

Analysis of Complex Discriminator Architectures in Generative Adversarial Networks

BITS F464 - Machine Learning (Term Paper)

1 NOV 2020

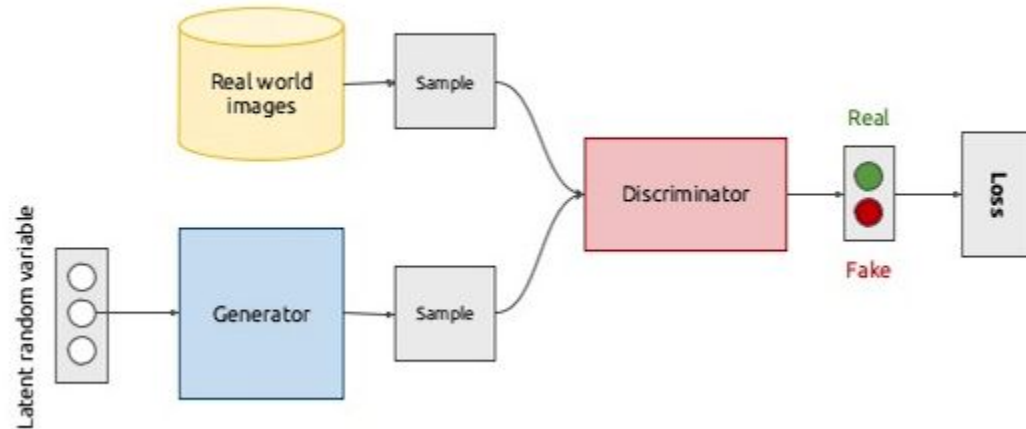
Prateek Grover

Introduction

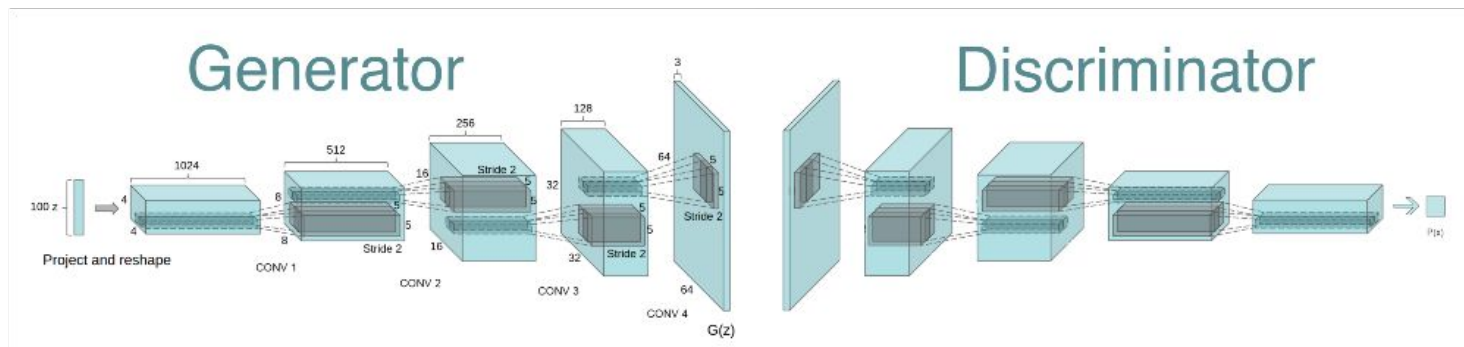
Analysis of Complex Discriminator Architectures in Generative Adversarial Networks

Prateek Grover

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

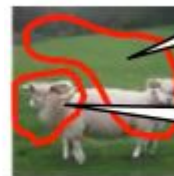


Traditional
Discriminators



real or
fake?

