Xi'an Jiaotong-Liverpool University
西交利物浦大學

SCHOOL OF ADVANCED TECHNOLOGY

STYLEDIFFUSER: CARTOON-STYLE IMAGE CREATION WITH DIFFUSION MODEL AND GAN FUSIONS

AUTHOR: DONGHENG LIN(ID: 1929066)
PROJECT SUPERVISOR: DR. ERICK PURWANTO

Declaration

I confirm that I have read and understood the University's Academic Integrity Policy.
I confirm that I have acted honestly, ethically and professionally in conduct leading to assessment for the programme of study.
I confirm that I have not copied material from another source nor committed plagiarism nor fabricated, falsified or embellished data when completing the attached piece of work.
I confirm that I have not copied material from another source, nor colluded with any other student in the preparation and production of this work.

Signed: Dongheng Lin Date: October 30, 2023

OCTOBER 30, 2023

Abstract

This final year project proposes a novel method, called StyleDiffuser, combining the Generative Adversarial Networks with Stable Diffusion Model which gives appropriate constraints to elastic and enormous Stable Diffusion Network with StyleGAN2 generated feature maps and corresponding feature metadata, therefore, leveraging the burden of diffusion steps required for output convergence and no longer requires verbose prompts to realize reasonable output control. Apart from the algorithm level contribution with the feasibility and improvement of the algorithm discussed, this work also involves corresponding data science procedure which constructs a dedicated dataset and developed a corresponding Web GUI for usability validation which is also included in this dissertation.

Contents

List of Figures

List of Tables

List of Acronyms

Table 1: List of Acronyms

Acronym	Terminology
GAN	Generative Adversarial Networks
GUI	Graphical User Interface
SFW	Suitable For Work
SOTA	State-of-the-art
GPU	Graphics Processing Unit
CPU	Central Processing Unit
ADA	Adaptive Discriminator Augment
LoRA	Low-Rank Adaptation
VAE	Variational autoencoder
YOLO	You Only Look Once
CNN	Convolutional Neural Network
R-CNN	Regions with CNN Features
FID	Fréchet Inception Distance
AFHQ	Animal FacesHQ
FFHQ	Flickr-Faces-HQ
PGGAN	Progressive Growing GAN
DDPM	Denoising Diffusion Probabilistic Models
PIC	Perceptual Image Compression
CDA	Conditional Denoising Autoencode

DSE	Domain Specific Encoder
DE	Diffusion Enhancer
MVC	Model-View-Controller
ORM	Object-Relational Mapping
HTML	HyperText Mark-up Language
SFTP	SSH File Transfer Protocol

1 | Introduction

1.1 Motivation, Aims and Objective

Nowadays, many people are immersed in the world of the internet and many of them want a unique avatar for themselves to distinguish others from themselves. However, not everyone is a versed artist who has such proficiency. Therefore, this research could provide an example for people who wants identical personal icon without expert experience in the art field by constructing and training on customized dataset and model. The implementation of this project does not require any verbose and lengthy prompts as textual input to constraint the model for consistent output. The project takes advantages of asymmetry of the strong generative model with larger parameter space and weaker model which is more focused on specific features instead. Thus, in this project, the author would like to propose a new approach to Generative Adversarial Network (GAN) [7] architecture adopting mechanisms in Diffusion Model [8] by using a weaker but more style-consistent GAN model to cast stronger supervision over a stronger but less consistent Diffusion Model. This approach will then be evaluated by experiments on the self-collected and processed dataset together with a widely used benchmark dataset to examine the possible improvement of combining 2 prevailing generative networks in computer vision and provide insights to solve the dilemma, in which fidelity and creativity seem to contradict and cannot co-exist for one generator, of image generator models.

Therefore, this project aims to propose a new approach that fuses the diffusion process [8] and the Convolutional Neural Network based GAN models [7], and validate such a model on relatively small datasets that are planned to be dedicated to this project. The model is expected to have the advantages of both GAN and diffusion models. Besides, it is a Web GUI and provided for the user for the sake of accessibility of this work.

For all these works mentioned specifically, the technical details of this project is based on the conventional deep learning frameworks together with other supportive libraries as depicted in the table below. As mentioned above, the project covers full-stack experiments on the fusion of advantages of generative models which includes data engineering, model design, and deployment. By the end of these projects, there is a complete novel model with highly-usable web GUI access. A full list is attached below for detailed reference of all the deliverable.

Table 1.1: Experiment Platform Specification

	Experiment Environment
Pytorch	1.11.0
Python	3.8
CUDA	11.3
GPU	NVIDIA A40 & Tesla P100
CPU	AMD EPYC 7543 32-Core Processor

Table 1.2: Deliverables Done

Deliverables
- 512*512 Image Dataset
- A new model
- Functional Web GUI
- FYP Thesis for summary

To this end, the proposed project aims to develop a novel approach to Generative Adversarial Network (GAN) architecture that combines the strengths of GAN and diffusion models to generate personalized avatars for individuals without requiring expert experience in art. The project involves collecting and processing datasets, designing and training the model using deep learning frameworks and supportive libraries, and developing a web GUI to make the model accessible. The project seeks to solve the dilemma where fidelity and creativity seem to contradict and cannot co-exist for one generator. By the end of the project, there will be a complete novel model with highly-usable web GUI access.

1.2 Literature Review

There are a lot of successful GAN-variants that even works perfectly under several image dataset [5]. Those achievements earn GAN models a great reputation for image visual fidelity while they may still suffer from lacking mode coverage. These are the so-called inherent over-fitting, which results in lacking diversity in final image outputs, problems of GAN networks that require Adversarial Optimization as depicted in Fig.1.1 The GAN can be further illustrated as an optimization game in Eq.1 where D means Discriminator and G represents Generator whereas L is the loss function, which is used to measure the difference between the generated data and the real data. The parameters μ_G and μ_D represent the weights and biases of the generator and discriminator networks, respectively [7].

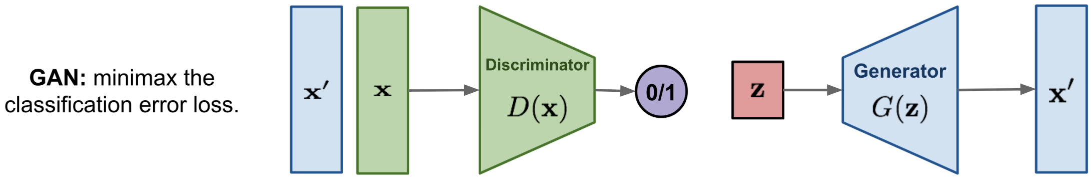


Figure 1.1: Generator-Discriminator Equilibrium in GAN model

$$E = min_{\mu_G} max_{\mu_D} L(\mu_G, \mu_D) \tag{1}$$

As for Diffusion models [8] shown in Fig.1.2 that are famous for their creativity, they are relatively more aesthetically satisfactory compared with their GAN counterparts. And more elegant in terms of model architecture which consists of only one model that needs to be optimized. However, due to their creativity, the model may return images with chaotic visual details, which are significantly not as good as GAN models. Apart from that, they cost much more computational power to converge in training time. Assume ϵ represents the noise generated randomly by standard Gaussian distribution, $\bar{x}$ is the result after n diffusion step and z stands for the final outcome.

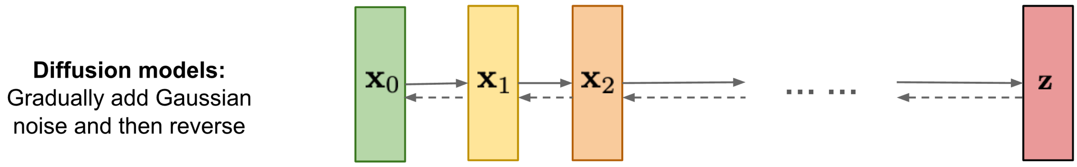


Figure 1.2: Diffusion Model Encoder-Decoder

$$\text{Loss } = E_{\bar{z},\epsilon}\left[\left\|\epsilon - \epsilon_0\left(\sqrt{\bar{x}_n}y_0 + \sqrt{1-\bar{x}_n}\epsilon, \sqrt{\bar{x}_n}\right)\right\|_1\right] \tag{2}$$

There have been some related works about the combination of these models, Denoising Diffusion GANs [9] proposed earlier shows the availability of using the Denoising Diffusion Probabilistic Model [8] in GAN structures to solve the trilemmas of Generative models. This work have pointed out the potential of combining these 2 types of models. Furthermore, there are also techniques like Low-rank approximation[6] and Adaptive Discriminator[2] to train the models on limited data on both types of model which have illustrated possible efficient implementation of each type of model with shows the feasibility of experiment. These techniques involve methods for augmentation, regularization, and low-rank adaptation of large language models. With such inspiration, the objectives of this project have become realistic and achievable.

Adopting GAN-like network structure and optimization techniques to Diffusion models have been proved by the state-of-the-art research, DiffusionGAN [10], which uses adversarial supervision to the diffusion steps themselves to check whether every interval of diffusion is still under control based on a discriminator checking. This work have shown excellent benchmark improvement which exploit the representation potential of diffusion model by a strong supervision from vanilla GAN-like architecture. However, it does not involve more advanced GAN-like neural networks' design but only applies the basic idea of adversarial training to the Diffusion process.

The most important idea of this project's approach involves using a weaker but more consistent GAN model to provide stronger supervision over a stronger Diffusion Model by technologies mentioned above. This basic idea is inspired by the work in [11], which adds block-wise zero-convolution layers to conduct control over huge diffusion models pre-trained while without additional input-level guidance as needed in ControlNet [11].

There is also a parallel work in fusion of both networks[12] published earlier before this dissertation, in which combines the GAN model and Diffusion Model by direct summations and concatenations between both networks which have been used for domain adaptation problems in the field of image generation and shows strong results proving the fusion mechanism, while this method is relative costly in terms of excessive dataset and prompts needed. In contrast, the project proposes a method requiring relatively smaller datasets and require nearly no additional textual constraints apart from metadata collected.

