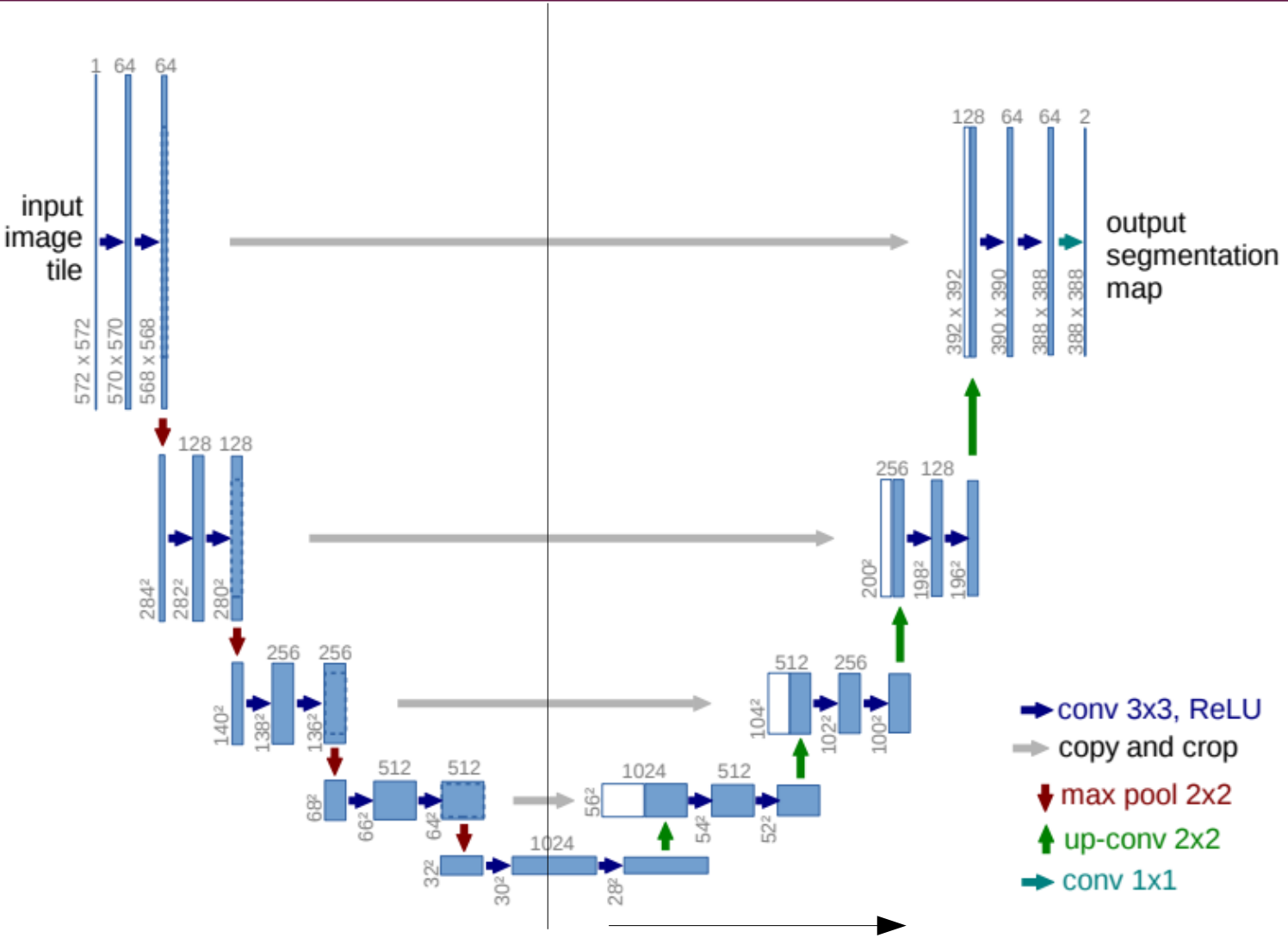
Increasing Output Scale in Networks



First half of U-Net:

“classic” CNN topology

Increasing Output Scale in Networks



First half of U-Net:

“classic” CNN topology

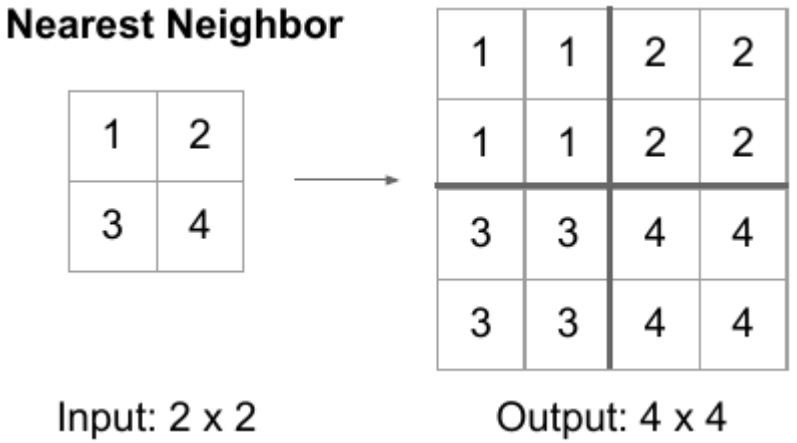
Second half:

“inverse” CNN

Increasing Output Scale in Networks

Upsampling Techniques

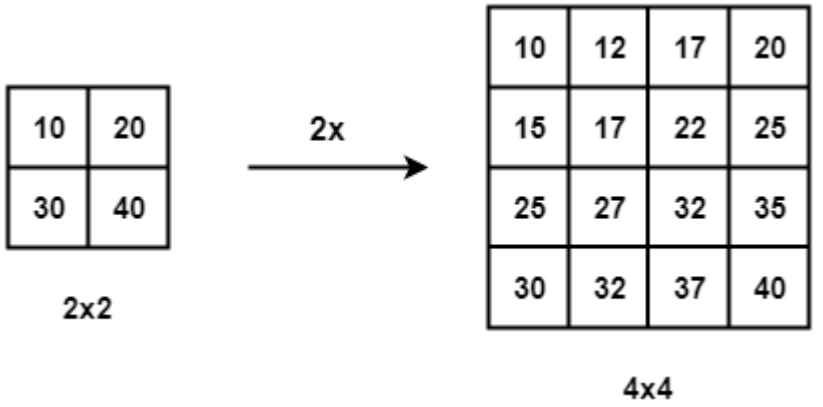
Nearest Neighbor



Increasing Output Scale in Networks

Upsampling Techniques

Bilinear Interpolation



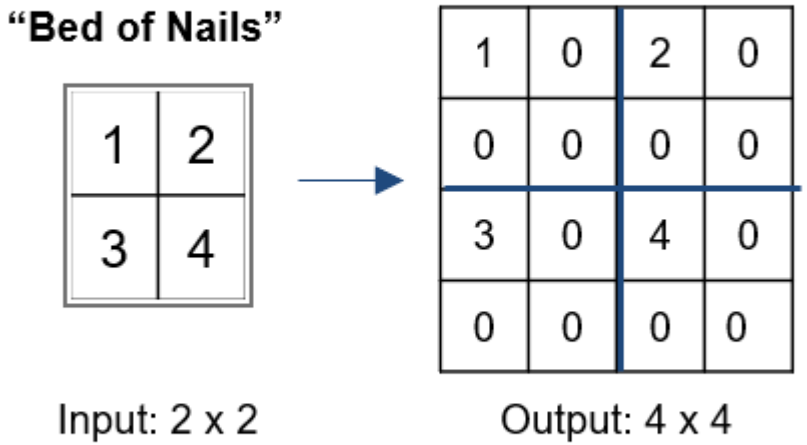
2x2

4x4

Increasing Output Scale in Networks

Upsampling Techniques

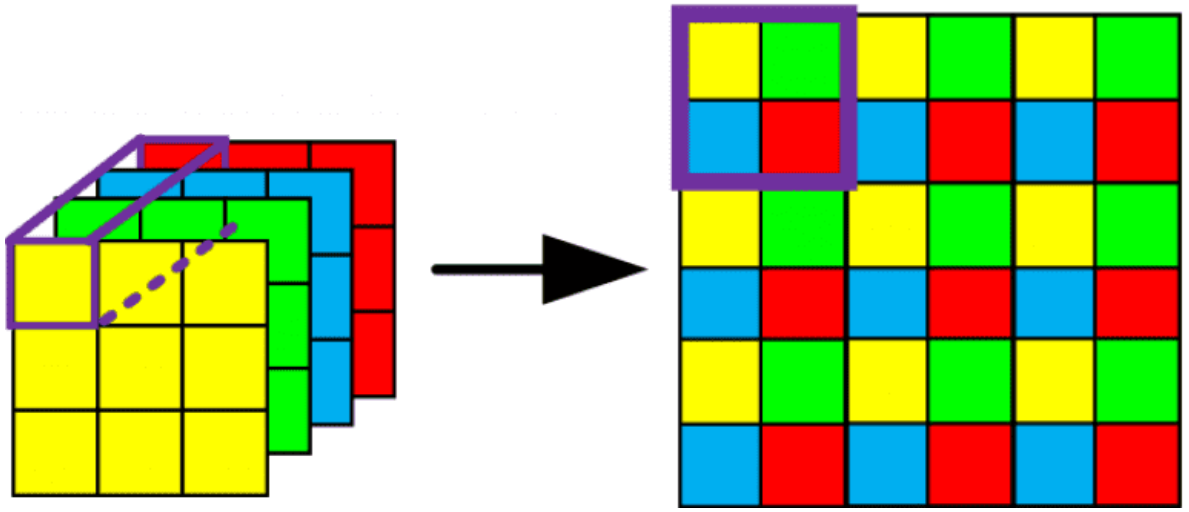
Bed of Nails



Increasing Output Scale in Networks

Upsampling Techniques

Pixel Shuffle



https://pyimagesearch.com/2021/09/27/pixel-shuffle-super-resolution-with-tensorflow-keras-and-deep-learning/

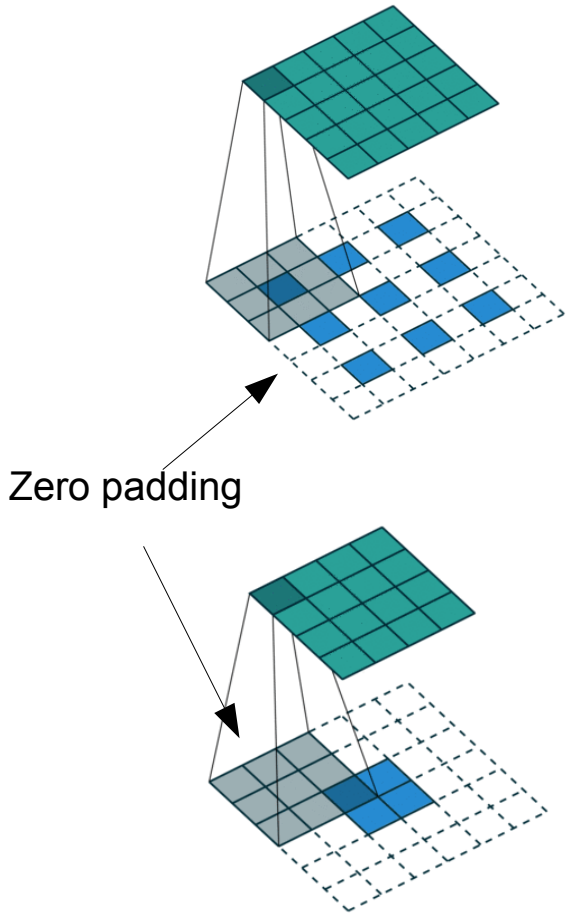
Increasing Output Scale in Networks

Up-Convolution

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

→ “Transposed Convolution”

Transposed convolutions



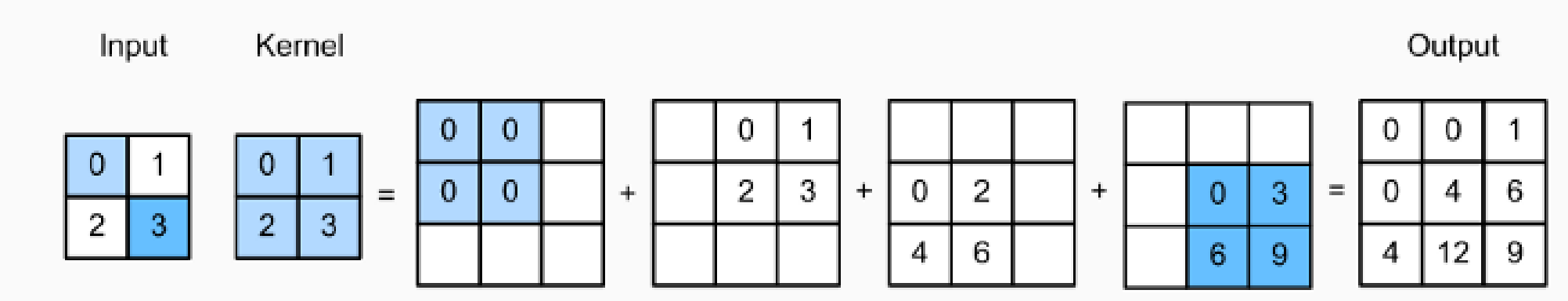
Animations: https://github.com/vdumoulin/conv_arithmetic

Increasing Output Scale in Networks

Up-Convolution

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

→ "Transposed Convolution"



Increasing Output Scale in Networks

Other Applications of Transposed Convolutions

Super-Resolution

Ground Truth

¼ Sized Input

Bicubic

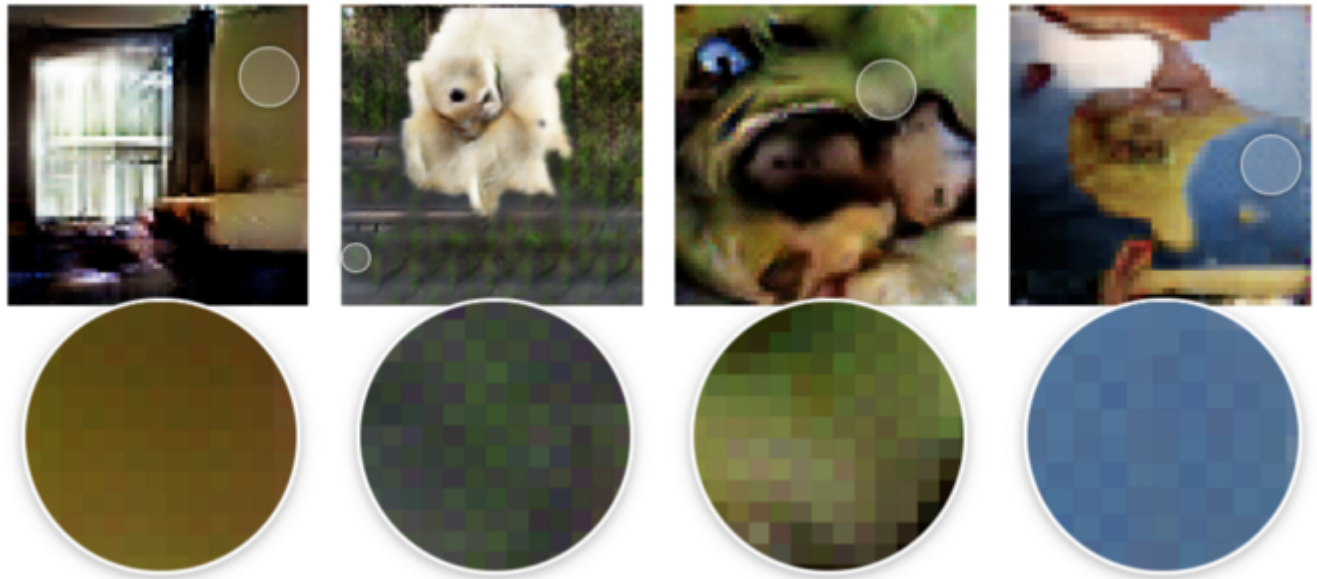
Super Resolution Network

SRFeat: Single Image Super-Resolution with Feature Discrimination, Seong-Jin Park, Hyeongseok Son, Sunghyun Cho, Ki-Sang Hong, Seungyong Lee; Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 439-455

