Generative AI with Diffusion Models

Part 1: From U-Nets to Diffusion

Agenda

- Part 1: From U-Nets to Diffusion
- Part 2: Denoising Diffusion Probabilistic Models
- Part 3: Optimizations
- Part 4: Classifier Free Diffusion
- Part 5: CLIP
- Part 6: Wrap-up & Assessment

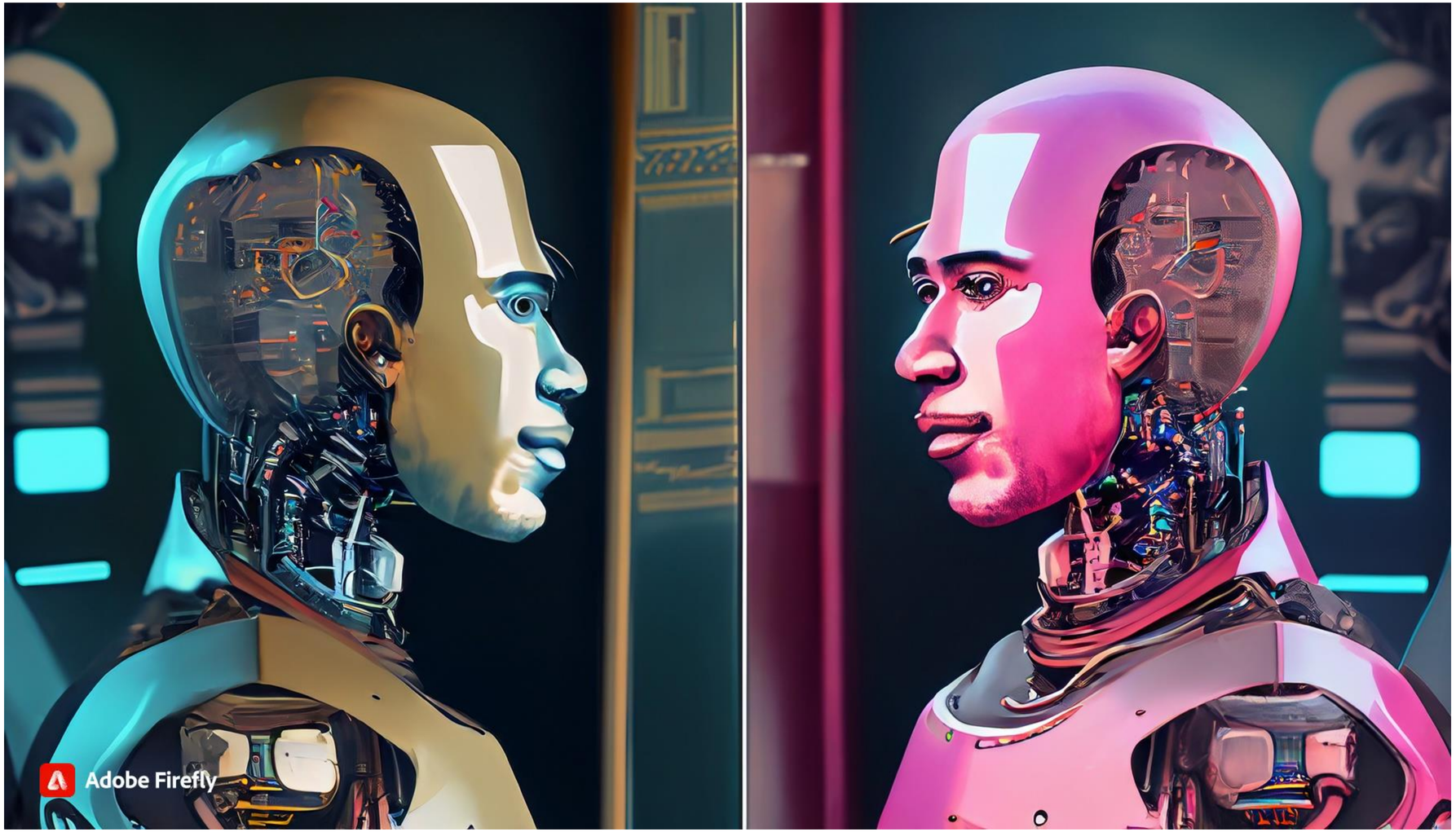
Prerequisites

- Basic familiarity with convolutional neural networks (CNNs)
- Basic familiarity with a deep learning framework such as:
  - PyTorch
  - TensorFlow

A Brief History of Generative AI

The Imitation Game

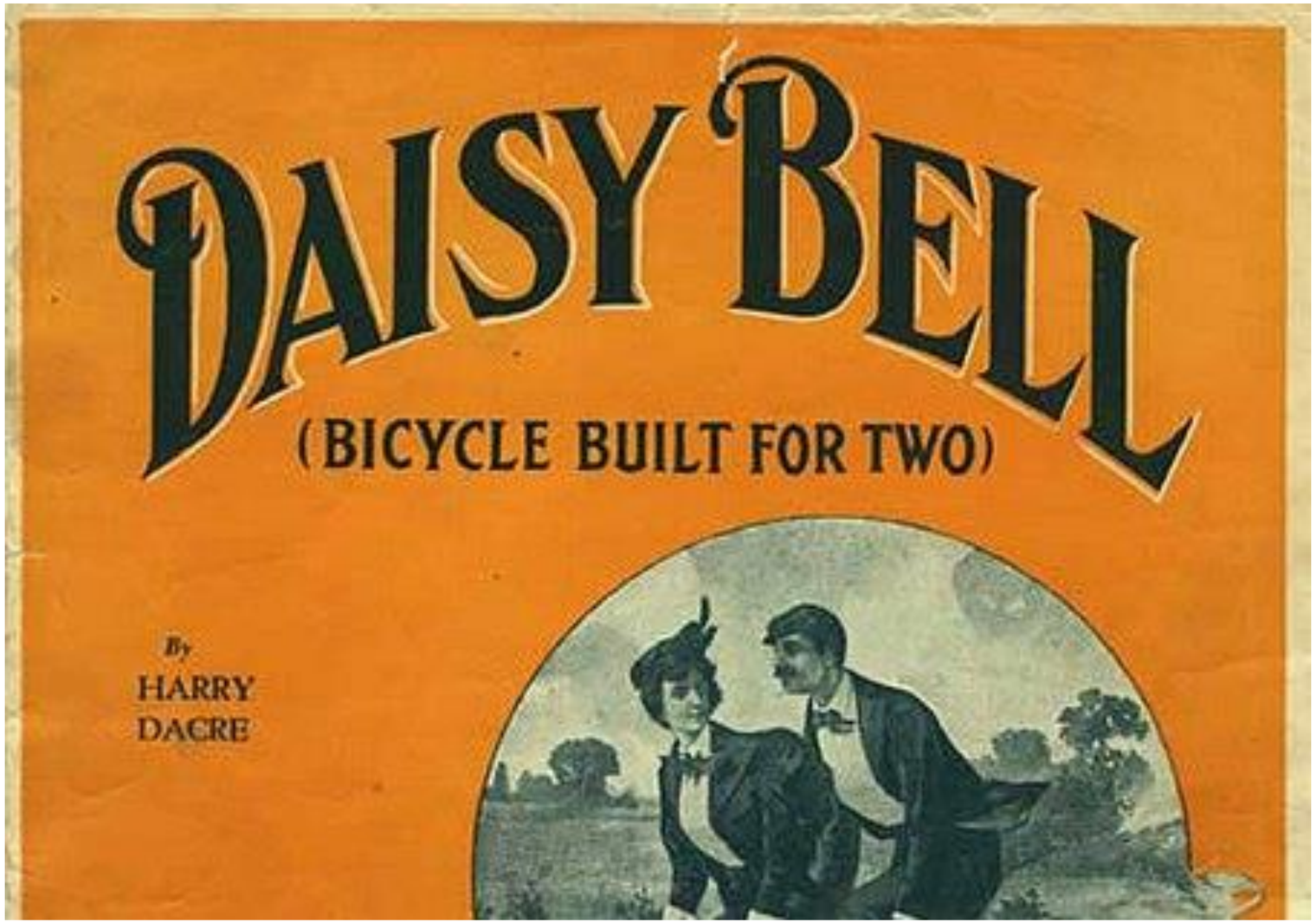
A.K.A The Turing Test



A robot looks in a mirror and the reflection is human, cyberpunk

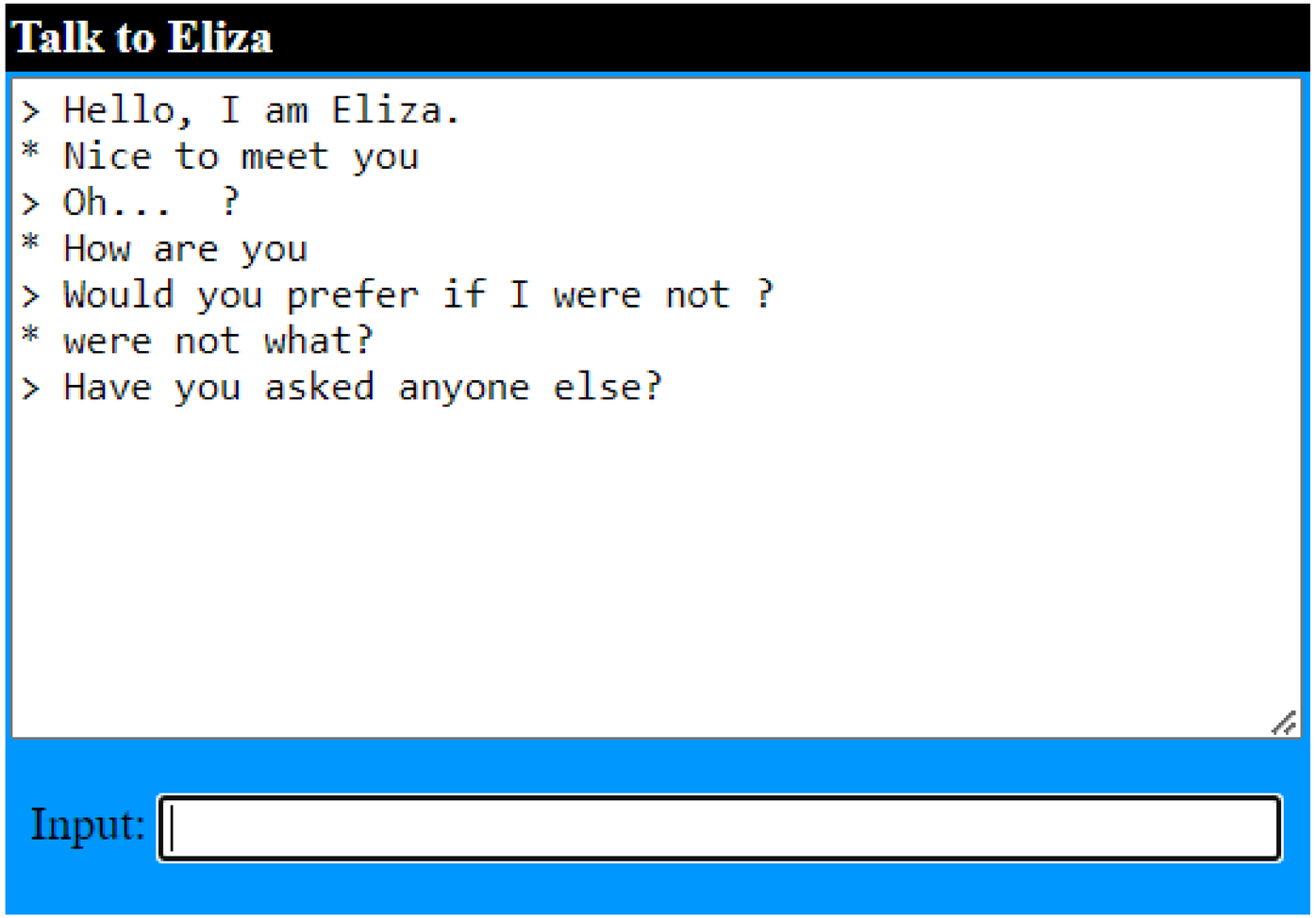
IBM 704

The First Singing Computer



Eliza

The First Gen AI Chatbot?



