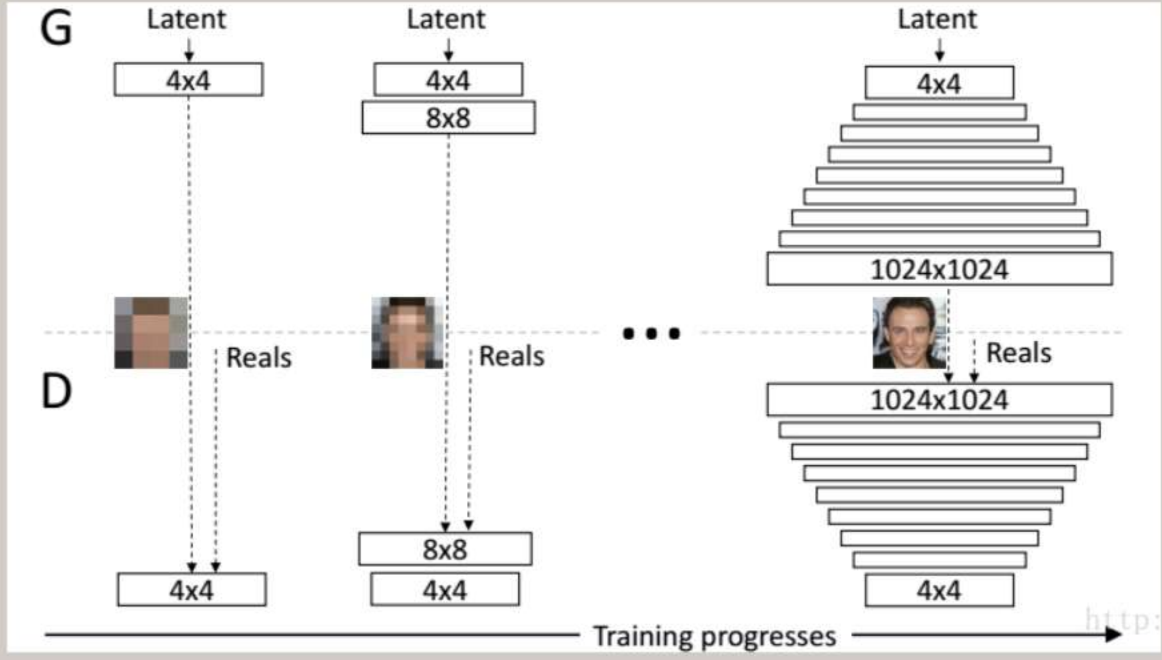
SYNTHETIC DATA GENERATION USING NEURAL NETWORKS



Index

- What is Generative AI?
- Techniques used for AI based Image Generation:
    - Vanilla GAN
    - DCGAN
    - Diffusion-based Model
    - ProGAN
- Techniques used for AI based Style Transfer:
    - cycleGAN
    - Pix2Pix
- Some Applications in military domain
- Final Remarks

Can you spot which one is real and which one is AI generated?

Techniques used for image generation:

GANs:
Uses two neural network one called **discriminator** and one called **generator**.
Generator- takes a random vector of a pre-defined latent space dimension as an input and upscales the image using various neural network layers to ouput a tensor defining an image.
Discriminator- is a simple classifier which outputs whether an image is real(1) or fake(0). Takes a real image from the training dataset and an image produced by passing a random vector through the generator and classifies them as real or fake.

The generator's task is to fool the discriminator into assigning maximum number of false positives as it's predictions. Three different types of loss functions are defined to train our generator and discriminator...

Techniques used for image generation:

Diffusion Models:
Diffusion Models are generative models which have been gaining significant popularity in the past several years, and for good reason. A handful of seminal papers released in the 2020s alone have shown the world what Diffusion models are capable of, such as beating GANs on image synthesis.

- Diffusion Models work by destroying training data through the successive addition of Gaussian noise, and then learning to recover the data by ... After training, we ... generate data by ... noise through ...

- Forward Process -> Adding Gaussian Noise

- Backward Process-> Predicting image from noise through a fixed markov chain

Final Remarks VAE v/s GANs v/s Diffusion Models

Paradigm	Quality	Diversity	Speed
VAE	✗	✓	✓
GAN	✓	✗	✓
Diffusion	✓	✓	✗

- For these reasons we will be majorly focusing on GANs for our application. We will also be experimenting with diffusion models as they have shown to beat some GAN models in terms of quality of image synthesis. The model we have selected to investigate and fine tune for our application is proGAN with some ideas inspired from the recent papers of styleGAN generated by nvidia.

The goal of this project is to make a gui based application which can help develop datasets for training different models making them more robust to unseen scenarios. We are also investigating image translation techniques such as cycleGAN and pix2pix for conversion of one style of images to another for example. Day -> Night, Summer -> Winter, Aerial -> Map etc.
These models can significantly improve the models implementing on military technologies and hence make them more reliable in unseen environments.

Some Applications of Generative AI

- Image Generation
- Semantic Image to Photo generation
- Super Resolution Image
- 3D Shape Generation
- Text-to-Speech Generation
- Text Generation, Code Generation
- Music, Video Generation
- DeepFake
- Image-to-Image Translation

