GAN мрежи

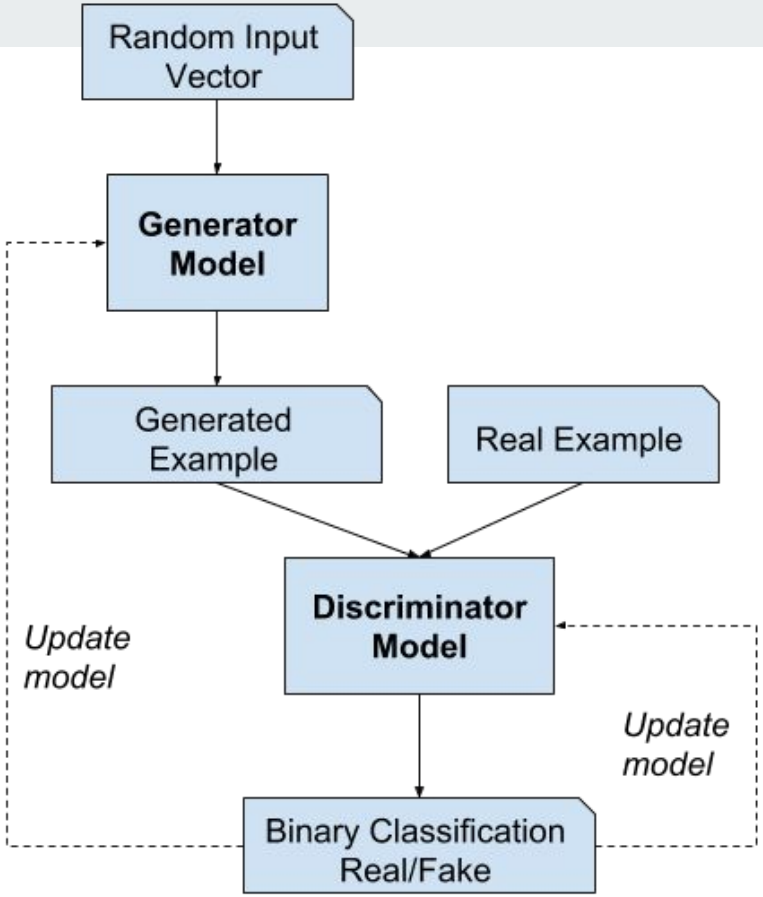
Любомир Ружински, ФН: 0MI3400156

GAN architecture

- The generator model takes a fixed-length random vector as input and generates a sample in the domain.

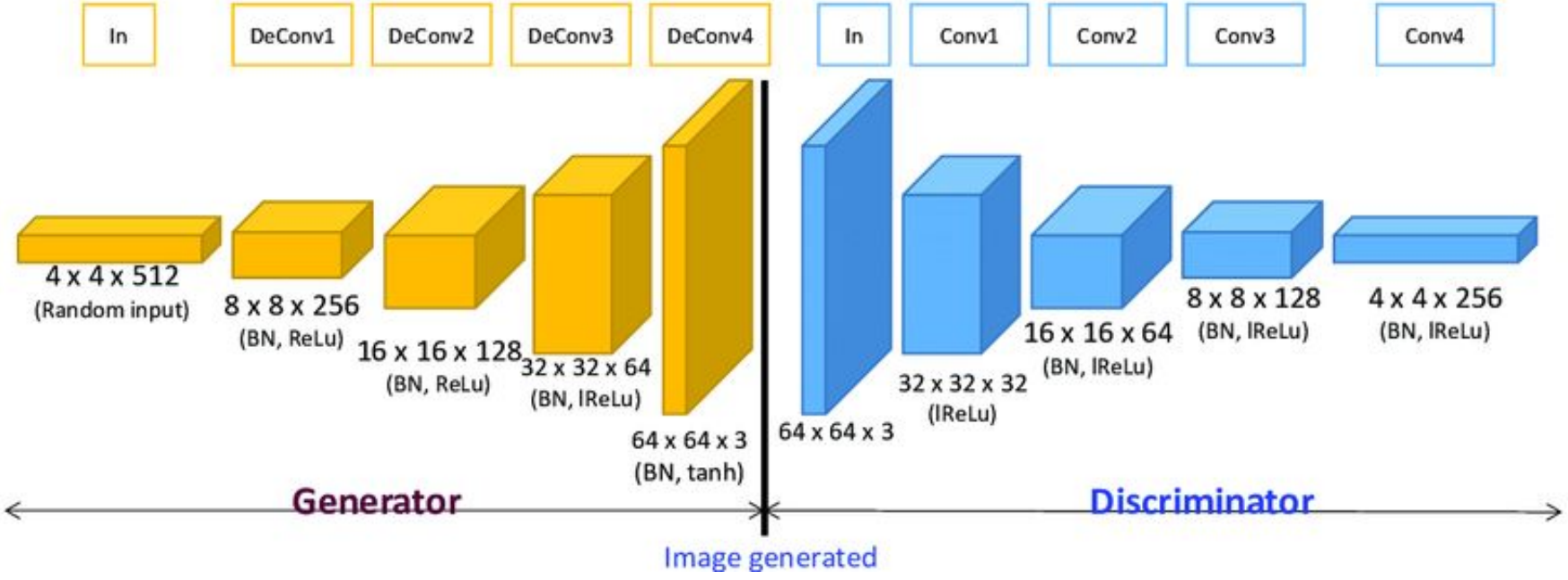
- The discriminator model takes an example from the domain as input (real or generated) and predicts a binary class label of real or fake (generated).

