Autoencoders and GAN

Autoencoders

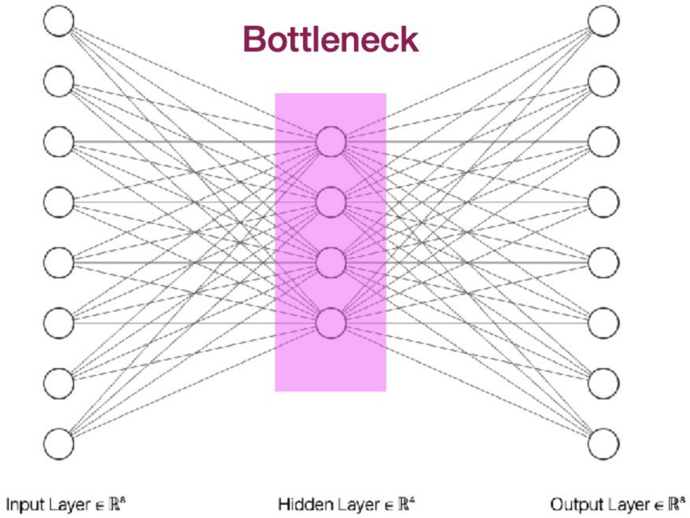
- An Unsupervised Learning technique that is used for representational learning
- Recall that our CNN filters act as feature detectors:
  - high level such as patterns
  - low level such as edges or blobs
- What if we could exploit what a CNN learns about a dataset so that it acts as a method of compression?

What do Autoencoders do?

- They learn to compress data based on their correlations between input features
- Some applications include:
  - Denoising (images, even audio)
  - Image Inpainting
  - Information Retrieval
  - Anomaly Detection

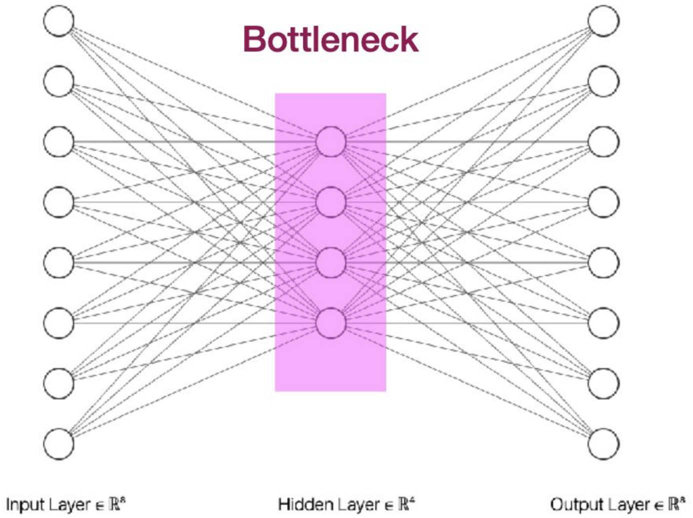
Autoencoders Architecture

- Autoencoders are neural networks that use a bottleneck architecture which forces a compressed knowledge representation of the input data.
- Autoencoders work very well with data that has correlated input features (i.e. not independent).


Bottleneck
Input Layer $\in \mathbb{R}^8$
Hidden Layer $\in \mathbb{R}^4$
Output Layer $\in \mathbb{R}^8$

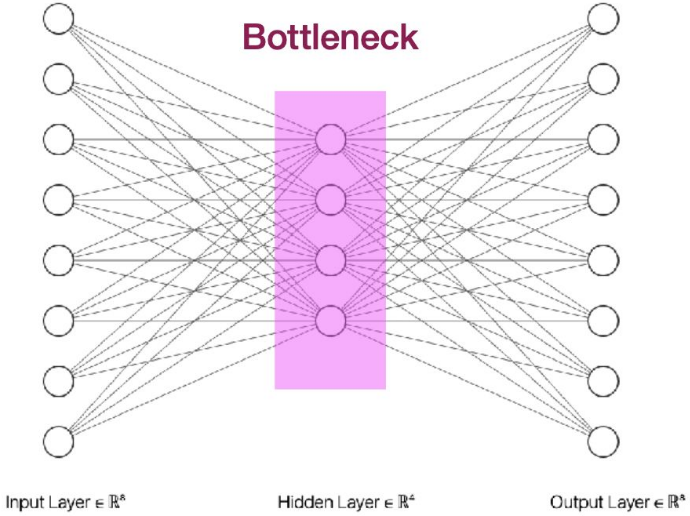
Bottleneck

- The bottleneck constrains the amount of information that is able to traverse the full network.
- This enables the hidden bottleneck layer(s) to learn a compressed representation of the input data


Bottleneck
Input Layer $\in \mathbb{R}^8$
Hidden Layer $\in \mathbb{R}^4$
Output Layer $\in \mathbb{R}^8$

Autoencoders Architecture

- Notice our input and output match in dimensions, that is because we’re reconstructing the input
- Our loss function here penalises reconstruction erorr
- This allows the model to learn the most important features needed to reconstruct the data/image

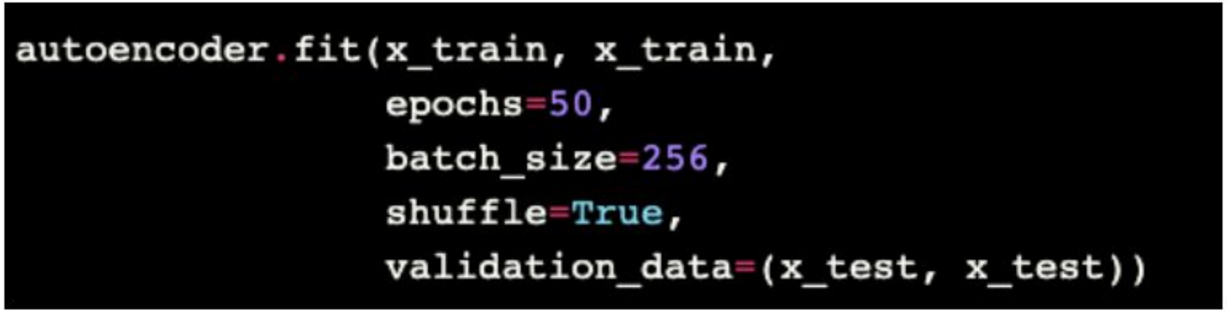

Bottleneck
Input Layer $\in \mathbb{R}^8$
Hidden Layer $\in \mathbb{R}^4$
Output Layer $\in \mathbb{R}^8$

CNN Autoencoder

- Given that our inputs in Computer Vision applications are images, using Convolution Neural Networks makes total sense
- Using Conv layers provides much better performance
- Encoder + Decoder

Training an Autoencoder

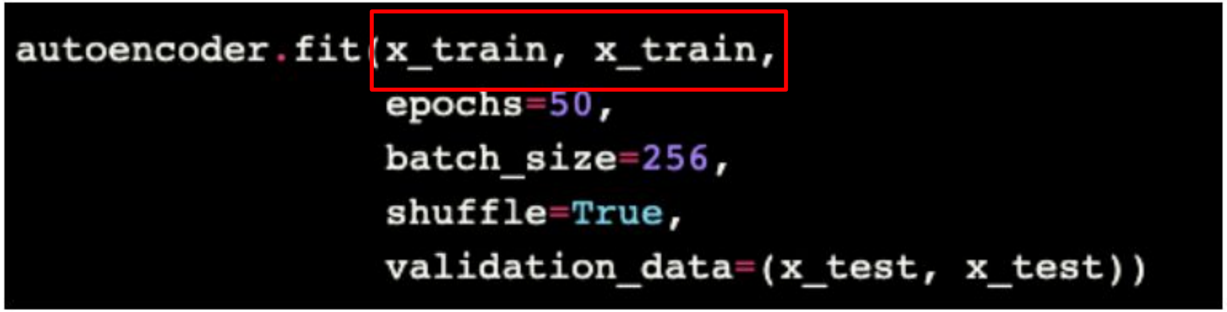
- The training process is simple, however there are few differences.
- The target data is the same as the training data
- Likewise for validation, as you’re testing how well your encoder-decoder model works
- The loss function can be binary cross entropy or even MSE.



```
autoencoder.fit(x_train, x_train,
                epochs=50,
                batch_size=256,
                shuffle=True,
                validation_data=(x_test, x_test))
```

Training an Autoencoder

- The training process is simple, however there are few differences.
- The target data is the same as the training data
- Likewise for validation, as you’re testing how well your encoder-decoder model works
- The loss function can be binary cross entropy or even MSE.



```
autoencoder.fit(x_train, x_train,
                epochs=50,
                batch_size=256,
                shuffle=True,
                validation_data=(x_test, x_test))
```

Autoencoder Limitations

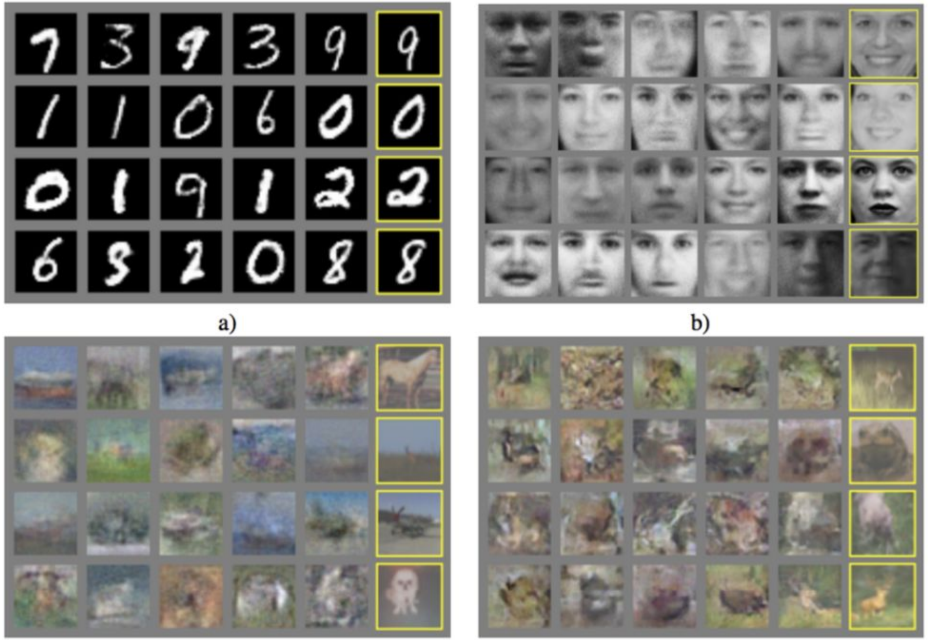
- Autoencoders are lossy, meaning the decompressed outputs are degraded compared to the original
- Data-specific meaning that it learns the representation in a specific domain
  - Ex: MNIST → hand written character (X)

What are GANs?

