Neural Networks and Fuzzy Logic
BITS F312

Research Assignment

Training Generative Adversarial Networks with Limited Data

Paper ID: 59

Group Members

NAME	***ID***
Aadit Maniar	2018A7PS0205P
Nikhil Bhamwani	2018B1A80686P
Prajwal Ranjan	2018B4A80414P

Title and Affiliation of Authors

Training Generative Adversarial Networks with Limited Data

AUTHOR	*AFFILIATED TO*
Tero Karras	NVIDIA
Miika Aittala	NVIDIA
Janne Hellsten	NVIDIA
Samuli Laine	NVIDIA
Timo Aila	NVIDIA
Jaakko Lehtinen	NVIDIA and Aalto University

Description of the Paper

Aim

Training Generative Adversarial Networks (or GANs) with limited data often leads to overfitting. This results in divergence in training. Collecting large amounts of data (of order 1-10 million) is an extremely difficult task. The aim of this paper is to train GANs using limited data. With the help of the methods proposed in this paper, we can train GANs using few thousand images, which can be collected easily from sources such as the Internet.

Proposed Methodology

Aim

In order to deal with overfitting, the data can be augmented. Variations, such as noise, rotations etc. can be incorporated into the dataset in order to improve the Generator training. Sometimes, however, augmenting the data results in the Generator reproducing images with the applied variations. This paper suggests the use of an Adaptive Discriminator Augmentation (ADA) mechanism for training GANs. This mechanism would significantly stabilize training processes when limited data is available.

Proposed Methodology

Aim

In the ADA method, the augmentation strength is varied during training, i.e., at different points in the training process, the augmentation amount is adjusted and applied. This training method is used on the StyleGAN2 model, developed by NVIDIA, that is used to train GANs with large datasets. The StyleGAN2 model is the successor to the StyleGAN model, and has trained many other models, such as GauGAN, GameGAN, GANimal etc.

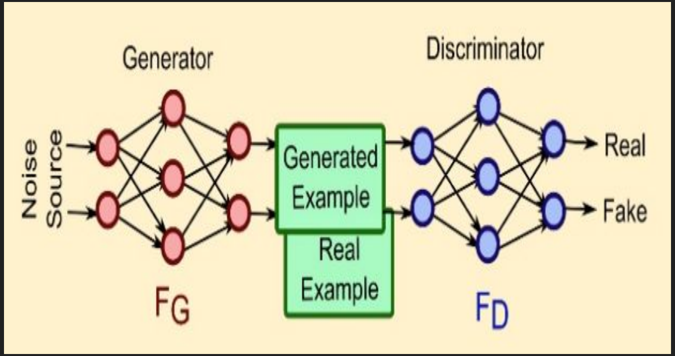
Final Outcome

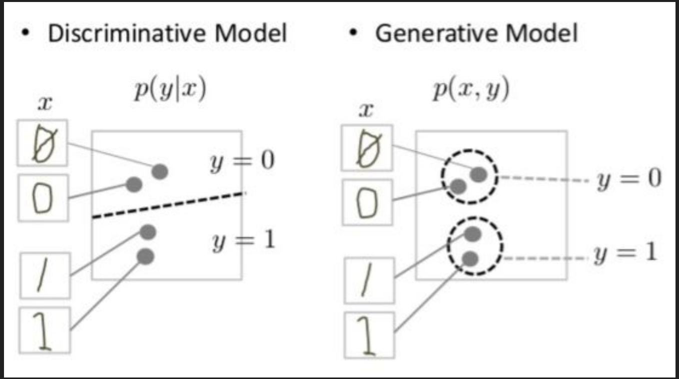
Using the ADA implementation while training GANs (such as StyleGAN2) produces high quality results with only a few thousand images. It reduces the required dataset size while producing excellent results, similar to those of models trained with significantly larger datasets. This method when applied with the CIFAR-10 dataset, produces results having FID scores of only 2.42.

Background Concepts

Generative Adversarial Network (GAN)

A Generative Adversarial Network, or GAN, is a machine learning implementation, where two neural networks compete with each other to become more accurate. GANs primarily consist of a generator network, G, and a discriminator network, D. A GAN consists of a Generator and a Discriminator. The Generator produces replicas of the original data, which is then fed to the Discriminator as input. The Discriminator then identifies whether the input is original or artificially generated.



