Analysis of Complex Discriminator Architectures in Generative Adversarial Networks

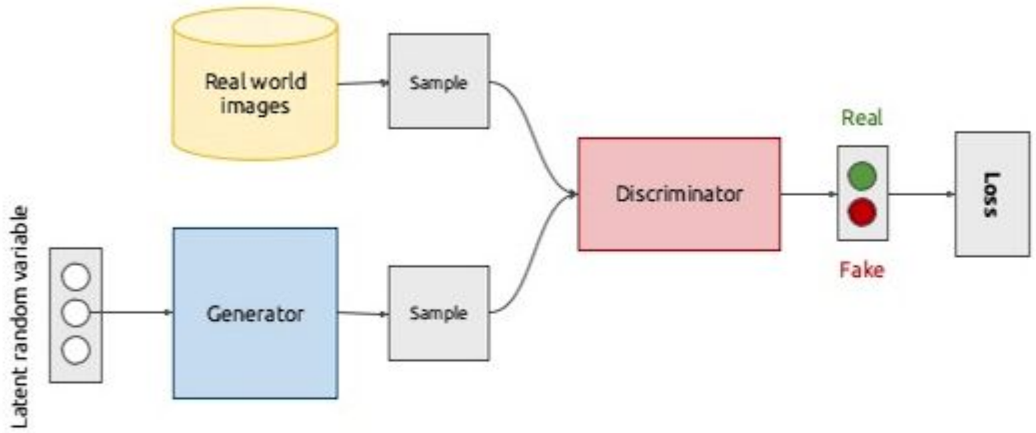
BITS F464 - Machine Learning (Term Paper)

1 NOV 2020

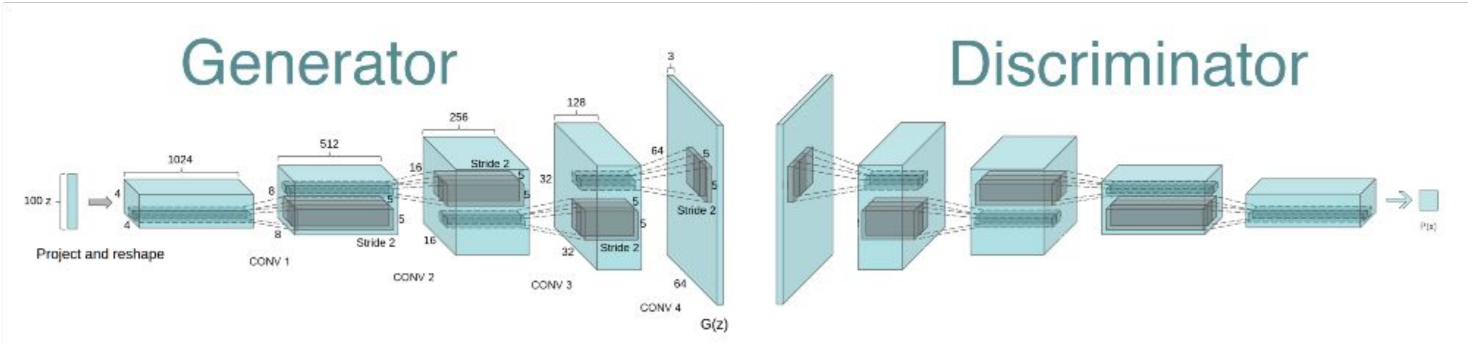
Prateek Grover

Introduction

Two-Player Game of Generative Adversarial Networks



Deep Convolutional Generative Adversarial Networks



Discussion

Problem Statement

Problem Statement

Improper generation of images that consists of all the actual parts that should be present in an image but in an improbable order or arrangement, yet the generator manages to convince the discriminator, that the image is real.

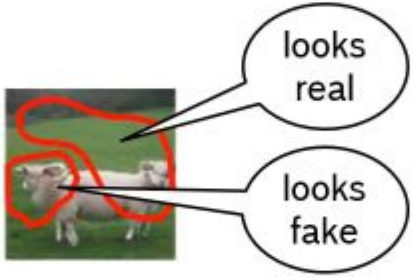
Problem Statement

Improper generation of images that consists of all the actual parts that should be present in an image but in an improbable order or arrangement, yet the generator manages to convince the discriminator, that the image is real.

