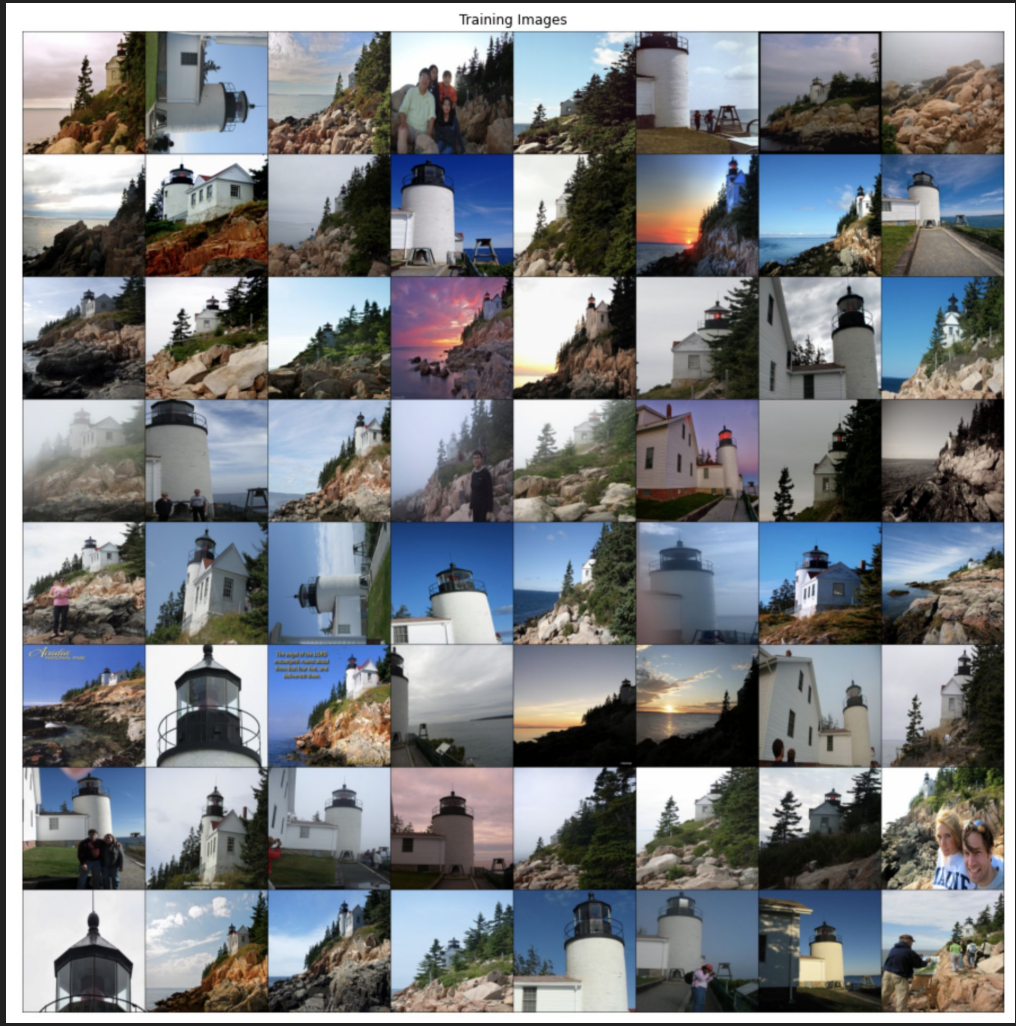
# Generative Adversarial Networks (GANs)

