

# Machine learning

## Lecture 10

source

destination



The StyleGAN creates spectacular synthesized images.

Generative Adversary Networks GAN

Marcin Wolter, *IFJ PAN*, 12 January 2021

# Your projects

- Any ideas? Or maybe some projects already prepared?
- If you have a proposal for project we can discuss it now.



# Generative Adversarial Nets (GANs)

- GANs were introduced by Ian Goodfellow and others in 2014. Yann LeCun called adversarial training “the most interesting idea in the last 10 years in ML.”  
<https://arxiv.org/abs/1406.2661>
- GANs can learn to mimic any distribution of data. They can be taught to create worlds similar to our own in any domain: images, music, speech, prose. They are robot artists!



# Next Rembrandt project



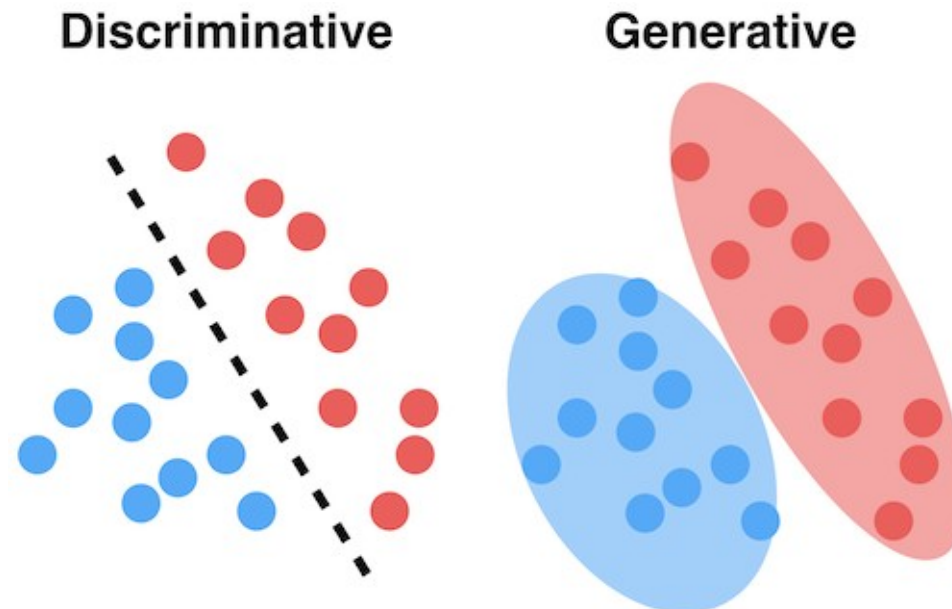
- Create automatically a new Rembrandt portrait based on a collection of Rembrandt works:

<https://www.nextrembrandt.com/>

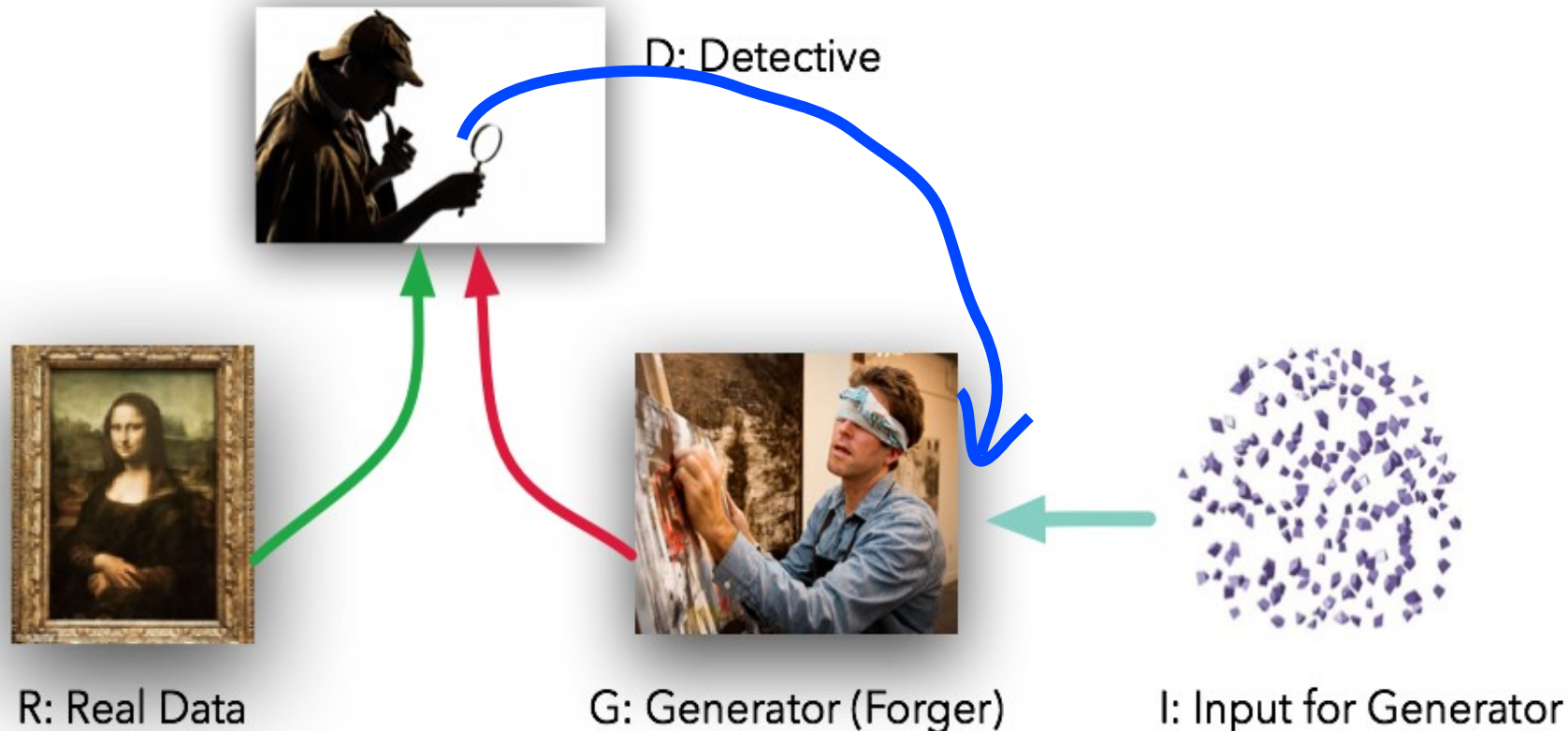


# How do GANs work?

- **Discriminative algorithms** - classify input data; given the features, they predict a label or category to which that data belongs (*signal* or *background*)
- **Generative algorithms** – do the opposite, assuming the event is a *signal*, how likely are these features?
- A way to distinguish discriminative from generative algorithms is:
  - **Discriminative models** learn the boundary between classes
  - **Generative models** model the distribution of individual classes



