deep imaging

Machine Learning and Imaging

BME 548L
Roarke Horstmeyer

Generative models, adversarial examples and GANs

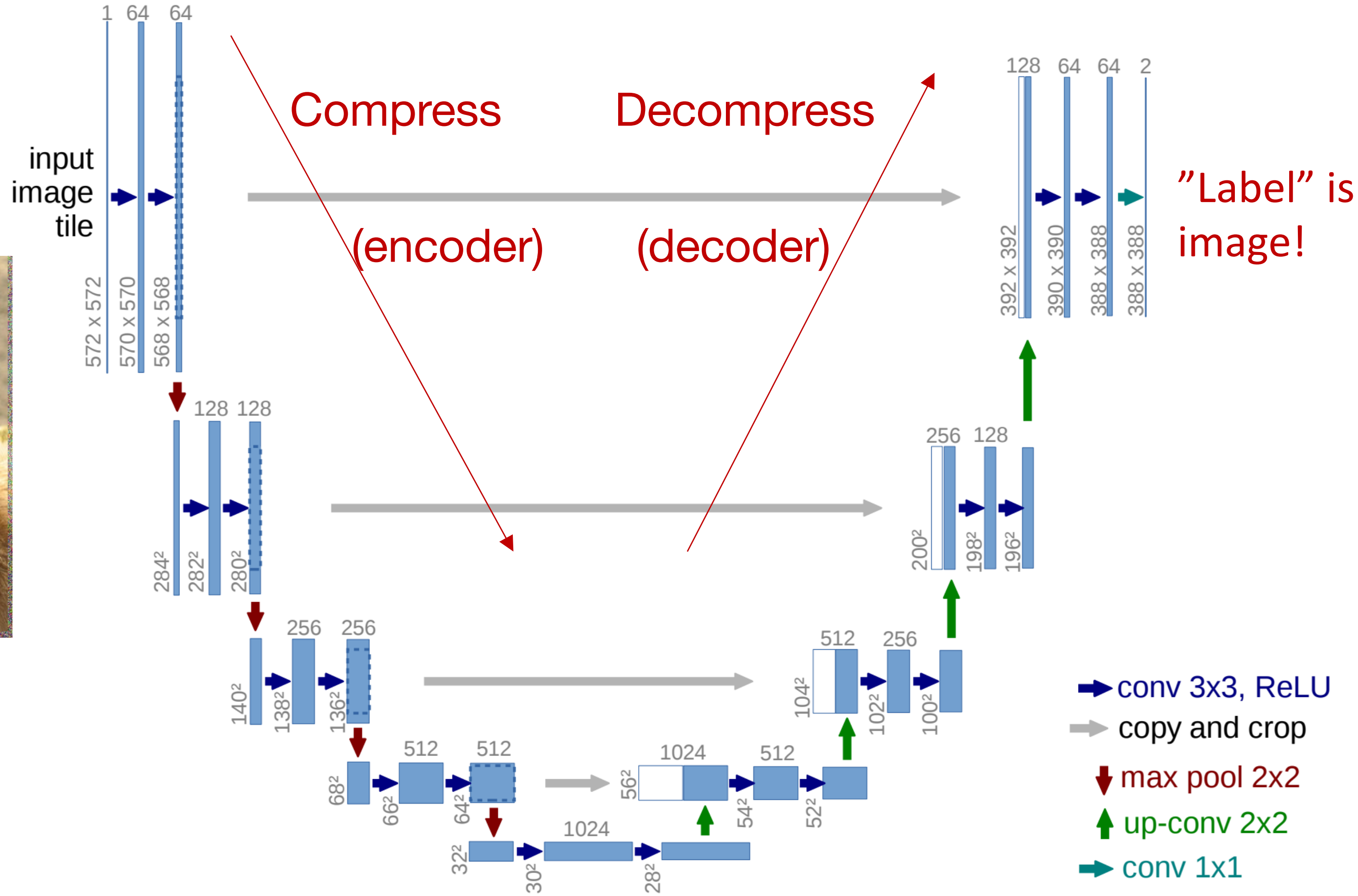
Announcements

- Homework #4 due tonight
- Homework #5 released tonight and will be due **Wednesday April 24 at 11:59pm**
- Final project details have been updated, please look at instructions here: https://deepimaging.github.io/data/BME548_Project_Instructions.pdf

Review from last class: Autoencoders

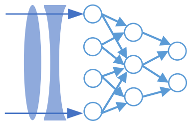
U-Net Architecture

Input image

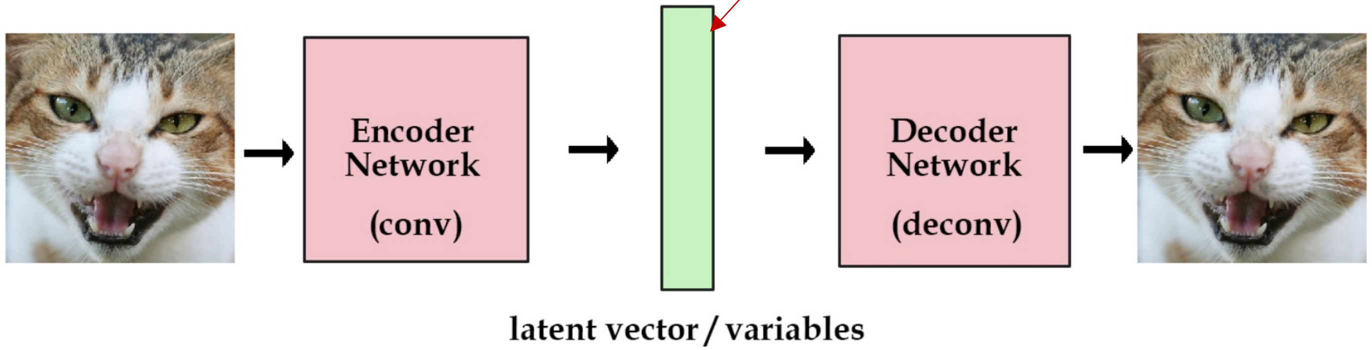


Input image

Example: Variational Autoencoder (VAE)



Force this vector to follow a Gaussian PDF



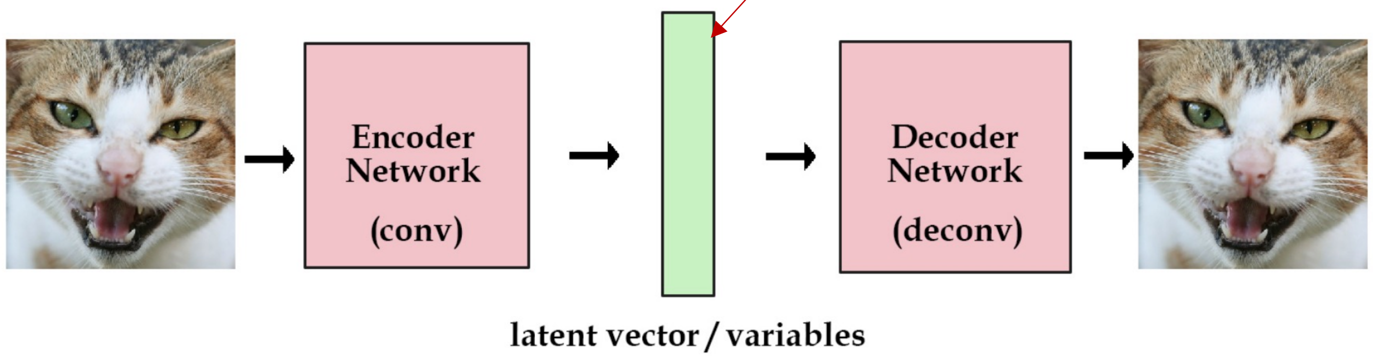
- Good *generative model*
- Have a clean probability distribution to select from to generate new examples

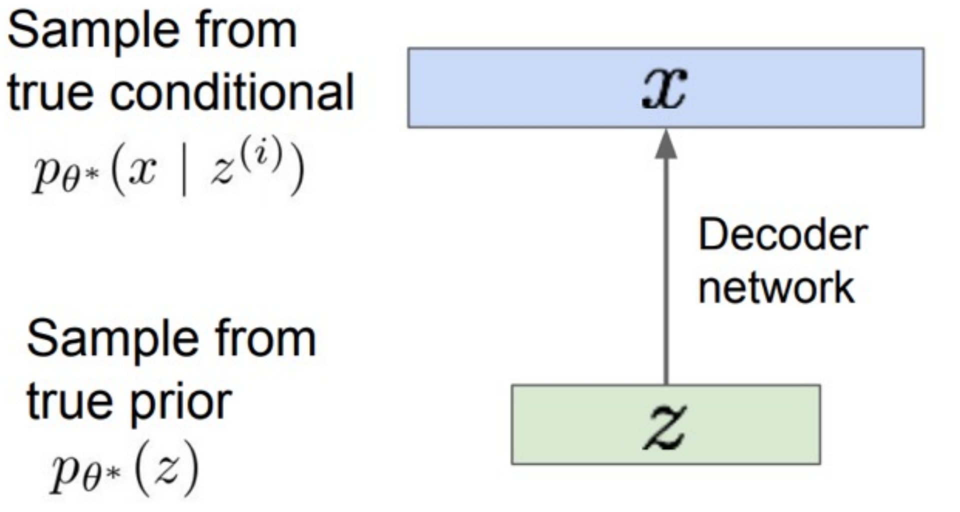
Minimize (KL) distance between latent vector and Gaussian normal

Example: Variational Autoencoder (VAE)

deep imaging

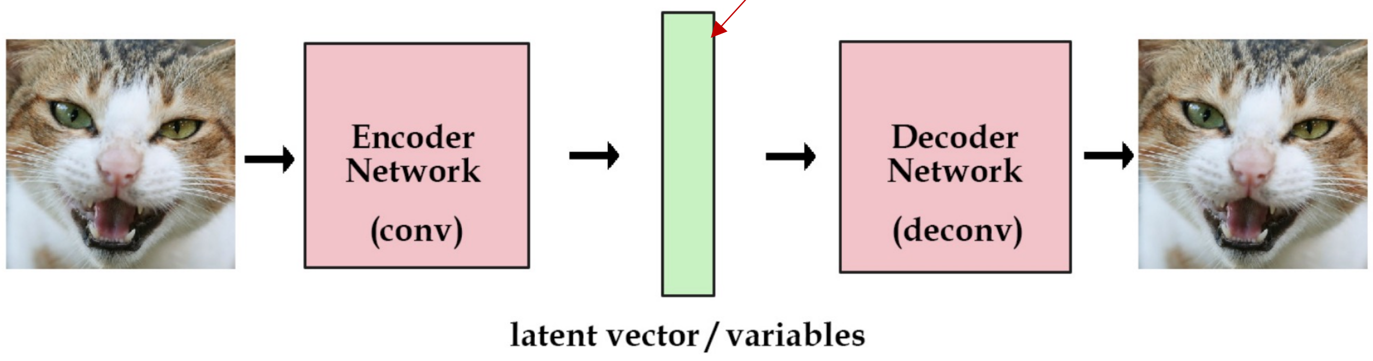
Force this vector to follow a Gaussian PDF





Example: Variational Autoencoder (VAE)

Force this vector to follow a Gaussian PDF

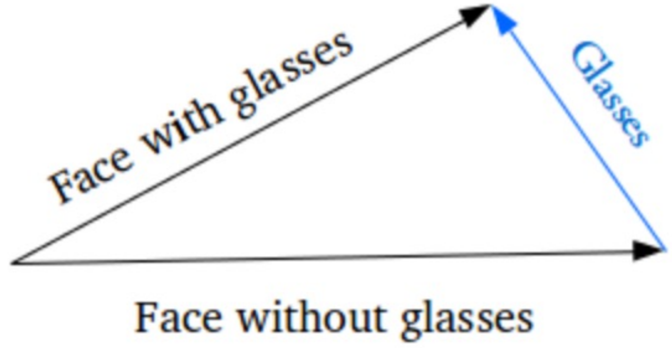


Minimize (KL) distance between latent vector and Gaussian normal

Generative Example (once trained):

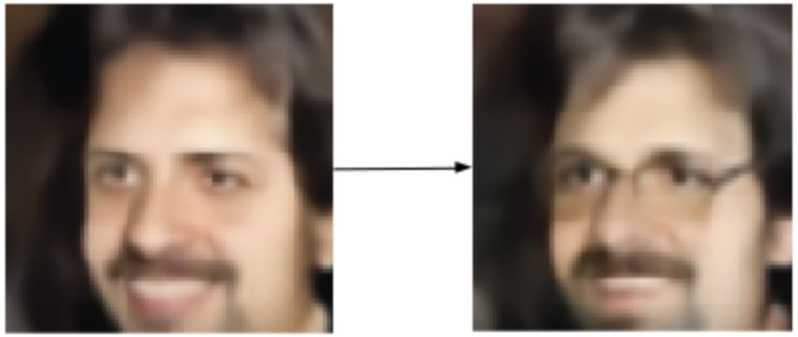
- Encode image with glasses, obtain latent vector PDF $\mathbf{P_g}$
- Encode image without glasses, obtain PDF $\mathbf{P_{ng}}$
- Compute **diff** = $\mathbf{P_g}$- $\mathbf{P_{ng}}$
- Encode new image to obtain $\mathbf{P_{new}}$, add in **diff**
- Decode $\mathbf{P_{new}}$ + **diff** to get guy with glasses!