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$



Deep Convolutional GANs

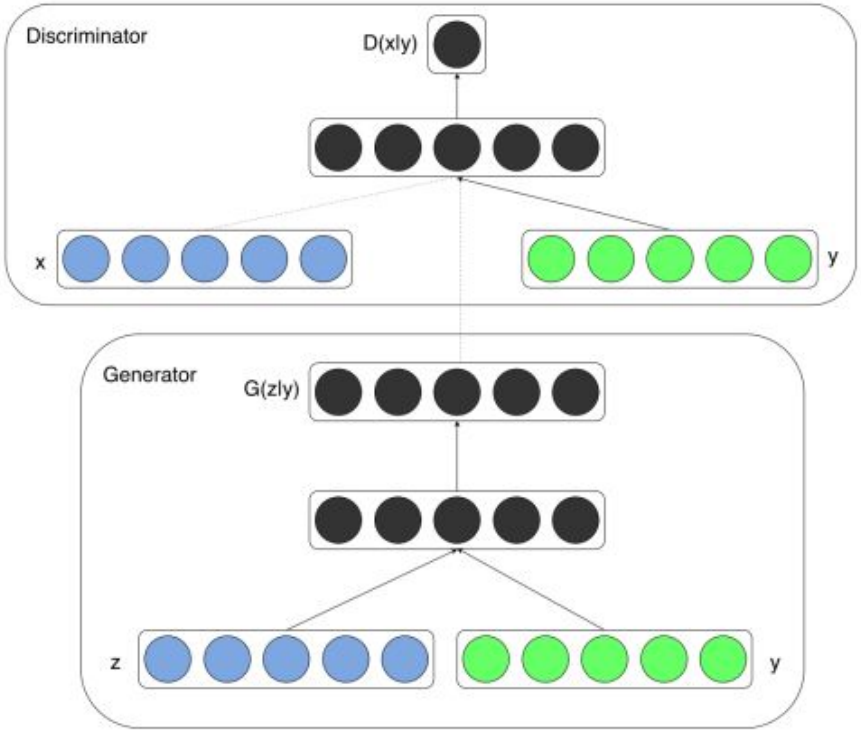
- Use deep CNNs for the generator and the discriminator.
  - CONV and Transposed CONV layers



Conditional GANs

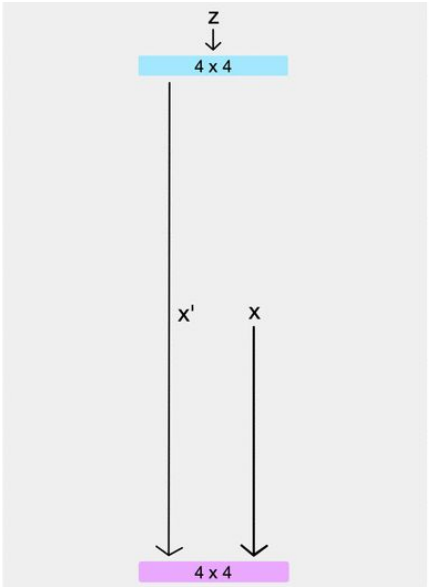
- In an unconditioned generative model, there is no control on modes of the data being generated.
- By conditioning the model on additional information it is possible to direct the data generation process. Such conditioning could be based on class labels.

