DEV DAY

DEV DAY

MLT4

An Introduction to Generative Adversarial Networks

Julien Simon
Global Evangelist, AI & Machine Learning
Amazon Web Services

@julsimon

What are Generative Adversarial Networks?

- Most Machine Learning problems try to make sense of a real-life dataset
  - Tabular data: regression, classification, clustering
  - Text: sentiment analysis, entity extraction
  - Image: object detection, image segmentation

- Generative Adversarial Networks are fundamentally different
  - Goodfellow, 2014 https://arxiv.org/abs/1406.2661

- A GAN uses a real-life dataset to generate new lookalike samples
  - Not based on statistical analysis, e.g. filling in missing values during feature engineering

DEV DAY

A Light Introduction to GAN Theory

Adversarial Networks

- Two deep learning networks compete against one another

- The Generator network creates new samples from random data, not from the data set
  - Start from a random vector (say, 100 bytes) and generate a tensor with the same shape as dataset samples (say, 3x256x256)

- The Discriminator learns how to classify samples
  - Real ones, coming from the data set
  - Fake ones, generated by the Generator
  - Typical Deep Learning process: mini-batch training, backpropagation, etc.

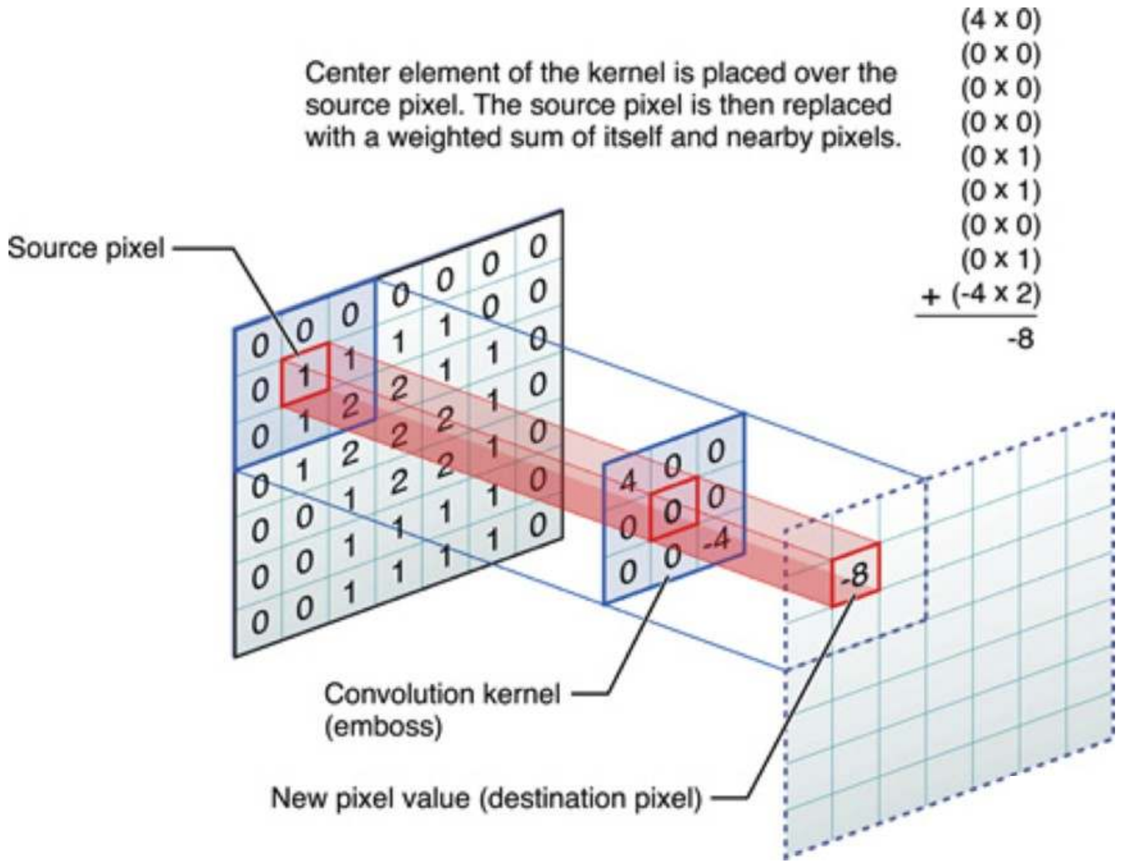
« How do you create something from nothing? »

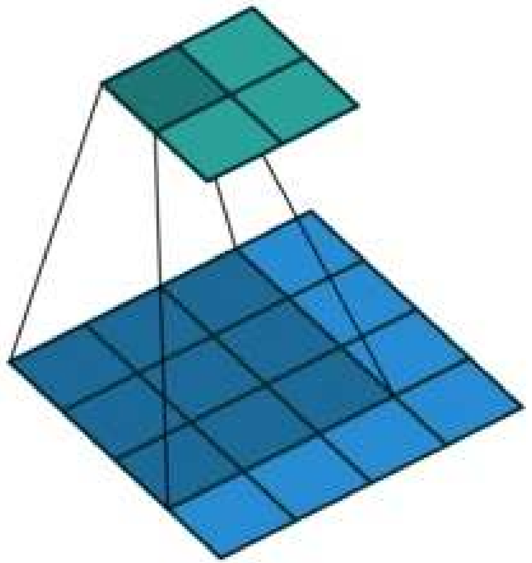
« How can fake samples look remotely convincing? »

« How does the Generator learn? »

Let’s take about convolution for a minute

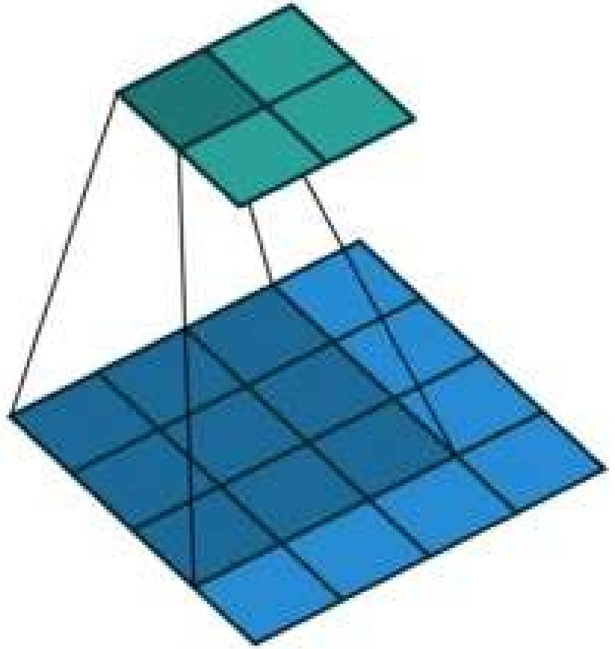
- Convolution extracts features from complex data samples
- It reduces the total amount of information
- Samples can shrink, keeping only the « good stuff »
  - Padding and stride
- So we know how to go from (3, 256, 256) to [0,1]
  - From images to probabilities
- Can we do the opposite?


Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.
(4 x 0)
(0 x 0)
(0 x 0)
(0 x 0)
(0 x 1)
(0 x 1)
(0 x 0)
(0 x 1)
+ (-4 x 2)
-8
Source pixel
Convolution kernel (emboss)
New pixel value (destination pixel)



Transposed convolution

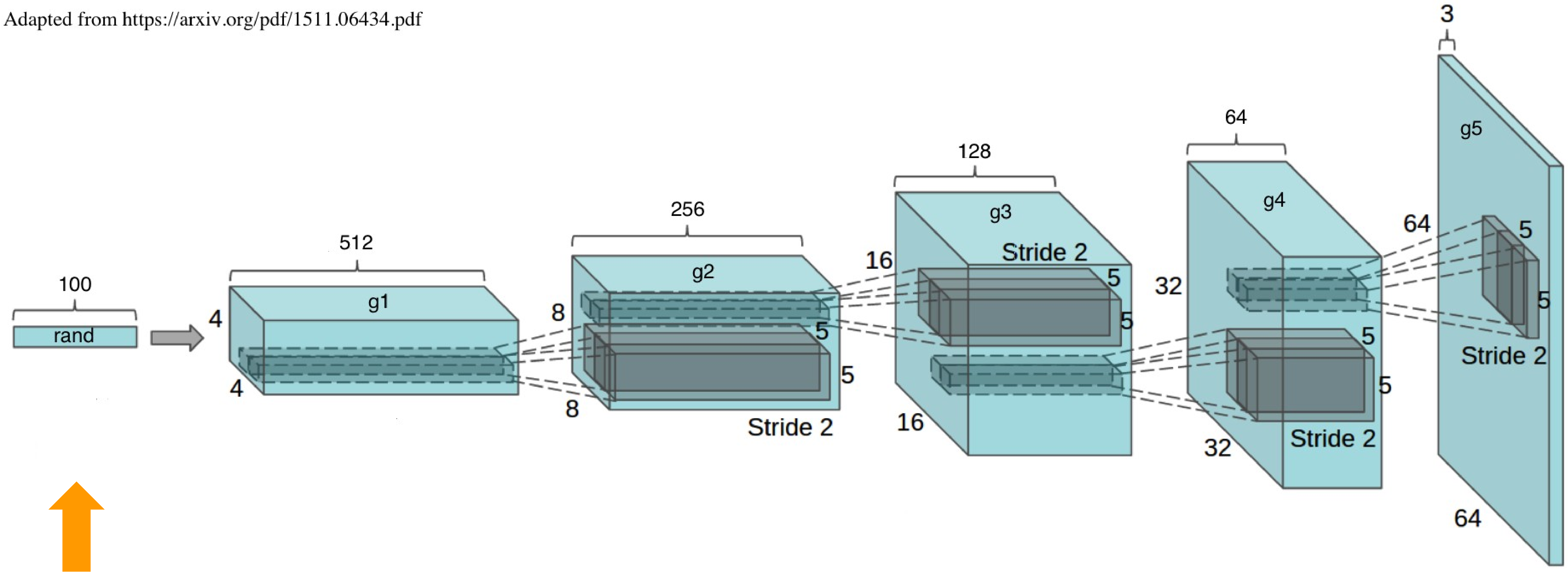
- Transposed convolution is the reverse operation
- It increases the total amount of information
- Samples grow
- So we know how to go from (100,) to (3, 256, 256)



The Deep Convolutional Generator (DCGAN)

https://arxiv.org/abs/1511.06434

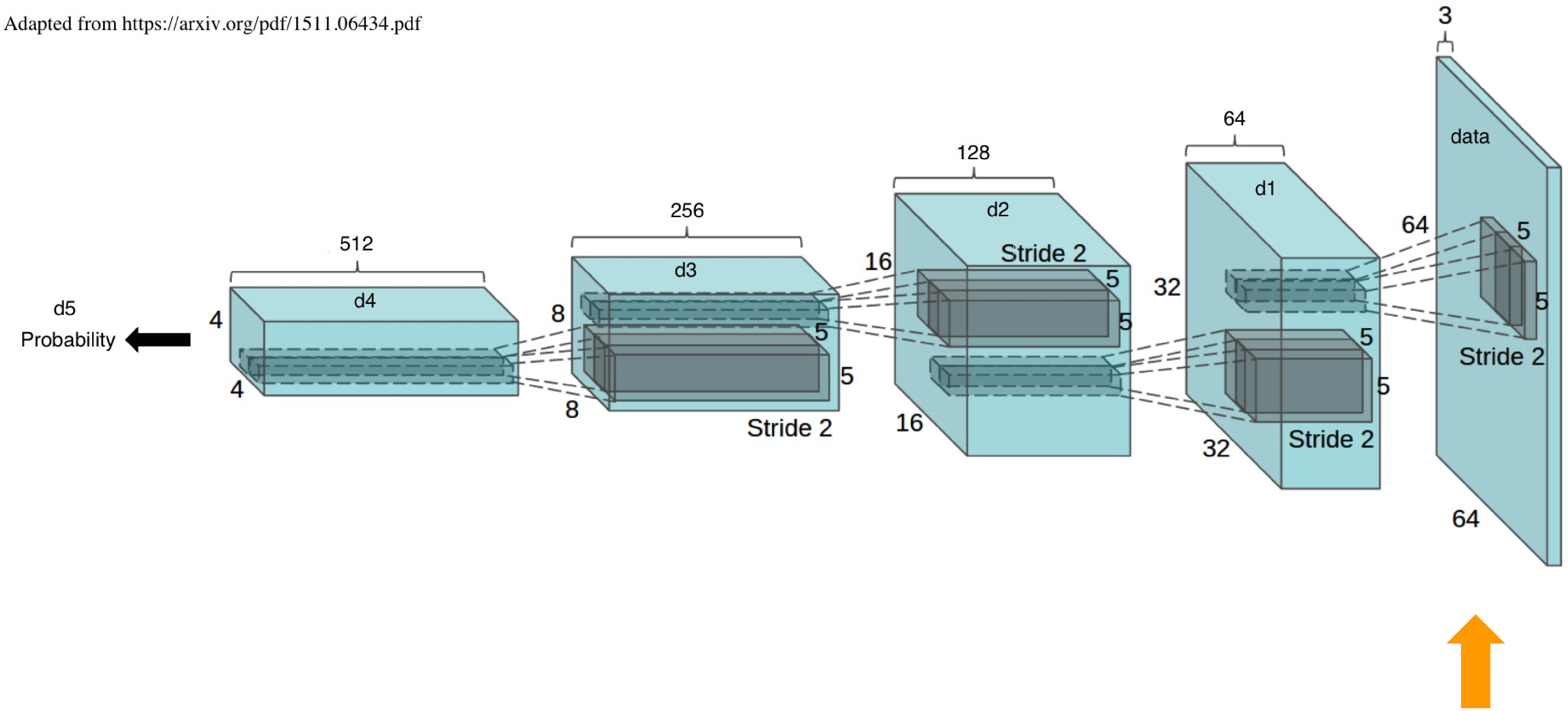
Adapted from https://arxiv.org/pdf/1511.06434.pdf