Generative AI of the 70’s, 80’s and 90s?

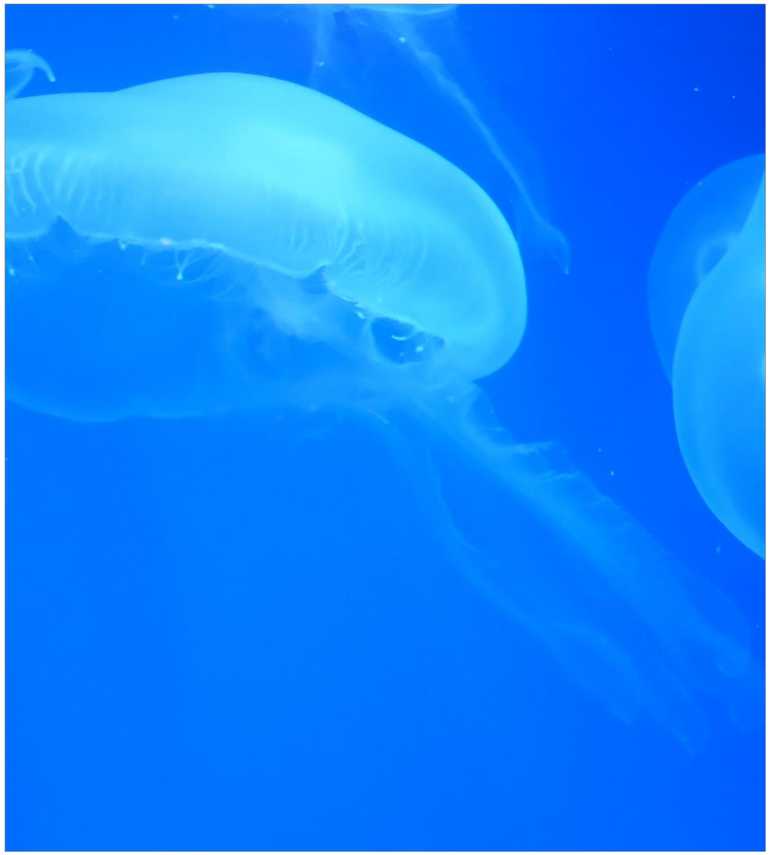
- Electronic music
- Video games graphics
- Video game AI
- Computer animation
- Instant messaging chatbots

