Exploring Generative Deep Learning for Character Design

Dhvani Thakkar
C-046
6004220038

Rashi Rana
C-129
6004220175

Vanshika Dhruv
C-171
6004220091

Vruddhi Shah
C-186
6004220215

***Abstract*—The project explores multiple deep learning architectures for multimodal character generation, combining text and image synthesis to create creative assets inspired by fantasy universes such as *Genshin Impact* and *Pokémon*. Using Stable Diffusion fine-tuned with Low-Rank Adaptation (LoRA), the system efficiently generates domain-specific visual styles. In parallel, an LSTM-based model is trained to generate structured textual descriptions that resemble official character bios. Additionally, CycleGAN and StyleGAN2-ADA architectures are employed for style transfer and unconditional image generation respectively, enabling stylistic consistency even with limited datasets. The overall objective is to demonstrate how diffusion-based and generative models can be integrated into a single pipeline for multimodal creative generation.**

I. Introduction

Recent advances in deep learning have revolutionised content generation by bridging the gap between vision and language. Models such as diffusion networks and GANs have demonstrated remarkable capabilities in producing realistic and stylistically coherent visuals, while recurrent networks continue to be strong baselines for structured text generation. Our project aims to explore how these architectures can collaborate to produce original fantasy-style characters both visually and textually. Through a series of experiments involving fine-tuning Stable Diffusion with LoRA, training LSTMs for textual bios, and employing GANs for style transfer, we examine how multimodal systems can mimic the creative process of artists and writers in generating consistent fictional universes.

II. Background

This section reviews the key deep learning architectures and methods that form the foundation of our project.

A. Diffusion Models

Diffusion models are a class of generative models that iteratively denoise random noise to produce high-quality images. They have gained attention for their ability to capture complex data distributions and generate coherent visual details. Stable Diffusion, in particular, combines a diffusion process with a text encoder, allowing text-conditioned image synthesis.

B. Low-Rank Adaptation (LoRA)

LoRA is a parameter-efficient fine-tuning technique that inserts trainable low-rank matrices into specific layers of a pretrained model. Instead of retraining the entire model, LoRA focuses on a small subset of weights, significantly reducing computational cost and memory usage. This approach is ideal for domain adaptation tasks like generating Genshin-style art.

C. Cycle-Consistent Generative Adversarial Networks (CycleGAN)

CycleGANs enable unpaired image-to-image translation by enforcing cycle consistency between two domains. This allows style transfer without the need for paired training data, making it suitable for transforming Pokémon-style art into Genshin-style characters or vice versa.

D. StyleGAN

StyleGAN and its improved variant, StyleGAN2-ADA, generate high-resolution images by mapping latent vectors into detailed, hierarchical styles. ADA (Adaptive Discriminator Augmentation) helps stabilize training on small datasets, a crucial factor in artistic domains with limited samples.

E. Long Short-Term Memory Networks (LSTMs)

LSTMs are a type of recurrent neural network capable of learning long-term dependencies in sequential data. They are commonly used in text generation tasks due to their ability to maintain context across extended sequences. In this project, LSTMs are used to produce structured fantasy-style character descriptions that serve as prompts for image generation.

III. Dataset

Two distinct datasets were utilized:

- one containing *Genshin Impact* character data,
- and another consisting of *Pokémon* character data.

Each dataset contained both visual and textual attributes, allowing the generative models to learn image characteristics and semantic associations.

A. Genshin Impact dataset

We scraped the game's official website to create the dataset. Each entry contained character name, image, element type, weapon class, rarity, and textual descriptions.

B. Pokemon dataset

This dataset was similarly compiled through automated scraping of official Pokémon resources.
Each record contained an image, name, elemental types, and other attributes such as category and weaknesses.

IV. Methodology

A. Genshin Character Image Generation using Diffusion and LoRA

To adapt a pretrained diffusion model to the Genshin Impact character domain, we employed **Low-Rank Adaptation (LoRA)** on the Stable Diffusion v1.5 backbone. The approach fine-tunes only a small subset of parameters—inserted as low-rank adapters—while keeping the base model frozen, resulting in efficient and lightweight domain specialization.

Data Preparation: A custom `GenshinDataset` class was implemented in PyTorch to read entries from the CSV file containing image paths and their corresponding textual prompts. Each image was loaded, converted to RGB, and resized to 512×512 pixels. The prompt served as the conditioning text for the diffusion model.