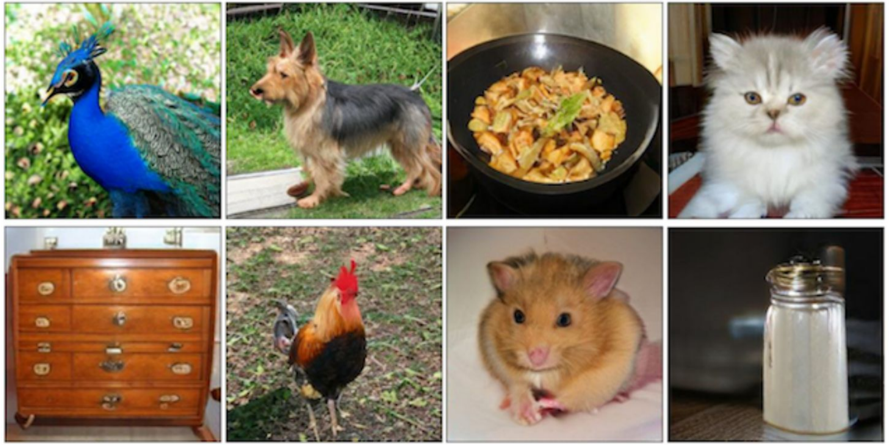
- Generative Adversarial Network (GAN) is a type of neural network that generates data that plausibly comes from an existing distribution of samples.

Google’s BigGAN

Face generation performed by GANs.
Taken from Progressive Growing of GANs for Improved Quality, Stability, and Variation, 2017.



Examples of GANs used to Generate New Plausible Examples for Image Datasets.



Example of Realistic Synthetic Photographs Generated with BigGAN. Taken from Large Scale GAN Training for High Fidelity Natural Image Synthesis, 2018.

(a)

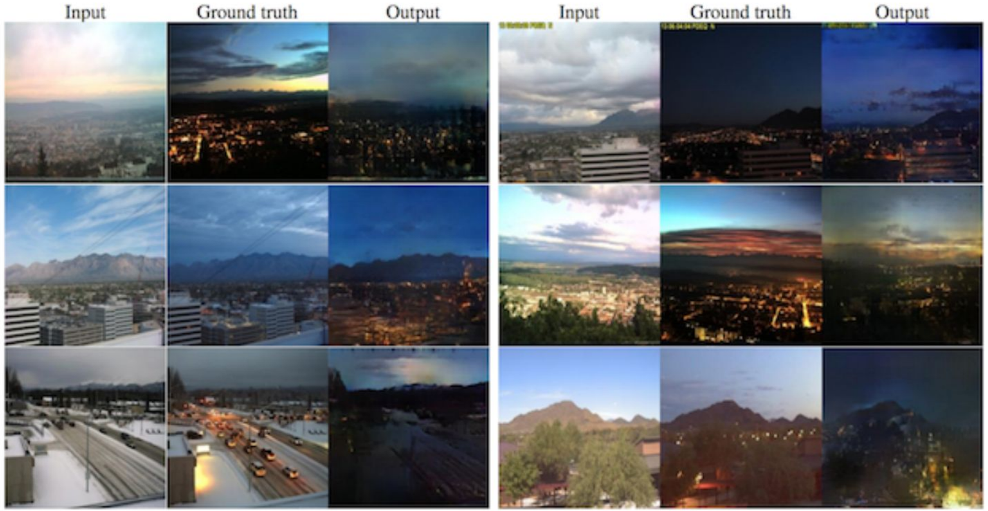
(b)

(c)

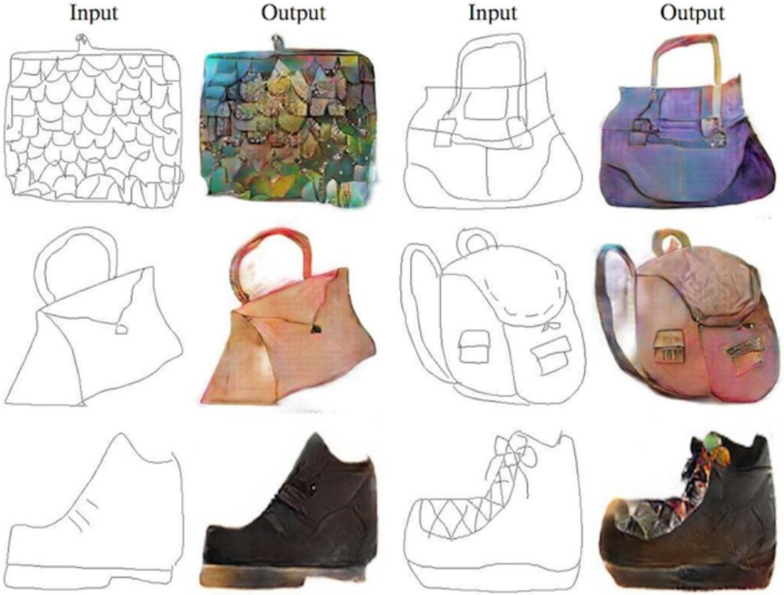
(d)

Examples of GANs used to Generate New Plausible Examples for Image Datasets.

Image-to-Image Translation

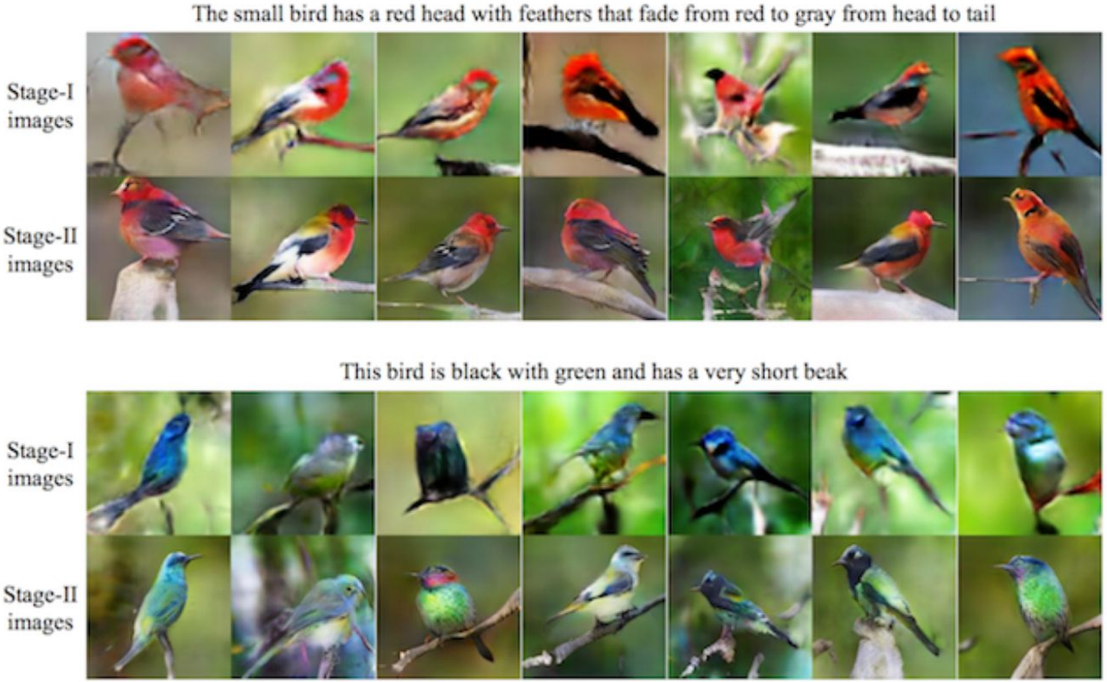


Example of Photographs of Daytime Cityscapes to Nighttime With pix2pix. Taken from Image-to-Image Translation with Conditional Adversarial Networks, 2016.



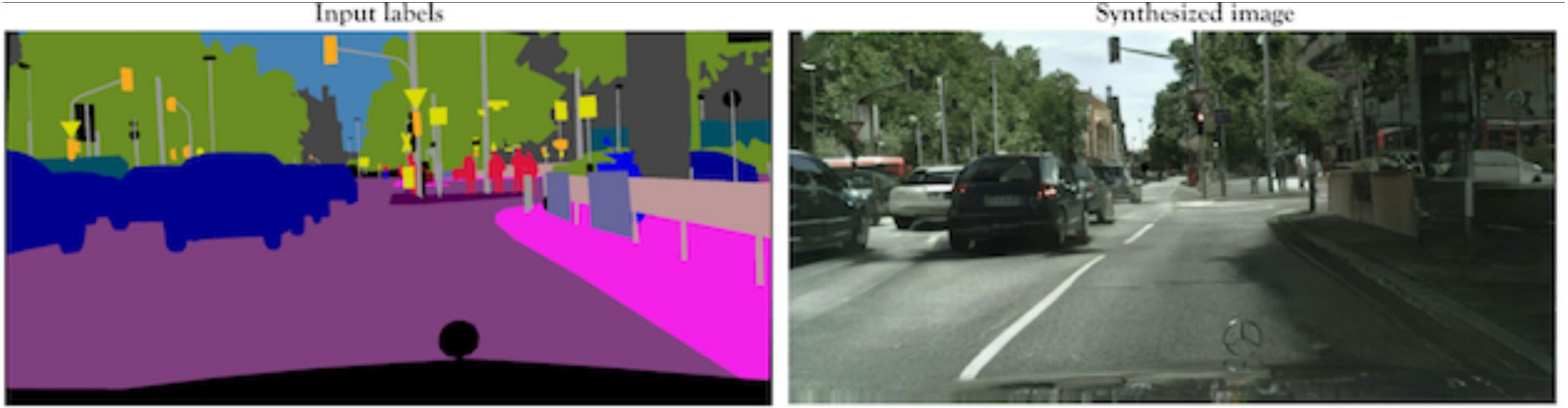
Example of Sketches to Color Photographs With pix2pix.Taken from Image-to-Image Translation with Conditional Adversarial Networks, 2016.