To sum up for this literature review, we have compared GAN and Diffusion models and highlights their respective strengths and weaknesses. While GANs produce visually appealing images, they may suffer from mode collapse and insufficient mode coverage, while Diffusion models require more computational power to train and may produce chaotic visual details. Techniques such as Denoising Diffusion GANs, Low-rank adaptation, and Adaptive Discriminator are discussed as potential solutions to limitations in training these models [8] [6] [2]. The proposed approach involves using a weaker but more consistent GAN model to provide stronger supervision over a stronger Diffusion Model, inspired by the work in ControllNet [11].

2 | Methodologies

The project can be separated into several stages which involve different topics of data science and machine learning. It will start with big-data-based data scraping and dataset construction. After proper datasets are made, corresponding models are designed to make predictions based on domain knowledge. Once the model is proven to be good enough to improve image generation, subsequent deployment will be implemented to provide convincing practical values for this project. At this stage of task fulfillment, in the sub sections below, we will discuss both the data science methodologies in collecting data needed for this project as well as the deep-learning model design, theories and experiments done.

2.1 Dataset Methodologies

Firstly, the dataset collection is based on data-scrapping with an open anime illustration database, Danbooru [13], which consists of various images with corresponding labels and metadata indicating art genre, image style, theme, parodies, etc. In order to construct a limited dataset with relatively consistent styles, a filter based on these labels is set and therefore ensures the dataset fidelity and coherence which defined a specific character-based problem domain. As a result, there are in total of 5138 images collected satisfying a 5 set of keywords which are "Patchouli Knowledge", "1girl" and "SFW (suitable for work)". Part of the images collected is listed in Fig.2.1 below.

Figure 2.1: Part of preliminary collected images

However, as illustrated above, the scraped whole images are not suitable for this project aiming to use minimum cost to build an efficient avatar-only generator based on few-shot image samples. Therefore, an automatic image cropping algorithm is adopted and implemented specifically for avatar cropping.

Therefore, anime-face cropping in dataset construction utilized a deep-learning approach which is by running the YOLOv5 network [1] on all the initial samples collected. The YOLOv5 neural network is trained dedicatedly for the anime faces by taking samples in another public dataset, Danbooru 2020 anime face dataset [14]. Such a YOLOv5 network redefines object detection as a regression problem. It applies a convolutional neural network (CNN) to the whole image which divides the image into grids and predicts the class probability and bounding box of each grid. YOLO models are comparably efficient since it transforms the detection problem into a common regression problem, in which there is no need for complex pipelines, which makes it significantly faster than traditional R-CNN [15] counterparts.

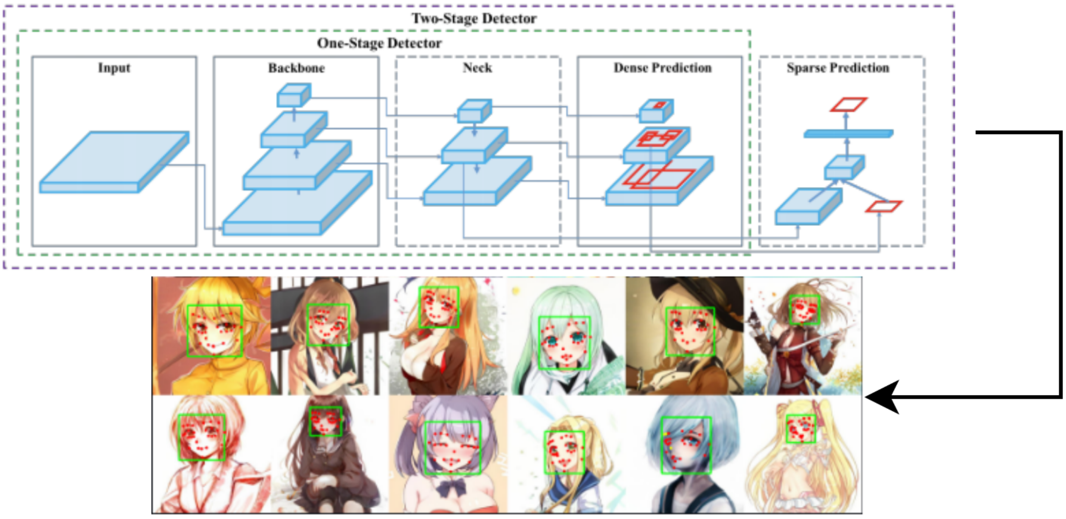
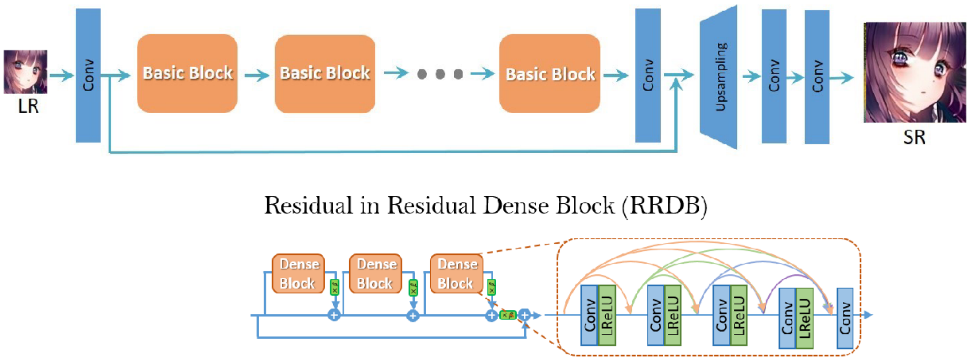


Figure 2.2: YOLO-based Anime Detector [1]

However, it is not enough to simply have the faces cropped, the resolution of the cropped faces ranges from 34*34 to 782*972, which needs to be homogenized to a specific resolution to be used for training. Therefore, it is necessary to conduct up-scaling for small samples and then resize them back into the proper size. Specifically, the Super-resolution reconstruction algorithm used in this process is an Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) [16] based approach that efficiently rebuilds the detailed textures from blurred patterns as shown in Fig.2.3 which illustrated the procedures in a 4x up-scaling. The images reconstructed were then resized to 512*512 which is a typical resolution used for such a dataset.



Residual in Residual Dense Block (RRDB)

Figure 2.3: ESRGAN Algorithm for Dataset construction

To this end, with the successfully collected and homogenized image samples. It is necessary to prune the dataset for there still exists data inconsistency in the dataset which contains images of different art styles. Therefore, an additional step of pruning the dataset is needed. Manual editing is not reasonable and exhausting, so a k-means image clustering model [17] is used to quickly identify image classes that differ from each other as shown in Fig.2.4. Then the minority classes are deleted. Finally, the image dataset is pruned from 5138 to the size of 1236 with a reasonable and consistent pattern as depicted in Fig.2.5 below.

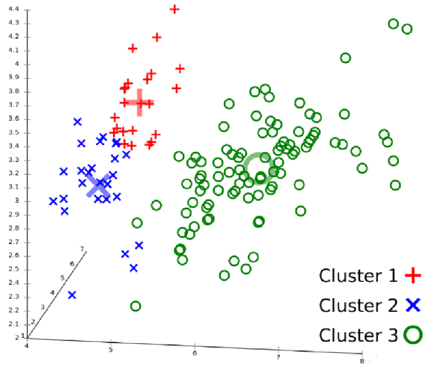


Figure 2.4: K-means Clustering when k = 3

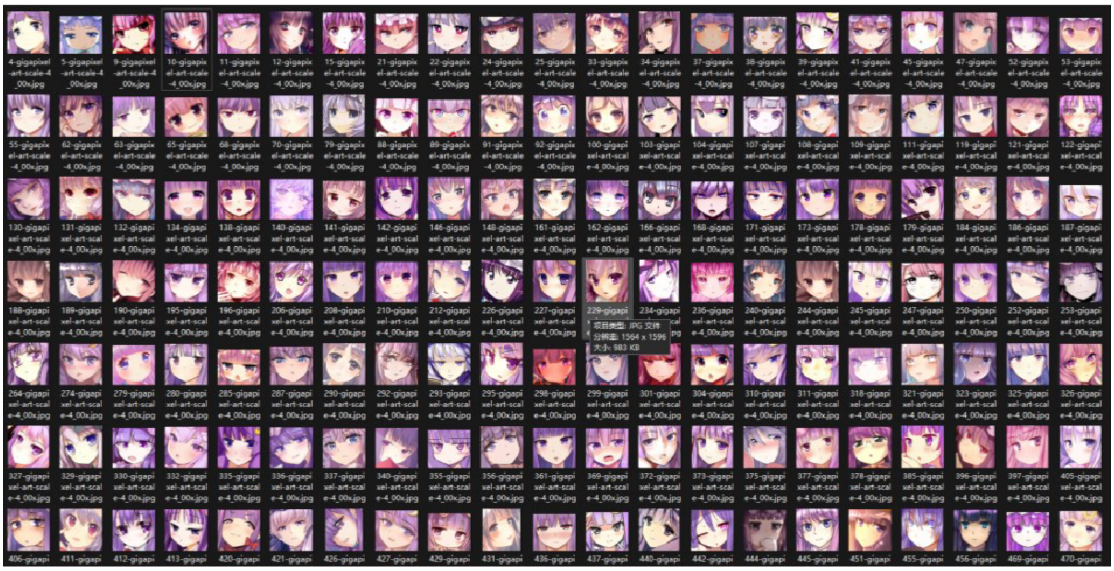
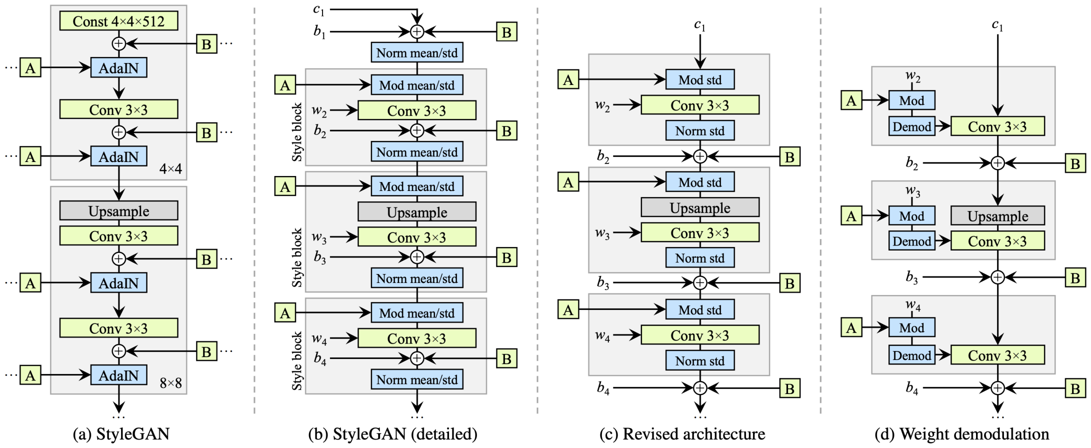