Proposed
Discriminator



looks
real

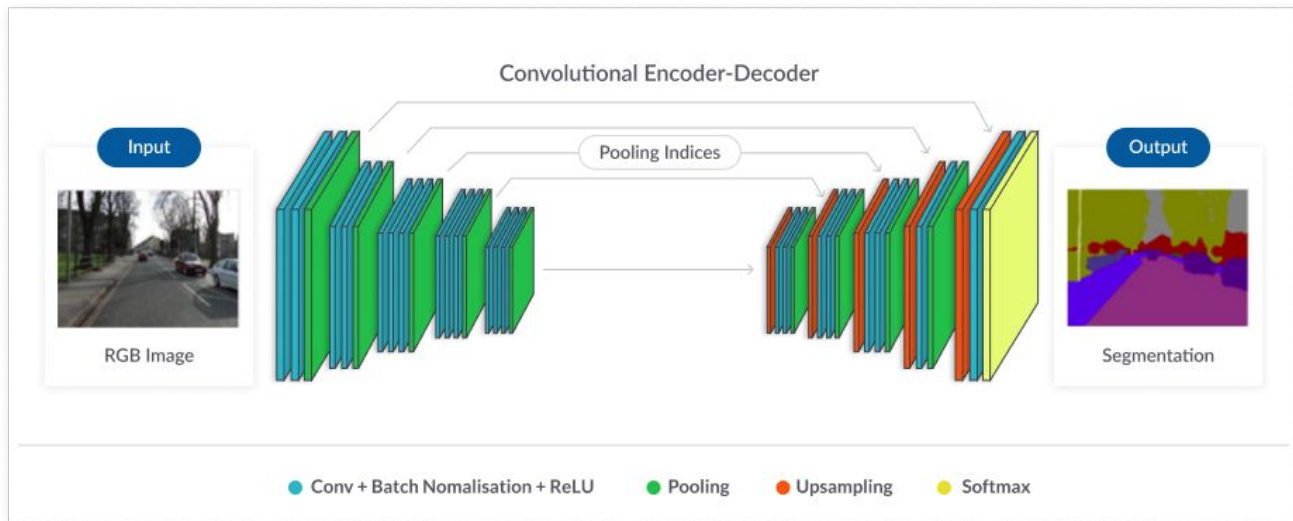
looks
fake

Per-Pixel Classification in Discriminators

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

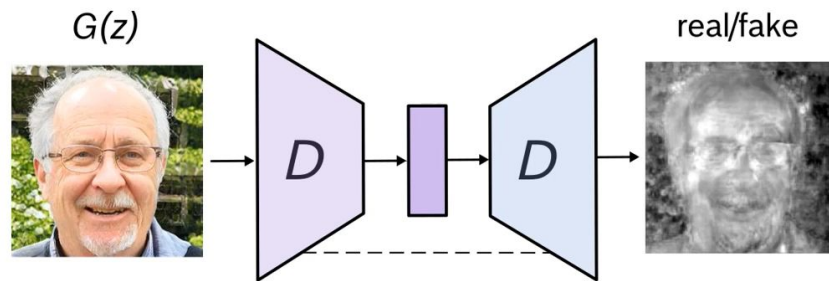


Image Segmentation using Deep Learning



a typical Encoder-Decoder CNN Architecture used in Image Segmentation

Proposed Solution

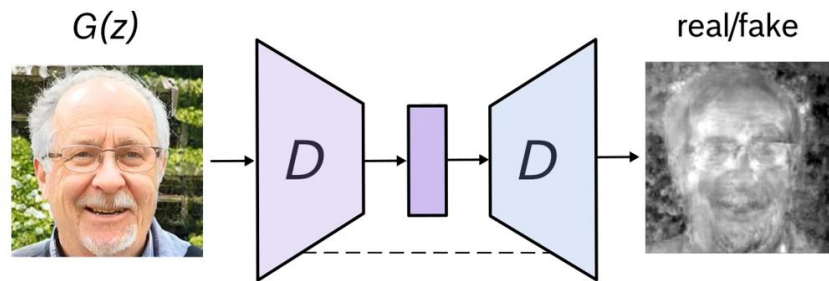


an encoder-decoder based Discriminator

Proposed Solution

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