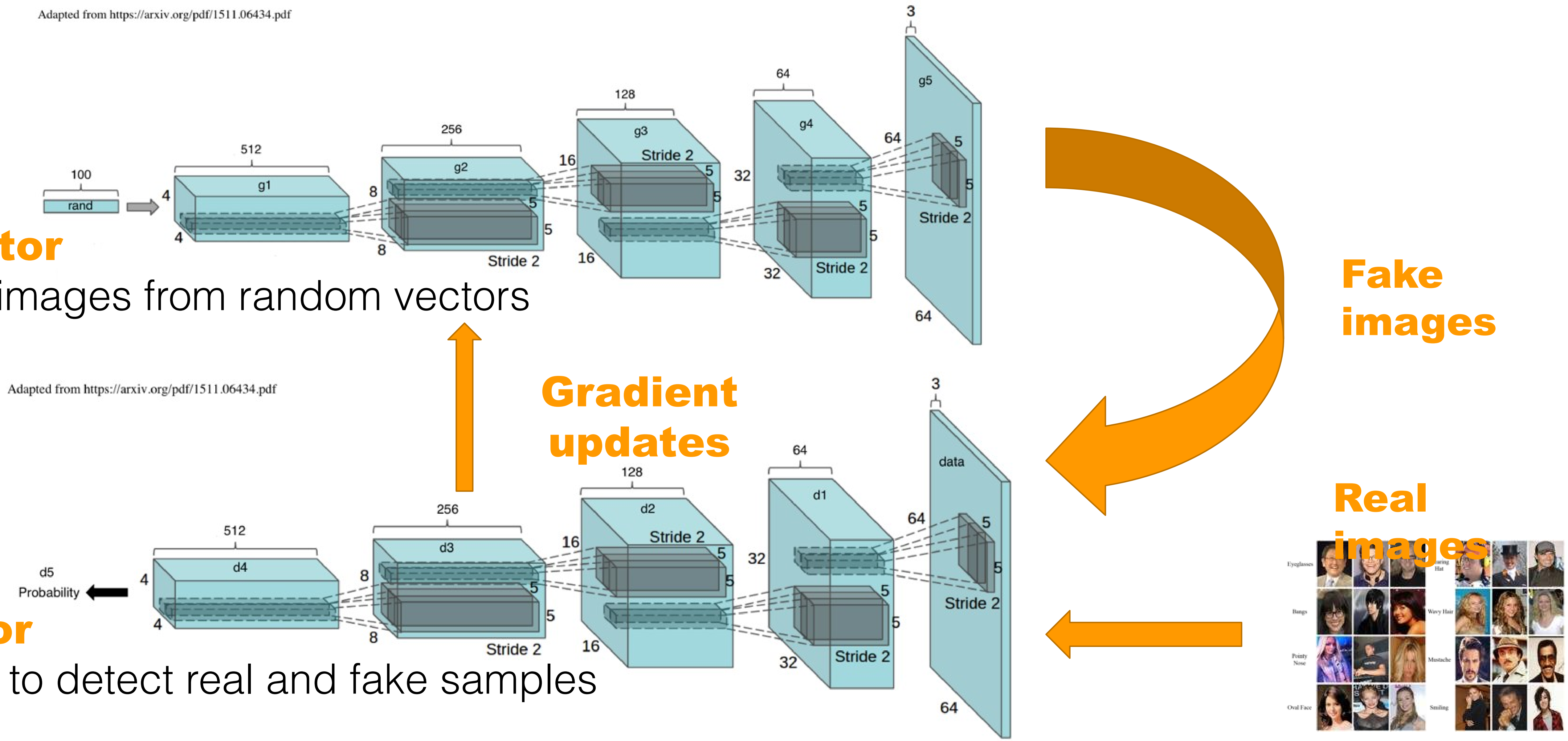
The Deep Convolutional Discriminator (DCGAN)

https://arxiv.org/abs/1511.06434

Adapted from https://arxiv.org/pdf/1511.06434.pdf



Training DCGAN



Adapted from https://arxiv.org/pdf/1511.06434.pdf

Generator

Building images from random vectors

Fake images

Adapted from https://arxiv.org/pdf/1511.06434.pdf

Gradient updates

Real images

Detector

Learning to detect real and fake samples

https://medium.com/@julsimon/generative-adversarial-networks-on-apache-mxnet-part-1-b6d39e6b5df1