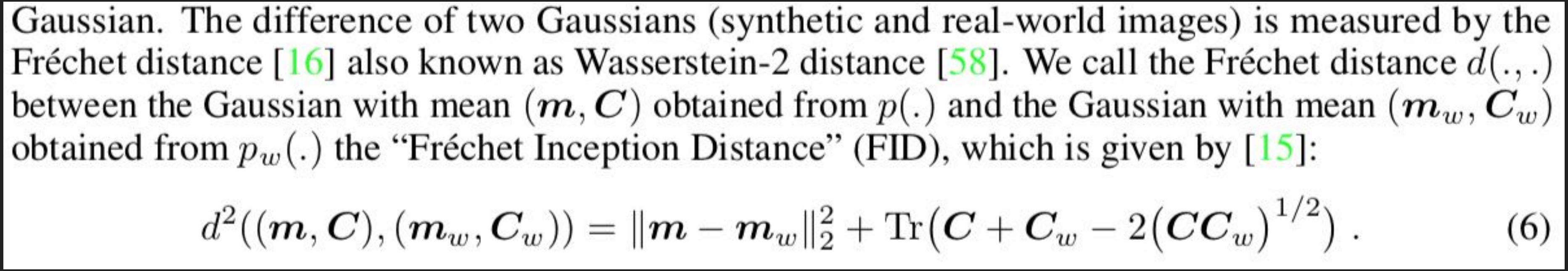


Transfer Learning

Transfer Learning is a machine learning method in which a model, that is developed and trained for a task, is used as the starting point for another task. Currently, many state-of-the-art models exist, that produce human-like results. At industrial levels, these efficient pre-trained models can be used as starting points in training processes, therefore increasing the efficiency, quality and speed of the task. This method is widely being implemented in areas such as Self-Driving Cars (by Udacity or Google), Robotic Simulations etc.

Frechet Inception Distance (FID)

The Frechet Inception Distance score, or FID score, is a metric, that is used to assess the quality of the images generated by the Generator network of GANs. It does so, by comparing the Gaussian Distribution of the real images with that of the generated images. Below, is a snipped briefly explaining it.



Gaussian. The difference of two Gaussians (synthetic and real-world images) is measured by the Fréchet distance [16] also known as Wasserstein-2 distance [58]. We call the Fréchet distance $d(.,.)$ between the Gaussian with mean $(\boldsymbol{m}, \boldsymbol{C})$ obtained from $p(.)$ and the Gaussian with mean $(\boldsymbol{m}_w, \boldsymbol{C}_w)$ obtained from $p_w(.)$ the "Fréchet Inception Distance" (FID), which is given by [15]:

$$d^2((\boldsymbol{m}, \boldsymbol{C}), (\boldsymbol{m}_w, \boldsymbol{C}_w)) = \|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \operatorname{Tr}\left(\boldsymbol{C} + \boldsymbol{C}_w - 2\left(\boldsymbol{C}\boldsymbol{C}_w\right)^{1/2}\right) . \qquad (6)$$

Kernel Inception Distance (KID)

Kernel Inception Distance score, or KID score, serves as an image assessment metric by measuring the dissimilarity between two probability distributions Pr and Pg using samples drawn independently from each distribution. It is considered to be an improvement on FID for small validation datasets.

Dataset Details

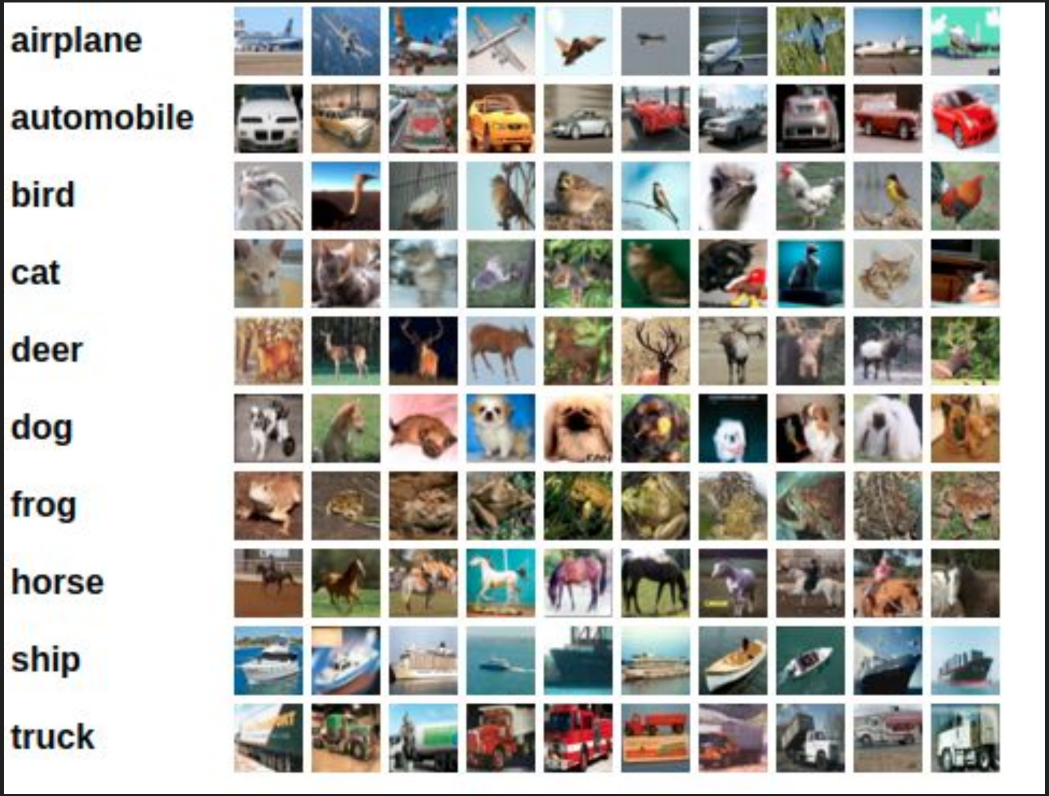
The Dataset

- We have used the CIFAR-10 dataset for training the discriminator.
- This happens to be the dataset on which the original StyleGAN implementation by NVIDIA has achieved state of the art results.
- The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class.
- There are 50000 training images and 10000 test images.