- With Gaussian PDF, can start to add/subtract latent vector in a normalized vector space



Adding new features to samples

Glasses



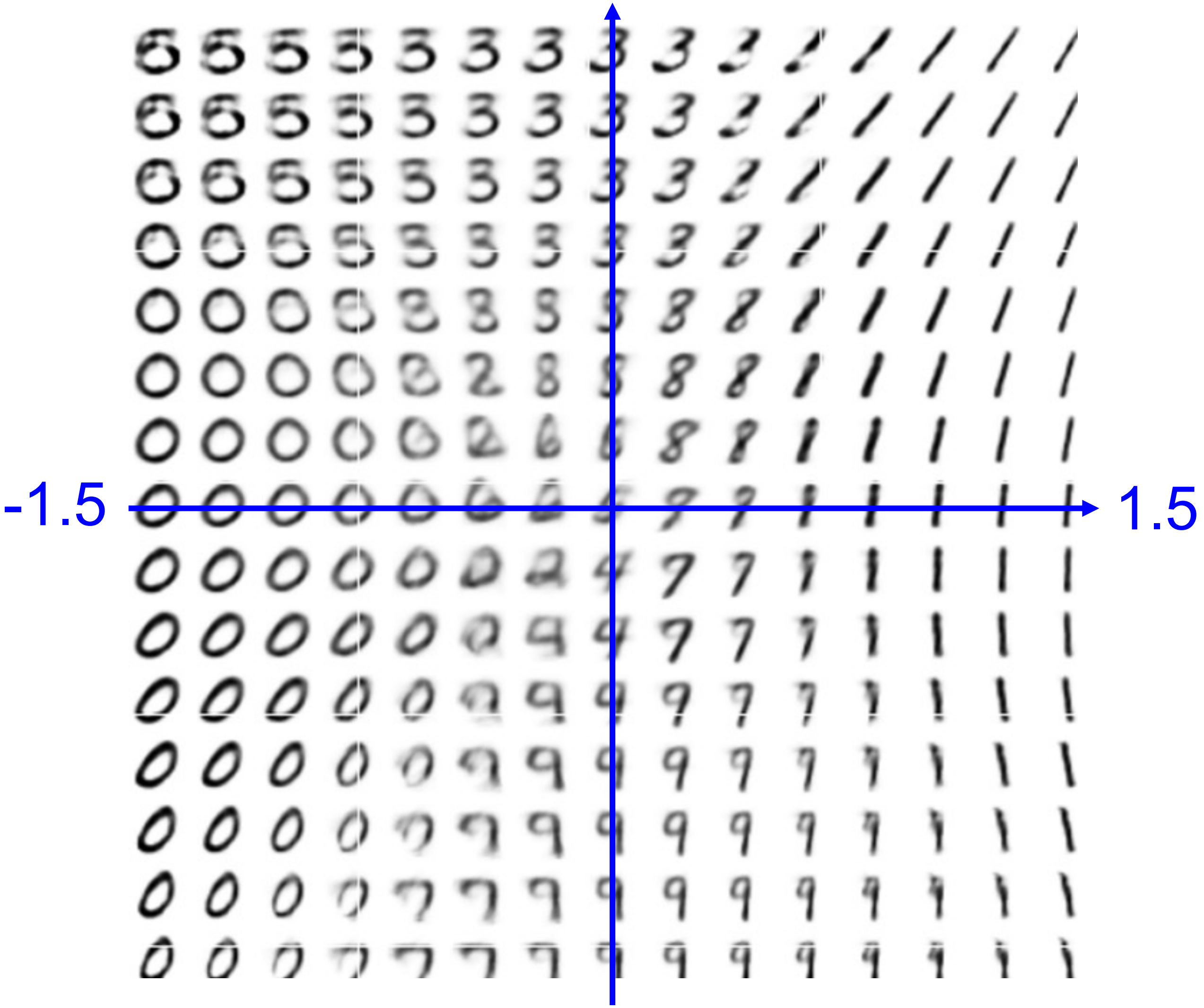
Exploring a specific variation of input data[1]

A very simple example – MNIST digits

See https://deepimaging.github.io/data/Simple_Autoencoder.ipynb

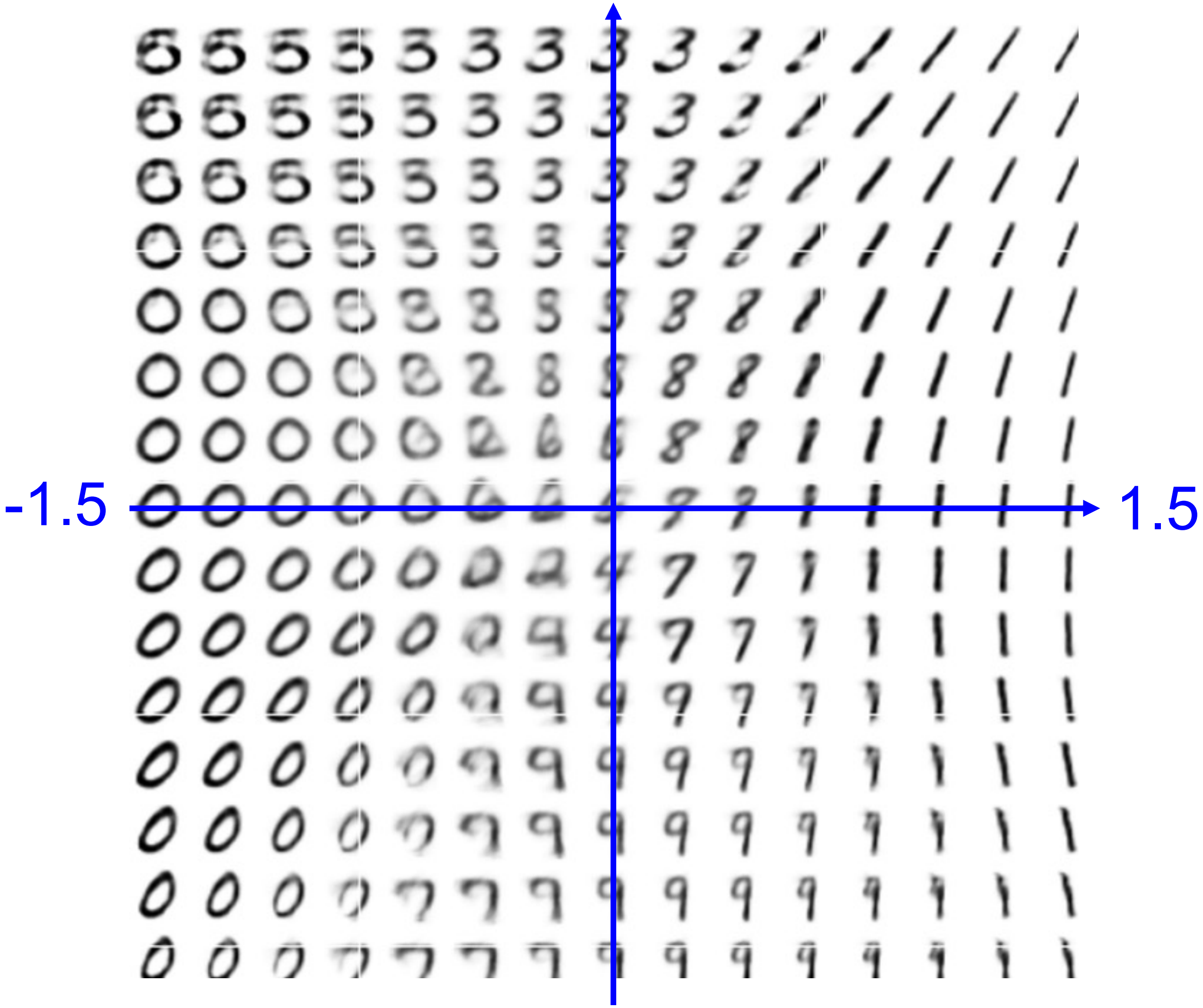
On the way towards generating “fake” images of handwritten digits?

Navigating across the manifold of handwritten digits



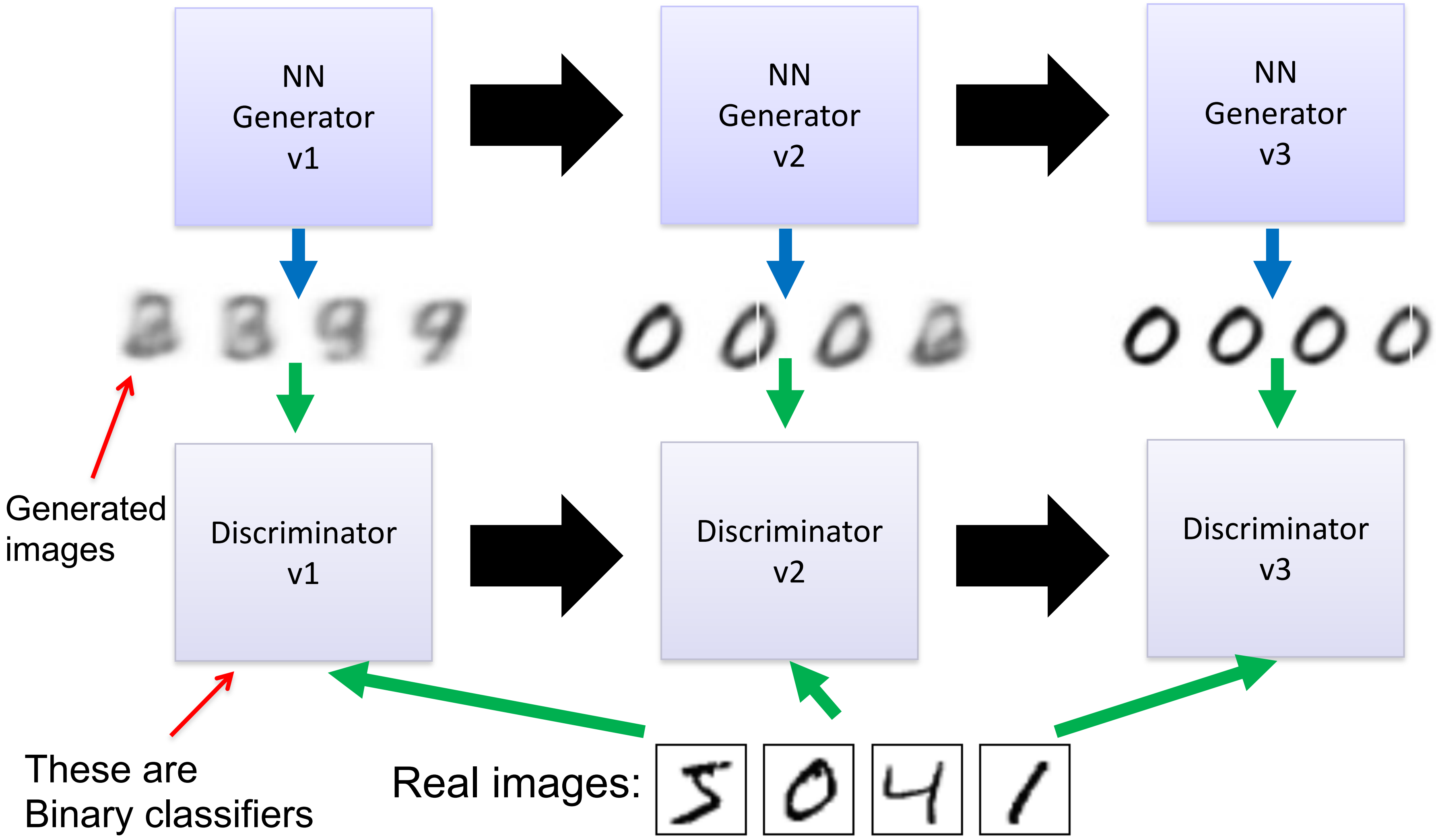
- ”Selection” – starting with some vector in latent space
- “Navigation” – movement along vectors
- Selection and Navigation within the latent space produces “visually interesting” results

Navigating across the manifold of handwritten digits



- ”Selection” – starting with some vector in latent space
- “Navigation” – movement along vectors
- Selection and Navigation within the latent space produces “visually interesting” results
- If we wanted to make fake digits, **How might we ensure that they are actually correct?**

One approach: sequentially check if “generated” images match ground-truth with a classifier



I. A. Attacking a network with adversarial examples

Goal: Given a network pretrained on ImageNet, find an input image that is not a iguana but will be classified as an iguana.

