GENERATIVE AI

DOGS IMAGE GENERATION

~ a project work by GARIMA SOHI

GitHub: https://github.com/sohi-g/image-generation

TABLE OF CONTENTS

PROBLEM STATEMENT

- The objective of this project is to utilize Generative Adversarial Networks (GANs) for generating realistic and diverse images of dogs. This undertaking presents a unique challenge due to the absence of any ground truth data for prediction. The generated images will be evaluated based on their realism and their successful classification as dogs.

- The choice of dogs as the subject of image generation is driven by two factors. First, the inherent appeal of dogs and their universal appeal across audiences. Second, the complexity and diversity associated with dogs, which can be categorized into numerous sub-categories based on breed, colour, size, etc., offer a challenging yet interesting dataset for image generation.

- The project aims to demonstrate the creative capabilities of GANs, showcasing their potential in generating lifelike images and potentially, creating digital worlds. This project lays the foundation of implementing a basic GAN, which will be implemented on MNIST dataset also, to understand it's performance on simpler vs complex images.

ASSUMPTIONS / HYPOTHESIS

1. Data Availability	2. Image Quality	3. Performance of GANs
The project assumes that enough diverse and representative dog images are available for training the GAN model. This includes variety in breed, color, size, pose, and environment.	The quality of the generated images will be high enough to be convincingly classified as dogs. This involves not only the general shape and form of dogs, but also more detailed features such as fur texture and eye color.	It is assumed that GANs will be effective in generating realistic and diverse images of dogs. The success of this project hinges on the effectiveness of GANs in mimicking the intricate features of different dog breeds.

EXPLORATORY DATA ANALYSIS

A. DATA SOURCE OVERVIEW

DATA SOURCE

kaggle

20,655
IMAGE FILES
(.JPG)

120 DOG BREEEDS
(.xml annotation files)

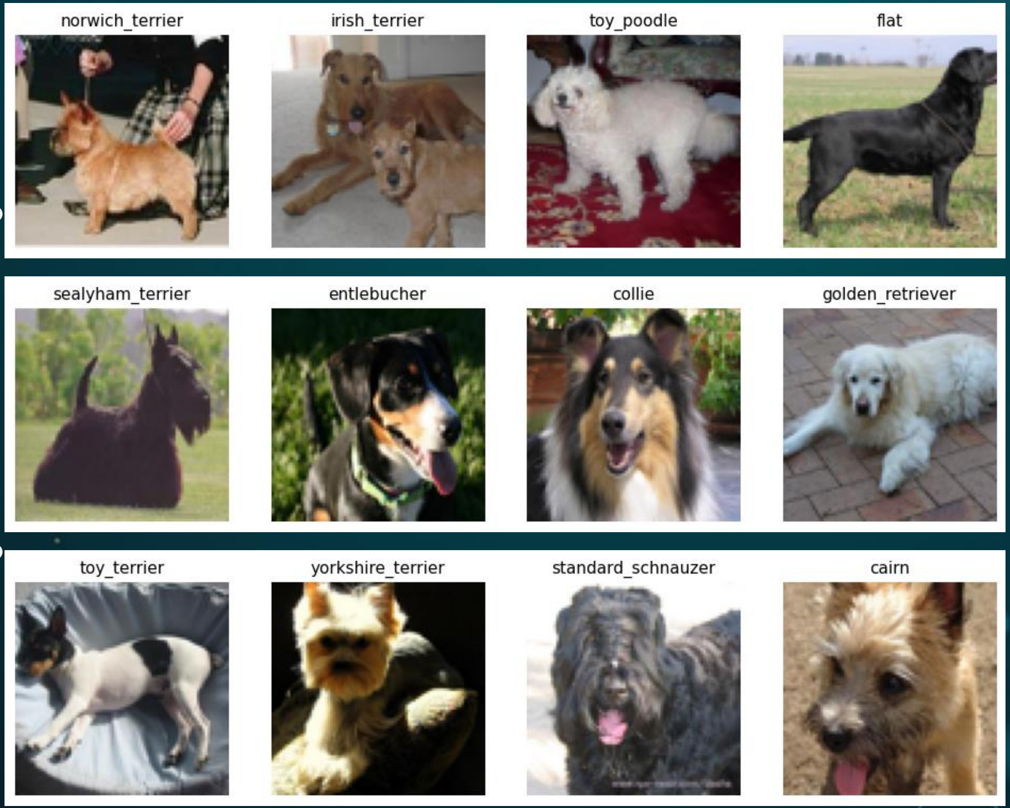
SAMPLE IMAGES

GitHub: https://github.com/sohi-g/image-generation/pre-processing/Pre-Processing.ipynb