The Dataset

This is what some of the images belonging to the CIFAR-10 dataset look like.

For our application in GANs, the class information has very little significance.



Allotted Tasks

Allotted Tasks

1. Performing data augmentation
2. Training the GAN
3. Evaluating the trained network on the following metrics
    a. Inception Score (I.S.)
    b. Frechet Inception Distance (F.I.D.)
    c. Kernel Inception Distance (K.I.D.)
4. Comparison of our results with those of the authors.

Progress in Allotted Tasks

Progress in Allotted Tasks

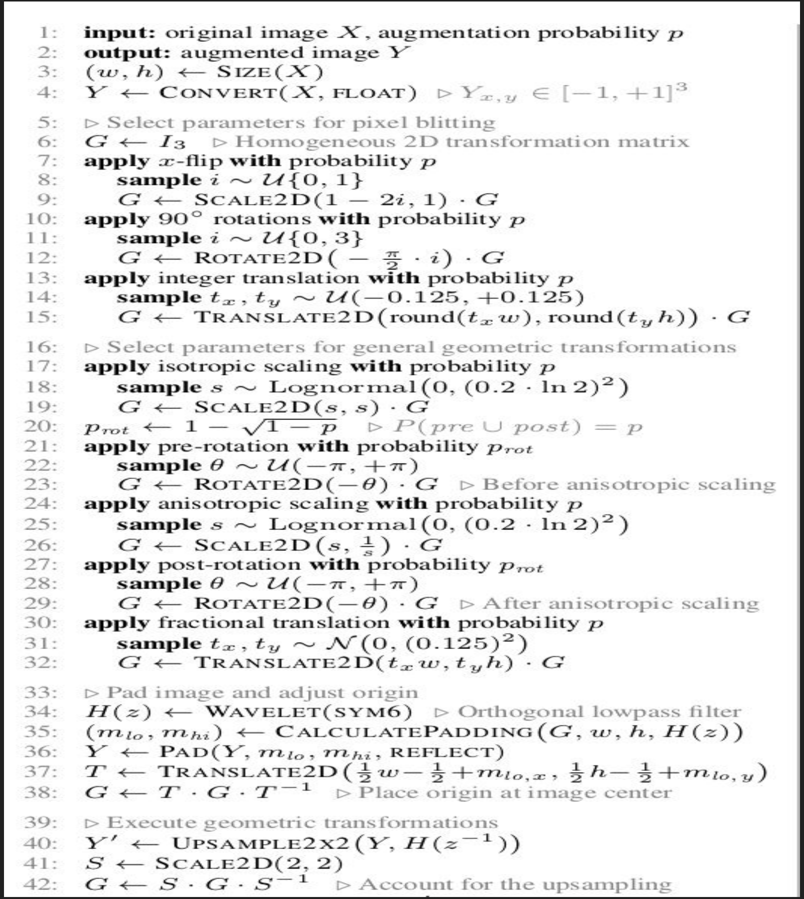
1. Performing data augmentation - Complete
2. Training the GAN - Incomplete, we used a pre-trained model due to hardware limitations.
3. Evaluating the trained network on the following metrics
    a. Inception Score (I.S.) - Incomplete due to hardware limitations
    b. Frechet Inception Distance (F.I.D.) - Complete
    c. Kernel Inception Distance (K.I.D.) - Complete
4. Comparison of our results with those of the authors. - Complete