An 80’s arcade with lots of machines

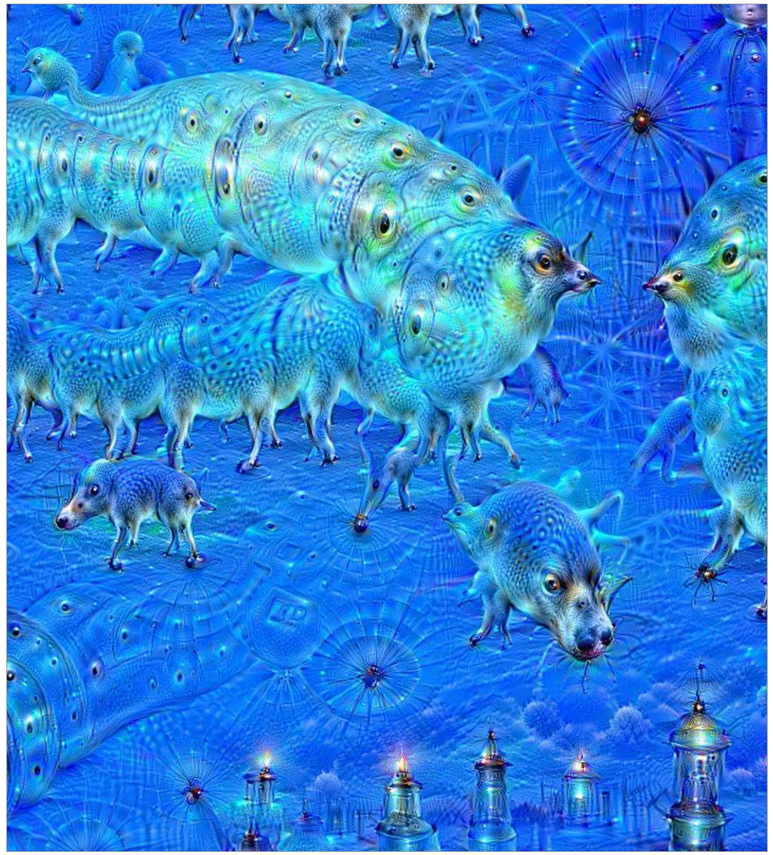
The Rise of Neural Networks

Deep Dreaming

Images by Martin Thoma



Original

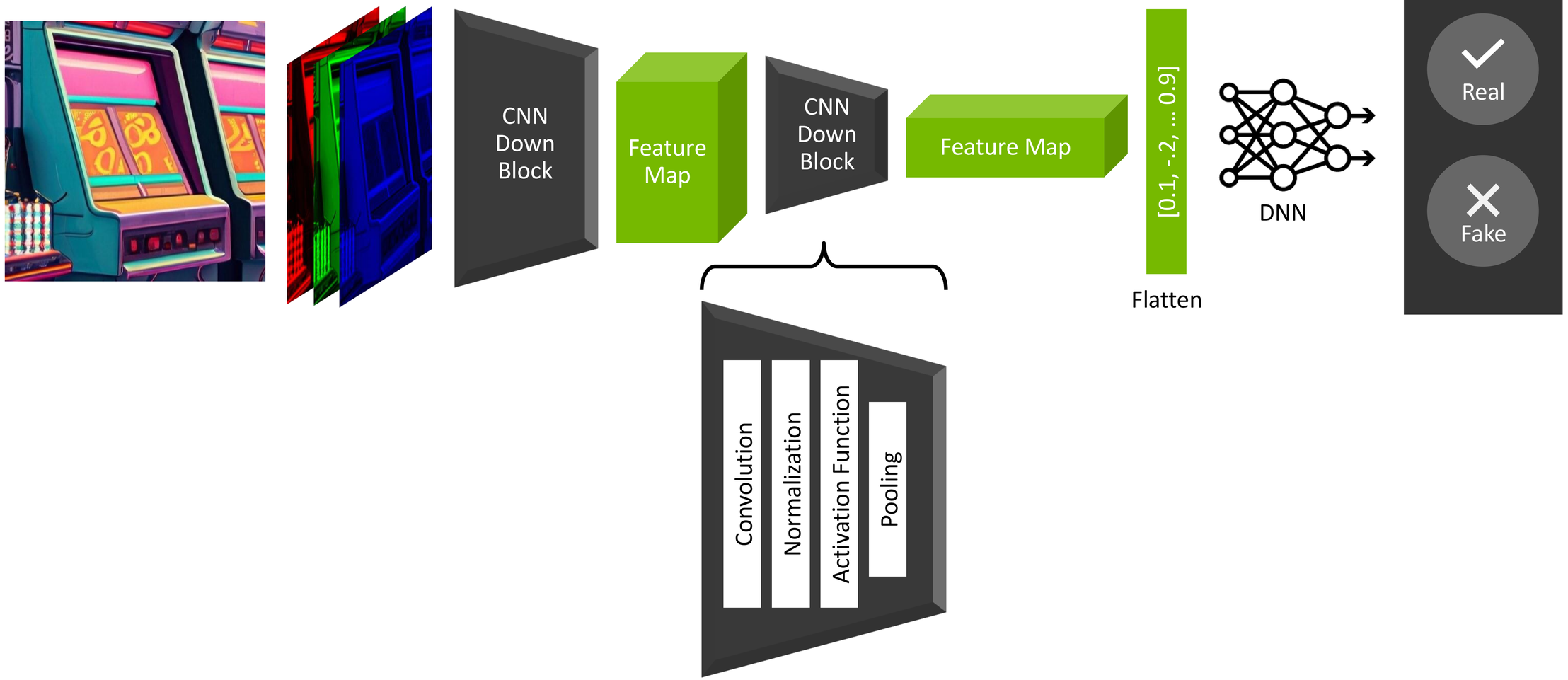


10 Iterations

50 Iterations

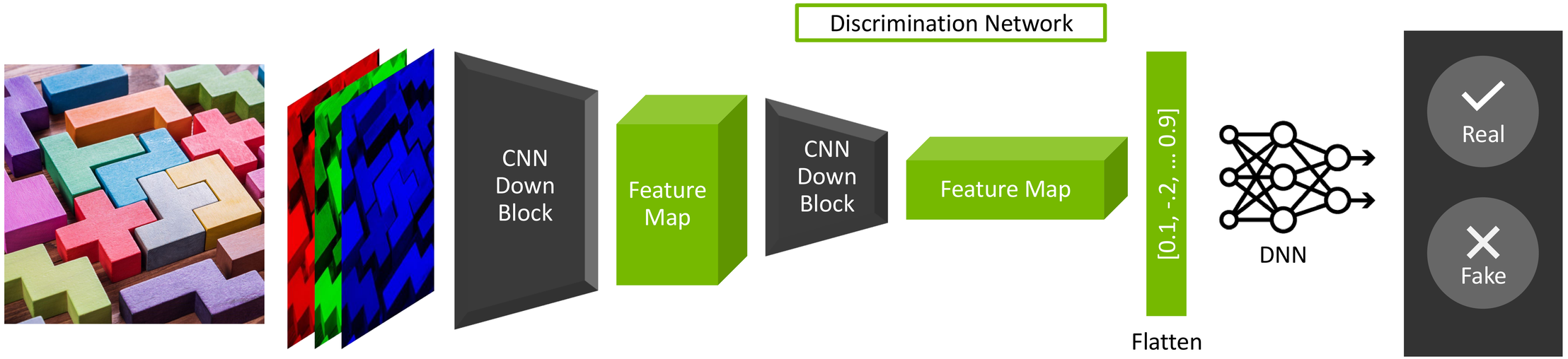
GANs

Generative Adversarial Networks



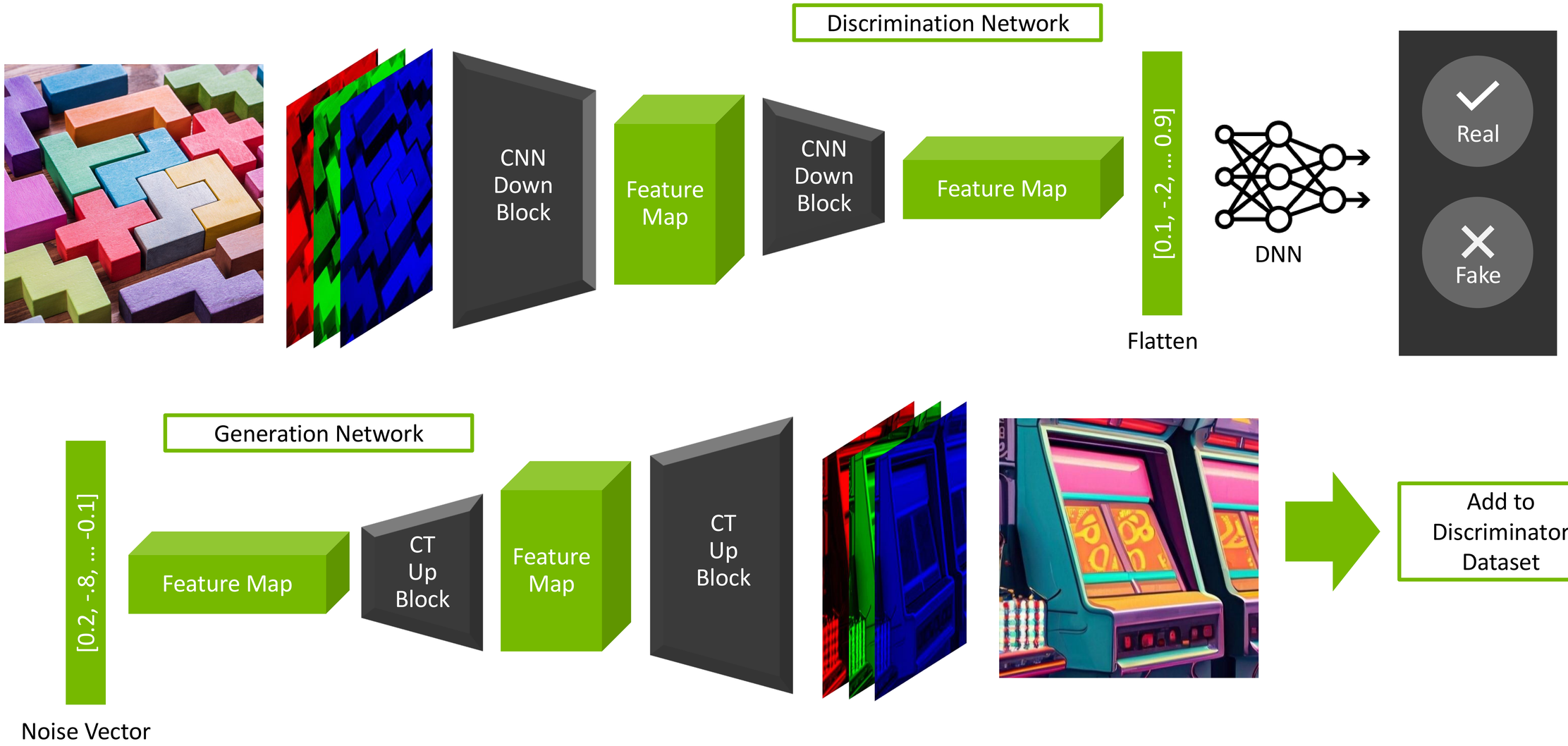