Increasing Output Scale in Networks

Problems with Transpose Convolutions

Checkerboard patterns



http://distill.pub/2016/deconv-checkerboard

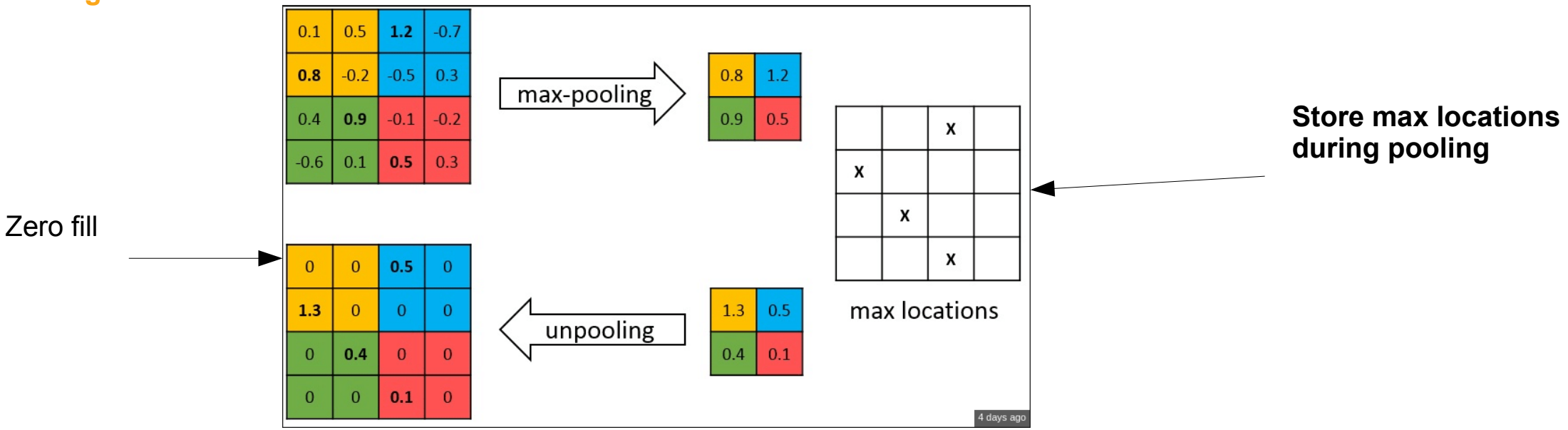
Increasing Output Scale in Networks

Up-Convolution

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

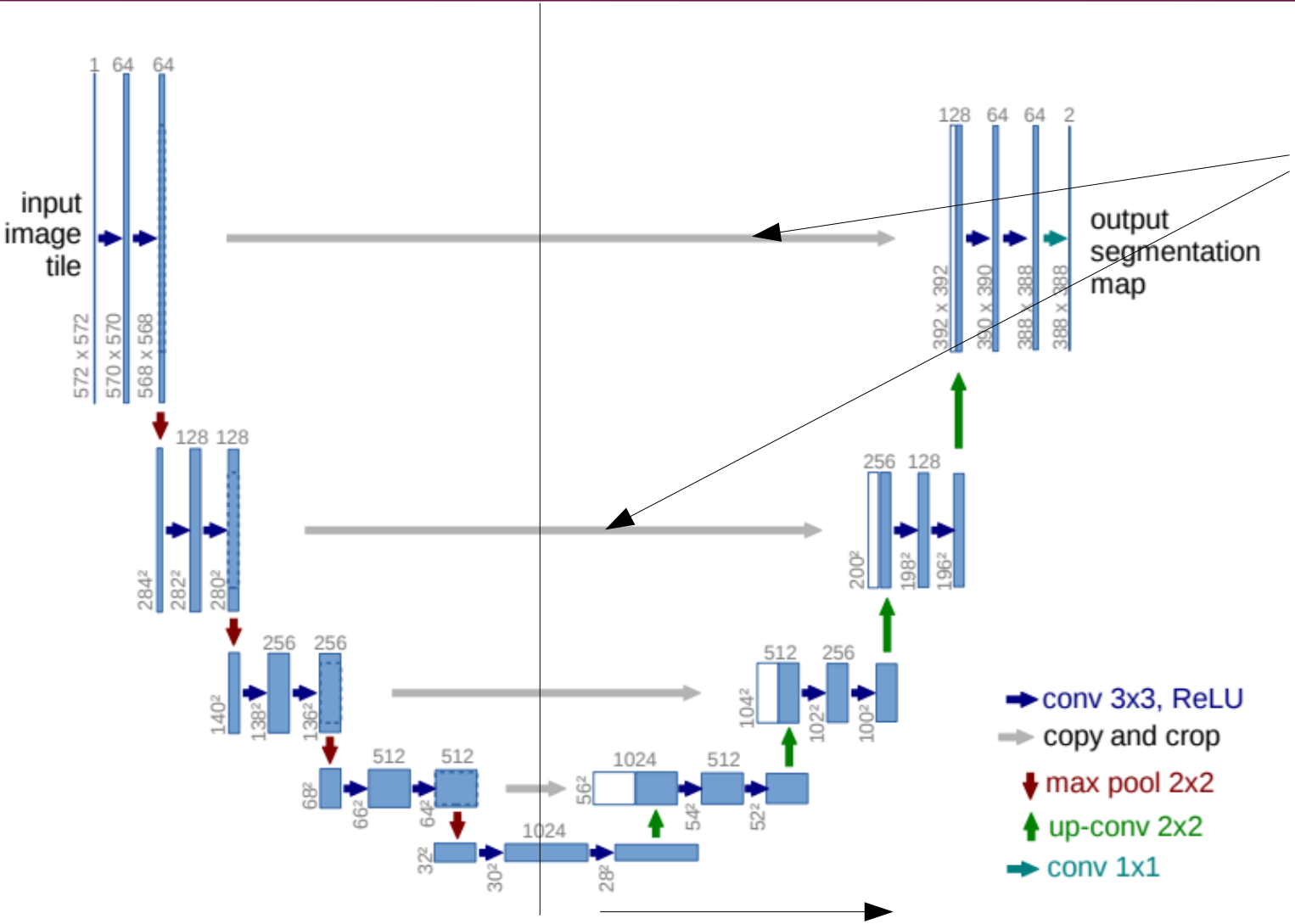
→ "Transposed Convolution"

Un-Pooling



Animations: https://github.com/vdumoulin/conv_arithmetic

Increasing Output Scale in Networks



Important:

“Residual” connections

Deliver high resolution information for generator

U-Net Application: Semantic Segmentation

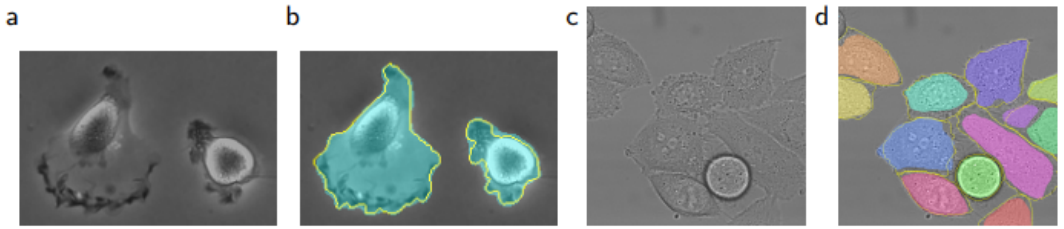


Fig. 4. Result on the ISBI cell tracking challenge. (**a**) part of an input image of the "PhC-U373" data set. (**b**) Segmentation result (cyan mask) with manual ground truth (yellow border) (**c**) input image of the "DIC-HeLa" data set. (**d**) Segmentation result (random colored masks) with manual ground truth (yellow border).

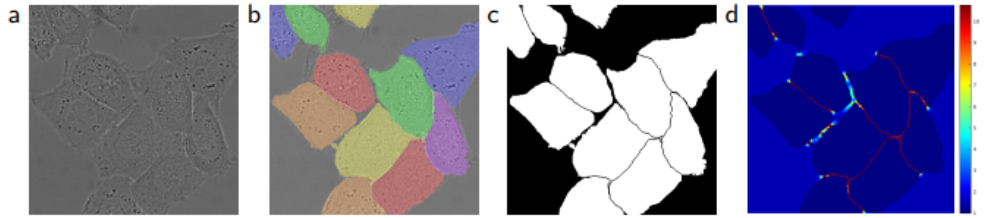
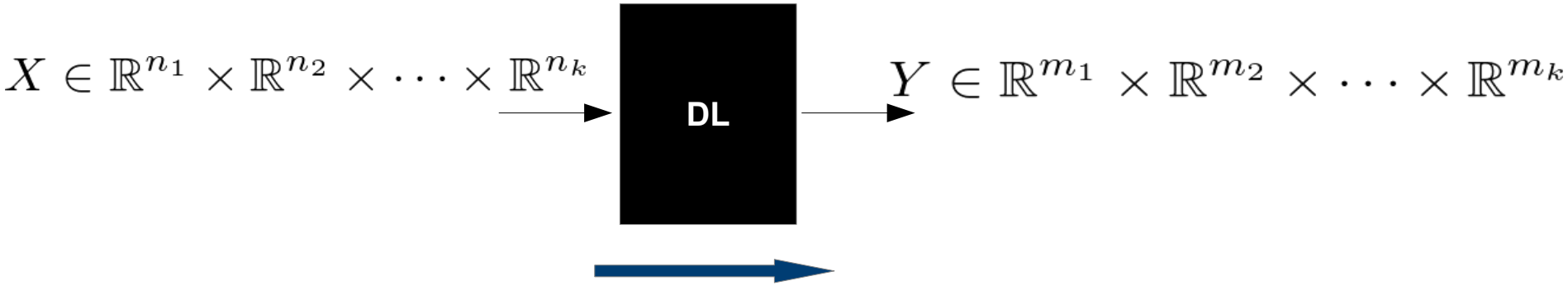


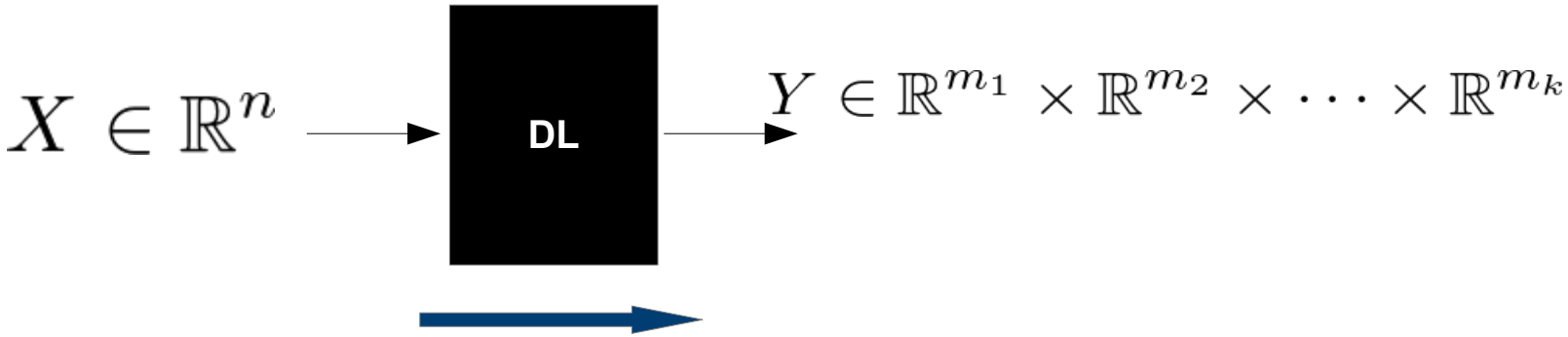
Fig. 3. HeLa cells on glass recorded with DIC (differential interference contrast) microscopy. (**a**) raw image. (**b**) overlay with ground truth segmentation. Different colors indicate different instances of the HeLa cells. (**c**) generated segmentation mask (white: foreground, black: background). (**d**) map with a pixel-wise loss weight to force the network to learn the border pixels.

Motivation

Let's see



Now



Generator Networks

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

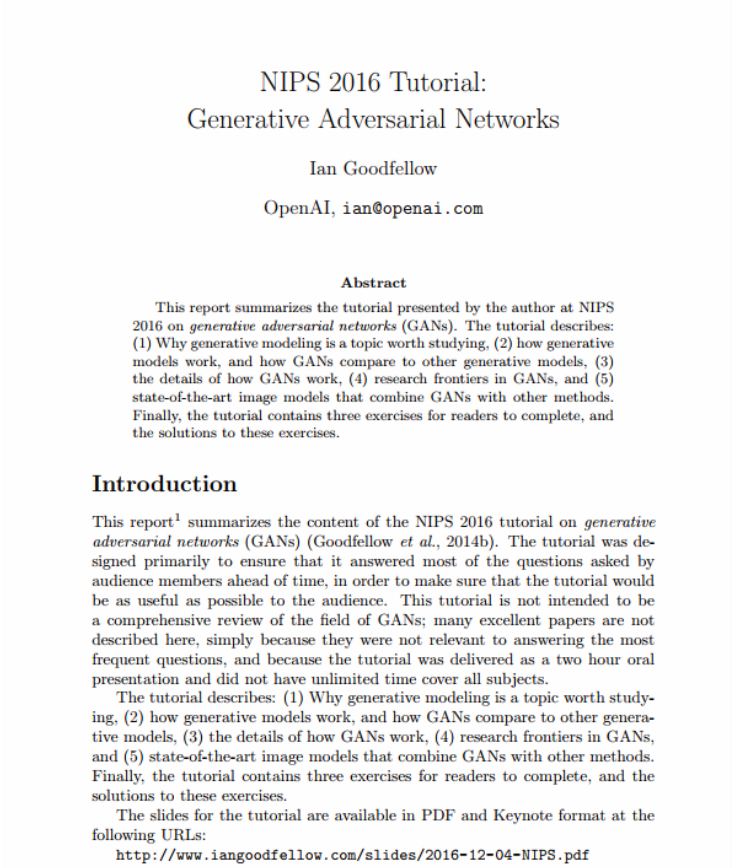
We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than 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

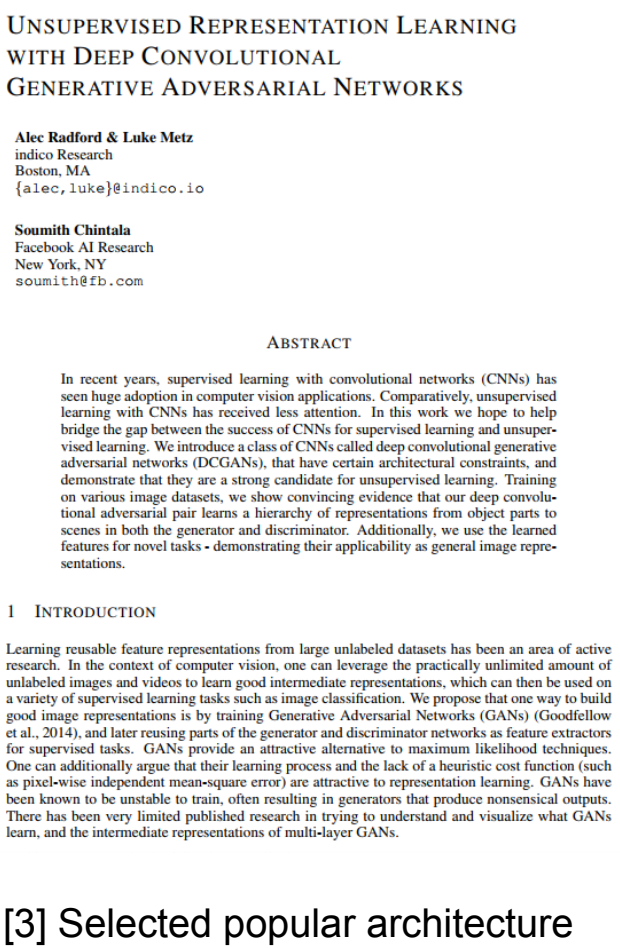
This report[1] summarizes the content of the NIPS 2016 tutorial on *generative adversarial networks* (GANs) (Goodfellow *et al.*, 2014b). The tutorial was designed primarily to ensure that it answered most of the questions asked by audience members ahead of time, in order to make sure that the tutorial 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