Model Initialization: The `StableDiffusionPipeline` from the `diffusers` library was initialized with the base checkpoint `runwayml/stable-diffusion-v1-5`.

LoRA Configuration: A `LoraConfig` was defined with the following hyperparameters:

```
r = 4, lora_alpha = 4, lora_dropout
= 0.1, target_modules = ["to_q",
"to_k", "to_v", "to_out.0"], bias =
"none"
```

This configuration injects low-rank adapters into the UNet attention layers responsible for denoising.

Training Process: The fine-tuning loop iterated through up to 200 training steps with a learning rate of 1×10^{-4}. For each step:

1) The input image was encoded into latent space using the VAE encoder and scaled by 0.18215.
2) The text prompt was tokenized and embedded via the CLIP text encoder.
3) Random Gaussian noise was added to the latent representation at a random timestep according to the diffusion scheduler.
4) The UNet, augmented with LoRA layers, predicted the noise component.
5) The **mean squared error (MSE)** between the predicted and true noise was computed as the training loss.
6) Gradients were backpropagated, and weights of the LoRA layers were updated using the Adam optimizer.

Output: Training progress was logged every 20 steps to monitor loss convergence. Upon completion, the LoRA-adapted UNet weights were saved as:

```
genshin_dataset/lora_output/lora_genshin.pt
```

This fine-tuned model could then generate new Genshin-style character portraits when prompted with descriptive text.

B. Character Description Generation using LSTMs

Our objective was to produce creative yet structured textual outputs that resemble official character bios, forming the text backbone for subsequent image generation.

Dataset Preparation: A combined textual corpus was created by merging two sources:

- A curated *fantasy corpus* compiled from various fantasy novels, to capture linguistic structure, grammar, and other elements of fantasy writing.
- A structured dataset of Genshin Impact characters (`characters.csv`), containing fields such as `name`, `element`, `weapon`, `rarity`, `description`, and `prompt`.

Each entry was converted into a standardized text format:

```
Name: [name]
Element: [element]
Weapon: [weapon]
Rarity: [rarity]
Description: [description]
Prompt: [prompt]
```

Model Architecture: We implemented two-layer LSTM network in PyTorch consisting of:

- Character-level embedding layer of size 128.
- Two LSTM layers with a hidden size of 256.
- Fully connected output layer projecting to the vocabulary dimension.

Character-level modeling allowed fine-grained control over text generation and ensured the model could synthesize proper names and stylistic tokens uncommon in word-level models.

Training Strategy: The model was trained in two stages:

1) *Pretraining Phase:* The model was first trained on the fantasy corpus for 50 epochs with a learning rate of 0.003 to acquire general language patterns.
2) *Fine-tuning Phase:* The pretrained weights were then fine-tuned on the Genshin dataset for 300 epochs using a reduced learning rate of 0.001.

The optimization used Adam with cross-entropy loss, trained on character sequences of length 100 sampled from the combined corpus.

Text Generation: After fine-tuning, the model generated novel character descriptions via autoregressive sampling. A temperature-controlled sampling mechanism was employed. Each generation began with the seed string "`Name: `", producing approximately 300 characters of text.

Post-processing: Generated text samples were augmented with randomly assigned attributes (Element, Weapon, Rarity) drawn from predefined lists to maintain structural consistency.

C. Pokemon Style transfer using CycleGAN

The objective of this component was to perform unpaired image-to-image translation between Pokémon types, such as transforming *Water-type* Pokémon into visually consistent *Fire-type* designs. This was implemented using a Cycle-Consistent Generative Adversarial Network (CycleGAN), which allows bidirectional translation between two visual domains without requiring paired examples.

Data Preparation: We divided the Pokémon dataset into subfolders corresponding to elemental types (e.g., Fire, Water, Grass, Electric). Each domain contained approximately 100–200 images resized to 256×256 pixels. The dataset was balanced across type pairs to ensure fair translation performance.

Model Architecture: The CycleGAN framework consists of two generators (G_{XY} and G_{YX}) and two discriminators (D_X and D_Y).

- G_{XY} learns to translate an image from domain X (e.g., Water) to domain Y (e.g., Fire).
- G_{YX} performs the reverse mapping.
- Discriminators D_X and D_Y evaluate the realism of generated images in their respective domains.