EXPLORATORY DATA ANALYSIS

B. PRE-PROCESSING

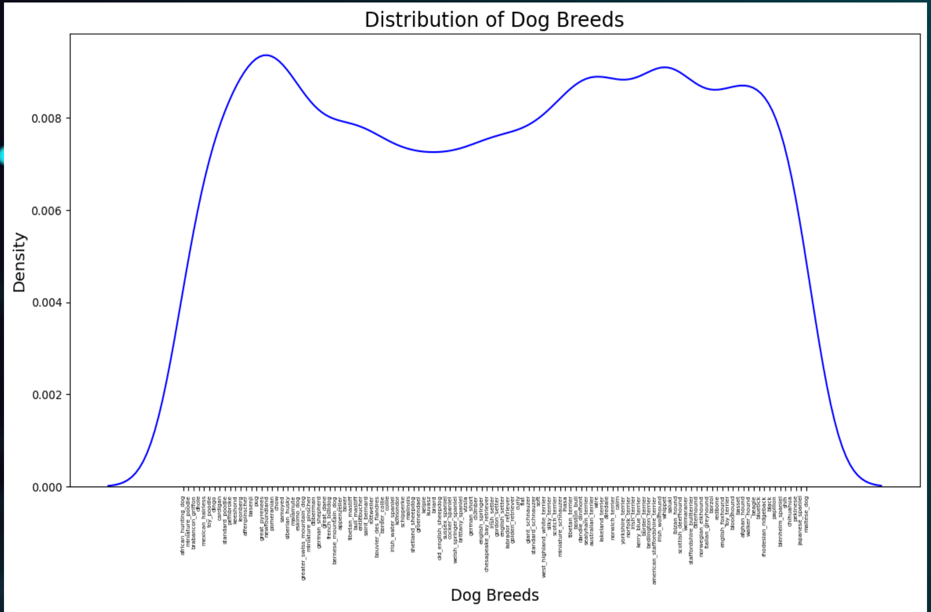
IMAGE RESCALING AND CROPPING BASED ON BOUNDING BOX ANNOTATIONS



Sample Processed Images

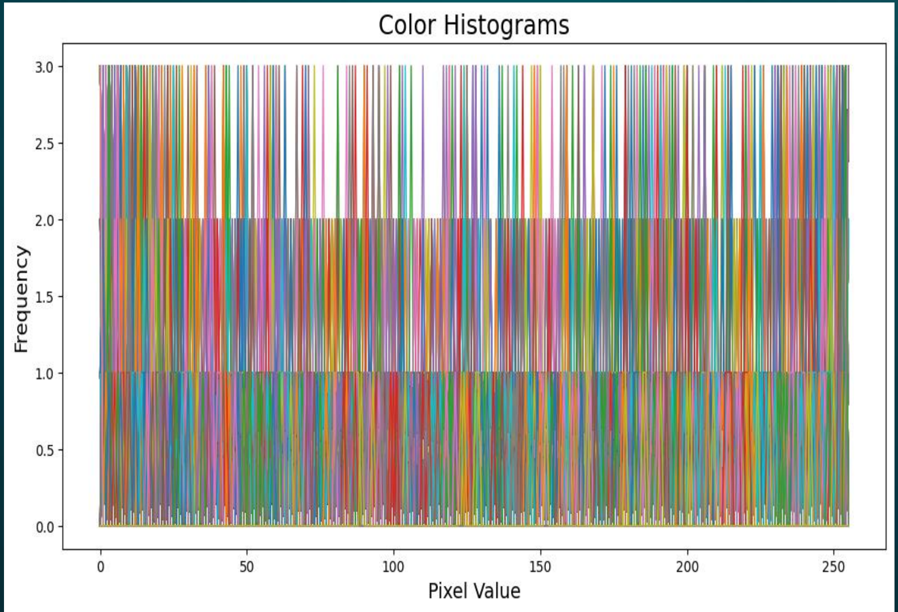
GitHub: https://github.com/sohi-g/image-generation/pre-processing/Pre-Processing.ipynb

EXPLORATORY DATA ANALYSIS
C. INSIGHTS DISCOVERY



Dogs Breed Distribution across 120 types

There is no bias towards in any specific breed type. All breeds have comparable number of images for training. Thus, there is no class imbalance.



