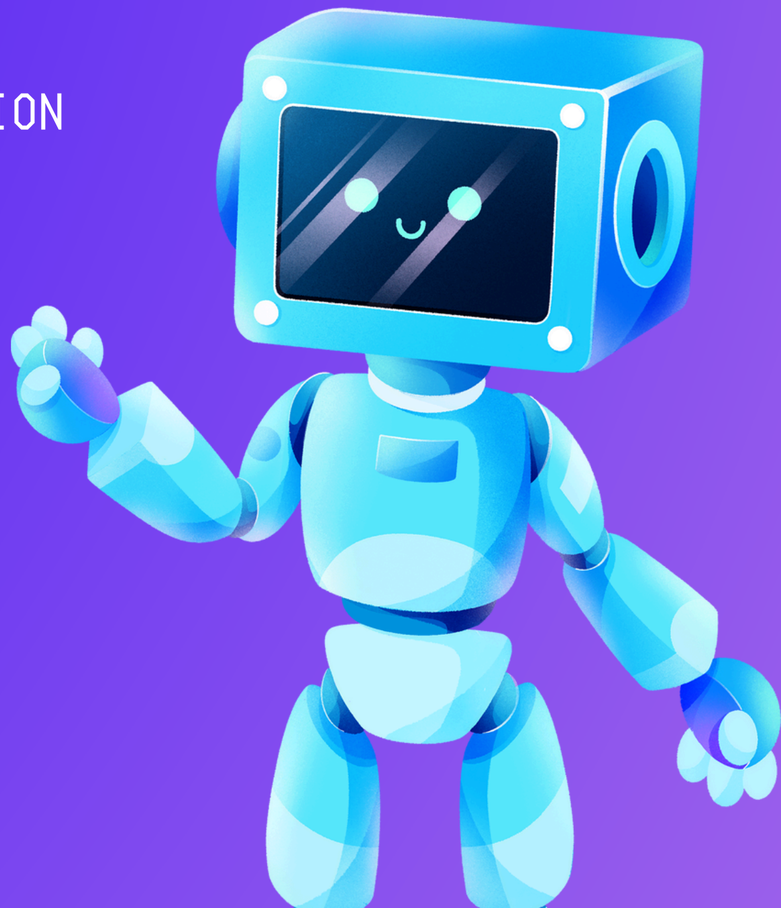
GENERATIVE ADVERSARIAL NETWORKS VS. STABLE DIFFUSION:

A COMPARATIVE ANALYSIS FOR IMAGE GENERATION

By Sage Woodard



UNDERSTANDING GENERATIVE AI

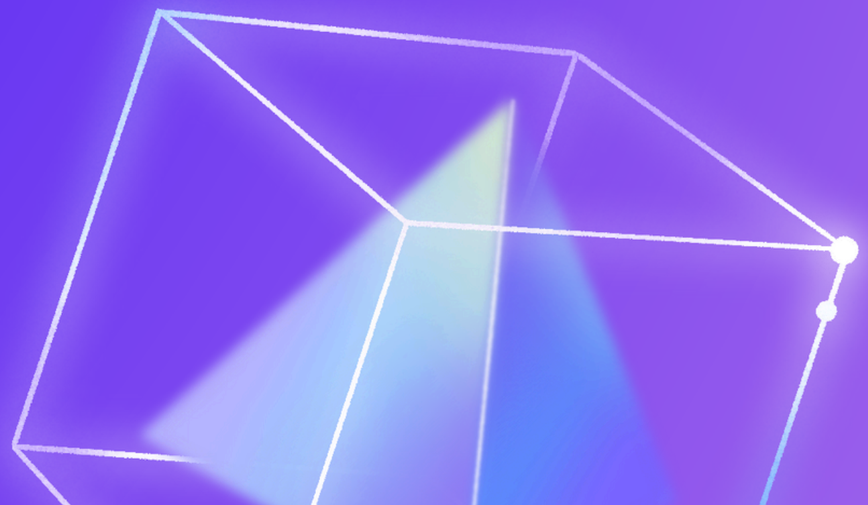
- What is Generative AI?