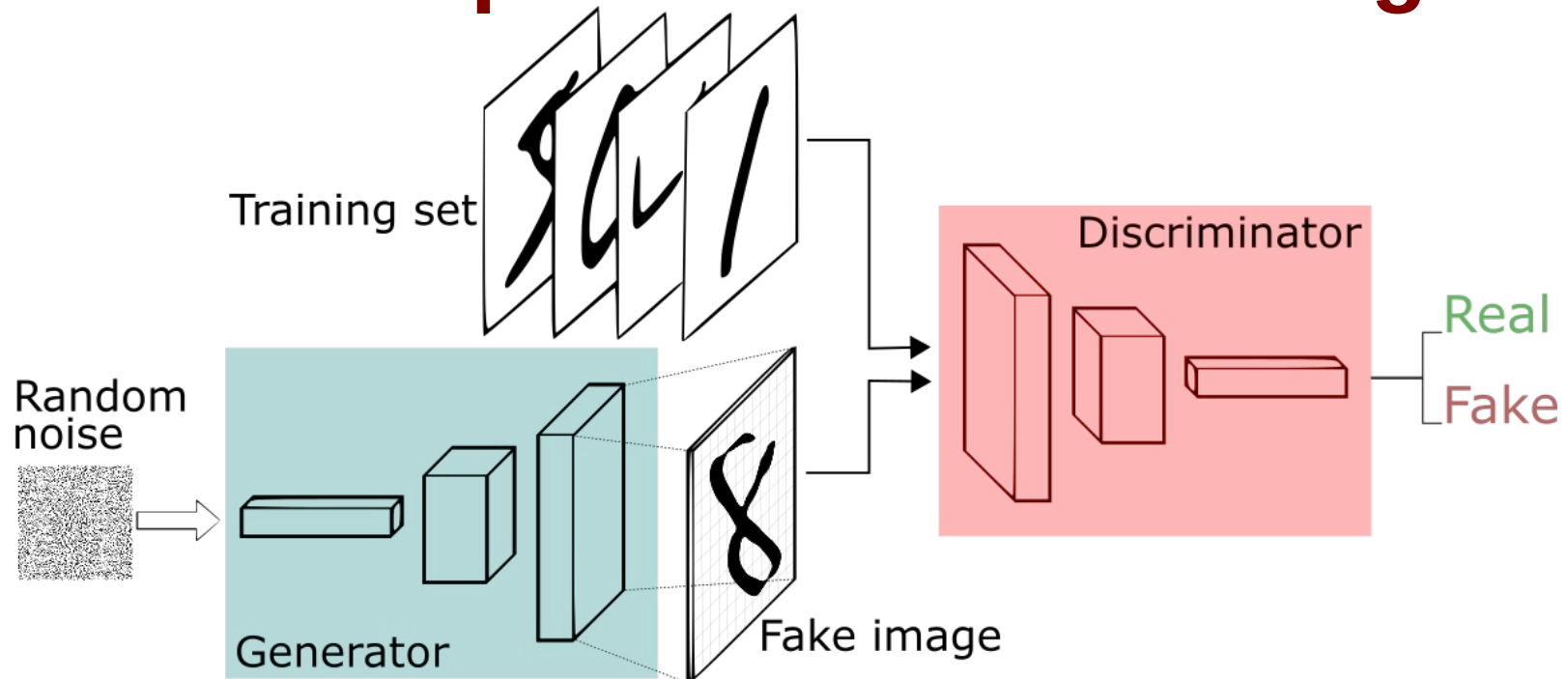
# Blind forger and detective



**The forger has never seen Mona Lisa, but gets the judgments of detective and tries to fool him (i.e. paint something that looks like Mona Lisa).**

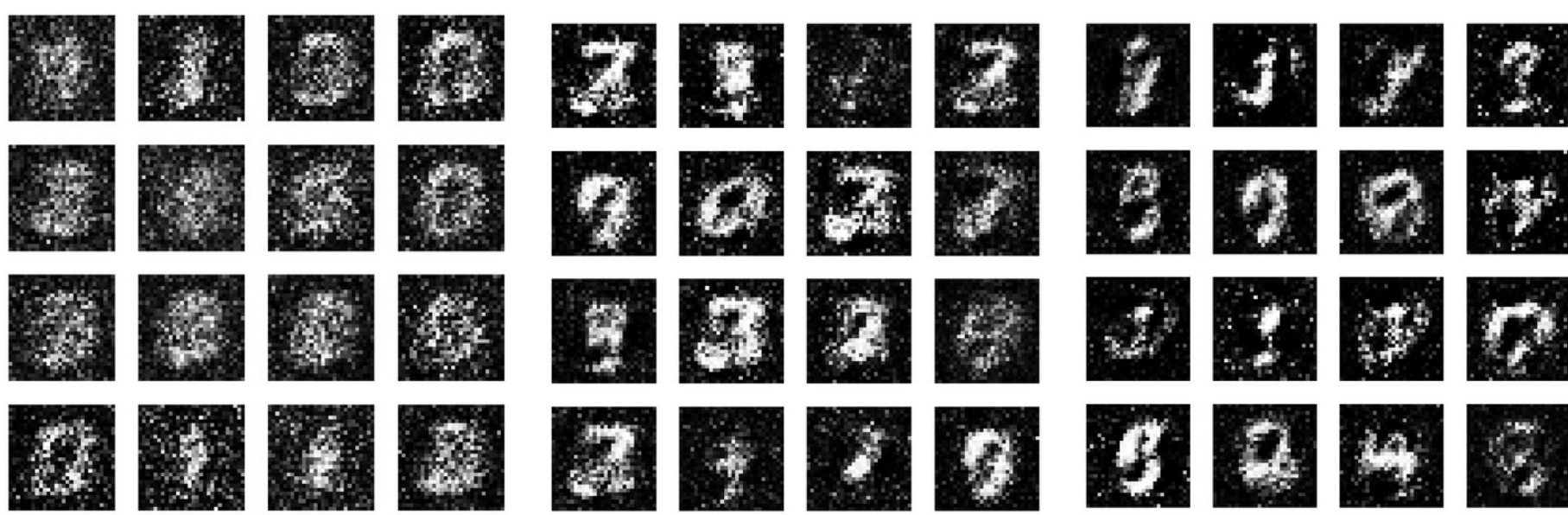
They both (forger and detective) have to train in parallel (important), since if detective is too clever the forger will never paint anything acceptable.