Training Objective: Training used the standard CycleGAN loss function, which combines:

- **Adversarial Loss:** Encourages generators to produce images indistinguishable from real samples in the target domain.
- **Cycle Consistency Loss:** Ensures that translating an image to another domain and back reconstructs the original image, i.e.,

$$\mathcal{L}_{cyc}(G_{XY}, G_{YX}) = \mathbb{E}_{x\sim p_{data}(x)}[\|G_{YX}(G_{XY}(x)) - x\|_1].$$

- **Identity Loss:** Encourages color and texture preservation when translating between visually similar types.

Training Details: The model was trained for 100 epochs using the Adam optimizer ($\beta_1 = 0.5$, $\beta_2 = 0.999$) with a learning rate of 2×10^{-4}. Training alternated between generator and discriminator updates for stable convergence. Image augmentations such as random horizontal flips and color jittering were applied to improve generalization.

D. Unconditional Character Generation using StyleGAN2-ADA

To explore unconditional image generation of Genshin-style characters, we employed **StyleGAN2-ADA** (Adaptive Discriminator Augmentation), a GAN architecture designed to perform robustly on limited datasets. The model was trained exclusively on character images from our curated dataset, without any text.

- **Data Preparation:** Images were resized to 512×512 pixels. The dataset size was relatively small (˜93 images), making ADA critical to prevent overfitting.
- **Model Configuration:** StyleGAN2-ADA was trained using default generator and discriminator settings, with adaptive augmentation enabled. Training ran for a few hundred images until visually plausible samples were obtained.

E. Multimodal pipeline: Text-to-Image Character Generation

Our overarching objective was to integrate text and image generation into a single creative workflow. The process was designed as a two-stage multimodal pipeline:

Stage 1 - Text Generation:
An text generator produces new character names and attributes and descriptions.

Stage 2 - Image Generation:
The generated description is used as a prompt for the image generation model, producing a character portrait consistent with the text.

This sequential approach demonstrates how language and vision models can collaborate: text defines concept and lore, while diffusion synthesizes appearance and style. Even though both models are trained independently, they can be connected through prompts to create a unified character creation system.

V. Results

A. Genshin Character Image Generation using Diffusion and LoRA

After training the LoRA-augmented Stable Diffusion model for 200 steps on the Genshin Impact character dataset, the model demonstrated the ability to generate visually coherent and stylistically consistent character portraits.
Manual descriptive prompts were used to qualitatively evaluate generation performance. The model successfully preserved the art style and responded well to text-based conditioning on attributes such as element, weapon type, and rarity.

Example Prompts and Outputs:

> **Prompt 1:** "Barbara, a 4-star Hydro Catalyst from Genshin Impact. Every denizen of Mondstadt adores Barbara. However, she learned the word "idol" from a magazine."
> **Prompt 2:** "Rosaria, a 4-star Cryo Polearm from Genshin Impact. A sister of the church, though you wouldn't know it if it weren't for her attire. Known for her sharp, cold words and manner, she often works alone."
> **Prompt 3:** "Qiqi, a 5-star Cryo Sword from Genshin Impact. An apprentice and herb gatherer at Bubu Pharmacy. An undead with a bone-white complexion, she seldom has much in the way of words or emotion."

B. Character Description Generation using LSTMs

The LSTM model was intended to generate novel character descriptions based on a combination of a general fantasy corpus and Genshin Impact character data. While it learned some structural patterns—such as consistently producing sections labeled `Name`, `Element`, `Weapon`, and `Rarity`—the majority of generated outputs were incoherent or contained nonsensical text.

Sample Outputs:

```
Name: Elevently
Element: Clyor
Description: A denizung able of the
Knights and a bseamber of Favonius.
Prompt: She Collei, a 4-star Anemo
Catalyst from Genshin Impact. A rught
in sening.
Element: Cryo
Weapon: Catalyst
Rarity: 4-star

Name: Reazan
Element: Cryo
Weapon: Catalyst
Rarity: 4-star
Description: A traveler from another
world of Fontaine.
```

Fig. 1. Sample generations from the LoRA fine-tuned diffusion model using manual prompts

```
Prompt: Elethe, a 4-star Electro Sword
from Genshin Impact. One of the Speral
Mighty Pusilat, a is her lagixular. A
pight with sand.
```

Limitations:

- **Incoherent Language:** Most outputs contained non-words or misspellings (*e.g.,* "denizung able," "bseamber," "Clyor"), making the descriptions largely unreadable.
- **Structural Errors:** Some generations merged multiple profiles, repeated attribute labels inconsistently, or ended abruptly.
- **Semantic Failure:** Descriptions often lacked meaningful content or logical narrative; they did not convey plausible character traits.