Problem Statement

Traditional Discriminators

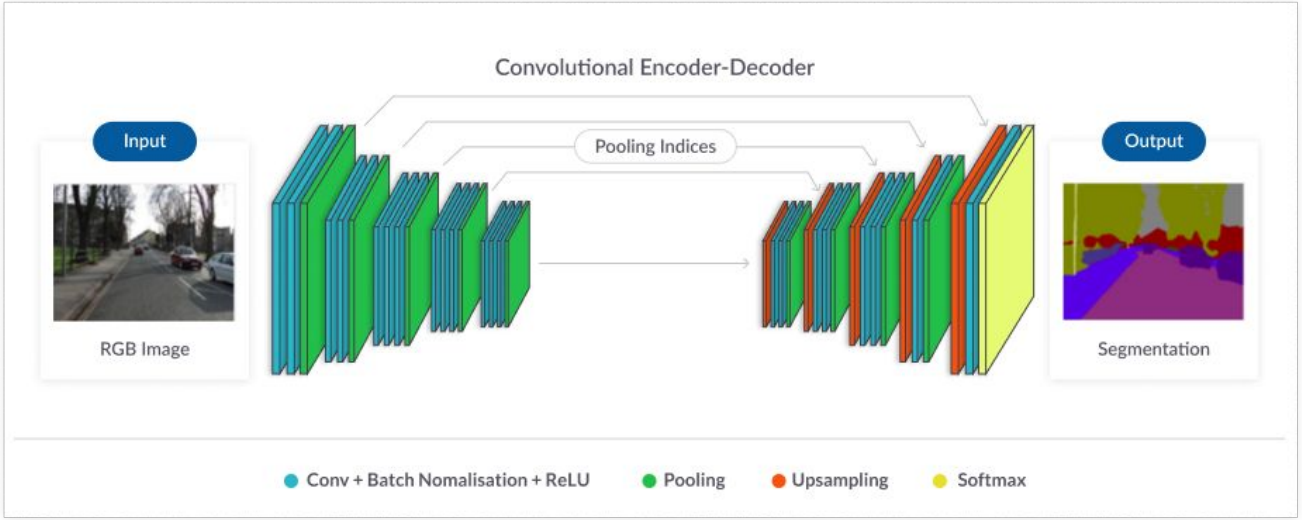
Proposed Discriminator



Per-Pixel Classification in Discriminators

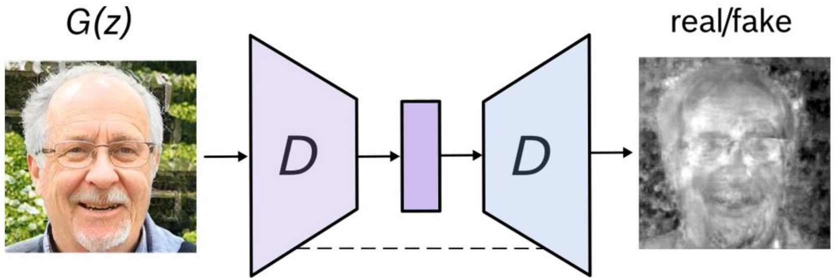
Classifies each pixel in the generated image a value between 0 and 1 indicating its degree of ‘realness’.