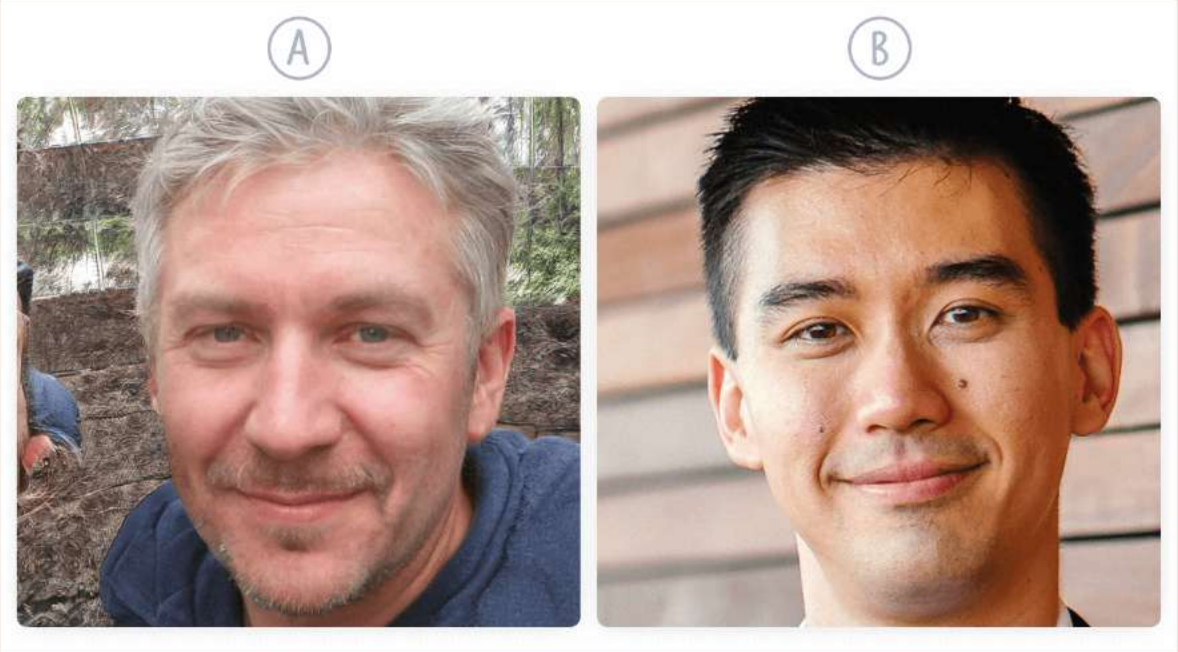
Night-to-Day Conversion

Generative AI in real life

Potential Use Cases

Index

- What is Generative AI?
- Techniques used for AI based Image Generation:
    - Vanilla GAN
    - DCGAN
    - Diffusion-based Model
    - ProGAN
- Techniques used for AI based Style Transfer:
    - cycleGAN
    - Pix2Pix
- Some Applications in military domain
- Final Remarks



Can you spot which one is real and which one is AI generated?

What is Generative AI?

Generative AI is a sub-branch of artificial intelligence study where we try to create synthetic data using various techniques. These techniques then them be used to produce various types of data such as images, audio, video, tabular data, etc.

These techniques majorly include:

1) **Variational AutoEncoders**- A technique based on Bayesian machine learning. Incapable of high quality image generation.

2) **Generative Adversarial Networks**- Utilizes two networks in an adversarial(competing) environment to generate high quality images.

3) **Diffusion Models**- A latent variable model that learns to recover data from a random noise using a fixed markov chain. The model learns by first destroying the training data through successive addition of gaussian noise.

What is Generative AI?

Generative AI is a sub-branch of artificial intelligence study where we try to create synthetic data using various techniques. These techniques then them be used to produce various types of data such as images, audio, video, tabular data, etc.

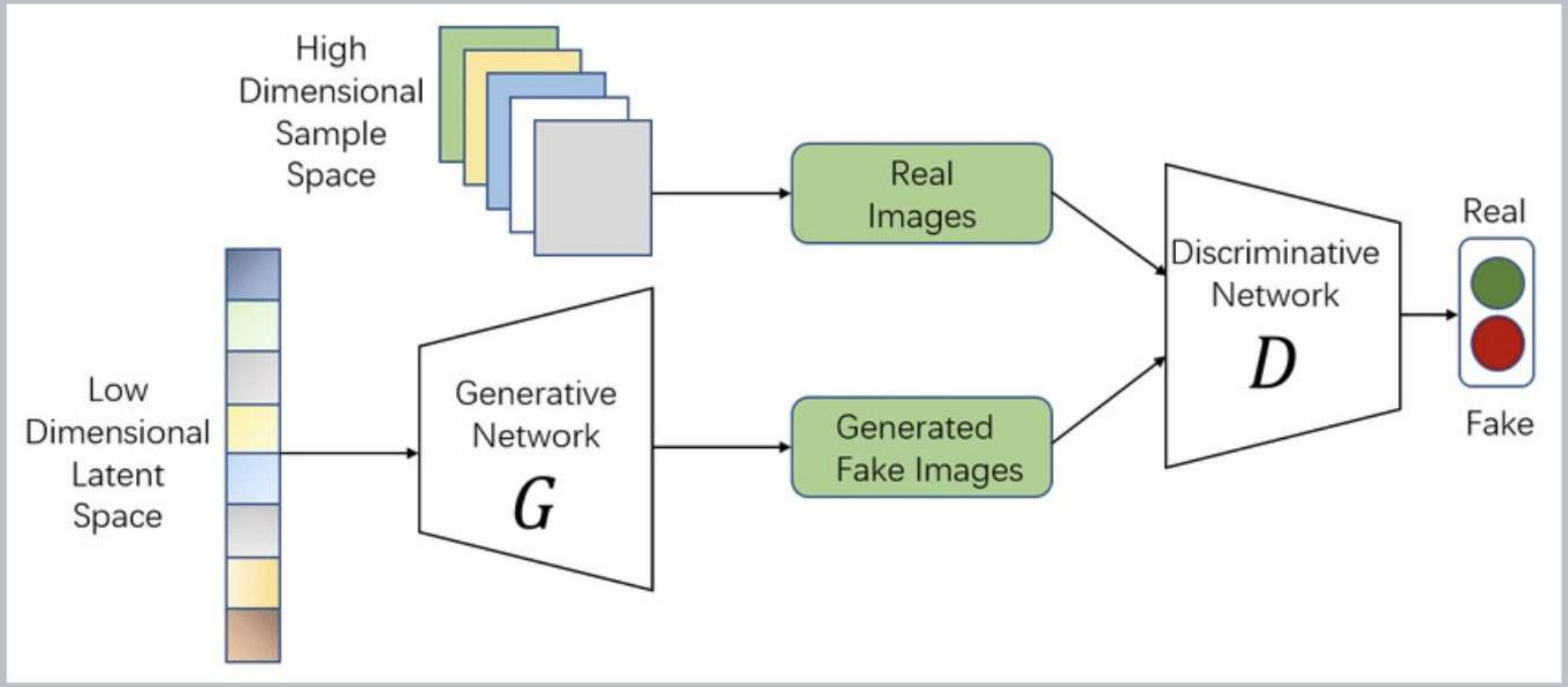
These techniques majorly include:

1) **Variational AutoEncoders**- A technique based on Bayesian machine learning. Incapable of high quality image generation.