Figure 2.5: Part of the Produced dataset

2.2 Prior Experiment and Metrics

2.2.1 StyleGAN2 Prior Experiments

To identify problems of the state-of-the-art models and validate the dataset potential, experiments over a cutting-edge model with the dataset are conducted. The tested model is StyleGAN2. In 2018, NVIDIA released the StyleGAN [5] paper, Style-Based-GAN Architecture. This paper proposes a new GAN generator architecture, which allows them to control different levels of detail of generated samples. Then, StyleGAN2 [3] is proposed in 2020 which fixed some issues of previous StyleGAN including the water droplet artifacts problem, and realized scale-specific feature controlled by introducing the direct connection of the input with the output instead of continuing using the idea of PGGAN [4] in which training layer by layer fixed small size pictures that are sent to the discriminator for judgment, and then back-propagated as in the predecessor StyleGAN.



(a) StyleGAN (b) StyleGAN (detailed) (c) Revised architecture (d) Weight demodulation

Figure 2.6: StyleGAN Generator Evolution [2]

In StyleGAN, the generator model includes a "latent space" consisting of a set of vectors that are randomly sampled from a normal distribution. These vectors are then transformed by several layers of non-linear functions to generate an image. Latent noise refers to small random variations in this latent space vector that can be added to the generator's input during image synthesis. Essentially, the generator model maps a random vector z from a standard normal distribution to an output image X. This mapping is achieved through a series of non-linear transformations applied to intermediate latent vectors W. Therefore, these variations do not significantly change the overall structure or content of the generated image, but rather introduce subtle changes in its appearance such as texture, lighting, or color which gives a consistent style for the images generated.

Additionally, StyleGAN2 introduces a new normalization term is introduced to achieve smoother potential spatial interpolation by adding potential spatial interpolation and describes how changes in the source vector z led to changes in the generated image. This is achieved by adding the following loss items to the generator. Eq. 1 shows a Jacobian matrix used for that which compares the small change in w with the change in the resulting image. This matrix is multiplied by a random image to avoid falling into the local optimum. The l2 norm of this matrix is multiplied by its exponential moving average.

$$\mathrm{E}_{w,y\sim N(0,I)}\left(||J_w^T y\big|_2 - a\right)^2 \tag{1}$$

Since the dataset is quite limited compared with the original dataset the papers used, this project adopts an Adaptive Discriminator Augment mechanism (ADA), which can achieve significantly stable training even under conditions where data is quite limited [2]. During training, the ADA method dynamically adjusts the intensity of these random variations based on the performance of the discriminator network. Specifically, if the discriminator is performing well at distinguishing real and fake images, the intensity of the random variations is increased, making it more difficult for the discriminator to continue improving its accuracy. Conversely, if the discriminator is struggling to distinguish real and fake images, the intensity of the random variations is decreased, making it easier for the discriminator to learn.

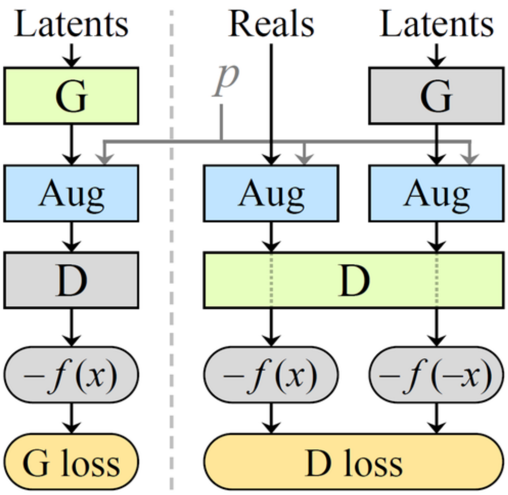


Figure 2.7: Adaptive Discriminator Augment (ADA) Structure [3]

2.2.2 Metrics

There are several formal metrics used in the evaluation of this project that can be considered benchmark milestones of the model proposed. We will first look at the Fréchet Inception Distance (FID) score [18] which evaluates the fidelity of generated images. In addition, mode coverage [19] will be another important benchmark that reflects how creative the model is when producing images compared to previous baselines. In FID, we extract features from an intermediate layer using an Inception network [20] in which distribution for these features are modeled using a multivariate Gaussian distribution with mean μ and covariance σ.

$$\mathrm{F}ID(x,g) = \|\mu_x - \mu_g\|_2^2 + Tr\left(\sum x + \sum g - 2\left(\sum x \sum g\right)^{\frac{1}{2}}\right) \tag{2}$$

The model's FID score is based on the inception distance against 50k real images. Therefore, after 160kimg the model seems fully converged to a stage where the outputs are reasonable avatar images with a corresponding FID score of 9.685418 against real images, which is comparably lower than the official results based on other open datasets [5]. Given the inconsistency and limitation of the dataset collected, which consist of only limited visual feature and styles compared with open datasets like AFHQ [21], CelebA-HQ [4], etc., such a degradation is expected and tolerable.

However, the current StyleGAN2 model reveals some shortcomings to be improved. As depicted in Fig.2.8, the model suffers greatly from occasional mode collapse, eye distortion, and even trails of glasses that do not supposed to be there. Therefore, there is still space for improvement with this cutting-edge implementation of GAN.

These problems are caused by the inherent problems of the GAN-based model. The GAN-based model hardly learns any actual relations between vi-

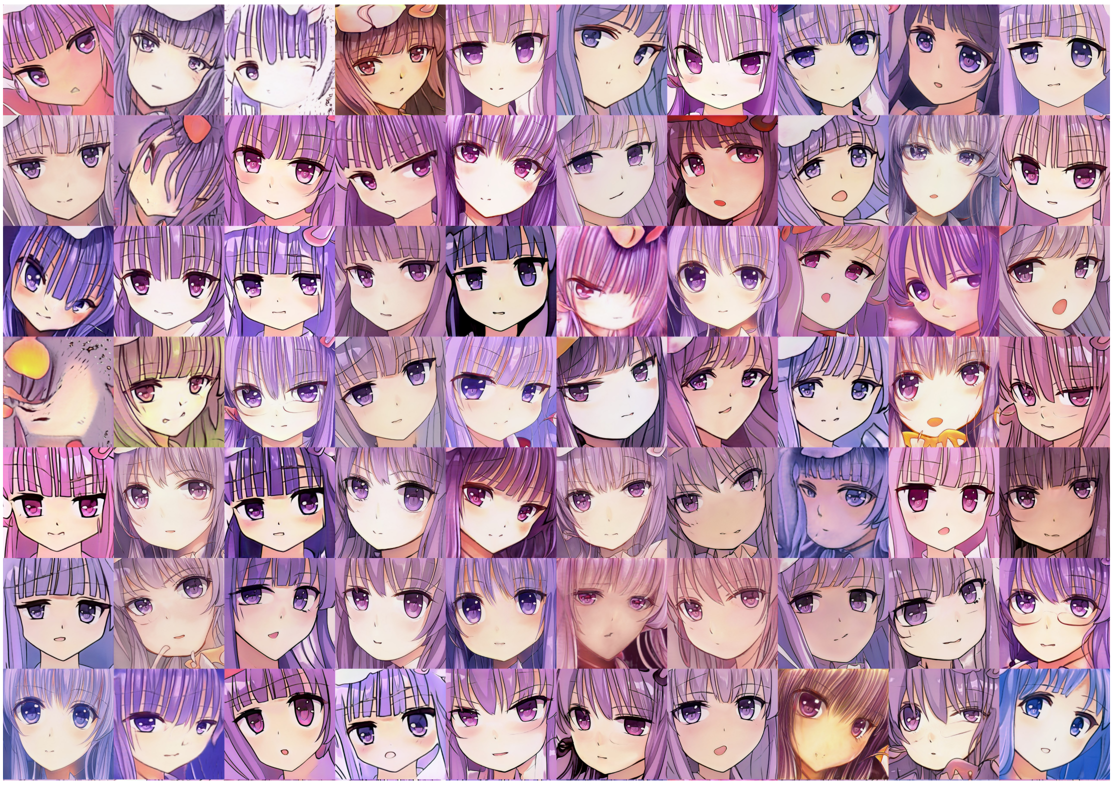


Figure 2.8: Random Images Generated

sual objects. It most likely absorbs features haphazardly as much as it can to fool the discriminator. This caused the so-called mode collapse and lacking mode coverage problem of Generative models before the introduction of Diffusion models[8] which successfully adapted Diffusion steps in noise processing steps, which models the reconstruction as a denoising procedure. The diffusion approach is more robust when dealing with inter-dataset inconsistency but is much more expensive in terms of computational power, which makes it nearly impossible for persistent web-application deployment without GPU server.

Therefore, starting from this dilemma, we may notice that generating by diffusion usually assumes that the distribution of each denoising steps can be modeled by Gaussian distribution. This assumption is applicable to PGGAN-like [4] generator generation steps, in which thousands of layers are passed in the generative process. Therefore, GAN generators can be an accelerator or even the main branch of a generative model. Attaching more stable GAN models to diffusion process is a valid solution that may provide more consistent lower-bound for the diffusion steps. We will discuss our approach in the section below.

2.3 Model Methodologies

In the literature review and prior experiments related to image generative models, it has been identified that the existing solutions to the problems are not perfect. These solutions may have some inherent drawbacks that restrict their performance in terms of generating high-quality images with good visual fidelity.