# GANs example – hand-written digits



- **Training set** – MNIST: hand-written digits supplied by US post.
- **Discriminator** – convolutional neural network labeling images as real or fake.
- **Generator** - inverse convolutional network (while a standard convolutional classifier takes an image and downsamples it to produce a probability, the generator takes a vector of random noise and upsamples it to an image).

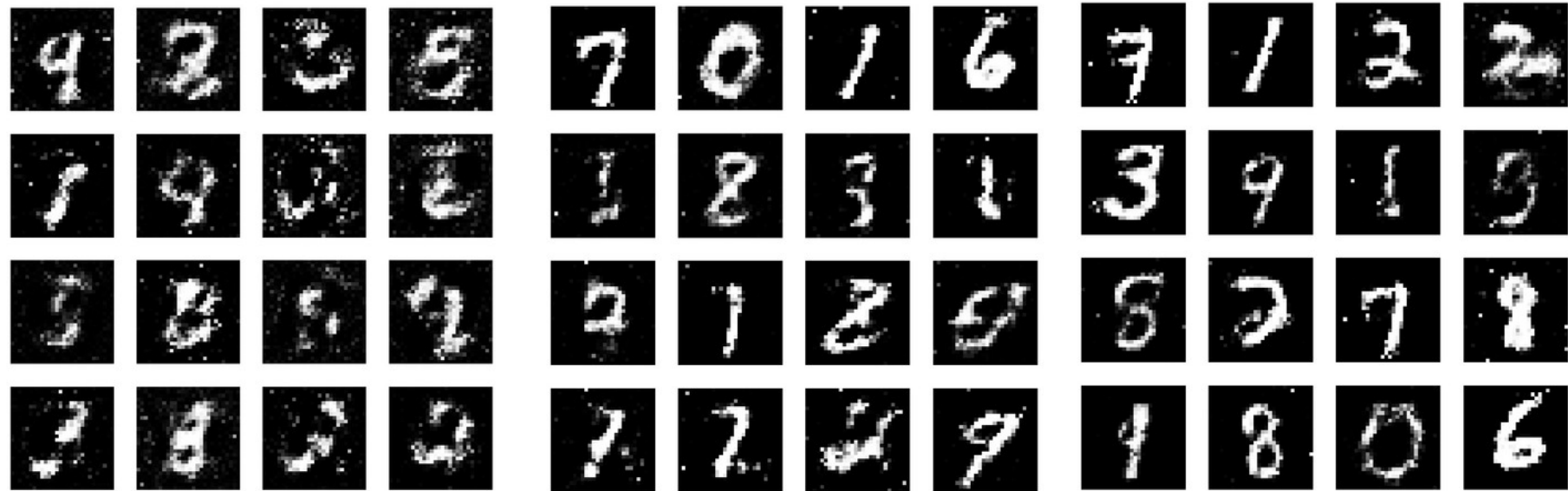
*Implementation: Python code using Keras interface and TensorFlow backend.*



400 cycles

800 cycles

1200 cycles



2400 cycles

8000 cycles

19900 cycles

Each cycle digits look more and more realistic.

Updated code from the net:

[https://github.com/marcinwolter/MachineLearning2020/blob/main/gan\\_generate\\_letters.ipynb](https://github.com/marcinwolter/MachineLearning2020/blob/main/gan_generate_letters.ipynb)

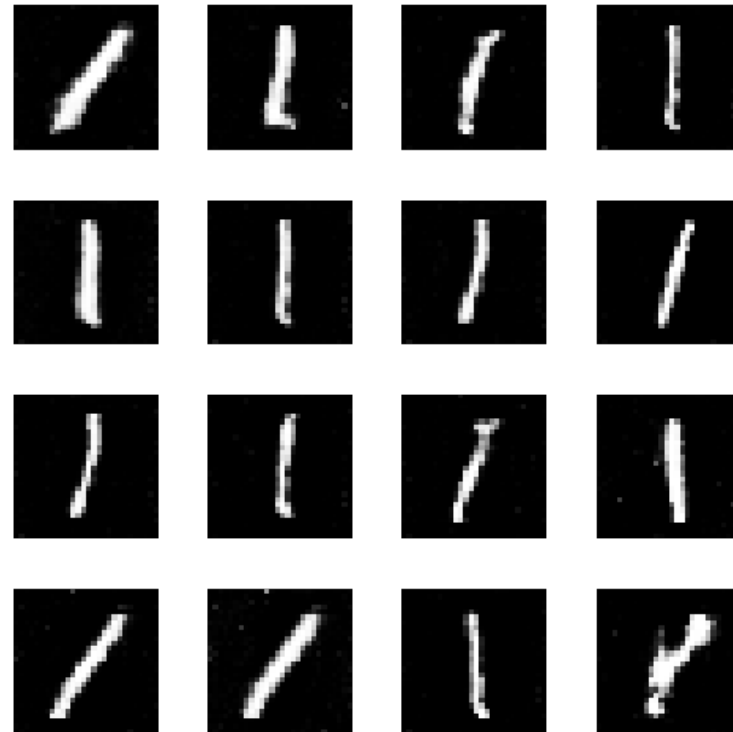


# Training a GAN

- It is important to keep balance between generator and discriminator – none of them should be too smart!
- If in the previous example we modify discriminator to make it much better (ConvDNN, many layers) the GAN starts to generate just one digit.