X ⇒ **Neural network** (pretrained on ImageNet) ⇒

$$\begin{pmatrix} 0.04 \\ 0.85 \\ 0.02 \\ 0.07 \\ 0.81 \\ \vdots \\ 0.02 \end{pmatrix} \begin{matrix} \text{“car”} \\ \text{“iguana} \\ \text{“tomato”} \\ \text{“bike”} \\ \text{“cat”} \\ \\ \text{“crab”} \end{matrix}$$

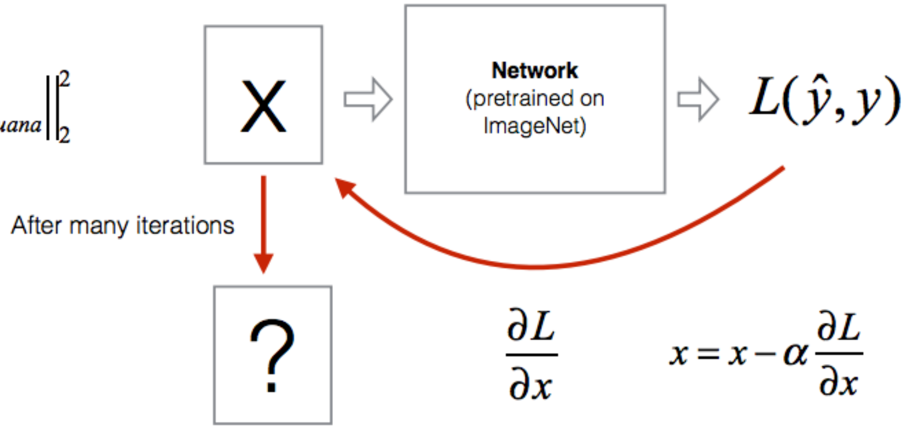
1. Rephrasing what we want:

Find x such that: $\hat{y}(x) = y_{iguana} = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

2. Defining the loss function

$$L(\hat{y}, y) = \frac{1}{2}\left\| \hat{y}(W,b,x) - y_{iguana} \right\|_2^2$$

3. Optimize the image



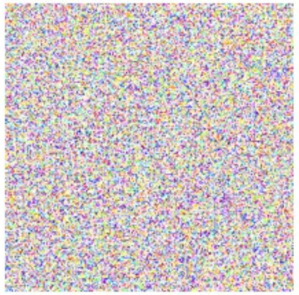
[Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy (2015): Explaining and harnessing adversarial examples]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Stanford CS230, Lecture 3

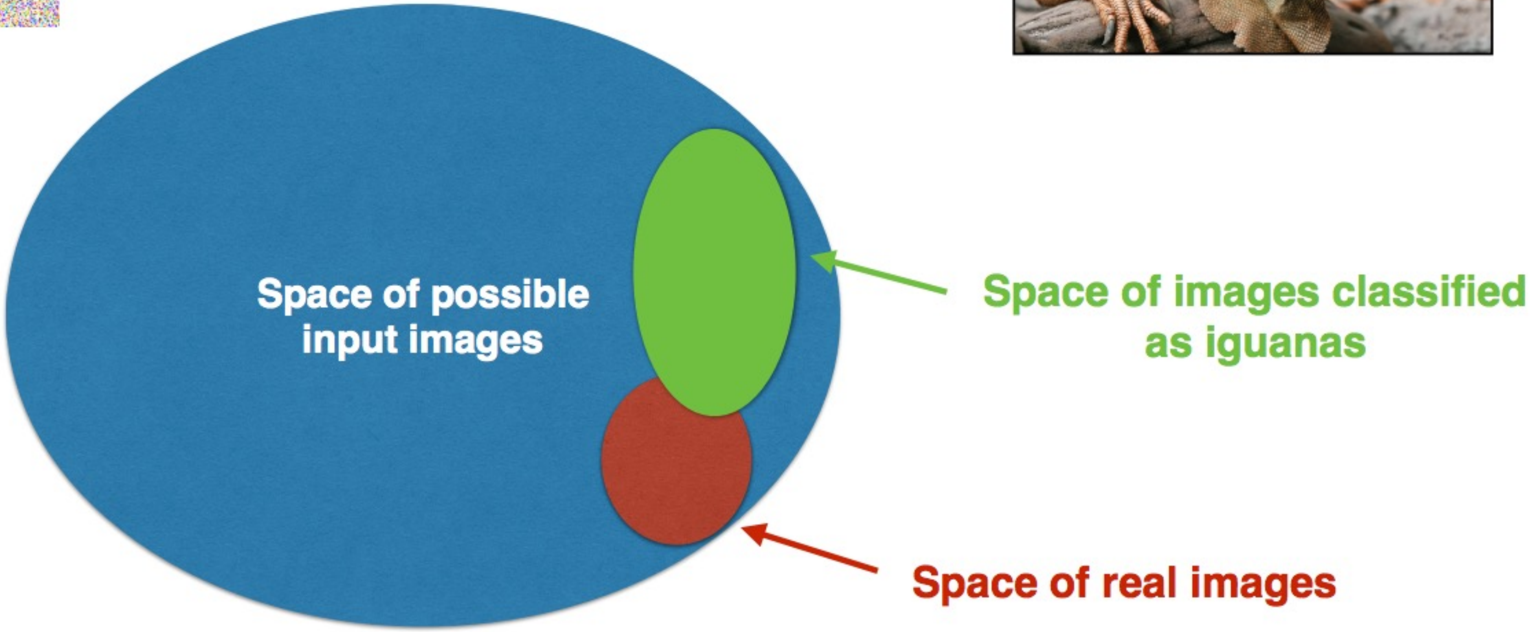
I. A. Attacking a network with adversarial examples

Question: Will the forged image **x** look like an **iguana**?



?

$$256^{32\times32\times3} \approx 10^{7400}$$



I. A. Attacking a network with adversarial examples

deep imaging

Goal: Given a network pretrained on ImageNet, find an input image that is a cat but will be classify as an iguana.

Neural network (pretrained on ImageNet)

$$\begin{pmatrix} 0.04 \\ 0.85 \\ 0.02 \\ 0.07 \\ 0.81 \\ \vdots \\ 0.02 \end{pmatrix} \begin{array}{l} \text{"car"} \\ \text{"iguana} \\ \text{"tomato"} \\ \text{"bike"} \\ \text{"cat"} \\ \\ \text{"crab"} \end{array}$$

1. Rephrasing what we want:

Find x such that: $\hat{y}(x) = y_{iguana} = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

And: $x = x_{cat}$

2. Defining the loss function

$$L(\hat{y}, y) = \frac{1}{2}\left\|\hat{y}(W,b,x) - y_{iguana}\right\|_2^2 + \lambda\left\|x - x_{cat}\right\|_2^2$$

3. Optimize the image

Network (pretrained on ImageNet)

$L(\hat{y}, y)$

After many iterations

$$\frac{\partial L}{\partial x} \qquad x = x - \alpha\frac{\partial L}{\partial x}$$

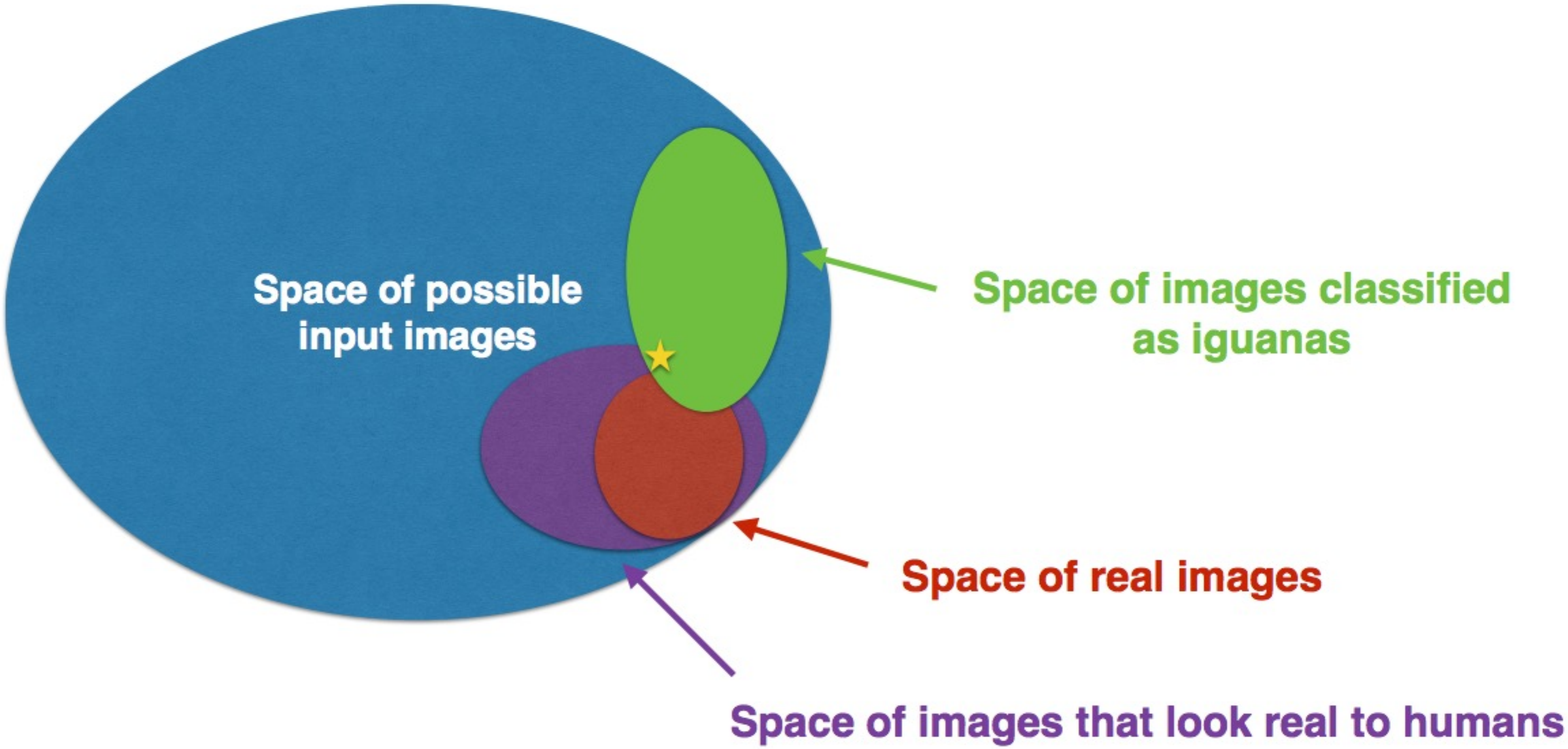
[Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy (2015): Explaining and harnessing adversarial examples]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Stanford CS230, Lecture 3

I. A. Attacking a network with adversarial examples

$$256^{32\times32\times3} \approx 10^{7400}$$



I. B. Defenses against adversarial examples

Menti

Knowledge of the attacker:

- White-box
- Black-box

Solution 1

- Create a SafetyNet

Solution 2

- Train on correctly labelled adversarial examples

$x =$ 

y = cat

Solution 3

- Adversarial training $L_{new} = L(W,b,x,y) + \lambda L(W,b,x_{adv},y)$
- Adversarial logit pairing $L_{new} = L(W,b,x,y) + \lambda \left\| f(x;W,b) - f(x_{adv};W,b) \right\|_2^2$

[Lu et al. (2017): SafetyNet: Detecting and Rejecting Adversarial Examples Robustly]
[Harini Kannan et al. (2018): Adversarial Logit Pairing]

NO Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles

Jiajun Lu,* Hussein Sibai,* Evan Fabry, David Forsyth
University of Illinois at Urbana Champaign
{jlu23, sibai2, efabry2, daf}@illinois.edu

deep imaging

It has been shown that most machine learning algorithms are susceptible to adversarial perturbations. Slightly perturbing an image in a carefully chosen direction in the image space may cause a trained neural network model to misclassify it. Recently, it was shown that physical adversarial examples exist: printing perturbed images then taking pictures of them would still result in misclassification. This raises security and safety concerns.

However, these experiments ignore a crucial property of physical objects: the camera can view objects from different distances and at different angles. In this paper, we show experiments that suggest that current constructions of physical adversarial examples do not disrupt object detection from a moving platform. Instead, a trained neural network classifies most of the pictures taken from different distances and angles of a perturbed image correctly. We believe this is because the adversarial property of the perturbation is sensitive to the scale at which the perturbed picture is viewed, so (for example) an autonomous car will misclassify a stop sign only from a small range of distances.

4 different adversarial examples for object detector:

4 different adversarial examples for object classifier:

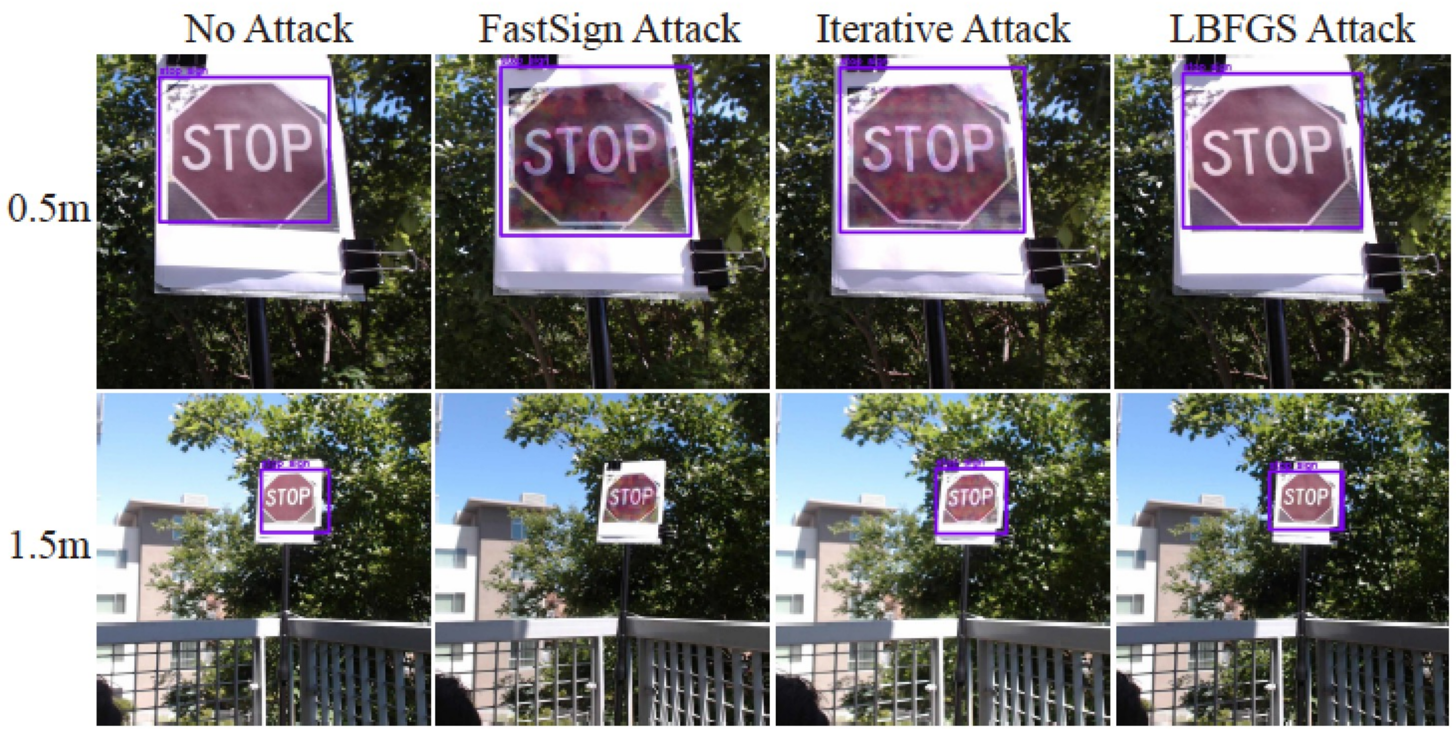


Figure 3: This figure shows experiment setup, and we use the printed stop signs to simulate real stop signs with natural background. These are examples for successful 0.5 meters and 1.5 meters detection: both original images and adversarial examples are detected in both distances. It demonstrates that adversarial examples in a physical setting do not reliably fool stop sign detectors.