Pixel Distribution

Color histograms are plotted to understand if there is any dominant color of breed which might impact the model training, but it turned out all colors are decently spread across in the dataset.

GitHub: https://github.com/sohi-g/image-generation/insights-discovery/Insights-Discovery.ipynb

FEATURE ENGINEERING & TRANSFORMATIONS

Dataset was transformed using following feature engineering and transformation pipeline:

1. **Label Encoding**: Converted categorical labels to numerical labels using a label encoder.

2. **Image Normalization**: Normalized input images by subtracting the mean value (127.5) and dividing by the standard deviation (127.5). This scales the pixel values to the range of -1 to 1.

3. **Data Augmentation**:
- Flip: Randomly flip the input images horizontally to increase dataset diversity.
- Crop Randomly crop the input images to a size of 64x64 pixels, introducing spatial variations.

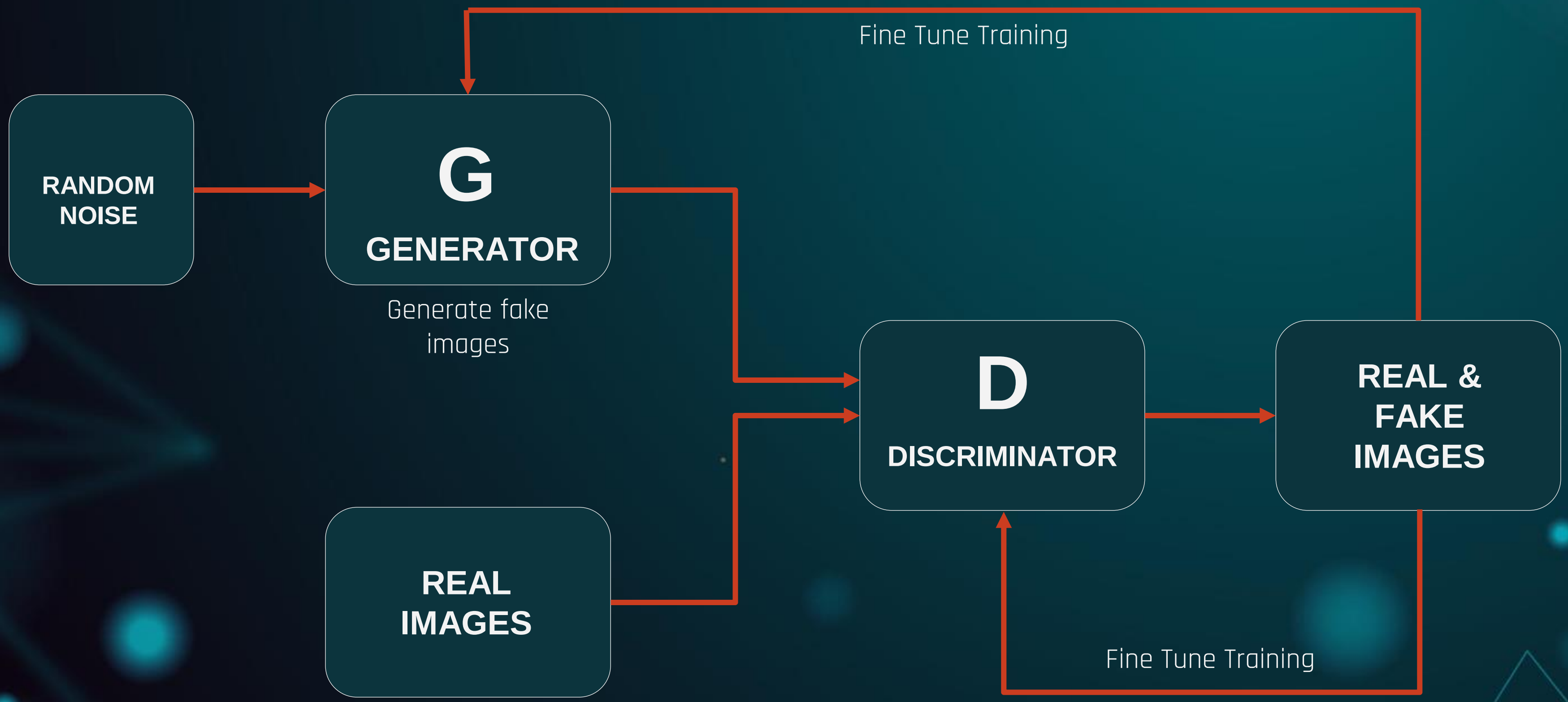
4. **Create TensorFlow Dataset**: Created a TensorFlow dataset from the pre-processed images and labels.