[https://github.com/marcinwolter/MachineLearning2020/blob/master/gan\\_generate\\_letters\\_failing.ipynb](https://github.com/marcinwolter/MachineLearning2020/blob/master/gan_generate_letters_failing.ipynb)

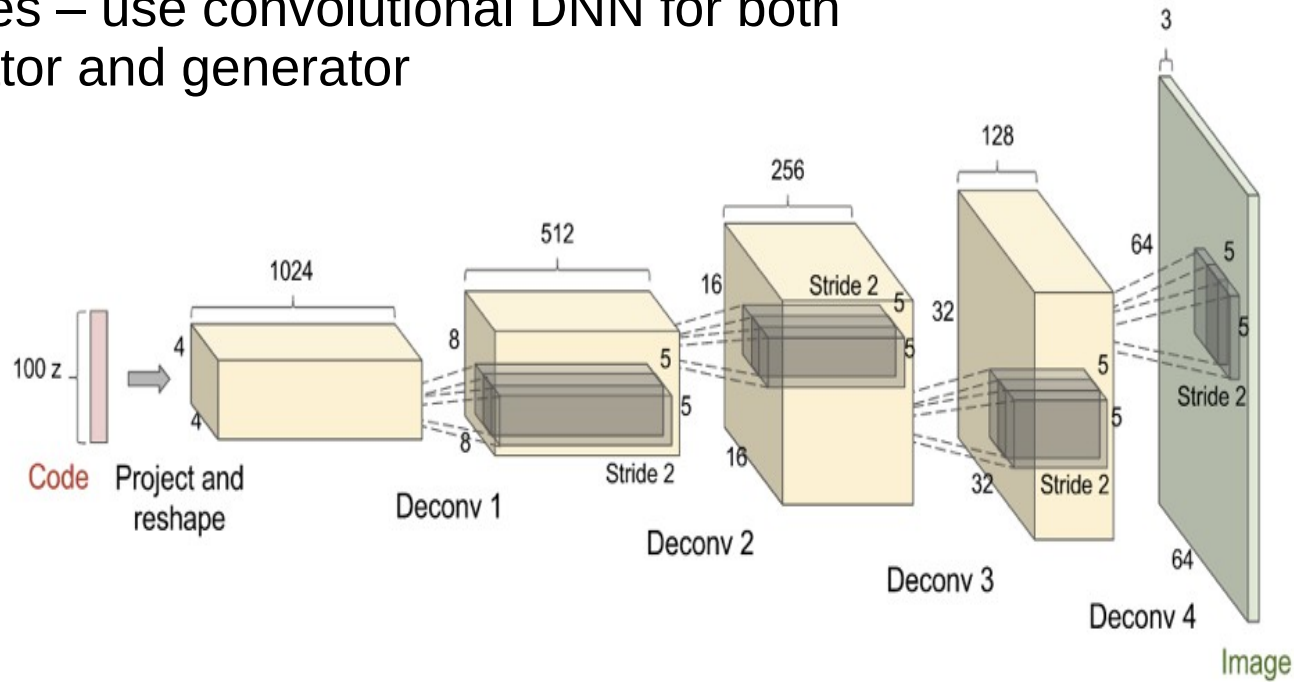
**FAILING to generate different digits**



# Deconvolutional GANs (DCGAN) (Radford et al., 2015)



For pictures – use convolutional DNN for both discriminator and generator



Goodfellow (2017)

# Deconvolutional GANs (DCGAN)

(Radford et al., 2015)

Goodfellow (2017)

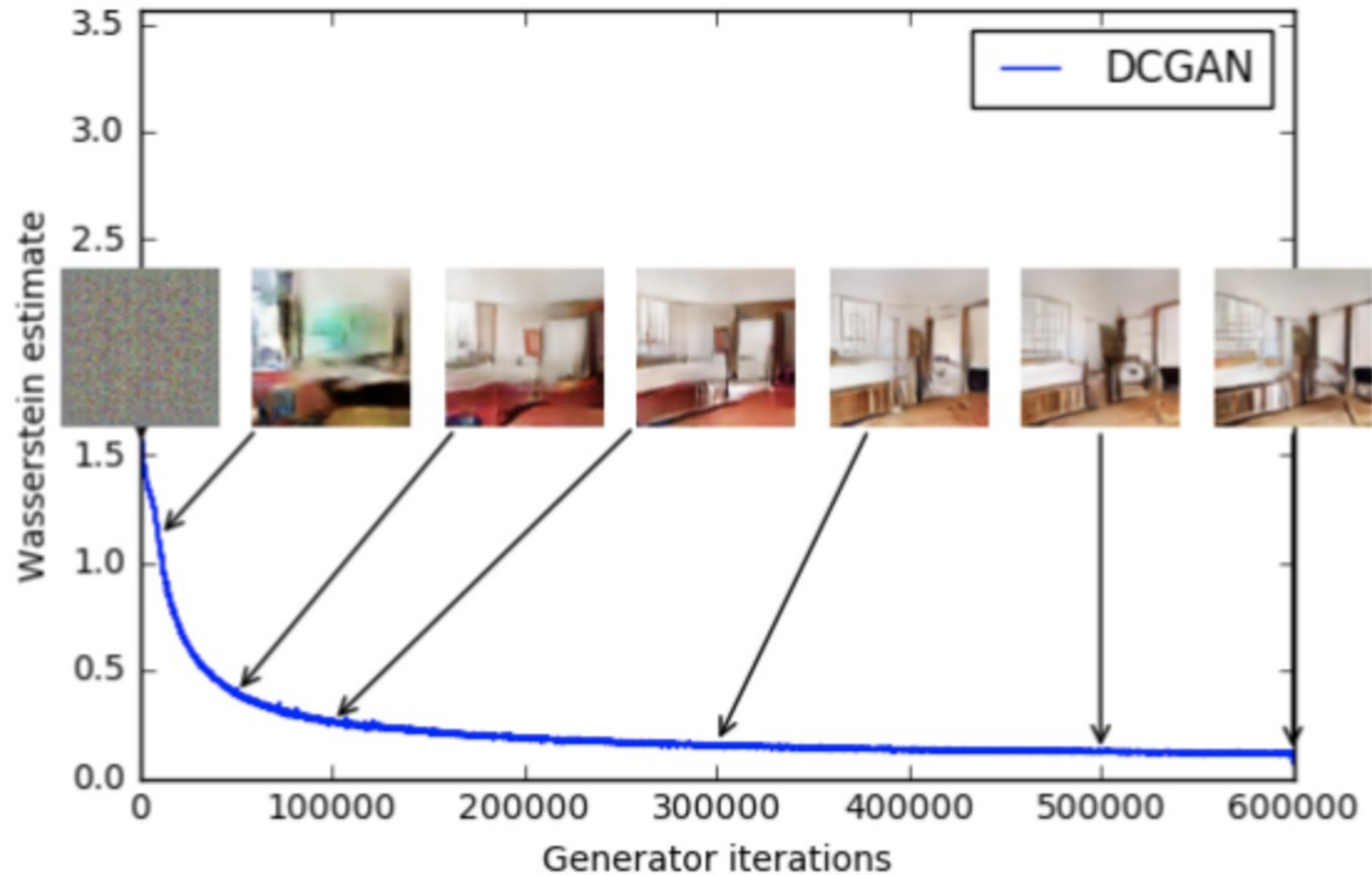


Based on LSUN  
data set

- 10 scene categories
- 20 object categories

[ArXiv](#)  
[1506.03365](#)