GANs

Generative Adversarial Networks



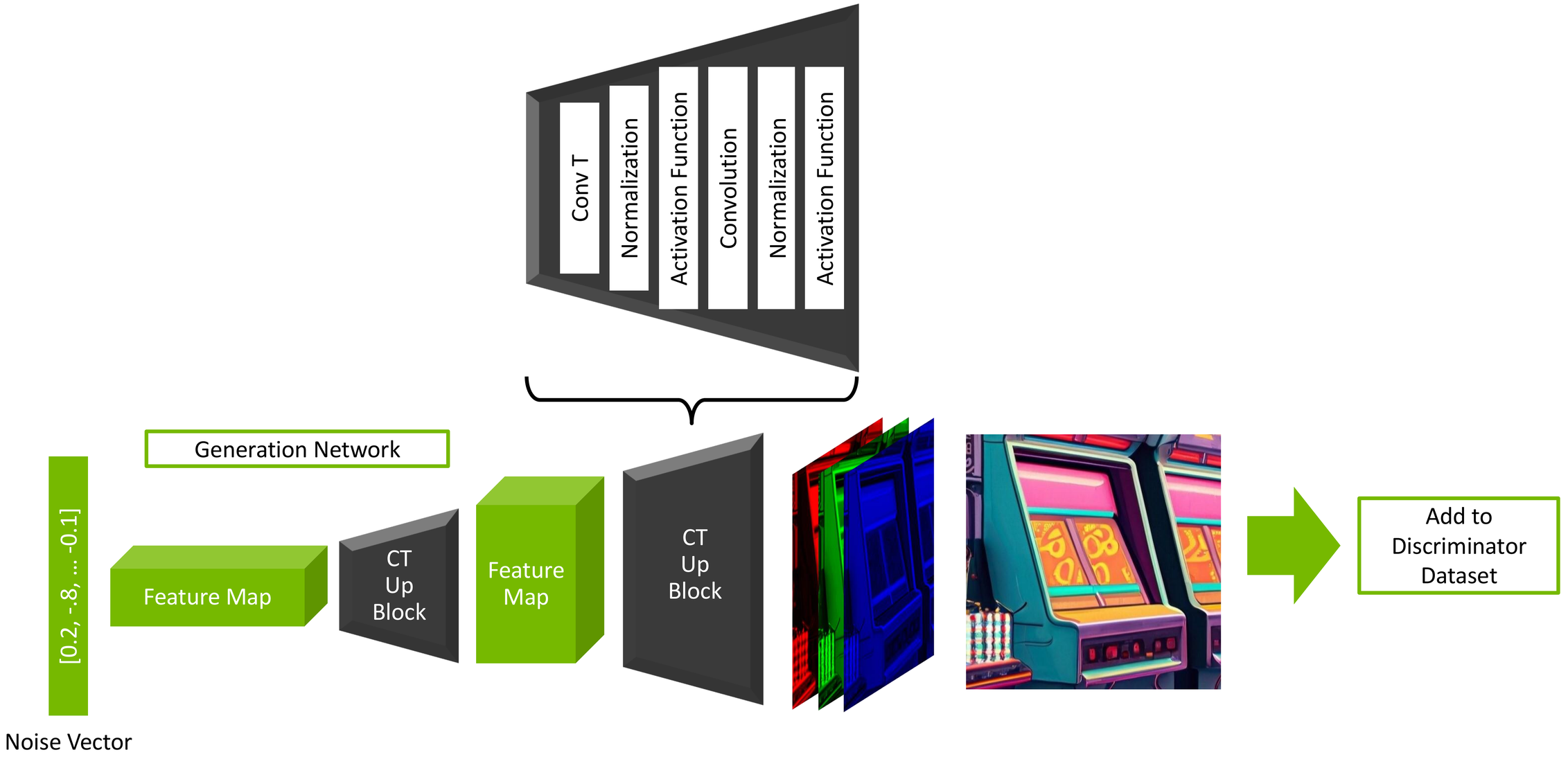
GANs

Generative Adversarial Networks



GANs

Generative Adversarial Networks



GANs

Generative Adversarial Networks

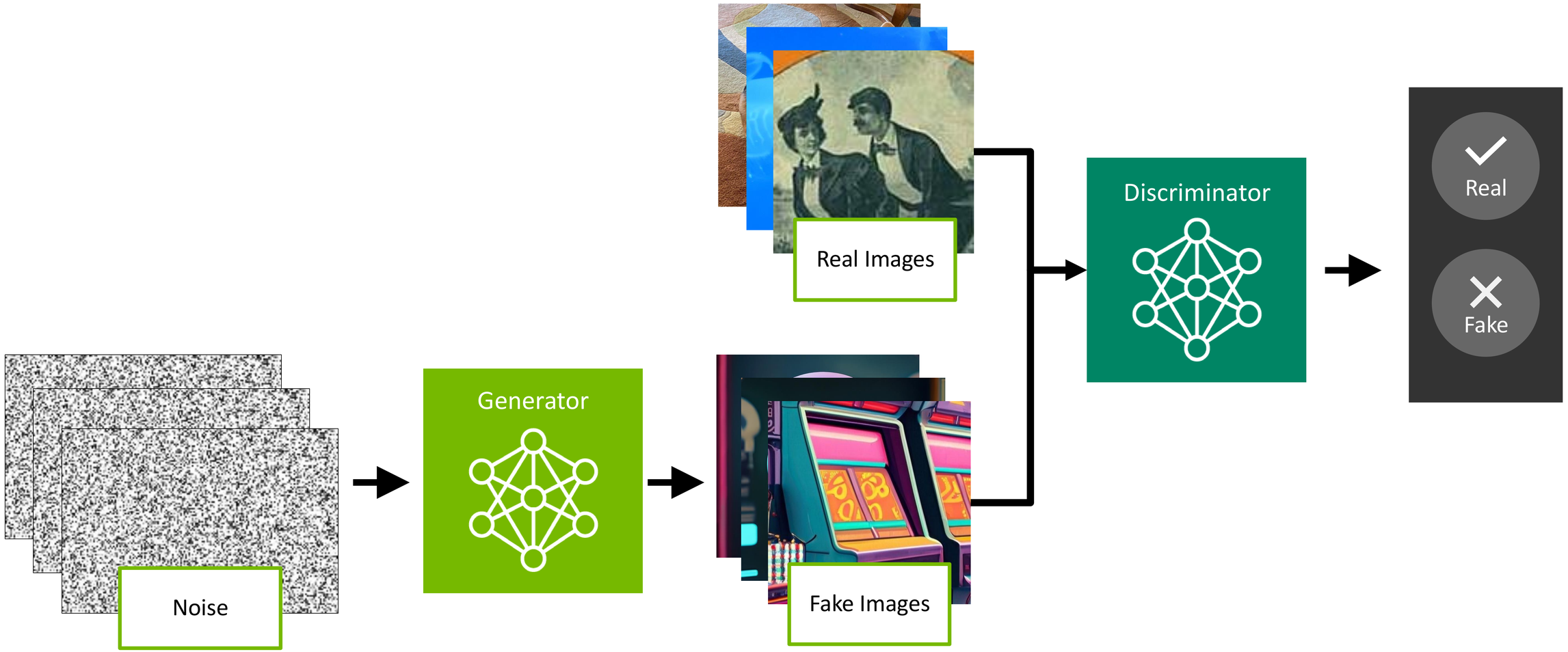


Image Segmentation

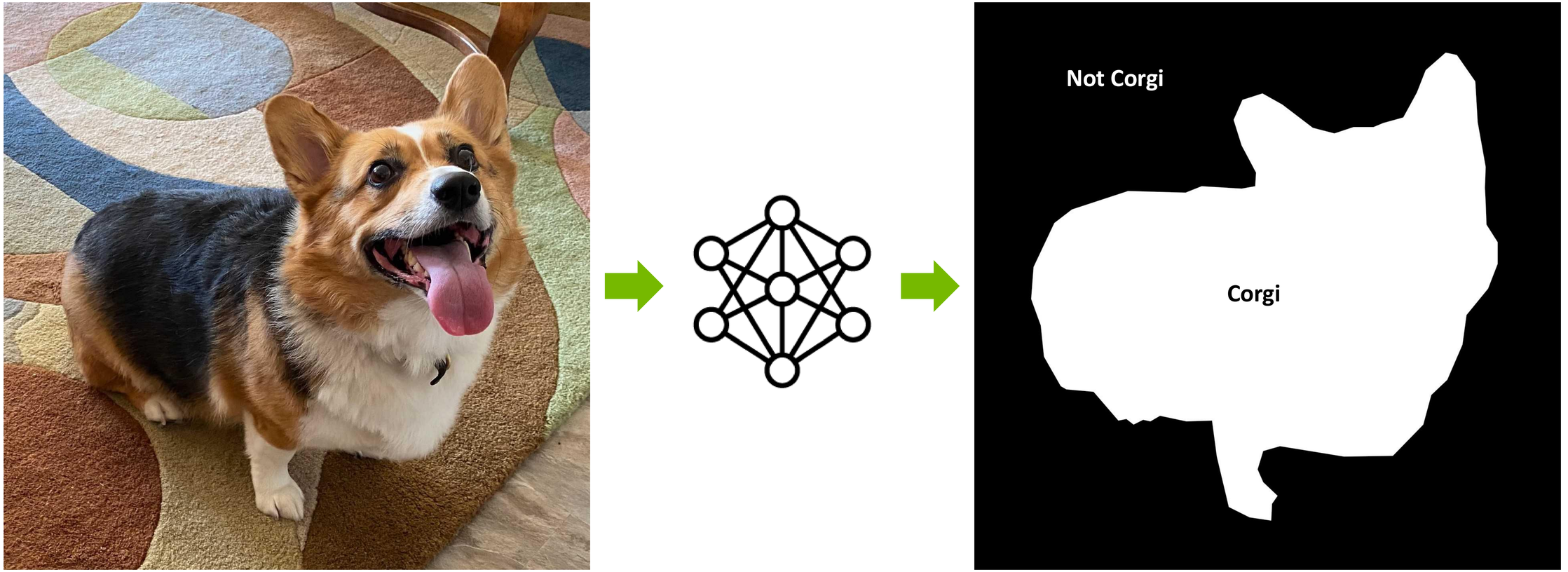
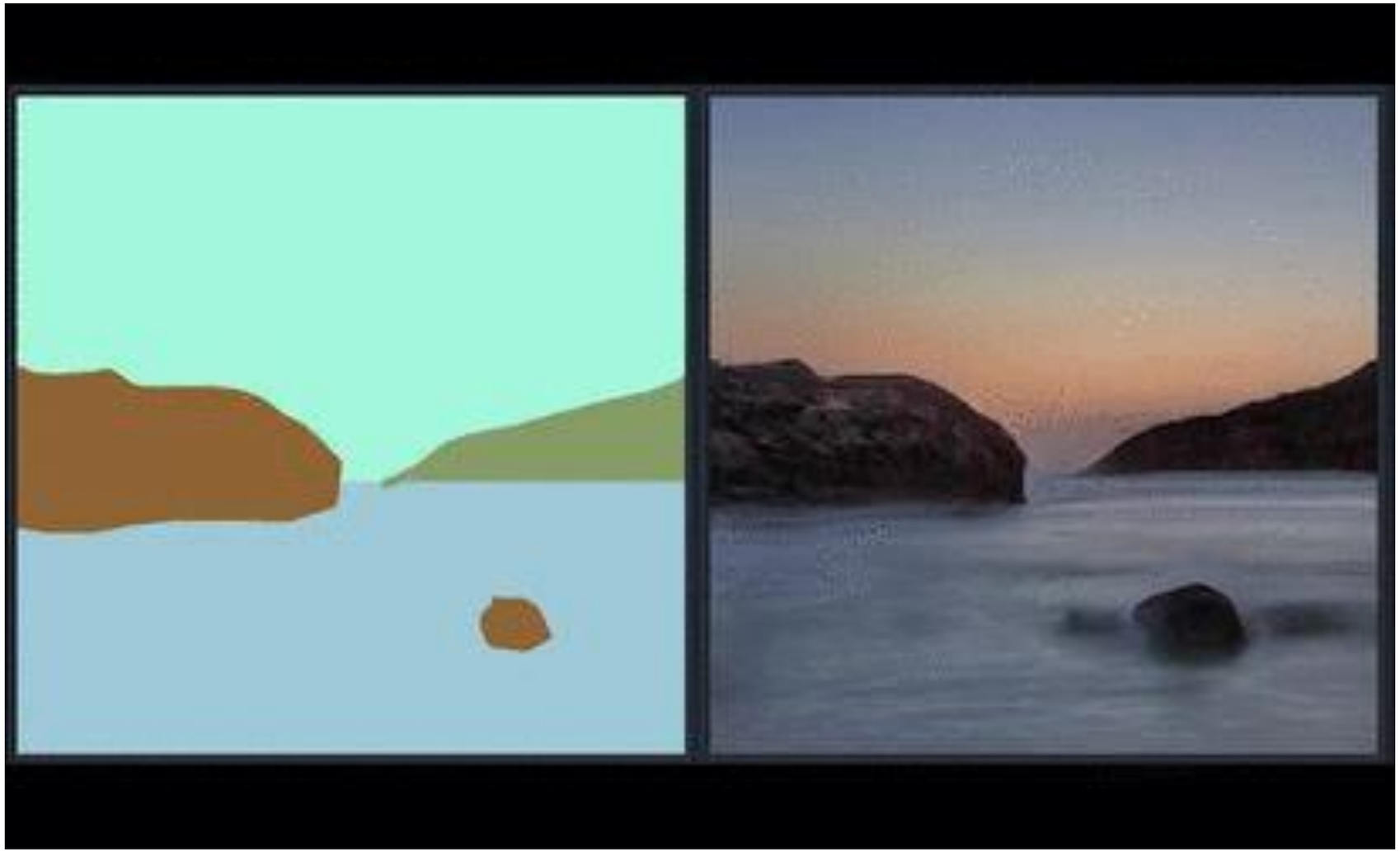