Within 5 days (!), a blog post from OpenAI:

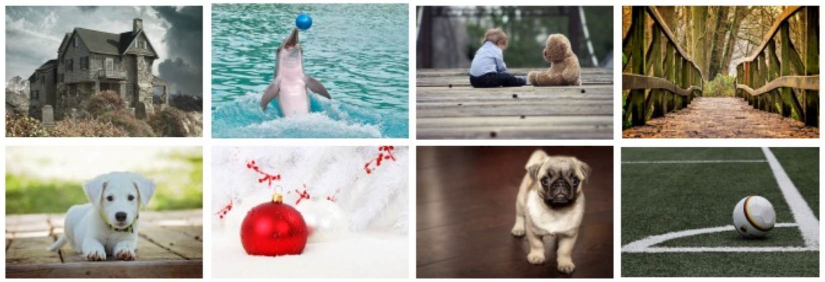
https://openai.com/research/robust-adversarial-inputs

Generative adversarial networks

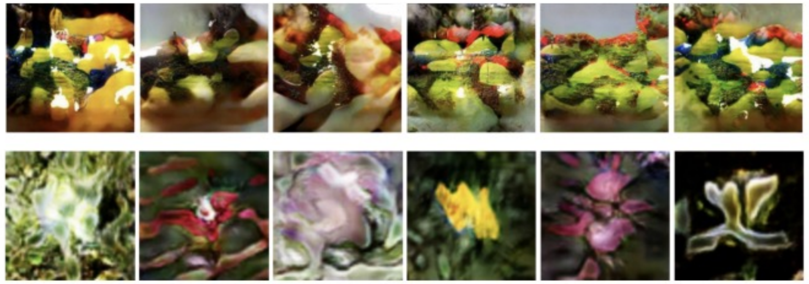
II.A - Motivation

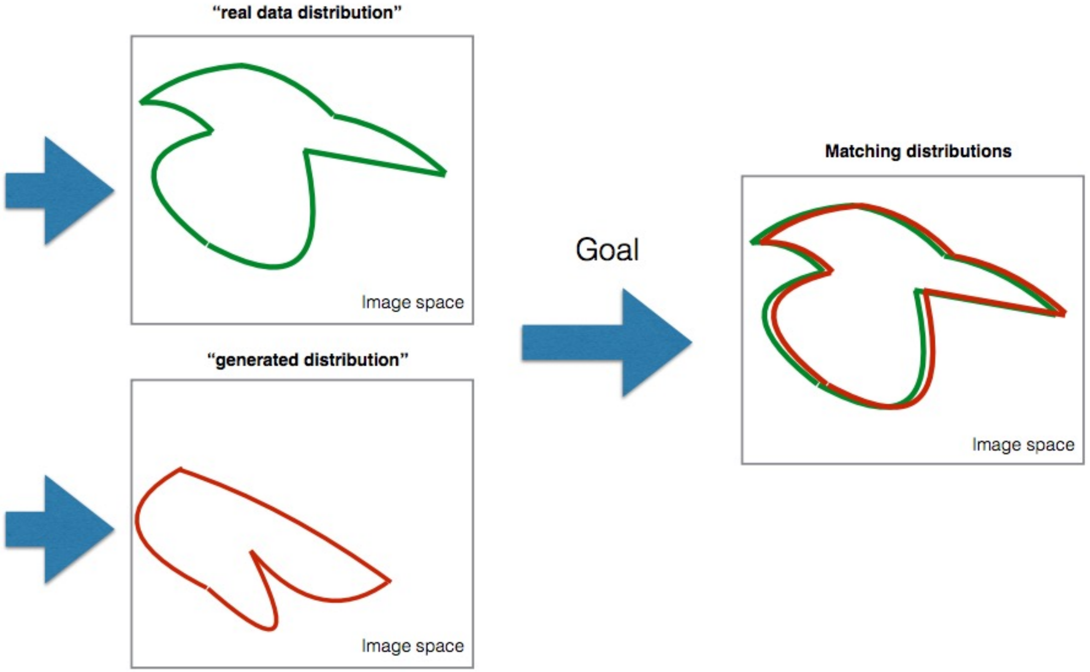
Probability distributions:

Samples from the "real data distribution"



Samples from the "generated distribution"



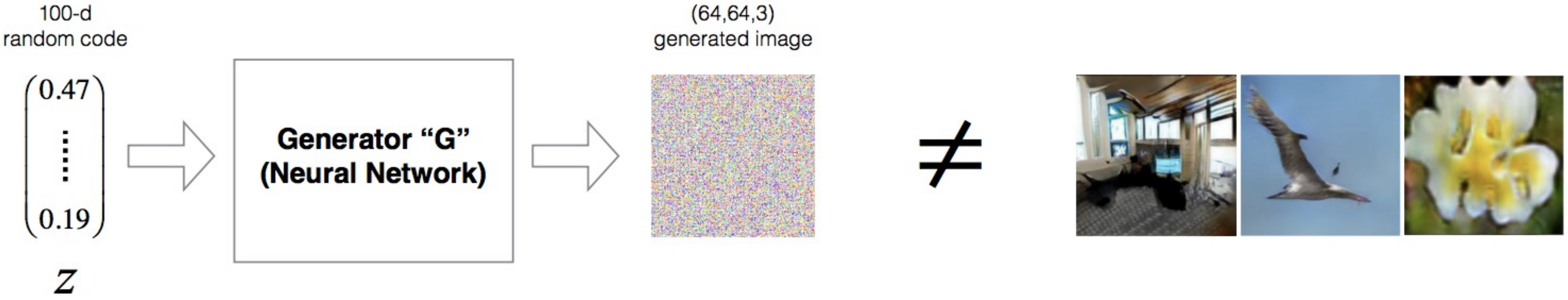


[Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris Metaxas (2017): StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Stanford CS230, Lecture 3

II.B - G/D Game

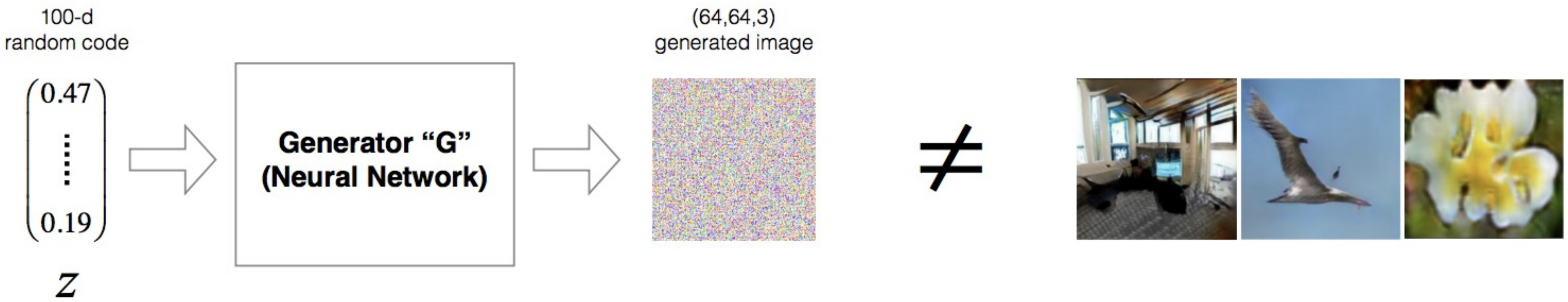


How can we train G to generate images from the true data distributions?

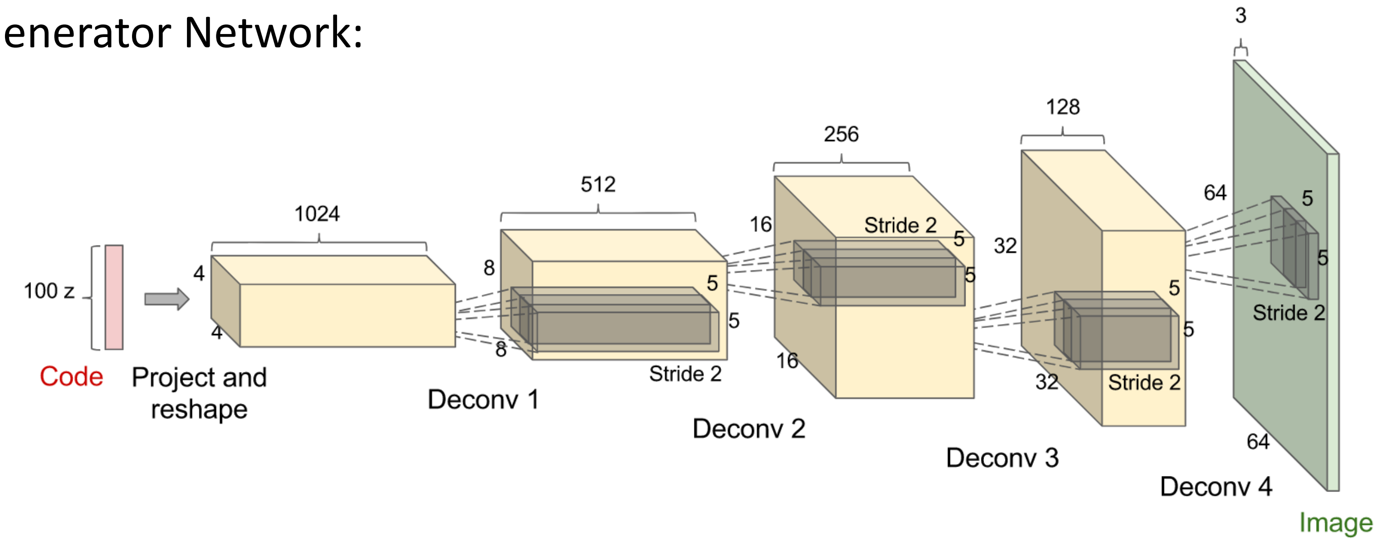
[Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris Metaxas (2017): StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

II.B - G/D Game



DCGAN Generator Network:



[Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris Metaxas (2017): StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Stanford CS230, Lecture 3

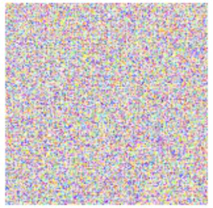
II.B - G/D Game

100-d random code

$$z = \begin{pmatrix} 0.47 \\ \vdots \\ 0.19 \end{pmatrix}$$

Generator "G" (Neural Network)

(64,64,3) generated image



Run Adam simultaneously on two minibatches (true data / generated data)

x

Gradients

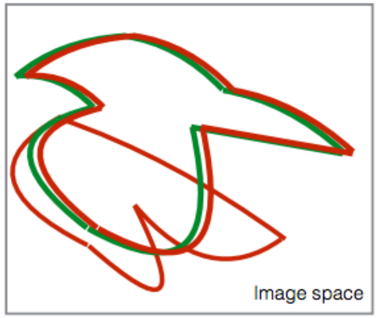
Discriminator "D" (Neural Network)

Binary classification

$$\begin{cases} y = 0 & if \quad x = G(z) \\ y = 1 & otherwise \end{cases}$$

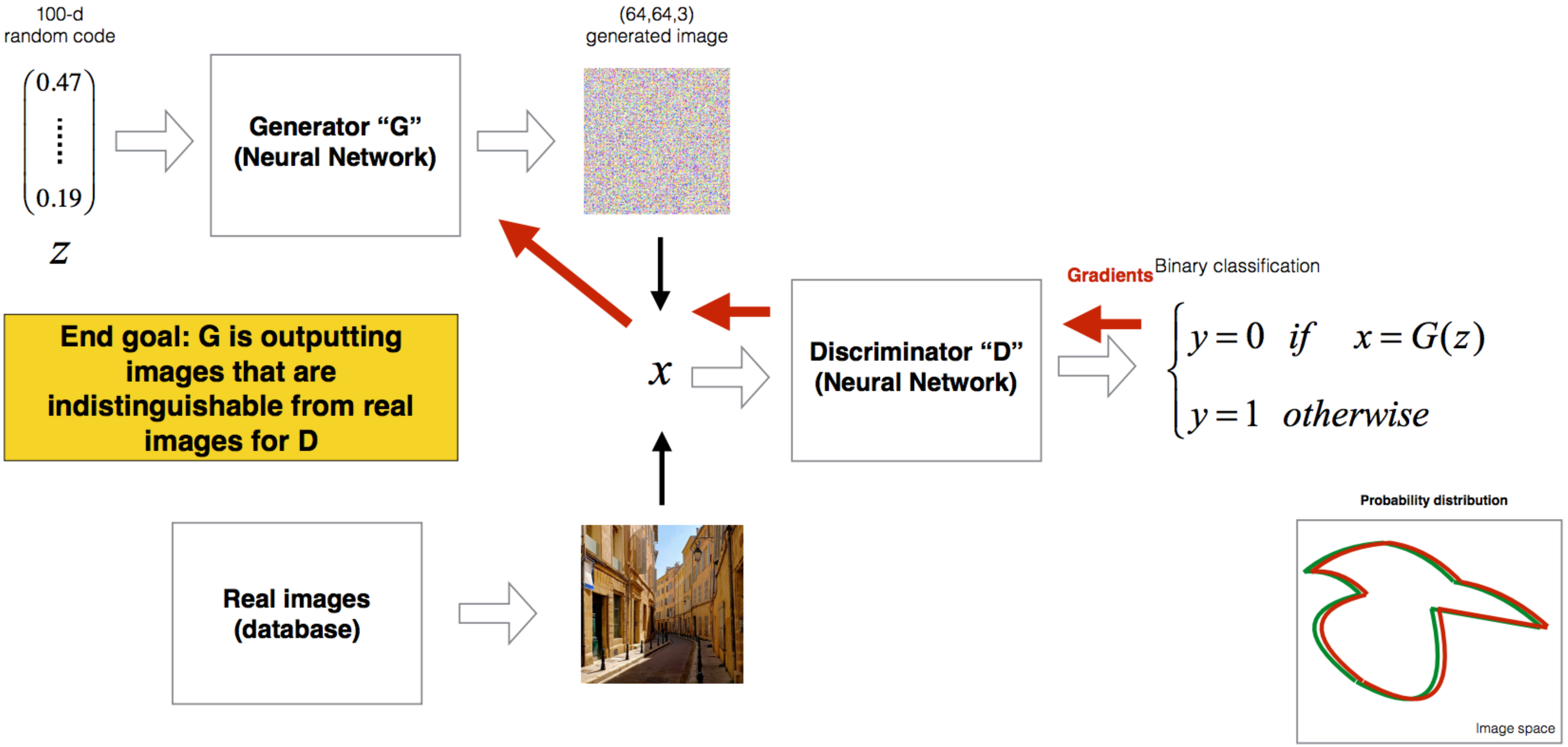
Real images (database)