Text-to-Image Translation


The small bird has a red head with feathers that fade from red to gray from head to tail
Stage-I images
Stage-II images
This bird is black with green and has a very short beak
Stage-I images
Stage-II images

Example of Textual Descriptions and GAN-Generated Photographs of BirdsTaken from StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, 2016.

Semantic-Image-to-Photo Translation



Example of Semantic Image and GAN-Generated Cityscape Photograph.Taken from
High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs, 2017.

Super Resolution (SRGAN)

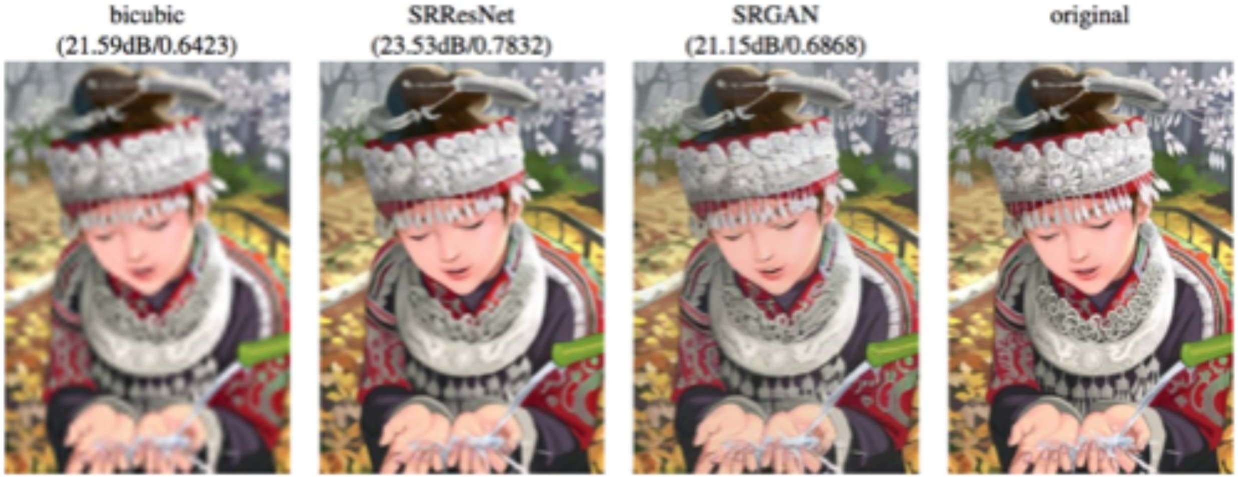
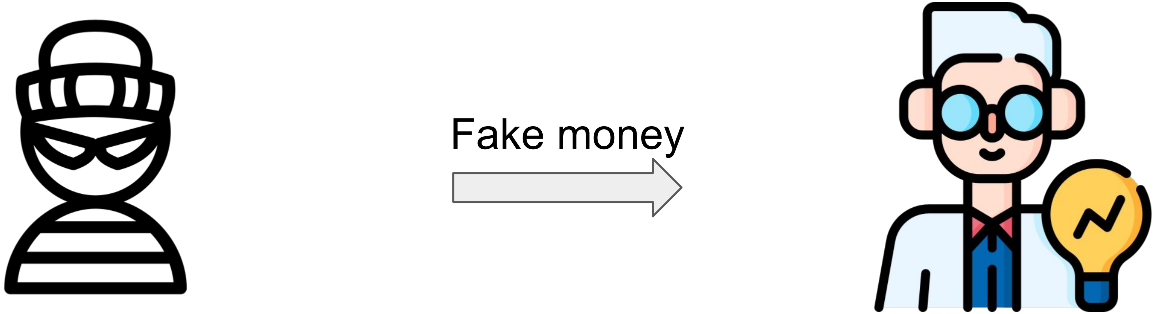


Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]

GANs Analogy



Counterfeiter

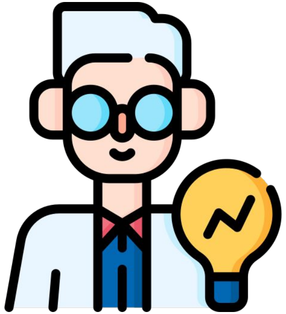
Fake money

Expert

GANs Analogy

- At first, he is no good

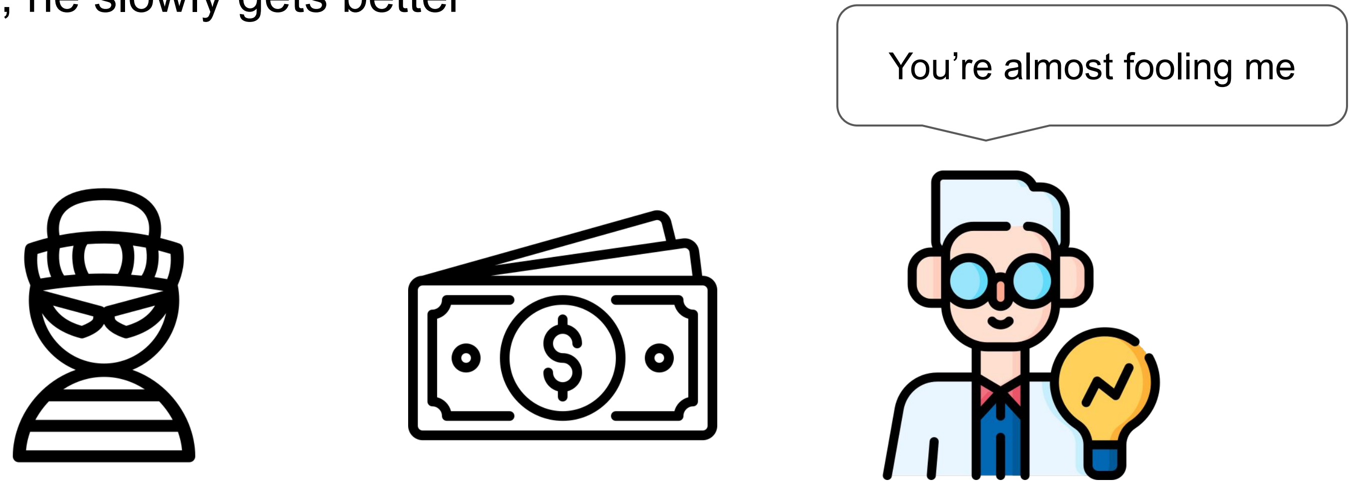
Counterfeiter



Expert

GANs Analogy

- Then, he slowly gets better



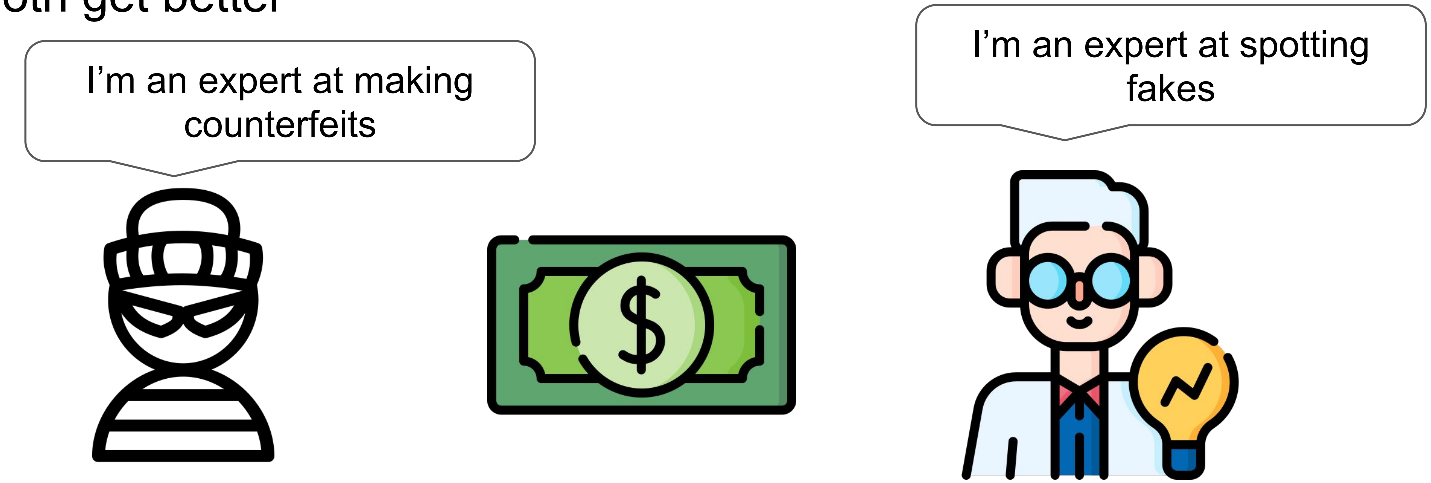
You’re almost fooling me

Counterfeiter

Expert

GANs Analogy

- Both get better



I’m an expert at making counterfeits

I’m an expert at spotting fakes

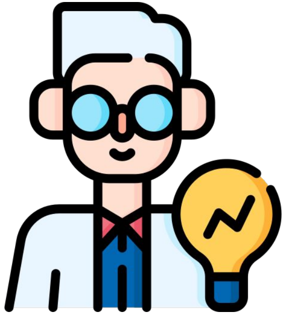
Counterfeiter

Expert

Two Componets of GANs

- In our analogy, we have two antagonistic networks contesting against each other
- Counterfeiter was the Genertator Network and Expert was the Discriminator