# DCGAN training



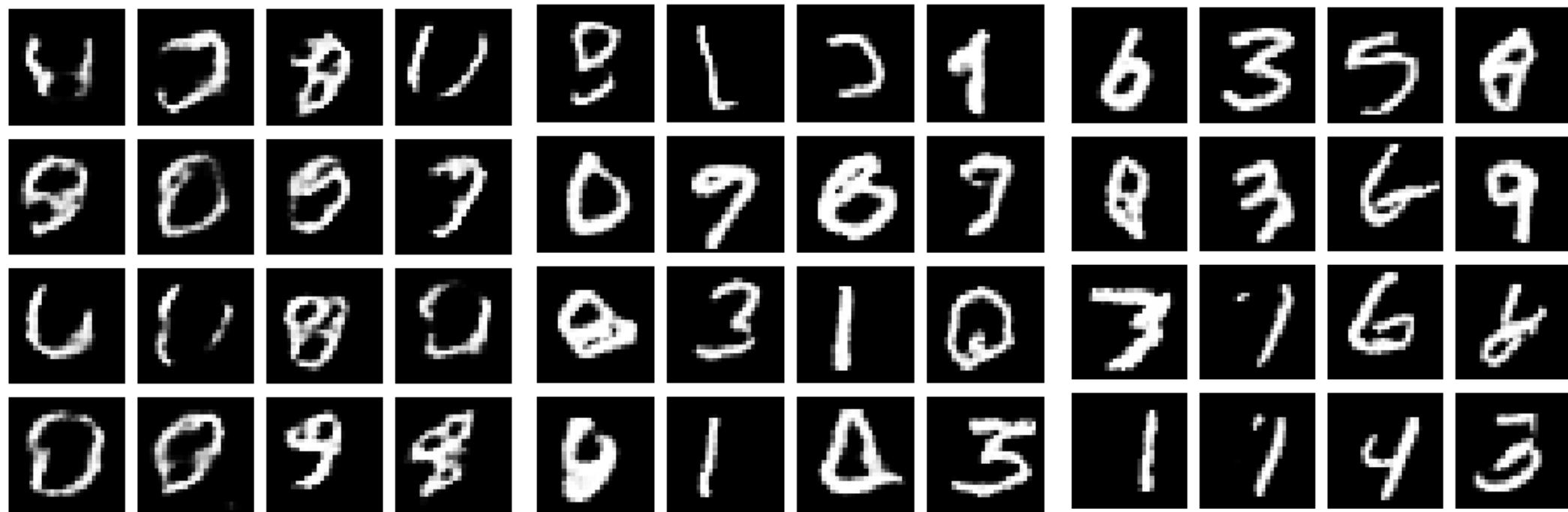
An example of a DC-GAN training and improving its quality "Improving the GAN training process"



# DCGAN example

Generating hand-written letters again:

[https://github.com/marcinwolter/MachineLearnin2019/blob/master/dcgan\\_cn\\_n\\_mnist.ipynb](https://github.com/marcinwolter/MachineLearnin2019/blob/master/dcgan_cn_n_mnist.ipynb)

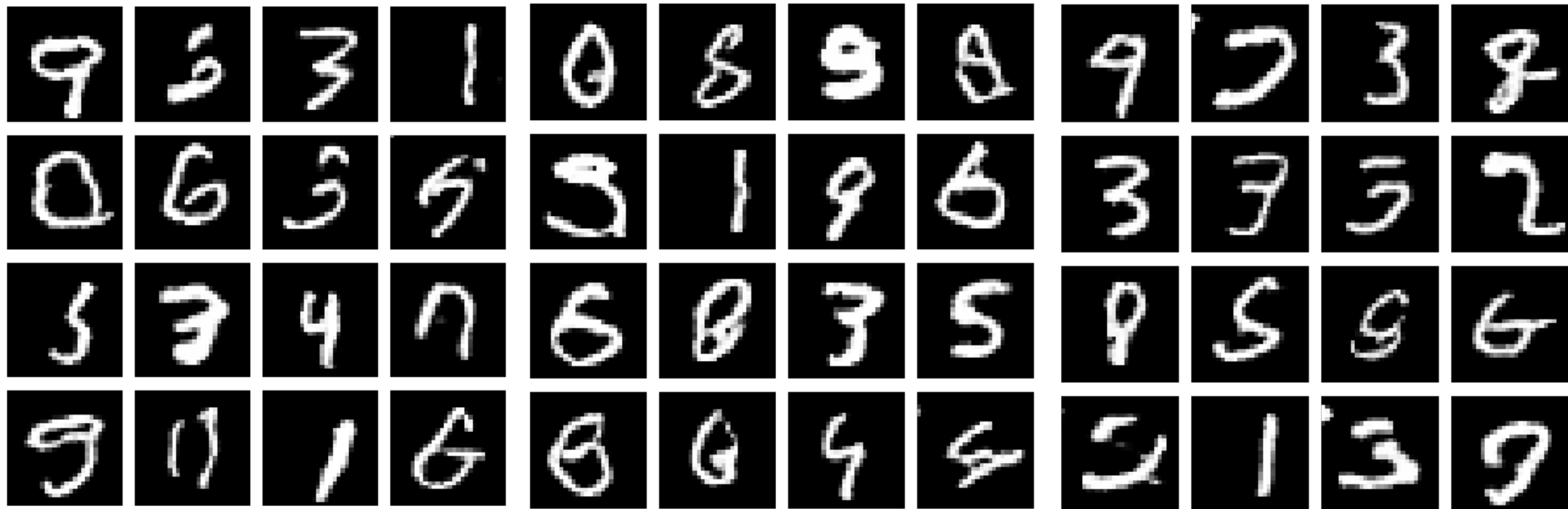


6 epochs

30 epochs

50 epochs

# DCGAN example cont.



55 epochs

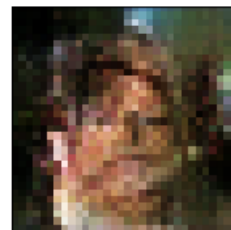
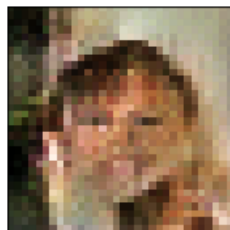
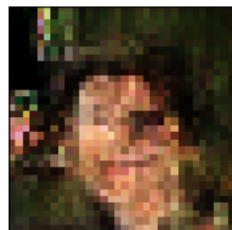
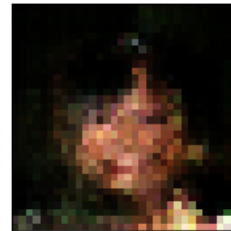
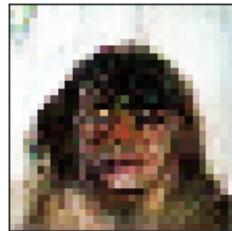
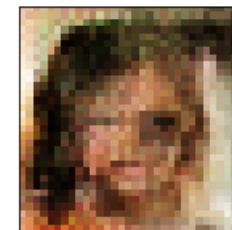
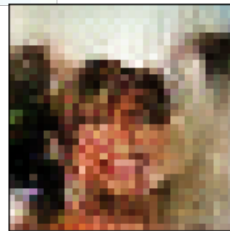
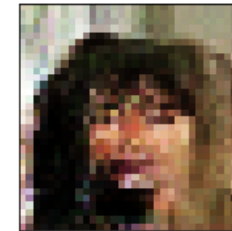
75 epochs