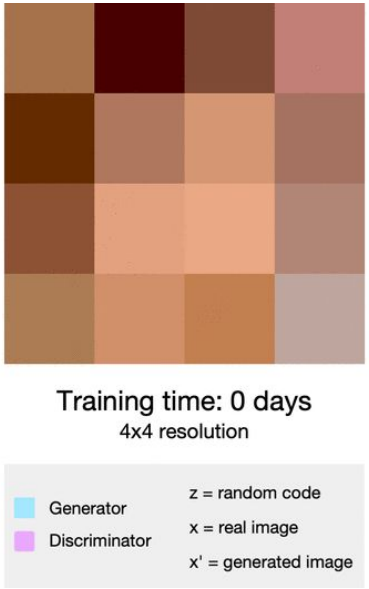
Figure 2: Generated MNIST digits, each row conditioned on one label



StyleGAN

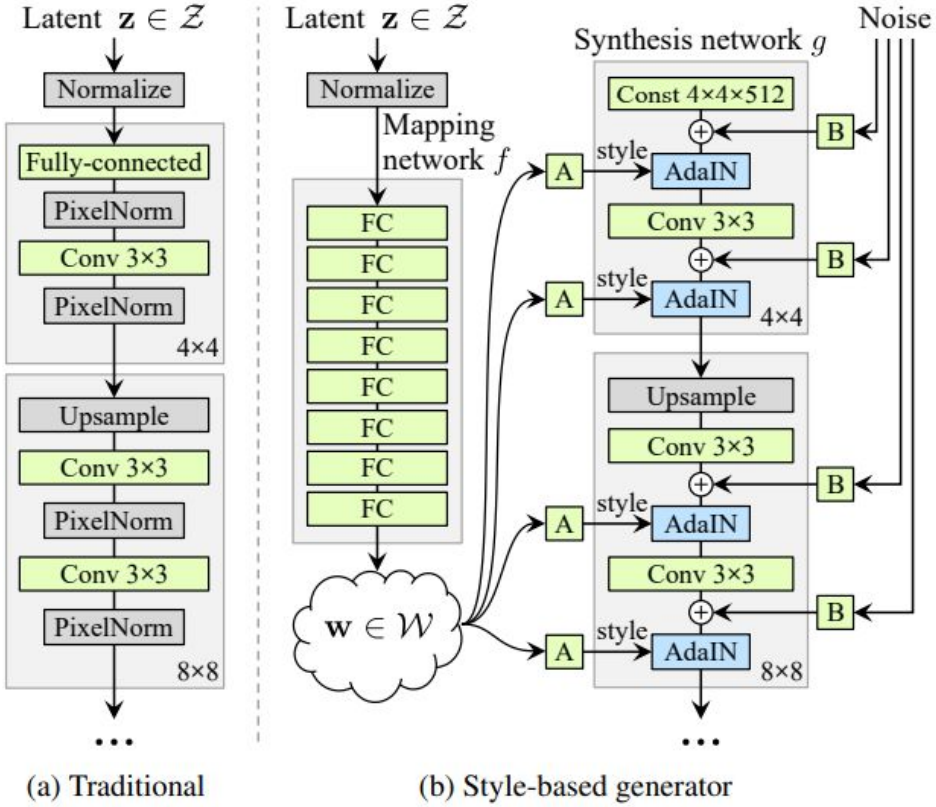
- An architecture by Nvidia which allows controlling the "style" of the GAN output by applying adaptive instance normalization at different layers of the network.
- Using new generator architecture.
- Incremental growing architecture.





Generator

- The Mapping Network's goal is to encode the input vector into an intermediate vector whose different elements control different visual features.
- The lower the layer (and the resolution), the coarser the features it affects:
  - Coarse - resolution of up to 8 - affects pose, general hair style, face shape, etc
  - Middle - resolution of 16 to 32 - affects finer facial features, hair style, eyes open/closed, etc.
  - Fine - resolution of 64 to 1024 - affects color scheme (eye, hair and skin) and micro features.