Counterfeiter

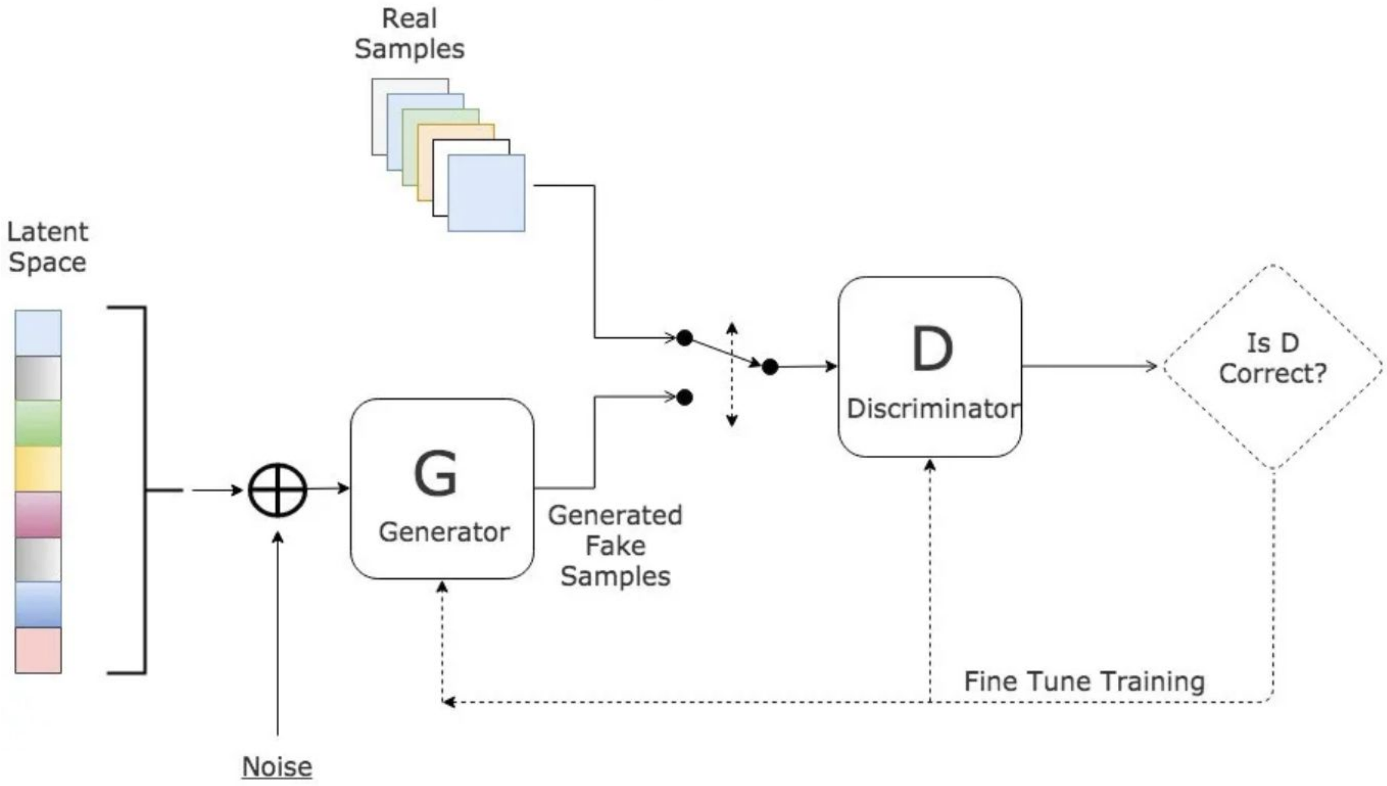


Expert

Generator & Descriminator Networks

- The purpose of Generator Network is to take random data initialization and decode it into a synthetic sample.
- The purpose of the Discriminator Network is to take this input from the Generator and predict whether this sample comes from the real data set.

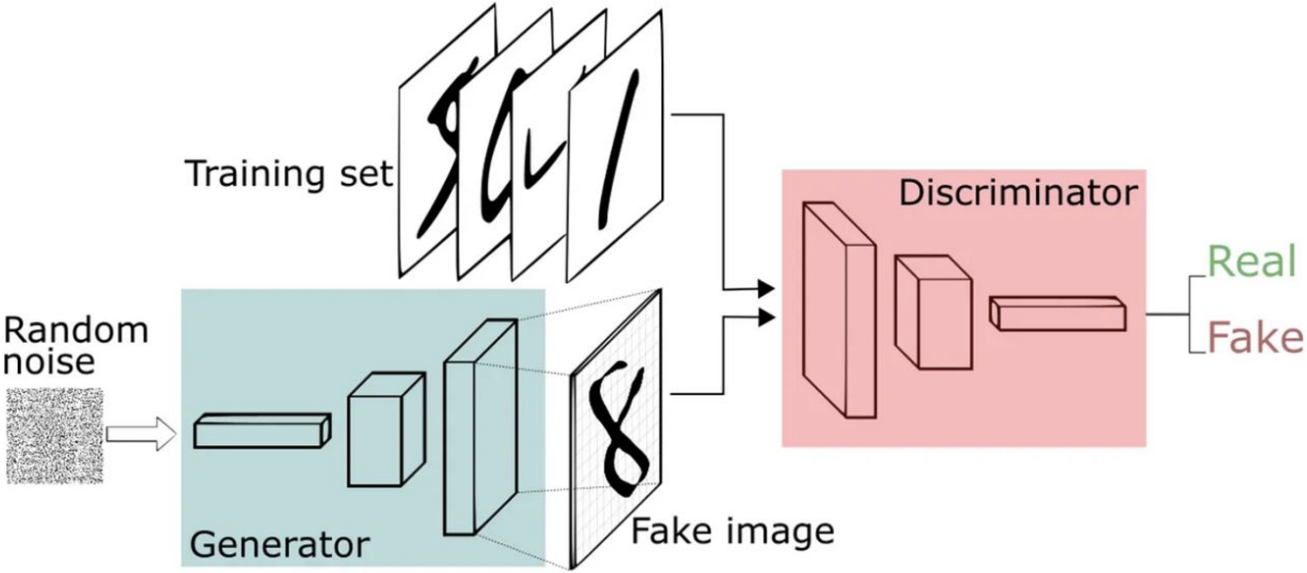
The Basic GAN Architecture



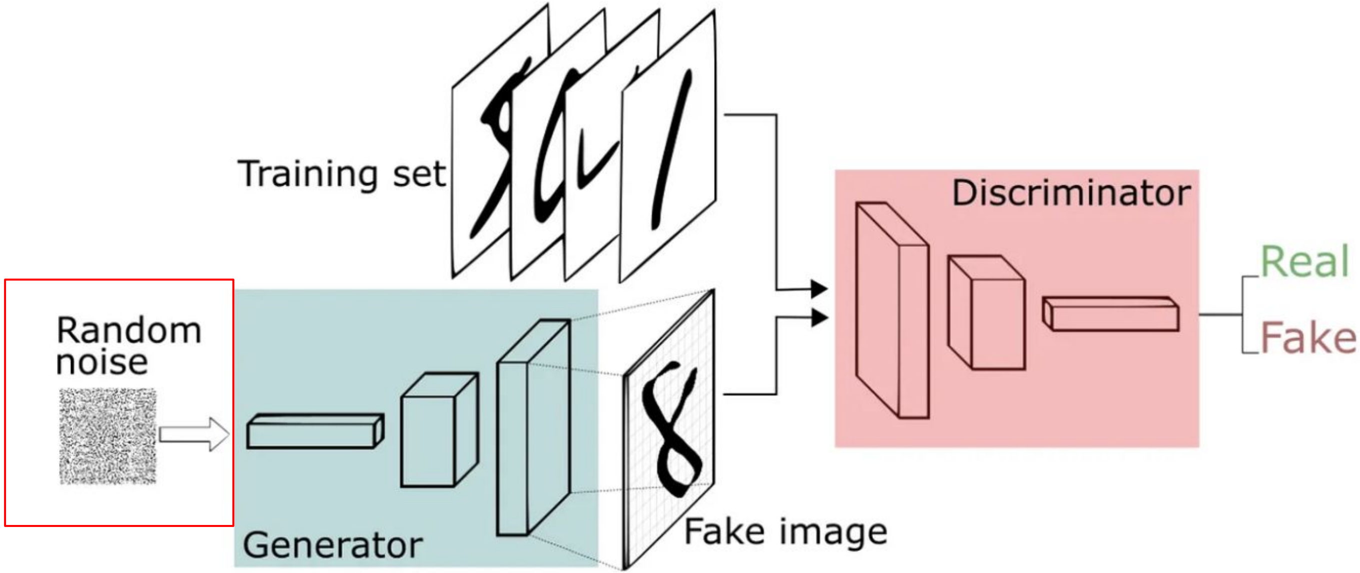
Training GANs

- Training GANs is notoriously difficult compared to neural networks, where we use gradient descent to change weights to reduce losses.
- In a GAN, each change in weight changes the overall balance of the dynamic system.
- Rather than seeking to minimize losses, we seek to find a equilibrium between our two opposing networks.
- Training stops when your discriminator (or you) can't tell the difference between real and fake data