Image Segmentation + GANs

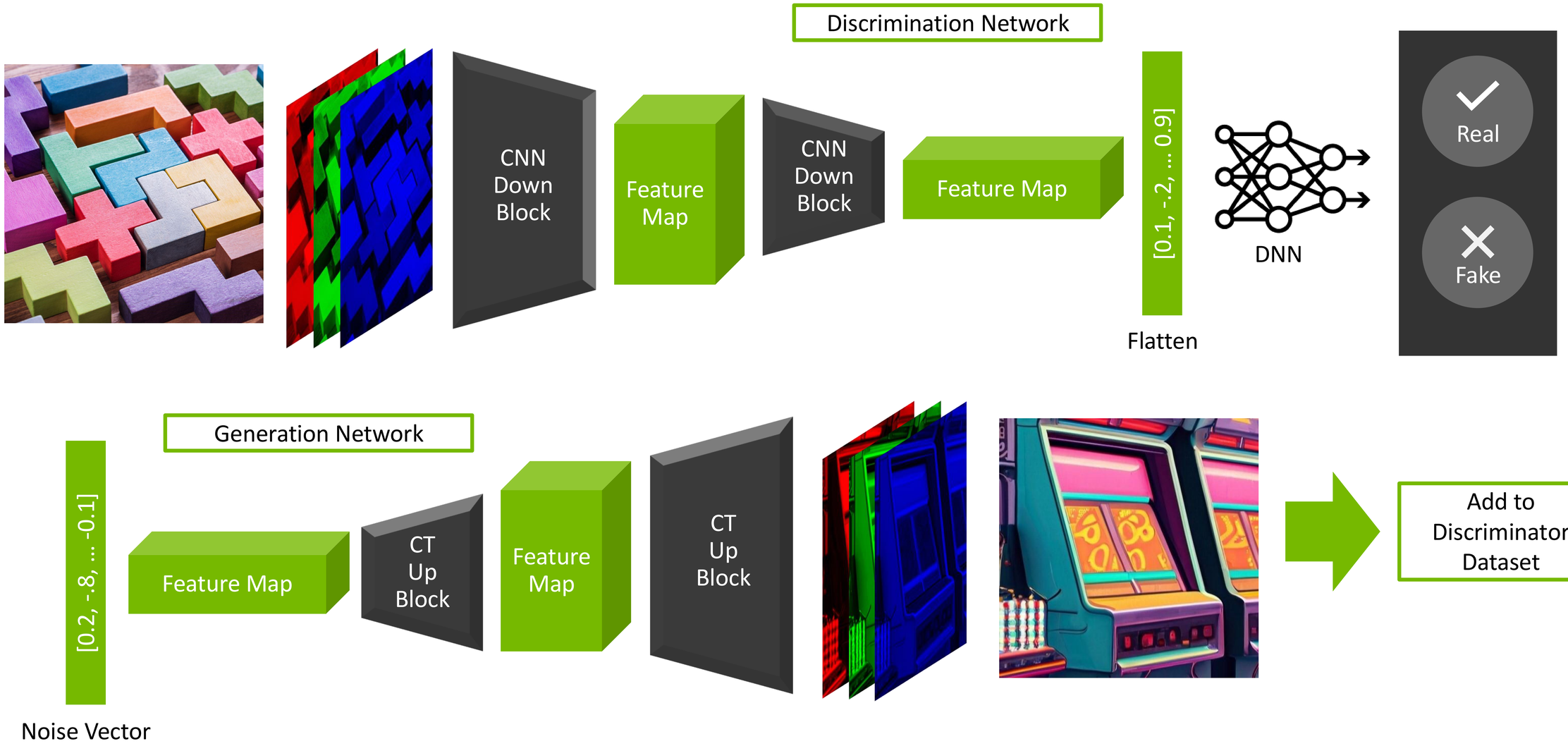
NVIDIA Spade



U-Nets

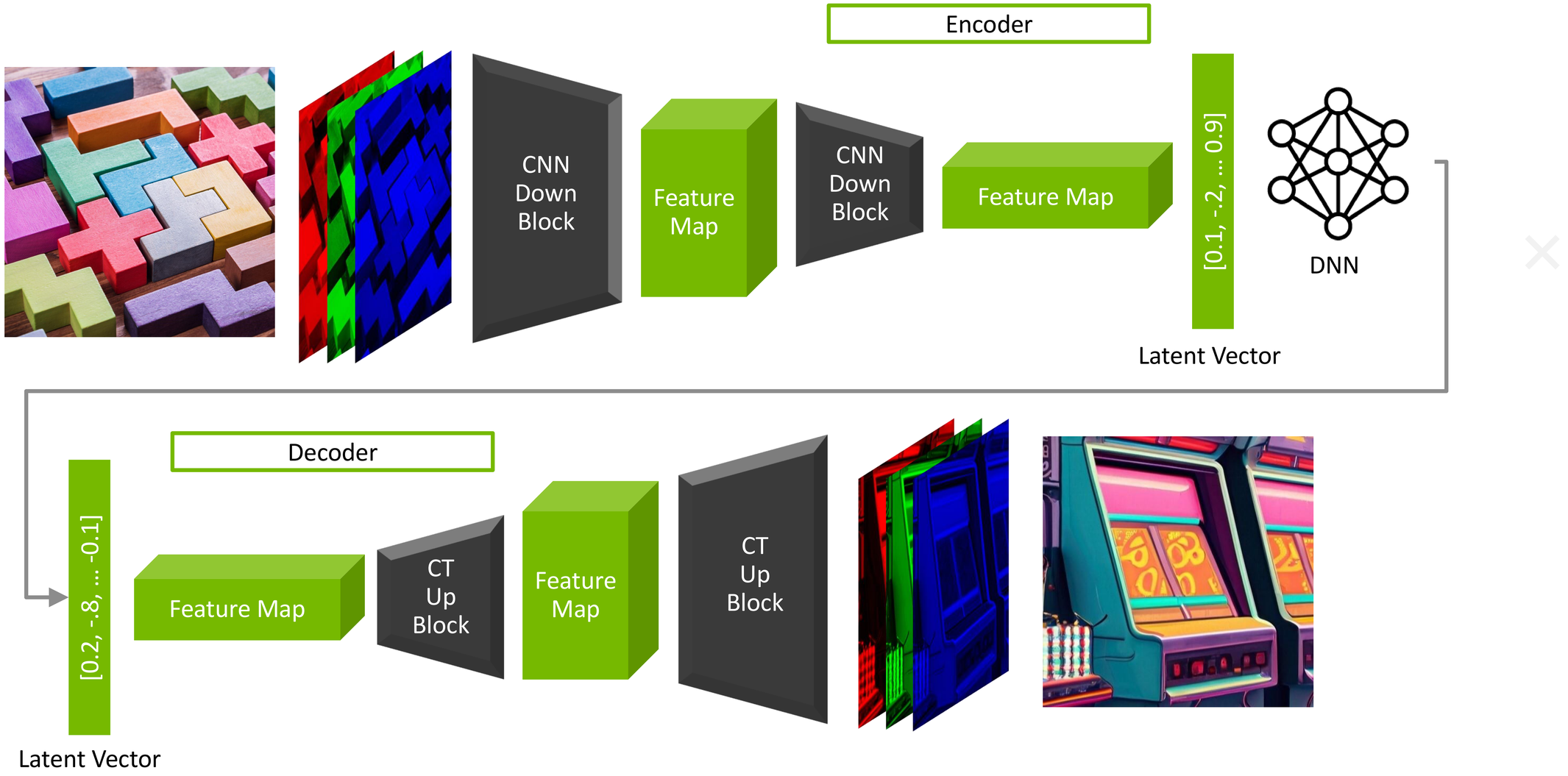
GANs

Generative Adversarial Networks



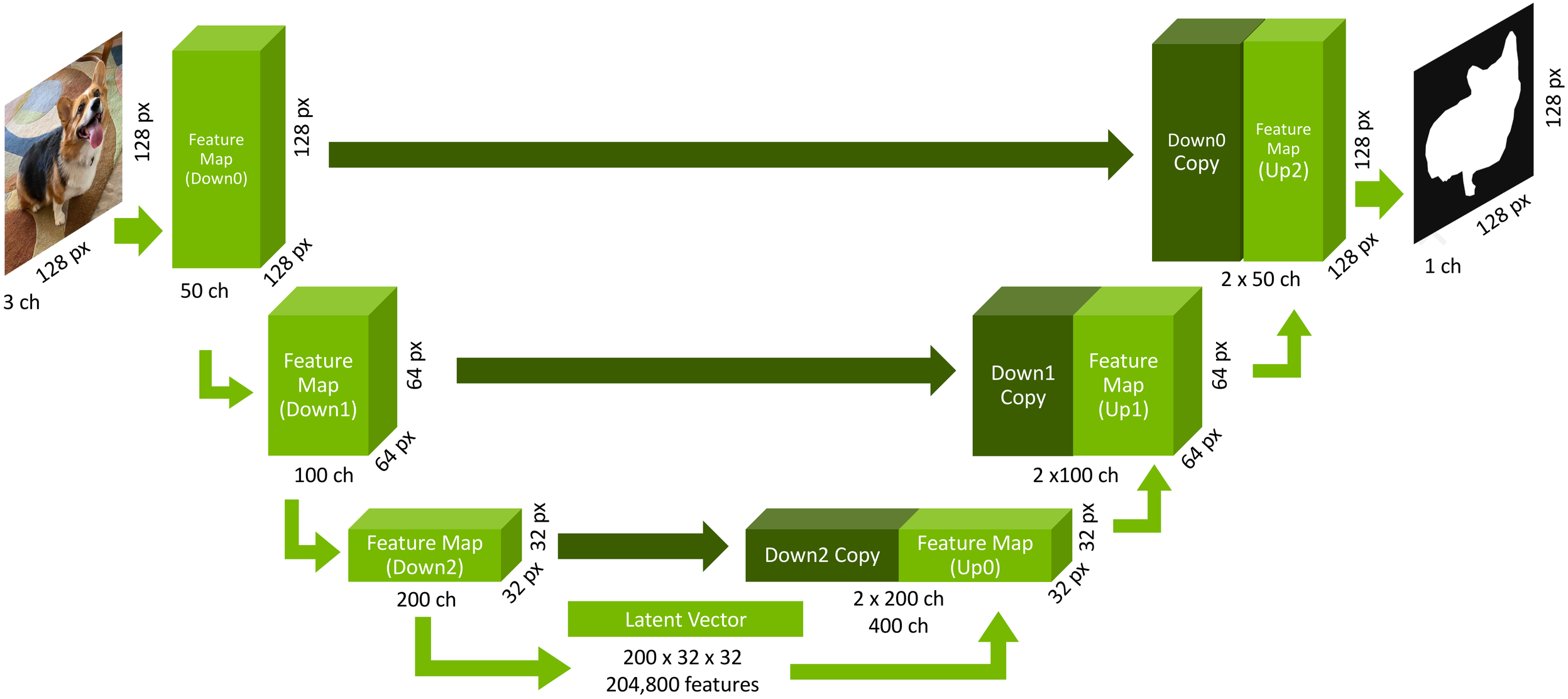
~~GANs~~ U-Nets

The U shaped Autoencoder



U-Nets

The U shaped Autoencoder



Transposed Convolution

Transposed Convolution

Convolution Review

Kernel

.25	.25
.25	.25

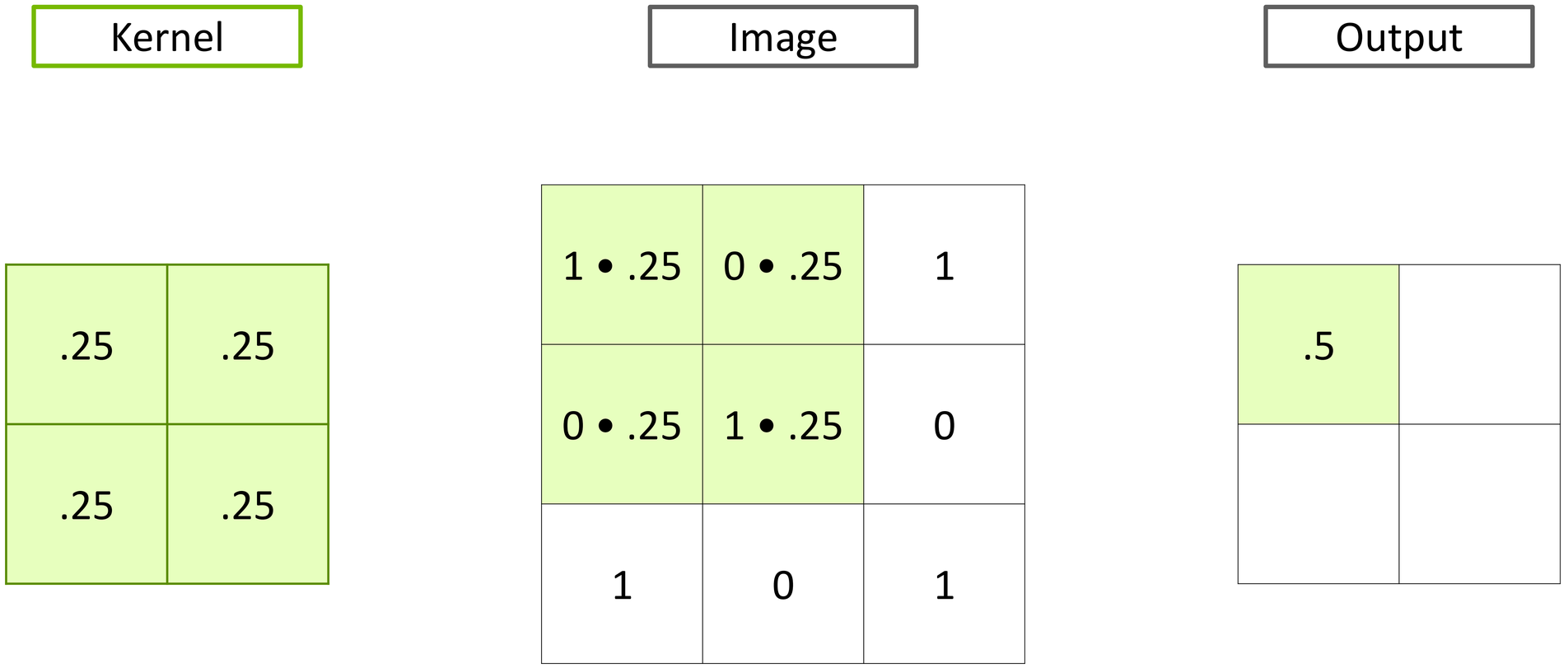
Image

1	0	1
0	1	0
1	0	1

Output

Transposed Convolution

Convolution Review



Transposed Convolution

Convolution Review

Kernel

.25	.25
.25	.25

Image

1	0 • .25	1 • .25
0	1 • .25	0 • .25
1	0	1

Output

.5	.5

Transposed Convolution

Convolution Review

Kernel

.25	.25
.25	.25

Image

1	0	1
0 • .25	1 • .25	0
1 • .25	0 • .25	1

Output

.5	.5
.5	

Transposed Convolution

Convolution Review

Kernel

.25	.25
.25	.25

Image

1	0	1
0	1 • .25	0 • .25
1	0 • .25	1 • .25

Output

.5	.5
.5	.5

Transposed Convolution

Image Upscaling

Kernel

.25	.25
.25	.25

Image

1	0	1
0	1	0
1	0	1

Output

Transposed Convolution

Image Upscaling

Kernel

.25	.25
.25	.25

Image

Stride = 2

1	0	0	0	1
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
1	0	0	0	1

Output

Transposed Convolution

Image Upscaling

Kernel

.25	.25
.25	.25

Image

Stride = 2

1 • .25	0 • .25	0	0	1
0 • .25	0 • .25	0	0	0
0	0	1	0	0
0	0	0	0	0
1	0	0	0	1

Output

.25			

Transposed Convolution

Image Upscaling

Kernel

.25	.25
.25	.25

Image

Stride = 2

1	0 • .25	0 • .25	0	1
0	0 • .25	0 • .25	0	0
0	0	1	0	0
0	0	0	0	0
1	0	0	0	1

Output

.25	0		

Transposed Convolution

Image Upscaling

Kernel

.25	.25
.25	.25

Image

Stride = 2

1	0	0	0	1
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
1	0	0	0	1

Output

.25	0	0	.25
0	.25	.25	0
0	.25	.25	0
.25	0	0	.25

Transposed Convolution

Stride

Image

Stride = 2

1	0	0	0	1
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
1	0	0	0	1

Image

Stride = 3

1	0	0	0	0	0	1
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
1	0	0	0	0	0	1

Transposed Convolution

Padding

Image

Stride = 3

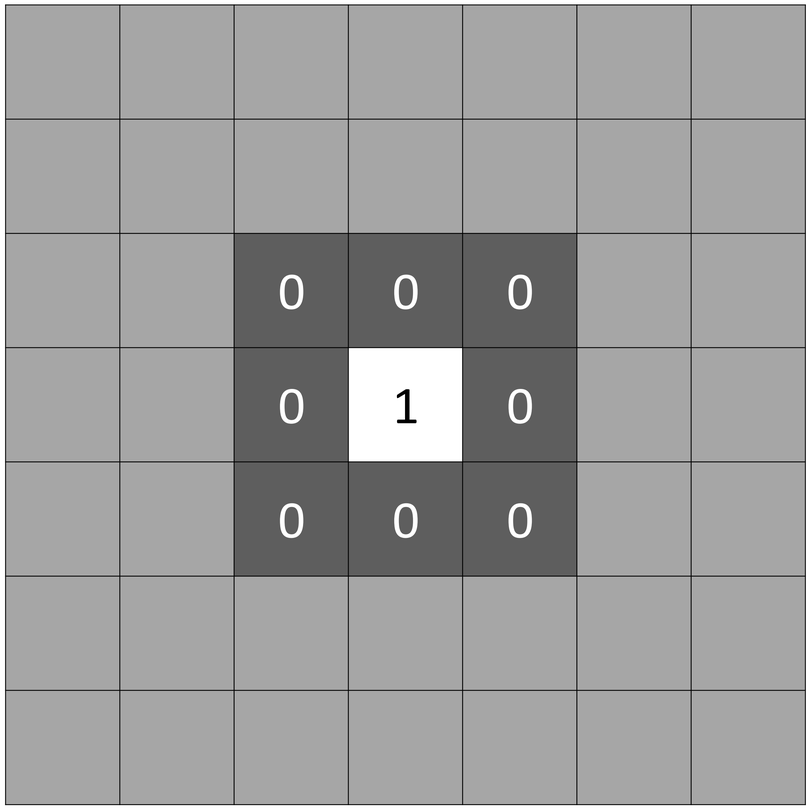
Padding = 0

1	0	0	0	0	0	1
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
1	0	0	0	0	0	1