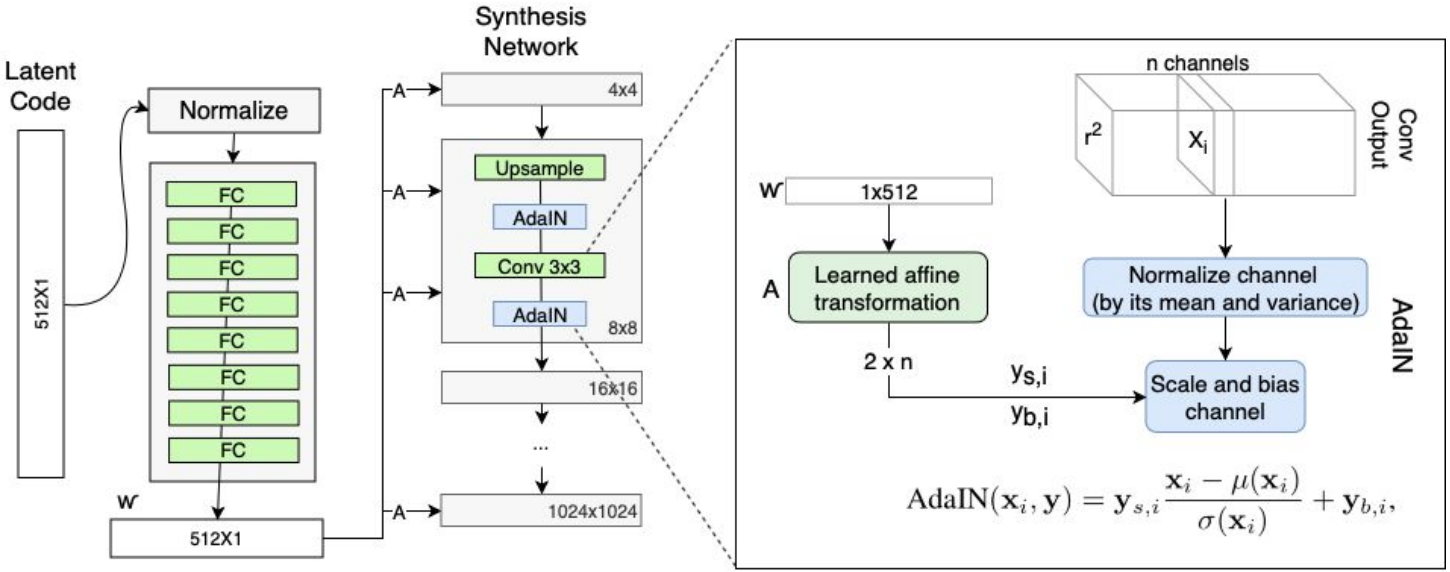


Adaptive Instance Normalization

- Each AdaIN block takes as input the latent style w and the feature map x.

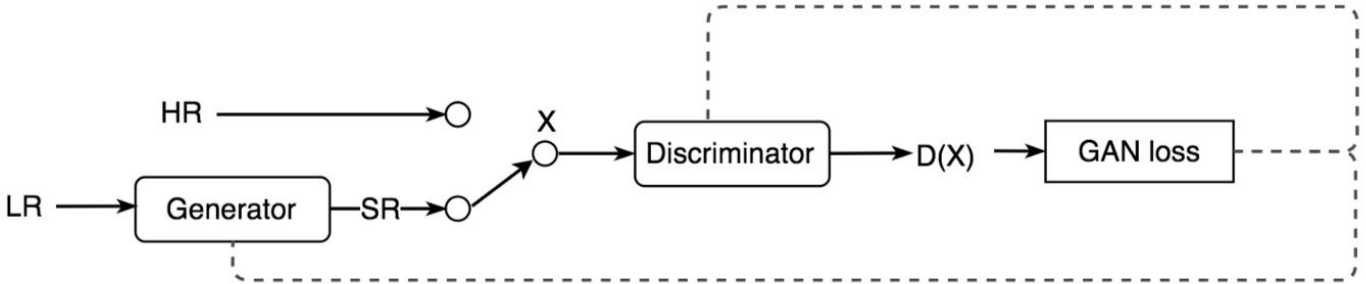
- An affine layer (fully connected with no activation, block A in the figure) converts the style to a mean and standard deviation . Then the feature map is shifted and scaled to have this mean and standard deviation.

- demo



Super Resolution GAN (SRGAN)

- SRGAN applies a deep network in combination with an adversary network to produce higher resolution images
- During the training, A high-resolution image (HR) is downsampled to a low-resolution image (LR). A GAN generator upsamples LR images to super-resolution images (SR). We use a discriminator to distinguish the HR images and backpropagate the GAN loss to train the discriminator and the generator.