To address these issues, a new solution called StyleDiffuser Model has been proposed, which combines the strengths of both GAN and Diffusion Models while mitigating their corresponding limitations. The proposed model architecture is depicted in Fig.2.9, which serves as a basic blueprint for understanding how the model works.

The StyleDiffuser Model aims to overcome the limitations of the traditional GAN-based approaches, which suffer from instability and mode collapse issues, leading to poor quality and unrealistic outputs. Additionally, Diffusion Models also have certain shortcomings, such as style-mixing and training tradeoff, which can limit their scalability and applicability.

The model works by combining a weaker but more consistent GAN model with a stronger Diffusion Model. This is achieved by concatenating the two models, with the GAN model producing "prototype" feature maps instead of final outputs. These feature maps are then used as input for the Diffusion Model along with relevant conditional metadata information, which is proved to be an effective activation for neural networks which is supported [22]. The GAN model essentially serves as a style extractor in this setup.

To balance the excessive training tradeoff, not only different training ap-

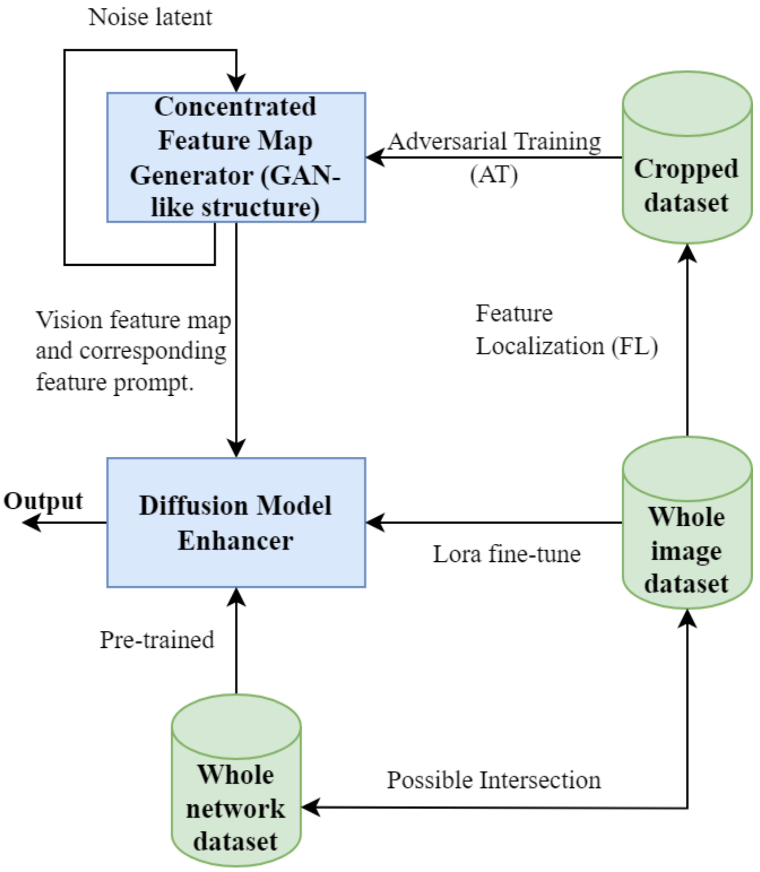

Noise latent
Concentrated Feature Map Generator (GAN-like structure)
Adversarial Training (AT)
Cropped dataset
Vision feature map and corresponding feature prompt.
Feature Localization (FL)
Output
Diffusion Model Enhancer
Lora fine-tune
Whole image dataset
Pre-trained
Whole network dataset
Possible Intersection

Figure 2.9: Basic Structure of StyleDiffuser

proaches that ensure low-cost fine-tuning including Low-rank Approximation[6] and Adaptive Discriminator Augmentation[2] are used. But also the proposed model uses a localized dataset with limited feature coverage for the GAN model and a whole-image dataset with more feature context for the Diffusion Model. For example, for the task of generating Anime Character Faces we are dealing with, the localized dataset is the dataset cropped using YOLOV5[1], while the Whole-image dataset consists of original images before cropping. This asymmetry in model-level knowledge ensures that the stronger model knows more than the weaker model, while the weaker model helps to keep the stronger model focused and consistent. This approach allows for faster convergence during training.

With the StyleDiffuser Model, the authors aim to create a novel approach that overcomes these challenges and delivers state-of-the-art performance in terms of image generation. In the following sections, more detailed explanations and descriptions will be provided for each part of the model, enabling readers to gain a comprehensive understanding of its functionality and design principles.

2.3.1 StyleGAN2 Feature Map Latents

As we have shown in the model architecture in Fig.2.9 above, The first component we would like to introduce here is the GAN-like structured feature map generator based on localized dataset. In reflecting on the essence of Generative Adversarial Networks (GAN), it is important to note that the aim of the generator is to deceive the discriminator using an image generated from a latent vector with dimensions W * H that correspond to the size of the desired image [7]. In the early days of GAN research, most models used random noise as the initial input without any additional operations to make it representative of the characteristics of the image dataset. The Conditional GAN was the first attempt to introduce additional class labels (y) as input, which allowed for some control over general image semantic classes. However, this approach did not involve any style-related information that may be irrelevant to the object classes.

It wasn't until the release of PGGAN [4] that the concept of "latent" was introduced as a trainable transformation process that extracts features from the original dataset. With the inspiration from "style-transfer" [23] and emphasis on the "style latent", StyleGAN was then proposed, which used a brand new architecture to extract and embed style latents between every block to achieve style consistency even among different objects of the same style. Building upon this idea [24], the author suggests that it is possible to treat the entire GAN model as a more compressed view of the portrait of the entire dataset, summarizing all the possible art styles, compositions, etc.

Recent work in the field of Diffusion models has also embraced the con-

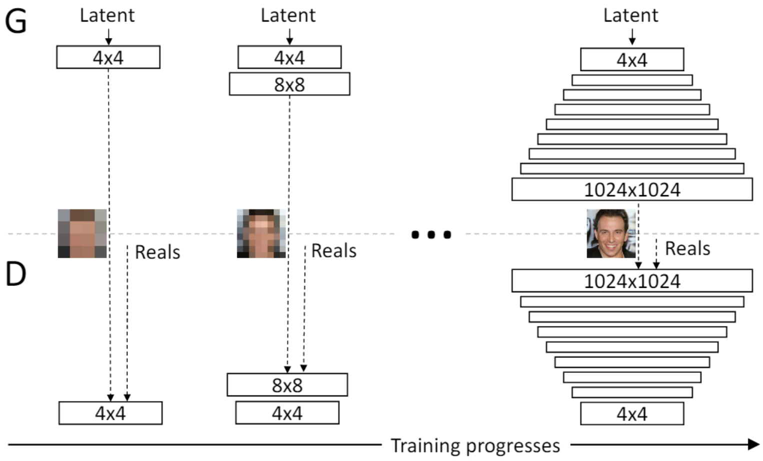


Figure 2.10: PGGAN Latent [4]

cept of latents, making it natural to consider a StyleGAN-latent based diffusion approach. This would involve introducing a diffusion process to the StyleGAN2 model that allows for better modeling of complex data distributions while maintaining control over the desired features. Overall, the development of GAN models has evolved significantly over the years, and the introduction of new concepts such as latent vectors and diffusion processes continue to push the boundaries of what is possible with generative models.

Specifically, the Derivation of the StyleGAN Feature Latents still follows the original GAN-like models, while treating the final output as the Feature Map instead of final outcome. Denote this intermediate prototype as $G(z)$ where G stands for Generator in StyleGAN [5] and z be the noise map with size = $W * H$. The StyleGAN starts with a z and it will be converted into W by nonlinear affine transformation controlled by part of the neural networks,

$$z \to w(w \in W) \tag{3}$$

For simplicity, the dimensions of both spaces are set to 512 and an 8-layer MLP is used to implement the mapping. Note that this is a trainable process determined by θ the parameter of the network. Thus, this W could be customized by such transformation into style information

$$y = (y_s, y_b) \tag{4}$$

which is used after each convolutional layer in the generative network g to control adaptive instance normalization (AdaIN). The AdaIN operation is defined by the following equation. Each feature map x_i is normalized separately and then scaled using the corresponding scalar in y. Therefore, the dimensions of y are twice that of the feature maps in that layer. Unlike style transfer, StyleGAN is calculated from the vector w rather than from style images. Eq.5 and Fig.2.11 below summarized this process.

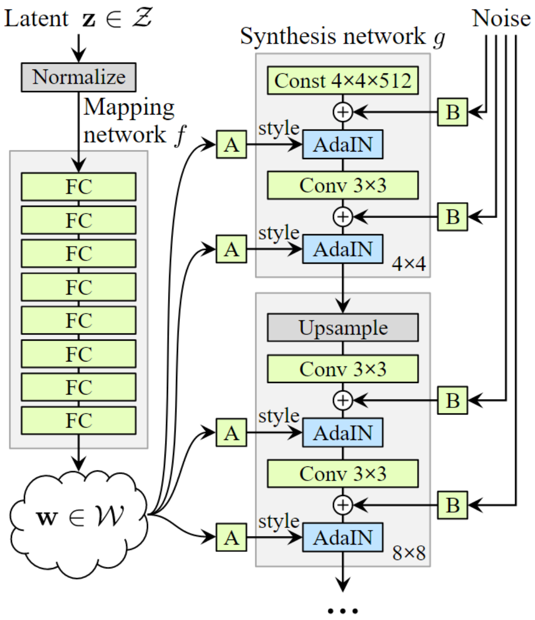


Figure 2.11: StyleGAN Generation [5]

$$\begin{gathered}
F_{\Theta_{affine}}(\mathbf{w}) = \mathbf{y} \in (\mathbf{y_b}, \mathbf{y_s}) \\
\text{AdaIN}\,(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i}\frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i} \\
G(\mathbf{z}) = F_{\Theta}(\mathbf{z}, \mathbf{y})
\end{gathered} \tag{5}$$

Following this style-base generation idea, the StyleGAN2 [3] have made further improvement on the generator mechanism. This model has made further improvements to the generator mechanism, making it more powerful and efficient than its predecessor, StyleGAN [5]. As a result, this project has chosen to use StyleGAN2 as the baseline backbone model for the StyleDiffuser, taking into account factors such as benchmarking and training costs.