Image

Stride = 3

Padding = 2



Transposed Convolution

Padding

Image

Stride = 3

Padding = 0

1	0	0	0	0	0	1
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
1	0	0	0	0	0	1

Image

Stride = 3

Padding = 2

0	0	0
0	1	0
0	0	0

Transposed Convolution

Out Padding

Image

Out Padding = 0

1	0	1
0	1	0
1	0	1

Image

Out Padding = 1

1	0	1	0
0	1	0	0
1	0	1	0
0	0	0	0

Image

Out Padding = 2

1	0	1	0	0
0	1	0	0	0
1	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Image Resizing

Upsampling

64 px

64 px

128 px

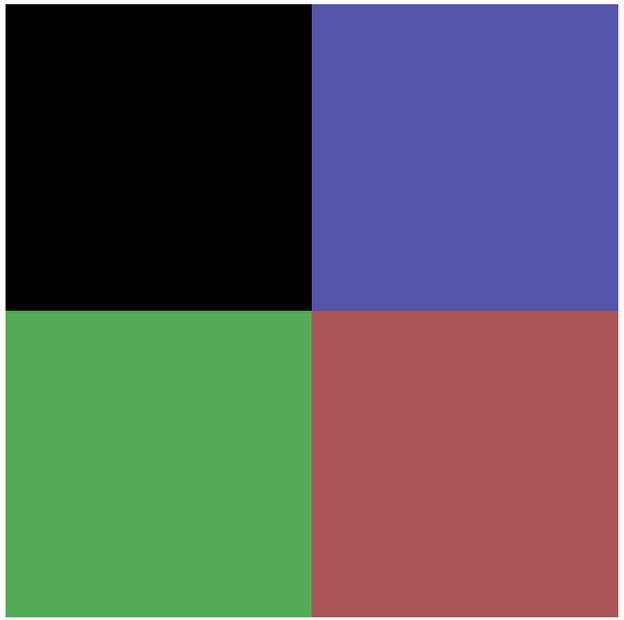
128 px

192 px

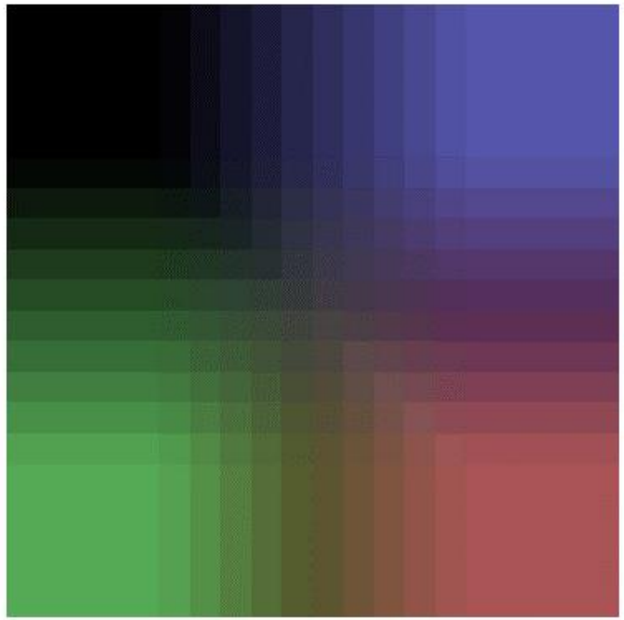
192 px

Image Resizing

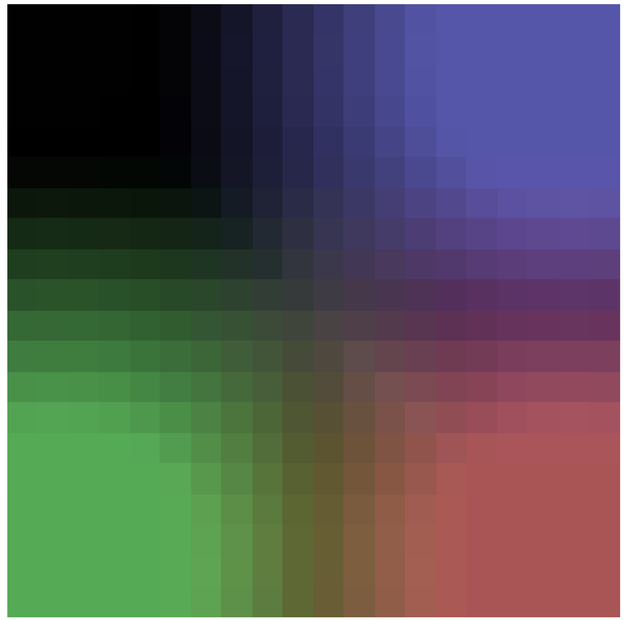
Upsampling



Nearest



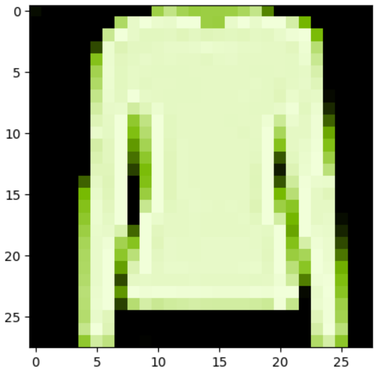
Bilinear



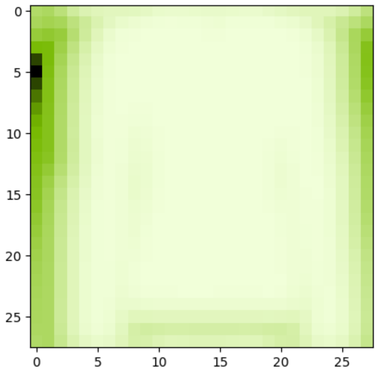
Bicubic

Deconvolution?

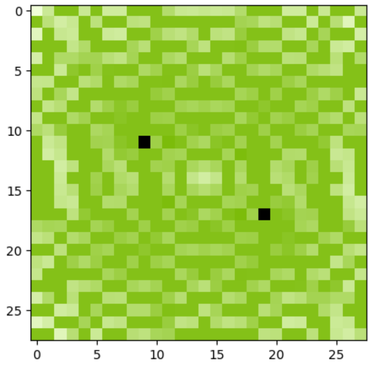
Same as Transposed Convolution?