1. Neglecting Global Realism
2. Encouraging Texture Bias

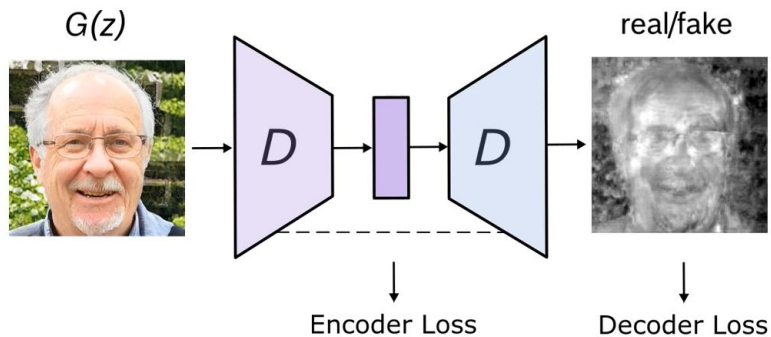


an encoder-decoder based Discriminator

Proposed Solution

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

1. Neglecting Global Realism
2. Encouraging Texture Bias



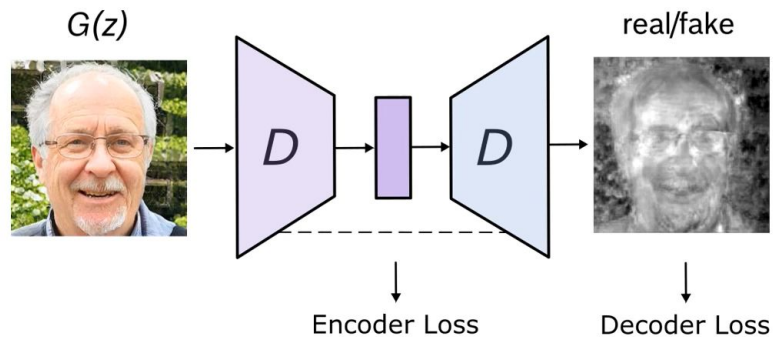
Solution:

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



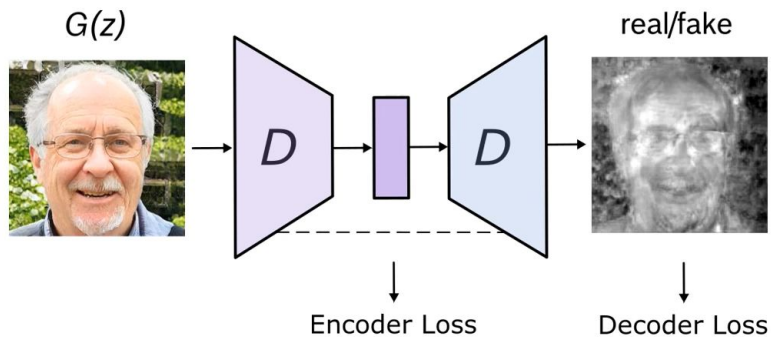
Solution:

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



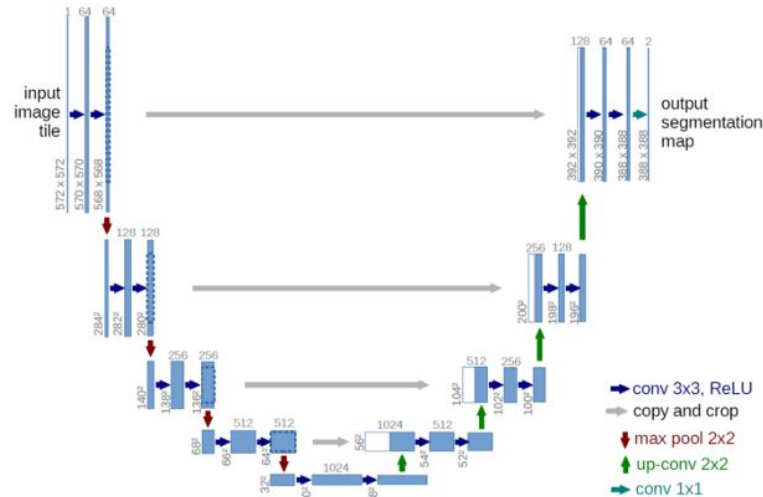
Solution:

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

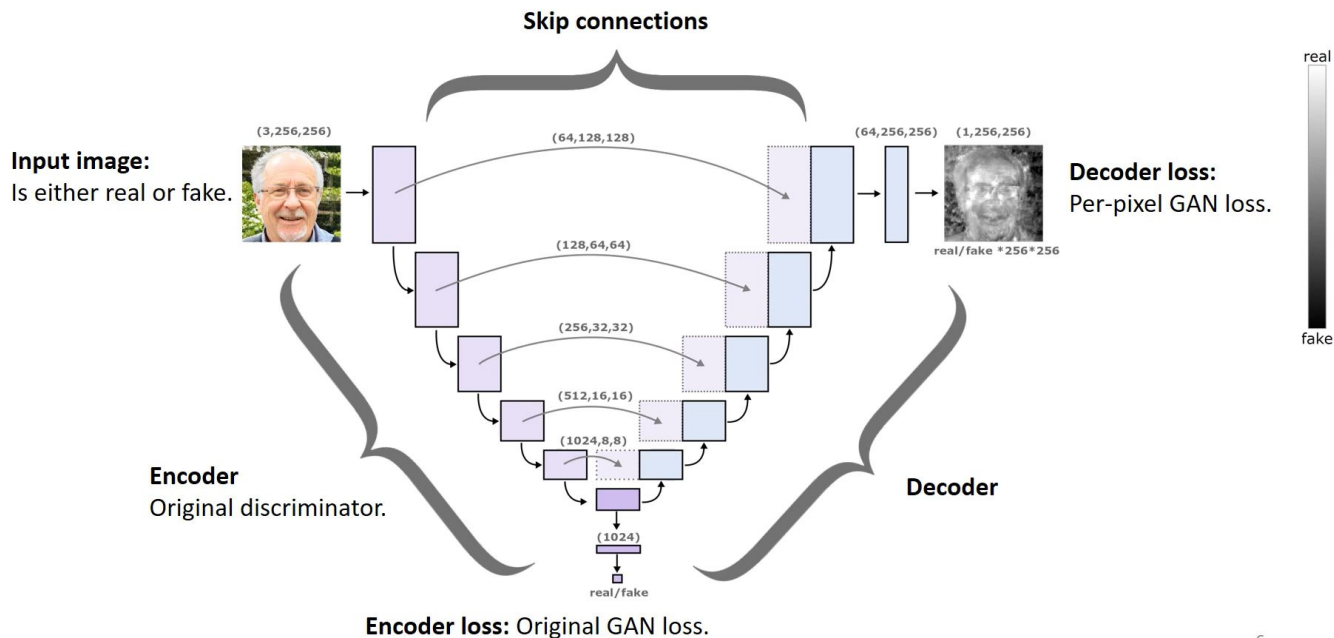
What is U-Net?

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

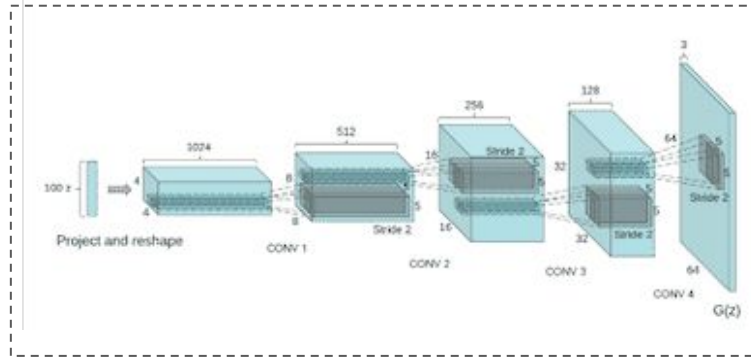


architecture of a typical U-Net

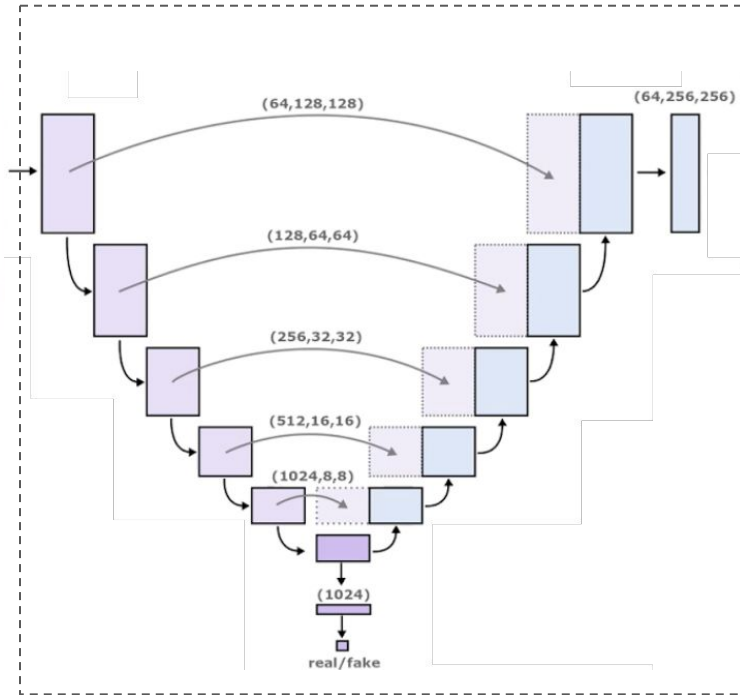
U-Net Based Discriminator



CNN Based Generator + U-Net Based Discriminator



Primary Reference utilizes a BigGAN Generator



Implementation

Analysis of Complex Discriminator Architectures in Generative Adversarial Networks