2) **Generative Adversarial Networks**- Utilizes two networks in an adversarial(competing) environment to generate high quality images.

3) **Diffusion Models-** A latent variable model that learns to recover data from a random noise using a fixed markov chain. The model learns by first destroying the training data through successive addition of gaussian noise.

Different neural network architectures used in GAN:



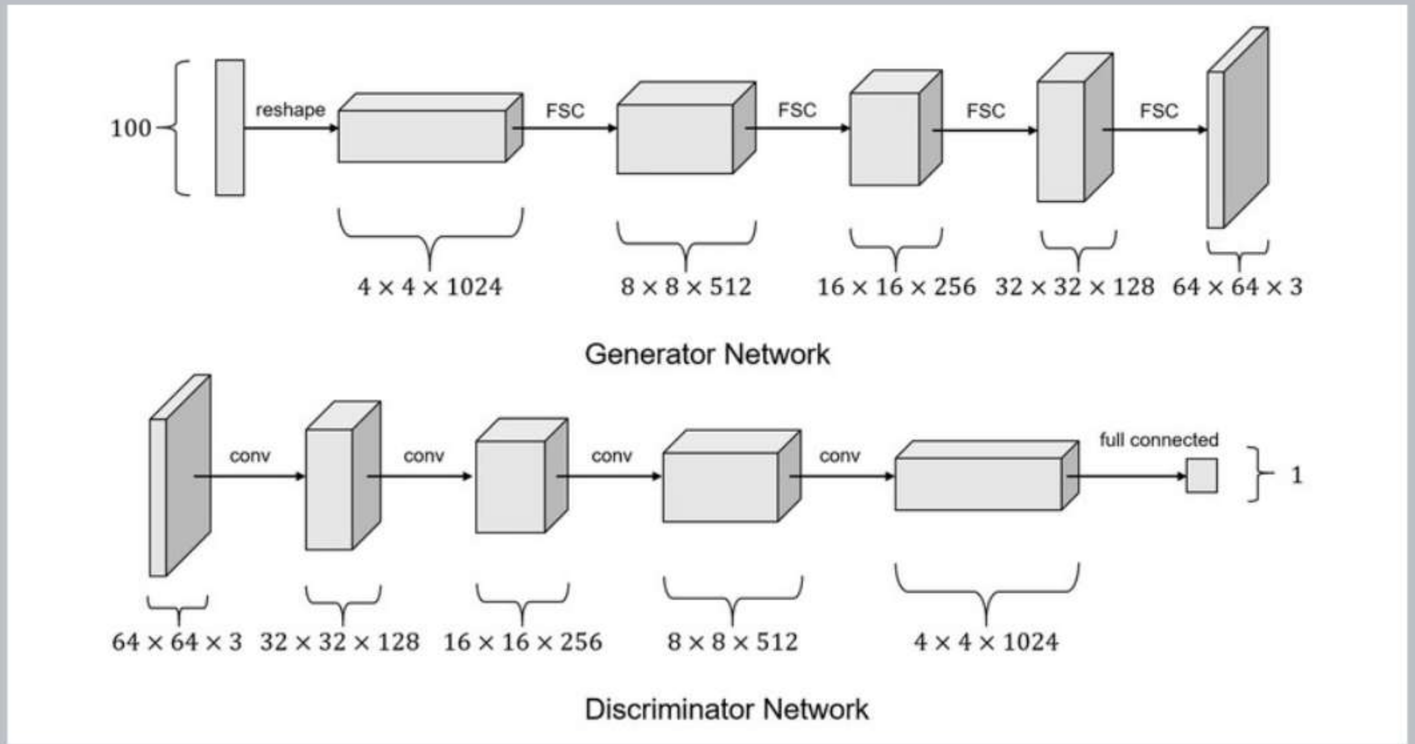
General Network of GAN

Vanilla GAN:

- It is the simplest GAN architecture where we use simple artificial neural networks layers to train the generator and discriminator. Mostly **NS-Loss** is used with Vanilla GAN but **WGAN-GP** can also be used for better results.

Deep Convolutional GAN:

- This architecture uses convolutional and transpose convolutional layers in the discriminator and generator respectively to generate higher quality images than **Vanilla GAN**.