Image Segmentation using Deep Learning



a typical Encoder-Decoder CNN Architecture used in Image Segmentation

Proposed Solution

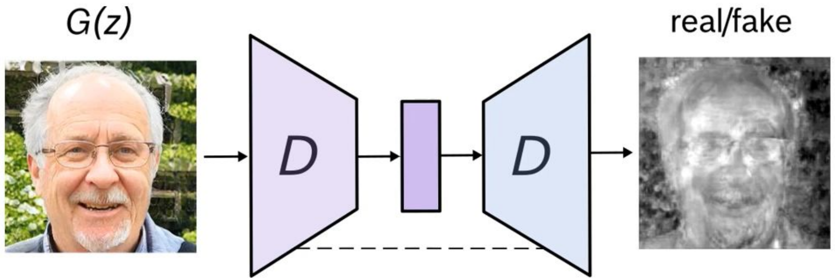


an encoder-decoder based Discriminator

Proposed Solution

Possible Problems that we could face while using a Segmentation Network -

1. Neglecting Global Realism
2. Encouraging Texture Bias

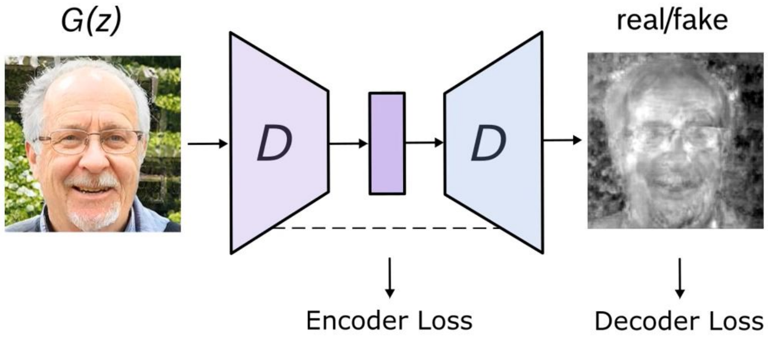


an encoder-decoder based Discriminator

Proposed Solution

Possible Problems that we could
face while using a Segmentation Network -

1. Neglecting Global Realism
2. Encouraging Texture Bias



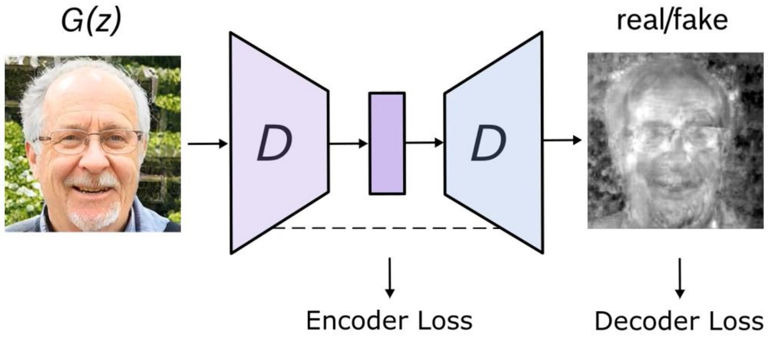
Solution:

1. Global Loss from Encoder & Local Loss from Decoder Network

Proposed Solution

Possible Problems that we could
face while using a Segmentation Network -

1. Neglecting Global Realism
2. Encouraging Texture Bias



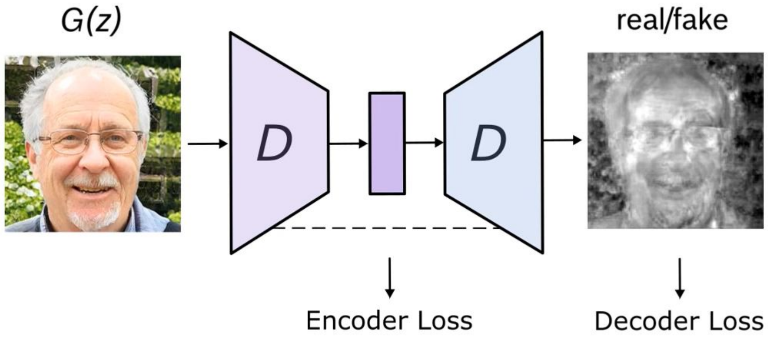
Solution:

1. Global Loss from Encoder & Local Loss from Decoder Network
2. Skip Connections that can combine Local and Global Features

Proposed Solution

Possible Problems that we could face while using a Segmentation Network -

1. Neglecting Global Realism
2. Encouraging Texture Bias



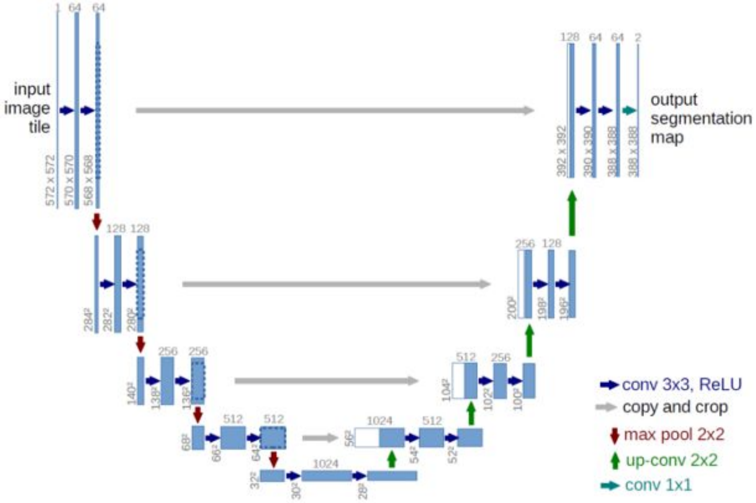
Solution:

1. Global Loss from Encoder & Local Loss from Decoder Network
2. Skip Connections that can combine Local and Global Features