The Training Process

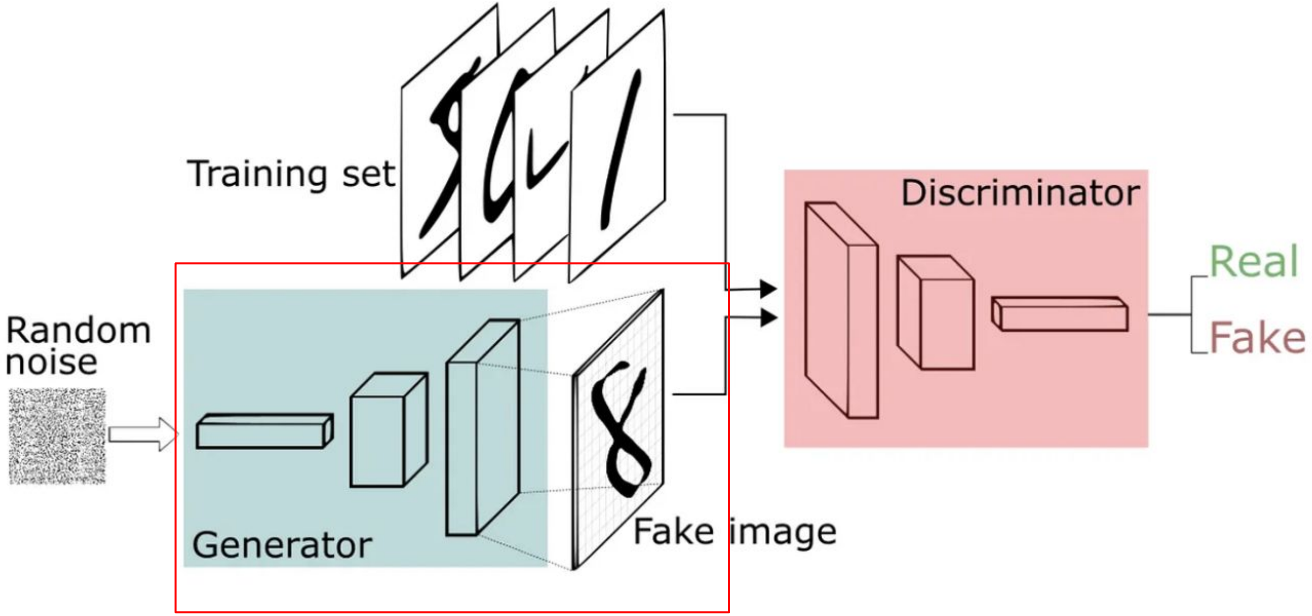


The Training Process



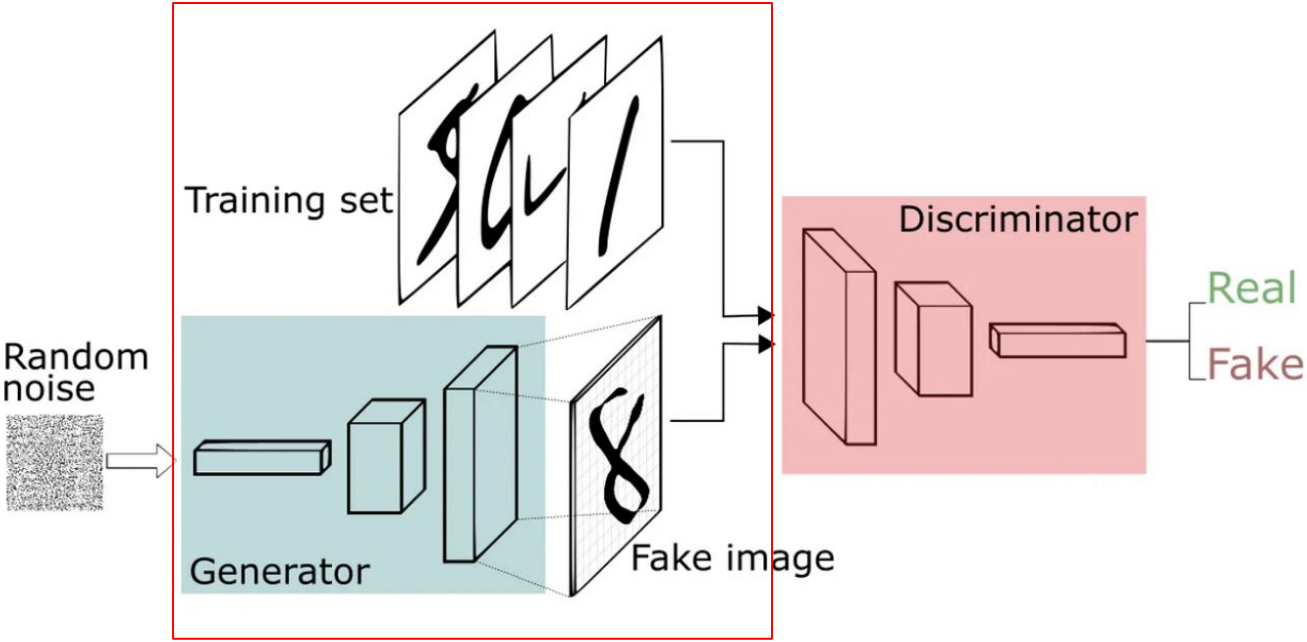
1. We randomly generate a noisy vector

The Training Process



2. Input this into Generator Network to generate a sample data

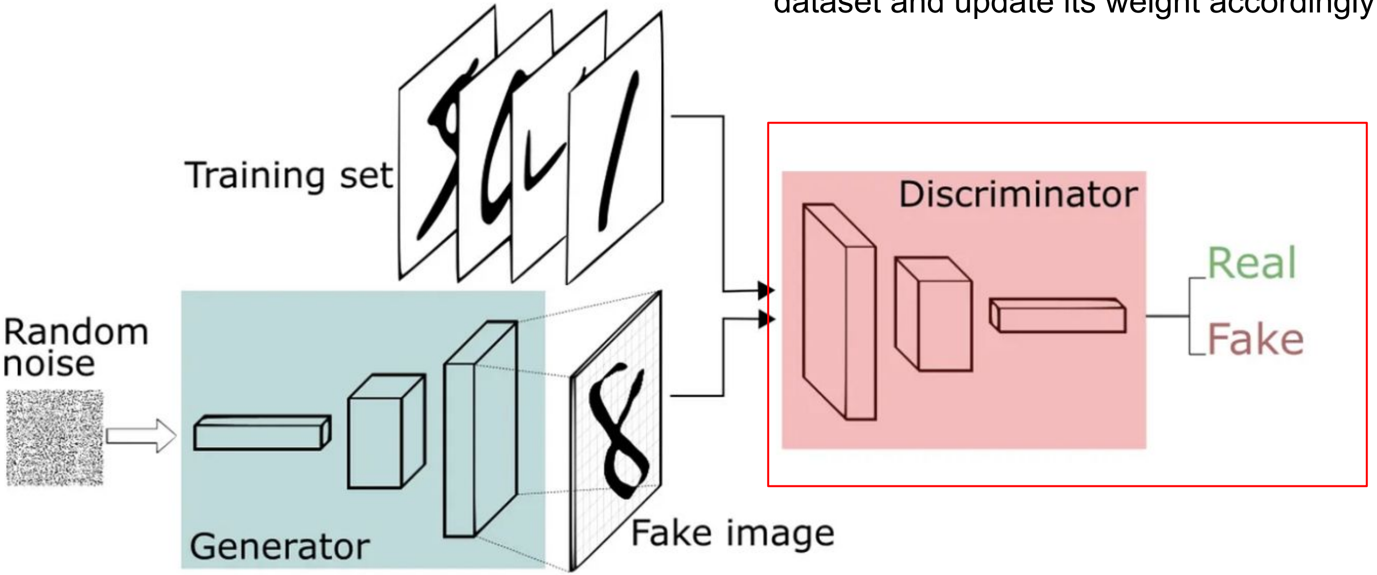
The Training Process



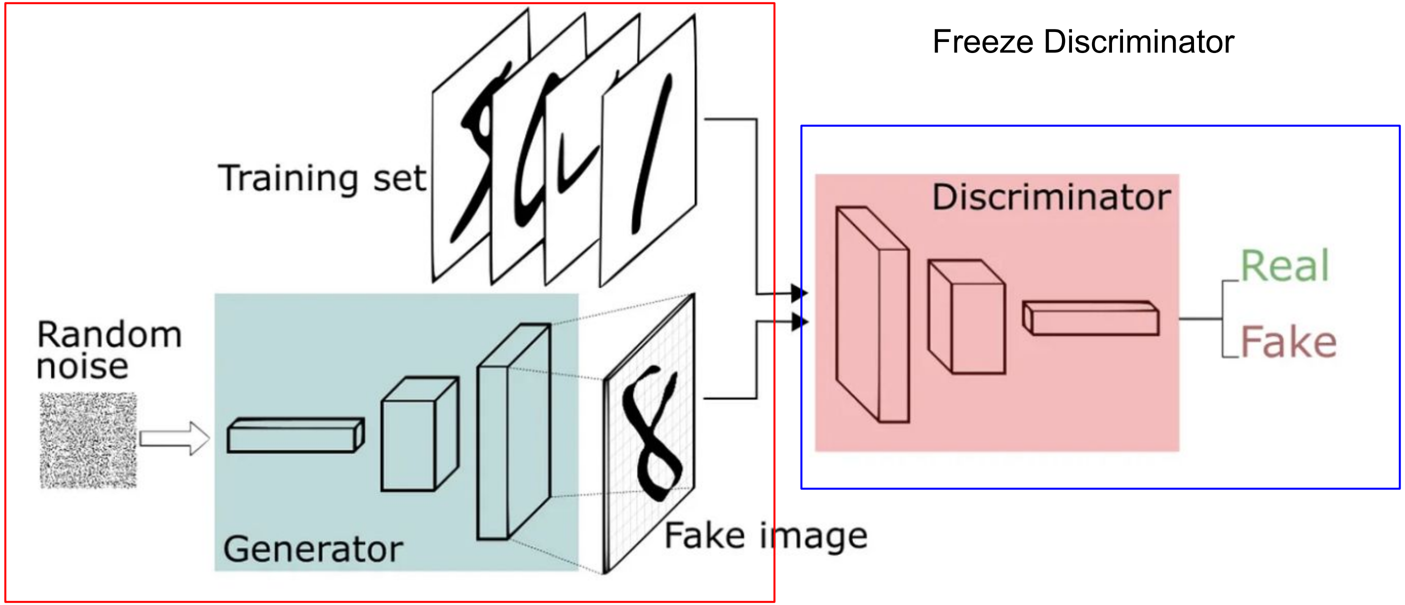
3. Take some sample data from our real data and mix it with some of our generated data