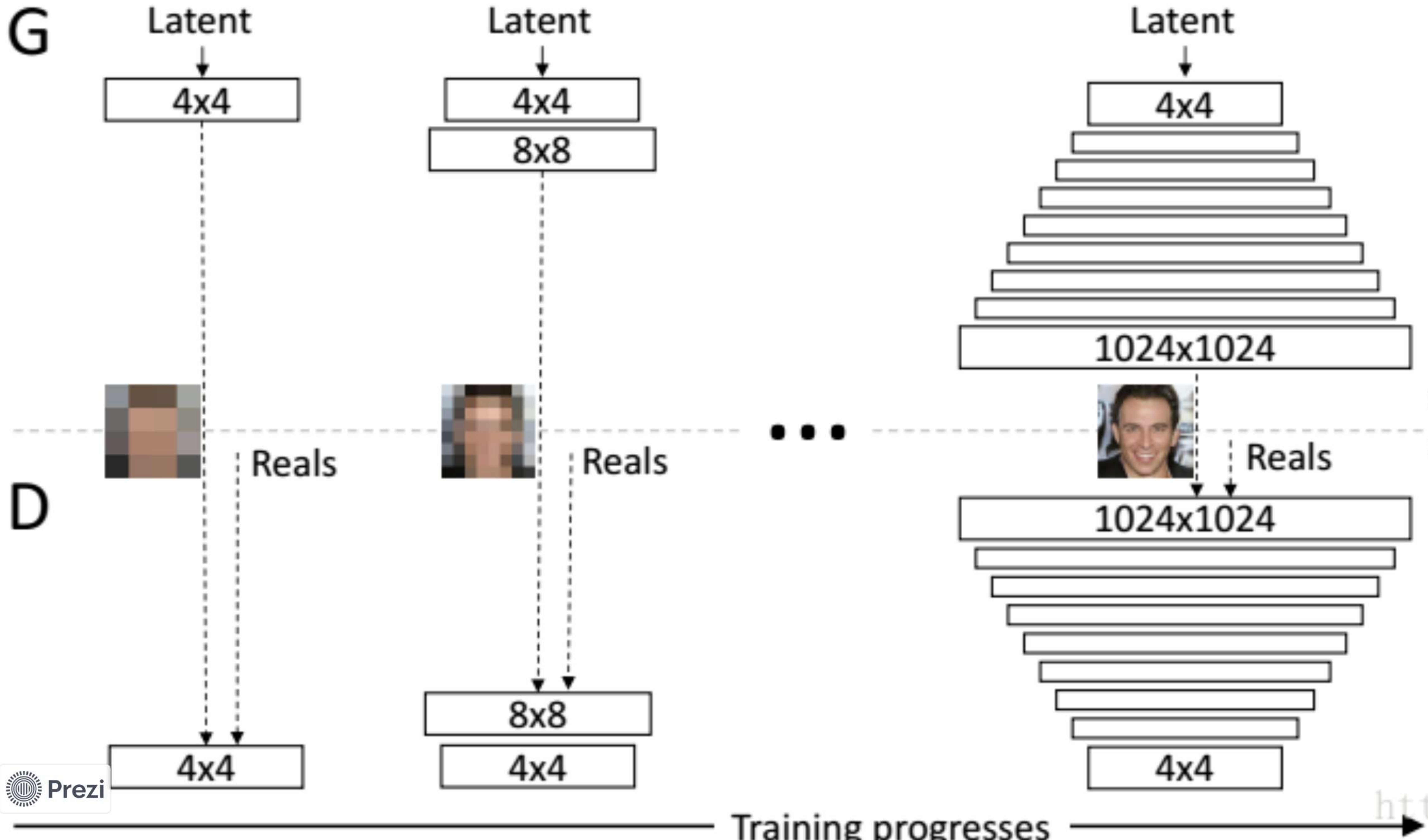
Prezi

Progressively Growing GAN(proGAN):

proGAN is one of the best GAN architecture available which improves on previously mentioned architectures in terms of quality, speed, and diversity in image generation. The key idea of progressively growing GAN is to first train a model on a small dataset where the discriminator is the mirror of the generator network and more convolutional layers are added in the network after the model has gained some decent results for the previous resolution. When the resolution is doubled after training of the lower resolution a transition layer with a parameter alpha is introduced to slowly transition from current resolution to next resolution.

This method seems more natural as humans also learn in a progressively growing manner. The research paper on proGAN described three more important ideas for implementation of proGAN ensuring training convergence and stability:

1) Mini-batch standard deviation- Variance of a generated batch of images is calculated and added as a layer towards the end in the discriminator term allowing more dirverse examples to be produced.
2) Pixel Normalization- All batch normalization layers were replaced by a pixel normalization technique which doesn't have any learnable parameter giving more stable results.
3) Equalized Learning Rate: Before every forward pass learning rates are equalized by scaling the weights. Ensures that the learning rate doesn't depend on the dynamic range of that feature.
Progam architecture uses **WGAN-GP loss** for optimization.


G
Latent
4x4
Latent
4x4
8x8
Latent
4x4
1024x1024
Reals
Reals
Reals
D
4x4
8x8
4x4
1024x1024
4x4
Training progresses

Techniques used for image generation:

GANs:
Uses two neural network one called **discriminator** and one called **generator.**
Generator- takes a random vector of a pre-defined latent space dimension as an input and upscales the image using various neural network layers to ouput a tensor defining an image.
Discriminator- is a simple classifier which outputs whether an image is real(1) or fake(0). Takes a real image from the training dataset and an image produced by passing a random vector through the generator and classifies them as real or fake.

The generator's task is to fool the discriminator into assigning maximum number of false positives as it's predictions. Three different types of loss functions are defined to train our generator and discriminator...

Losses for GAN:

1) MM Loss:
Min-Max is very common loss function for adversarial environments and here the generators loss is negative of discriminators loss. Here, discriminator is trying to minimize it's loss and at the same time generator is trying to minimize the negative of that loss. This is rarely used in practice and it can blow up training due to competing weight changes among discriminator and generator and hence they can get stuck. Here, Jd describes the maximum likelihood loss of the discriminator and Jg is the loss of the generator.

$J^D = E_{x\sim p_r}\log[D(x)] + E_{z\sim p_g}\log[1 - D(G(z))]$ $J^G = -J^D$

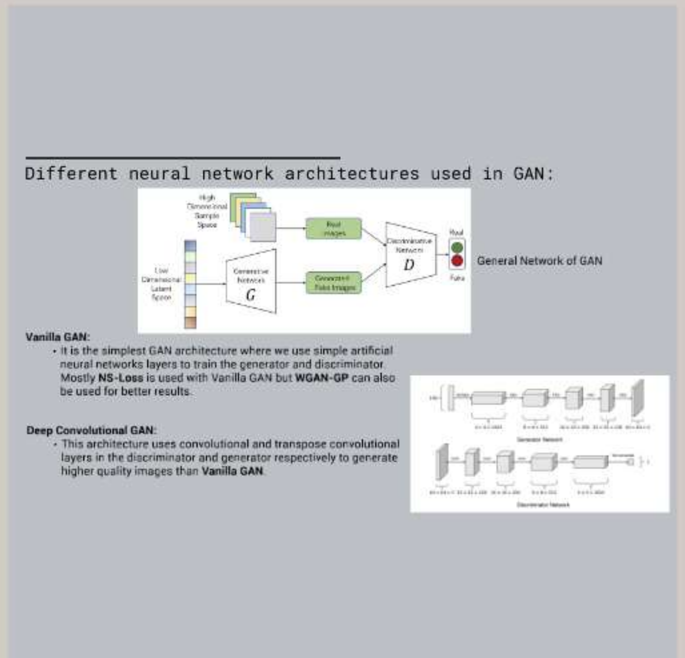
2) Non Saturation Loss:
A slight modification is made to the min-max loss where instead of using the negative of discriminator loss as generator loss just the false positives part of the loss is analyzes and the generator tries to maximize the false positives of the generator. This is the most commonly used loss function for training GANs.