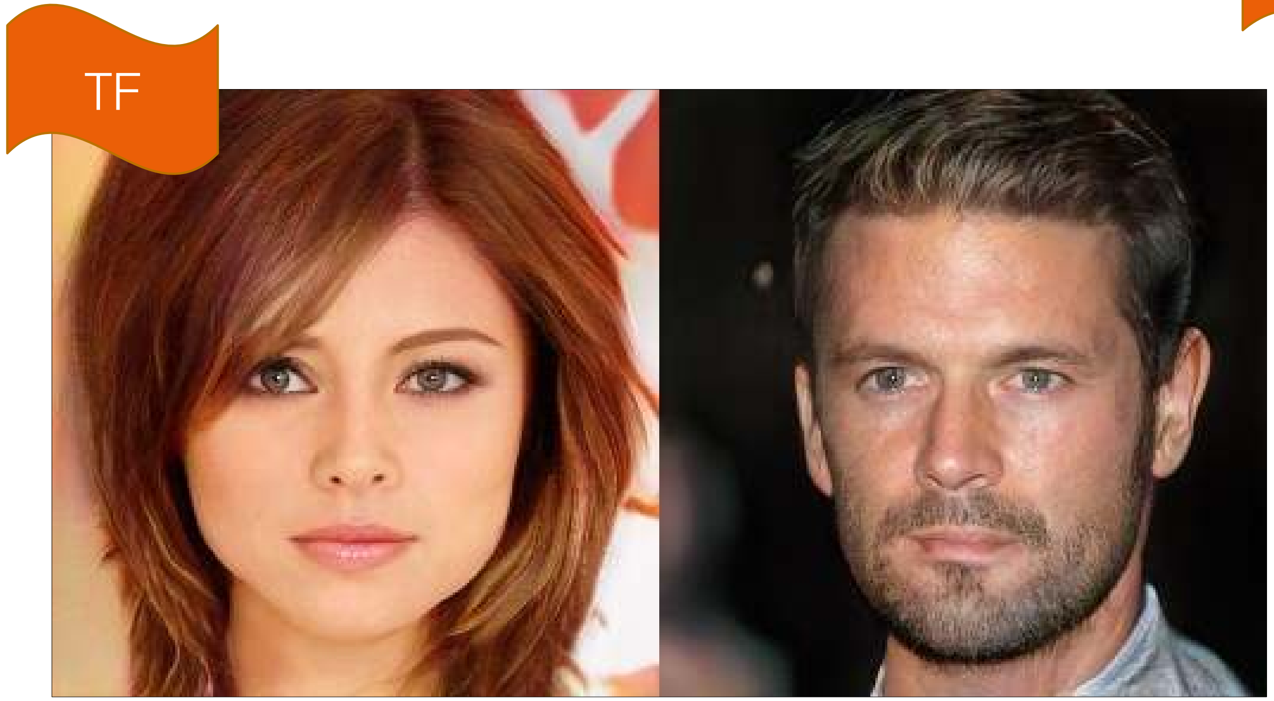
Demo:
training DCGAN with MXNet Gluon

https://gitlab.com/juliensimon/dlnotebooks/-/blob/master/mxnet/07%20-%20DCGAN%20Gluon.ipynb

DEV DAY

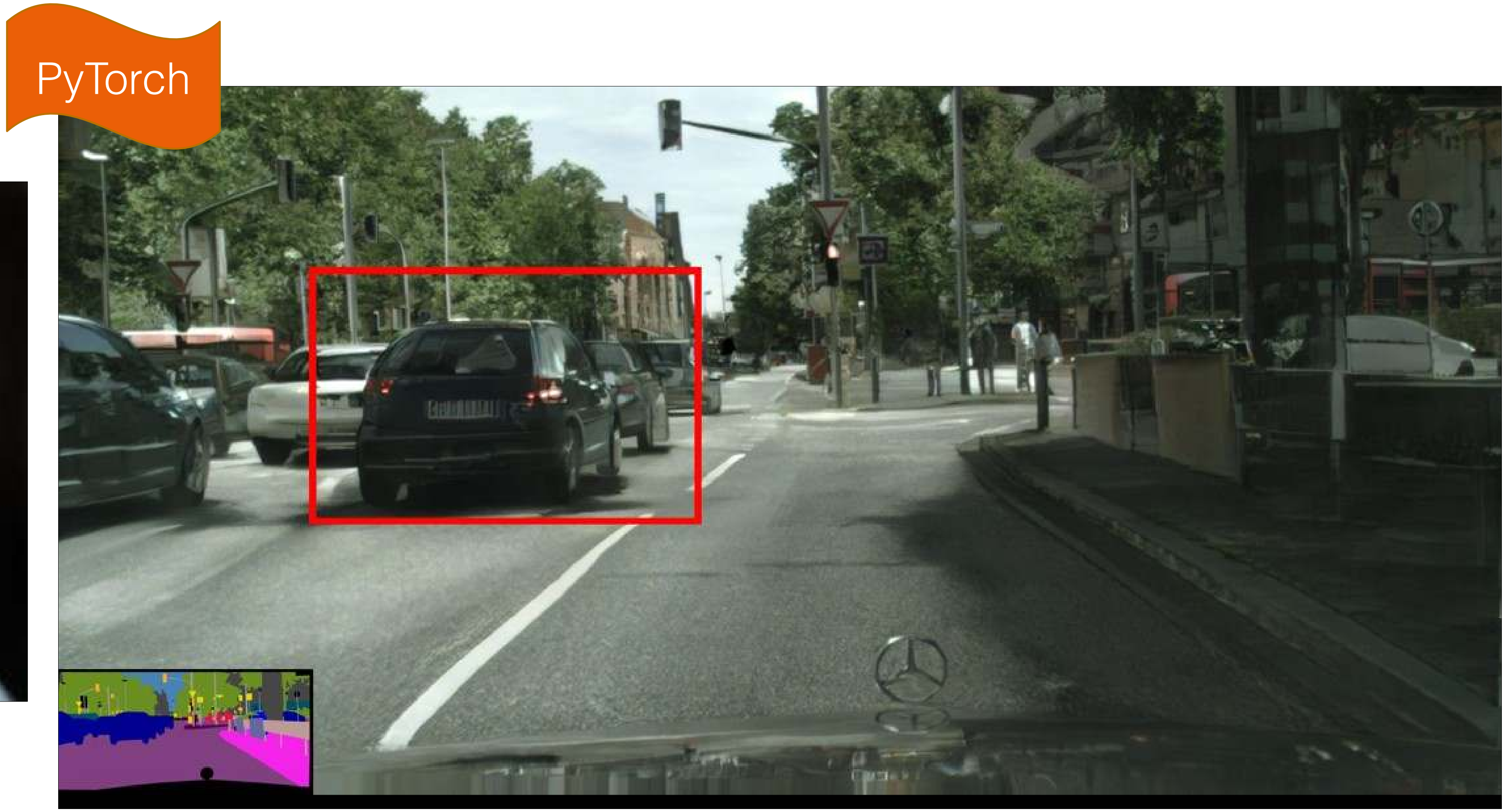
Sample Projects

GAN: Welcome to the (un)real world, Neo



Generating new ”celebrity” faceshttps://github.com/tkarras/progressive_growing_of_gans