Architecture

It mostly composes of convolution layers, batch normalization and parameterized ReLU (PReIU). The generator also implements skip connections similar to ResNet.

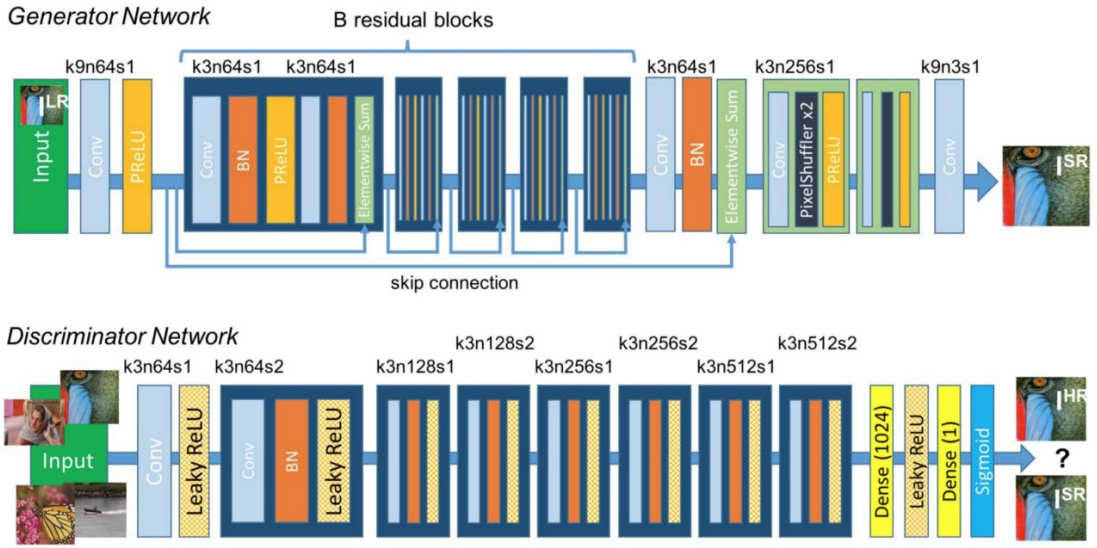
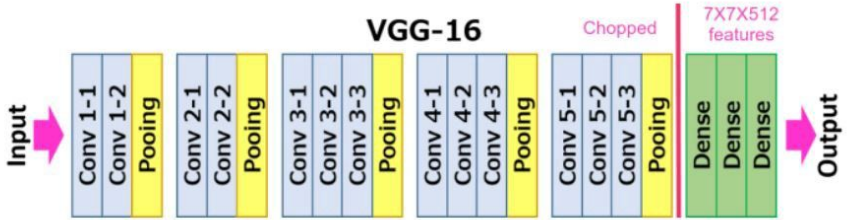


Figure 4: Architecture of Generator and Discriminator Network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.

Loss

SRGAN uses a perceptual loss measuring the MSE of features extracted by a VGG-19 network. For a specific layer within VGG-19, we want their features to be matched (Minimum MSE for features).

$$l^{SR} = \underbrace{\underbrace{l_{\mathrm{X}}^{SR}}_{\text{content loss}} + \underbrace{10^{-3} l_{Gen}^{SR}}_{\text{adversarial loss}}}_{\text{perceptual loss (for VGG based content losses)}}$$



$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

$$l_{VGG/i.j}^{SR} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2$$

$\phi_{i,j}$ **The feature map for the j-th convolution (after activation) before the i-th maxpooling layer.**