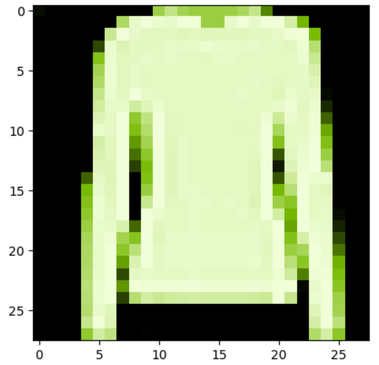
Image



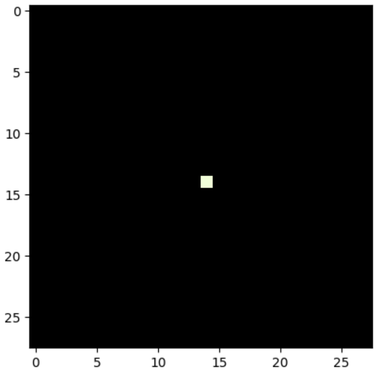
Convolved Image



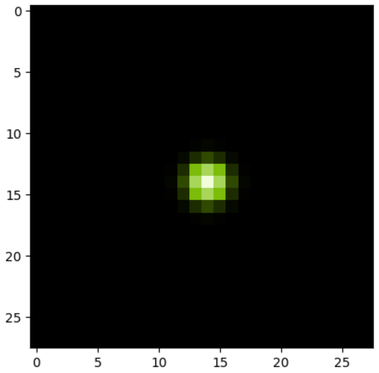
FFT Convolved



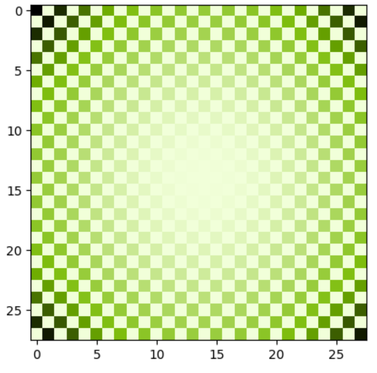
Deconvolved Convolved Image



Point



Point Spread Function

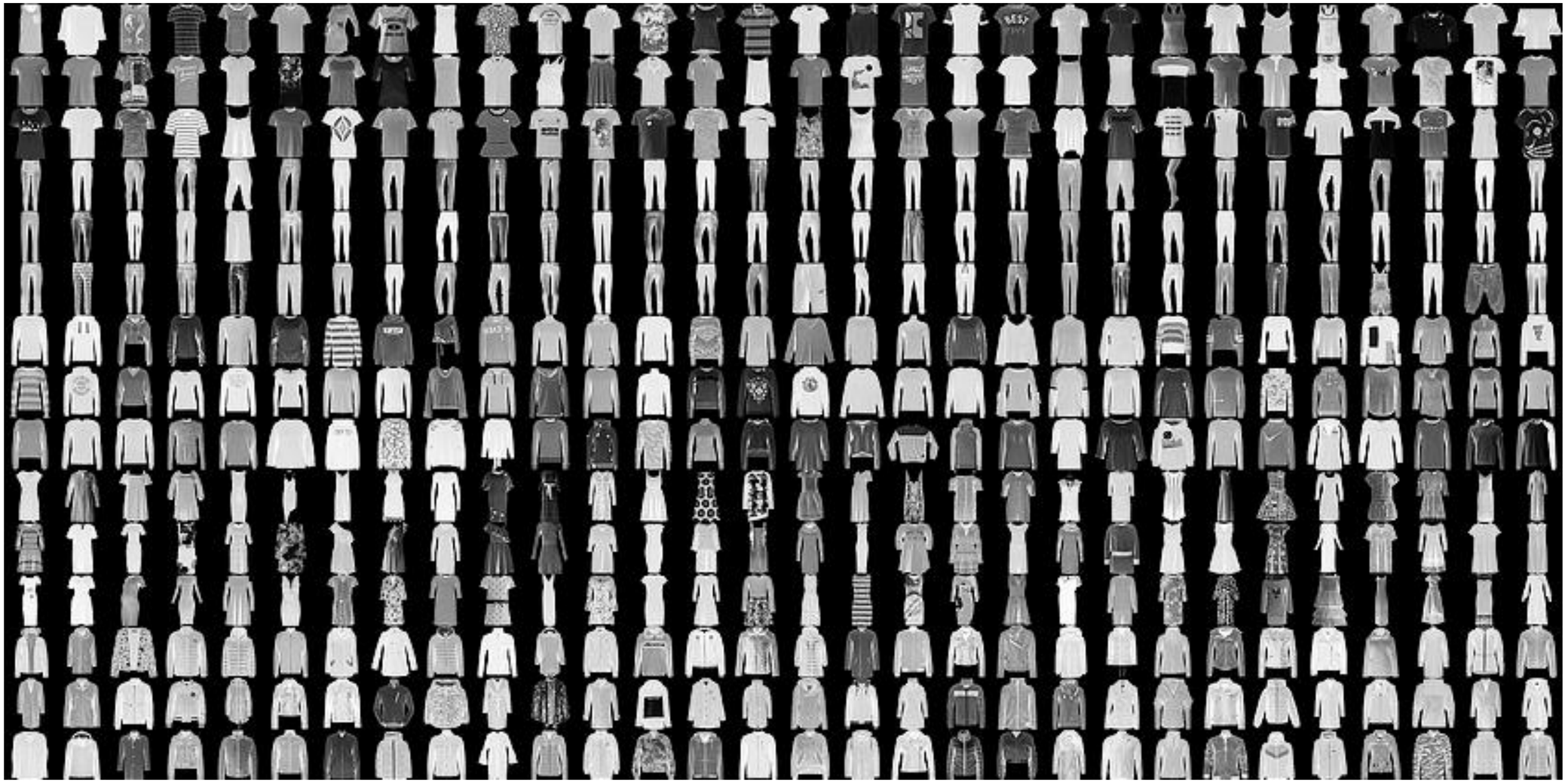


FFT Point Spread Function

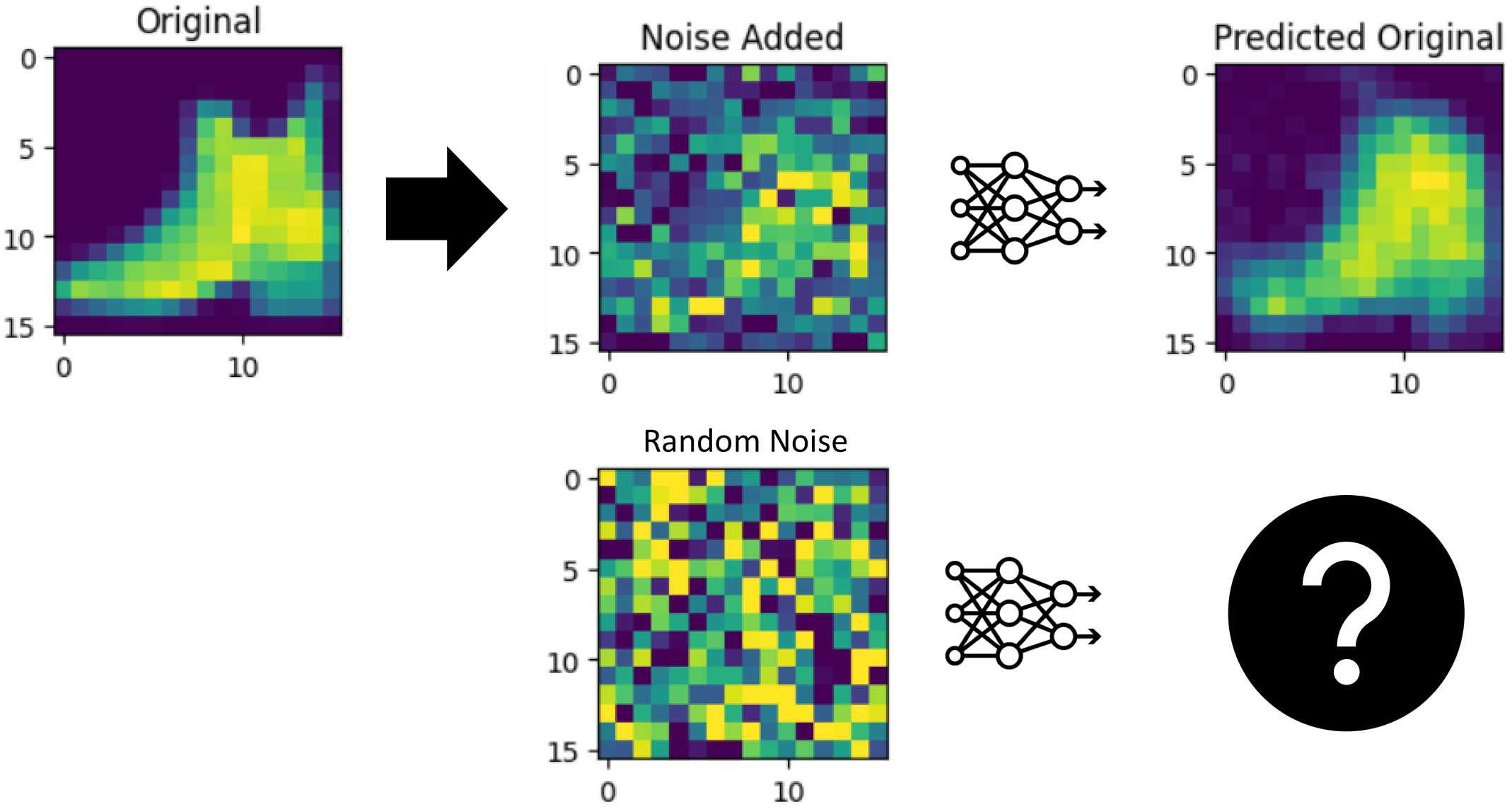
Lab

FashionMNIST

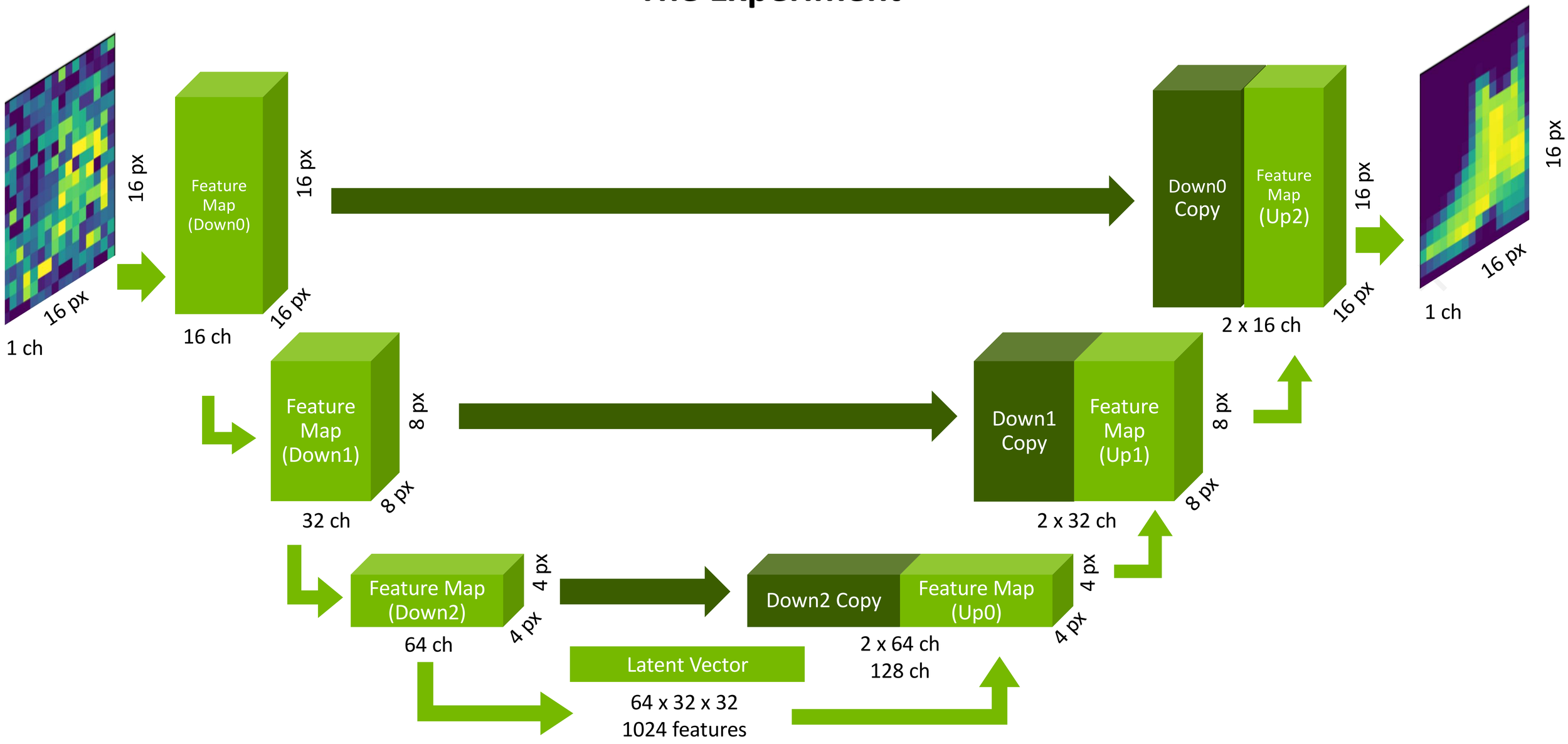
Convolutional Neural Network “Hello World”



Hypothesis: Generate an image from Noise



The Experiment

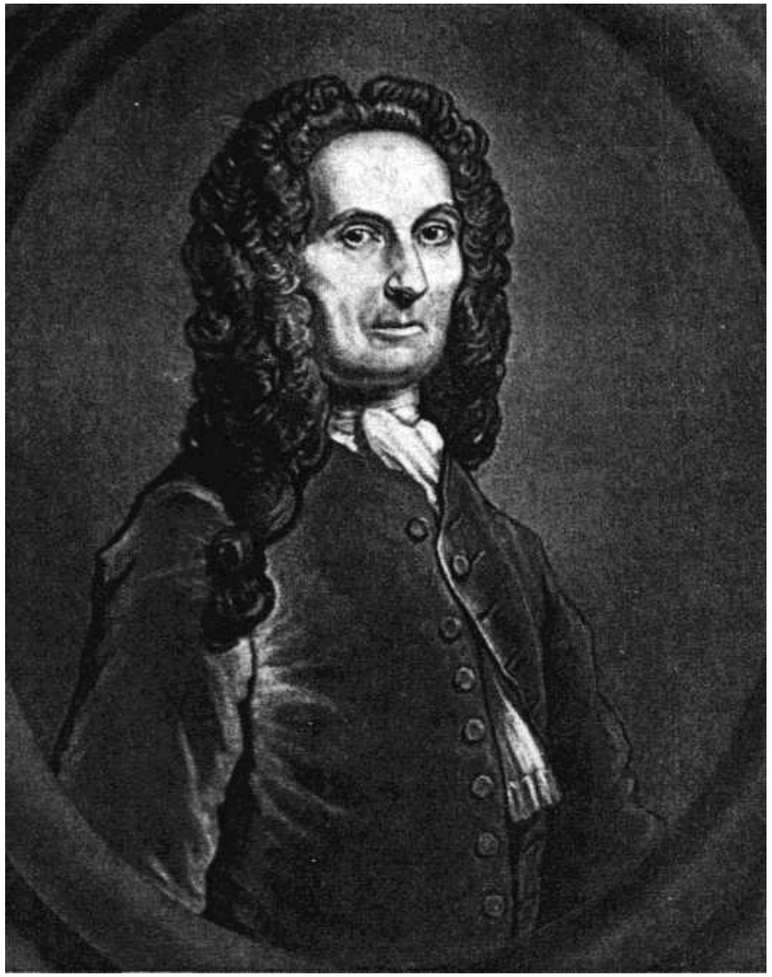


Let’s get started!