Prateek Grover

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

The screenshot displays the Microsoft Azure portal interface. On the left is a navigation sidebar with options like 'Create a resource', 'Home', 'Dashboard', 'All services', and 'FAVORITES'. The main area shows the 'Training-Server' virtual machine details. At the top, there are action buttons: Connect, Start, Restart, Stop, Capture, Delete, Refresh, and Open in mobile. Below this, the 'Essentials' section provides a summary of the VM's configuration. The 'Properties' tab is selected, showing detailed settings for the virtual machine and its networking.

Essentials	
Resource group (change)	ML-Implementation
Status	Stopped (deallocated)
Location	East US
Subscription (change)	Azure for Students
Subscription ID	a5ae2bbb-9b66-46a5-bda7-070b394ecd2
Tags (change)	Click here to add tags
Operating system	Linux
Size	Standard NC6_Promo (6 vcpus, 56 GiB memory)
Public IP address	104.211.16.124
Virtual network/subnet	ML-Implementation-vnet/default
DNS name	Configure

Virtual machine	
Computer name	Training-Server
Operating system	Linux
Publisher	Canonical
Offer	UbuntuServer
Plan	18.04-LTS
VM generation	V1
Host	-
Proximity placement group	N/A
Colocation status	N/A

Networking	
Public IP address	104.211.16.124
Public IP address (IPv6)	-
Private IP address	10.0.0.4
Private IP address (IPv6)	-
Virtual network/subnet	ML-Implementation-vnet/default
DNS name	Configure

Size	
Size	Standard NC6_Promo
vCPUs	6

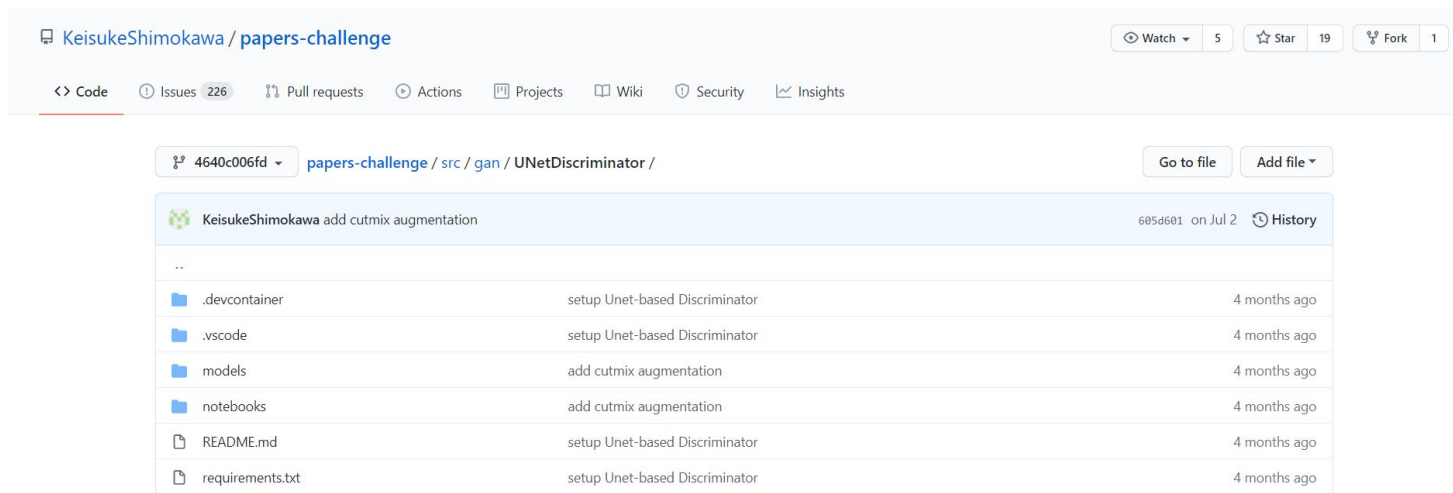
Obtaining Datasets

- **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.