Implementation Details

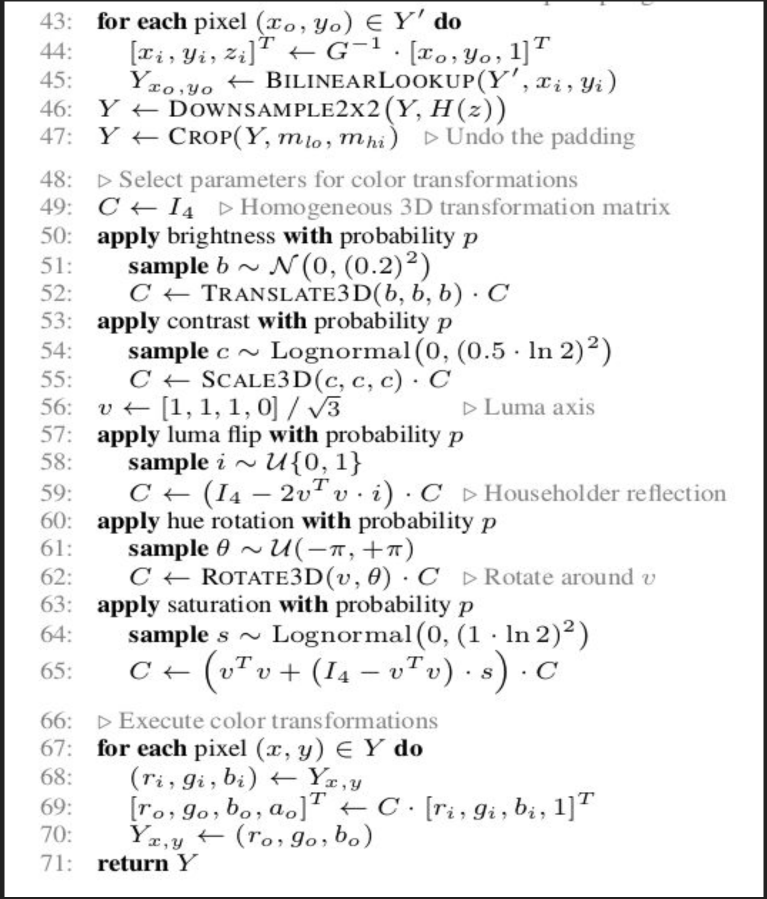
Implementation Details

Our techniques have been implemented on top of the official StyleGAN2 implementation. We have used Pytorch for the implementation. We kept most of the details unchanged, including network architectures, weight demodulation, path length regularization, lazy regularization, style mixing regularization, bi-linear filtering in all up/downsampling layers, equalized learning rate for all trainable parameters, minibatch standard deviation layer at the end of the discriminator, exponential moving average of generator weights, non-saturating logistic loss with R 1 regularization, and Adam optimizer with $\beta1 = 0$, $\beta2 = 0.99$, and epsilon = 10(E −8). Computation of Inception Score was implemented using the pre-trained inception-v3 model.