Compared to vanilla StyleGAN [5], StyleGAN2's most significant improvement is its revised generator architecture. The AdaIN module between convolutional blocks has been replaced with Weight Demodulation (WD), and random noises B have been relocated outside of the style activation component. These changes can be observed in Fig.2.12 and Fig.2.6. Additionally, the normalization of A no longer calculates the mean value; instead, only the necessary standard deviation is used to scale the weight w for convolution layers. Modulation and demodulation are then conducted based on σ, which is the expected standard deviation of the output activation, as shown in Eq.6. Here, j, k and i enumerate the output and input feature maps correspondingly [3].

Furthermore, StyleGAN2 has incorporated a ResNet-like residual network architecture for progressive growing, inspired by MSGGAN [25]. This architecture maps low-resolution features on the final output via skip-connections. These efforts not only enable the model to produce higher quality images with finer details and textures [3], but also allow it to extract and utilize the style-information present within samples. Therefore, StyleGAN2 has made significant improvements to the style-based image generation process, making it a more robust and efficient model than its predecessor, StyleGAN. Its revised generator architecture, ResNet-like residual network [26], and improved use of style-information make it an ideal choice for the backbone model architecture for the StyleDiffuser project.

$$
\begin{aligned}
\mathbf{w}'_{ijk} &= \mathbf{s}_i \times \mathbf{w}_{ijk} \\
\sigma_j &= \sqrt{(\textstyle\sum_{i,k}(\mathbf{w}'_{ijk})^2)} \\
\mathbf{w}''_{ijk} &= \mathbf{w}'_{ijk}/\sqrt{(\textstyle\sum_{i,k}(\mathbf{w}'_{ijk})^2 + \epsilon)}
\end{aligned}
\tag{6}
$$

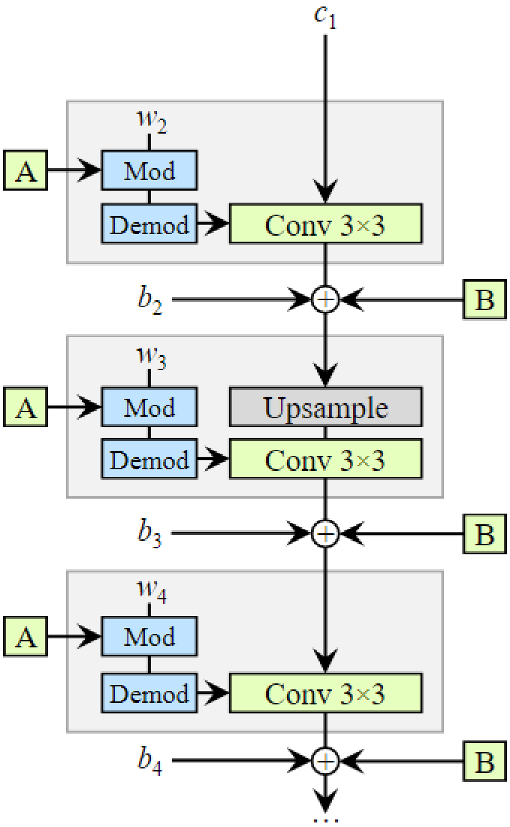


Figure 2.12: Weight Demodulation [3]

As illustrated above, the basic idea of style-based image generation is proposed and further improved by StyleGAN series[5] [3] while there is still difficulties when it comes to implementations on smaller datasets under vanilla StyleGAN training approaches. Apart from the styled Generator, even though there is not specific style-related mechanism designed for discriminators, the discriminator part also need to be customized for this project. With regards to what we have introduced in the part of prior experiment and literature review, Adaptive Discriminator Augmentation(ADA) [2] is implemented for limited datasets, which, in our case, consists of only 1236 samples with limited features without normalization. The structure of ADA network is depicted in Fig.2.7. The basic ideas of ADA is realized by augmentation over the discriminator. Given that the data augmentation deals with over-fitting, the ADA methods introduced 2 overfitting heuristics r_v and r_t where r ranges from 0-1 indicating overfitting probability on validation and training set. r_v, r_t will jointly influence the probability p of augmentation [2]. In that way, the StyleGAN2 assisted with ADA training, could comprehensively express all the valuable patterns with in 1236 image dataset without focusing on some specific features only, and therefore provides fairly good supervision over a stronger model with a more compressed view of real images.

$$\begin{aligned} r_v &= \frac{\mathbb{E}[D_{\text{train}}]-\mathbb{E}[D_{\text{validation}}]}{\mathbb{E}[D_{\text{train}}]-\mathbb{E}[D_{\text{generated}}]} \\ r_t &= \mathbb{E}\left[\operatorname{sign}\left(D_{\text{train}}\right)\right] \end{aligned} \tag{7}$$

2.3.2 Diffusion Enhancer

It is manifest that the StyleDiffuser uses a Diffusion Enhancer (DE) before the final output of the model. Diffusion models have gained significant attention in recent years due to their ability to effectively learn complex data distributions and generate high-quality samples. These models utilise diffusion processes to model data generation, focusing on the transition between noise and data. In this project, we will give justifications on the necessity of the component of StableDiffusion based Diffusion Enhancer discuss the key advancements in diffusion models, specifically Denoising Diffusion Probabilistic Models (DDPM) [8] Stable Diffusion [27], and illustrate the diffusion model implementation in DE component.

Denoising Diffusion Probabilistic Models (DDPM) [8], is a generative model that employs a diffusion process to model the data distribution. This model is built upon the idea of transforming a simple noise distribution into the target data distribution through a series of denoising steps. DDPM utilises a continuous-time stochastic process, known as the diffusion process, to construct a Markov chain that transitions from noise to data. This process is governed by a partial differential equation, the Fokker-Planck equation [28], which describes the time evolution of the probability density of a stochastic process.

Essentially, a diffusion process is a forward process that gradually adds Gaussion Noise modelled by $q(x_t|x_{t_1})$ onto the initial image x_0 into a noise map x_T step-by-step. After all the $x_t, t \in (0, T)$ is derived in forward process, in backward process, which we denoise the noise map, the model aims to fit a model ϵ_θ to a process $p_\theta(x_{t_1}|x_t)$ governed by t and x_t that recovers image step-by-step in a Encoder-Decoder architecture [29] as represented in Eq.8 where α_t is a hyper-parameter called noise schedule and $\epsilon_{t-1} \sim N(0,1)$, $\sigma_t \sim N(0,1)$ represent Gaussian noises in forward and backward process correspondingly. The Eq.9 below shows the loss function for such a DDPM network, which clearly shows that DDPM have a more elegant mathematical form of optimization problem in Deep Learning compared with counterpart generative models like GAN [7] and Variational Auto-Encoder [30]. Due to the advantages brought by these traits, DDPM has been used in various applications, such as image synthesis[31], denoising[32], and inpainting[33].

$$\begin{aligned} x_t &= \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\ x_{t-1} &= \tfrac{1}{\sqrt{\alpha_t}} x_t - \tfrac{\sqrt{1-\alpha_t}}{\sqrt{\alpha_t}} \epsilon_\theta\left(x_t, t\right) + \sigma_t \end{aligned} \tag{8}$$

$$L_{DM} = \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}\left[\left\|\epsilon - \epsilon_\theta\left(x_t, t\right)\right\|_2^2\right] \tag{9}$$

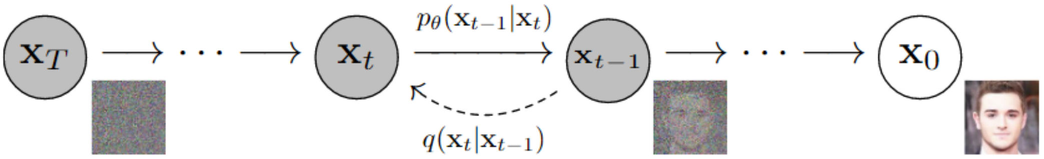

$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_0$

Figure 2.13: Denoising Process [3]

Even though DDPM indicates a new direction for generative models, it is still incapable for down-stream works like text-to-image, image-to-image, etc. which is required for the StyleDiffuser to take inputs. Stable Diffusion [27] is an extension of DDPM that addresses the issue of instability during the generation process trained on a subset of LAION-5B[34]. Stable Diffusion extends the DDPM framework by introducing a latent-space, which uses a Perceptual Image Compression (PIC) mechanism to compress the original images into latent vectors while eliminating some irrelevent high-frequency features. This term ensures that the noise injected during each step of the diffusion process is stable and avoids the accumulation of errors that can occur in DDPM as we summarized in the loss function in Eq.10 i.e. Given images:

$$x \in \mathbb{R}_{H*W*3} \tag{10}$$

The image will first go through a encoder ζ that encodes the image to the latent representation space as shown in Eq.14 below.

$$z = \zeta(x), z \in \mathbb{R}_{H*W*c} \tag{11}$$

The latent generated is then being used to reconstruct a transformation governed by $\mathcal{D}$ in Eq.15 .

$$\tilde{x} = \mathcal{D}(z) = \mathcal{D}(\zeta(x)) \tag{12}$$

Therefore, the idea of using latent space in Stable Diffusion is, to some extent, similar to what we have seen in StyleGAN2 latent representation.