KeisukeShimokawa / papers-challenge

Watch 5 Star 19 Fork 1

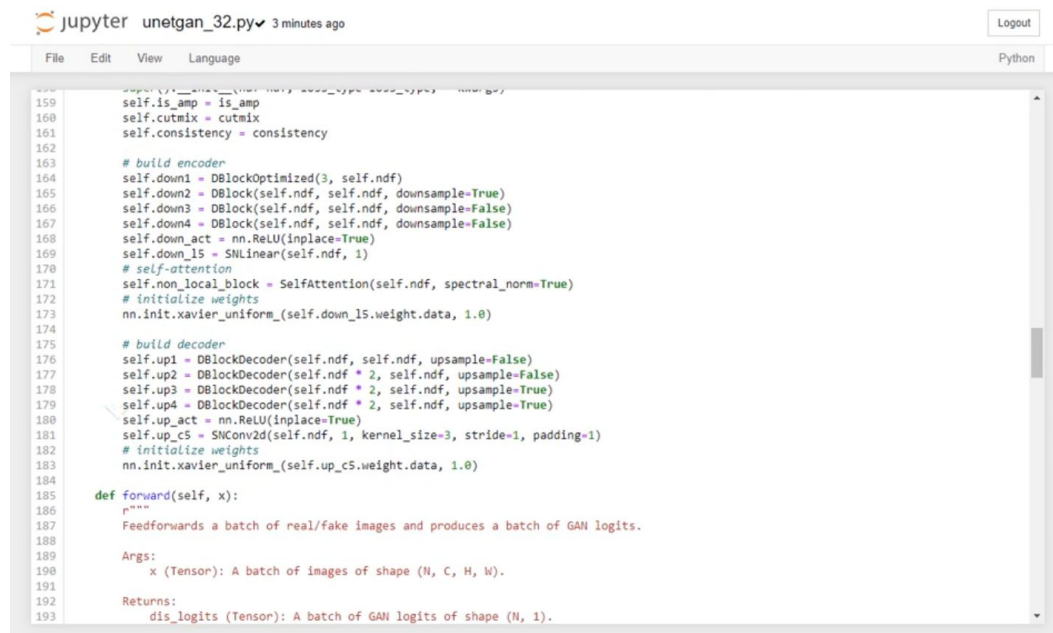
<> Code Issues 226 Pull requests Actions Projects Wiki Security Insights

4640c006fd papers-challenge / src / gan / UNetDiscriminator /

Go to file Add file

KeisukeShimokawa add cutmix augmentation	605d601 on Jul 2	History
..		
.devcontainer	setup Unet-based Discriminator	4 months ago
.vscode	setup Unet-based Discriminator	4 months ago
models	add cutmix augmentation	4 months ago
notebooks	add cutmix augmentation	4 months ago
README.md	setup Unet-based Discriminator	4 months ago
requirements.txt	setup Unet-based Discriminator	4 months ago

Jupyter Notebook on a Hosted Server



```
159 self.is_amp = is_amp
160 self.cutmix = cutmix
161 self.consistency = consistency
162
163 # build encoder
164 self.down1 = DBlockOptimized(3, self.ndf)
165 self.down2 = DBlock(self.ndf, self.ndf, downsample=True)
166 self.down3 = DBlock(self.ndf, self.ndf, downsample=False)
167 self.down4 = DBlock(self.ndf, self.ndf, downsample=False)
168 self.down_act = nn.ReLU(inplace=True)
169 self.down_15 = SNLinear(self.ndf, 1)
170 # self-attention
171 self.non_local_block = SelfAttention(self.ndf, spectral_norm=True)
172 # initialize weights
173 nn.init.xavier_uniform_(self.down_15.weight.data, 1.0)
174
175 # build decoder
176 self.up1 = DBlockDecoder(self.ndf, self.ndf, upsample=False)
177 self.up2 = DBlockDecoder(self.ndf * 2, self.ndf, upsample=False)
178 self.up3 = DBlockDecoder(self.ndf * 2, self.ndf, upsample=True)
179 self.up4 = DBlockDecoder(self.ndf * 2, self.ndf, upsample=True)
180 self.up_act = nn.ReLU(inplace=True)
181 self.up_c5 = SNConv2d(self.ndf, 1, kernel_size=3, stride=1, padding=1)
182 # initialize weights
183 nn.init.xavier_uniform_(self.up_c5.weight.data, 1.0)
184
185 def forward(self, x):
186     """
187     Feedforwards a batch of real/fake images and produces a batch of GAN logits.
188
189     Args:
190         x (Tensor): A batch of images of shape (N, C, H, W).
191
192     Returns:
193         dis_logits (Tensor): A batch of GAN logits of shape (N, 1).
```