## Generating Synthetic Lighthouses

# DATASET

Dataset: Google Landmarks

~1200 photos of lighthouses



# TASK Generate Synthetic Images of Lighthouses

## Generator

## Discriminator






# DCGAN vs Vanilla GAN

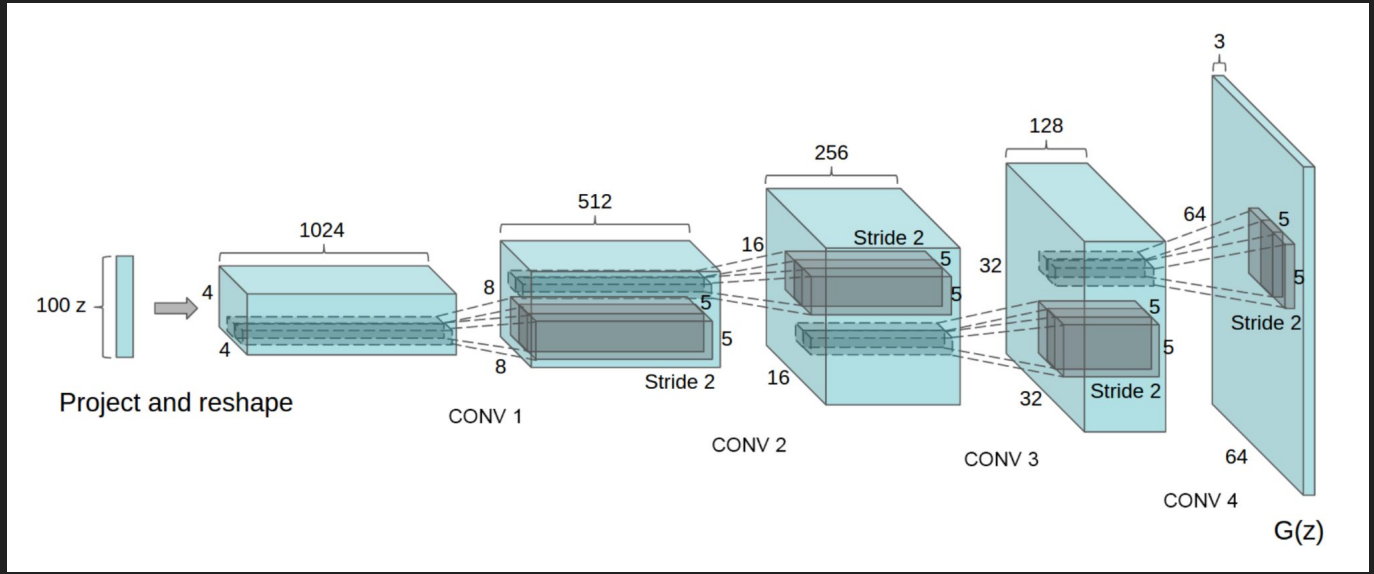
**Vanilla GAN Generator**

```
class VanillaGAN_Generator(nn.Module):
    def __init__(self, z_dim, img_dim):
        super().__init__()
        self.gen = nn.Sequential(
            nn.Linear(z_dim, 256),
            nn.LeakyReLU(0.01),
            nn.Linear(256, img_dim),
            nn.Tanh(),
        )

    def forward(self, x):
        return self.gen(x)
```