  Generative AI leverages advanced neural networks to create new data, such as images, videos, or music, that closely resembles real-world examples.

- Why it Matters:

  These models enable innovations in etertainment, science, and virtual reality by automating processes and solving complex problems.

- Prominent Models:
  - Generative Aversarial Networks (GANs)
  - Stable Diffusion Models



REFERENCES: [5], [7]

GENERATIVE ADVERSARIAL NETWORKS

- GANs (Generative Adversarial Networks) are a type of deep learning model designed to generate realistic synthetic data, such as images, by learning the distribution of real data.
- They consist of two components: a Generator that creates synthetic data and a Discriminator that evaluates whether the data is real or fake.
- The model is trained through an adversarial process, where the Generator improves by creating convincing outputs, and the Discriminator enhances its ability to differentiate between real and generated data.
- GANs have applications in image generation, data augmentation, style transfer, and video synthesis.
- They are widely used in creative industries, gaming, and fields requiring high-quality, realistic synthetic data.

REFERENCES: [5], [8]

STABLE DIFFUSION

- Stable Diffusion is a deep learning model designed for generating high-quality images from text prompts or transforming existing images.
- It uses a diffusion process, iteratively denoising a random noise image guided by the input prompt to produce coherent outputs.
- Unlike pixel-based methods, it operates in a latent space (compressed image representation), making it computationally efficient for high-resolution images.
- Applications include text-to-image generation, inpainting, and image-to-image translation.
- It enables creative and practical uses across industries, from art creation to scientific visualization.

SOURCE: HTTPS://WWW.TENSORFLOW.ORG/TUTORIALS/GENERATIVE/GENERATE_IMAGES_WITH_STABLE_DIFFUSION

METHODOLOGY

01

Programming Language:
Python

02

Deep Learning Framework:
PyTorch and Tensorflow

03

Dataset:
EuroSAT satellite imagery

04

Implementation:
GAN and Stable Diffusion

PYTORCH AND TENSORFLOW

- PyTorch: An open-source deep learning framework that provides flexibility and a dynamic computation graph, making it ideal for research and experimentation. PyTorch is intuitive, uses Pythonic syntax, and offers strong GPR support and integration with tools like Hugging Face for state-of-the-art AI models.
- TensorFlow: An open-source machine learning framework that excels in production-level deployment with static computation graphs for efficiency and scalability. TensorFlow offers a wide range of applications, strong support for distributed computing, and a vast ecosystem.

SOURCES: PASZKE, ADAM, ET AL. (2019) AND ABADI, MARTIN, ET AL. (2016)

EUROSAT DATASET

- Overview: The EuroSAT dataset is derived from Sentinel-2 satellite imager, containing 27,000 labeled and georeferenced samples across 10 land use and land cover classes (e.g., agricultural land, water bodies, and urban areas).
- Key Features: The EuroSAT dataset covers 13 spectral bands, including RGB, making it versatile for various applications and is preprocessed for immediate use, reducing computational overhead for model training.
- Limitations: While EuroSAT offers moderate resolution, higher-resolution datasets exist for tasts requiring finer detail.

SOURCE: HELBER, PATRICK, ET AL. (2019)

EUROSAT SAMPLE VIA TENSORFLOW

GAN IMPLEMENTATION

```python
# Define the generator model
def build_generator():
  model = tf.keras.Sequential()
  model.add(layers.Dense(8 * 8 * 256, use_bias=False, input_shape=(100,)))
  model.add(layers.BatchNormalization())
  model.add(layers.LeakyReLU())

  model.add(layers.Reshape((8, 8, 256)))
  model.add(layers.Conv2DTranspose(128, (5, 5), strides=(2, 2), padding='same', use_bias=False))
  model.add(layers.BatchNormalization())
  model.add(layers.LeakyReLU())

  model.add(layers.Conv2DTranspose(64, (5, 5), strides=(2, 2), padding='same', use_bias=False))
  model.add(layers.BatchNormalization())
  model.add(layers.LeakyReLU())

  model.add(layers.Conv2DTranspose(3, (5, 5), strides=(2, 2), padding='same', use_bias=False, activation='tanh'))
  return model
```