Implementation Details - Pseudocode for Geometric and Colour Transformations

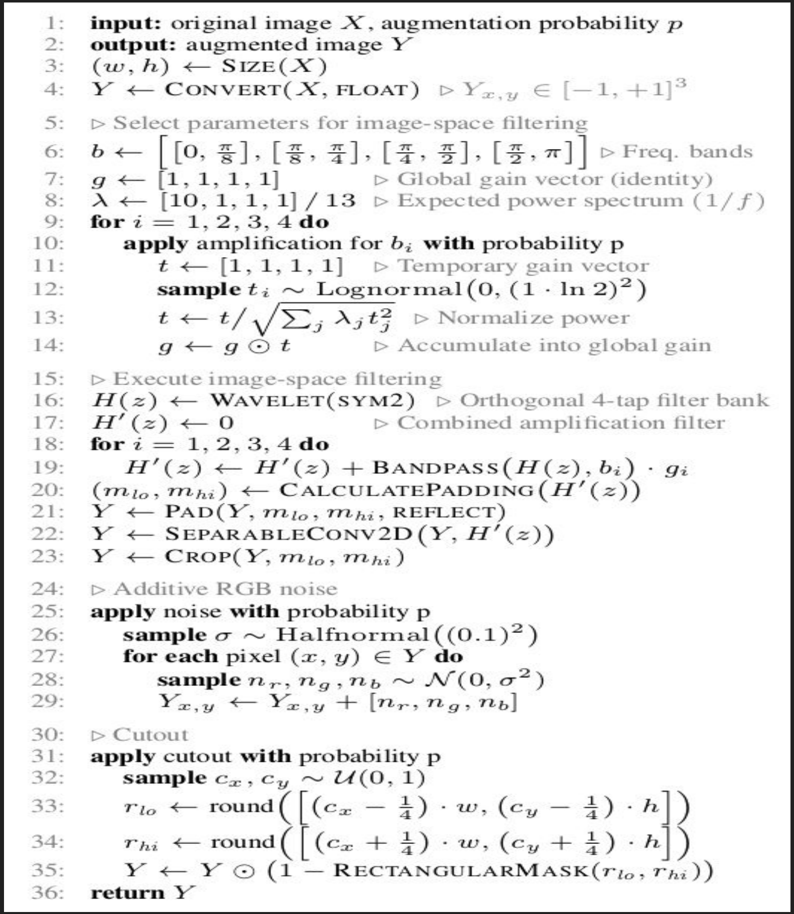


```
input: original image $X$, augmentation probability $p$
output: augmented image $Y$
$(w, h) \leftarrow$ SIZE($X$)
$Y \leftarrow$ CONVERT($X$, FLOAT)   ▷ $Y_{x,y} \in [-1,+1]^3$

▷ Select parameters for pixel blitting
$G \leftarrow I_3$   ▷ Homogeneous 2D transformation matrix
apply $x$-flip with probability $p$
    sample $i \sim \mathcal{U}\{0,1\}$
    $G \leftarrow$ SCALE2D$(1-2i, 1) \cdot G$
apply 90° rotations with probability $p$
    sample $i \sim \mathcal{U}\{0,3\}$
    $G \leftarrow$ ROTATE2D$(-\frac{\pi}{2} \cdot i) \cdot G$
apply integer translation with probability $p$
    sample $t_x, t_y \sim \mathcal{U}(-0.125, +0.125)$
    $G \leftarrow$ TRANSLATE2D$(\text{round}(t_x w), \text{round}(t_y h)) \cdot G$

▷ Select parameters for general geometric transformations
apply isotropic scaling with probability $p$
    sample $s \sim \text{Lognormal}(0, (0.2 \cdot \ln 2)^2)$
    $G \leftarrow$ SCALE2D$(s, s) \cdot G$
$p_{rot} \leftarrow 1 - \sqrt{1-p}$   ▷ $P(pre \cup post) = p$
apply pre-rotation with probability $p_{rot}$
    sample $\theta \sim \mathcal{U}(-\pi, +\pi)$
    $G \leftarrow$ ROTATE2D$(-\theta) \cdot G$   ▷ Before anisotropic scaling
apply anisotropic scaling with probability $p$
    sample $s \sim \text{Lognormal}(0, (0.2 \cdot \ln 2)^2)$
    $G \leftarrow$ SCALE2D$(s, \frac{1}{s}) \cdot G$
apply post-rotation with probability $p_{rot}$
    sample $\theta \sim \mathcal{U}(-\pi, +\pi)$
    $G \leftarrow$ ROTATE2D$(-\theta) \cdot G$   ▷ After anisotropic scaling
apply fractional translation with probability $p$
    sample $t_x, t_y \sim \mathcal{N}(0, (0.125)^2)$
    $G \leftarrow$ TRANSLATE2D$(t_x w, t_y h) \cdot G$

▷ Pad image and adjust origin
$H(z) \leftarrow$ WAVELET(SYM6)   ▷ Orthogonal lowpass filter
$(m_{lo}, m_{hi}) \leftarrow$ CALCULATEPADDING$(G, w, h, H(z))$
$Y \leftarrow$ PAD$(Y, m_{lo}, m_{hi}$, REFLECT)
$T \leftarrow$ TRANSLATE2D$(\frac{1}{2}w - \frac{1}{2} + m_{lo,x}, \frac{1}{2}h - \frac{1}{2} + m_{lo,y})$
$G \leftarrow T \cdot G \cdot T^{-1}$   ▷ Place origin at image center

▷ Execute geometric transformations
$Y' \leftarrow$ UPSAMPLE2X2$(Y, H(z^{-1}))$
$S \leftarrow$ SCALE2D$(2, 2)$
$G \leftarrow S \cdot G \cdot S^{-1}$   ▷ Account for the upsampling
```



```
for each pixel $(x_o, y_o) \in Y'$ do
    $[x_i, y_i, z_i]^T \leftarrow G^{-1} \cdot [x_o, y_o, 1]^T$
    $Y_{x_o,y_o} \leftarrow$ BILINEARLOOKUP$(Y', x_i, y_i)$
$Y \leftarrow$ DOWNSAMPLE2X2$(Y, H(z))$
$Y \leftarrow$ CROP$(Y, m_{lo}, m_{hi})$   ▷ Undo the padding

▷ Select parameters for color transformations
$C \leftarrow I_4$   ▷ Homogeneous 3D transformation matrix
apply brightness with probability $p$
    sample $b \sim \mathcal{N}(0, (0.2)^2)$
    $C \leftarrow$ TRANSLATE3D$(b, b, b) \cdot C$
apply contrast with probability $p$
    sample $c \sim \text{Lognormal}(0, (0.5 \cdot \ln 2)^2)$
    $C \leftarrow$ SCALE3D$(c, c, c) \cdot C$
$v \leftarrow [1, 1, 1, 0] / \sqrt{3}$   ▷ Luma axis
apply luma flip with probability $p$
    sample $i \sim \mathcal{U}\{0,1\}$
    $C \leftarrow (I_4 - 2v^T v \cdot i) \cdot C$   ▷ Householder reflection
apply hue rotation with probability $p$
    sample $\theta \sim \mathcal{U}(-\pi, +\pi)$
    $C \leftarrow$ ROTATE3D$(v, \theta) \cdot C$   ▷ Rotate around $v$
apply saturation with probability $p$
    sample $s \sim \text{Lognormal}(0, (1 \cdot \ln 2)^2)$
    $C \leftarrow \left(v^T v + (I_4 - v^T v) \cdot s\right) \cdot C$

▷ Execute color transformations
for each pixel $(x, y) \in Y$ do
    $(r_i, g_i, b_i) \leftarrow Y_{x,y}$
    $[r_o, g_o, b_o, a_o]^T \leftarrow C \cdot [r_i, g_i, b_i, 1]^T$
    $Y_{x,y} \leftarrow (r_o, g_o, b_o)$
return $Y$
```