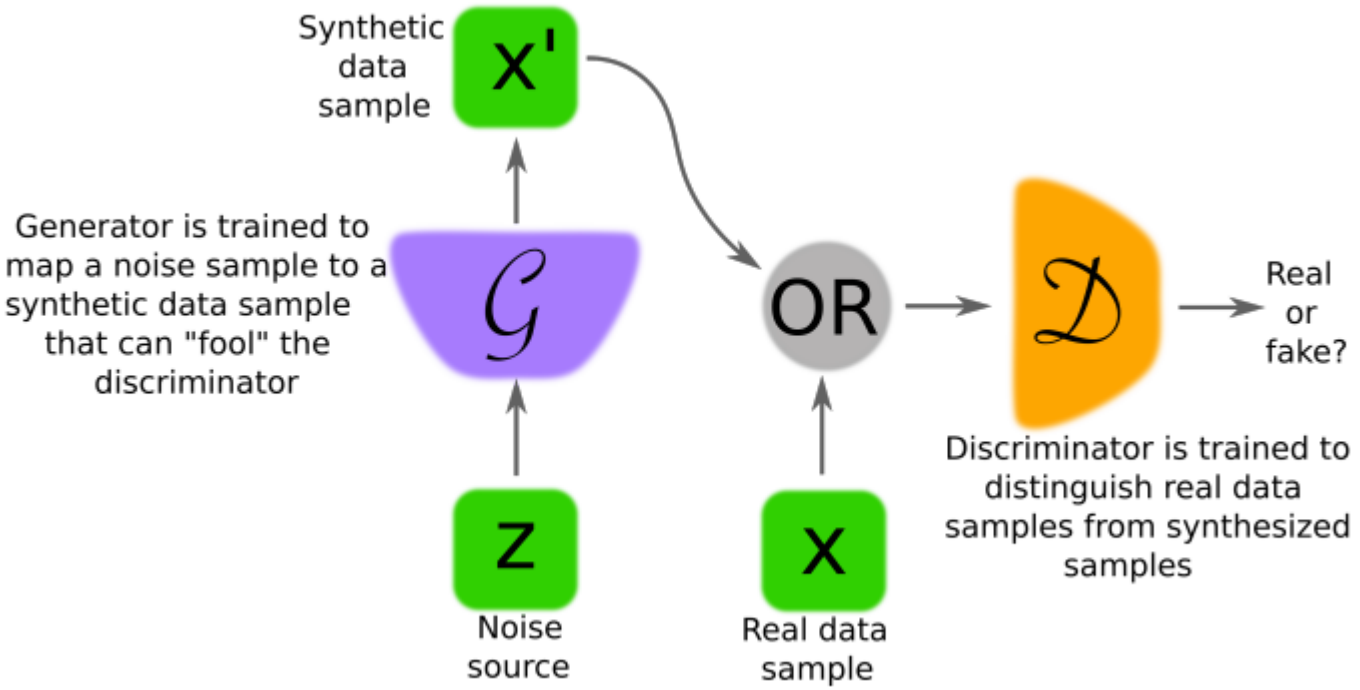
Learning reusable feature representations from large unlabeled datasets has been an area of active research. In the context of computer vision, one can leverage the practically unlimited amount of unlabeled images and videos to learn good intermediate representations, which can then be used on a variety 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

What is a GAN ?

- In a Nutshell -
Generative Adversarial Nets are:

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



What is a GAN ?

- In a Nutshell -
Generative Adversarial Nets are:

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

Generative Adversarial Networks: An Overview

Antonia Creswell[§], Tom White[¶],
Vincent Dumoulin[‡], Kai Arulkumaran[§], Biswa Sengupta[†§] and Anil A Bharath[§], *Member IEEE*
[§] BICV Group, Dept. of Bioengineering, Imperial College London
[¶] School of Design, Victoria University of Wellington, New Zealand
[‡] MILA, University of Montreal, Montreal H3T 1N8
[†] Cortexica Vision Systems Ltd., London, United Kingdom

***Abstract*—Generative adversarial networks (GANs) provide a way to learn deep representations without extensively annotated training data. They achieve this through deriving backpropagation signals through a competitive process involving a pair of networks. The representations that can be learned by GANs may be used in a variety of applications, including image synthesis, semantic image editing, style transfer, image super-resolution and classification. The aim of this review paper is to provide an overview of GANs for the signal processing community, drawing on familiar analogies and concepts where possible. In addition to identifying different methods for training and constructing GANs, we also point to remaining challenges in their theory and application.**

***Index Terms*—neural networks, unsupervised learning, semi-supervised learning.**

I. Introduction

GENERATIVE adversarial networks (GANs) are an emerging technique for both semi-supervised and unsupervised learning. They achieve this through implicitly modelling high-dimensional distributions of data. Proposed in 2014 [1], they can be characterized by training a pair of networks in competition with each other. A common analogy, apt for visual data, is to think of one network as an art forger, and the other as an art expert. The forger, known in the GAN literature as the generator, $\mathcal{G}$, creates forgeries, with the aim of making realistic images. The expert, known as the discriminator, $\mathcal{D}$, receives both forgeries and real (authentic) images, and aims to tell them apart (see Fig. 1). Both are trained simultaneously, and in competition with each other.

Crucially, the generator has no direct access to real images - the only way it learns is through its interaction with the discriminator. The discriminator has access to both the synthetic samples and samples drawn from the stack of real images. The error signal to the discriminator is provided through the simple ground truth of knowing whether the image came from the real stack or from the generator. The same error signal, via the discriminator, can be used to train the generator, leading it towards being able to produce forgeries of better quality.

The networks that represent the generator and discriminator are typically implemented by multi-layer networks consisting of convolutional and/or fully-connected layers. The generator and discriminator networks must be differentiable, though it is not necessary for them to be directly invertible. If one considers the generator network as mapping from some representation space, called a latent space, to the space of the data (we shall focus on images), then we may express this more formally as $\mathcal{G} : \mathcal{G}(\mathbf{z}) \rightarrow R^{|\mathbf{x}|}$, where $\mathbf{z} \in R^{|\mathbf{z}|}$ is a sample from the latent space, $\mathbf{x} \in R^{|\mathbf{x}|}$ is an image and $|\cdot|$ denotes the number of dimensions.

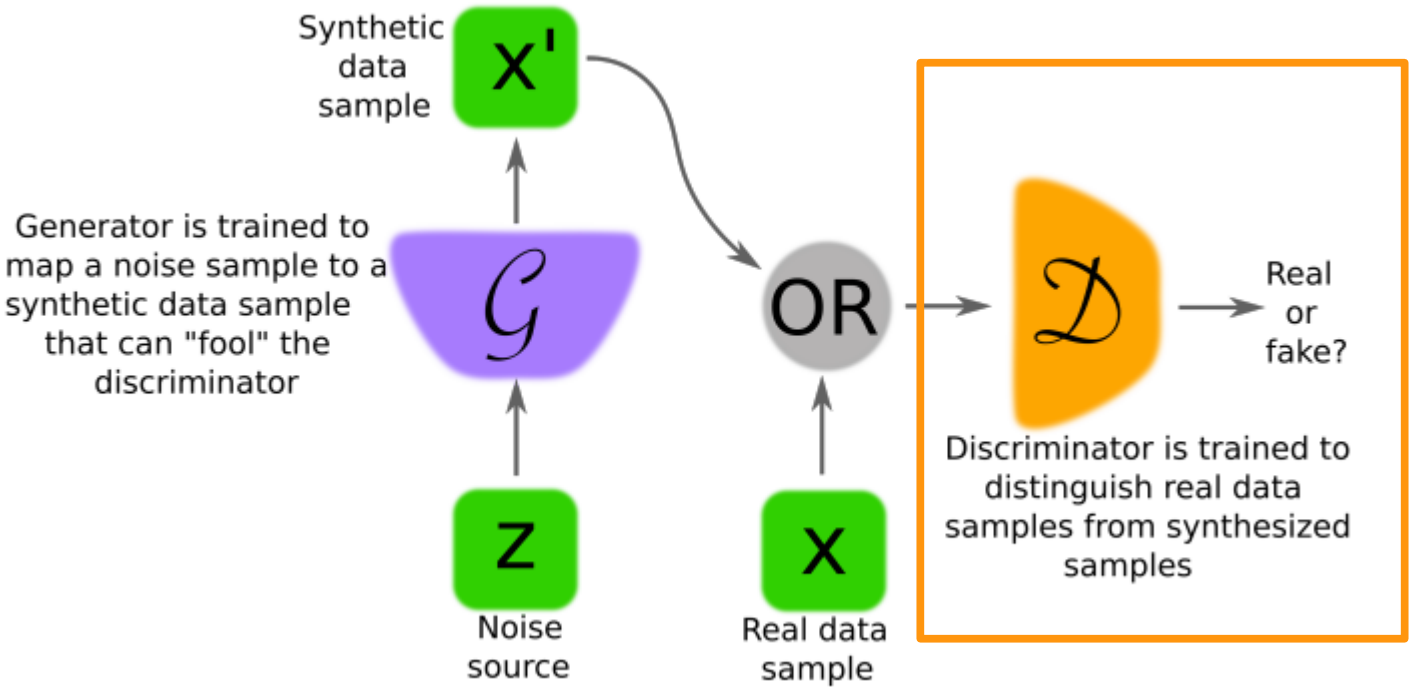
In a basic GAN, the discriminator network, $\mathcal{D}$, may be similarly characterized as a function that maps from image data to a probability that the image is from the real data distribution, rather than the generator distribution: $\mathcal{D} : \mathcal{D}(\mathbf{x}) \rightarrow (0, 1)$. For a fixed generator, $\mathcal{G}$, the discriminator, $\mathcal{D}$, may be trained to classify images as either being from the training data (real, close to 1) or from a fixed generator (fake, close to 0). When the discriminator is optimal, it may be frozen, and the generator, $\mathcal{G}$, may continue to be trained so as to lower the accuracy of the discriminator. If the generator distribution is able to match the real data distribution perfectly then the discriminator will be maximally confused, predicting 0.5 for all inputs. In practice, the discriminator might not be trained until it is optimal; we explore the training process in more depth in Section IV.

On top of the interesting academic problems related to training and constructing GANs, the motivations behind training GANs may not necessarily be the generator or the discriminator *per se*: the representations embodied by either of the pair of networks can be used in a variety of subsequent tasks. We explore the applications of these representations in Section VI.

What is a GAN ?

- In a Nutshell -
Generative Adversarial Nets are:

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

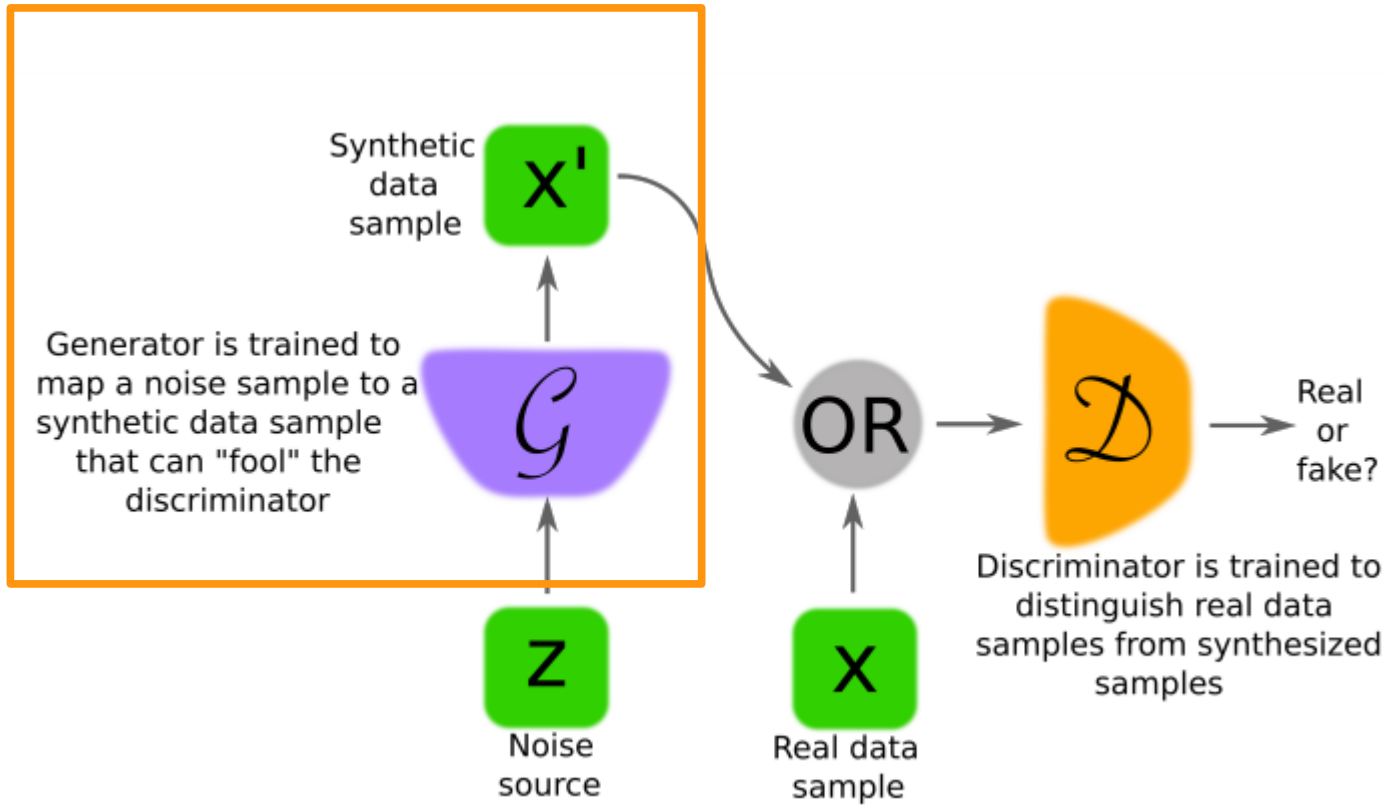


GAN schemes from: https://arxiv.org/pdf/1710.07035.pdf

What is a GAN ?