Probability distributions



Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

II.B - G/D Game



Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Stanford CS230, Lecture 3

II.B - G/D Game

Training procedure, we want to minimize:

Labels: $\begin{cases} y_{real} & \text{is always 1} \\ y_{gen} & \text{is always 0} \end{cases}$

- The cost of the discriminator

$$J^{(D)} = \underbrace{-\frac{1}{m_{real}} \sum_{i=1}^{m_{real}} y_{real}^{(i)} . \log(D(x^{(i)}))}_{\substack{\text{cross-entropy 1:} \\ \text{"D should correctly label real data as 1"}}} \underbrace{- \frac{1}{m_{gen}} \sum_{i=1}^{m_{gen}} (1 - y_{gen}^{(i)}) . \log(1 - D(G(z^{(i)})))}_{\substack{\text{cross-entropy 2:} \\ \text{"D should correctly label generated data as 0"}}}$$

- The cost of the generator

$$J^{(G)} = -J^{(D)} = \frac{1}{m_{gen}} \sum_{i=1}^{m_{gen}} \log(1 - D(G(z^{(i)})))$$

"G should try to fool D: by minimizing the opposite of what D is trying to minimize"

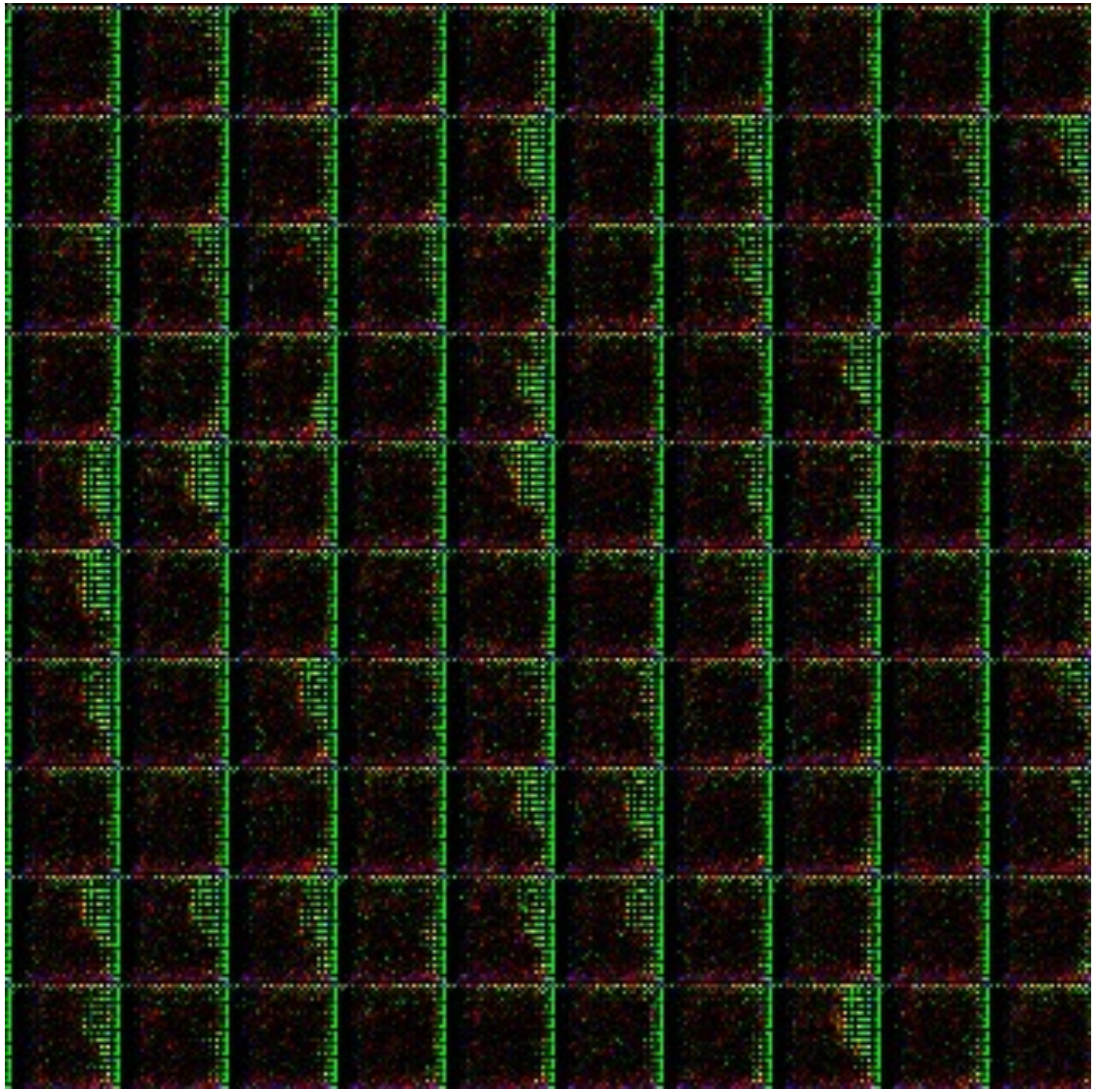
"Maximize probability that the discriminator is wrong and labels the fake example as a real example"

- Typical # of weights: 100 Million

- Typical training dataset:
20GB (e.g., Imagenet)

- Lots of code and resources available to get started

- Increasing integration with large language models



II. D. In terms of code

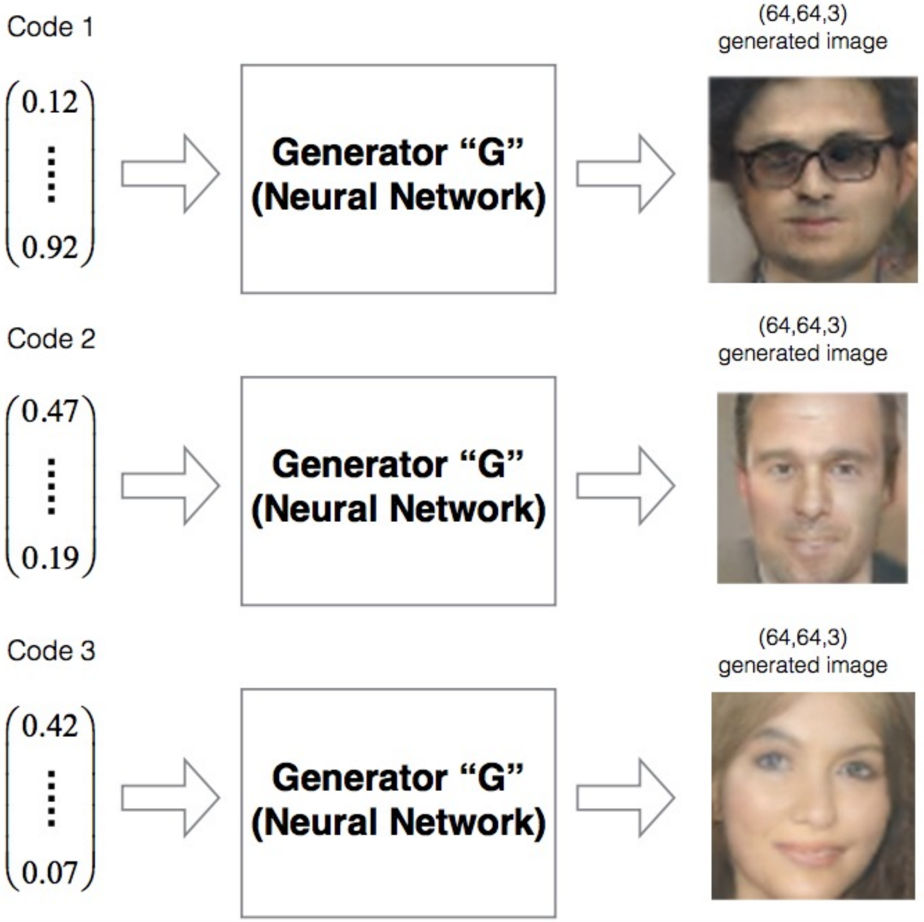
```python
# Build and compile the discriminator
self.discriminator = self.build_discriminator()
self.discriminator.compile(loss='binary_crossentropy',
    optimizer=optimizer,
    metrics=['accuracy'])

# Build the generator
self.generator = self.build_generator()

# The generator takes noise as input and generates imgs
z = Input(shape=(self.latent_dim,))
img = self.generator(z)

# For the combined model we will only train the generator
self.discriminator.trainable = False

# The discriminator takes generated images as input and determines validity
validity = self.discriminator(img)

# The combined model  (stacked generator and discriminator)
# Trains the generator to fool the discriminator
self.combined = Model(z, validity)
self.combined.compile(loss='binary_crossentropy', optimizer=optimizer)
```

```python
def build_discriminator(self):

    model = Sequential()

    model.add(Flatten(input_shape=self.img_shape))
    model.add(Dense(512))
    model.add(LeakyReLU(alpha=0.2))
    model.add(Dense(256))
    model.add(LeakyReLU(alpha=0.2))
    model.add(Dense(1, activation='sigmoid'))
    model.summary()

    img = Input(shape=self.img_shape)
    validity = model(img)

    return Model(img, validity)
```

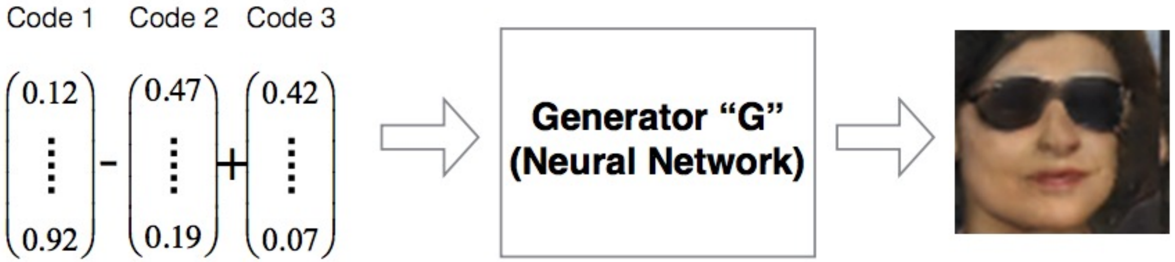
Erik Linder-Norén (Github): eriklindernoren/Keras-GAN: link]

II.E - Nice results

Operation on codes



Note: this is like the previously mentioned selection and navigation process!



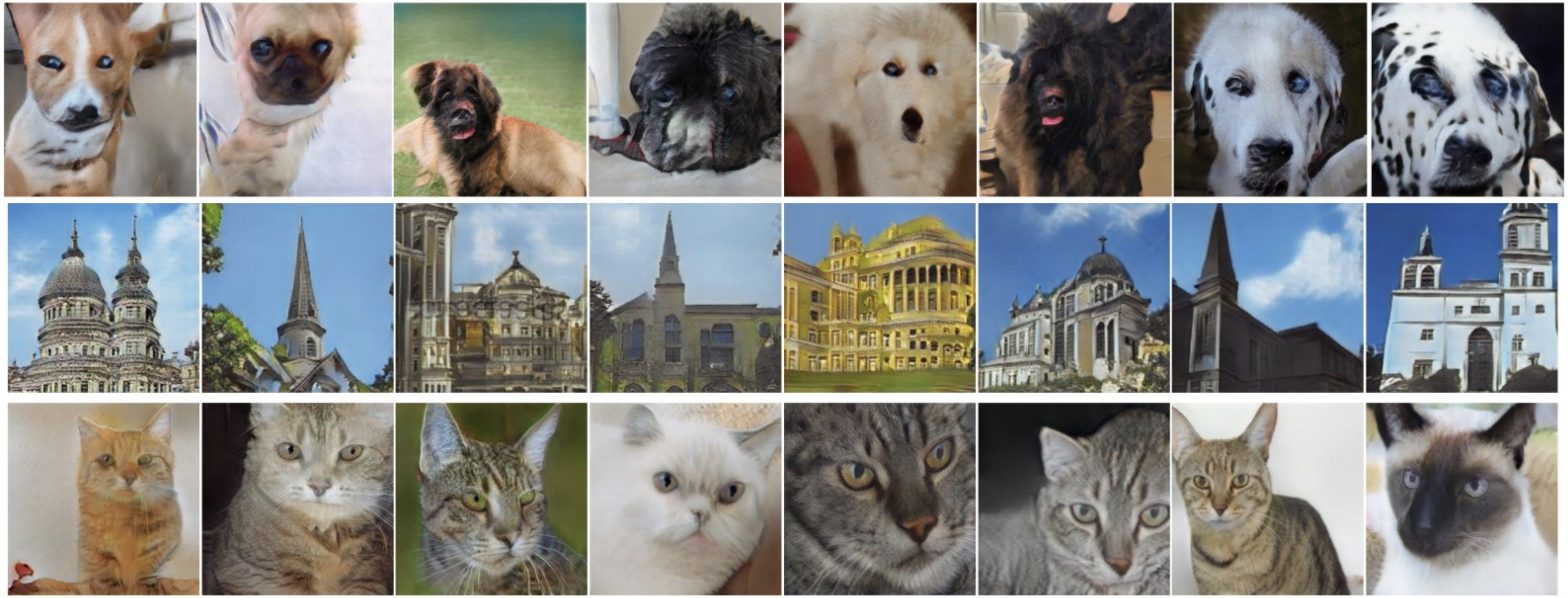
Man with glasses - man + woman = woman with glasses

[Radford et al. (2015): UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS]

II.E - Nice results

Image Generation:

Samples from the "generated distribution"



[Zhang et al. (2017): StackGAN++]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Example of the Progression in the Capabilities of GANs from 2014 to 2017.Taken from The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation, 2018.