U - Net Architecture used widely in Biomedical Image Segmentation

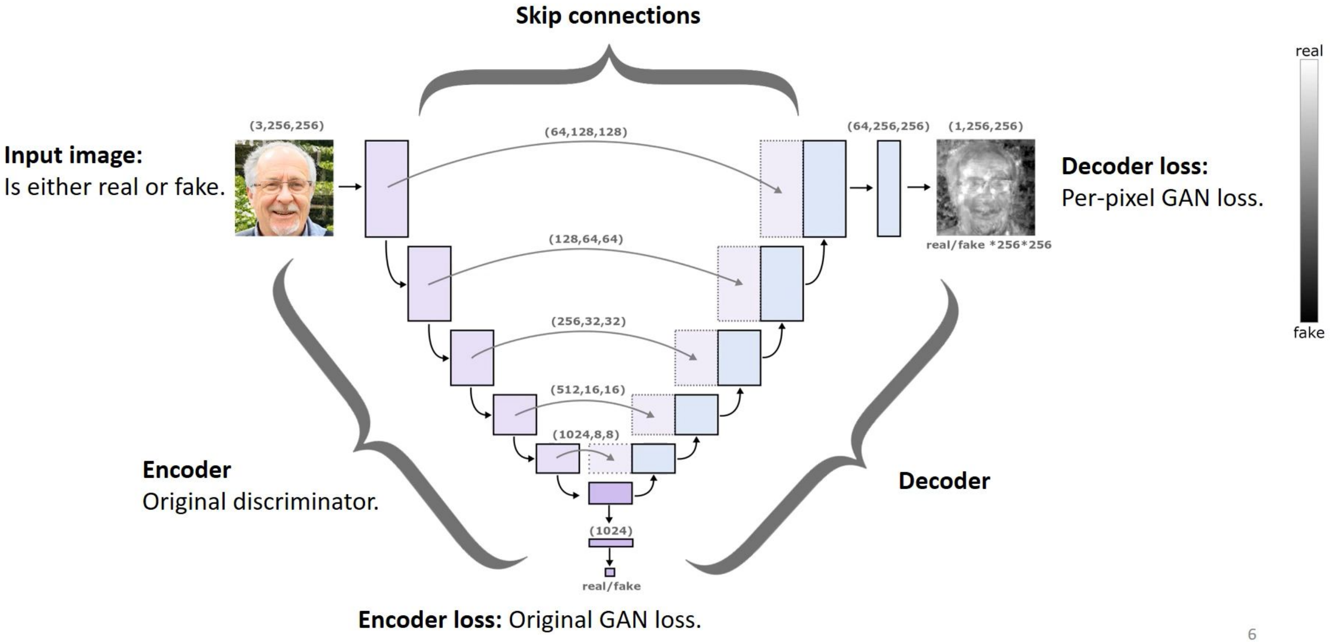
What is U-Net?

The main idea is to supplement a usual contracting network by successive layers, where pooling operations are replaced by upsampling operators. Hence these layers increase the resolution of the output