100 epochs

# Generating faces

- Use DCGAN to generate faces using the CELEBA dataset.
- [https://github.com/marcinwolter/MachineLearning2020/blob/main/dcgan\\_overriding\\_train\\_step.ipynb](https://github.com/marcinwolter/MachineLearning2020/blob/main/dcgan_overriding_train_step.ipynb)
- Instead of letters we generate faces (resolution highly reduced to speed-up the generation process)

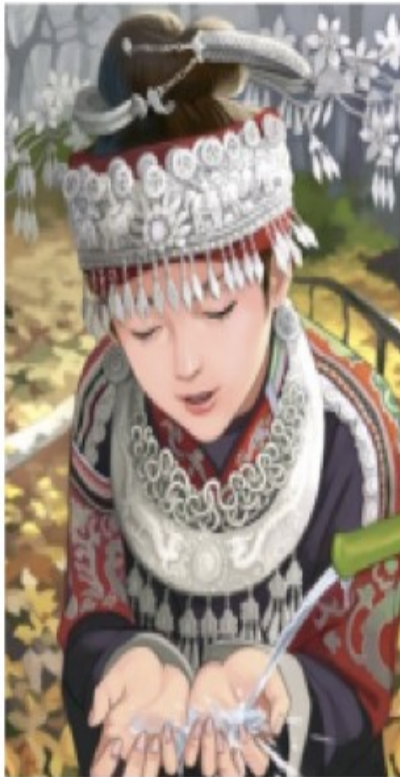
Original



After 20 epochs (we need much more epochs to generate decent faces).

# Beyond Labels: Providing Images as Input to Generator: Image Super-Resolution (Ledig et al., 2016)

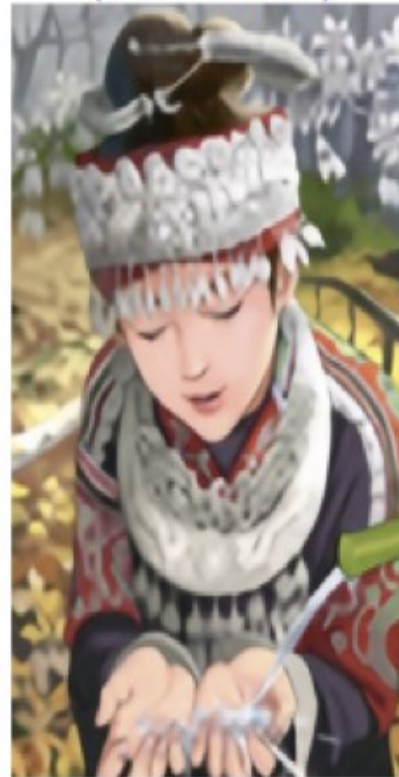
original



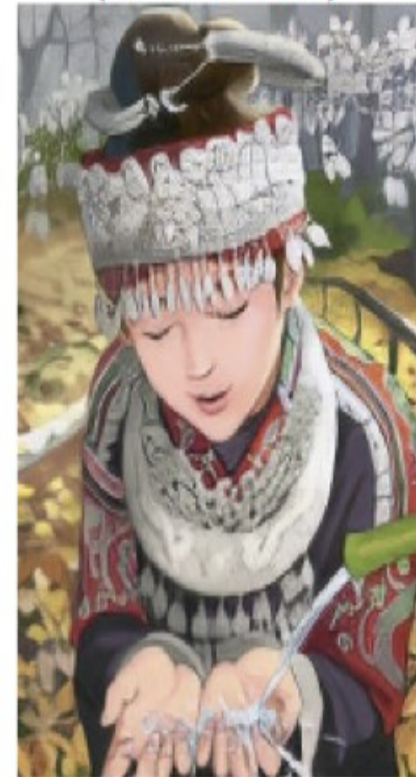
bicubic  
(21.59dB/0.6423)



SRResNet  
(23.44dB/0.7777)



SRGAN  
(20.34dB/0.6562)