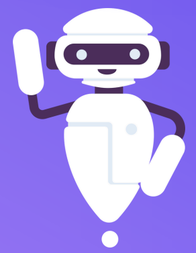
GITHUB: HTTPS://GITHUB.COM/SAGEWOODARD/EUROSAT_GAN/BLOB/MAIN/GAN_11_2024.IPYNB

STABLE DIFFUSION IMPLEMENTATION

```
# Load and configure the stable diffusion model
# Initialize the stable diffusion image-to-image pipeline
from diffusers import StableDiffusionImg2ImgPipeline
import torch

# Load stable diffusion image-to-image pipeline
pipe = StableDiffusionImg2ImgPipeline.from_pretrained('CompVis/stable-diffusion-v1-4', torch_dtype=torch.float16)

# Set the model to use the GPU
pipe = pipe.to('cuda') # This moves the model to the A100 GPU in Google Colab
print('Using GPU:', torch.cuda.get_device_name(0))

# Optionally disable the safety checker
pipe.safety_checker = None # Disable safety checker
```

GITHUB: HTTPS://GITHUB.COM/SAGEWOODARD/EUROSAT_STABLEDIFFUSION/BLOB/MAIN/SDM_11_2024.IPYNB

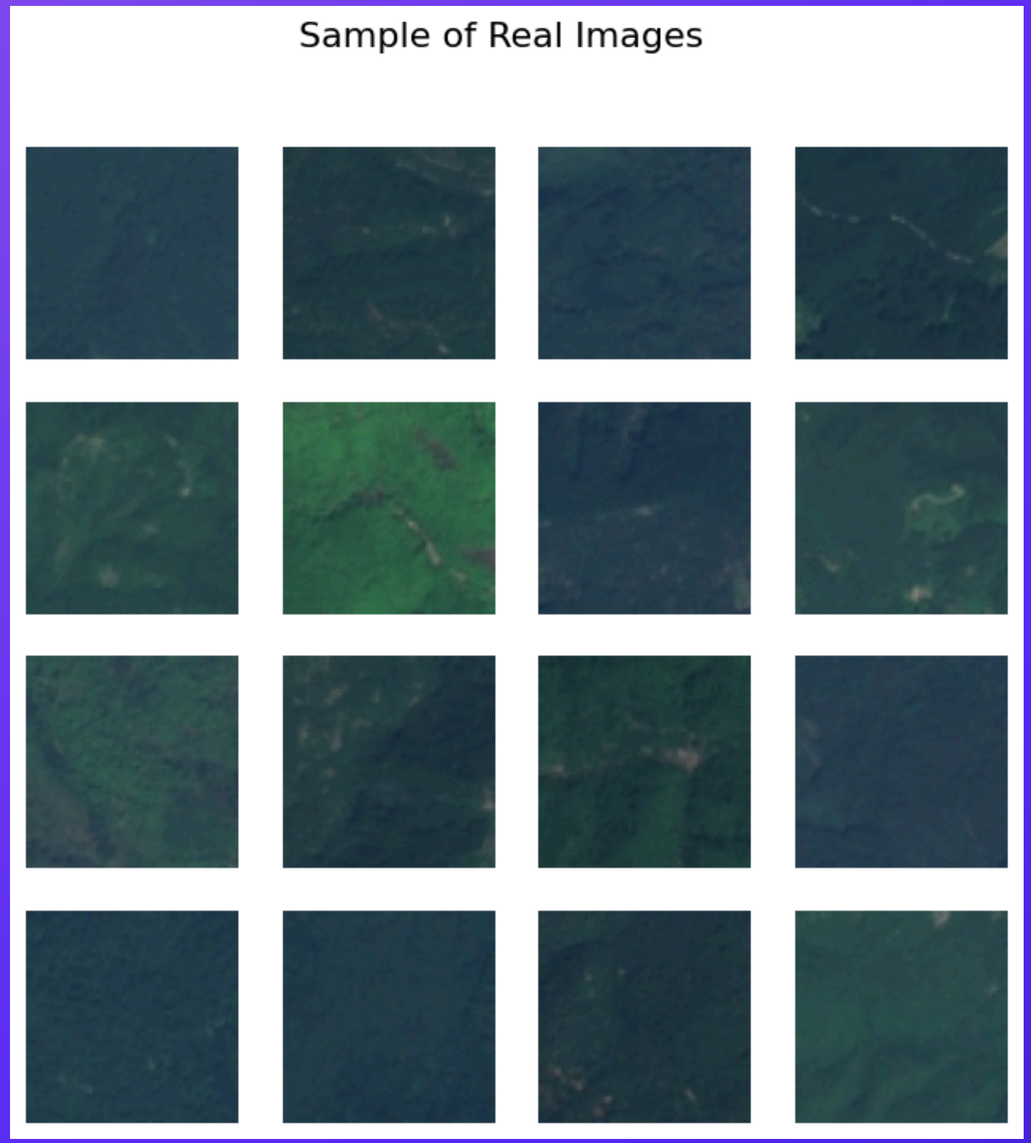