Evaluation Metrics

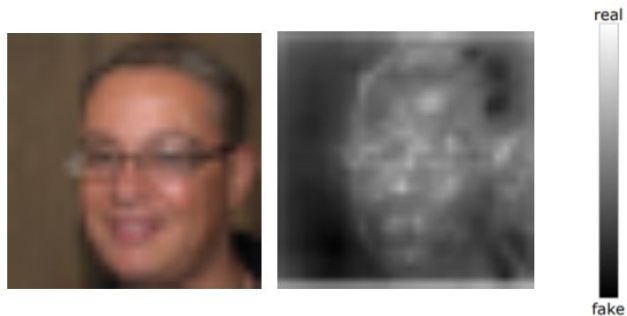
Inception Distance is defined as the exponential of KL Divergence between the label distribution, $P(y|x)$ and the marginal distribution, $P(y)$.

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

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, $\mathcal{N}(\mu, \Sigma)$.

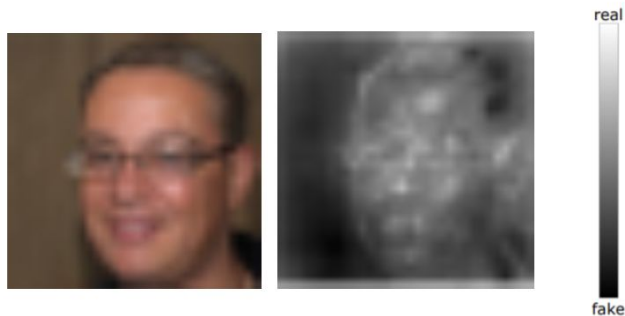
$$FID = |\mu - \mu_w|^2 + \text{tr}(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{1/2}).$$

Training and Results

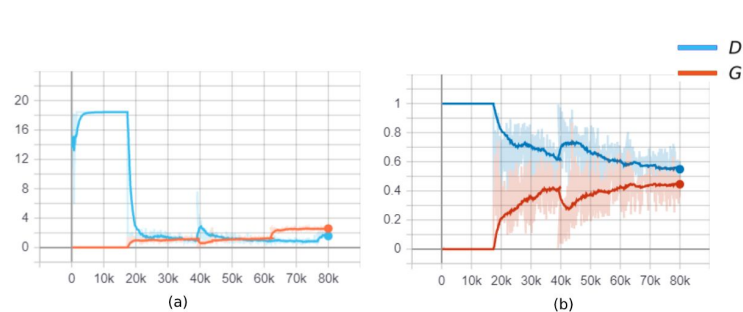


Per-Pixel Classification of Generated Image

Training and Results



Per-Pixel Classification of Generated Image

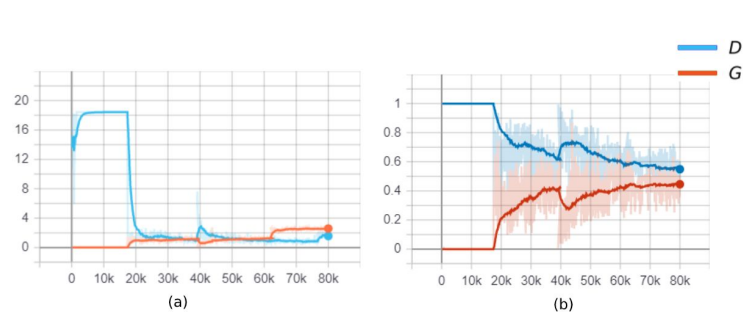


(a) Loss Curve obtained and (b) Mean Probability of Correct Classification during training on CelebA Dataset visualized using Tensorboard