$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x), \epsilon \sim \mathcal{N}(0,1), t}\left[\|\epsilon - \epsilon_\theta\left(z_t, t\right)\|_2^2\right] \tag{13}$$

Stable Diffusion has been shown to improve the quality of samples in image synthesis [27]. Stable Diffusion also supports conditional generation which is crucial in realizing Styled generation in StyleDiffuser which enforces its balance by takes additional supervision of vision feature and metadata information. Stable Diffusion is realized by Conditional Denoising Autoencoder (CDA) $\epsilon_\theta\left(z_t, t, y\right)$ where y is used for image synthesis control where y could be a multi-modal embedding which will go through a Domain Specific Encoder (DSE)τ_θ

that maps different y_{text}, y_{image} and y_{class} into $\tau_\theta(y) \in \mathbb{R}^{M\times d_\tau}$. CDA is eventually implemented into the network by cross-attention [35] in Eq.14 [27].

$$Attention(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)\cdot V, \tag{14}$$

Where Q, K and V are correspondingly, in which $\varphi_i\left(z_t\right) \in \mathbb{R}^{N\times d_\epsilon^i}$ is an intermediate feature representation of Encoder-Decoder U-Net [29] architecture:

$$\begin{aligned} Q &= W_Q^{(i)}\cdot\varphi_i\left(z_t\right), \\ K &= W_K^{(i)}\cdot\tau_\theta(y), \\ V &= W_V^{(i)}\cdot\tau_\theta(y) \end{aligned} \tag{15}$$

Therefore, taking y into consideration, the $LLDM$ can be rewritten as:

$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[\left\|\epsilon-\epsilon_\theta\left(z_t,t,\tau_\theta(y)\right)\right\|_2^2\right] \tag{16}$$

Moreover, There has been state-of-the-art checkpoints pre-trained for specific tasks including Anime style generation developed by industry [27] [36]. However, the checkpoint models are learned from a database which is too big for the model to represent. Therefore resulting inconsistent and chaotic outputs occasionally, it is natural to consider fine-tune and feature constraint to resolve this issue in the StyleDiffuser project's DE component.

Specifically, This project implements an efficient approach to fine-tune the Stable Diffusion baseline based on the collected dataset, Low-Rank Adaptation (LoRA) is a technique developed to address the problem of fine-tuning large language models, such as GPT-3 [37], by reducing the number of parameters that need to be adapted during the fine-tuning process [6]. Although LoRA is primarily applied to large language models, it can potentially be adapted for use in fine-tuning diffusion models, like Stable Diffusion, for improved efficiency.

LoRA works by introducing a low-rank bottleneck in the fully connected layers of the model, significantly reducing the number of parameters needed to be updated during fine-tuning. This allows for more efficient training while maintaining model performance [6]. For each of the selected fully-connected layers that are applicable to adopt LoRA, LoRA introduce a low-rank bottleneck by decomposing the weight matrix into two smaller matrices with dimensions $dim_{input} * rank$ and $rank * dim_{output}$, where dim_{input} is the hidden size of the pre-trained model, dim_{output} is the output size of the layer, and $rank$ is the chosen rank for the low-rank approximation [6]. During the model's forward pass, the original fully connected layer is replaced by the low-rank bottleneck, where the input is first projected onto the low-rank space and then being added back to the original output space [6] as shown in Fig.2.14 below. Therefore, the LoRA is implemented in the diffusion enhancer block for the sake of efficiency and stability such that it takes in the feature map which represents a more focused view over dataset and uses task-specific data to Fine-tune the model, and updates only the low-rank bottleneck parameters instead of the entire weight matrix of the fully connected layers while training. In that way, the diffusion enhance not only absorbs knowledge from the collected dataset but also retains comprehensive knowledge from previously pre-trained massive dataset, making the enhancer could be creative on feature-constrained tasks.

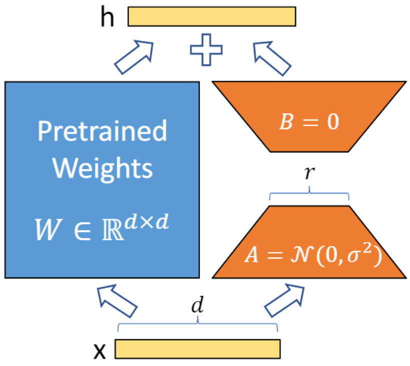

h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

Figure 2.14: LoRA Architecture[6]

To this end, two major components of StyleDiffuser are introduced with corresponding justification based on their theories, while we still haven't covered the connection mechanism between them in detail. Although both of them do have affinities on style related tasks, the depth, latent space and representation of styles of them are asymmetric. Therefore, the project designed a mechanism to make use of such asymmetry for anime character face generation in a more reasonable and consistent way.

2.3.3 Feature Localization, Context and Connection

As depicted in the architecture illustration above in Fig.2.9, the image datasets for 2 components are actually different. Given the asymmetry of the model depth, style latent representation of them, the relative shallower and weaker GAN-based feature generator is trained with image with feature localized by feature "face" by a YOLO-structured object detection model [1], while the diffusion enhancer is trained over uncropped image with full feature coverage. Denote such feature selection localization function as $C_{\theta_{cropper}}$ with input x and query $q_{feature}$, feature map generator as $G_{\theta_{gan}}$, substitution into $Loss_{LDM}$ gives the final formation of the loss function StyleDiffuser model in Eq.17.

$$
\begin{aligned}
X &= F(G(C_{\theta_{cropper}}(x, q_{feature})), q^t_{feature}) \\
L_{LDM} &:= \mathbb{E}_{\mathcal{E}(x), X, \epsilon \sim \mathcal{N}(0,1), t}\left[\|\epsilon - \epsilon_\theta\left(z_t, t, \tau_\theta(X)\right)\|_2^2\right]
\end{aligned}
\tag{17}
$$

Here, X means extracted feature map set from feature localized dataset after with additional query $q^t_{feature}$ embedded. The generator is optimized via ADA mechanism to avoid over-fitting. The z_t is still governed by original image before localization and its diffusion process $q(x_t|x_{t_1})$ fine-tuned by LoRA process.

2.4 Experimental Methods

The experiments is conducted over three dimensions, In the first dimension, the model will be tested for its quality results in generating anime character faces through qualitative evaluation. This will help determine the model's effectiveness and feasibility for the problem stated above.

In the second dimension, the model will be tested on other public test sets to evaluate its performance on different types of dataset and benchmark its improvement in similar tasks. In this dissertation, we will focus on an open animal face dataset, AFHQ[21]. Examine the Fréchet Inception Distance (FID) score [18] over AFHQ datasets, to measure the fidelity of the generated images by comparing the statistical distribution of features extracted from real and generated images.

In the final dimension, a web-based Graphical User Interface (GUI) will be developed to make the generative model accessible and user-friendly for researchers and other users who may not possess specific technical knowledge. The primary aim of this GUI is to provide an intuitive interface that enables users to effortlessly access and operate the model, facilitating the exploration and utilization of its features and capabilities.

To achieve this, the web GUI will be built using a combination of Django[38] and jQuery[39]. Django is a high-level Python web framework that promotes rapid development and clean, pragmatic design. It follows the Model-View-Controller (MVC) architectural pattern, which separates the application's data, user interface, and control logic into independent components. This separation allows for easier maintenance, modification, and scaling of the application. Django provides a wide range of built-in features and tools, including a powerful Object-Relational Mapping (ORM) system, support for user authentication and authorization, and customizable templates for rendering HTML pages.

jQuery, on the other hand, is a popular and lightweight JavaScript library that simplifies various tasks, such as HTML document traversal and manipulation, event handling, and animation. It is designed to make it easier to work with HTML documents and interact with web services through Ajax, a technique that enables asynchronous data exchange between the client and server without the need for page reloads. jQuery's simple syntax and extensive library of plugins streamline the process of creating rich, interactive, and responsive user interfaces.

By leveraging the power of Django and jQuery, the Web GUI will offer an interactive platform where users can easily input data, configure model parameters, initiate model training or inference, and visualize results. This seamless integration of technologies will ensure that the generative model is both accessible and user-friendly, catering to a wide range of users, including researchers, developers, and enthusiasts.

Overall, this experiment aims to evaluate the effectiveness and feasibility of StyleDiffuser through analyzing their performance and usability across different dimensions. It is expected to provide new insights for research in this field and have a positive impact on its future development.

3 | Results And Calculations

The improvements of StyleDiffuser compared with predecessor models will be illustrated in the section below with both qualitative and quantitative evaluations.

3.1 Qualitative Evaluations

Firstly, The StyleDiffuser is tested over the dedicatedly collected and processed dataset. After 12 hour training with 144kimgs, it return excellent image samples as shown in Fig.3.1. To quantify the improvement on image quality. Given that it is hard to construct meaningful inception model for such a limited dataset, the prevailing Fréchet Inception Distance (FID) [18] is not very indicative for the performance of model on this limited task. Thus, the experiments takes out the discriminator component in the Style-feature Map generator as a image score calculator which is trained independently. The score calculator will return a value between 0 and 1 for each image sample to indicate the probability of each image is a real sample or poor patterns depicted by deep learning models. Taking 0.7 as judgement threshold results in the benchmark calculated as illustrated in table below.

Clearly, the model is more consistent than both standalone StyleGAN2 and Stable Diffusion model with higher benign rates, with intuitively better visual effect as shown in Fig.3.2.