EVALUATION CRITERIA



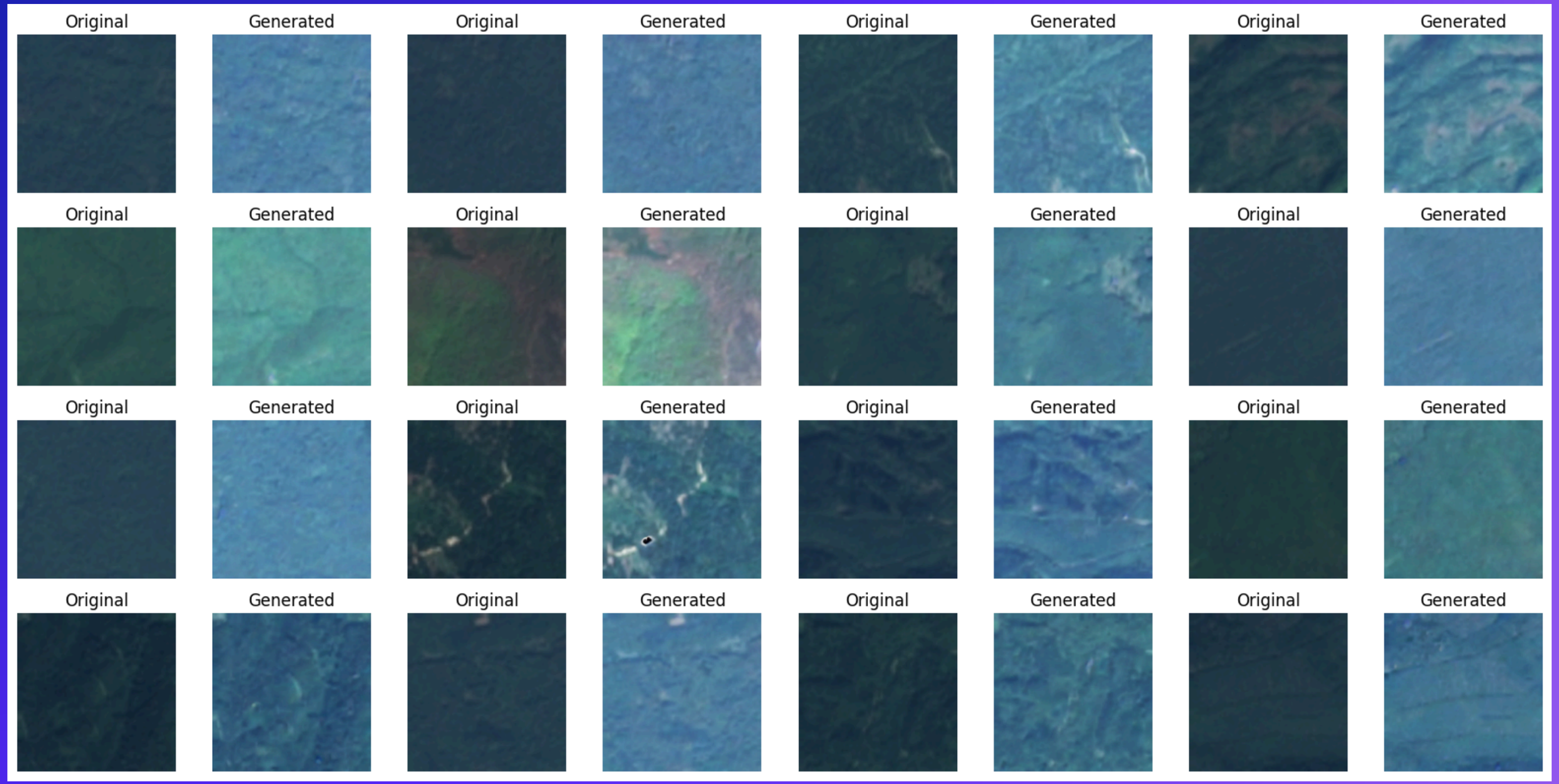
VISUAL INSPECTION

LOSS METRIC CALCULATIONS

FRECHET INCEPTION DISANCE



GAN VISUALIZATION

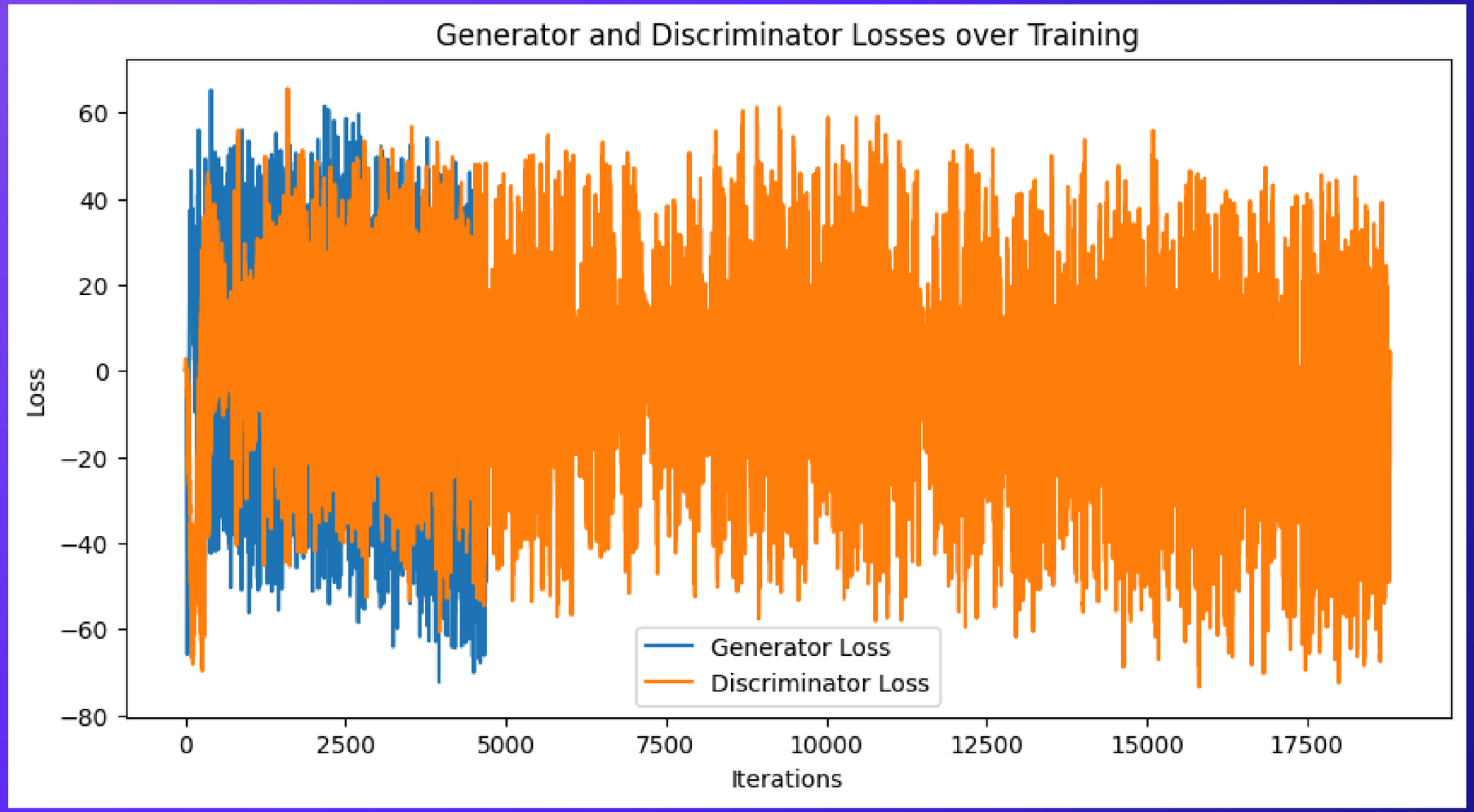


STABLE DIFFUSION VISUALIZATION

LOSS METRIC CALCULATIONS

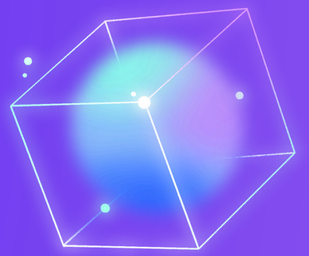
- Wasserstein loss was used as the loss metric for the GAN model, as the model was a WGAN model specifically. For the generator, using Wasserstein loss encourages the generator to produce realistic images by maximizing the discriminator's evaluation. For the discriminator, Wasserstein loss with a gradient penalty (WGAN-GP) ensures continuity and helps stabilize training.
- Because the stable diffusion model was designed using the Hugging Face implementation, the denoising loss is hidden behind predefined pipelines.

SOURCES: ARJOVSKY, MARTIN, ET AL (2017).