Bicubic upsampling

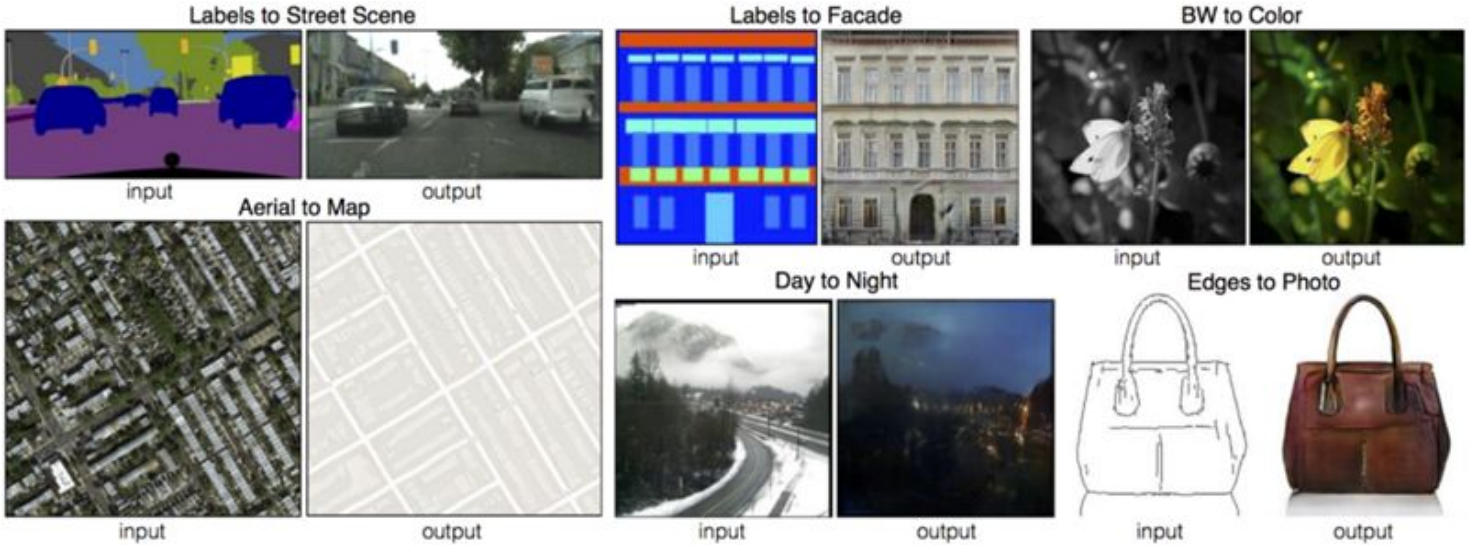
SRGAN

Original

Pix2Pix

The goal of this model is to convert from one image to another image, in other words the goal is to learn the mapping from an input image to an output image.

Pix2Pix GAN is an implementation of the cGAN where the generation of an image is conditional on a given image.



Architecture

- U- Net Generator
- The discriminator network uses the PatchGAN network. instead of predicting the whole image as fake or real at the discriminator, the model takes a N*N patch image and predicts every pixel in that patch if its real or fake