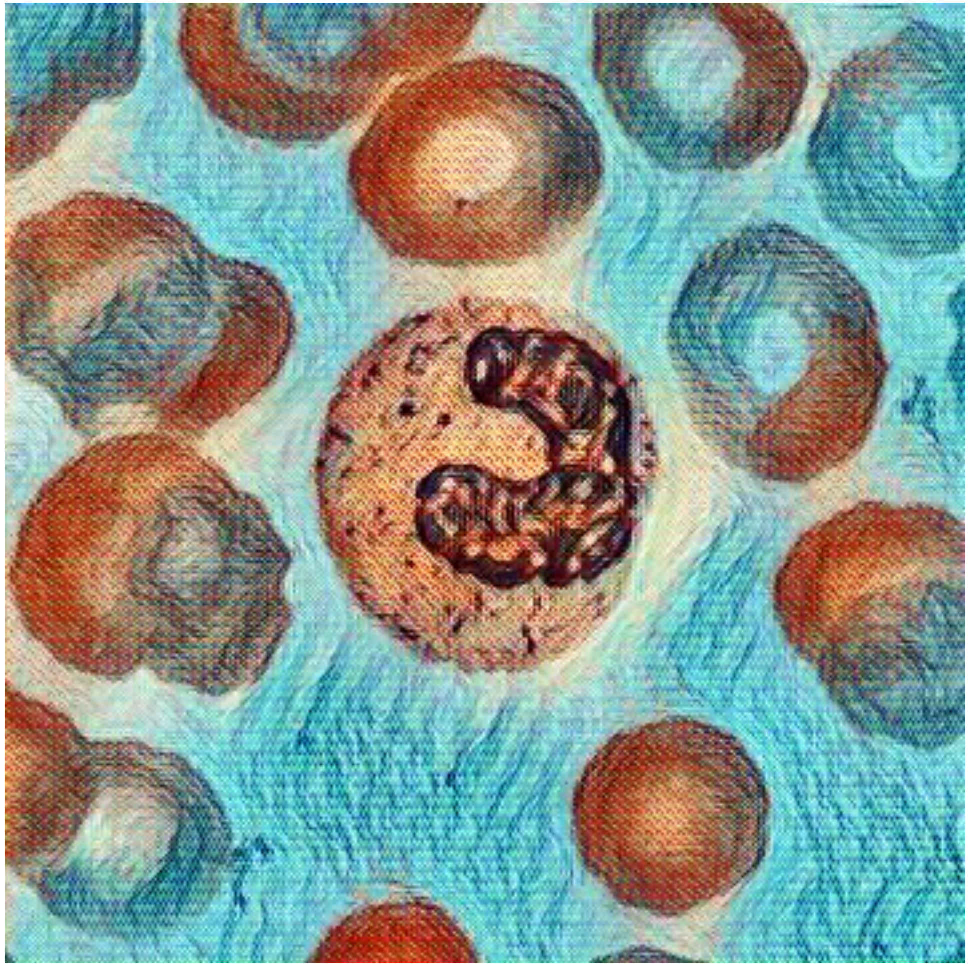
In 2024…

https://www.youtube.com/watch?v=HK6y8DAPN_0

deep imaging

Applications in biomedical/scientific imaging?

- Generating labeled examples when annotation is difficult/impossible
- Transferring images of one type to another ("style transfer") for human viewing
- Increasing dataset sizes (augmentation, but better)



Domain Transfer and Virtual "Staining"/Fluorescence

CycleGANs

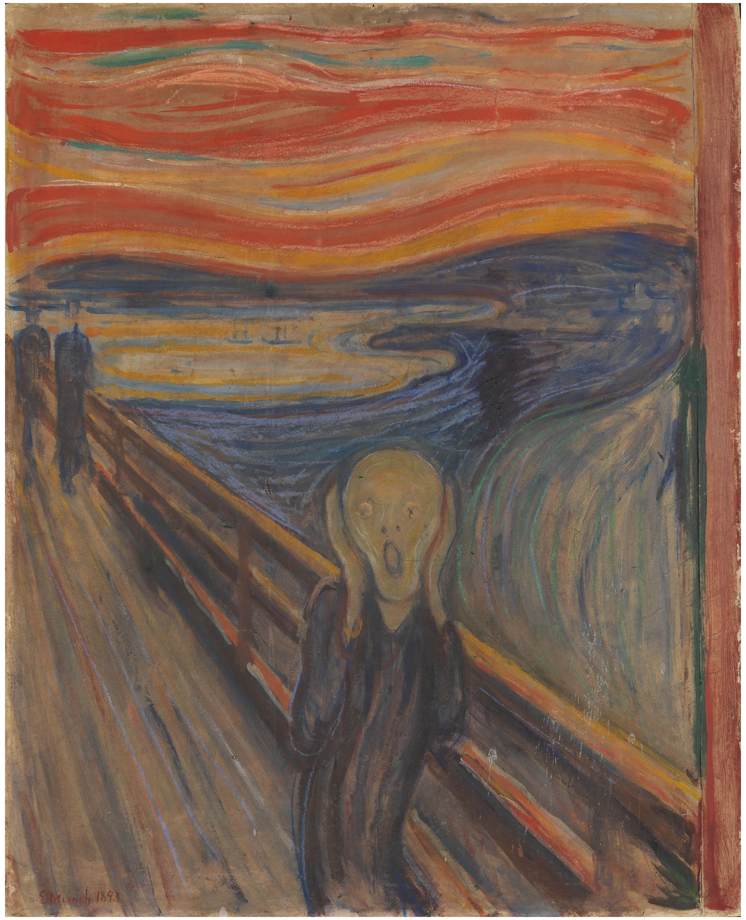
Picture 1

What is content in this picture?

What is the style?

CycleGANs

deep imaging



Picture 2

What is content in this picture?

What is the style?

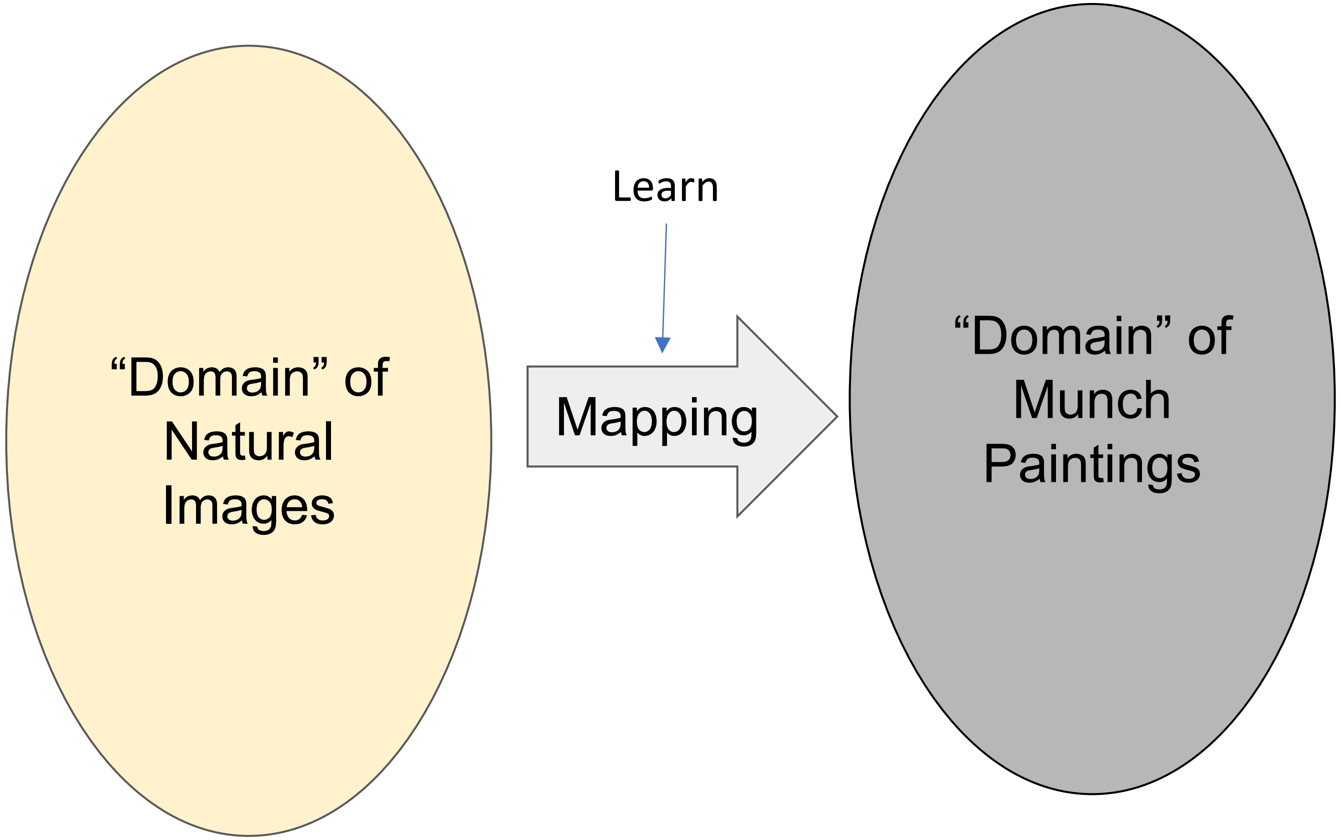
CycleGANs

What would it look like if the landscape was painted like The Scream?

Or, what would the content from Picture 1 look like in the style of Picture 2?

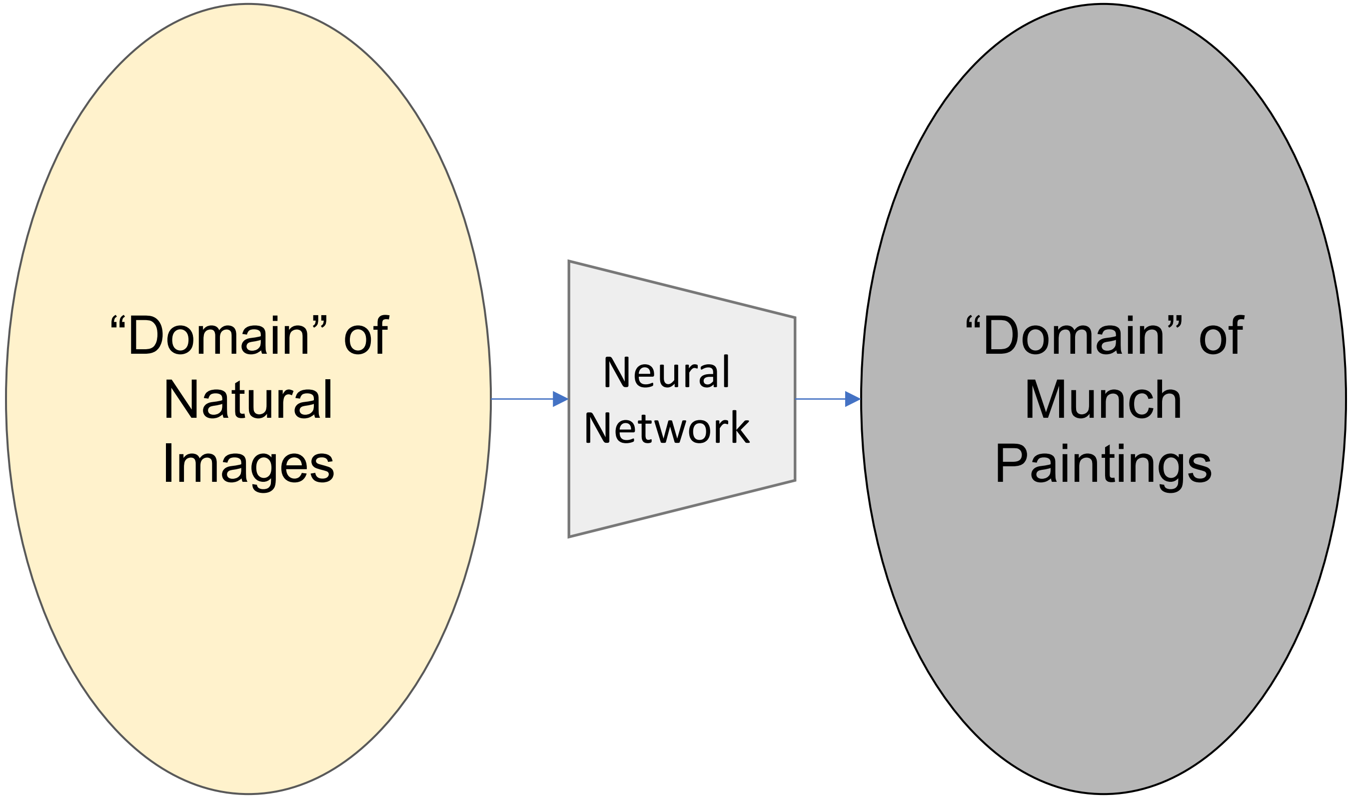
We do not have paired data!

CycleGANs – how?



CycleGANs – how?