ROMBACH, ROBIN, ANDREAS BLATTMANN, DOMINIK LORENZ, PATRICK ESSER, AND BJÖRN OMMER (2022).



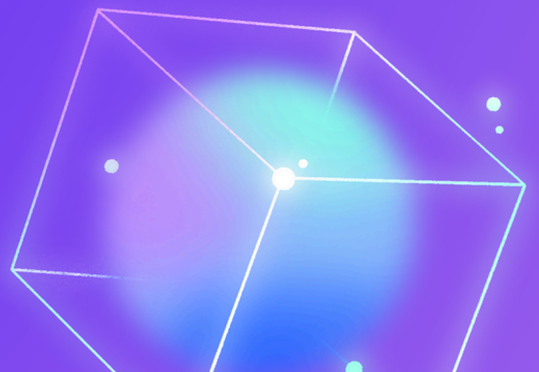
GAN LOSS GRAPH

FRECHET INCEPTION DISTANCE



- The Fréchet Inception Distance (FID) is a metric used to evaluate the quality of images generated by generative models like GANs and SDMs. It measures the similarity between the distribution of real images and generated images by comparing their feature representations.
- FID is widely used because it aligns well with human perception of image quality.
- A lower FID score means higher image quality and closer alignment with real-world data.

SOURCE: HEUSEL, M., RAMSAUER, H., UNTERTHINER, T., NESSLER, B., & HOCHREITER, S. (2017).



FID RESULTS

- GAN: 1846.37
- Stable diffusion: 165.74
- s-value (Difference): 1680.63
- s-value (ratio) 11.14
- There is a large difference between the gan fid and the stable diffusion fid (1680.63, specifically). The gan fid is 11 times higher than the stable diffusion fid.

```python
import matplotlib.pyplot as plt

# Example FID scores
fid_gan = 1846.365750809347
fid_sd = 165.73935229494703

# Compute s-values
s_difference = fid_gan - fid_sd
s_ratio = fid_gan / fid_sd

# Print Results
print(f"FID (GAN): {fid_gan}")
print(f"FID (Stable Diffusion): {fid_sd}")
print(f"s-value (Difference): {s_difference}")
print(f"s-value (Ratio): {s_ratio}")

# Plot Results
models = ['GAN', 'Stable Diffusion']
fid_scores = [fid_gan, fid_sd]

plt.bar(models, fid_scores, color=['blue', 'green'])
plt.ylabel('FID Score')
plt.title('FID Scores Comparison')
plt.show()
```

```
FID (GAN): 1846.365750809347
FID (Stable Diffusion): 165.73935229494703
s-value (Difference): 1680.6263985144
s-value (Ratio): 11.140177183289488
```