$J^D = E_{x\sim p_r}\log[D(x)] + E_{z\sim p_g}\log[1 - D(G(z))]$ $J^G = E_{z\sim p_g}\log[D(G(z))]$

3) Wasserstein Loss with Gradient Penalty:
This loss function was immediately popularized after getting published because of mathematically explainable loss convergence and emperically better results. We introduce a parametrized function as a critic which is looking at the maximum distance between the real and the fake image in a latent space. This loss is known as the earth mover's distance. This parametrized function needs to satisfy 1-Liptschitz to ensure convergence which basically means |fw(x1) - fw(x2)| <= x1 - x2 for all x1, x2.

- Discriminator Loss: $\max E_{x\sim P_r}[f_w(x)] - E_{z\sim p(z)}[f_w(g_\theta(z))]$
- Generator Loss: $\min E_{x\sim P_r}[f_w(x)] - E_{z\sim p(z)}[f_w(g_\theta(z))]$

The idea of **gradient-penalty** is to enforce a constraint with penalty on the gradient norm such that the gradients of the critic's output with respect to the input have a unit norm. Instead of using a hard constraint which limits learning we introduce this as a flexible soft constraing which can be learned by the model.



Different neural network architectures used in GAN:

General Network of GAN

Vanilla GAN:
- It is the simplest GAN architecture where we use simple artificial neural networks layers to train the generator and discriminator. Mostly **NS-Loss** is used with Vanilla GAN but **WGAN-GP** can also be used for better results.

Deep Convolutional GAN:
- This architecture uses convolutional and transpose convolutional layers in the discriminator and generator respectively to generate higher quality images than **Vanilla GAN**.

Progressively Growing GAN(proGAN):
proGAN is one of the best GAN architecture available which improves on previously mentioned architectures in terms of quality, speed, and diversity in image generation. The key idea of progressively growing GAN is to first train a model on a small dataset where the discriminator is the mirror of the generator network and more convolutional layers are added in the network after the model has gained some decent results for the previous resolution. When the resolution is doubled after training of the lower resolution a transition layer with a parameter alpha is introduced to slowly transition from current resolution to next resolution.
This method seems more natural as humans also learn in a progressively growing manner. The research paper on proGAN described three more important ideas for implementation of proGAN ensuring training convergence and stability:

1) **Mini-batch standard deviation-** Variance of a generated batch of images is calculated and added as a layer towards the end in the discriminator term allowing more dirverse examples to be produced.
2) **Pixel Normalization-** All batch normalization layers were replaced by a pixel normalization technique which doesn't have any learnable parameter giving more stable results.
3) **Equalized Learning Rate:** Before every forward pass learning rates are equalized by scaling the weights. Ensures that the learning rate doesn't depend on the dynamic range of that feature.
Progam architecture uses **WGAN-GP loss** for optimization.

Losses for GAN:

1) MM Loss:

Min-Max is very common loss function for adversarial environments and here the generators loss is negative of discriminators loss. Here, discriminator is trying to minimize it's loss and at the same time generator is trying to minimize the negative of that loss. This is rarely used in practice and it can blow up training due to competing weight changes among discriminator and generator and hence they can get stuck. Here, Jd describes the maximum likelihood loss of the discriminator and Jg is the loss of the generator.

$$J^D = E_{x \sim p_r} \log[D(x)] + E_{z \sim p_g} \log[1 - D(G(z))]$$

$$J^G = -J^D$$

2) Non Saturation Loss:

A slight modification is made to the min-max loss where instead of using the negative of discriminator loss as generator loss just the false positives part of the loss is analyzes and the generator tries to maximize the false positives of the generator. This is the most commonly used loss function for training GANs.

$$J^D = E_{x \sim p_r} \log[D(x)] + E_{z \sim p_g} \log[1 - D(G(z))]$$

$$J^G = E_{z \sim p_g} \log[D(G(z))]$$

3) Wasserstein Loss with Gradient Penalty:

This loss function was immediately popularized after getting published because of mathematically explainable loss convergence and emperically better results. We introduce a parametrized function as a critic which is looking at the maximum distance between the real and the fake image in a latent space. This loss is known as the earth mover's distance. This parametrized function needs to satisfy 1-Liptschitz to ensure convergence which basically means |fw(x1) - fw(x2)| <= x1 - x2 for all x1, x2.

- Discriminator Loss: $\max E_{x \sim P_r}[f_w(x)] - E_{z \sim p(z)}[f_w(g_\theta(z))]$

- Generator Loss: $\min E_{x \sim P_r}[f_w(x)] - E_{z \sim p(z)}[f_w(g_\theta(z))]$

The idea of **gradient-penalty** is to enforce a constraint with penalty on the gradient norm such that the gradients of the critic's output with respect to the input have a unit norm. Instead of using a hard constraint which limits learning we introduce this as a flexible soft constraing which can be learned by the model.