5. **Batching**: Grouped the dataset into smaller batches with a specified batch size. The `drop_remainder=True` argument drops the last incomplete batch if the dataset size is not divisible by the batch size.

GitHub: https://github.com/sohi-g/image-generation/modelling/Modelling-CGAN.ipynb

PROPOSED MODEL

A. MODELLING PIPELINE



PROPOSED MODEL

B. EVALUATION METRIC

In a GAN (Generative Adversarial Network), The generator aims to create realistic synthetic data, while the discriminator tries to tell the difference between real and generated data. During training, the generator and discriminator compete and improve together. They learn from each other's mistakes through an adversarial process.

Evaluation Metric Used:

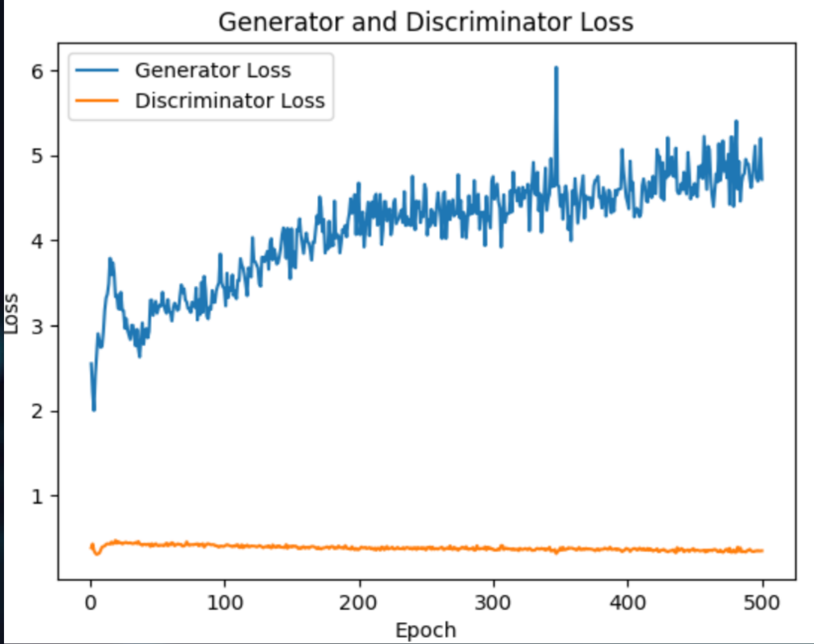
1. Generator Loss: It's calculated by comparing the generated samples with real samples using a specific loss function like mean squared error or binary cross-entropy. The generator learns to improve by minimizing this loss.

2. Discriminator Loss: The discriminator wants to minimize this loss by correctly classifying samples. It's calculated by comparing the discriminator's predictions with the true labels of the samples.

Here, I have used binary_crossentropy for both models (generator & discriminator).

A decreasing generator loss indicates that the generator is getting better at generating realistic samples. We also look at the discriminator loss to ensure a balanced training process where the discriminator doesn't overpower the generator.

PROPOSED MODEL
C. BASIC GAN MODEL



Simple GAN Loss Visualization

GAN Model Params:

- Input Image Format – (22109, 19200)
- Simple Dense Layers with Leaky ReLU activation and Dropout Regularization
- Ran over 500 Epochs with 128 Batch Size
- Learning Rate = 0.0002, beta = 0.5
- Adam Optimizer

Loss Interpretation:

- Continuous increase in the generator loss suggests that the generator is struggling to produce realistic samples, possibly due to a lack of challenge for the discriminator, insufficient model capacity, or training instability.
- Data seems like overfitting because of high number of epochs with high batch size.