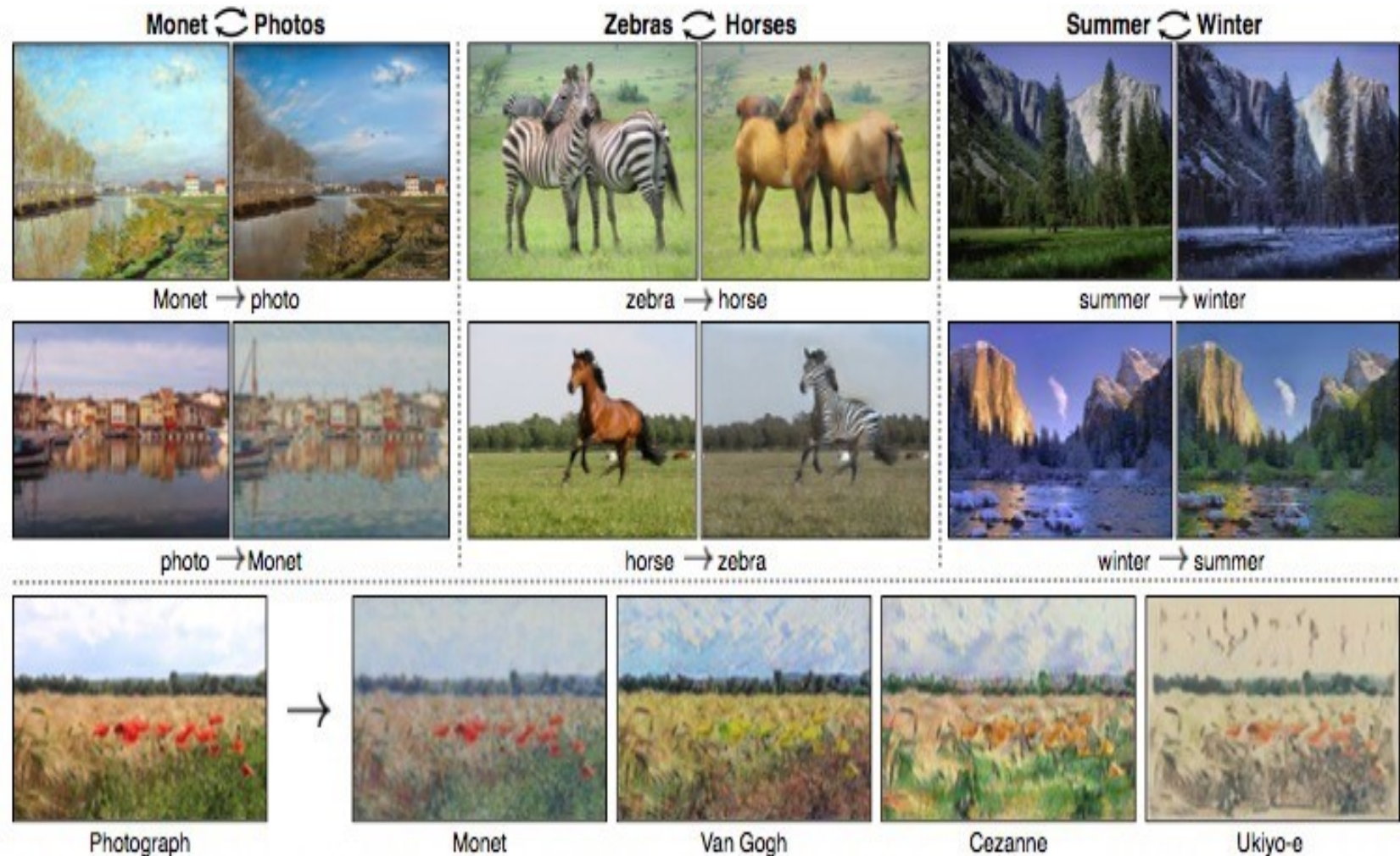


# Cycle GANs

(Zhu et al., 2017; arXiv:1703:10593v2 [cs.CV])

Given two image collections

- algorithm learns to translate an image from one collection to the other
- does not require correspondence between images