- In a Nutshell -
Generative Adversarial Nets are:

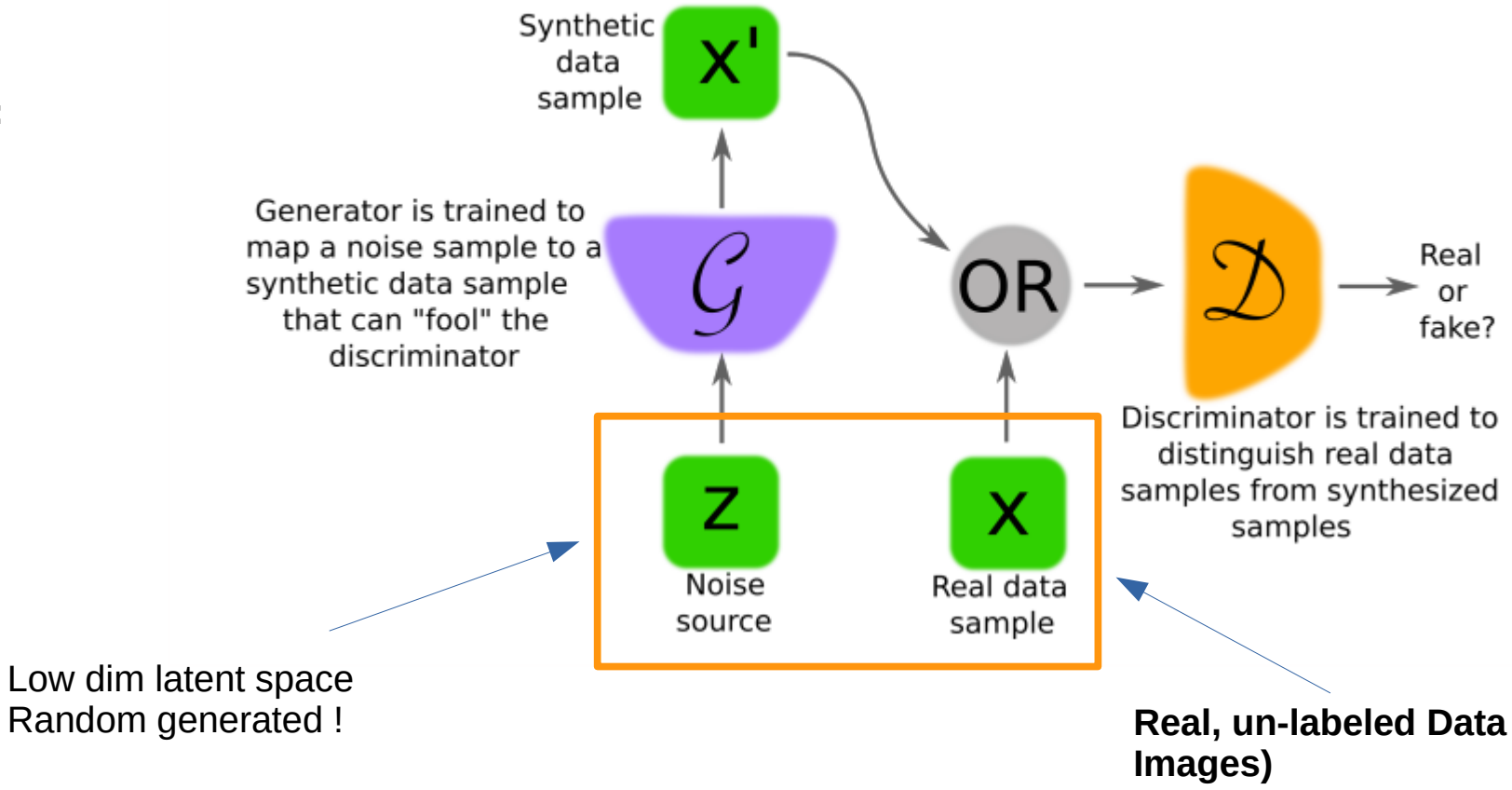
- Groups of DNNs (at least two)
- Working against each other !
- Min parts:
  - Discriminator Network
  - Generator Network



What is a GAN ?

- In a Nutshell -
Generative Adversarial Nets are:

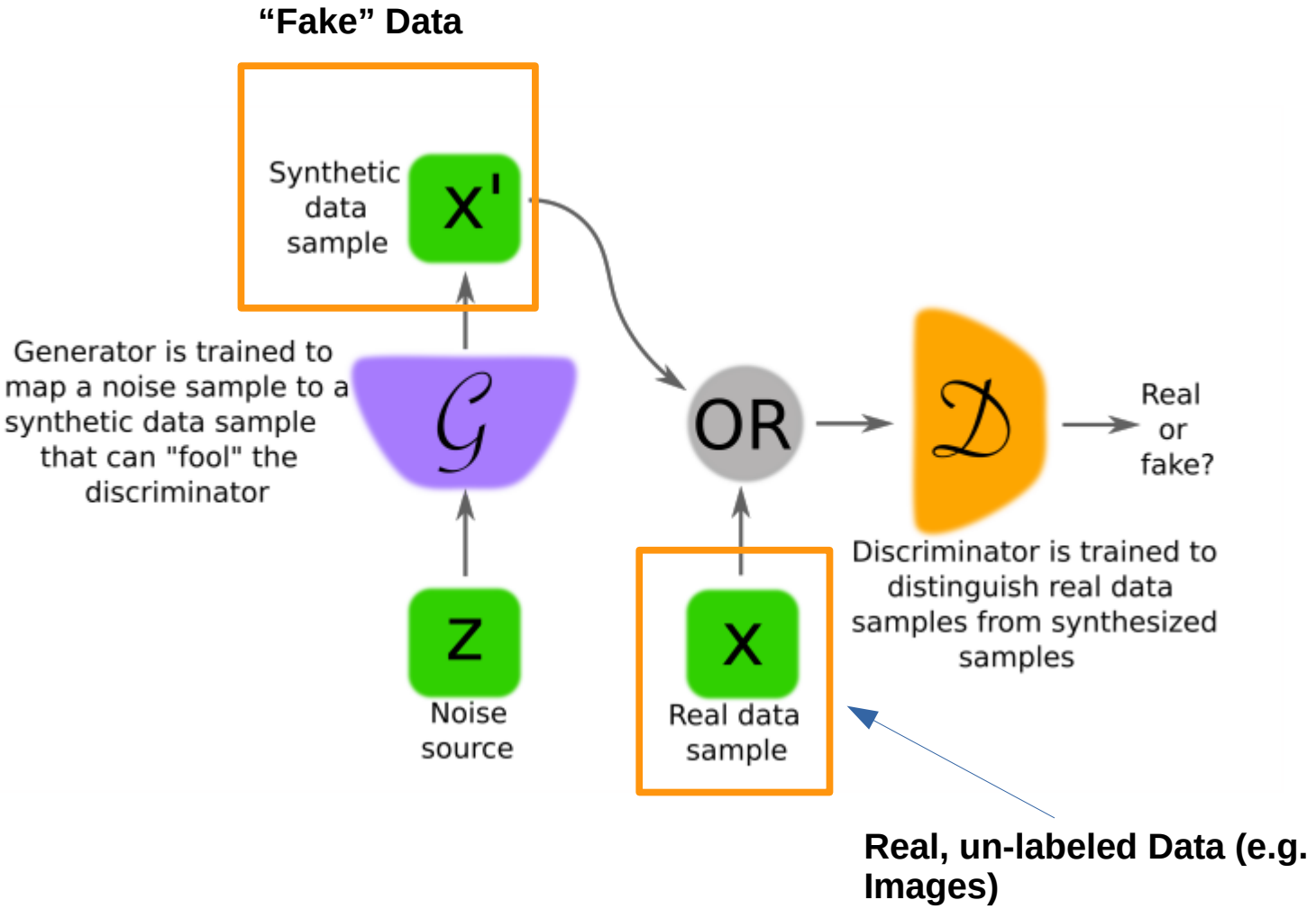
- Groups of DNNs (at least two)
- Working against each other !
- Min parts:
  - Discriminator Network
  - Generator Network



What is a GAN ?

- In a Nutshell -
Generative Adversarial Nets are:

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



Example generator Architecture [3]

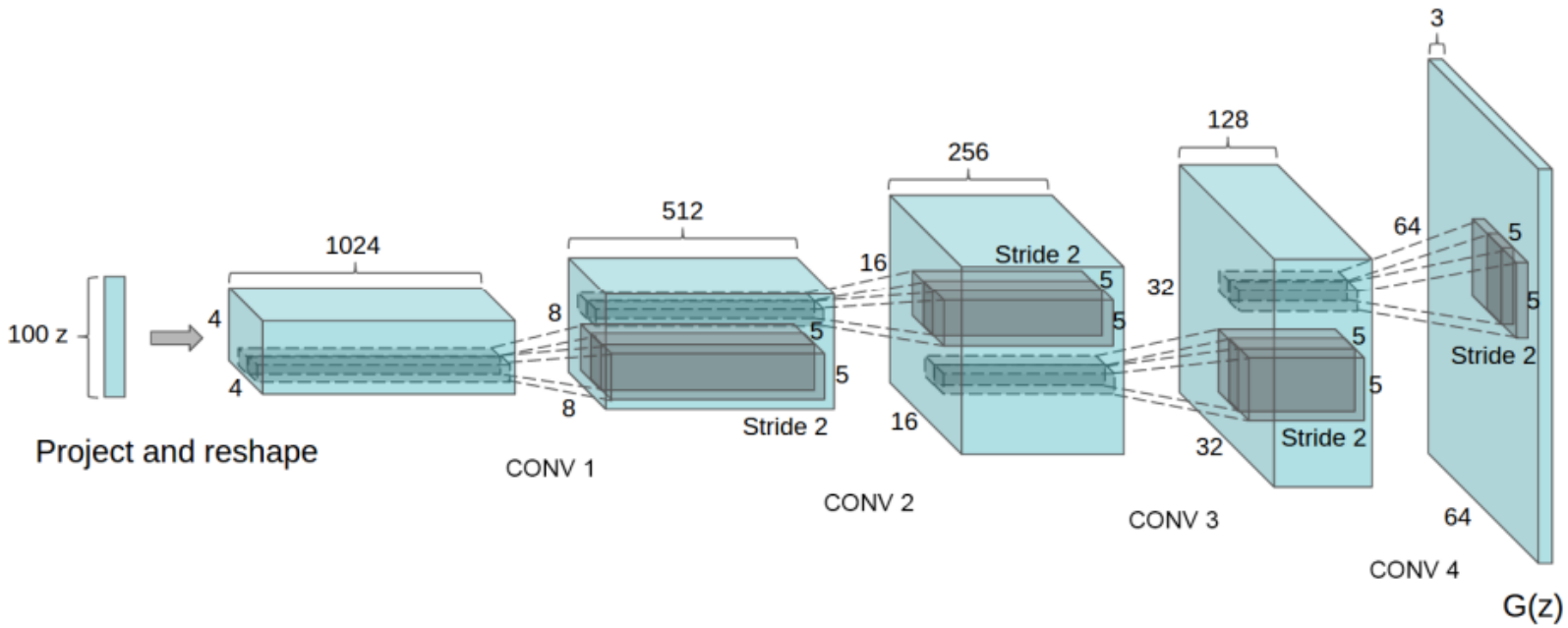
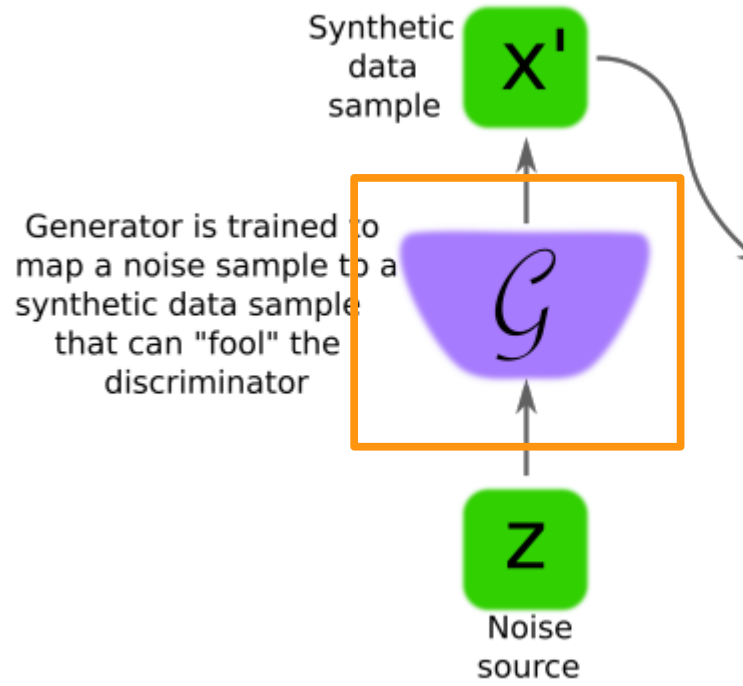


Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions)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



Generated Data (Examples)

Images of Bedrooms generated by a GAN [3]

Generated Data (Examples)

Images of Faces generated by a GAN [3]

Generated Data (Examples)

Beware of Checkerboard Patterns from Up-Convolution:
(Generated Image with amplified Contrast)

Generated Data (Examples)

Beware of Checkerboard Patterns from Up-Convolution:
(Generated Image with amplified Contrast)

Remember from earlier?

StyleGAN: state of the art DeepFakes

A Style-Based Generator Architecture for Generative Adversarial Networks

Tero Karras
NVIDIA
tkarras@nvidia.com

Samuli Laine
NVIDIA
slaine@nvidia.com

Timo Aila
NVIDIA
taila@nvidia.com

Abstract

We propose an alternative generator architecture for generative adversarial networks, borrowing from style transfer literature. The new architecture leads to an automatically learned, unsupervised separation of high-level attributes (e.g., pose and identity when trained on human faces) and stochastic variation in the generated images (e.g., freckles, hair), and it enables intuitive, scale-specific control of the synthesis. The new generator improves the state-of-the-art in terms of traditional distribution quality metrics, leads to demonstrably better interpolation properties, and also better disentangles the latent factors of variation. To quantify interpolation quality and disentanglement, we propose two new, automated methods that are applicable to any generator architecture. Finally, we introduce a new, highly varied and high-quality dataset of human faces.

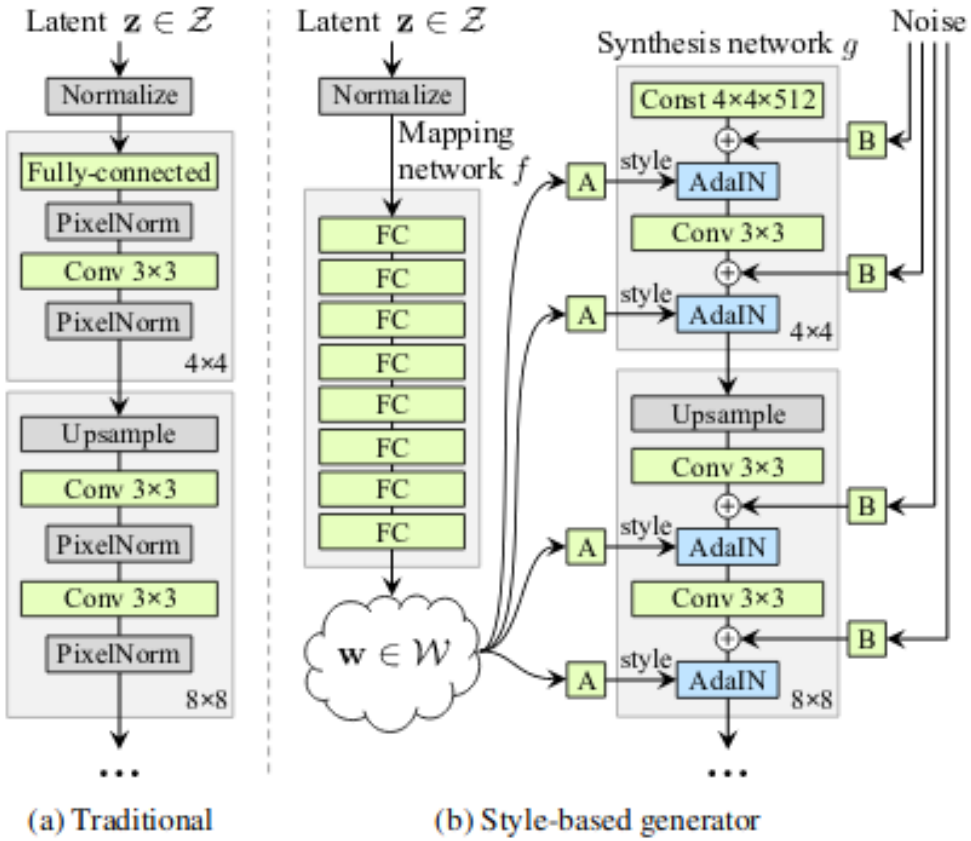
1. Introduction

The resolution and quality of images produced by generative methods — especially generative adversarial networks (GAN) [22] — have seen rapid improvement recently [30, 45, 5]. Yet the generators continue to operate as black

(e.g., pose, identity) from stochastic variation (e.g., freckles, hair) in the generated images, and enables intuitive scale-specific mixing and interpolation operations. We do not modify the discriminator or the loss function in any way, and our work is thus orthogonal to the ongoing discussion about GAN loss functions, regularization, and hyperparameters [24, 45, 5, 40, 44, 36].