The Training Process

4. Train the Discriminator to classify this mixed dataset and update its weight accordingly

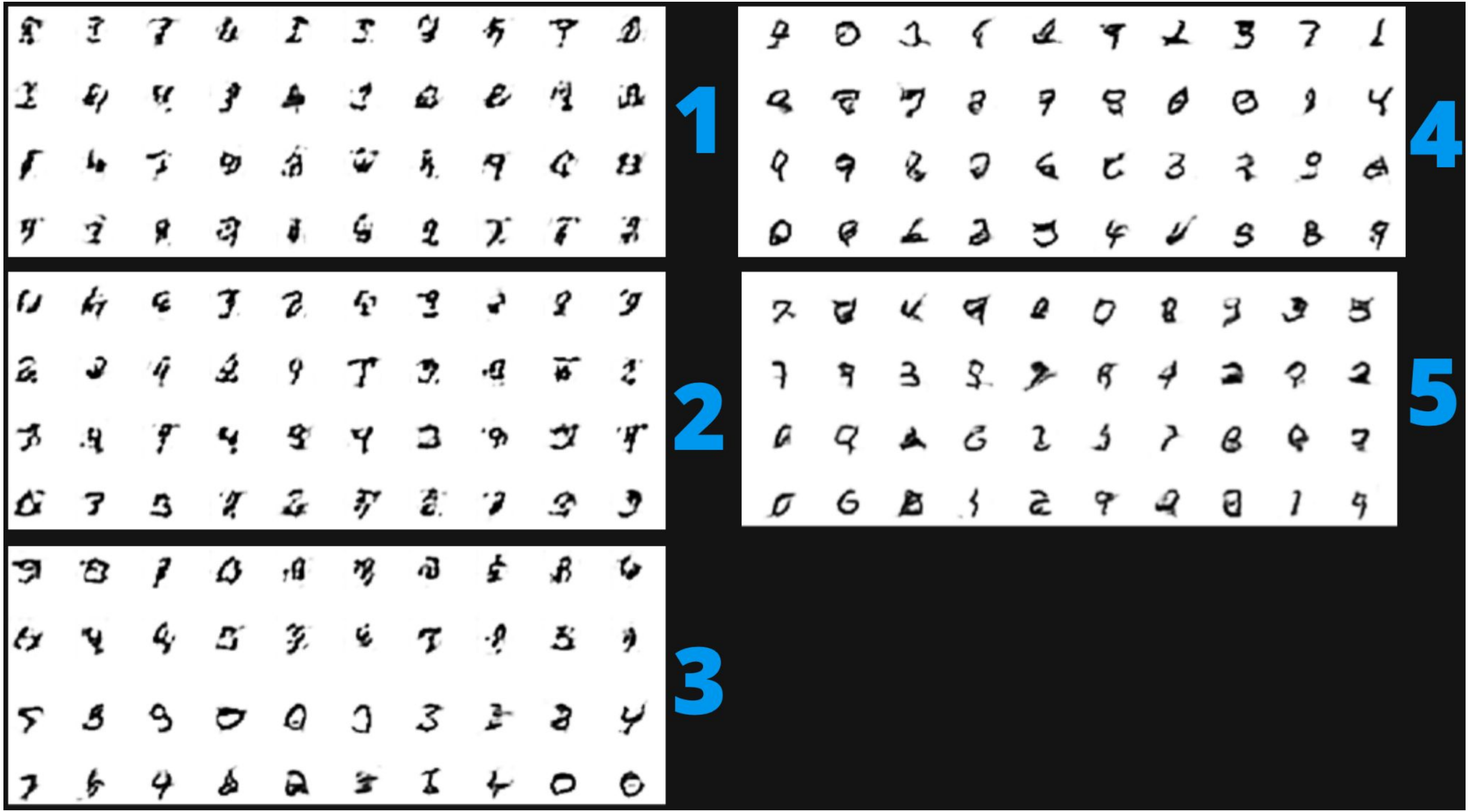


The Training Process



Freeze Discriminator

5. Now train the Generator. We make more random noisy vectors and create synthetic data. With the weights of Discriminator frozen, we use the feedback from the Discriminator to update the weights of Genertator.


1
2
3
4
5

Challenges in Training

- Achieving equilibrium: GAN training can be unstable due to the delicate balance between the generator and discriminator
- Hyperparameter tuning: Careful selection of hyperparameters such as learning rates, batch sizes, and the number of training steps is essential
- Mode Collapse and Non-Convergence: happens when regardless of the noisy fed into your generator, the generated output varies very little. It occurs when a small set of images look good to the Discriminator and get scored better than other images. The GAN simply learns to reproduce those images over and over (similar to overfitting)

Practical Use Case of GANs

- Creative industries - Art, music and design
- Deep Fakes - Replicating facial style in video
- Security - Privacy preserving, enhancing poor CCTV feeds
- Medical applications - Data augmentation and drug discovery
- Photography - Samrtphone and camera scene enhancement

Video and image effects: Special effect industries

Video games: Graphic enhancement using DLSS

Video compression: NVIDIA’s Maxine

Marketing Materials: Virtual try-ons

Autonomous vehicles

Space and physics: improve astronomical images

Creative Industries

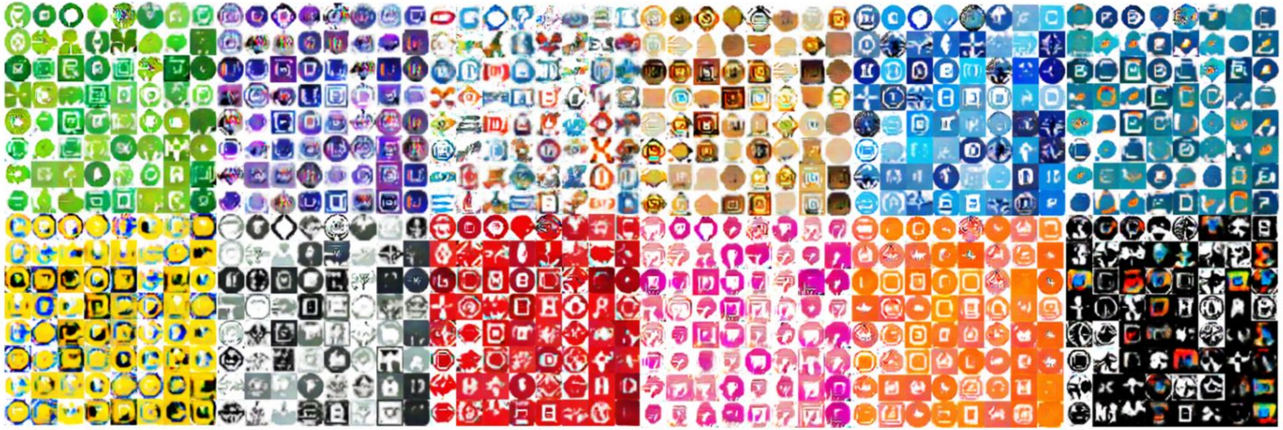
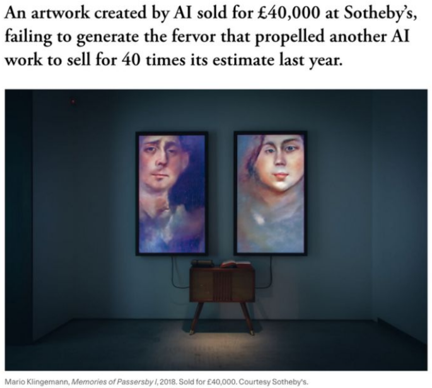
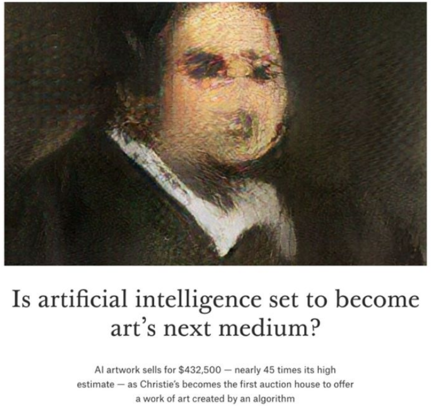


Fig. 6. Results from the generation of 64 logos per class after 400 epochs of training. Classes from left to right top to bottom: green, purple, white, brown, blue, cyan, yellow, gray, red, pink, orange, black.



An artwork created by AI sold for £40,000 at Sotheby's, failing to generate the fervor that propelled another AI work to sell for 40 times its estimate last year.

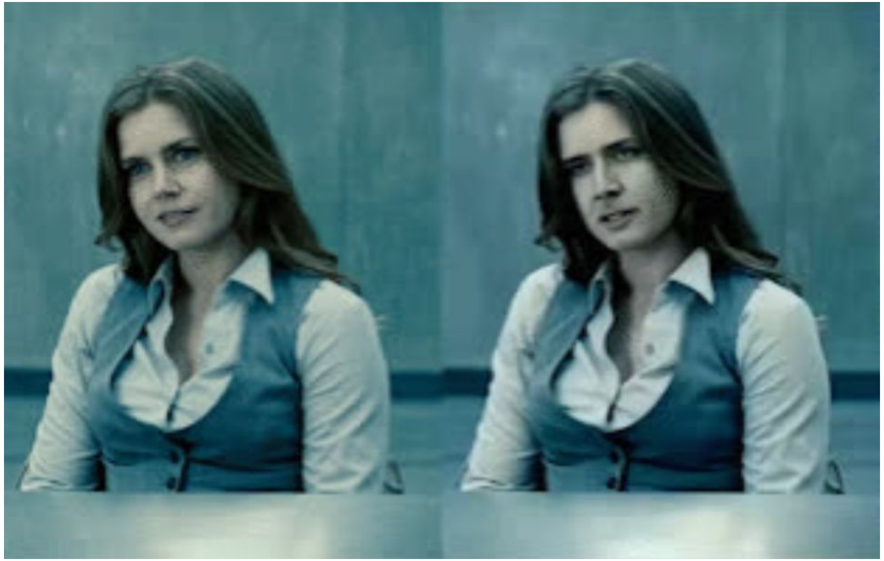
Mario Klingemann, *Memories of Passersby I*, 2018. Sold for £40,000. Courtesy Sotheby's.

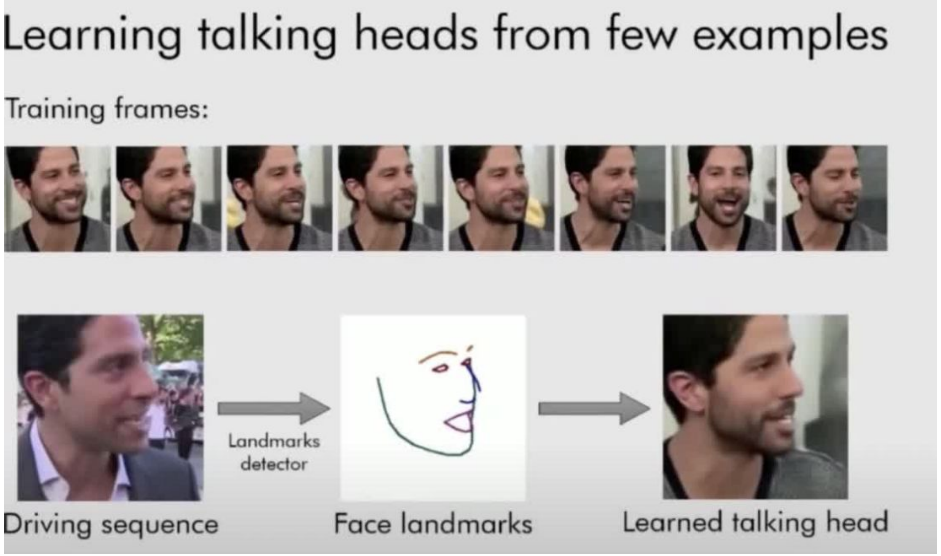


Is artificial intelligence set to become art's next medium?