Appendix:
The Normal Distribution

De Moivre

From Coin Flips to Bells



$$\Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

$$\Pr(X = k) = \frac{n!}{k!\,(n-k)!} p^k (1-p)^{n-k}$$

$$\Pr(X = 2) = \frac{4!}{2!\,(4-2)!} \left(\frac{1}{2}\right)^2 (1-\frac{1}{2})^{4-2}$$

$$\Pr(X = 2) = \frac{4 \cdot 3 \cdot 2 \cdot 1}{2 \cdot 1 \cdot 2 \cdot 1} \left(\frac{1}{4}\right)\left(\frac{1}{4}\right)$$

$$\Pr(X = 2) = \frac{6}{16}$$

$$p = \frac{1}{2}$$

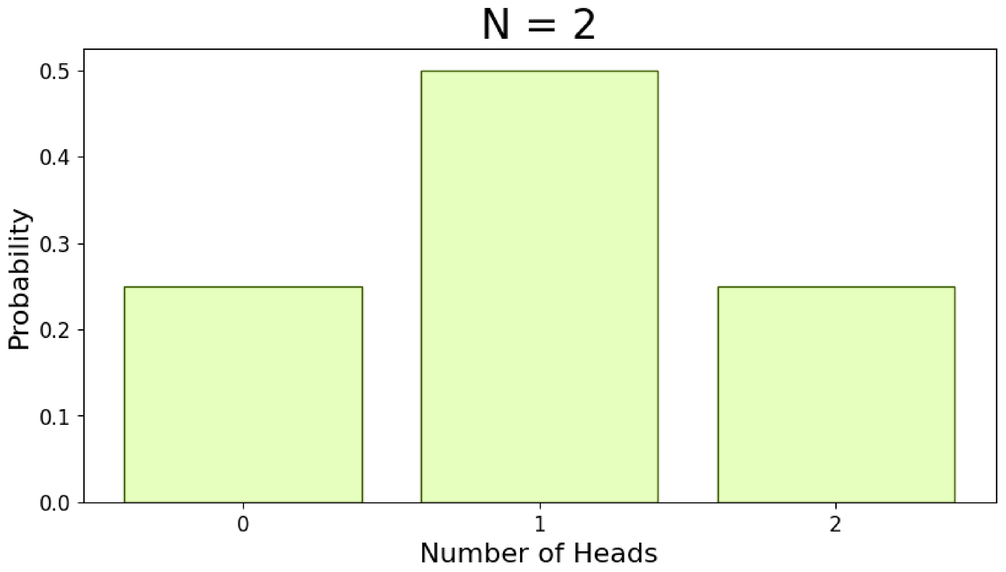
$$n = 4$$

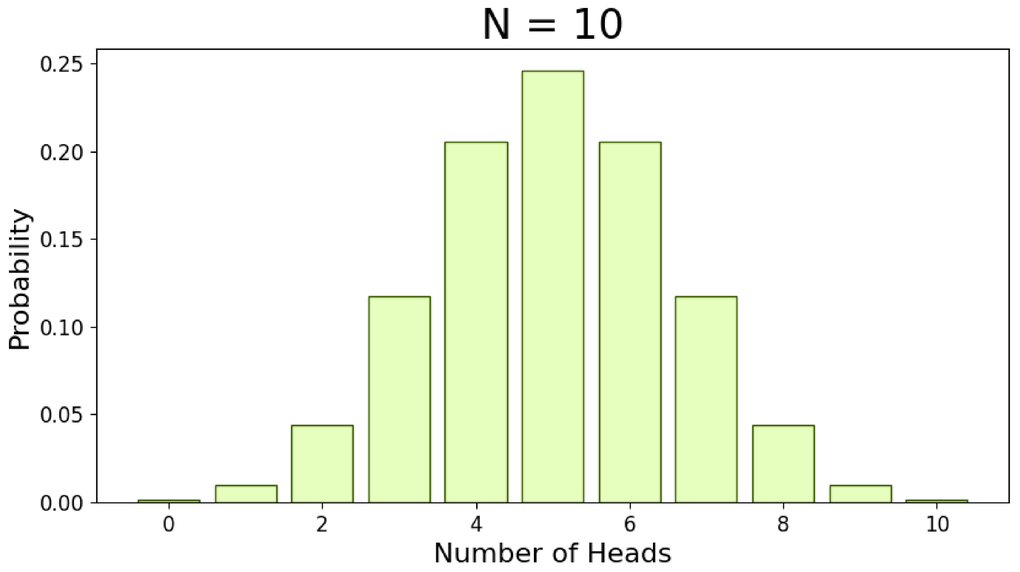
$$k = 2$$

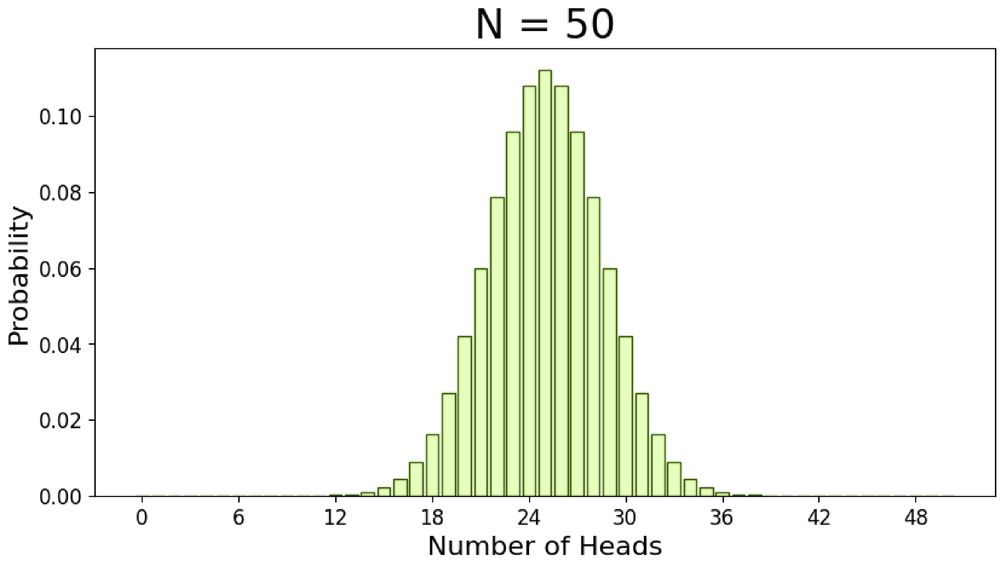
A weighted coin flipping through the air like a cartoon

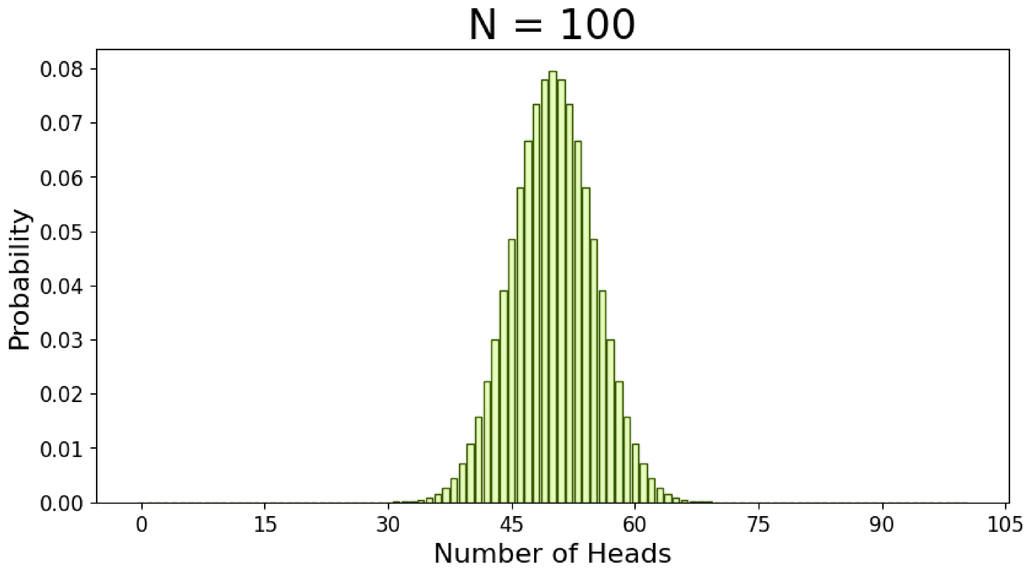
De Moivre

From Coin Flips to Bells



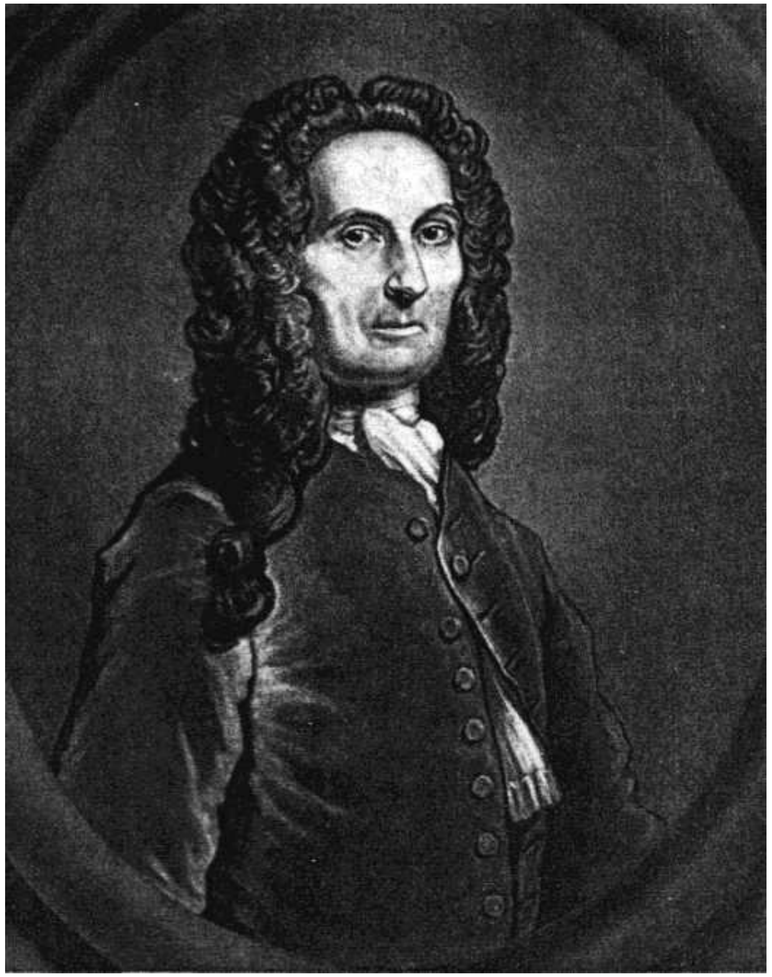






De Moivre

From Coin Flips to Bells



$$\Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

$$\Pr(X = k) = \frac{n!}{k!\,(n-k)!} p^k (1-p)^{n-k}$$

$$n! \approx \sqrt{2\pi}\left(\frac{n}{e}\right)^n$$

$$\binom{n}{k} p^k q^{n-k} \simeq \frac{1}{\sqrt{2\pi npq}} e^{-\frac{(k-np)^2}{2npq}}$$

$$N(\mathrm{x};\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

A weighted coin flipping through the air like a cartoon

De Moivre

From Coin Flips to Bells

$$N(\mathrm{x};\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$\mu = mean, a.k.a. average$ $\sigma = standard\ devaition, a.k.a. spread$

$$z = \frac{x\ -\mu}{\sigma}$$