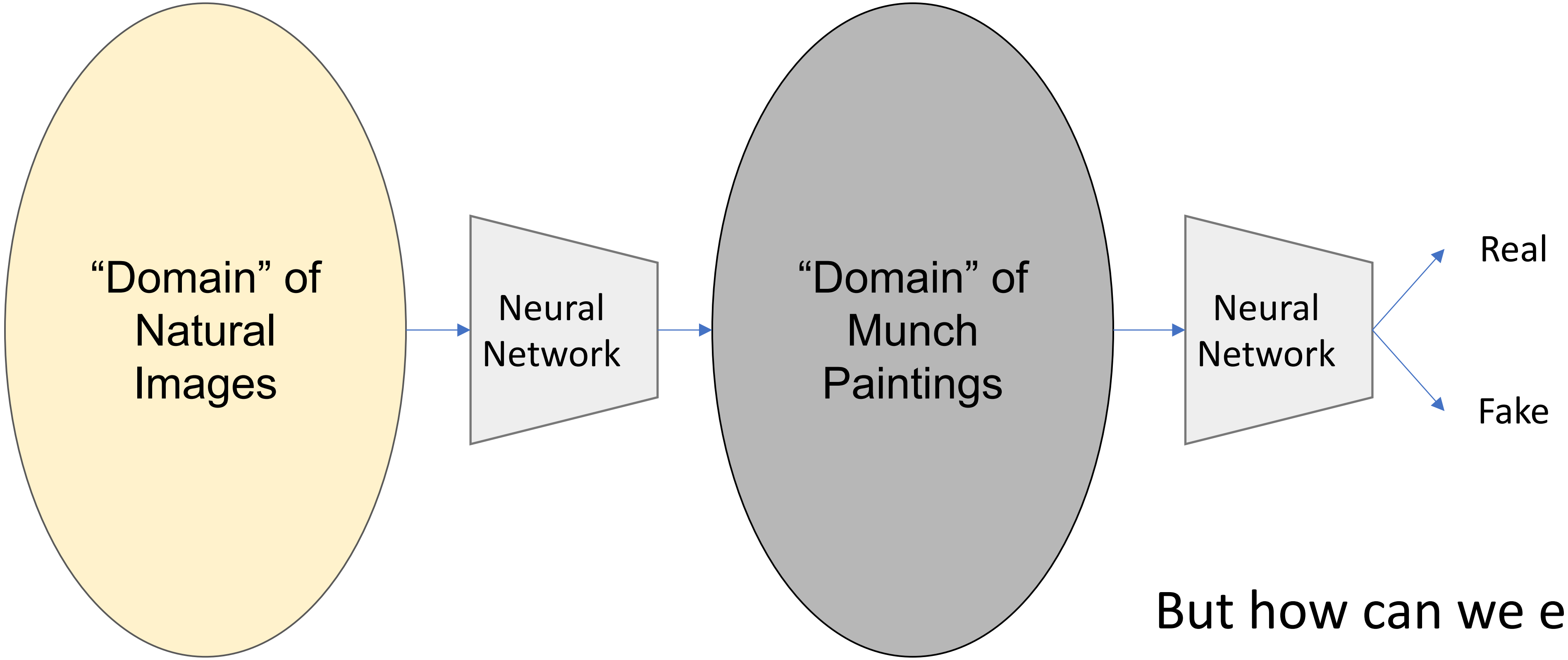
We can learn a mapping with supervised learning…



But that needs a lot of paired data!

CycleGANs – how?

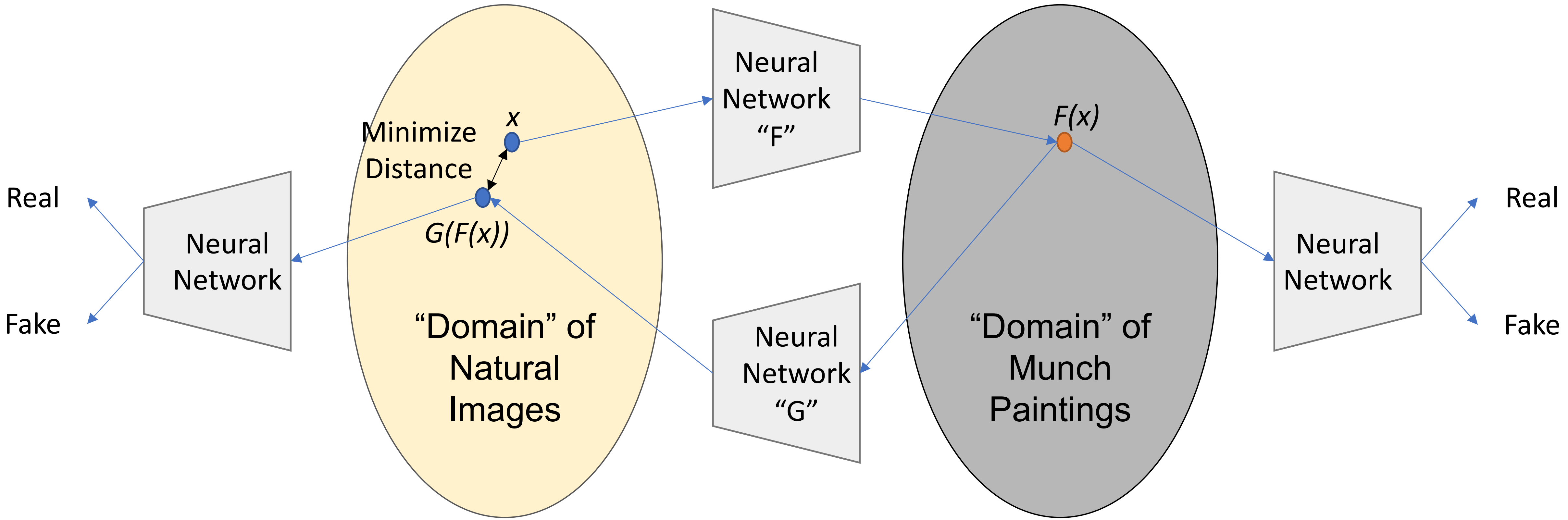
We could use a GAN instead:



But how can we ensure that the content is the same?

CycleGAN

Use a second GAN!

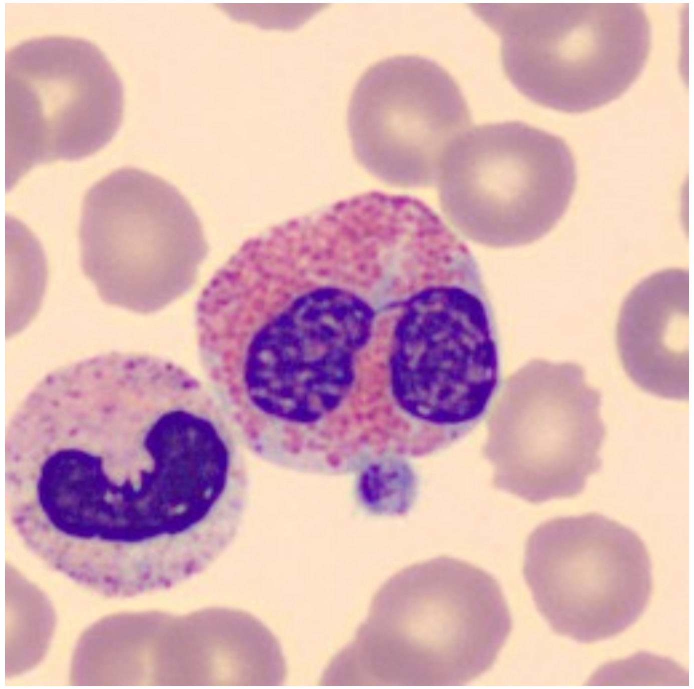


CycleGAN

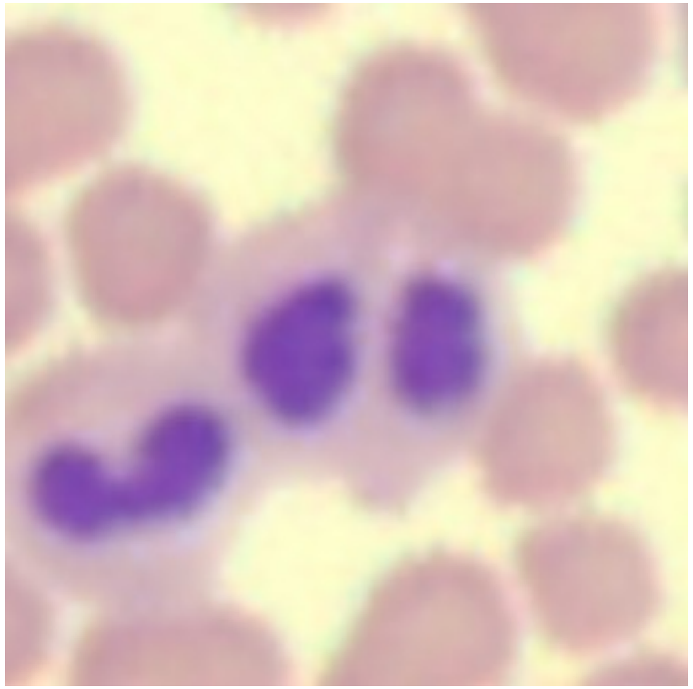
- Learn a forward mapping F and reverse mapping G
- Make sure that G(F(x)) is very close to x – **Imposes content preservation**
- Use discriminator with F(x) and known style image – **Imposes Style similarity**

Where is this useful? “Lossy learning” – let’s make use of annotated images online!

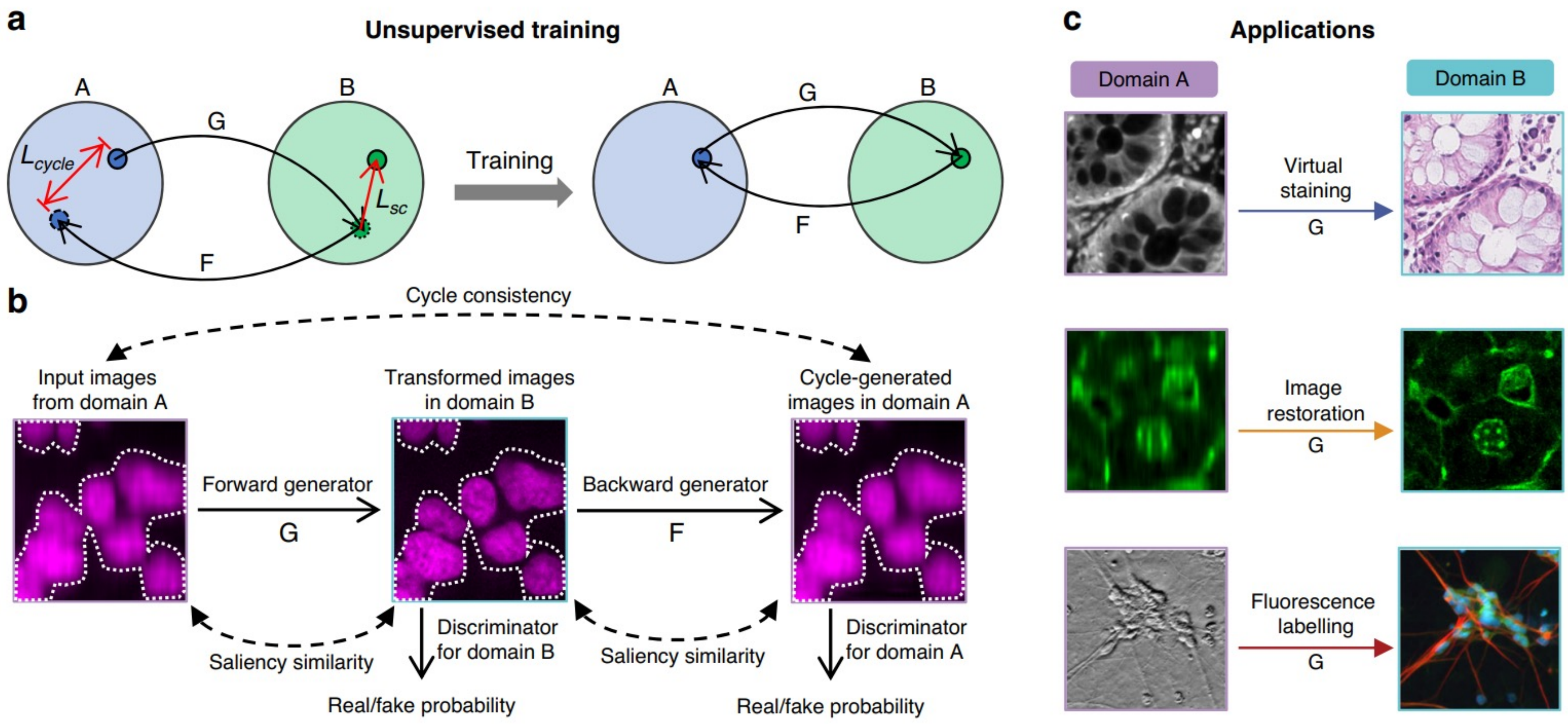
Easy to annotate



What my data looks like....