April 2018



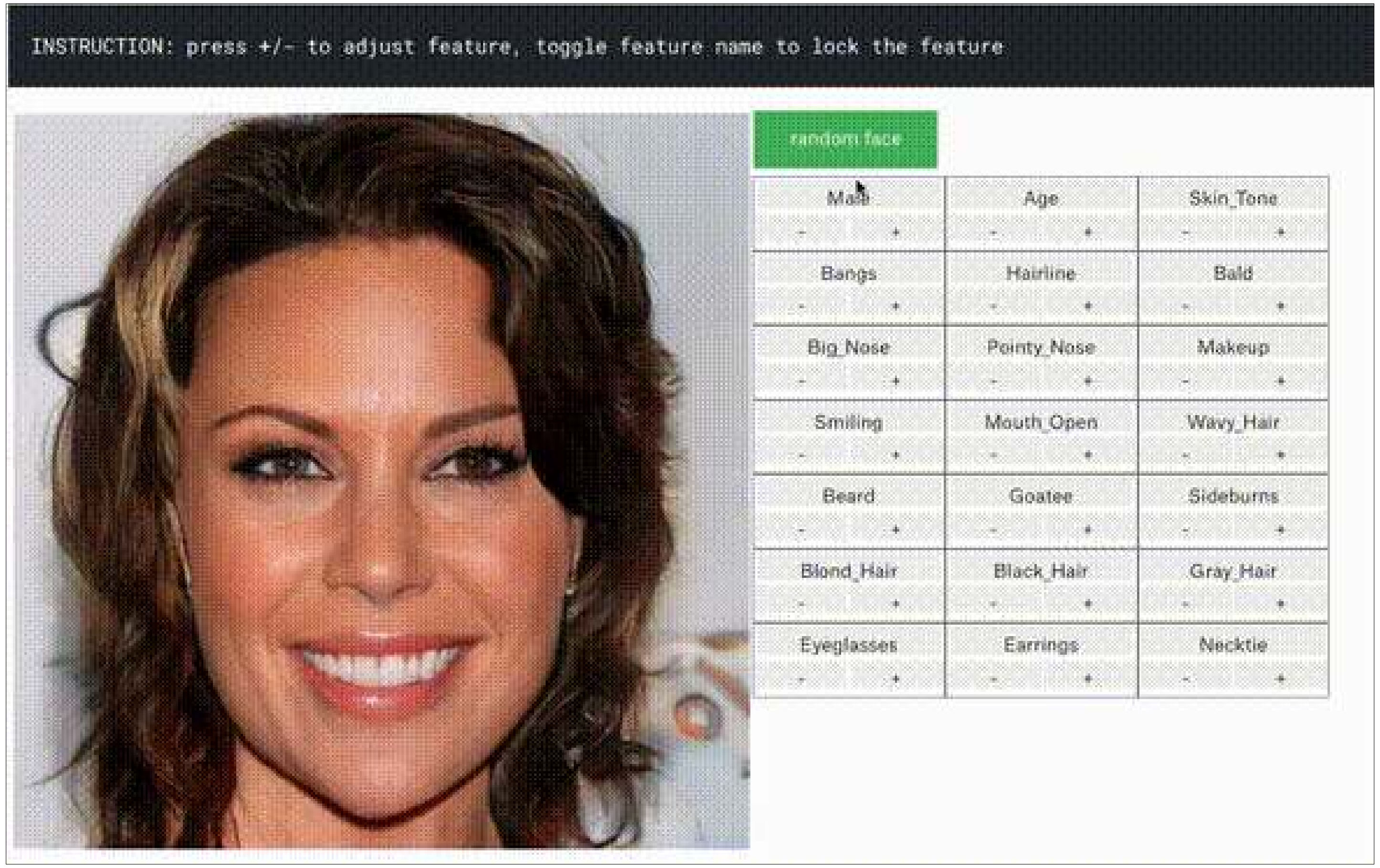
From semantic map to 2048x1024 picture

https://tcwang0509.github.io/pix2pixHD/

November 2017

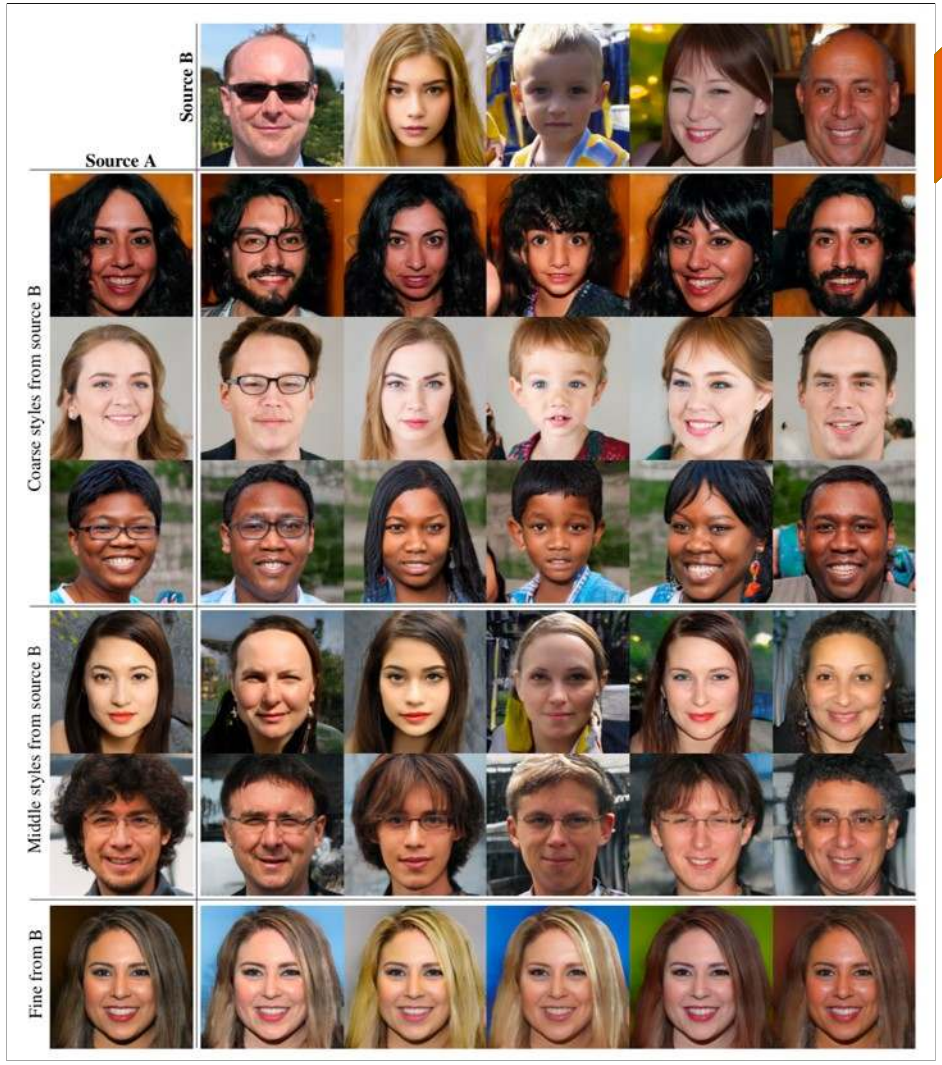
More face generation with GANs

TF



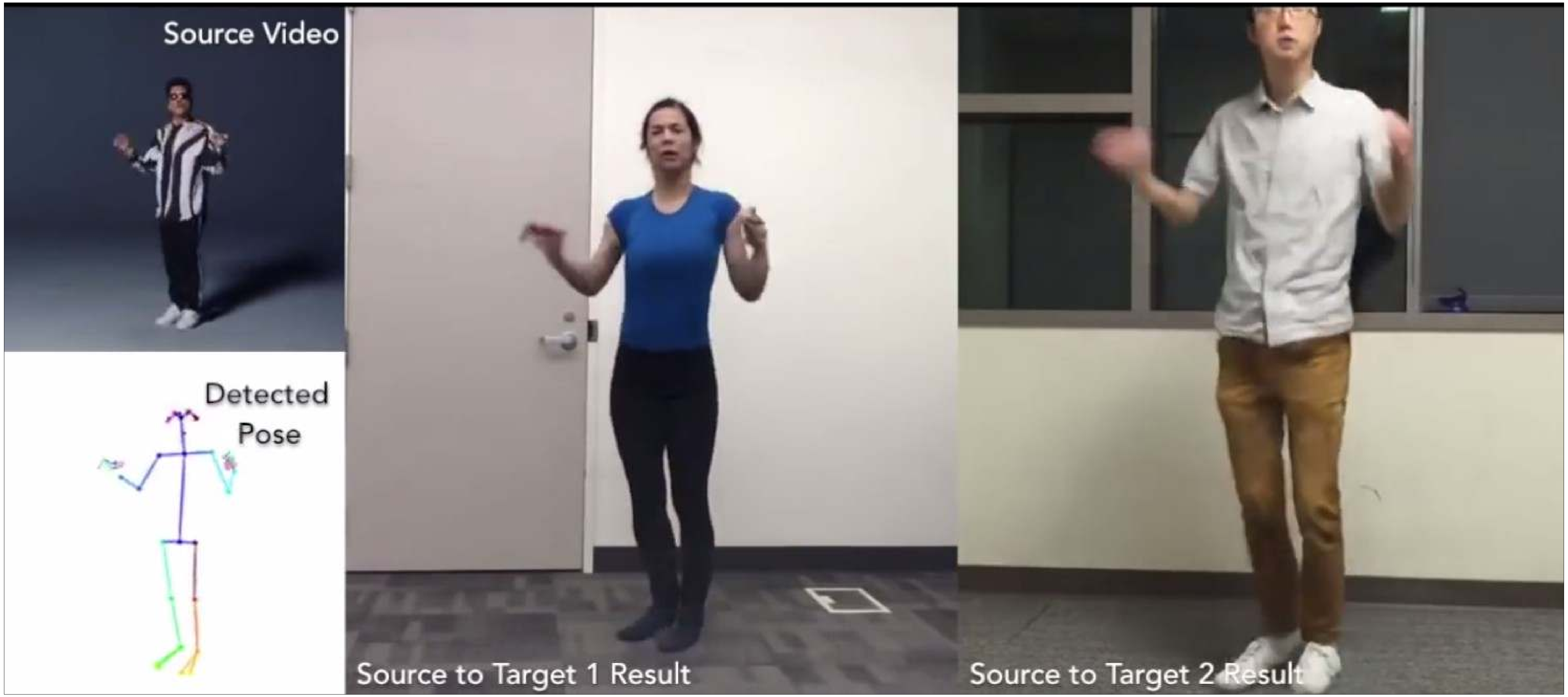
INSTRUCTION: press +/- to adjust feature, toggle feature name to lock the feature

Controlled Image Generation with TL-GAN
https://github.com/SummitKwan/transparent_latent_gan
October 2018



Applying the style of a face to another face https://github.com/NVlabs/stylegan
https://www.youtube.com/watch?v=kSLJriaOumA
March 2019