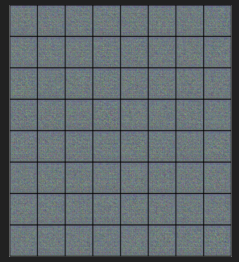
# DCGAN vs Vanilla GAN

**DCGAN Generator**

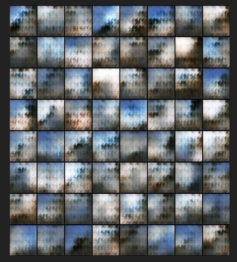


# Results

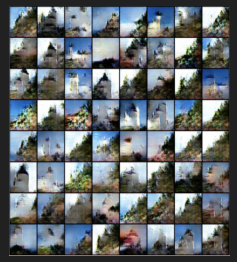
Epoch 5



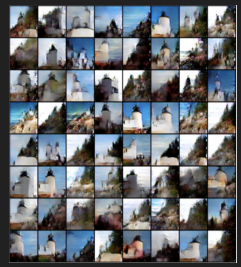
Epoch 55



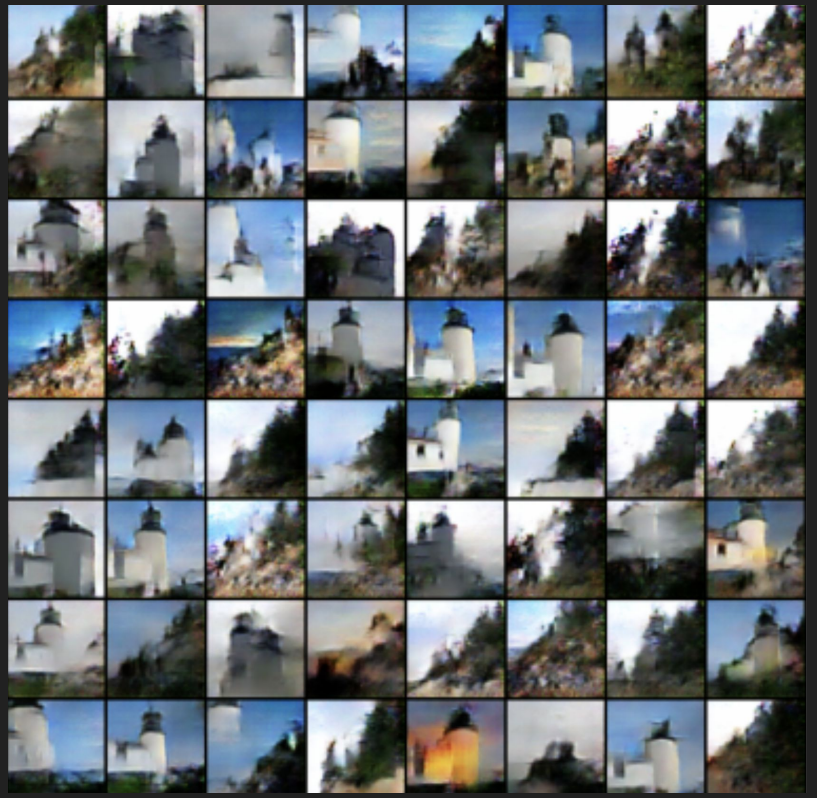
Epoch 155



Epoch 255



Epoch 500



# Process

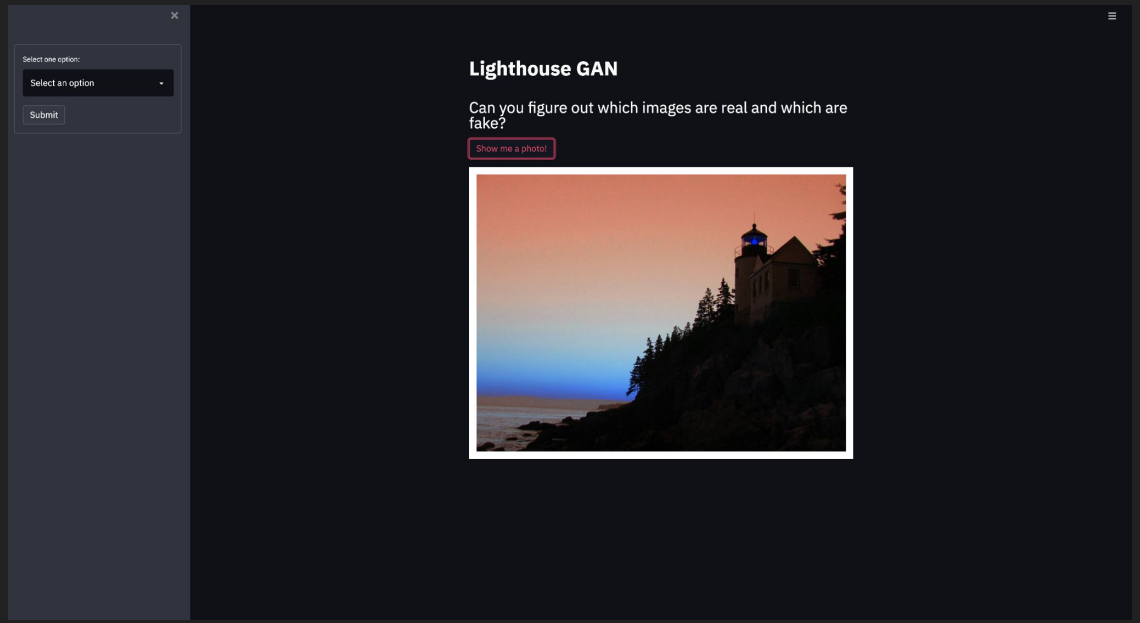
## What went wrong?

1. Trying to place more weight on generator
2. Trying to increase the learning rate of generator over the discriminator
3. Focusing too much on the loss

## What went right?

Not giving too much emphasis on the loss and making the learning rate and weights equal for the discriminator and generator

# Deploy


Select one option:
Select an option
Submit
Lighthouse GAN
Can you figure out which images are real and which are fake?
Show me a photo!

# Future Steps

1. Train with more data
2. Train with the entire landmark dataset
3. Explore more GAN architectures
4. Add features to the prediction for streamlit so users can choose characteristics of their lighthouse photograph i.e time of day, background landscape
5. ***Measure  quantitative metrics**

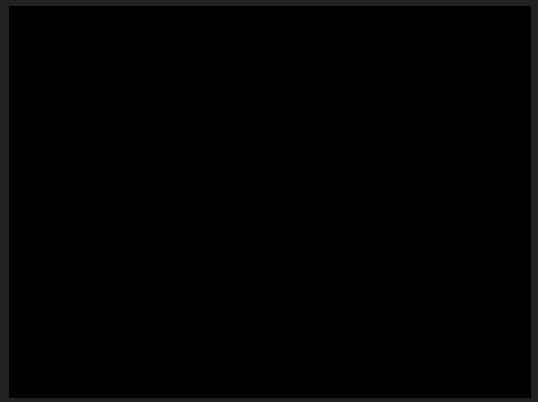
# Quantitative Metrics

1. **Peak Signal-to-Noise Ratio (PSNR)**
2. **Structural Similarity Index Measurement (SSIM)**

$$PSNR(\hat{y}, y) = 10 \log_{10}(\frac{\max(y)}{RMSE(\hat{y}, y)})$$

Where:

$$RMSE(\hat{y}, y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$



$$SSIM(\hat{y}, y) = \frac{(2\mu_{\hat{y}}\mu_y + C_1) + (2\sigma_{\hat{y}y} + C_2)}{(\mu_{\hat{y}}^2 + \mu_y^2 + C_1)(\sigma_{\hat{y}}^2 + \sigma_y^2 + C_2)}$$

# Thank you for listening!