Our generator embeds the input latent code into an intermediate latent space, which has a profound effect on how the factors of variation are represented in the network. The input latent space must follow the probability density of the training data, and we argue that this leads to some degree of unavoidable entanglement. Our intermediate latent space is free from that restriction and is therefore allowed to be disentangled. As previous methods for estimating the degree of latent space disentanglement are not directly applicable in our case, we propose two new automated metrics — perceptual path length and linear separability — for quantifying these aspects of the generator. Using these metrics, we show that compared to a traditional generator architecture, our generator admits a more linear, less entangled representation of different factors of variation.

Finally, we present a new dataset of human faces (Flickr-Faces-HQ, FFHQ) that offers much higher quality and covers considerably wider variation than existing high-resolution datasets (Appendix A). We have made this



(a) Traditional (b) Style-based generator

Style

A Style-Bas

We propose an
generative adversar
transfer literature.
tomatically learned,
attributes (e.g., pose
faces) and stochasti
(e.g., freckles, hair),
control of the synthe
state-of-the-art in te
metrics, leads to den
ties, and also better
tion. To quantify inte
we propose two new,
ble to any generator
new, highly varied an

1. Introduction

The resolution an
erative methods — e
works (GAN) [22] —
[30, 45, 5]. Yet the g

https://neurohive.io/en/news/nvidia-released-state-of-the-art-stylegan2-model/

Data Augmentation – Count Conditioned GANs

UNIVERSITÄT SIEGEN

2021 ICCV OCTOBER 11-17 VIRTUAL

BOSCH

- Our aim is to guide the image generation process based on the number of instances of each object class
- Requires no additional information such as bounding box annotations
- Learning to count is challenging for computer vision and deep learning models
- Contributions
  - Modular extension to StyleGAN2[1] for high quality image generation based on the **M**ultiple **C**lass object **C**ount - MC² - StyleGAN2
  - A new and challenging count based real-world dataset derived from the street scenes dataset Cityscapes[2] - CityCount

Figure 1: Real and generated CityCount images by our model based on the multiple-class count input

MC²-StyleGAN2

- Generator (G)
  - Extension to StyleGAN2 generator with dense like skip connections
  - Input comprises of Multiple-Class Count vector (**c**) and latent noise (Z)
  - **c** concatenated to every layer of mapping network
- Discriminator (D)
  - An adversarial pathway – To distinguish real/fake image
  - A count regression pathway (C) – To predict object class and their multiplicity

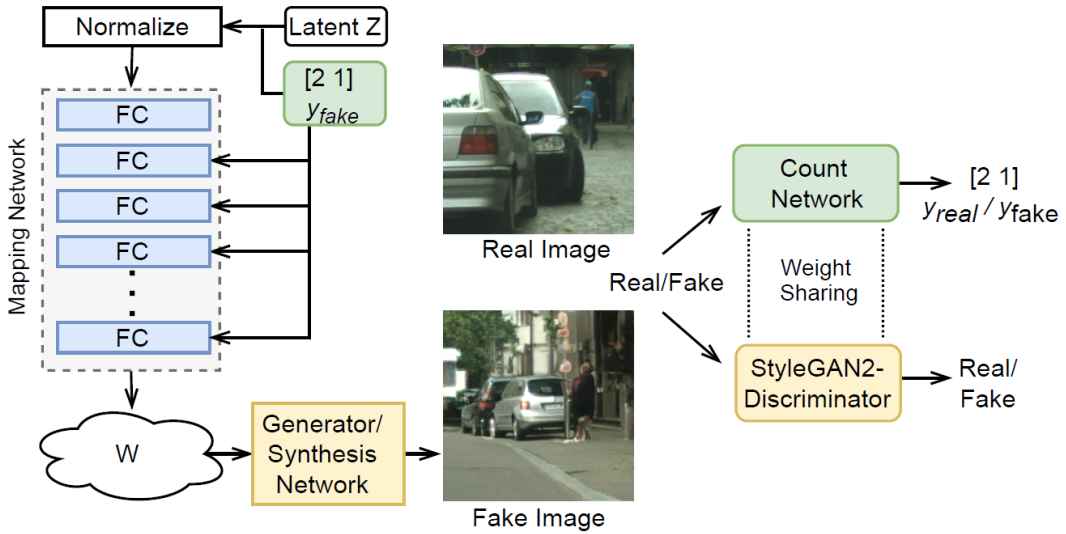


Figure 1: MC²-StyleGAN2 architecture – The input count vector [2,1] corresponds to 2 cars and 1 person

$$\mathcal{L}_{MC^2}(C) = ||C(x) - \mathbf{c}||_2$$

$$\mathcal{L}_{MC^2-StyleGAN2}(G, D) = \mathcal{L}_{GAN}(G, D) + \lambda \mathcal{L}_{MC^2}(C)$$

Qualitative Evaluation

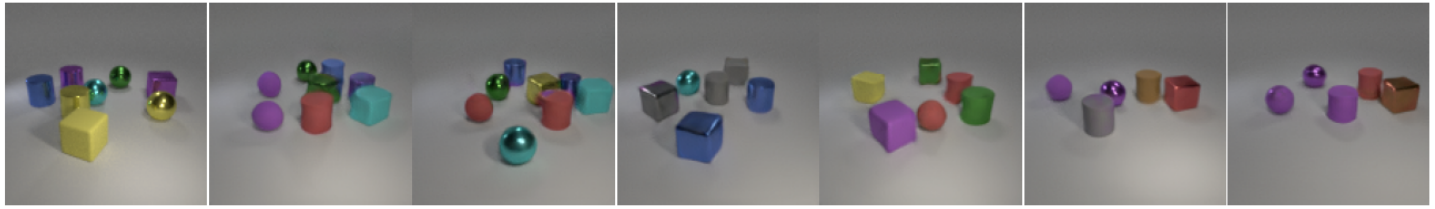
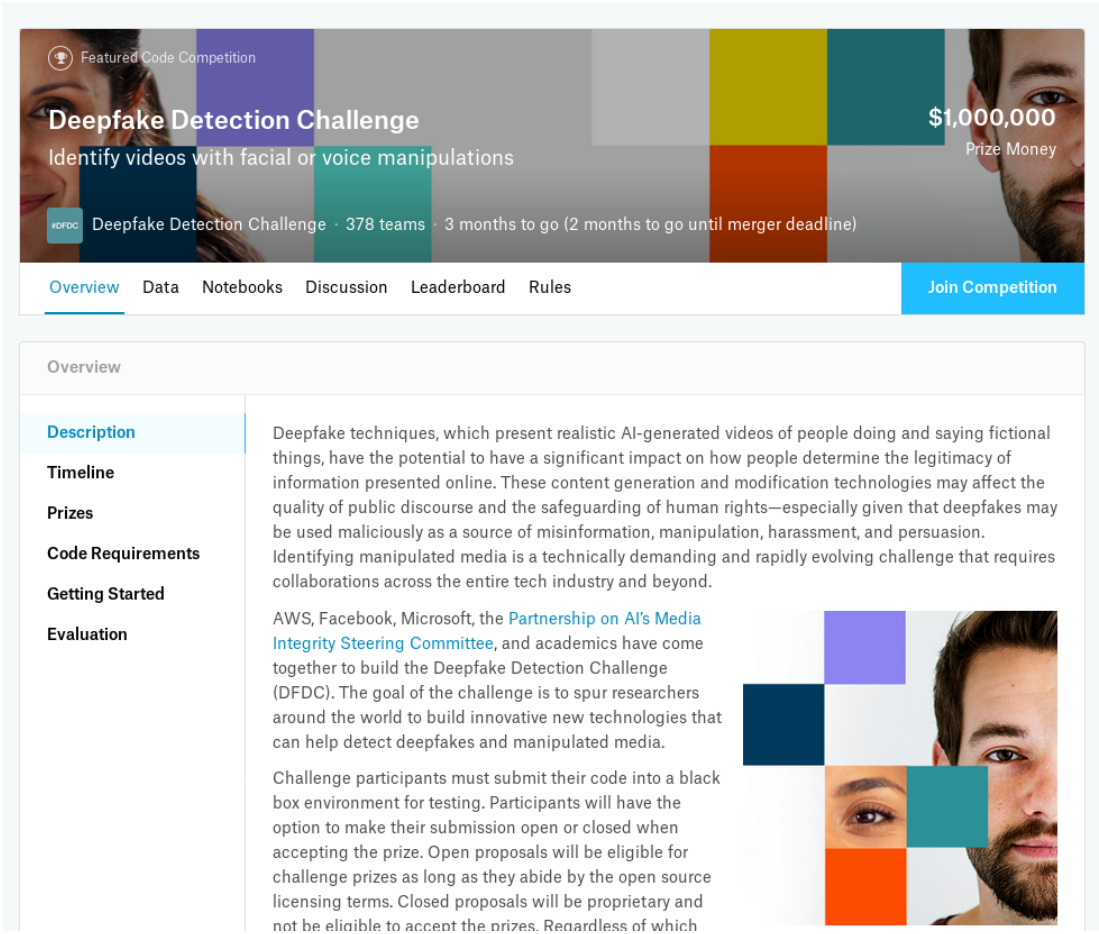


Figure 1: CLEVR - count vector corresponds to number of cylinders, spheres and cubes

Figure 2: SVHN - count vector corresponds to number of digits 1,2,...9

Figure 3: CityCount - count vector corresponds to number of cars and persons

Detecting DeepFakes



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

Detecting DeepFakes

Common approaches:

- Use Deep Neural Networks (CNNs) to classify fake content
  - Where does the training data come from?
  - Only known manipulations

Attributing Fake Images to GANs: Learning and Analyzing GAN Fingerprints

Ning Yu[1,2] Larry Davis[1] Mario Fritz[3]
[1]University of Maryland, College Park
[2]Max Planck Institute for Informatics
Saarland Informatics Campus, Germany
[3]CISPA Helmholtz Center for Information Security
Saarland Informatics Campus, Germany
ningyu@mpi-inf.mpg.de lsd@cs.umd.edu fritz@cispa.saarland

Abstract

Recent advances in Generative Adversarial Networks (GANs) have shown increasing success in generating photorealistic images. But they also raise challenges to visual forensics and model attribution. We present the first study of learning GAN fingerprints towards image attribution and using them to classify an image as real or GAN-generated. For GAN-generated images, we further identify their sources. Our experiments show that (1) GANs carry distinct model fingerprints and leave stable fingerprints in their generated images, which support image attribution; (2) even minor differences in GAN training can result in different fingerprints, which enables fine-grained model authentication; (3) fingerprints persist across different image frequencies and patches and are not biased by GAN artifacts; (4) fingerprint finetuning is effective in immunizing against five types of adversarial image perturbations; and (5) comparisons also show our learned fingerprints consistently outperform several baselines in a variety of setups [1].

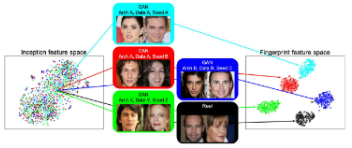


Figure 1. A t-SNE [43] visual comparison between our fingerprint features (right) and the baseline inception features [52] (left) for image attribution. Inception features are highly entangled, indicating the challenge to differentiate high-quality GAN-generated images from real ones. However, our result shows any single difference in GAN architectures, training sets, or even initialization seeds can result in distinct fingerprint features for effective attribution.

At the same time, however, the success of GANs has raised two challenges to the vision community: visual forensics and intellectual property protection.

GAN challenges to visual forensics. There is a widespread concern about the impact of this technology when used maliciously. This issue has also received in-

Detecting DeepFakes

- Two-player game, similar to Generator and Discriminator in GANs
- Scientific and societal interest in generating ever better and more realistic content (VR, games, illustrations, training better models, etc.)
- Scientific and societal interest in being able to verify content authenticity

Detecting DeepFakes

Common approaches:

- Use Deep Neural Networks (CNNs) to classify fake content
  - Where does the training data come from?
  - Only known manipulations

Attributing Fake Images to GANs: Learning and Analyzing GAN Fingerprints

Ning Yu[1,2] Larry Davis[1] Mario Fritz[3]
[1]University of Maryland, College Park
[2]Max Planck Institute for Informatics
Saarland Informatics Campus, Germany
[3]CISPA Helmholtz Center for Information Security
Saarland Informatics Campus, Germany
ningyu@mpi-inf.mpg.de lsd@cs.umd.edu fritz@cispa.saarland

Abstract

Recent advances in Generative Adversarial Networks (GANs) have shown increasing success in generating photorealistic images. But they also raise challenges to visual forensics and model attribution. We present the first study of learning GAN fingerprints towards image attribution and using them to classify an image as real or GAN-generated. For GAN-generated images, we further identify their sources. Our experiments show that (1) GANs carry distinct model fingerprints and leave stable fingerprints in their generated images, which support image attribution; (2) even minor differences in GAN training can result in different fingerprints, which enables fine-grained model authentication; (3) fingerprints persist across different image frequencies and patches and are not biased by GAN artifacts; (4) fingerprint finetuning is effective in immunizing against five types of adversarial image perturbations; and (5) comparisons also show our learned fingerprints consistently outperform several baselines in a variety of setups [1].

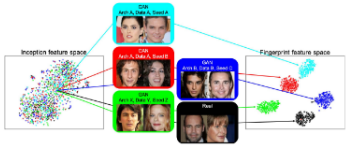


Figure 1. A t-SNE [43] visual comparison between our fingerprint features (right) and the baseline inception features [52] (left) for image attribution. Inception features are highly entangled, indicating the challenge to differentiate high-quality GAN-generated images from real ones. However, our result shows any single difference in GAN architectures, training sets, or even initialization seeds can result in distinct fingerprint features for effective attribution.

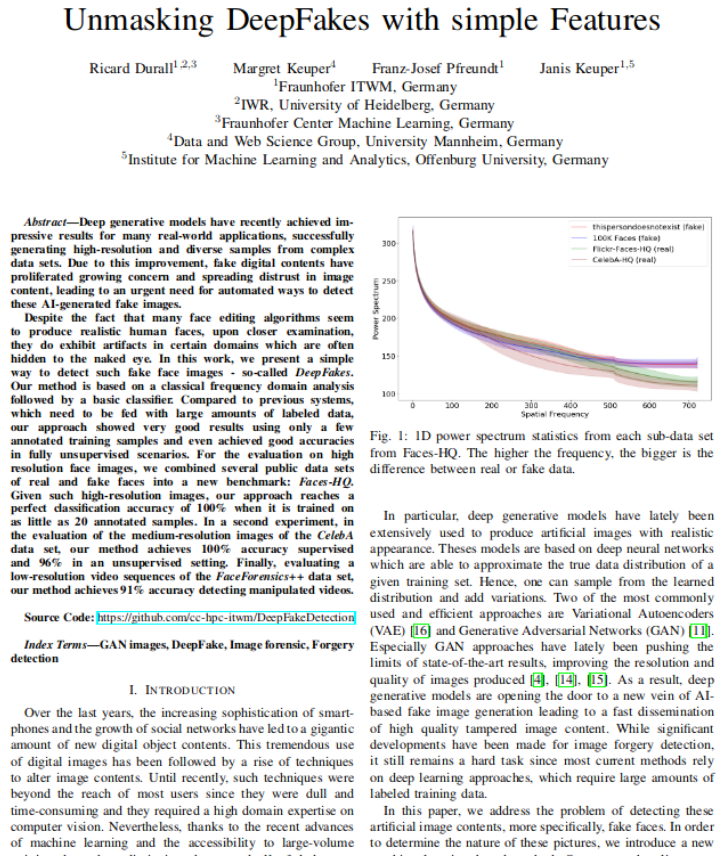
At the same time, however, the success of GANs has raised two challenges to the vision community: visual forensics and intellectual property protection.