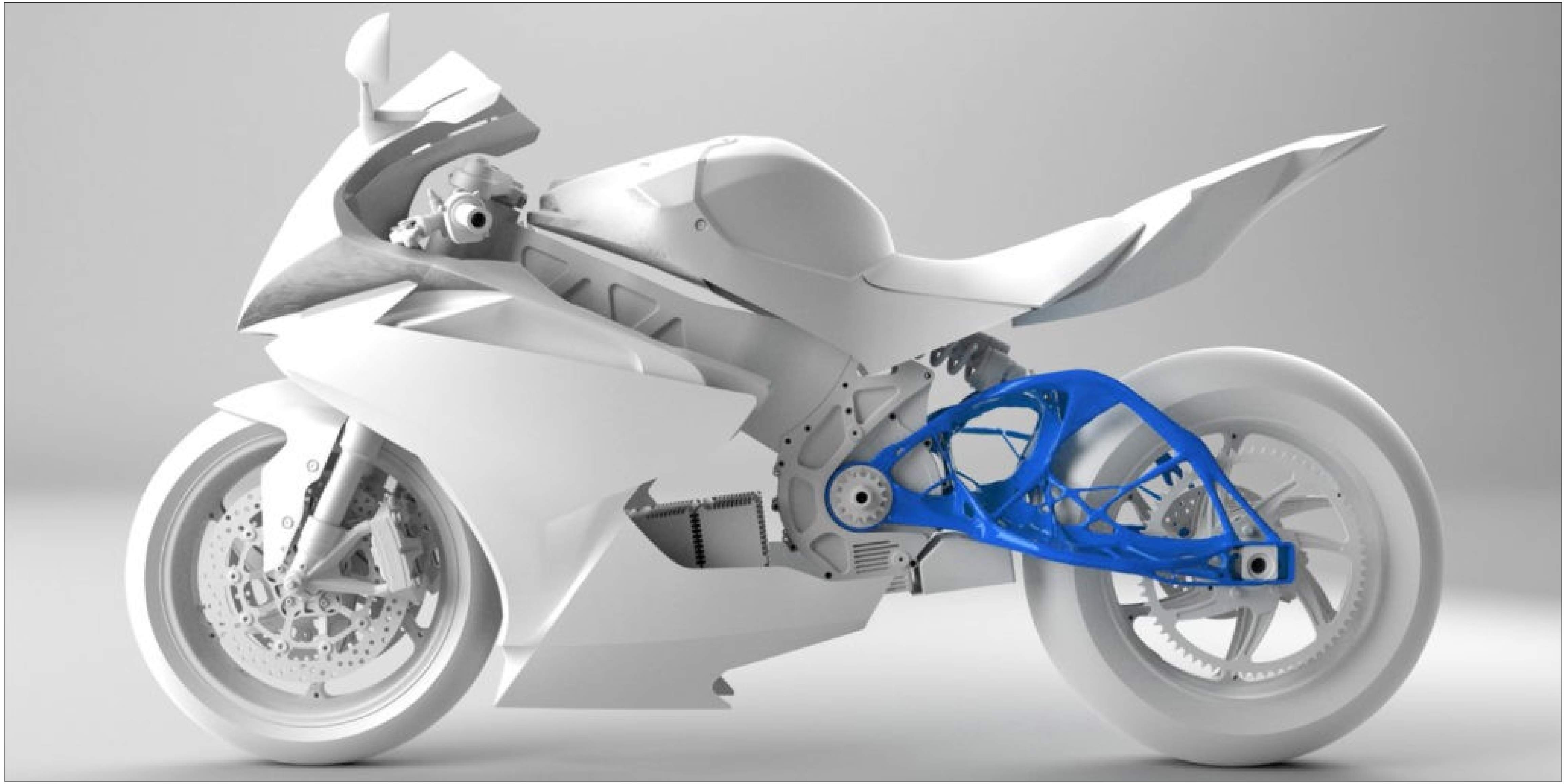
GAN: Everybody dance now



https://arxiv.org/abs/1808.07371
https://www.youtube.com/watch?v=PCBTZh41Ris
August 2018

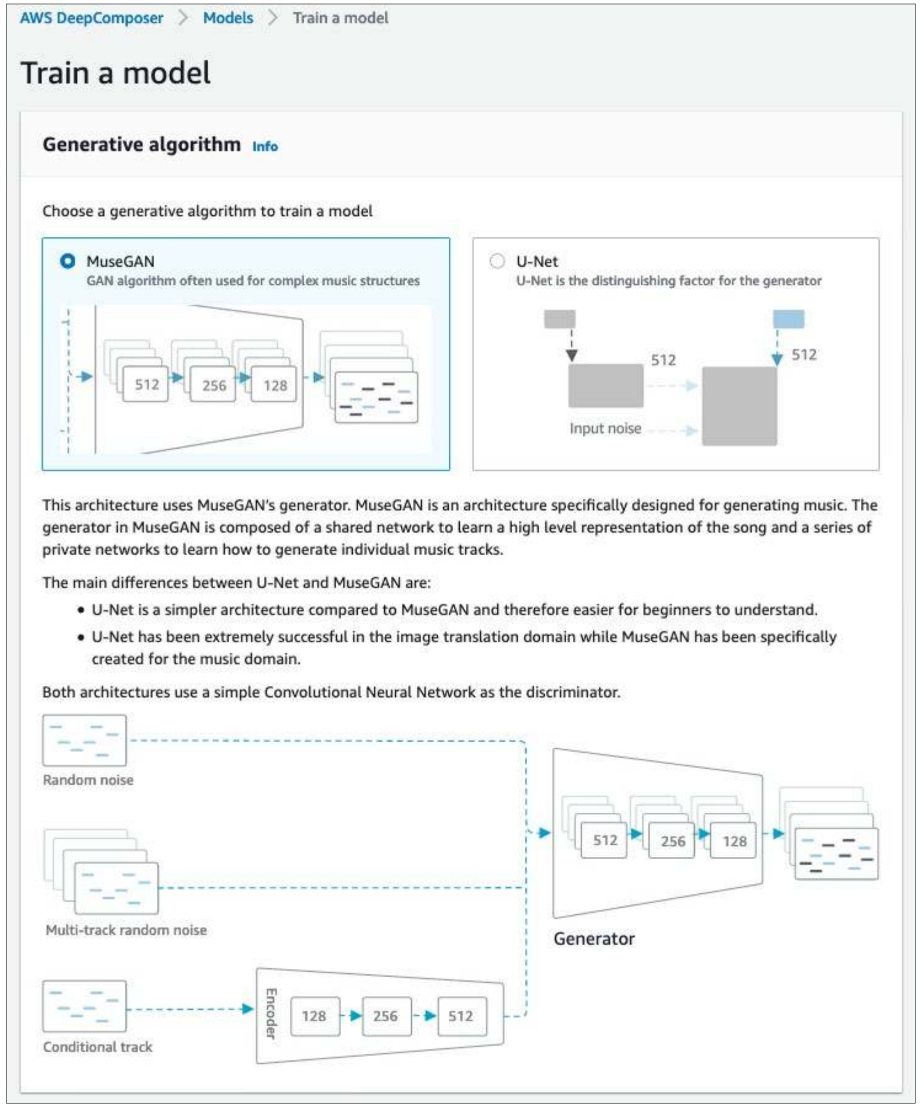
Autodesk - Generative Design

https://www.youtube.com/watch?v=A31A8KDC9S4



AWS DeepComposer

https://aws.amazon.com/deepcomposer
https://github.com/aws-samples/aws-deepcomposer-samples