Implementation Details - Pseudocode for Image-Space Filtering and Corruptions



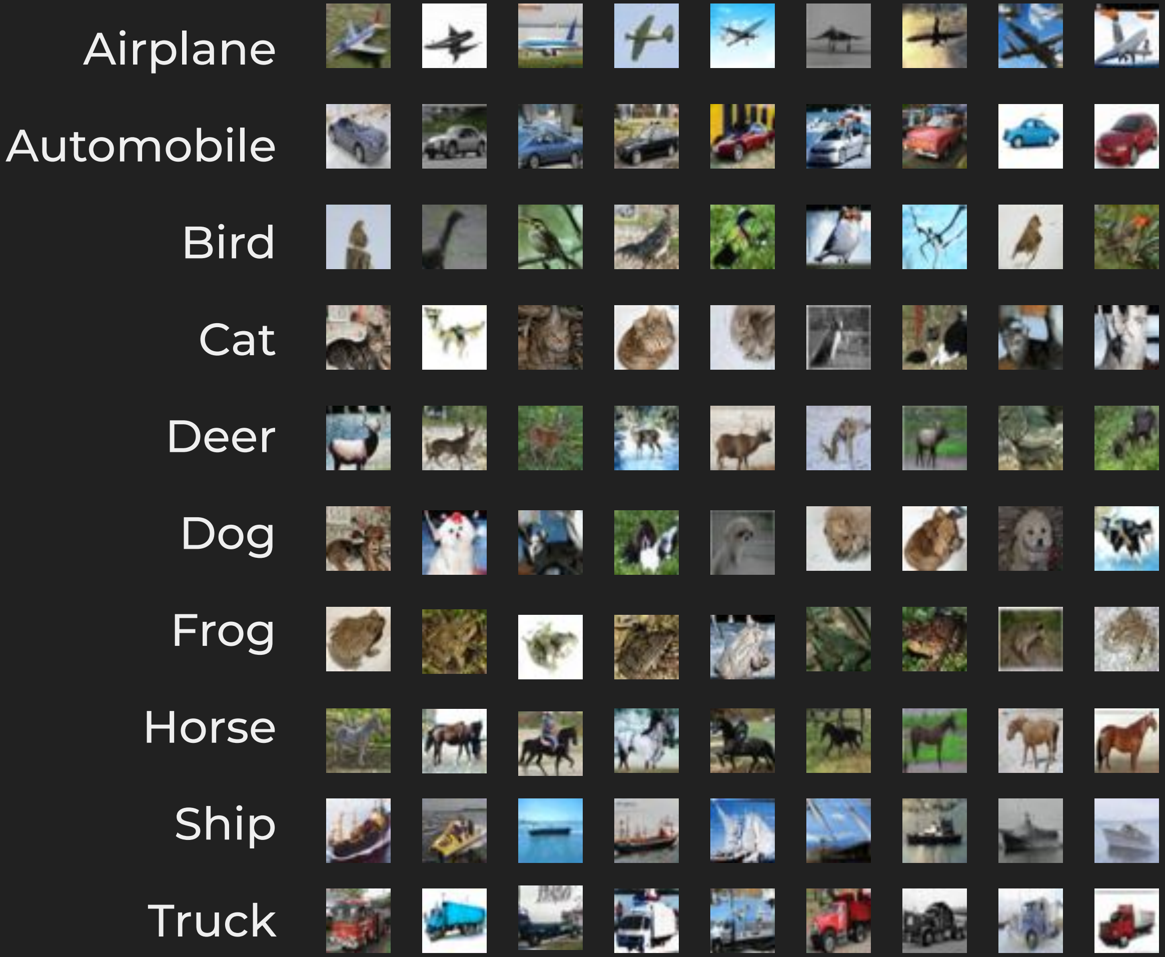
```
input: original image $X$, augmentation probability $p$
output: augmented image $Y$
$(w, h) \leftarrow$ SIZE($X$)
$Y \leftarrow$ CONVERT($X$, FLOAT)   $\triangleright$ $Y_{x,y} \in [-1, +1]^3$

$\triangleright$ Select parameters for image-space filtering
$b \leftarrow \left[ [0, \frac{\pi}{8}], [\frac{\pi}{8}, \frac{\pi}{4}], [\frac{\pi}{4}, \frac{\pi}{2}], [\frac{\pi}{2}, \pi] \right]$   $\triangleright$ Freq. bands
$g \leftarrow [1, 1, 1, 1]$   $\triangleright$ Global gain vector (identity)
$\lambda \leftarrow [10, 1, 1, 1] \,/\, 13$   $\triangleright$ Expected power spectrum ($1/f$)
for $i = 1, 2, 3, 4$ do
    apply amplification for $b_i$ with probability p
        $t \leftarrow [1, 1, 1, 1]$   $\triangleright$ Temporary gain vector
        sample $t_i \sim \text{Lognormal}(0, (1 \cdot \ln 2)^2)$
        $t \leftarrow t / \sqrt{\sum_j \lambda_j t_j^2}$   $\triangleright$ Normalize power
        $g \leftarrow g \odot t$   $\triangleright$ Accumulate into global gain

$\triangleright$ Execute image-space filtering
$H(z) \leftarrow$ WAVELET(SYM2)   $\triangleright$ Orthogonal 4-tap filter bank
$H'(z) \leftarrow 0$   $\triangleright$ Combined amplification filter
for $i = 1, 2, 3, 4$ do
    $H'(z) \leftarrow H'(z) +$ BANDPASS$(H(z), b_i) \cdot g_i$
$(m_{lo}, m_{hi}) \leftarrow$ CALCULATEPADDING$(H'(z))$
$Y \leftarrow$ PAD$(Y, m_{lo}, m_{hi}$, REFLECT$)$
$Y \leftarrow$ SEPARABLECONV2D$(Y, H'(z))$
$Y \leftarrow$ CROP$(Y, m_{lo}, m_{hi})$

$\triangleright$ Additive RGB noise
apply noise with probability p
    sample $\sigma \sim \text{Halfnormal}((0.1)^2)$
    for each pixel $(x, y) \in Y$ do
        sample $n_r, n_g, n_b \sim \mathcal{N}(0, \sigma^2)$
        $Y_{x,y} \leftarrow Y_{x,y} + [n_r, n_g, n_b]$

$\triangleright$ Cutout
apply cutout with probability p
    sample $c_x, c_y \sim \mathcal{U}(0, 1)$
    $r_{lo} \leftarrow \text{round}\left( \left[ (c_x - \frac{1}{4}) \cdot w, (c_y - \frac{1}{4}) \cdot h \right] \right)$
    $r_{hi} \leftarrow \text{round}\left( \left[ (c_x + \frac{1}{4}) \cdot w, (c_y + \frac{1}{4}) \cdot h \right] \right)$
    $Y \leftarrow Y \odot (1 -$ RECTANGULARMASK$(r_{lo}, r_{hi}))$
return $Y$
```