KEY OBSERVATIONS

- When visually comparing the images generated by the GAN model to the images generated by the stable diffusion model, it is clear that the images produced by the stable diffusion model include much more detail and are closer in appearance to the images from the EuroSAT datset.
- The conclusion drawn from the visual comparison of the images is reinforced by the results of the FID calculations and resulting s-score, as the GAN FID is much higher than the stable diffusion FID.
- These observations suggest that the stable diffusion model is better suited than the GAN model for generating images from the EuroSAT dataset.

KEY FINDINGS AND CONSIDERATIONS

- Due to the difficulty of balancing the generator and discriminator for GAN models to be efficient and work well, stable diffusion may be the better option for generating terrain images from satellite data
- PyTorch is better suited for this project than Tensorflow. The stable diffusion implementation utilized PyTorch, whereas the GAN implementation utilized Tensorflow.
- Regardless of the model being used, the image generation could benefit from a higher resolution dataset and a more complex model architecture
- A more advanced model will likely be needed for future projects

FUTURE DIRECTIONS AND APPLICATIONS

- Advancing Terrain Generation:
  - Consider a hybrid approach combining GANs and Stable Diffusion Models with high-resolution satellite data (e.g., Google Earth Engine) for realistic and dynamic terrains.
  - Focus on video game and VR environment applications, enhancing fidelity and interactivity.
- Future Plans:
  - Utilize TorchGeo to integrate geospatial data (elevation, land cover, slope).
  - Explore 3D reconstruction with deep learning for dynamic terrain models.
  - Apply findings to immersive gaming and environmental monitoring.

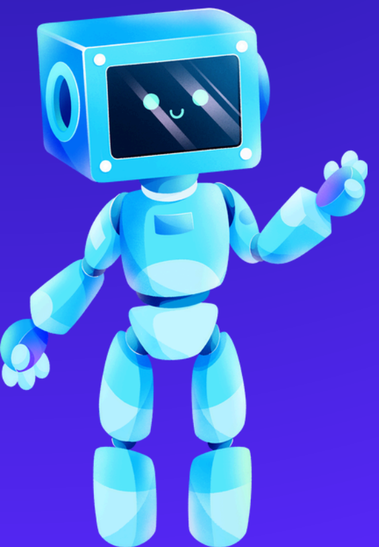
MODEL CODE

- GAN:

  HTTPS://GITHUB.COM/SAGEWOODARD/EUROSAT_GAN/BLOB/MAIN/GAN_11_2024.IPYNB

- STABLE DIFFUSION:

  HTTPS://GITHUB.COM/SAGEWOODARD/EUROSAT_STABLEDIFFUSION/BLOB/MAIN/SDM_11_2024.IPYNB



REFERENCES

[1] M. ABADI, ET AL., "TENSORFLOW: LARGE-SCALE MACHINE LEARNING ON HETEROGENEOUS SYSTEMS," SOFTWARE AVAILABLE FROM TENSORFLOW.ORG, 2016. AVAILABLE: HTTPS://WWW.TENSORFLOW.ORG/.

[2] M. ARJOVSKY, ET AL., "WASSERSTEIN GAN," ARXIV PREPRINT, ARXIV:1701.07875, 2017.

[3] GOOGLE EARTH ENGINE. AVAILABLE: HTTPS://EARTHENGINE.GOOGLE.COM/.

[4] I. GOODFELLOW, ET AL., "GENERATIVE ADVERSARIAL NETWORKS," COMMUNICATIONS OF THE ACM, 2014.

[5] I. GOODFELLOW, ET AL., DEEP LEARNING. MIT PRESS, 2016.

[6] I. GULRAJANI, ET AL., "IMPROVED TRAINING OF WASSERSTEIN GANS," ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, VOL. 30, PP. 5767–5777, 2017.