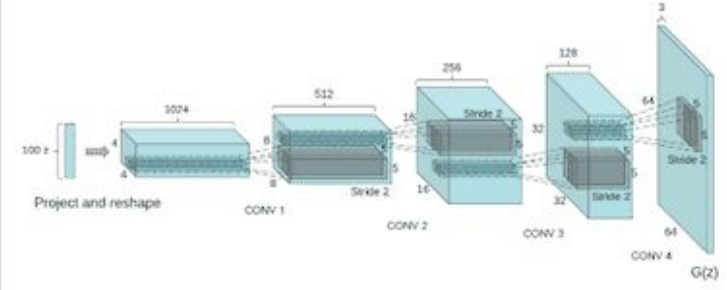


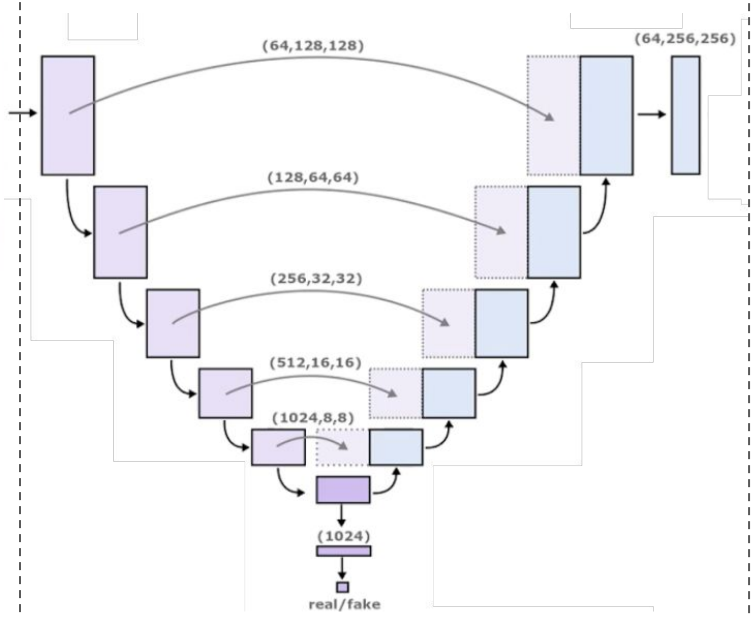
architecture of a typical U-Net

U-Net Based Discriminator



CNN Based Generator + U-Net Based Discriminator





Primary Reference utilizes a BigGAN Generator

Implementation

Implementation

Environment

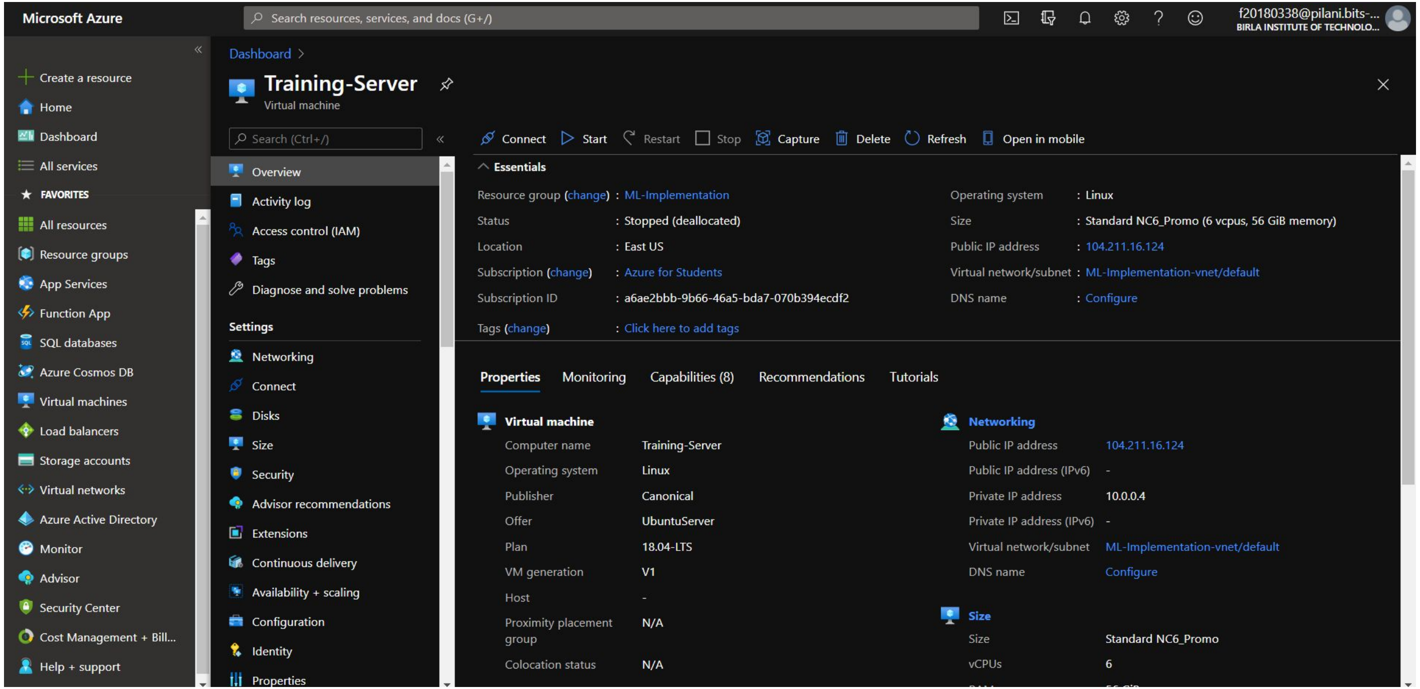
- Used Standard_NC6 of NC-Series Virtual Machines powered by NVIDIA Tesla K80 Card and Intel Xeon E5-2690 v3 (Haskell) Processor using Microsoft Azure cloud services.

Dependencies

- PyTorch 1.4 (stable), machine learning library
- CUDA 10.0, parallel computing platform
- OmegaConf, hierarchical configuration system
- apex.amp, for mixed precision training
- torch-mimicry, a specialized library for GAN Research

and other standard libraries

Setting up a Virtual Machine



Obtaining Datasets

- **Flickr Faces HQ Dataset (FFHQ) -** (https://github.com/NVlabs/ffhq-dataset)

- 70k Face Images: Original Resolution: 1024x1024, Used Resolution: 32x32

- **COCO Animals -** (http://cs231n.stanford.edu/coco-animals.zip)

- 38k Images of 10 Animal Classes: Original Resolution: 224x224, Used Resolution: 64x64

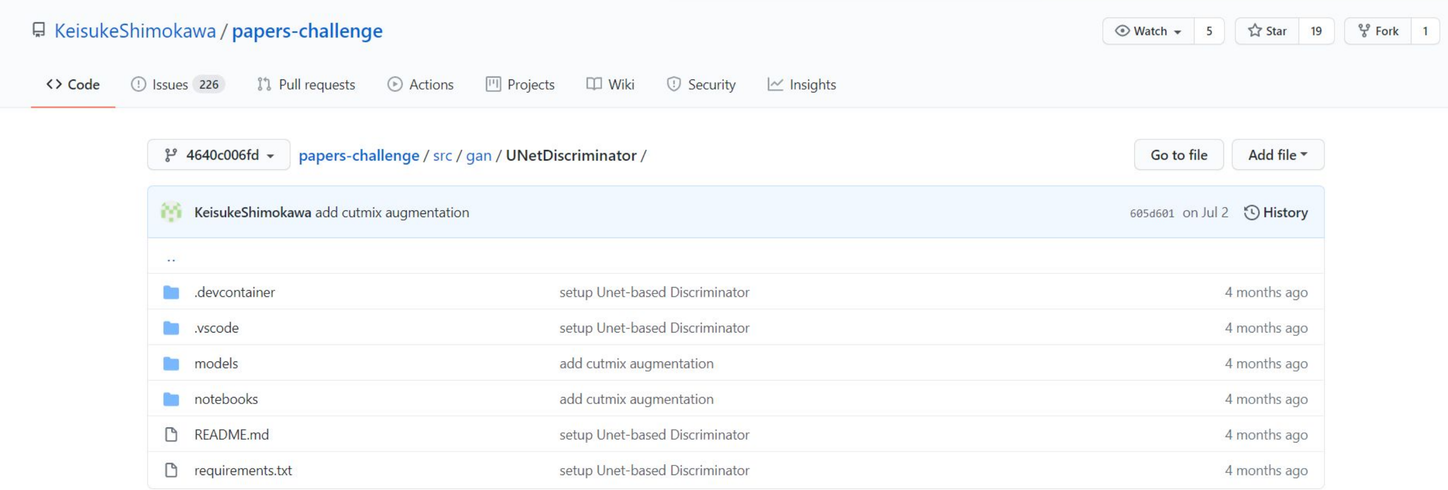
- **CelebA -** (http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html)