GAN challenges to visual forensics. There is a widespread concern about the impact of this technology when used maliciously. This issue has also received in-

Detecting DeepFakes

Common approaches:

- Use Deep Neural Networks (CNNs) to classify fake content
  - Where does the training data come from?
  - Only known manipulations
- Detecting Systematic Errors in generated Content



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

Idea: Use Simple Features

Detect Checkerboard Artifacts?

First:

- Look into the Fourier domain to understand their origin
- For simplicity:
  - example of Bed of Nails upsampling
  - 1D

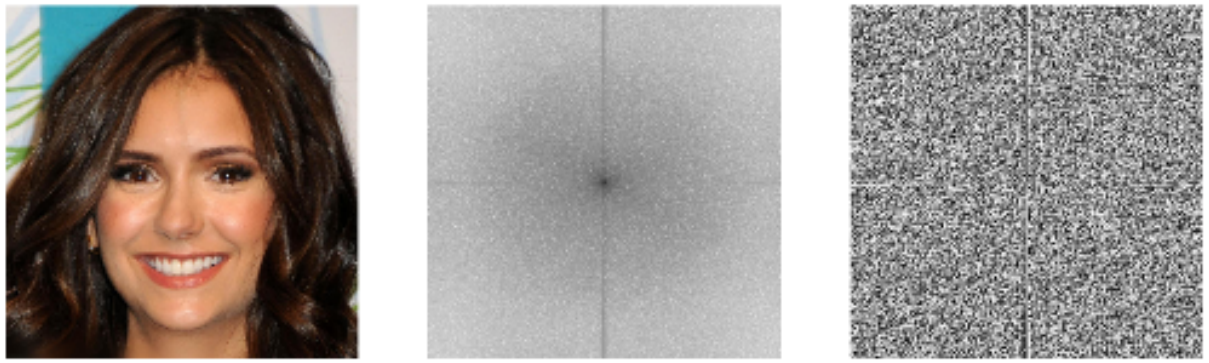


Fig. 3: Example of a DFT applied to a sample. (Left) Input image [1]. (Center) Power Spectrum. (Right) Phase Spectrum.

Fourier Transform of Image I

$$\mathcal{F}(I)(k,\ell) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} e^{-2\pi i \cdot \frac{jk}{M}} e^{-2\pi i \cdot \frac{j\ell}{N}} \cdot I(m,n), \quad (1)$$

$$\text{for} \quad k = 0,\ldots,M-1, \quad \ell = 0,\ldots,N-1,$$

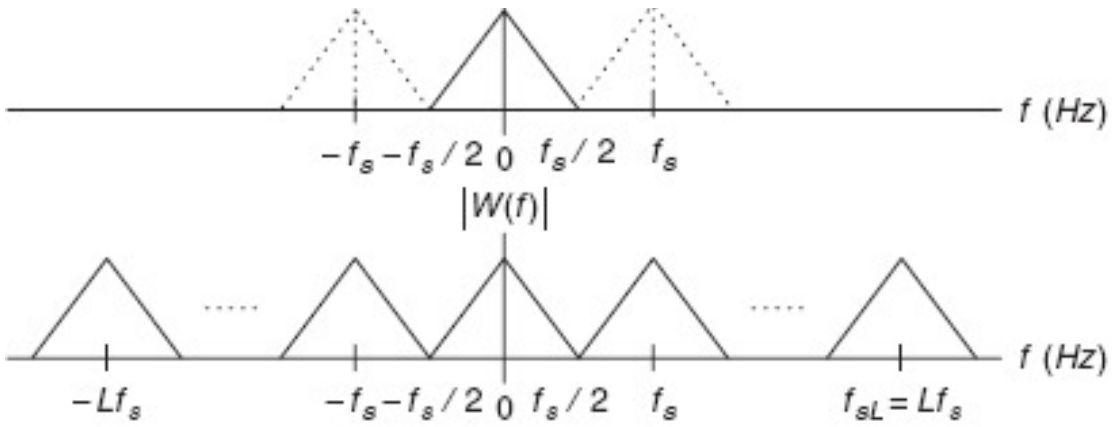
For a 2D image I of size MxN we get a 2D frequency representation using Fourier transform.

Idea: Use Simple Features

Detect Checkerboard Artifacts?

First:

- Look into the Fourier domain to understand their origin
- For simplicity:
  - example of Bed of Nails upsampling
  - 1D



Fourier Transform upsampled a

$$\hat{a}^{up}_{\bar{k}} = \sum_{j=0}^{2\cdot N-1} e^{-2\pi i \cdot \frac{j\bar{k}}{2\cdot N}} \cdot a^{up}_j$$

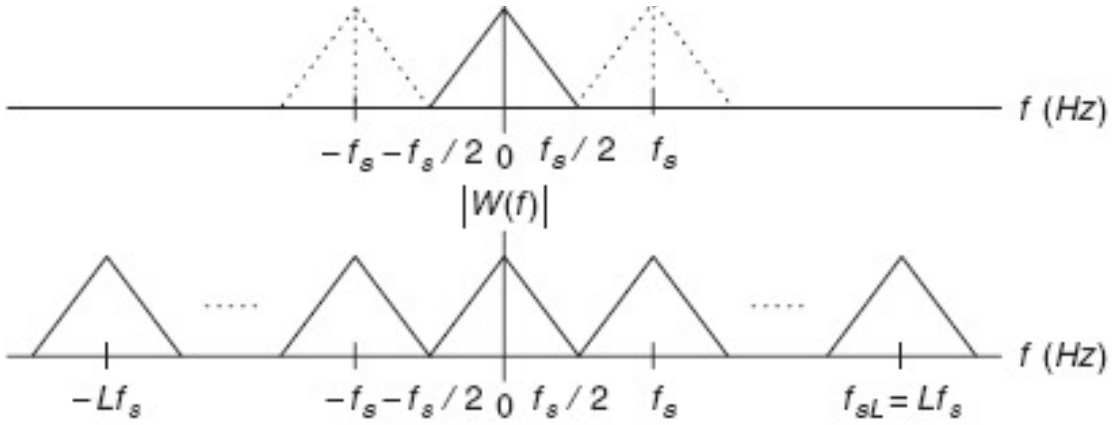
$$= \sum_{j=0}^{2\cdot N-1} e^{-2\pi i \cdot \frac{j\bar{k}}{2\cdot N}} \cdot \sum_{t=-\infty}^{\infty} a^{up}_j \cdot \delta(j-2t)$$

Idea: Use Simple Features

Detect Checkerboard Artifacts?

First:

- Look into the Fourier domain to understand their origin
- For simplicity:
  - example of Bed of Nails upsampling
  - 1D



Fourier Transform upsampled a

$$\hat{a}_{\bar{k}}^{up} = \sum_{j=0}^{2\cdot N-1} e^{-2\pi i \cdot \frac{j\bar{k}}{2\cdot N}} \cdot a_j^{up}$$

$$= \sum_{j=0}^{2\cdot N-1} e^{-2\pi i \cdot \frac{j\bar{k}}{2\cdot N}} \cdot \sum_{t=-\infty}^{\infty} a_j^{up} \cdot \delta(j-2t)$$

Assuming a periodic signal and applying the convolution theorem

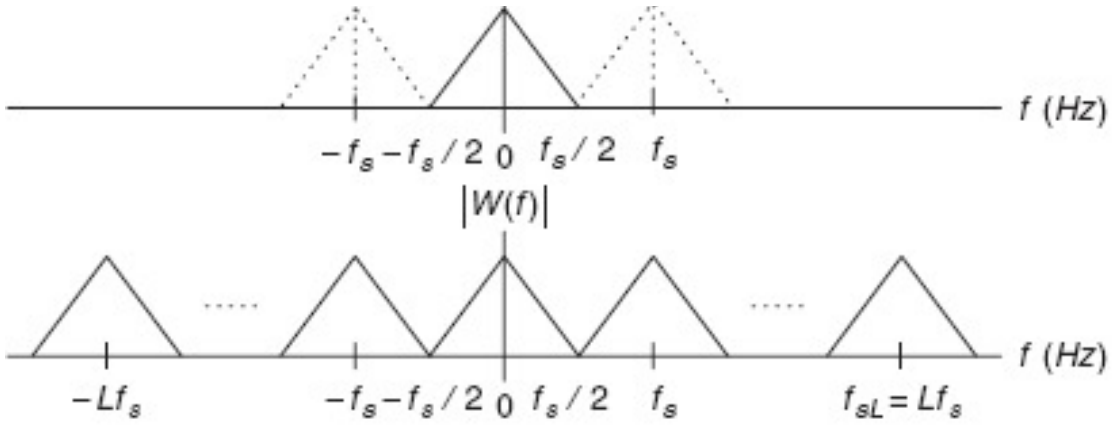
$$= \frac{1}{2} \cdot \sum_{t=-\infty}^{\infty} \left(\sum_{j=-\infty}^{\infty} e^{-2\pi i \cdot \frac{j\bar{k}}{2\cdot N}} a_j^{up} \right) \left(\bar{k} - \frac{t}{2} \right)$$

Idea: Use Simple Features

Detect Checkerboard Artifacts?

First:

- Look into the Fourier domain to understand their origin
- For simplicity:
  - example of Bed of Nails upsampling
  - 1D



Fourier Transform upsampled a

$$\hat{a}^{up}_{\bar{k}} = \sum_{j=0}^{2\cdot N-1} e^{-2\pi i \cdot \frac{j\bar{k}}{2\cdot N}} \cdot a^{up}_j$$

$$= \sum_{j=0}^{2\cdot N-1} e^{-2\pi i \cdot \frac{j\bar{k}}{2\cdot N}} \cdot \sum_{t=-\infty}^{\infty} a^{up}_j \cdot \delta(j-2t)$$

Assuming a periodic signal and applying the convolution theorem

$$\frac{1}{2} \cdot \sum_{t=-\infty}^{\infty} \left(\sum_{j=-\infty}^{\infty} e^{-2\pi i \cdot \frac{j\bar{k}}{N}} \cdot a_j \right) \left(\bar{k} - \frac{t}{2} \right)$$

Idea: Use Simple Features

Detect Checkerboard Artifacts?

Such upsampling is repeated several times in GANs, to get the final resolution

Convolutional layers in between are too few and have too small filters

Artifacts will not only appear in the highest frequencies

Simple Features ...

2D Discrete Fourier Transformation DFT

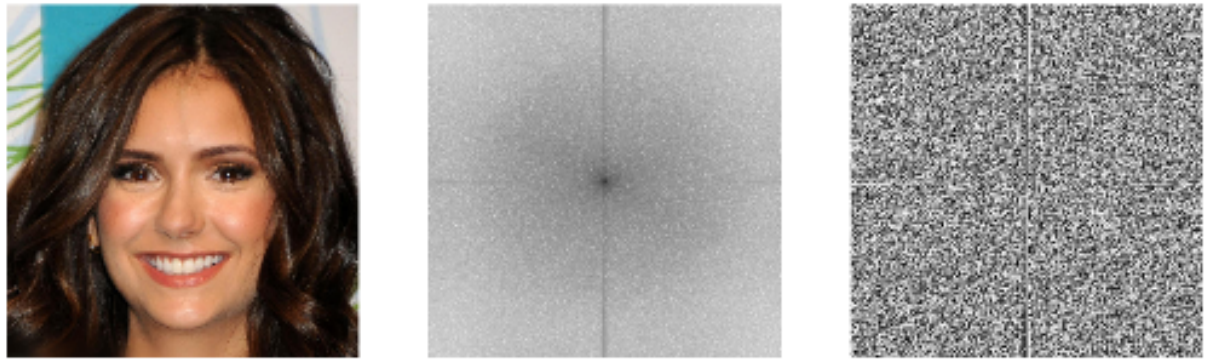


Fig. 3: Example of a DFT applied to a sample. (Left) Input image I. (Center) Power Spectrum. (Right) Phase Spectrum.

$$\mathcal{F}(I)(k,\ell) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} e^{-2\pi i \cdot \frac{mk}{M}} e^{-2\pi i \cdot \frac{n\ell}{N}} \cdot I(m,n), \quad (1)$$

$$\text{for} \quad k = 0,\ldots,M-1, \quad \ell = 0,\ldots,N-1,$$

For a 2D image I of size MxN we get a 2D frequency representation.

Durall, Keuper, Keuper: Watch your Up-Convolution: CNN-Based Methods are Failing to Reproduce Spectral Distributions, CVPR 2020.

Simple Features ...

2D DFT → 1D feature

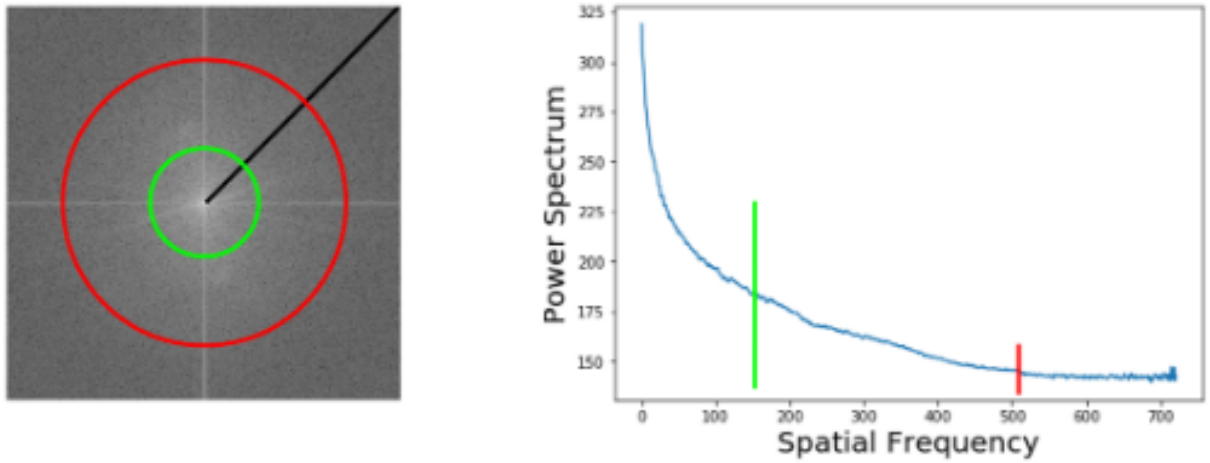


Fig. 4: Example of an azimuthal average. (Left) Power Spectrum 2D. (Right) Power Spectrum 1D. Each frequency component is the radial average from the 2D spectrum.

$$AI(\omega_k) = \int_0^{2\pi} \|\mathcal{F}(I)\,(\omega_k \cdot \cos(\phi), \omega_k \cdot \sin(\phi))\,\|^2 \mathrm{d}\phi$$
$$\text{for} \quad k = 0, \ldots, M/2 - 1\,, \qquad (2)$$

“Frequency profile of an Image”