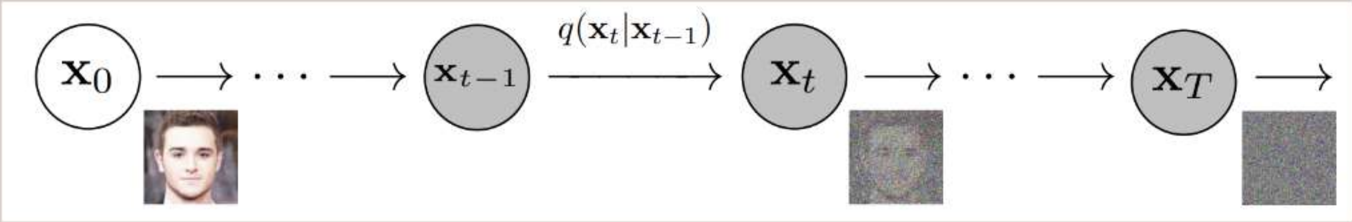
Techniques used for image generation:

Diffusion Models:

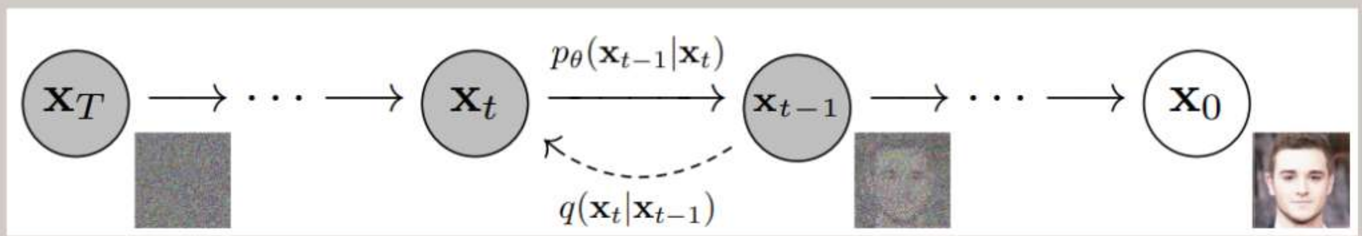
Diffusion Models are generative models which have been gaining significant popularity in the past several years, and for good reason. A handful of seminal papers released in the 2020s alone have shown the world what Diffusion models are capable of, such as beating GANs on image synthesis.

- Diffusion Models work by destroying training data through the successive addition of Gaussian noise, and then learning to recover the data by reversing this noising process. After training, we can use the Diffusion Model to generate data by simply passing randomly sampled noise through the learned denoising process.

- These models can produce high quality images with very high diversity the only problem with such models is that they are extremely slow to train and generate images.



- Forward Process-> Adding Gaussian Noise



- Backward Process-> Predicting image from noise through a fixed markov chain

Final Remarks VAE v/s GANs v/s Diffusion Models

Paradigm	Quality	Diversity	Speed
VAE	✗	✓	✓
GAN	✓	✗	✓
Diffusion	✓	✓	✗

- For these reasons we will be majorly focusing on GANs for our application. We will also be experimenting with diffusion models as they have shown to beat some GAN models in terms of quality of image synthesis. The model we have selected to investigate and fine tune for our application is proGAN with some ideas inspired from the recent papers of styleGAN generated by nvidia.

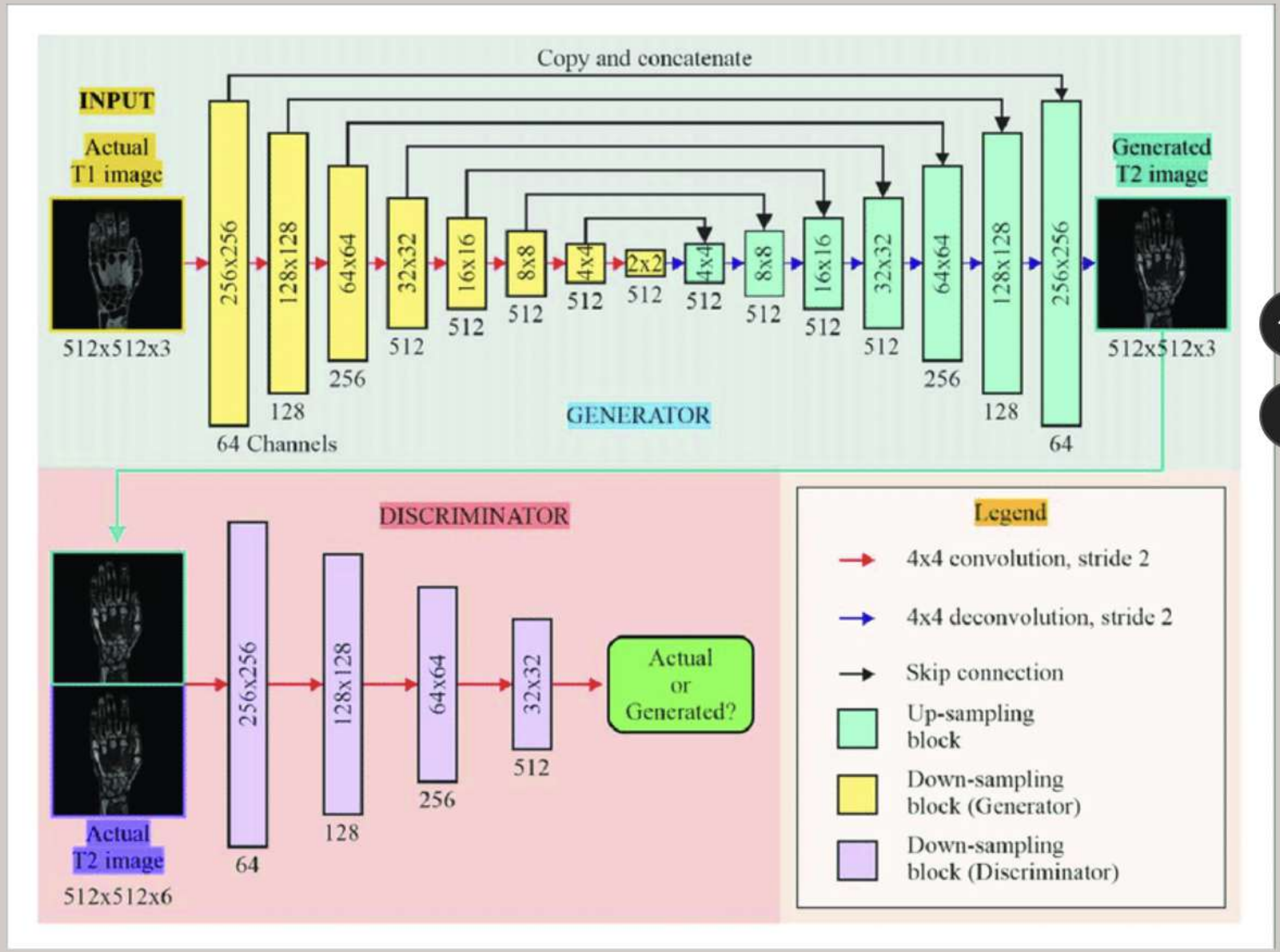
The goal of this project is to make a gui based application which can help develop datasets for training different models making them more robust to unseen scenarios. We are also investigating image translation techniques such as cycleGAN and pix2pix for conversion of one style of images to another for example. Day -> Night, Summer -> Winter, Aerial -> Map etc.
These models can significantly improve the models implementing on military technologies and hence make them more reliable in unseen environments.