GitHub: https://github.com/sohi-g/image-generation/modelling/Modelling-GAN.ipynb

PROPOSED MODEL

D. GAN MODEL RESULTS

Simple GAN Output of Generated Images

Generated Images are blurred, possibly because of the following reasons: Simple Model Training, High Generator Loss, Overfitting

GitHub: https://github.com/sohi-g/image-generation/modelling/Modelling-GAN.ipynb

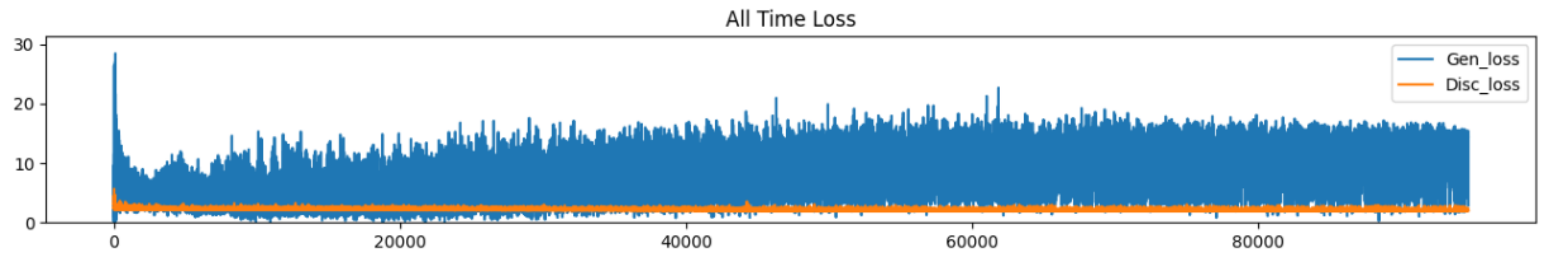
REGULARIZATION

Regularization Techniques used:

- Trained Data on Tensor Image with shape – (32, 64, 64, 3)
- Improvised Model to C-GAN (Conditional GAN), with conditional information of breeds of dogs with tensor input – (32,), for the purpose to generate more controlled and contextually relevant outputs
- Added new layers of Conv2DTranspose to upscale low-resolution feature maps to higher resolution, generating more detailed output in generator
- Added new layers of Conv2D that performs convolution operations to extract features from input images, helping classify real and generated images in discriminator
- Ran over 250 Epochs with 32 Batch Size (Reduced to prevent overfitting)
- Added a limit to learning rate, if epochs>180, exponentially reduce learning rate

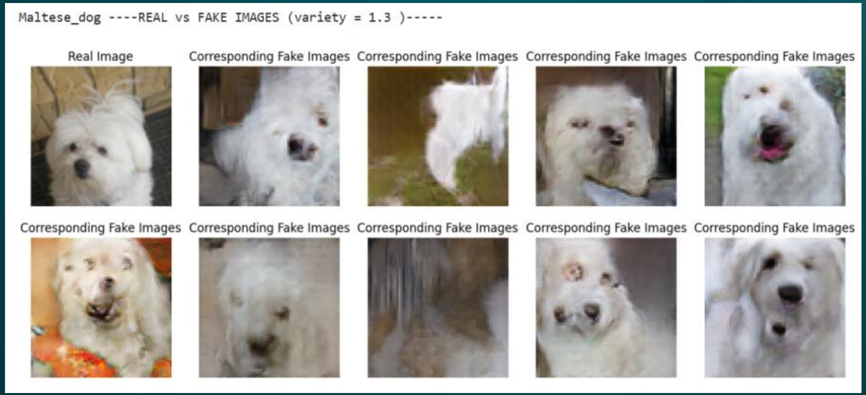
Loss Interpretation:

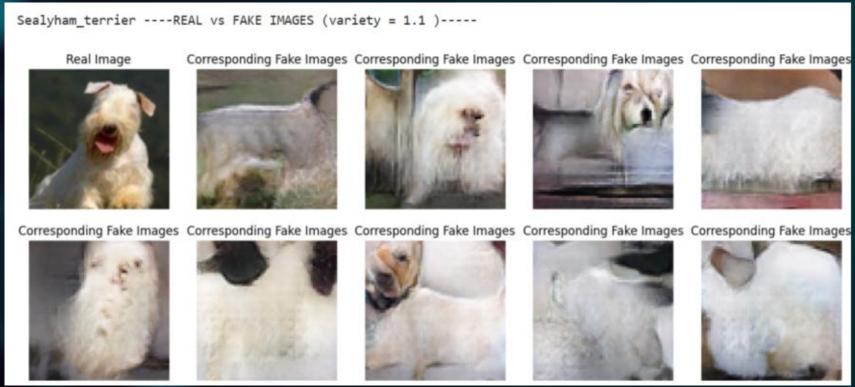
- It can be noted in below visualization that generator loss is better regulated, improving the performance of base model.