Implementation Details - Hyperparameters for training on CIFAR-10

PARAMETER	VALUE
Resolution	32x32
Number of GPUs used	1 (Tesla T4)
Training Length	50,000 images
Minibatch size	64
Learning Rate	2.5×10^{3}
Regularization Constant	0.01

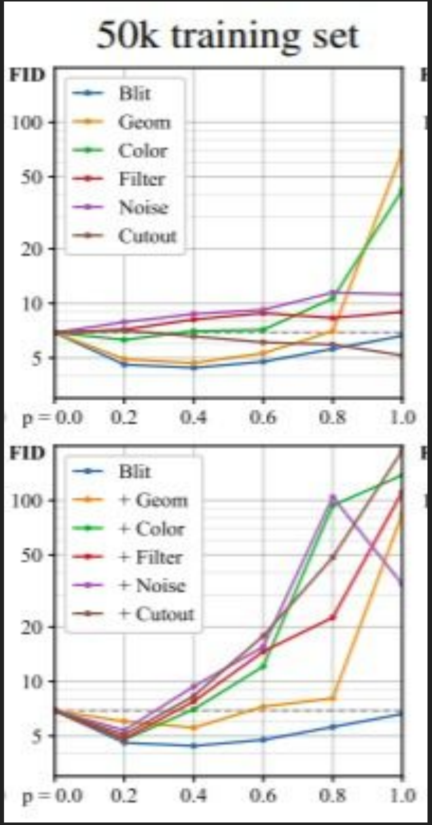
Results and Discussions

Results - Generated Images



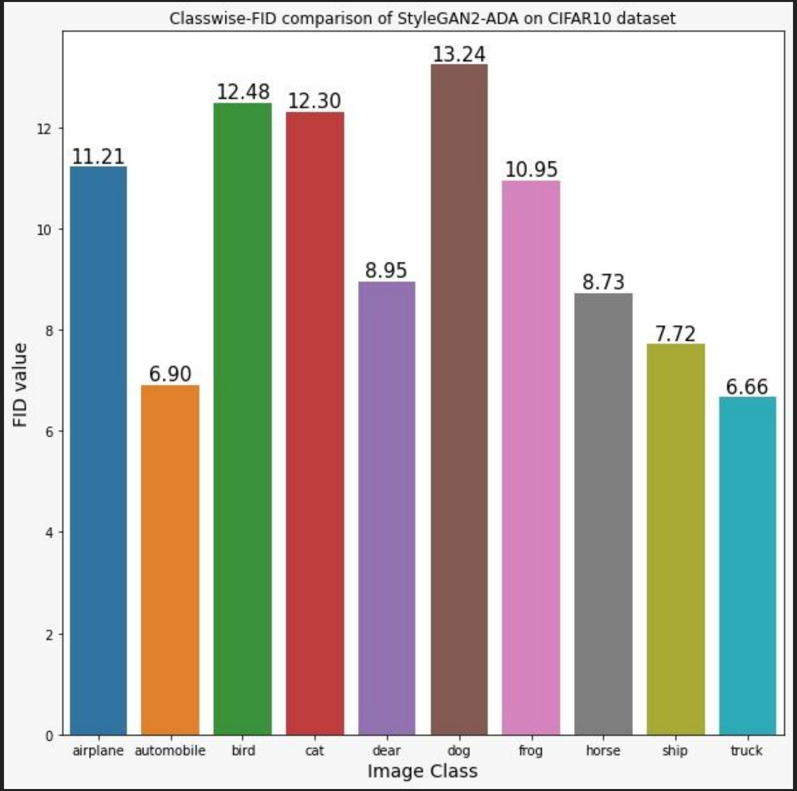
Results

The plot on the right depicts the Frechet Inception Distance (FID) values for images generated before and after augmentations. It gives an idea of how augmentations perform on 50000 data images.