AI artwork sells for $432,500 — nearly 45 times its high estimate — as Christie's becomes the first auction house to offer a work of art created by an algorithm

Deep Fakes



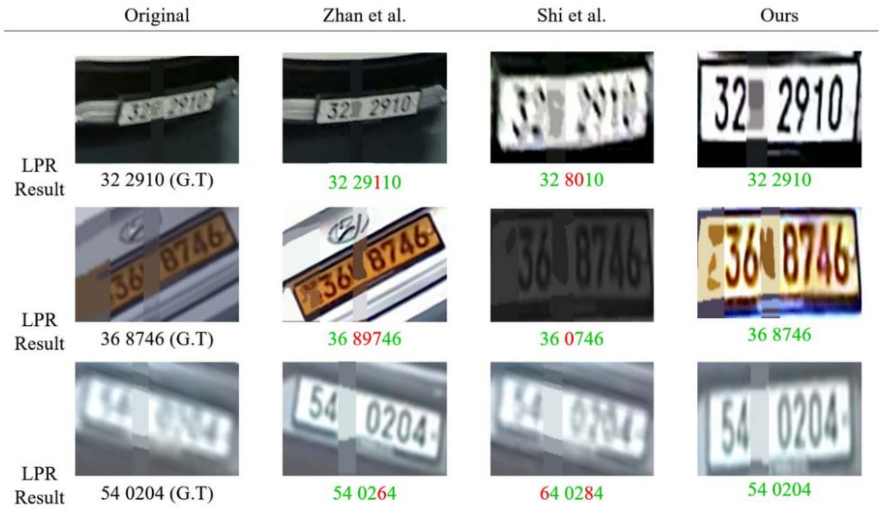


Learning talking heads from few examples

ArcaneGAN

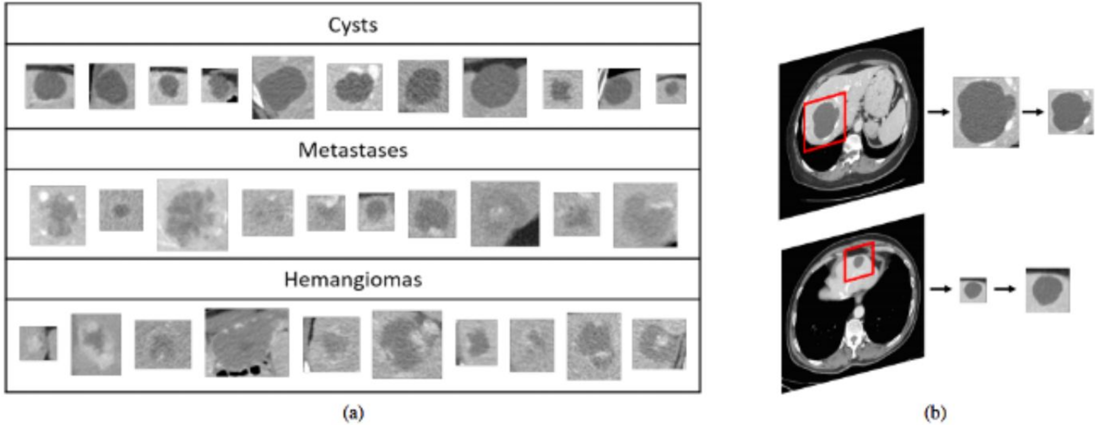
Security

- Privacy: Instead of sharing real data, we can share synthetic data that is indistinguishable from the real
- CCTV footage enhancement: SNIDER: Single Noisy Image Denoising and Rectification for Improving Liscence Plate Recognition

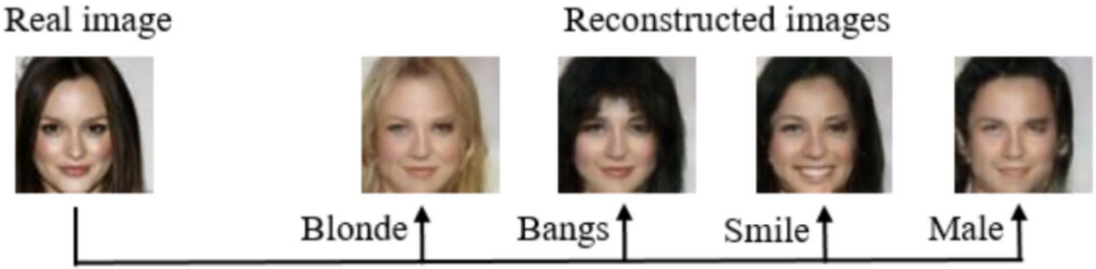


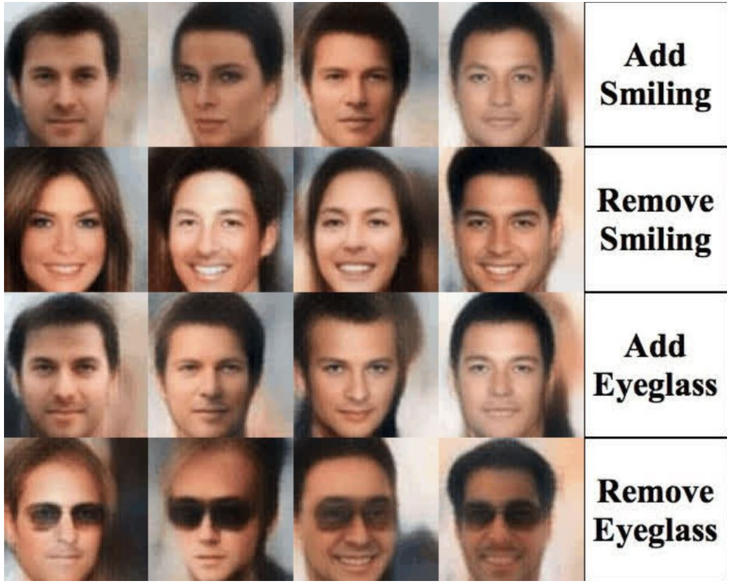
Medical Applications

- Drug discovery: GANs can quickly generate novel biologycal components to test hypothesis simultaneously
- Data augmentation: GANs were used to augment medical brain scan CT images which improved the sensitivity and specifity of their Brain Disease classifier to 85.7% and 92.4%, respectively

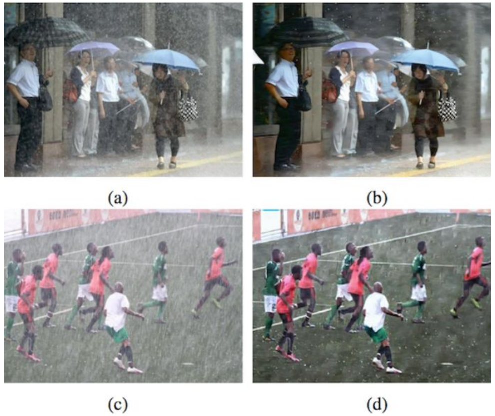


Photography





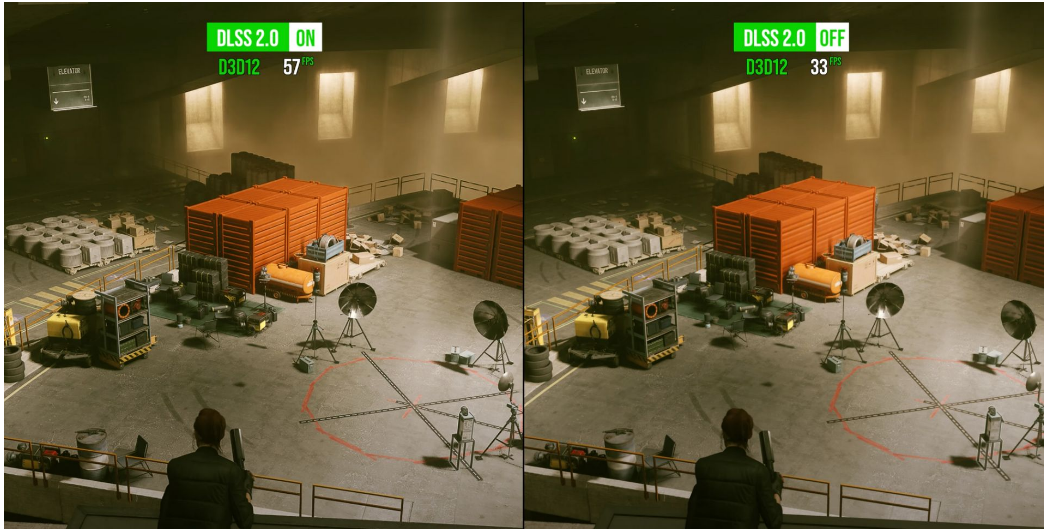
"Invertible Conditional GANs for Image Editing" https://arxiv.org/abs/1611.06355



“Image De-raining Using a Conditional Generative Adversarial Network" https://arxiv.org/abs/1701.05957

Videogame Graphics

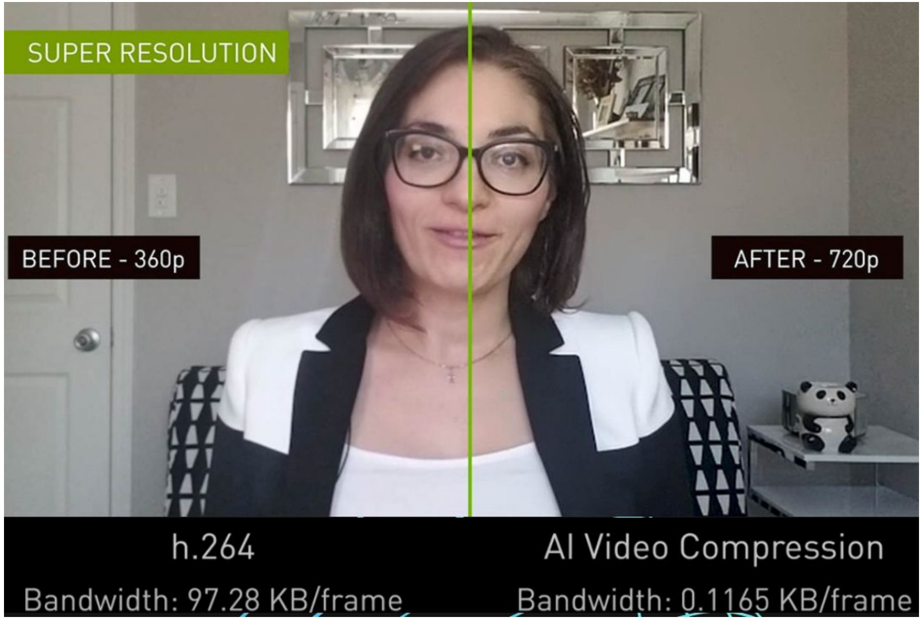
- Deep Learning Super Samplinig (DLSS): It enables the upscaling of lower-resolution images to a higher-resolution for display on higher-resolution displays (4K), while rendering natively at lower resolutions



https://www.nvidia.com/en-us/geforce/news/graphics- reinvented-new-technologies-in-rtx-graphics-cards/#dlss