- 200k Face Images: Original Resolution: 218x178 Used Resolution: 32x32 (aligned and cropped)

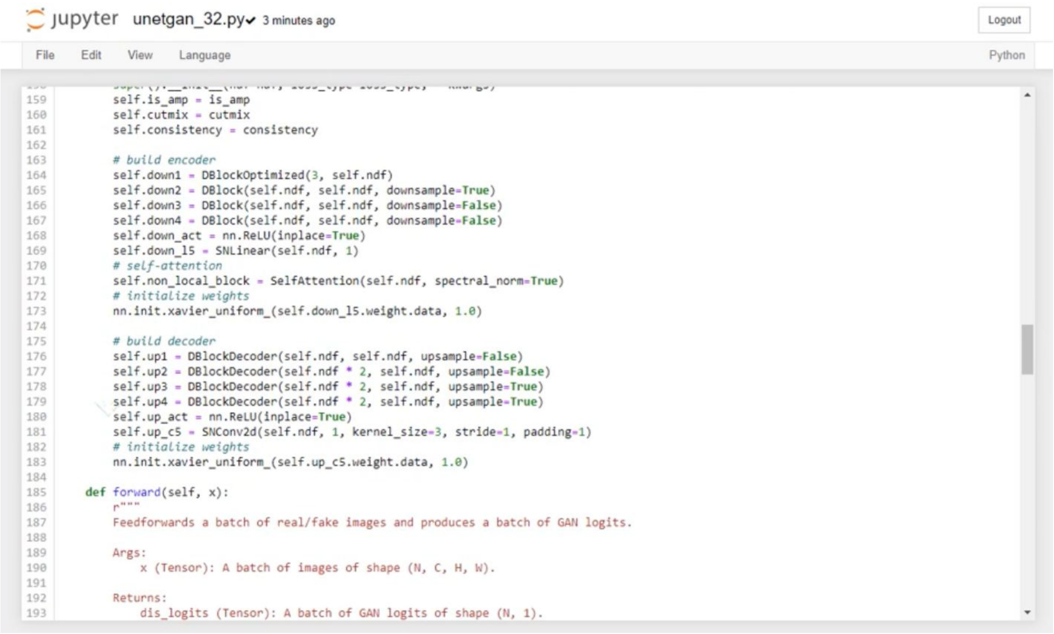
Obtaining Source Code

Unofficial Implementation on MNIST by Keisuke Shimokawa (https://github.com/KeisukeShimokawa/papers-challenge)

Official Implementation by Bosch Research was made public on October 15th. 2020.



Jupyter Notebook on a Hosted Server




```python
        self.is_amp = is_amp
        self.cutmix = cutmix
        self.consistency = consistency

        # build encoder
        self.down1 = DBlockOptimized(3, self.ndf)
        self.down2 = DBlock(self.ndf, self.ndf, downsample=True)
        self.down3 = DBlock(self.ndf, self.ndf, downsample=False)
        self.down4 = DBlock(self.ndf, self.ndf, downsample=False)
        self.down_act = nn.ReLU(inplace=True)
        self.down_l5 = SNLinear(self.ndf, 1)
        # self-attention
        self.non_local_block = SelfAttention(self.ndf, spectral_norm=True)
        # initialize weights
        nn.init.xavier_uniform_(self.down_l5.weight.data, 1.0)

        # build decoder
        self.up1 = DBlockDecoder(self.ndf, self.ndf, upsample=False)
        self.up2 = DBlockDecoder(self.ndf * 2, self.ndf, upsample=False)
        self.up3 = DBlockDecoder(self.ndf * 2, self.ndf, upsample=True)
        self.up4 = DBlockDecoder(self.ndf * 2, self.ndf, upsample=True)
        self.up_act = nn.ReLU(inplace=True)
        self.up_c5 = SNConv2d(self.ndf, 1, kernel_size=3, stride=1, padding=1)
        # initialize weights
        nn.init.xavier_uniform_(self.up_c5.weight.data, 1.0)

    def forward(self, x):
        r"""
        Feedforwards a batch of real/fake images and produces a batch of GAN logits.

        Args:
            x (Tensor): A batch of images of shape (N, C, H, W).

        Returns:
            dis_logits (Tensor): A batch of GAN logits of shape (N, 1).
```