Frequency properties of GAN outputs

… based on transposed convolutions for upsampling

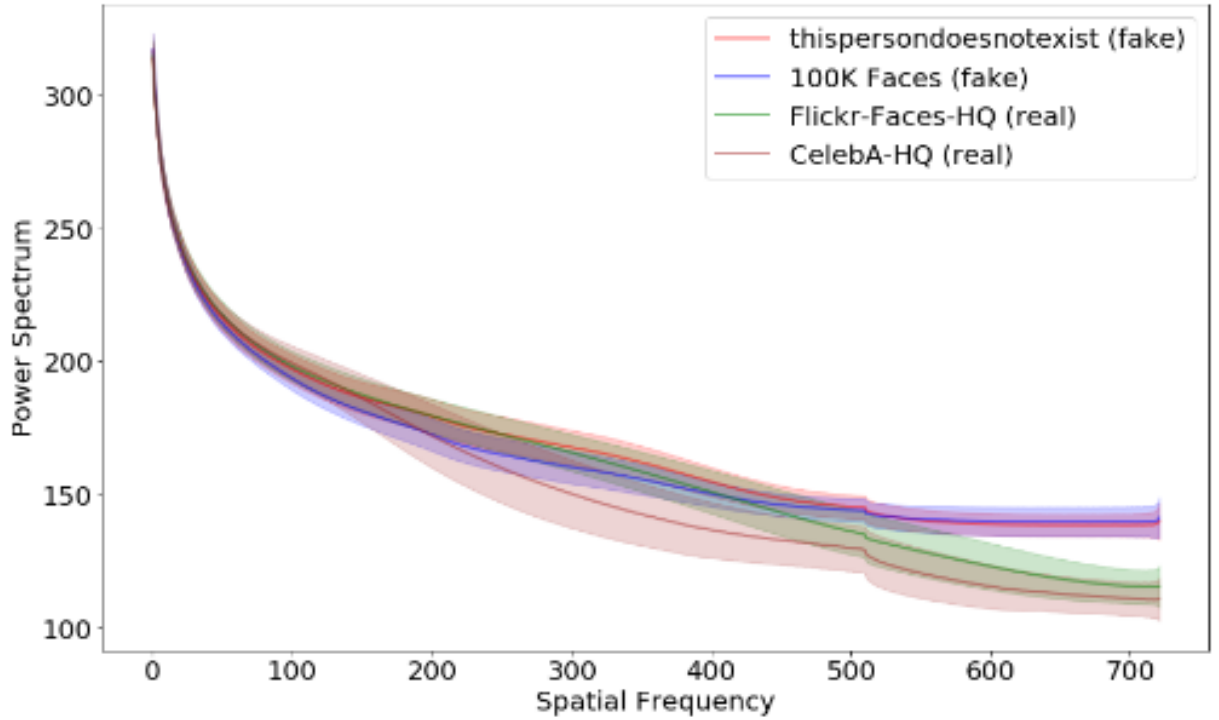
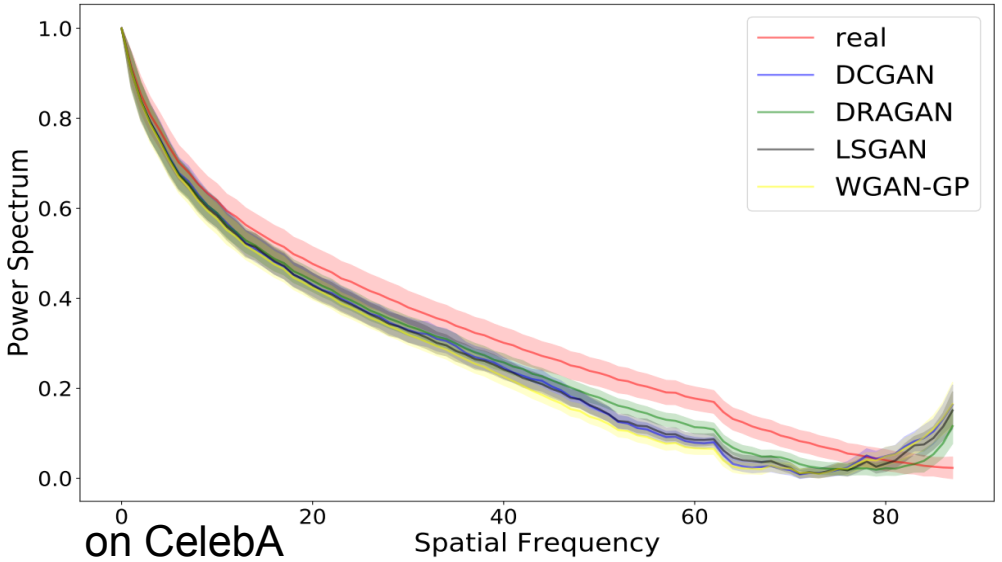


Fig. 1: 1D power spectrum statistics from each sub-data set from Faces-HQ. The higher the frequency, the bigger is the difference between real or fake data.

Frequency properties of GAN outputs

… based on transposed convolutions for upsampling



on CelebA

A. Radford et al.. Unsupervised representation learning with deep convolutional generative adversarial networks. 2015
N. Kodali et al.. On convergence and stability of gans. 2017
X. Mao et al.. Least squares generative adversarial networks. ICCV17
I. Gulrajani et al.. Improved training of wasserstein gans. 2017

Simple Features for detection

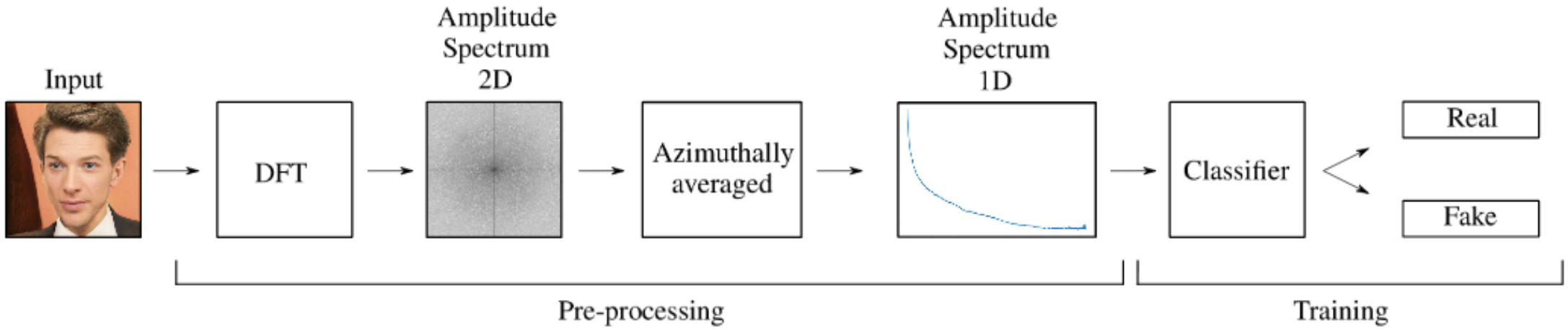


Fig. 2: Overview of the processing pipeline of our approach. It contains two main blocks, a feature extraction block using DFT and a training block, where a classifier uses the new transformed features to determine whether the face is real or not. Notice that input images are transformed to grey-scale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

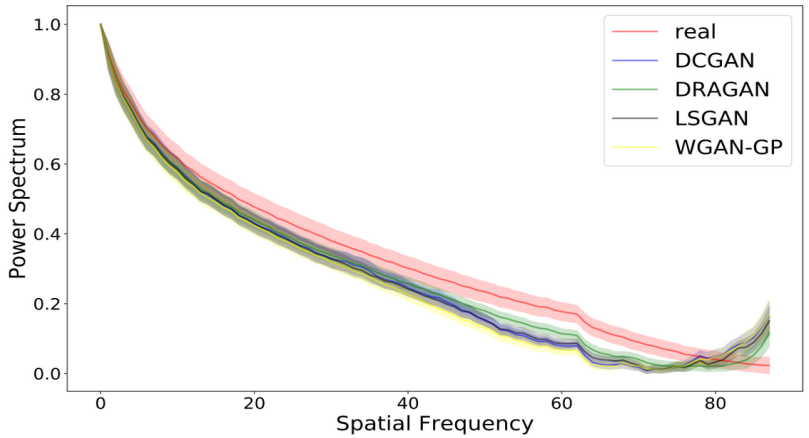
Frequency properties of GAN outputs

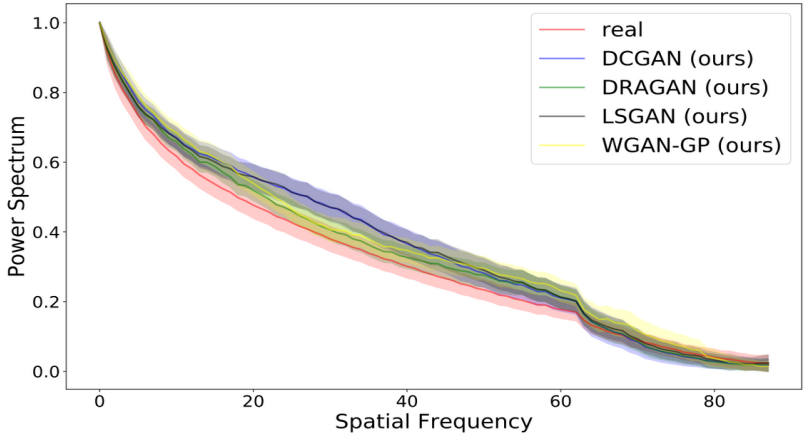
dataset	# train samples	SVM	k-means
Faces HQ	1000	100%	77%
	100	100%	75%
	20	100%	72%
CelebA	2000	100%	96%
DeepFakes (FaceForensics++)	2000	91%	82%

Frequency properties of GAN outputs

Problem Analysis

Proposed “fix” using a regularization (Cross-Entropy Loss on the frequency profile)

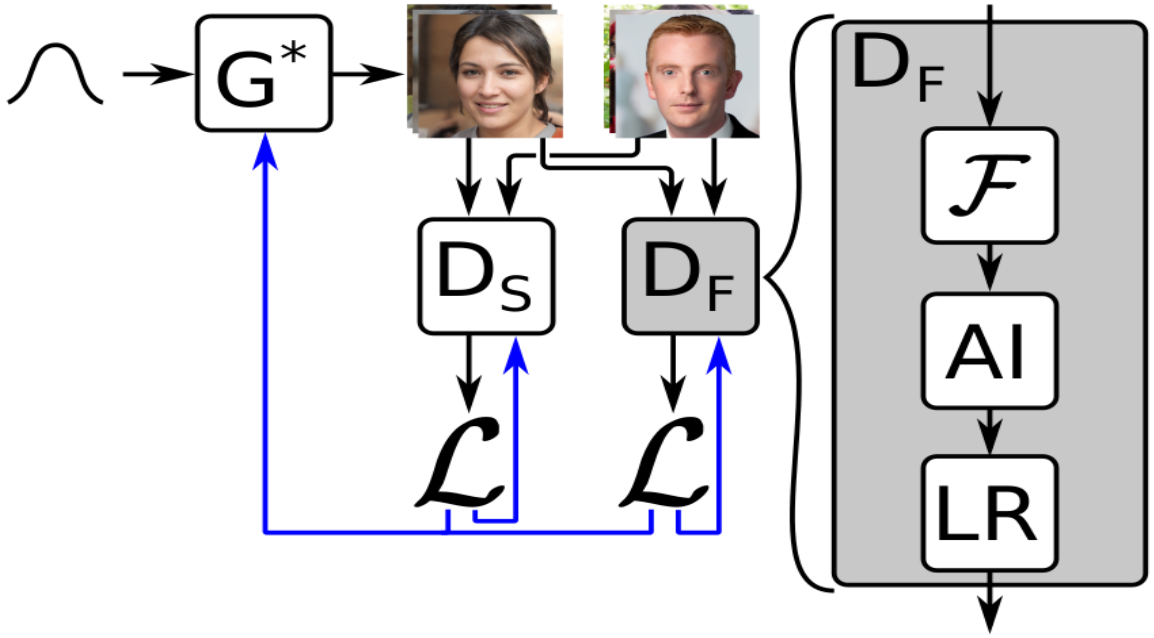


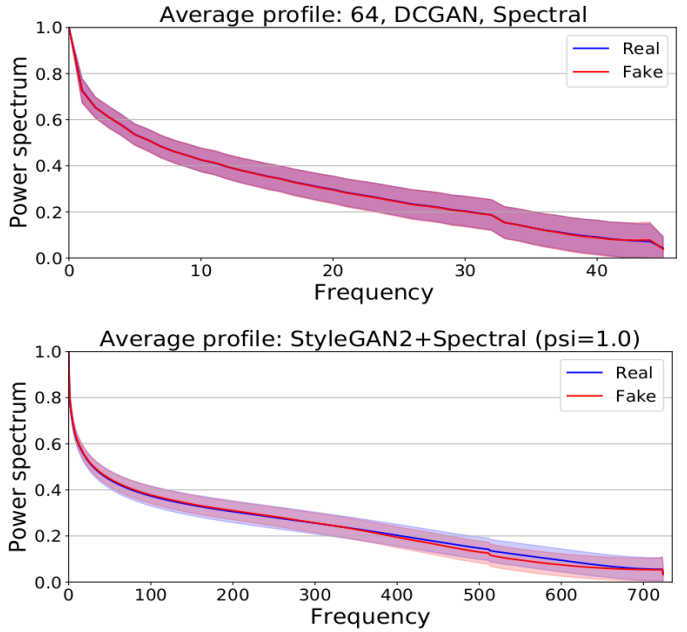


Durall, Keuper, Keuper:Watch your Up-Convolution: CNN-Based Methods are Failing to Reproduce Spectral Distributions, CVPR 2020.

Frequency properties of GAN outputs

Better Fix: **Spectral Discriminator Generative Adversarial Network**





Jung & Keuper: Spectral Distribution aware Image Generation, AAAI'21.

Task Description

Question: **Is the detection approach working for other upsampling approaches?**

Data:
Train and Test folder containing images from the ImageWoof dataset (low resolution dog images), and images generated using SNGAN, modified to use

- bilinear interpolation upsampling
- bicubic interpolation
- pixel shuffle upsampling

Task Description

Question: **Is the detection approach working for other upsampli approaches?**

Why SNGAN (Spectral Normalization GAN)?

- High quality class conditional samples at Imagenet scale
- First GAN to work on full Imagenet (million image dataset)
- Computational benefits over similarly performing GANs

SPECTRAL NORMALIZATION FOR GENERATIVE ADVERSARIAL NETWORKS

Takeru Miyato[1], Toshiki Kataoka[1], Masanori Koyama[2], Yuichi Yoshida[3]
{miyato, kataoka}@preferred.jp
koyama.masanori@gmail.com
yyoshida@nii.ac.jp
[1]Preferred Networks, Inc. [2]Ritsumeikan University [3]National Institute of Informatics

ABSTRACT

One of the challenges in the study of generative adversarial networks is the instability of its training. In this paper, we propose a novel weight normalization technique called spectral normalization to stabilize the training of the discriminator. Our new normalization technique is computationally light and easy to incorporate into existing implementations. We tested the efficacy of spectral normalization on CIFAR10, STL-10, and ILSVRC2012 dataset, and we experimentally confirmed that spectrally normalized GANs (SN-GANs) is capable of generating images of better or equal quality relative to the previous training stabilization techniques. The code with Chainer (Tokui et al., 2015), generated images and pretrained models are available at https://github.com/pfnet-research/sngan_projection.