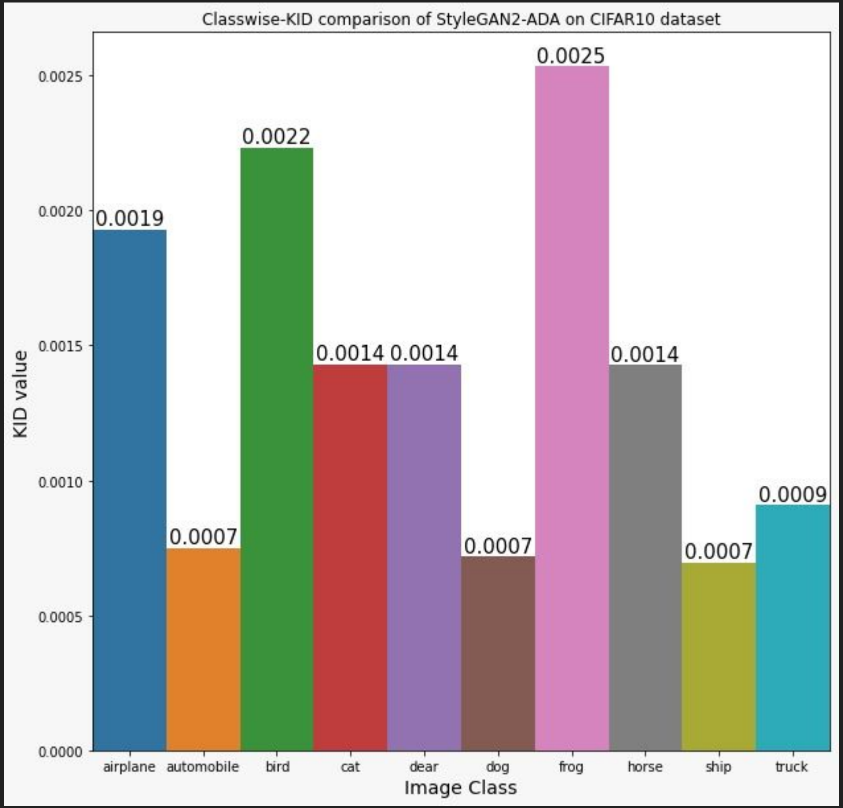
Results

The bar graph shows the Frechet Inception Distance (FID) values for the images generated corresponding to each of CIFAR-10 classes. 10,000 images per class were used to generate this score. Lower FID values indicate better performance.



Results

The bar graph shows the Kernel Inception Distance values for the images generated corresponding to each of CIFAR-10 classes. 3000 images per class were used to generate this score. Lower KID values indicate better performance.



Comparison with Authors

Comparison with Authors

Metric	Original performance (on 10k images)	Our performance
I.S. (higher is better)	10.02	2.825 on 500 images**
F.I.D. (lower is better)	5.33	9.913
K.I.D. (lower is better)	Not Available*	0.0014

* KID has not been specified for cifar-10. Testing their pre-trained model on 500 images gives KID = 0.073
** Hardware issues faced since IS uses all layers of inception-v3

Challenges Faced

Challenges Faced

- GAN Training
    - The training with adaptive discriminator augmentation turned out to be computationally very expensive and required at least 1 GPU.
    - Due to unavailability of GPUs on our local systems, we used Google colab with a GPU enabled runtime for most of the implementation which has its own limitations.
    - Since the paper recommended at least 16GB of GPU and colab provided only 12GB, training took drastically longer time than anticipated.

Challenges Faced

- GAN validation
    - Validation and performance evaluation for GANs is not as straightforward and needs complex metrics such as Inception Score, FID, KID etc.
    - The reliability of these metrics tend to increase with an increase in the number of real examples used for validation.
    - Loading all 10000 CIFAR-10 test set images and running the evaluation algorithm crashed the colab runtime.
    - Hence we had to use a smaller number of images for performance evaluation due to which their values appear worse than they should actually be.

Future Scope

Future Scope

- Improved metrics for performance evaluation of GANs or existing metrics using improved models could further boost the effect of adaptive discriminator augmentation in training and give better GANs.
- Such methods can be employed in healthcare industries, to develop better biological images. It can also be used along with Computer Vision in fault detection at unusual areas, such as looking for cracks in electrical poles etc.

Experience and Learning Outcomes

Experience and Learning Outcomes

- Got a good exposure to modern state of the art research in Artificial Intelligence.
- Learnt a lot about the working of GANs and the new challenges faced in training and evaluation.
- Achieved a good amount of confidence in practically implementing the concepts learnt.