Table 3.1: Benign rate over 100 images generated

Model	**StyleGAN2**	**StableDiffusion**	**StyleDiffuser (10 steps, Euler)**
Benign Rates	67%	71%	**80%**

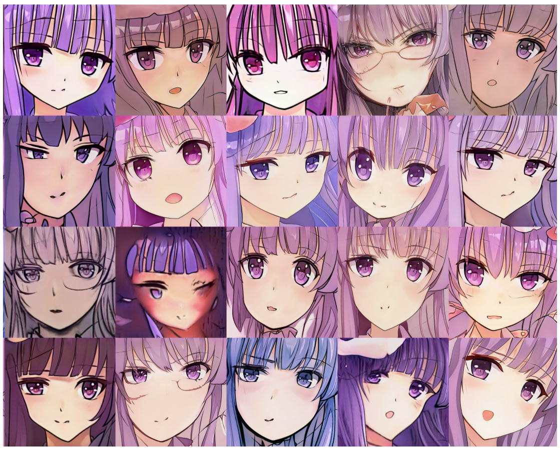


Figure 3.1: Part of images with scores below threshold

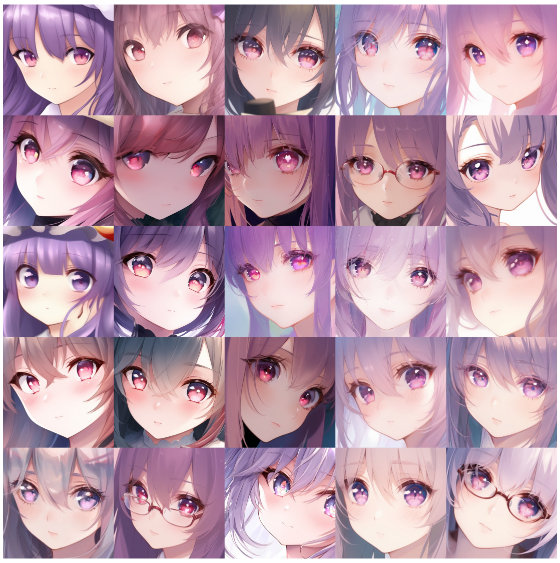


Figure 3.2: StyleDiffuser Outputs

3.2 Quantitative Evaluations

However, for massive datasets consist of more images with more comprehensive data and well-tested inception calculation model baselines, it is possible to calculate the FID score and evaluate the corresponding performance. This experiments takes AFHQ-dog[21] dataset, and calculates the FID following the equation below. After 150kimg training, we have get following FID scores in table below indicating may be a balance found by StyleDiffuser between visual creativity and fidelity in our solution which did not degraded too much even it was learnt from a checkpoint involves too much other dataset.

$$\mathrm{F}ID(x,g) = \|\mu_x - \mu_g\|_2^2 + Tr\left(\sum x + \sum g - 2\left(\sum x \sum g\right)^{\frac{1}{2}}\right) \tag{1}$$

The Inception network [20] plays a vital role in calculating the FID score. Inception networks, also known as Inception v1 or GoogleNet, are a type of deep convolutional neural network (CNN) architecture designed for image recognition and classification tasks. These networks are known for their unique architecture, which incorporates multiple convolutional layers with different kernel sizes in parallel, aiming to capture both local and global patterns in an image. This design allows the network to learn more complex and robust features from the input data.

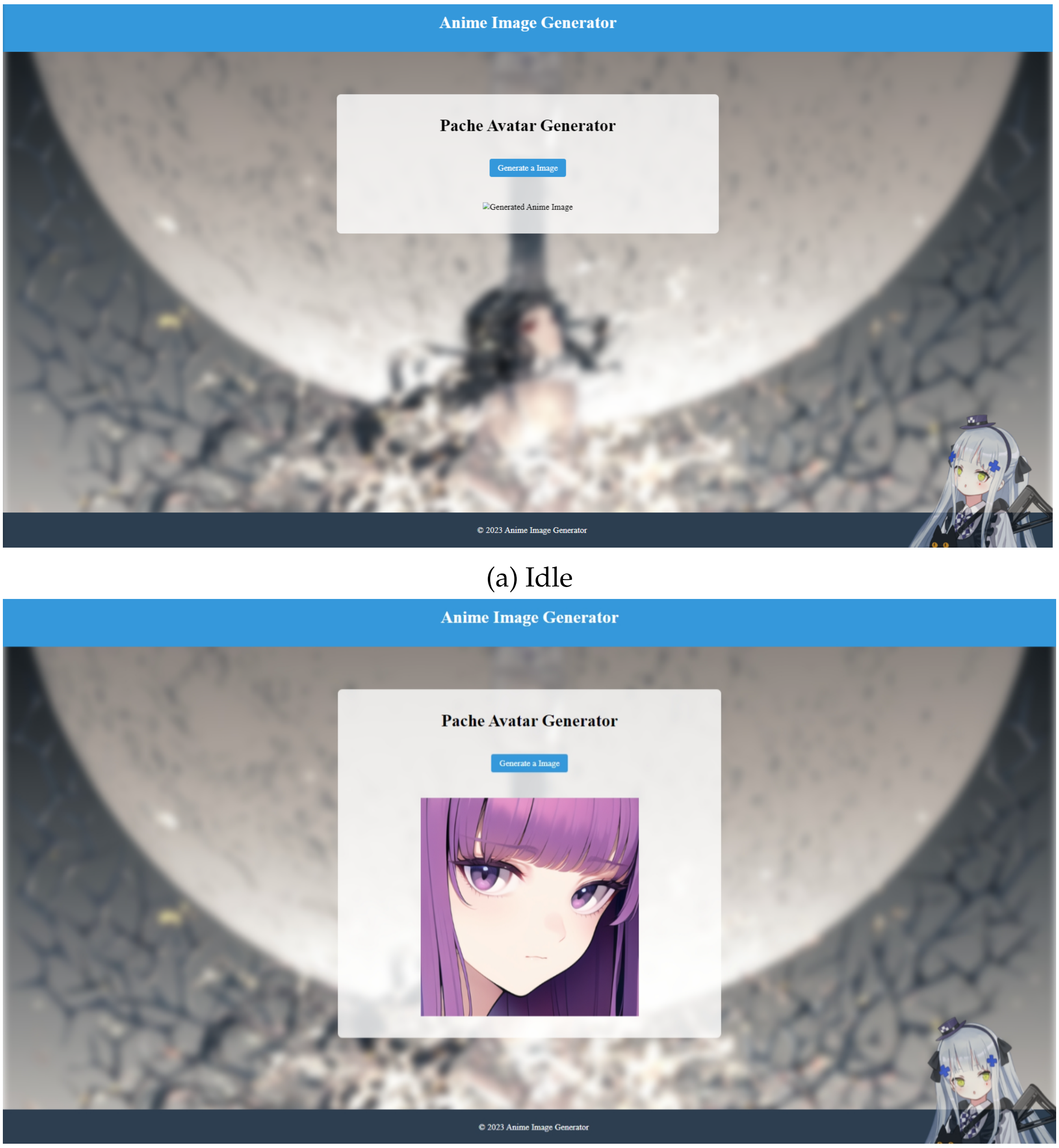
To compute the FID score, features are extracted from an intermediate layer of an Inception network [18]. The distribution of these features is then modeled using a multivariate Gaussian distribution, characterized by its mean (μ) and covariance (σ). By comparing the distributions of features from real and generated images, the FID score quantifies the similarity between them. Lower FID scores indicate a better match between the distributions, suggesting that the generated images are more similar to the real ones and hence, possess higher fidelity[18].

Table 3.2: FID scores over AFHQ-Dog

Model	**StyleGAN2**	**DiffusionGAN [10]**	**StyleDiffuser (10 steps, Euler)**
FID Score	7.40	4.83	**8.52**

3.3 Web GUI

Even though that with the introduction of Style Feature Latent can decrease the computation trade-off, because it can reduce the number of diffusion steps that is needed for diffusion model to converge on output. However, it is still not very applicable to typical CPU-only servers due to excessive memory consumption of diffusion models. Therefore, the real-time Web GUI is based on a buffer-mechanism, in which a remote agent server with GPU generates batches of images that will be delivered to the CPU server via SFTP protocol as a routine. The generation is done by randomly taking images from the batches to ensure its fast reaction and short execution time. As a result, the project have built a highly-usable Web GUI as depicted in Fig.3.3 below.



(a) Idle

(b) Image Generated

Figure 3.3: Web GUI Demo

4 | Discussions and Conclusions

The StyleDiffuser model is designed to combine the strengths of StyleGAN2[3] and Stable Diffusion models, aiming to produce high-quality and visually appealing images while maintaining a balance between creativity and fidelity. In the following discussion, we will delve deeper into the model design ideas that facilitate this combination.

4.1 Discussion

The project have go through all the necessary engineering procedures in Deep-Learning projects, in which consists of dataset construction, prior research on previous models, model design and evolution, and final deployment.

The dataset construction for this project involved several crucial steps, including data scrapping, image cropping, resolution homogenization, and dataset pruning. The data was collected from an open anime illustration database, Danbooru[13], which offers a variety of images with corresponding labels and metadata. By applying filters based on these labels, a limited dataset with consistent styles was created, consisting of 5138 images.

However, the initial dataset needed further refinement. An automatic image cropping algorithm using YOLOv5 [1] was implemented to focus on avatar-only generation. The YOLOv5 network was trained on the Danbooru 2020 anime face dataset and offered a more efficient object detection solution by transforming the problem into a regression problem. Following the cropping process, the images' resolution was homogenized using an Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) [16] approach. This method efficiently reconstructed detailed textures from blurred patterns, and the images were resized to a standard 512x512 resolution. Lastly, the dataset was pruned to eliminate data inconsistency and maintain a more uniform style across the images. A k-means image clustering model[17] was employed to identify and remove minority classes, resulting in a final dataset containing 1236 images with consistent patterns. The dataset construction process involved a series of carefully designed steps to create a high-quality and consistent dataset suitable for training the StyleDiffuser model. This well-crafted dataset played a crucial role in the project's success and the development of an efficient avatar-only generator.

After successful construction of dataset, we have conducted prior experiment of previous models including StyleGAN2. StyleGAN2 is a state-of-the-art generative adversarial network (GAN)[7] that has demonstrated remarkable success in generating high-quality images with impressive diversity. Its key innovations, such as the adaptive instance normalization (AdaIN) and the mapping network, contribute to its ability to generate diverse and visually appealing images. However, one of the limitations of GAN-based approaches is the potential for mode collapse, where the generator only produces a limited set of image samples and fails to cover the entire distribution of the training data.

On the other hand, the Stable Diffusion [27] model is a non-adversarial generative model based on the idea of denoising score matching. It employs a diffusion process to gradually transform an image from the target distribution to a predefined noise distribution, and then reverses the process to generate new samples. The diffusion model is known for its capability to generate high-fidelity images that closely resemble the training data, but it may lack the creativity and diversity offered by GANs.

The StyleDiffuser model aims to integrate the advantages of both StyleGAN2 and Stable Diffusion models, utilizing the creativity and diversity of StyleGAN2 and the high-fidelity image generation of Stable Diffusion models. To achieve this, the model design incorporates the style features extracted from the StyleGAN2 model into the diffusion process, guiding the diffusion model's denoising steps and providing a more creative and diverse set of generated images.