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$

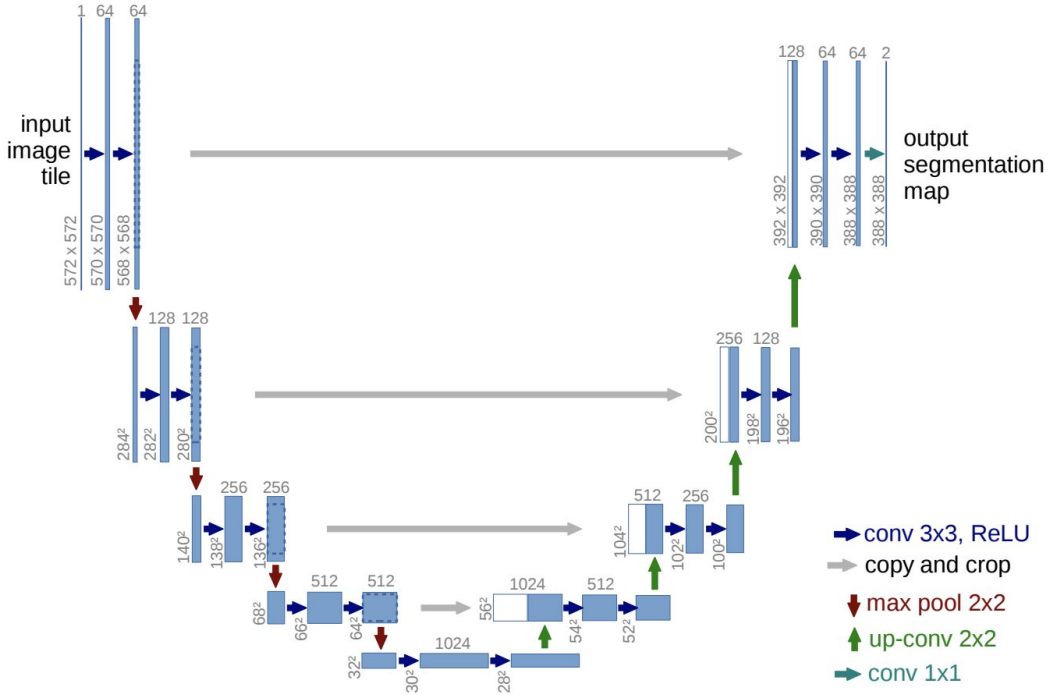
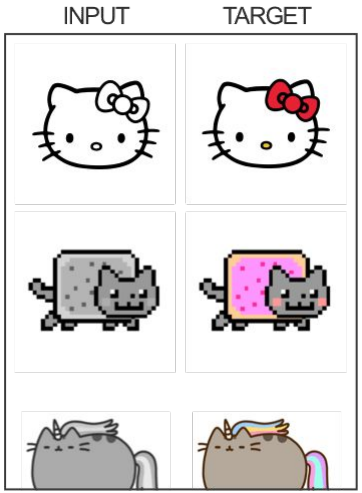
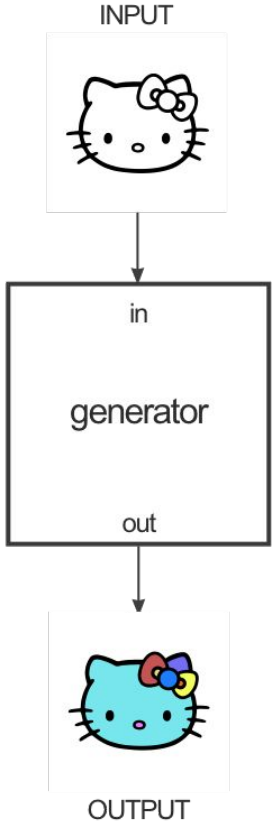
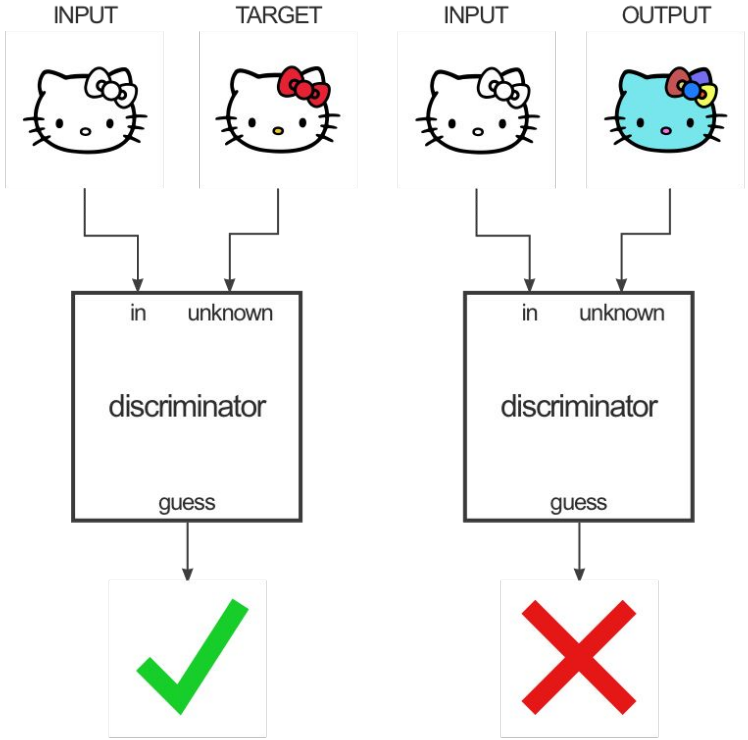


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Training







- https://affinelayer.com/pixsrv/

References

- Generative Adversarial Nets - https://arxiv.org/pdf/1406.2661.pdf
- Conditional Generative Adversarial Nets - https://arxiv.org/pdf/1411.1784.pdf
- Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks - https://arxiv.org/pdf/1511.06434.pdf
- A Style-Based Generator Architecture for Generative Adversarial Networks - https://arxiv.org/pdf/1812.04948.pdf
- Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network - https://arxiv.org/pdf/1609.04802.pdf
- Image-to-Image Translation with Conditional Adversarial Networks - https://arxiv.org/pdf/1611.07004.pdf