Techniques used for image translation:

pix2pix

Pix2Pix GAN has a generator and a discriminator just like a normal GAN would have. The generator takes in the input image as input and learns to map this image to target image. In Pix2Pix, the generator is a convolutional network with U-net architecture.

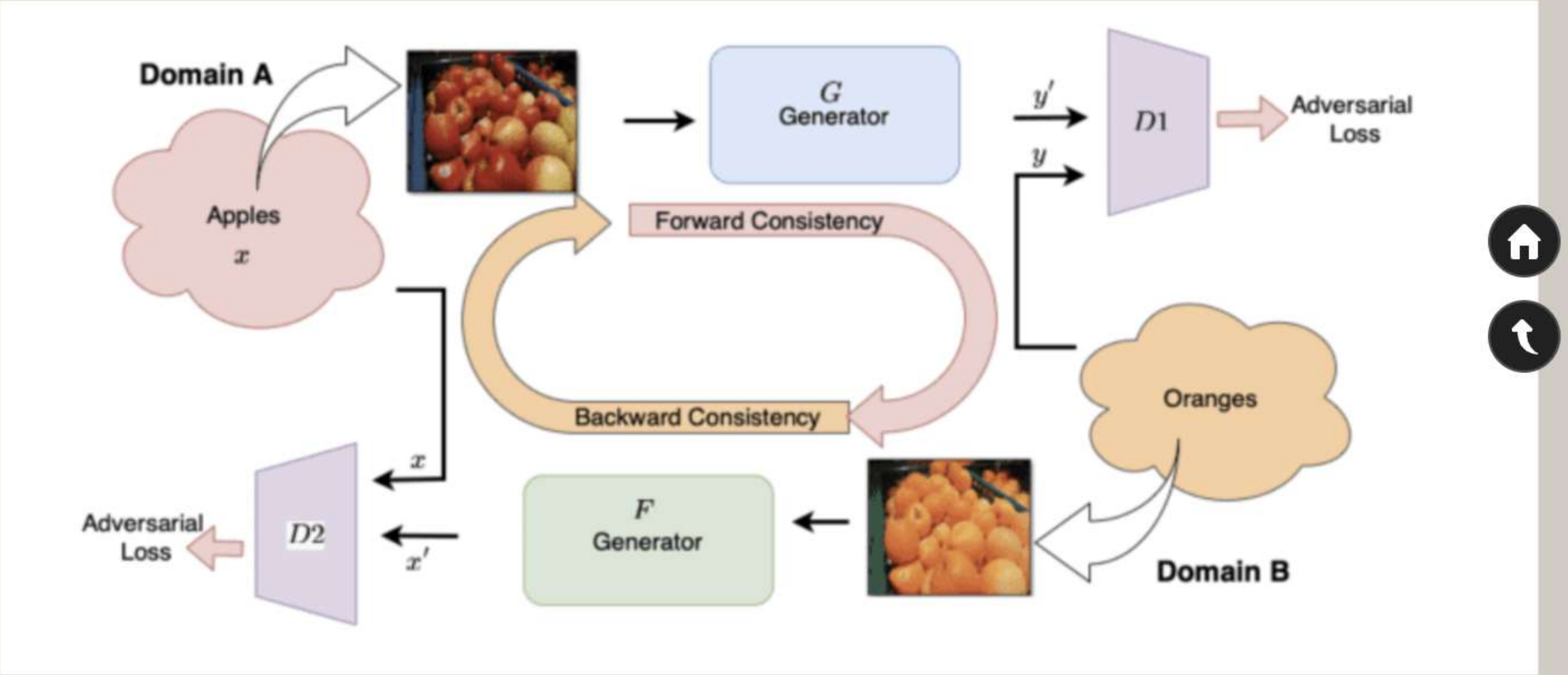
The discriminator of pix2pix network is just a couple of convolutional blocks. The discriminator rather than outputting a single integer between 1 and 0 representing the probability of image being real, outputs a matrix of such 1s and 0s for different patches of the image known as patchGAN discriminator.



Techniques used for image translation:

cycleGAN

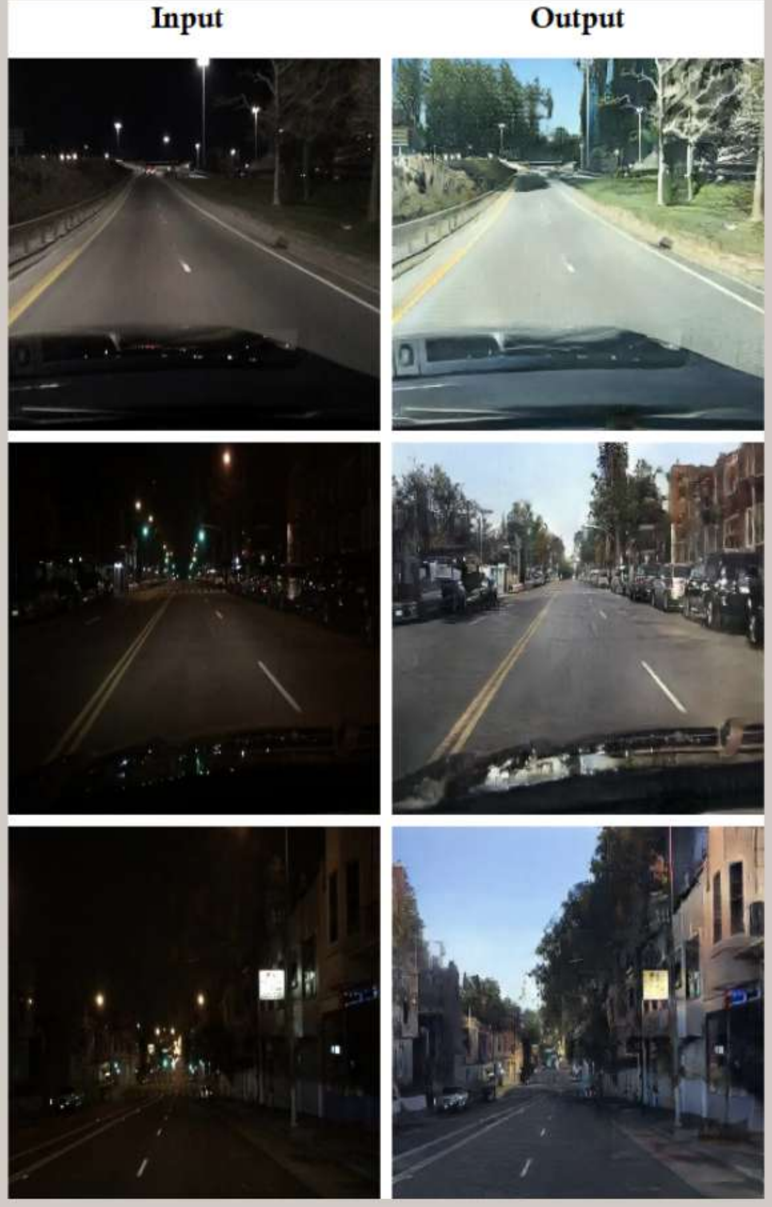
The only advantage of cycleGAN over pix2pix is that it can learn to map input image to target image for unpaired set of images. Therefore it reduced the manual time of pairing images with their exact desired output which is sometimes even not possible for some applications.



cycleGAN uses 2 generators and 2 discriminators to learn mapping from both input image to target image and target image to input image. Various losses such as cycle consistency loss, identity loss and MSE loss are used for training a cycleGAN

Some Applications of Generative AI

- Image Generation
- Semantic Image to Photo generation
- Super Resolution Image
- 3D Shape Generation
- Text-to-Speech Generation
- Text Generation, Code Generation
- Music, Video Generation
- DeepFake
- Image-to-Image Translation



Night-to-Day Conversion

Generative AI in real life

Deep Fakes

3D- GAN

Image Translation

Generative AI is used for a variety of applications by the industry leaders today itself. Here, we will be discussing image domain specific use cases which can be applied in military applications. However, we can utilize this technology for various other applications too:

- **Data Augmentation** for training another model on a more versatile dataset for making more robust models capable of generalizing to new cases even better.
- **Image to Image translation** for converting one domain of images to another for example: converting a dataset of day images to night images for training a model making it more adaptable to lighting conditions.
- **Improving the resolution** of a low-res image to make it more clear for human operators. Super Resolution GAN's can allow us to train a network which can improve the resolution of a low quality image.

Potential Use Cases

Generative AI in real life

Deep Fakes

3D- GAN

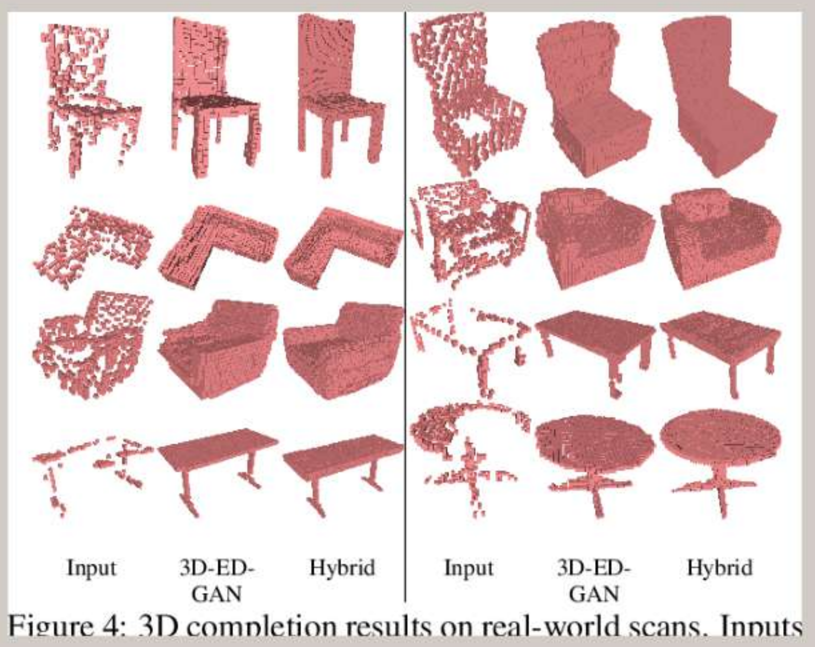


Image Translation

Potential Use Cases

Generative AI is used for a variety of applications by the industry leaders today itself. Here, we will be discussing image domain specific use cases which can be applied in military applications. However, we can utilize this technology for various other applications too:

- **Data Augmentation** for training another model on a more versatile dataset for making more robust models capable of generalizing to new cases even better.
- **Image to Image translation** for converting one domain of images to another for example: converting a dataset of day images to night images for training a model making it more adaptable to lighting conditions.
- **Improving the resolution** of a low-res image to make it more clear for human operators. Super Resolution GAN's can allow us to train a network which can improve the resolution of a low quality image.