Evaluation Metrics

Inception Distance is defined as the exponential of KL Divergence between the label distribution, $P(y|x)$ and the marginal distribution, $P(y)$.

$$IS = exp(\sum_{x \in \mathcal{X}} P(y|x) \log \left(\frac{P(y|x)}{P(y)} \right))$$

Frechet Inception Distance is defined as the Frechet Distance between the multidimensional Gaussian distributions, one depicting all the real images used to train the network $\mathcal{N}(\mu_w, \Sigma_w)$ and the distribution of images generated by the GAN, $\mathcal{N}(\mu, \Sigma)$.

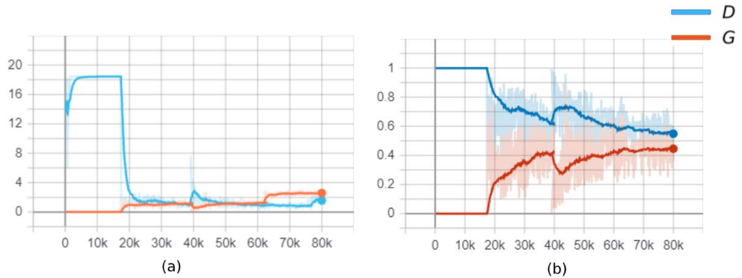
$$\mathrm{FID} = |\mu - \mu_w|^2 + \mathrm{tr}(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{1/2}).$$

Training and Results

Per-Pixel Classification of Generated Image

Training and Results

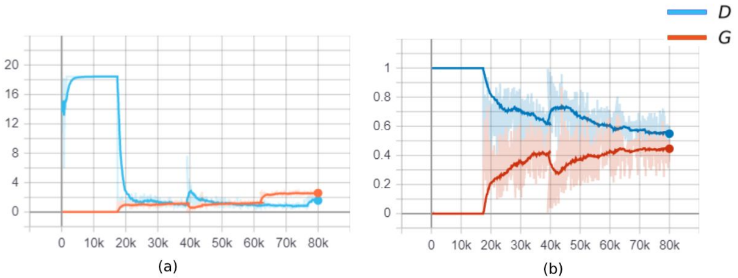
Per-Pixel Classification of Generated Image



(a) Loss Curve obtained and (b) Mean Probability of Correct Classification during training on CelebA Dataset visualized using Tensorboard

Training and Results

Per-Pixel Classification of Generated Image



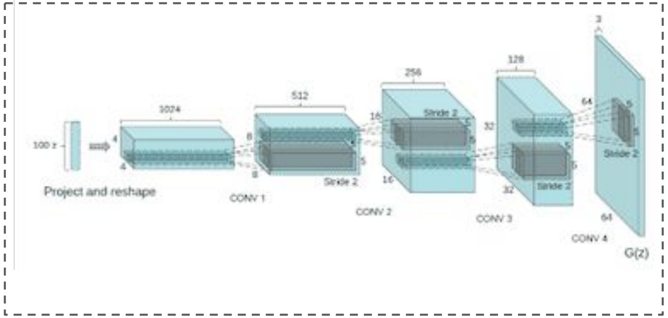
(a) Loss Curve obtained and (b) Mean Probability of Correct Classification during training on CelebA Dataset visualized using Tensorboard

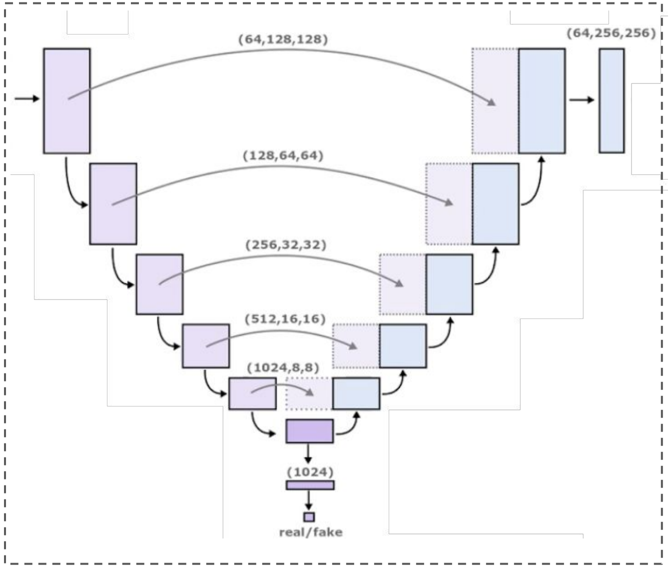
	COCO - Animals		Flickr Faces HQ		CelebA	
	FID	IS	FID	IS	FID	IS
BigGAN [18]	16.55	11.78	12.42	4.02	4.54	3.23
U-Net GAN with CutMix Augmentation	29.35 (13.87)	7.86 (12.31)	22.53 (7.63)	2.32 (4.47)	11.52 (2.95)	1.86 (3.43)