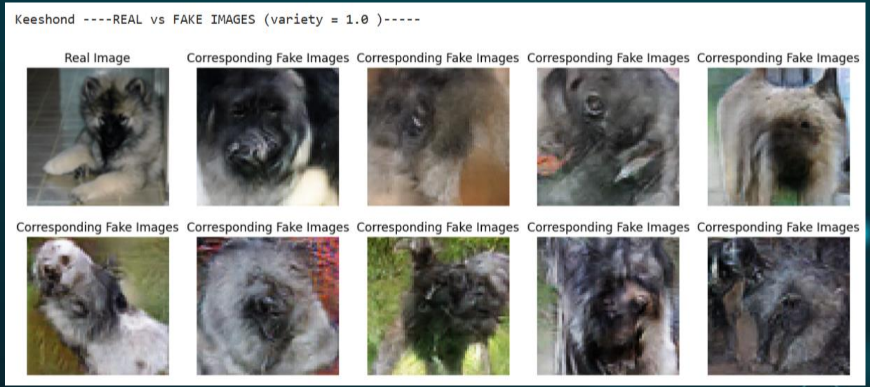


C-GAN Loss Visualization
GitHub: https://github.com/sohi-g/image-generation/modelling/Modelling-CGAN.ipynb

FINAL RESULTS - CGAN







C-GAN generates better images as can be seen above

GitHub: https://github.com/sohi-g/image-generation/modelling/Modelling-CGAN.ipynb

LEARNINGS & CHALLENGES

LEARNINGS:

- Gained insights into the workings of Generative Adversarial Networks (GANs) and their ability to create new data like a given training set.
- Explored Image generation techniques and manipulation of image attributes.
- Better understanding of Evaluation metrics for measuring the quality and diversity of generated images, in this case, generator and discriminator losses.
- Worked on Hyperparameter tuning and its impact on GAN performance to learn the importance of tuning hyperparameters, such as learning rates, regularization techniques.

CHALLENGES:

- The most challenging part was the computational resource requirements for training and iterating on models, since my system crashed a couple of times while running the model. Worked on multiple could platforms like Colab, GCP to successfully the models.
- Another challenge was mitigating overfitting, where the generated images were closely resembling the training data but lacked generalization to unseen examples. Reduced epochs and batch size to prevent overfitting.
- Bringing down the high generator loss was tricky in the beginning. But later resolved it by adding more accurate layers to model.

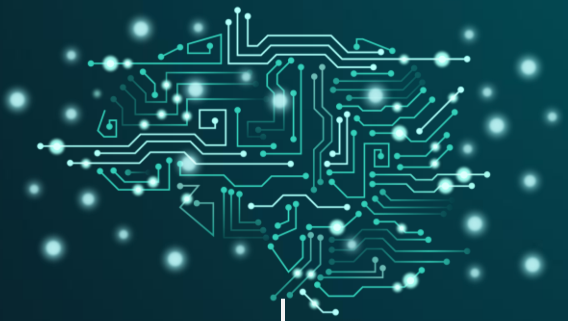
FUTURE WORKS

IMPROVING IMAGE QUALITY

There is scope to improve image resolution and detail, potentially by utilizing advanced GAN architectures like StyleGAN or BigGAN.

EXPANDING DATA SCOPE

Future projects could broaden the scope by training the GAN model on various datasets to generate images of different animals or even non-living objects.

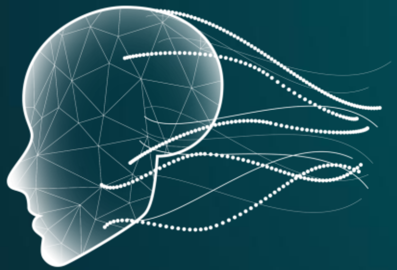


EVALUATING FAIRNESS AND BIAS

Future work could focus on assessing and mitigating potential bias in the GAN model to ensure it fairly represents all dog characteristics like breed, size, and colour.

VIDEO GENERATION

Upon mastering image generation, future work could explore creating short video clips, adding the complexity of producing consistent and realistic sequences of frames.



THANK YOU