In addition, the model design includes mechanisms to reduce the computational trade-offs often associated with diffusion models. By introducing Style Feature Latent, the number of diffusion steps required for the model to converge on the output can be decreased, leading to a more efficient generation process.

As observed in experiments and corresponding results, the StyleDiffuser model is compared to its predecessor models using both qualitative and quantitative evaluations. Initially, the StyleDiffuser is tested on a dedicated dataset, yielding excellent image samples after 12 hours of training. Since the FID score [18] is not very indicative for this limited dataset, an independent image score calculator is used. The results show that the StyleDiffuser model is more consistent and visually appealing than both standalone StyleGAN2 and Stable Diffusion models.

For larger datasets with more comprehensive data, FID scores can be calculated using an Inception network. The experiment uses the AFHQ-Dog [21] dataset to evaluate the model's performance, with results suggesting that the StyleDiffuser model strikes a balance between visual creativity and fidelity, combining the strengths of both GAN and diffusion models.

However, the model is not easily applicable to CPU-only servers due to the excessive memory consumption of diffusion models. To address this issue, a real-time Web GUI with a buffer-mechanism is implemented, where a remote agent server with a GPU generates batches of images that are delivered to the CPU via SFTP protocol. This ensures fast reaction times and short execution times, resulting in a highly-usable Web GUI.

In summary, as the experiment demonstrates that the StyleDiffuser model provides improved consistency and image quality compared to its predecessors. The StyleDiffuser model integrates the best aspects of both StyleGAN2 and Stable Diffusion models, aiming to create a generative model that balances creativity and fidelity, while also addressing some of the computational challenges associated with diffusion models. This design approach provides a promising direction for future development in the field of generative modeling.

4.1.1 Future Works & Potential Improvements

As demonstrated, the new model's generation capabilities are significantly better than those of standalone GAN and Diffusion models [3] [36]. However, there is still room for further improvements and future work to enhance the model's performance and applicability.

The first direction for future work involves conducting more extensive experiments. Although the Fréchet Inception Distance (FID) [18] score, the most commonly used metric, showed no improvement or even declined when adopting the new model, this indicates that the generated images are less similar to those in the original dataset from the perspective of another Inception model. Despite obtaining strong qualitative experimental results and proving that the StyleDiffuser found a balance between its two predecessor models in terms of creativity and fidelity when tested on the AFHQ-Dog dataset [21], we cannot claim that this model is superior to other existing generative models' approaches with such limited experimentation. More comprehensive experiments across a broader range of datasets, such as CELEB-A[40] and YoutubeFace[41], etc., are still needed to validate the model.

Additionally, future work can focus on investigating the scalability of the model structure by analyzing whether its two major sub-components can be interchanged for other types of tasks. For instance, researchers could explore the use of different GAN feature map generators or alternative diffusion models, or simply apply varying constraints to the asymmetry features. By examining these possibilities, the model's adaptability and performance can be further investigated and even improved, making it a more versatile and robust solution for a wider range of applications in the field of generative models.

In short, the new generative model shows improved performance over standalone GAN and Diffusion models, but further improvements and future work are necessary. More extensive experiments on various datasets are needed to validate the model, and future work can focus on investigating the model's scalability by examining the interchangeability of its major sub-components for different tasks.

4.2 Conclusions

In summary, this project has explored the potential of combining two distinct generative model architectures by concatenating asymmetry feature latent vectors. The project's methodology encompasses an extensive process of data collection, model design, training, testing experiments, and web-application development, which provides ample evidence to support the value and feasibility of this work.

By integrating the strengths of both generative models, the StyleDiffuser model aims to balance creativity and fidelity while generating high-quality images. The novel approach of combining asymmetric feature latent vectors from different models broadens the scope of generative modeling and paves the way for further advancements in the field.

The project's comprehensive experimentation and validation process not only highlights the improvements offered by the StyleDiffuser model but also identifies areas for future work and potential enhancements. Furthermore, the development of a user-friendly web-application demonstrates the practical applicability of the model, making it more accessible to researchers and users without specific technical knowledge.

Overall, this project serves as a testament to the potential of combining different generative model architectures to create more powerful, efficient, and versatile models that can cater to a wide range of applications and use cases.

Acknowledgement

At the end of this Final Year Project report as well as my undergraduate period, I would like to express my sincere gratitude to my supervisor, Dr.Erick Purwanto and Dr.Jianjun Chen, for their invaluable guidance, support, and encouragement throughout my final year project. Their expertise, insights, and patience have been instrumental in shaping this project, and their mentorship has been a source of inspiration and motivation.

I am also deeply thankful to all the lecturers and staff members here at XJTLU, especially my academic advisors, Dr.Yihong Wang and Dr.Jia Wang, for their dedication and commitment to providing an excellent educational experience. Their knowledge and enthusiasm for teaching have contributed significantly to my academic growth and personal development during my time at the university.

In addition, I would like to extend my appreciation to my lab co-workers in SC440 for their collaboration, assistance, and camaraderie in the research environment. Even though it may not be possible to attach a verbose name list for them, their expertise, advice, and willingness to share their knowledge have greatly enriched my learning experience and contributed to the success of this project.

My heartfelt thanks go out to my friends, who have been a constant source of encouragement and camaraderie throughout this journey. Even though I'm never a good friend to spend time with, they can still somehow tolerate my

cynical, mean and hysterical personality. Their friendship, understanding, and willingness to help have made my time at XJTLU truly memorable and rewarding.

I also would like to express my gratitude to my family for their unwavering love, support, and belief in my abilities. Their constant encouragement and faith in my potential have been a driving force behind my achievements and successes if there is any.

As I look back on my four-year journey at university, I have come to appreciate the profound personal growth I have experienced because of the unique culture and atmosphere of my Alma Mater allows student to have freedom to study and thrive, which extends my gain far beyond than just obtaining a bachelor's degree certificate. Admittedly, I have never been a naturally gifted engineering student and have made mistakes and faced numerous challenges in my academic endeavors. Despite doing my best to overcome setbacks and hardships, I have encountered failures that were beyond my control during these four years of engineering study, which nearly destroyed me and left me feeling humiliated and looked down upon.

Although despair and regret may haunt me for a period in the rest of my life, I am determined to persevere with all the scars and a wounded heart, knowing that, ultimately, these experiences have made me a better person.

Continuing my academic journey with regret and stigma is not as terrible as it may seem, especially since I have learned valuable lessons from my failures and have managed to stand up again with the help of those who offered their support. Therefore, I would like to reiterate my heartfelt gratitude to all the people I have mentioned who have helped me.

Bibliography

[1] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” 2020.

[2] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, “Training generative adversarial networks with limited data,” 2020.

[3] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and improving the image quality of stylegan,” 2020.

[4] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation,” 2018.

[5] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” 2019.

[6] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “Lora: Low-rank adaptation of large language models,” 2021.

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” 2014.

[8] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” 2020.

[9] Z. Xiao, K. Kreis, and A. Vahdat, “Tackling the generative learning trilemma with denoising diffusion gans,” 2022.

[10] Z. Wang, H. Zheng, P. He, W. Chen, and M. Zhou, "Diffusion-gan: Training gans with diffusion," 2022.

[11] L. Zhang and M. Agrawala, "Adding conditional control to text-to-image diffusion models," 2023.

[12] C. H. Wu and F. D. la Torre, "Unifying diffusion models' latent space, with applications to cyclediffusion and guidance," 2022.

[13] "Danbooru. anime image board," November 2022, accessed: DATE. [Online]. Available: https://danbooru.donmai.us/

[14] Anonymous, D. community, and G. Branwen, "Danbooru2020: A large-scale crowdsourced and tagged anime illustration dataset," https://www.gwern.net/Danbooru2020, January 2021, accessed: DATE. [Online]. Available: https://www.gwern.net/Danbooru2020

[15] R. Girshick, "Fast r-cnn," 2015.

[16] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. C. Loy, Y. Qiao, and X. Tang, "Esrgan: Enhanced super-resolution generative adversarial networks," 2018.

[17] "Groupimg: A k-means algorithm to separate them in clusters," December 2022, accessed: DATE. [Online]. Available: https://github.com/victorqribeiro/groupImg

[18] 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," 2018.

[19] P. Zhong, Y. Mo, C. Xiao, P. Chen, and C. Zheng, "Rethinking generative mode coverage: A pointwise guaranteed approach," 2019.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014.

[21] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[22] J. Wang, Q. Zhao, D. Lin, E. Purwanto, and K. L. Man, "Conditional metadata embedding data preprocessing method for semantic segmentation," in *2022 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE, 2022, pp. 303–311.

[23] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," 2015.

[24] F. Wang, M. Li, X. Lin, H. Lv, A. G. Schwing, and H. Ji, "Learning to decompose visual features with latent textual prompts," 2022.

[25] A. Karnewar and O. Wang, "Msg-gan: Multi-scale gradients for generative adversarial networks," 2020.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[27] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2021.

[28] H. Risken and H. Risken, *Fokker-planck equation*. Springer, 1996.

[29] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[30] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022.

[31] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," 2021.

[32] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," 01 2008, pp. 1096–1103.

[33] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," 2016.

[34] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, "Laion-5b: An open large-scale dataset for training next generation image-text models," 2022.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[36] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," 2022.

[37] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[38] "Django: The web framework for perfectionists with deadlines." April 2023, accessed: DATE. [Online]. Available: https://github.com/django/django

[39] "jquery: jquery javascript library." April 2023, accessed: DATE. [Online]. Available: https://github.com/jquery/jquery

[40] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[41] L. Wolf, T. Hassner, and I. Maoz, “Face recognition in unconstrained videos with matched background similarity,” in *CVPR 2011*, 2011, pp. 529–534.

A1 | Appendix - Code and Data

- Code: https://www.kaggle.com/datasets/rathgrith/stylediffuser-fyp
- CheckPoints:
  - https://www.kaggle.com/datasets/rathgrith/160klog
  - https://www.kaggle.com/datasets/rathgrith/afhq-pretrained
- Dataset: https://www.kaggle.com/datasets/rathgrith/pache512