Task Description

Question: **Is the detection approach working for other upsampling approaches?**

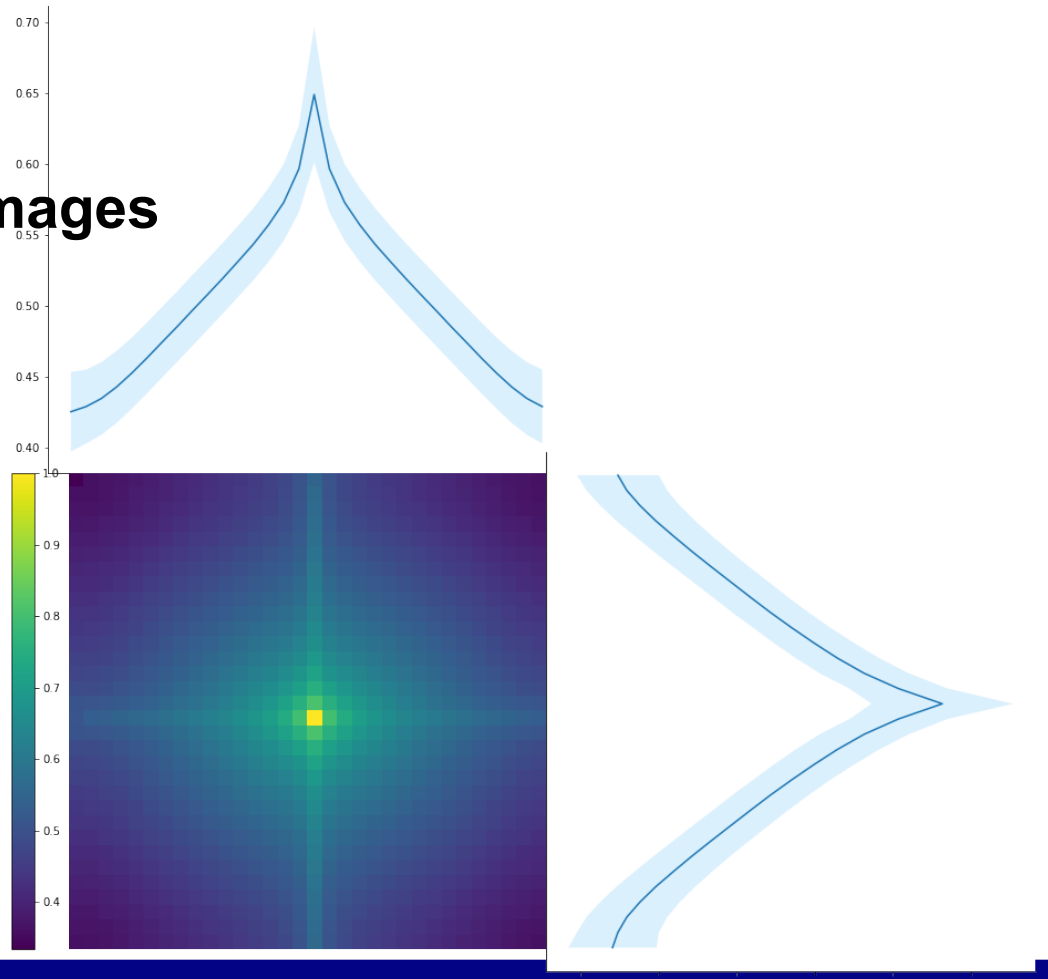
Data Analysis:

- bilinear interpolation upsampling
- bicubic interpolation
- pixel shuffle upsampling

Task Description

Question: **Is the detection approach working for other upsampling approaches?**

Data Analysis: mean spectra of original images

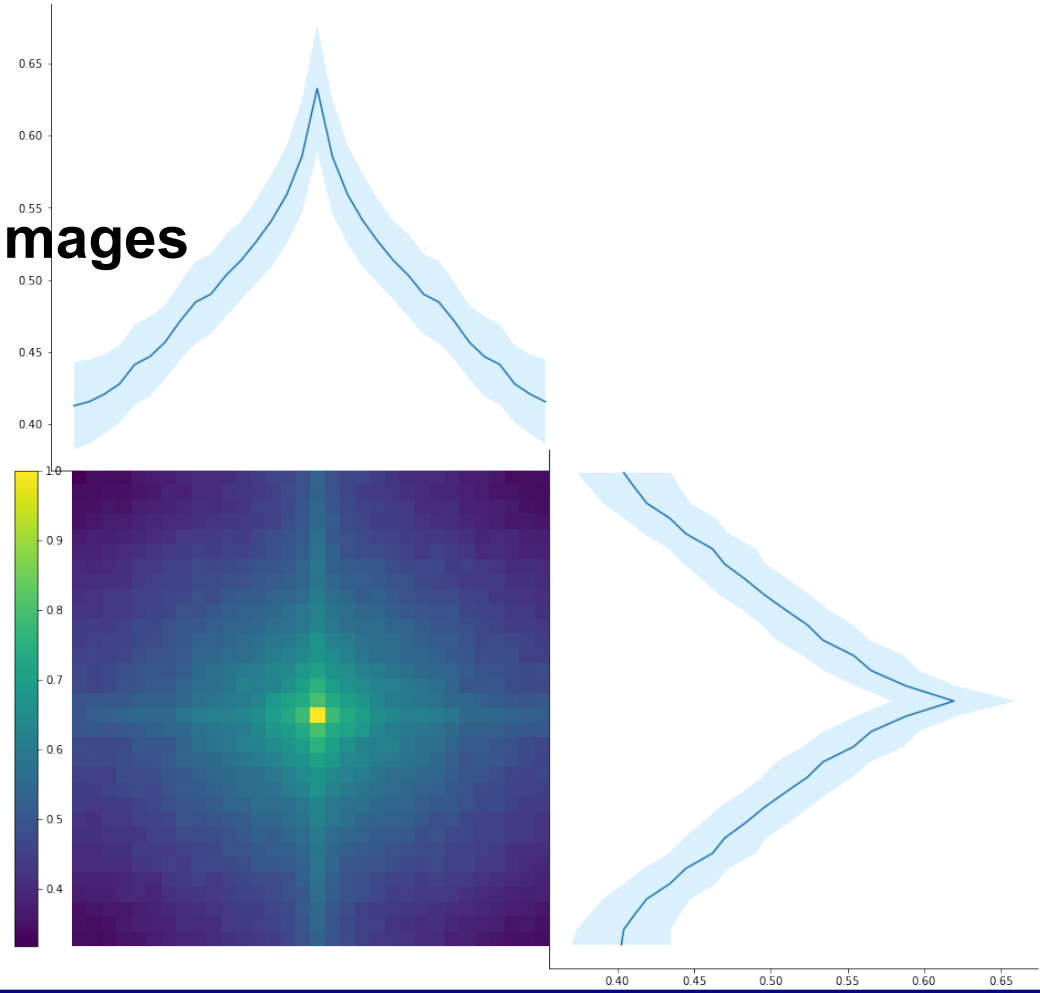


Task Description

Question: **Is the detection approach working for other upsampling approaches?**

Data Analysis: mean spectra of generated images

Bilinear Interpolation

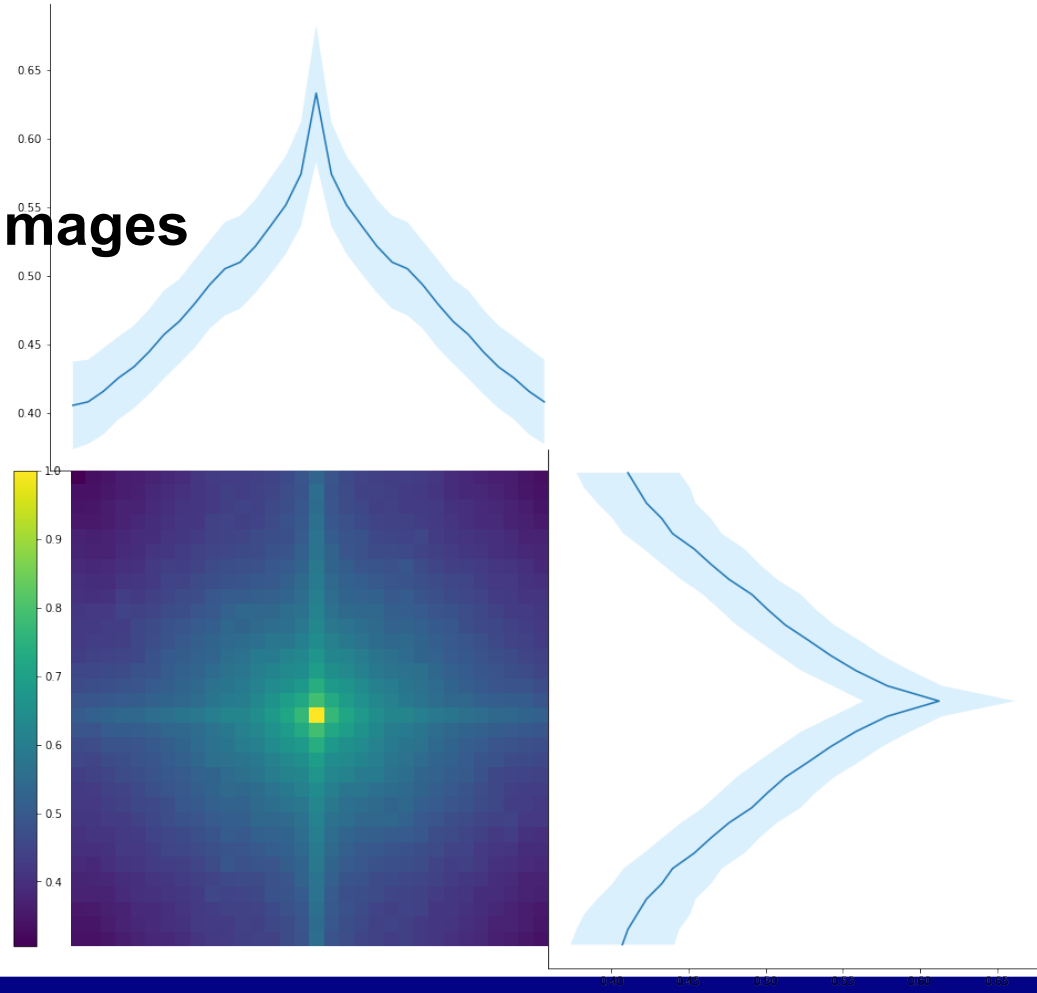


Task Description

Question: **Is the detection approach working for other upsampling approaches?**

Data Analysis: mean spectra of generated images

Bicubic Interpolation

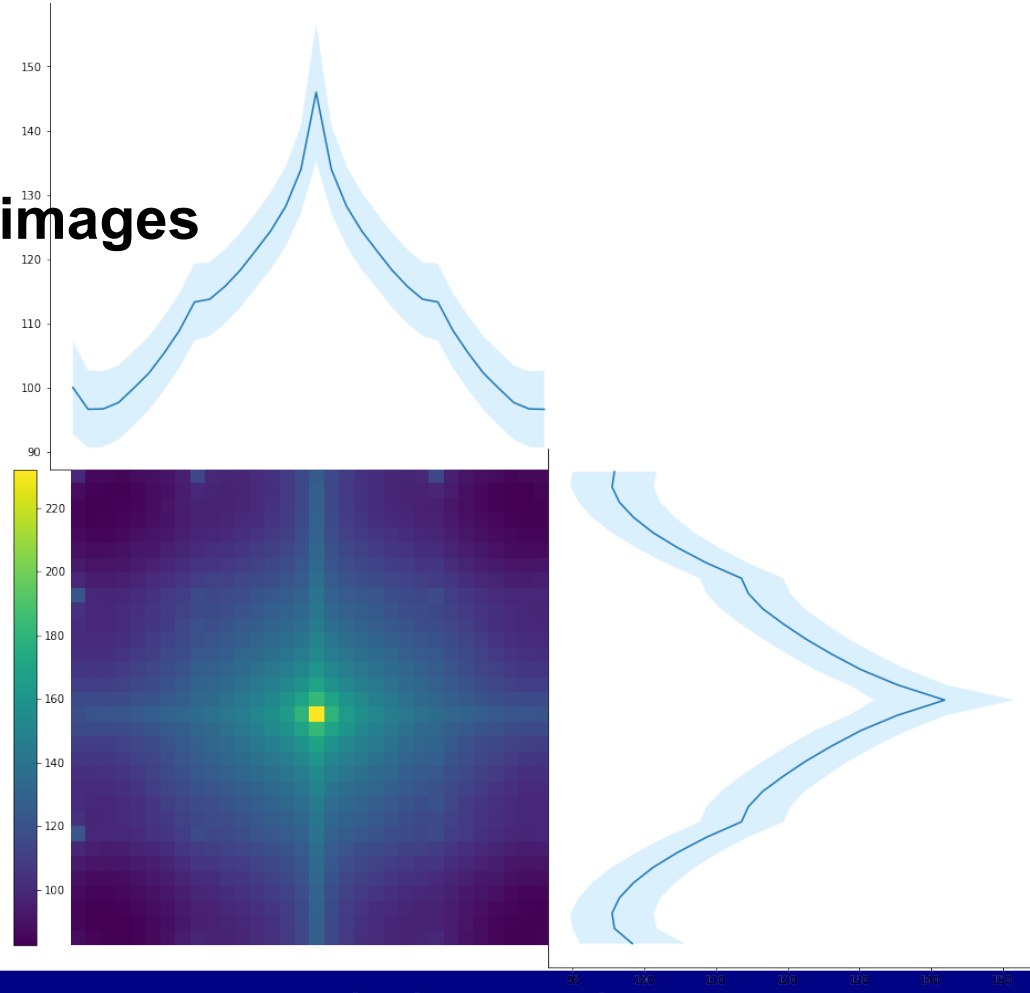


Task Description

Question: **Is the detection approach working for other upsampling approaches?**

Data Analysis: mean spectra of generated images

Pixel Shuffle

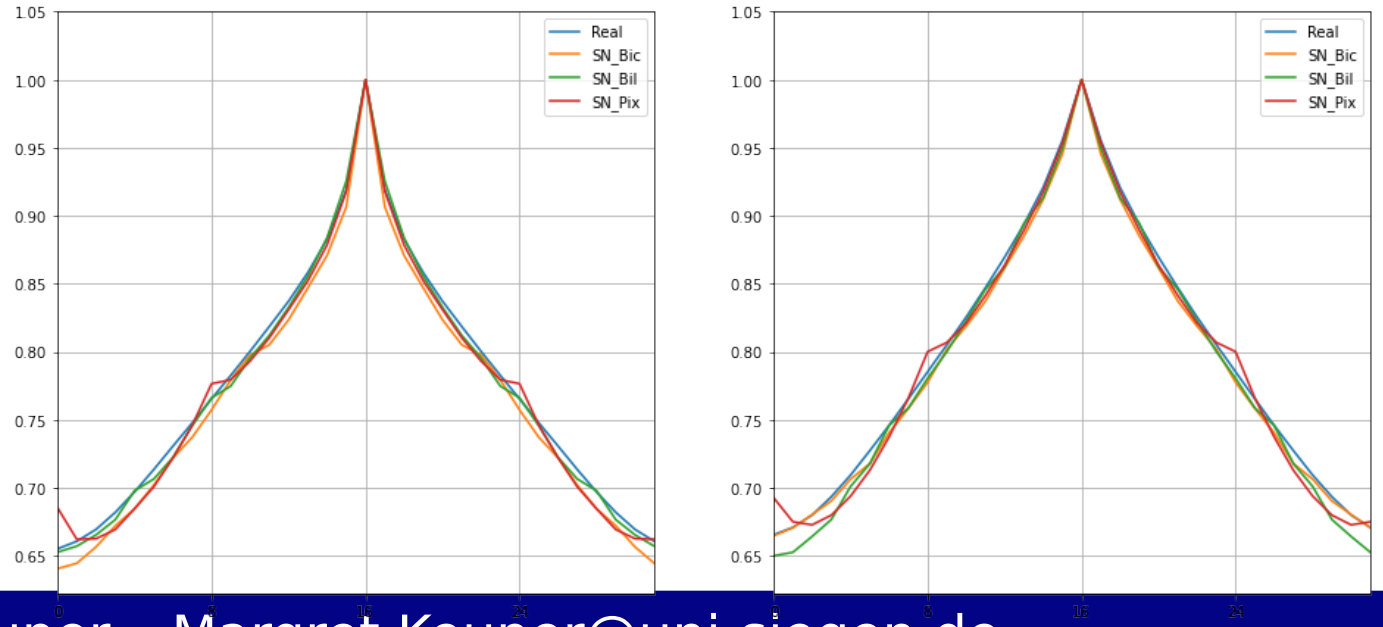


Task Description

Question: **Is the detection approach working for other upsampling approaches?**

Data Analysis: Comparison

Comparison between Real, SNGAN (bilinear, bicubic, pixelshuffle) - Normalized - left: colomn wise right: row-wise



Task Description

Question: **Is the detection approach working for other upsampling approaches?**
Can you get better results by using vertical and horizontal projections instead of radial ones?
Can you get better results by using a neural network for the detection?

Task Description

Approach:

Start from image pixels or from the Fourier transform (DFT).

Train a classifier to discriminate between the original images and generated images for each upsampling method separately.

- classifier can be a neural net, a cnn, or a classical model such as SVM and Random Forest.

Optimize for one setting that works well for the three generators (same hyerparameters).

Try to transfer a learned classifier from one generator to the other.
You can also train a classifier on the three generators (balance the data in this case!).

Conclusion

- Frequently used CNN architectures suffer from sampling issues in image classification and generation.
- Use such upsampling artifacts for DeepFake detection.

83

Thank You!