REFERENCES

[7] P. HELBER, ET AL., "EUROSAT: A NOVEL DATASET AND DEEP LEARNING BENCHMARK FOR LAND USE AND LAND COVER CLASSIFICATION," IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, VOL. 12, NO. 7, PP. 2217–2226, 2019. AVAILABLE: HTTPS://ARXIV.ORG/ABS/1709.00029.

[8] M. HEUSEL, H. RAMSAUER, T. UNTERTHINER, B. NESSLER, AND S. HOCHREITER, "GANS TRAINED BY A TWO TIME-SCALE UPDATE RULE CONVERGE TO A LOCAL NASH EQUILIBRIUM," ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, VOL. 30, PP. 6626–6637, 2017.

[9] A. PASZKE, ET AL., "PYTORCH: AN IMPERATIVE STYLE, HIGH-PERFORMANCE DEEP LEARNING LIBRARY," ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 2019. AVAILABLE: HTTPS://PYTORCH.ORG/.

[10] A. RADFORD, ET AL., "UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS," ARXIV PREPRINT, ARXIV:1511.06434, 2015.

REFERENCES

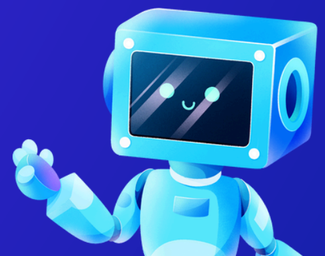
[11] R. ROMBACH, A. BLATTMANN, D. LORENZ, P. ESSER, AND B. OMMER, "HIGH-RESOLUTION IMAGE SYNTHESIS WITH LATENT DIFFUSION MODELS," IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2022. AVAILABLE: HTTPS://ARXIV.ORG/ABS/2112.10752.

[12] M. SEITZER, "PYTORCH-FID: FRÉCHET INCEPTION DISTANCE FOR PYTORCH," GITHUB, 2018. AVAILABLE: HTTPS://GITHUB.COM/MSEITZER/PYTORCH-FID.

[13] TENSORFLOW, "GENERATE IMAGES WITH STABLE DIFFUSION," TENSORFLOW. AVAILABLE: HTTPS://WWW.TENSORFLOW.ORG/TUTORIALS/GENERATIVE/GENERATE_IMAGES_WITH_STABLE_DIFFUSION.

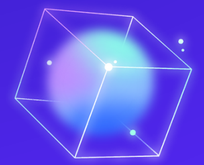
[14] TORCHGEO DOCUMENTATION. AVAILABLE: HTTPS://TORCHGEO.READTHEDOCS.IO/EN/LATEST/.

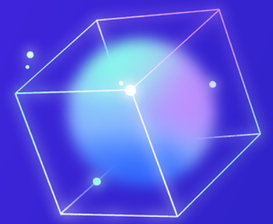
[15] P. VON PLATEN, ET AL., "DIFFUSERS: STATE-OF-THE-ART DIFFUSION MODELS FOR IMAGE GENERATION," HUGGING FACE, 2022. AVAILABLE: HTTPS://HUGGINGFACE.CO/DOCS/DIFFUSERS.

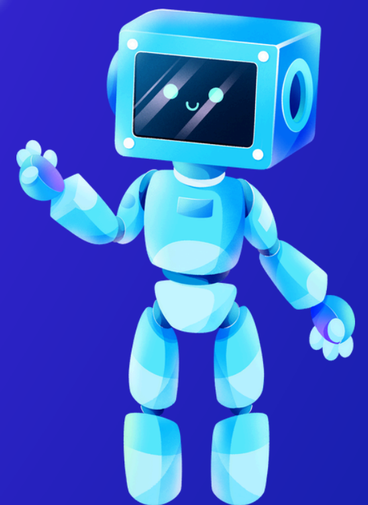


REFERENCES

[16] Z. WANG, ET AL., "IMAGE QUALITY ASSESSMENT: FROM ERROR VISIBILITY TO STRUCTURAL SIMILARITY," IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 13, NO. 4, PP. 600–612, 2004. AVAILABLE: HTTPS://IEEEXPLORE.IEEE.ORG/DOCUMENT/1284395.







THANK YOU!