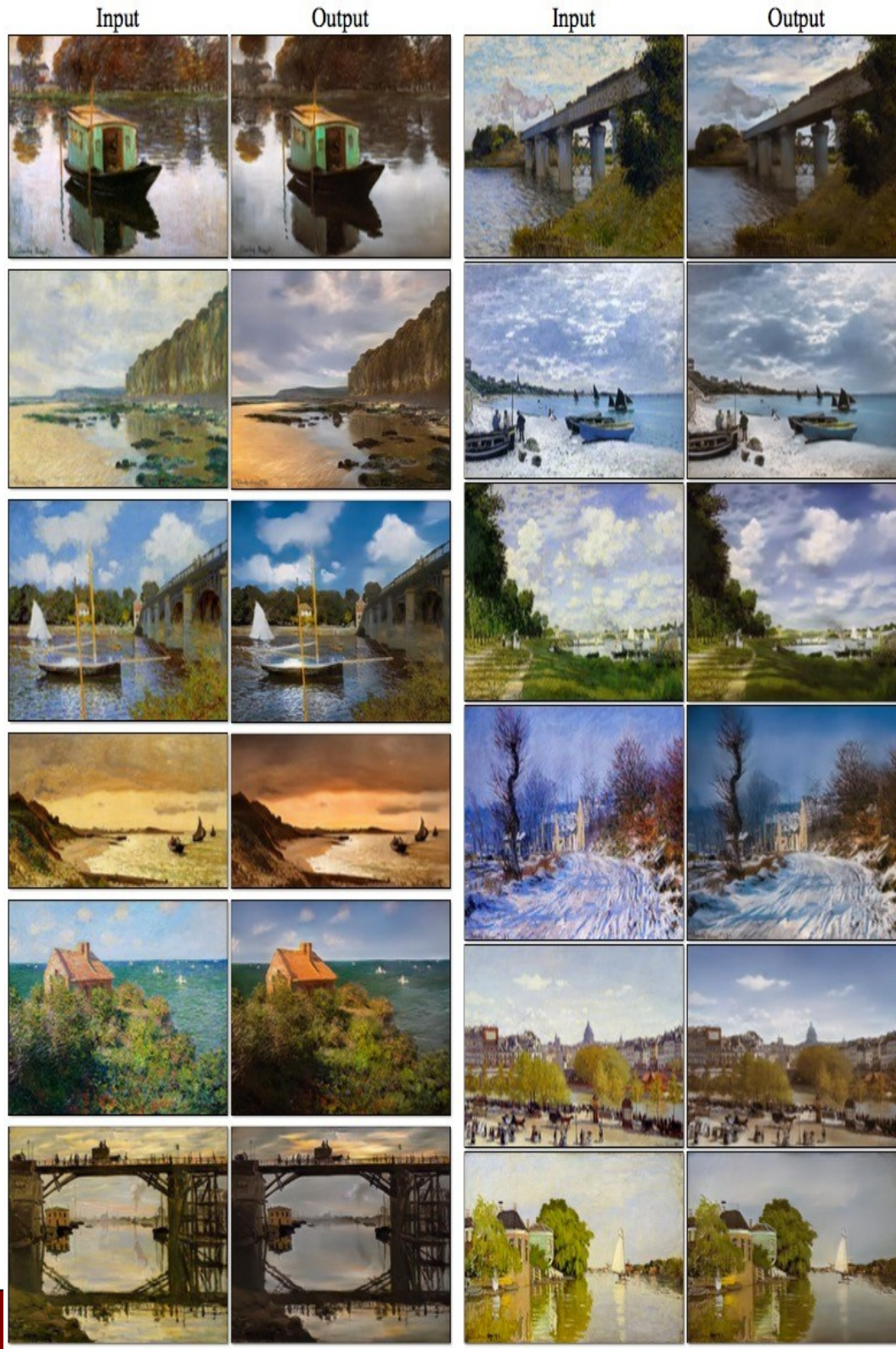


# Photos to paintings





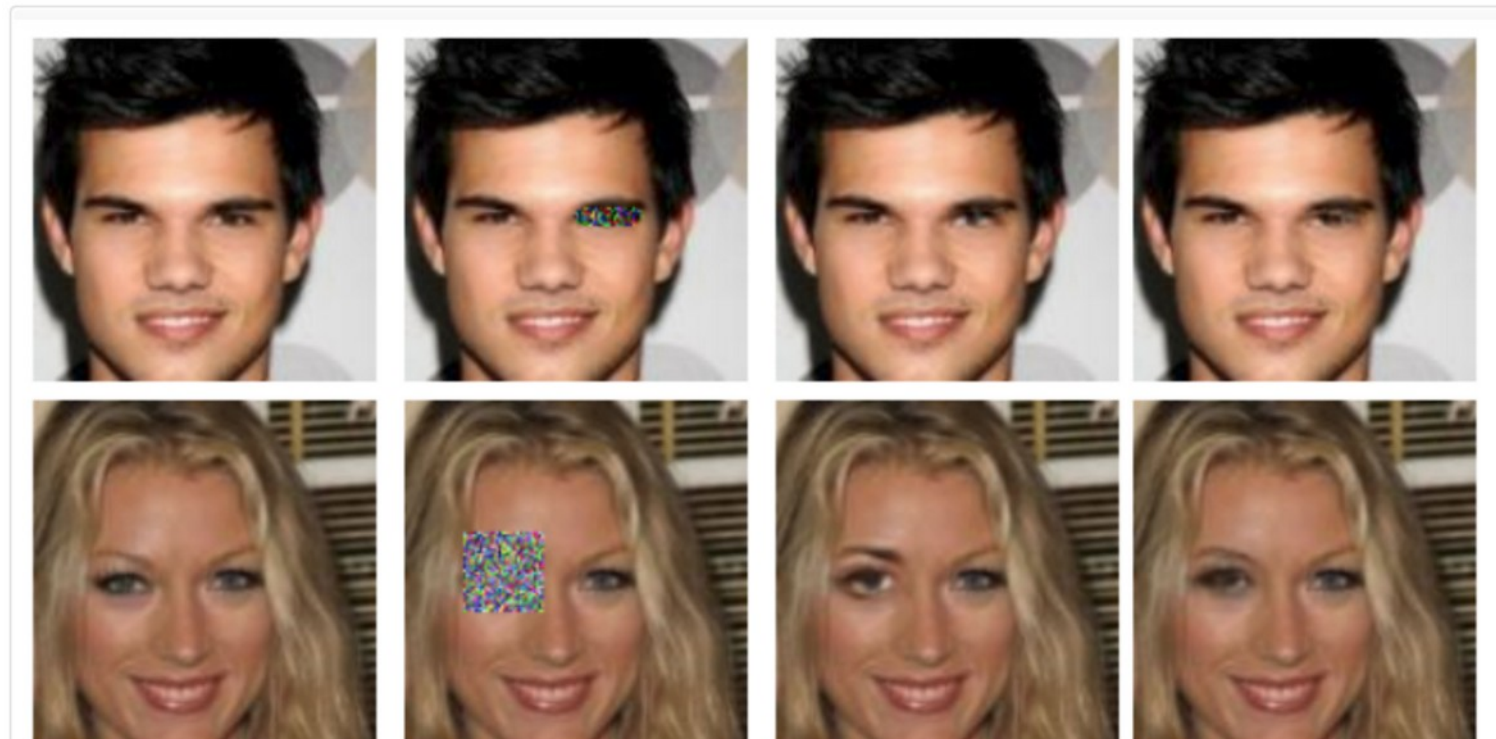
# Paintings to photos



# Applications of GAN

- In science - speed up generation of simulated data – new simulated look similar to the once already simulated / collected data.
- Overview of GAN applications:

<https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/>

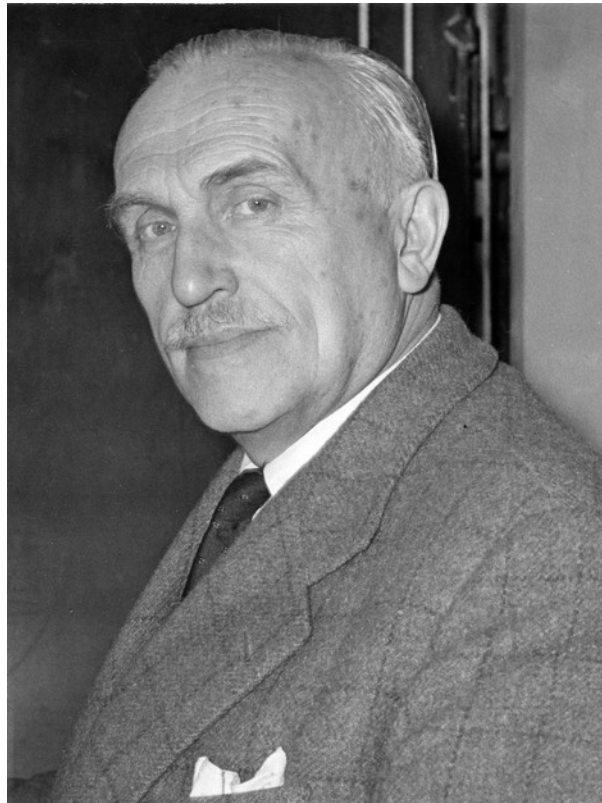


Example of GAN Reconstructed Photographs of Faces Taken from Generative Face Completion, 2017.

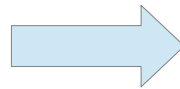
# Add colors to pictures - DeOldify

- <https://github.com/jantic/DeOldify>

open code project, adds color to pictures and to the movies. You can run it by yourself on Google Colab



Prof. Henryk Niewodniczański  
Founder of our Institute



# Make money with GAN!!!



Sold for a reported **\$432,500** at an auction!!!

"Edmond de Belamy" is part of a fictitious family created by a "generative adversarial network," of which there's ten other paintings. "Edmond" is one of the most striking of the paintings, and will likely become an important part of art history going forward thanks to its huge selling price. The generator behind the painting created new portraits based on 15,000 from the last 600 years, taking existing art and crafting something wholly original and quite alien.

# Next REMBRANDT

- Create a new Rembrandt painting based on all of his works  
<https://www.nextrembrandt.com/>



3D print





# Conclusions

- Many new methods were developed recently.
- Machine Learning approach becomes to be used not only for classification, but also for other tasks.
- Each month new application appear!
- The development is driven by AI applications (image recognition, autonomous cars etc). But the physics community can profit!
- More and more advanced ML techniques have application in HEP. Try to find a new one!

Artificial  
intelligence  
is no match for  
natural  
stupidity.