Train a model

Generative algorithm Info

Choose a generative algorithm to train a model

● MuseGAN
GAN algorithm often used for complex music structures

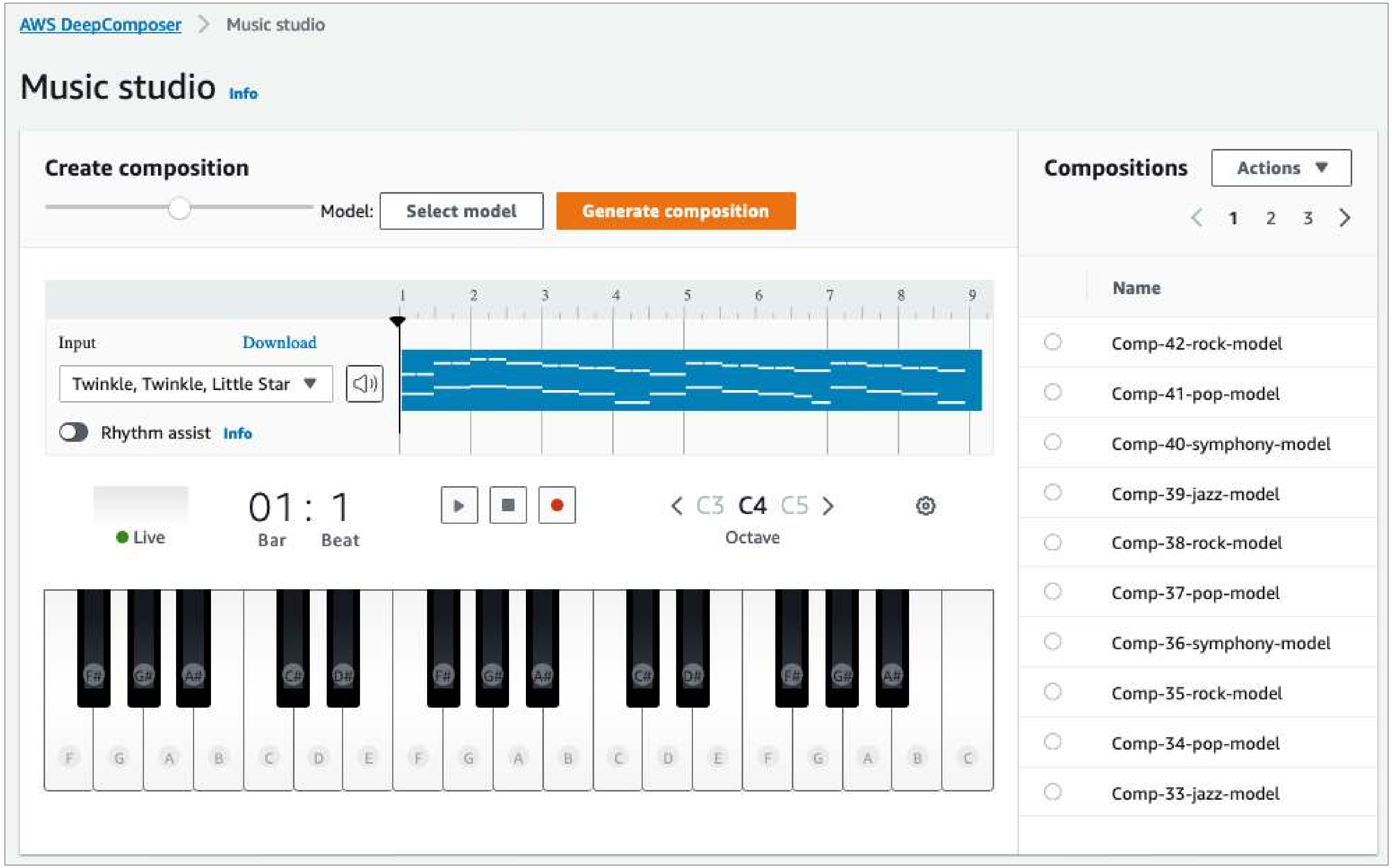
○ U-Net
U-Net is the distinguishing factor for the generator

This architecture uses MuseGAN's generator. MuseGAN is an architecture specifically designed for generating music. The generator in MuseGAN is composed of a shared network to learn a high level representation of the song and a series of private networks to learn how to generate individual music tracks.

The main differences between U-Net and MuseGAN are:

- U-Net is a simpler architecture compared to MuseGAN and therefore easier for beginners to understand.
- U-Net has been extremely successful in the image translation domain while MuseGAN has been specifically created for the music domain.

Both architectures use a simple Convolutional Neural Network as the discriminator.



AWS DeepComposer > Music studio
Select model
Generate composition
Actions ▼
‹ 1 2 3 ›
1 2 3 4 5 6 7 8 9
Download
Info
Live
‹ C3 C4 C5 ›
F# G# A# C# D# F# G# A# C# D# F# G# A#
F G A B C D E F G A B C D E F G A B C

Music studio Info

Create composition

Model: Select model

Input: Twinkle, Twinkle, Little Star

Rhythm assist: off

01 : 1
Bar Beat

Octave: C4

Compositions

	Name
○	Comp-42-rock-model
○	Comp-41-pop-model
○	Comp-40-symphony-model
○	Comp-39-jazz-model
○	Comp-38-rock-model
○	Comp-37-pop-model
○	Comp-36-symphony-model
○	Comp-35-rock-model
○	Comp-34-pop-model
○	Comp-33-jazz-model

DEV DAY

Demo

DEV DAY

Getting started

Resources

https://aws.training/machinelearning
https://deeplearning.ai
https://fast.ai

http://www.deeplearningbook.org/
https://d2l.ai/

https://gitlab.com/juliensimon/dlnotebooks

DEV DAY

Thank you!