Fig. 2. Pokemons water to fire

Advantages:

- The model maintained basic field structure, consistently labeling attributes like `Name`, `Element`, and `Weapon`.
- Certain words from the fantasy corpus, such as "Knight," "traveler," or "alchemist," appeared in outputs, showing that the model had learned some relevant vocabulary and stylistic elements, even if used inconsistently.

C. Pokemon Style transfer using CycleGAN

Observations during training:

- Sometimes identical mapping will be learned after a lot of training
- Sometimes colors are mapped indescriminantly (often blue and red in between fire and water)
- Change in learning rate can cause large changes at times

D. Unconditional Character Generation using StyleGAN2-ADA

The model successfully generated new character portraits in the style of Genshin Impact. Unconditional generation allows for diverse outputs that blend costume elements, color schemes, and hairstyles from the training set, though without explicit control over element, weapon, or rarity.

VI. Analysis and Conclusion

The experiments across different generative models reveal both the potential and limitations of multimodal deep learning in character generation.

A. Comparative Analysis

Diffusion + LoRA: The LoRA fine-tuned Stable Diffusion model produced the most visually appealing and stylistically consistent results. It successfully adapted to the Genshin Impact aesthetic using minimal compute and demonstrated effective text-conditioning capabilities.

LSTM Text Generation: While structurally successful in maintaining field labels (Name, Element, Weapon, etc.), the LSTM struggled with grammatical coherence and semantic consistency. This suggests that character-level LSTMs are insufficient for complex narrative generation compared to transformer-based models. LSTM also trained on the fantasy corpus worked much better at grammatical coherence than the one without.

CycleGAN Style Transfer: CycleGAN effectively transferred visual style between Pokémon domains without paired data, demonstrating clear color and texture adaptation between

Fig. 3. Sample character portraits generated by StyleGAN2-ADA

elemental types. However, training instability and occasional color mixing (for example, blending blue and red tones between Fire and Water domains) indicated that fine-tuning learning rates and adding perceptual losses could further improve results.

StyleGAN2-ADA: Despite the limited dataset size, StyleGAN2-ADA successfully generated high-quality Genshin-style characters, thanks to adaptive discriminator augmentation. However, the lack of text-conditioning resulted in uncontrolled attribute generation, limiting practical usability for guided design.

B. Concluding Remarks

Future improvements could involve:

- Replacing LSTMs with transformer-based models (e.g., GPT or T5) for coherent text synthesis.
- Using larger, high-quality datasets to stabilize CycleGAN mappings and StyleGAN generations.
- Integrating cross-modal embeddings to enable tighter coupling between textual and visual representations.

Overall, the results affirm that generative deep learning can meaningfully assist in artistic workflows, enabling scalable, stylistically consistent character design inspired by modern fantasy worlds.

REFERENCES

[1] J. Chen, G. Liu, and X. Chen, "AnimeGAN: A Novel Lightweight GAN for Photo Animation," in *Artificial Intelligence Algorithms and Applications (ISICA 2019)*, Communications in Computer and Information Science, vol. 1205, Springer, Singapore, 2020. doi: 10.1007/978-981-15-5577-0_18.

[2] J. Kleiber, "PokéGAN: Generating Fake Pokémon with a Generative Adversarial Network," *Jovian Blog*, Aug. 10, 2020. [Online]. Available: https://blog.jovian.ai/pokegan-generating-fake-pokemon-with-a-generative-adversarial-network-f540db8154

[3] C. Zhu, "Teaching AI to Generate New Pokémon," *Mage Blog*, Jun. 24, 2021. [Online]. Available: https://m.mage.ai/teaching-ai-to-generate-new-pokemon-7ee0ac02c514

[4] B. Shimanuki, "Joint Generation of Image and Text with GANs," Ph.D. dissertation, Massachusetts Institute of Technology, 2019.

[5] Pokémon Database, "National Pokédex," [Online]. Available: https://pokemondb.net/pokedex/national

[6] HoYoverse, "Genshin Impact Official Website," [Online]. Available: https://genshin.hoyoverse.com/en/