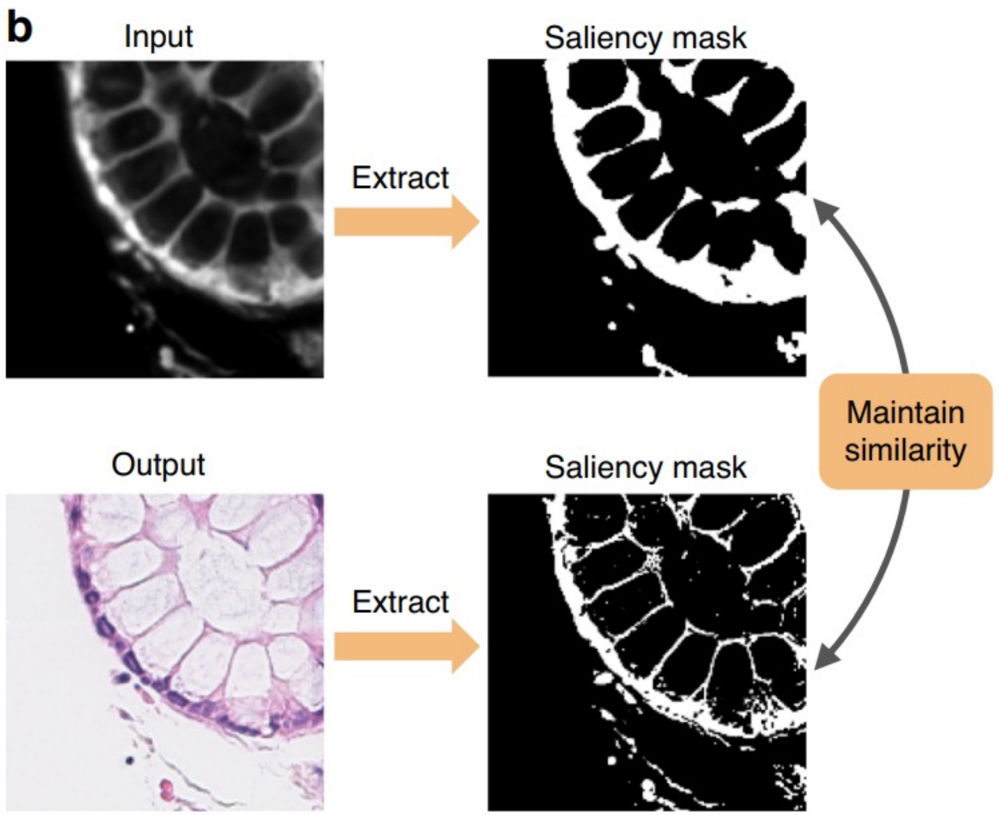
Application in Biomedical Context



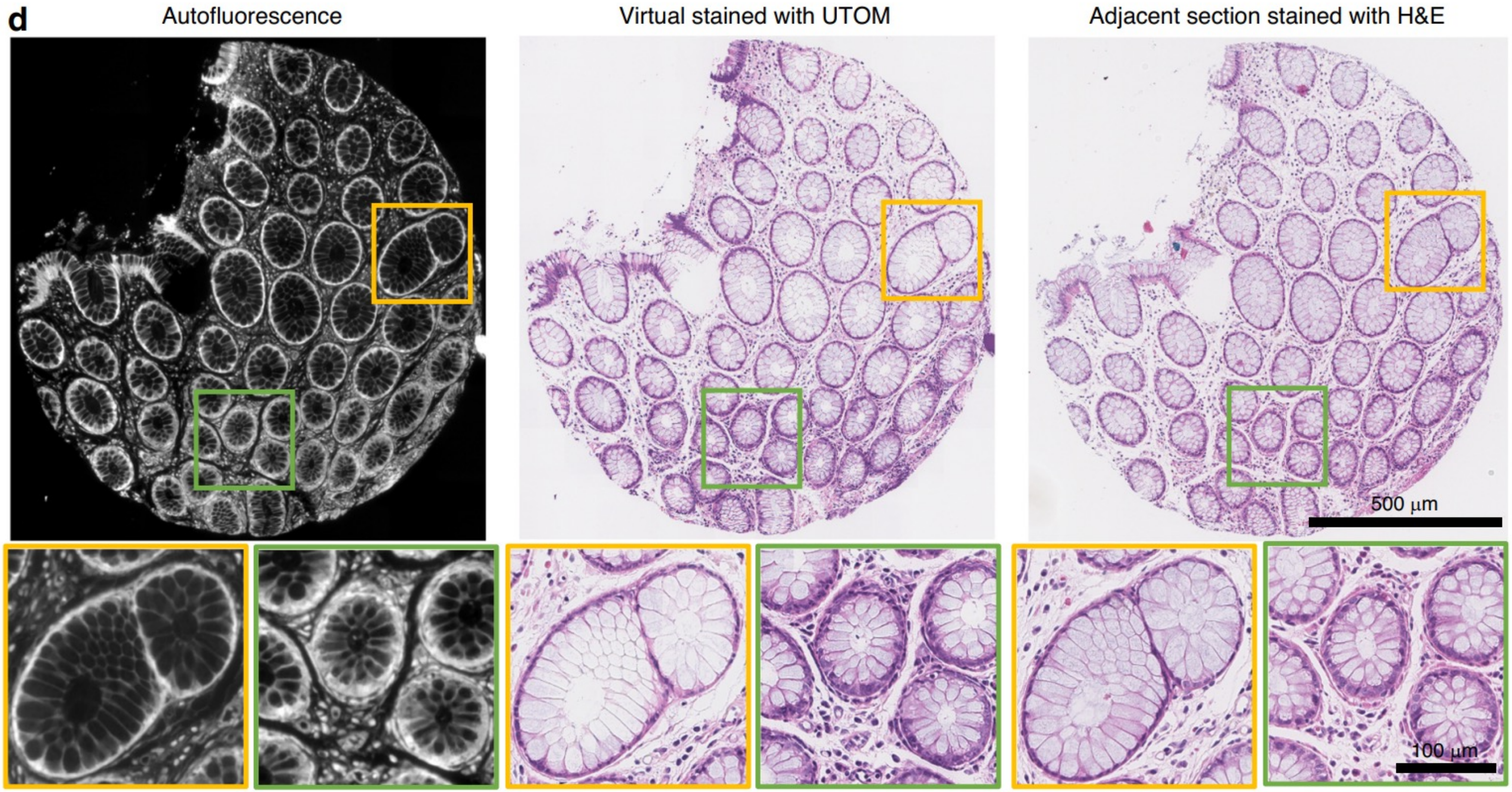
Li, Xinyang, et al. "Unsupervised content-preserving transformation for optical microscopy." *Light: Science & Applications* 10.1 (2021): 44.

Additional Step

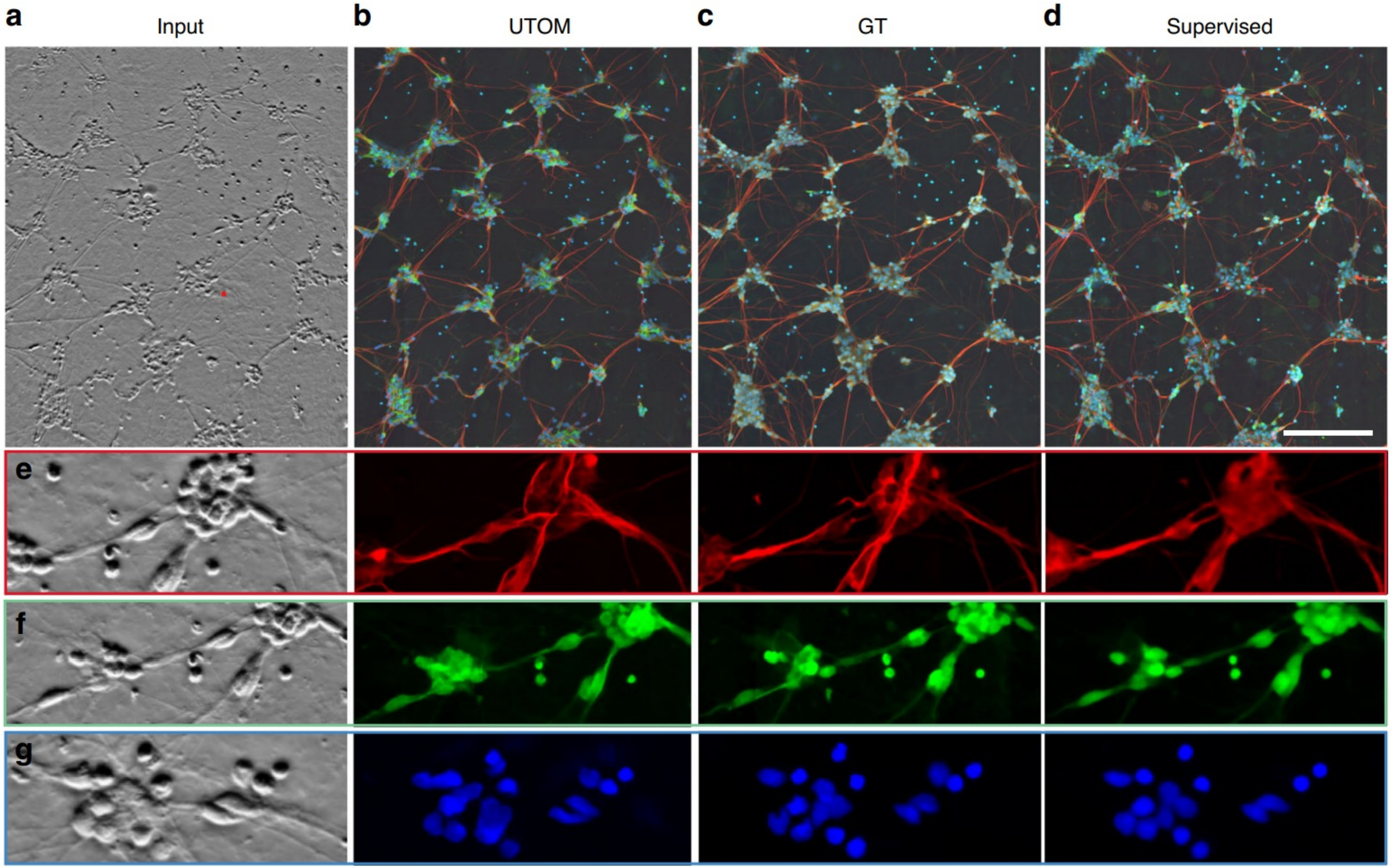
- In order to strongly impose content preservation, minimize saliency on input and generated images



Results – Autofluroscence to H&E



Results – Virtual Fluorescence



CycleGANs and Domain Adaptation

Machine Learning needs a ton of labeled data that is expensive to obtain

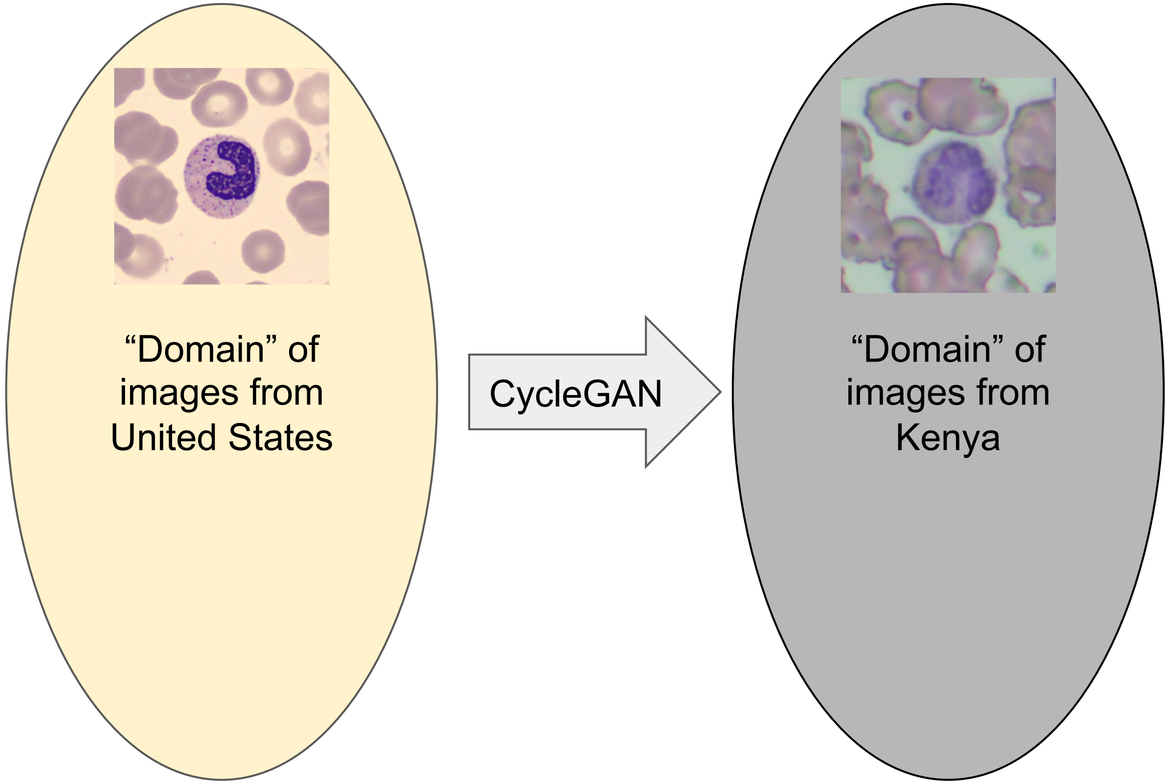
Machine Learning only works if it used on data that is similar to training data

Existing labeled data cannot be used if we develop a new imaging system

CycleGANs and Domain Adaptation

We could solve this problem by using CycleGANs to make the available data look like the data we want to learn

Lots of experts to label data

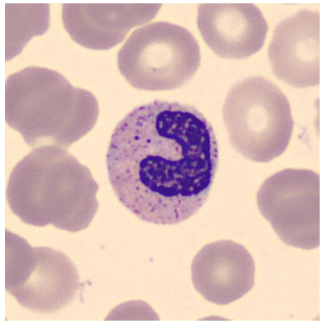


Few experts to label data

CycleGANs and Domain Adaptation

Using different source domains for the learning process of an algorithm, which is then applied to a different but related target domain is called **domain adaptation**.

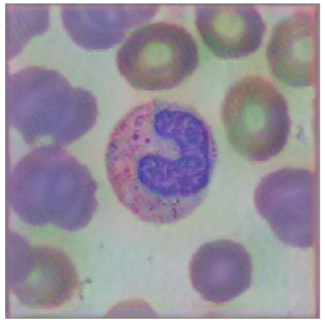
Image from
United States



Label: Basophil

“Make it look like it came from Kenya..”

Looks like image
from Kenya



Label: Basophil

Apply to novel machine learning tasks