Advances and Applications

Recall:

BigGAN Generator + U-Net Based Discriminator





Advancements in GANs

LOGAN: Latent Optimisation for Generative Adversarial Networks

Uses latent optimization to improve adversarial dynamics between Discriminator and Generator, produces vastly better results in terms of variety and fidelity.

Comparison between the images generated using BigGAN and using LOGAN

Yan Wu and Jeff Donahue and David Balduzzi and Karen Simonyan and Timothy Lillicrap, *LOGAN: Latent Optimisation for Generative Adversarial Networks*

Advancements in GANs

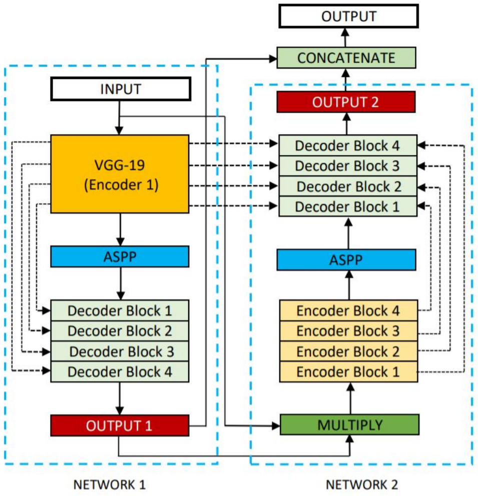
StyleGAN: A Style-Based Generator Architecture for Generative Adversarial Networks

Uses Structured Layer inputs to Generating thereby controlling the images generated, able to manipulate the age, sex, and angle of picture captured.

Karras, T., Laine, S., & Aila, T. (2019). *A Style-Based Generator Architecture for Generative Adversarial Networks. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

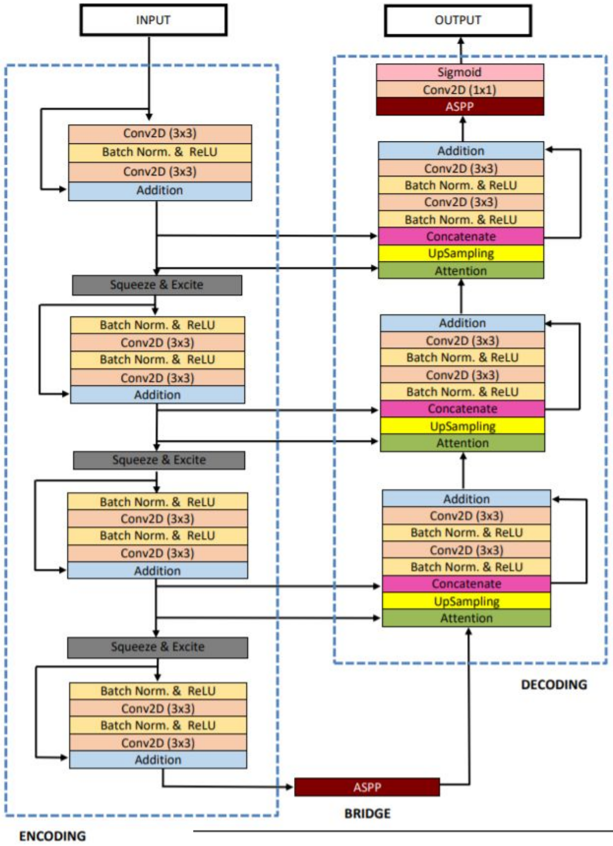
Advancements to U-Net in Biomedical Image Processing

Double U-Net: A Deep Convolutional Neural Network for Medical Image Segmentation



ASPP: Atrous Spatial Pyramid Pooling

Jha, D., Riegler, M. A., Johansen, D., Halvorsen, P., & Johansen, H. D. (2020). *DoubleU-Net: A Deep Convolutional Neural Network for Medical Image Segmentation. 2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS).*



ResUNet++: An Advanced Architecture for Medical Image Segmentation

Jha, D., Smedsrud, P. H., Riegler, M. A., Johansen, D., Lange, T. D., Halvorsen, P., & D. Johansen, H. (2019). *ResUNet++: An Advanced Architecture for Medical Image Segmentation. 2019 IEEE International Symposium on Multimedia (ISM).*

ASPP: Atrous Spatial Pyramid Pooling

Thanks!