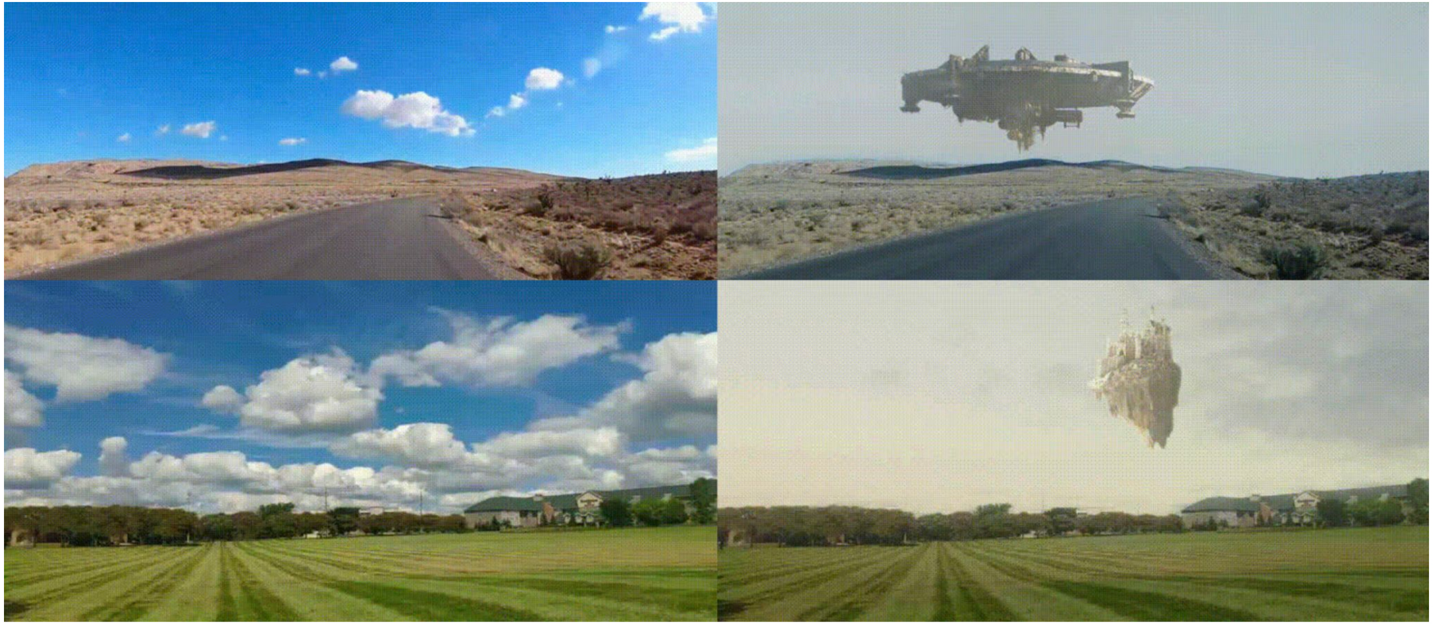
Video Compression - NVIDIA’s Maxine

- A generative adversarial network on the receiver’s side uses the initial image and the facial key points to reconstruct subsequent images on a local GPU.



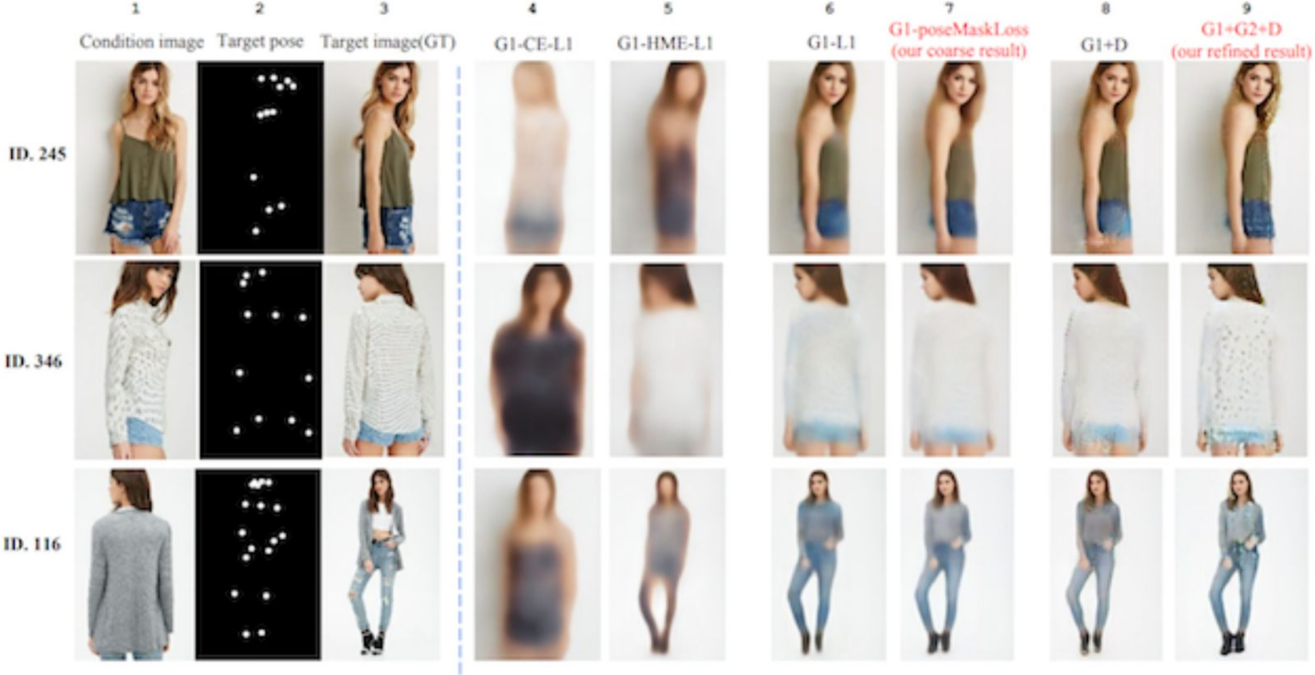
Video and Image Effects

- Sky Replacement - “Castle in the Sky: Dynamic Sky Replacement and Harmonization in Videos”, built using pytorch-CycleGAN-and-pix2pix



Marketing Materials

- Pose Guided Person Image Generation



Autonomous Vehicles

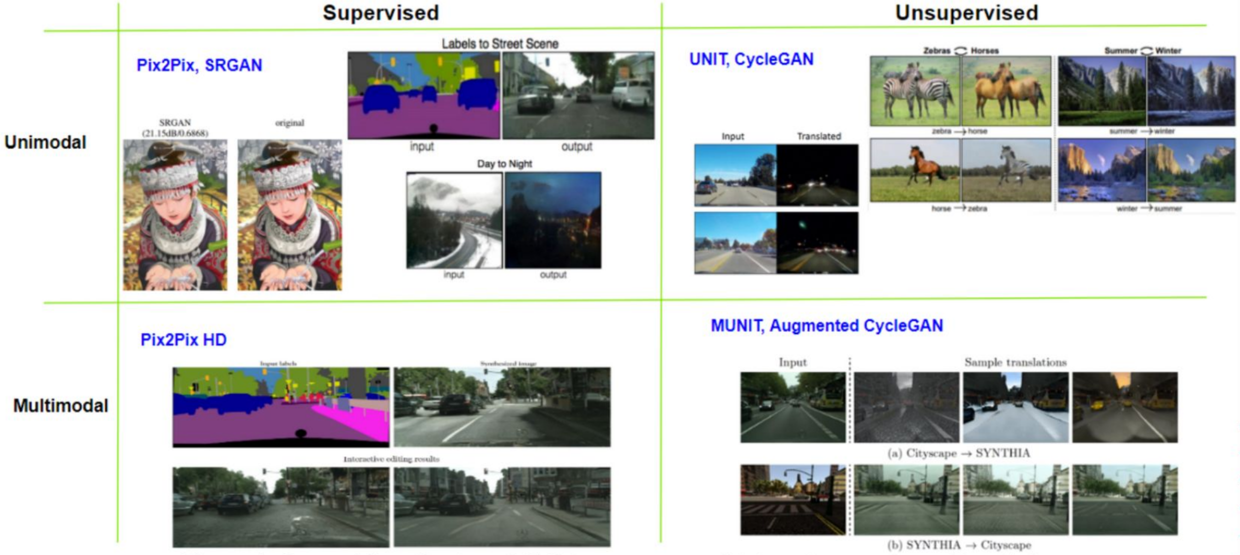
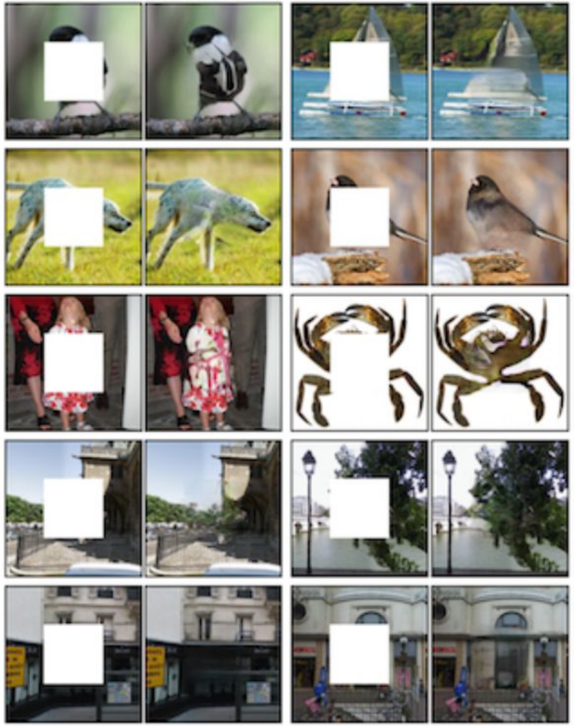


Figure 4: *Successful application of GAN for autonomous driving - Image to Image Translation*

GAN-Generated Photograph Inpainting Using Context Encoders.



Implementation

Download notebook at:

https://github.com/albert831229/nchu-computer-vision/tree/main/113/day