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

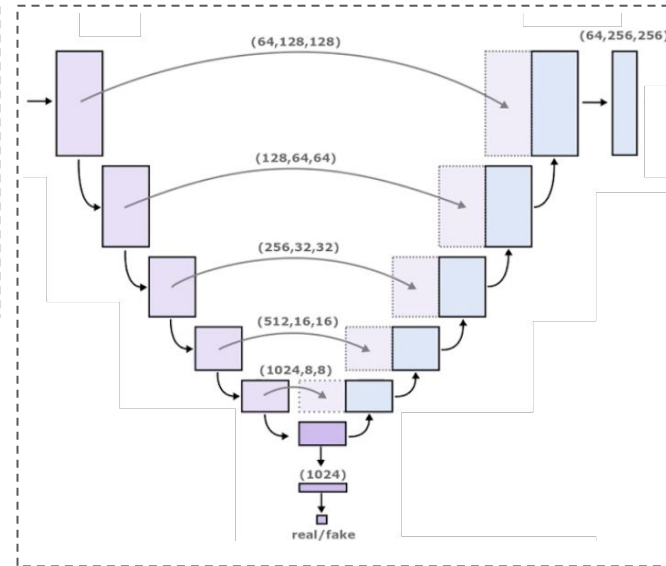
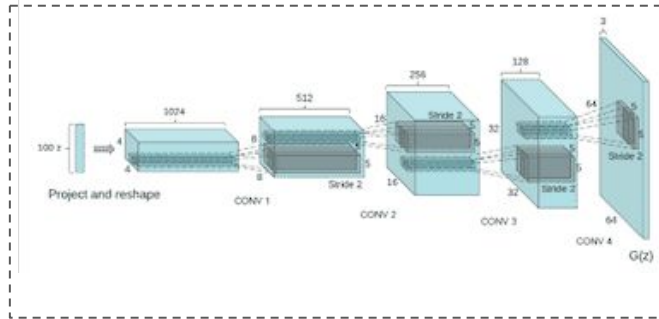
Analysis of Complex Discriminator Architectures in Generative Adversarial Networks

Prateek Grover

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.



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

Comparison between the images generated
using BigGAN and using LOGAN

Advancements in GANs

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

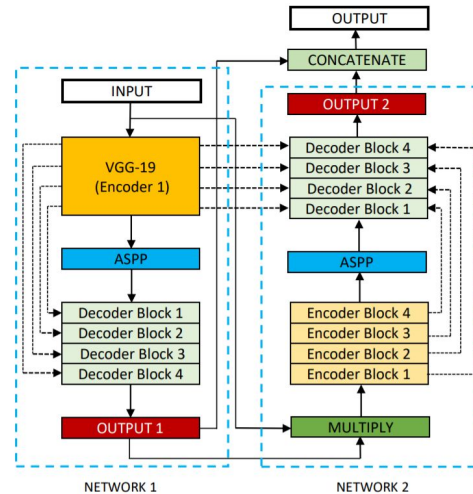
Uses Structured Layer inputs to Generating thereby controlling the images generated, able to manipulate the age, sex, and angle of picture captured.



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).

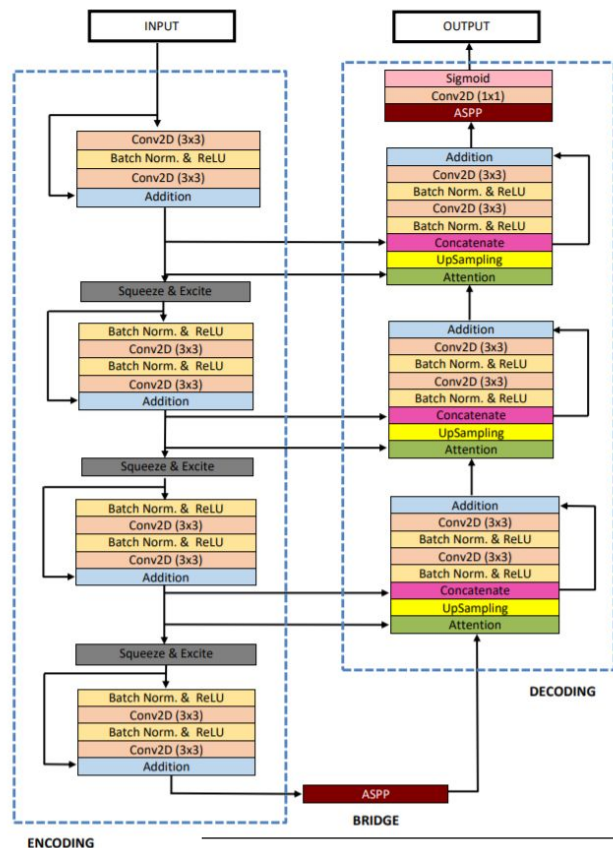
Advancements to U-Net in Biomedical Image Processing

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!