GHIBLI-STYLE FRAME-BY-FRAME ANIMATION GENERATION USING STABLE DIFFUSION WITH LORA FINE-TUNING

Ibrahim Mahmood

May 16, 2025

Abstract

This paper presents an end-to-end system for generating high-quality, Ghibli-style frame-by-frame animations using a fine-tuned Stable Diffusion model. We explore a novel approach that integrates LoRA (Low-Rank Adaptation) fine-tuning techniques on a Ghibli-themed image-caption dataset, ensuring stylistic consistency and low computational overhead. The system uses an LLM-powered narrative breakdown to convert user prompts into coherent animation frame contexts, which are then passed through a hybrid Stable Diffusion pipeline employing text-to-image and image-to-image modules, optionally guided by ControlNet (depth map) to enhance scene structure consistency. We detail the underlying architecture, model training process, and practical considerations, presenting a viable pipeline for personalized, stylized animation creation.

1. Introduction

Traditional animation production is labor-intensive, requiring frame-by-frame illustration and coloring. With advances in generative AI, particularly diffusion models, it has become feasible to automate significant portions of this workflow. However, maintaining stylistic consistency and narrative coherence across frames remains a challenge.

In this project, we aim to generate stylized animations in the visual style of Studio Ghibli using a hybrid pipeline based on Stable Diffusion. To achieve fidelity to the Ghibli style, we fine-tune the model using Low-Rank Adaptation (LoRA), a parameter-efficient method that enables quick adaptation to custom datasets. We also employ an LLM to break down user prompts into coherent sequences of frame-level contexts, each representing a still image to be generated and stitched together into an animation.

2. Related Work

- **Stable Diffusion:** A latent text-to-image diffusion model trained on a large corpus of image-caption pairs (e.g., LAION-5B). It works in a compressed latent space using a VAE, CLIP text encoder, and a UNet denoising backbone.
- **LoRA:** A technique to fine-tune large models using low-rank updates injected into attention layers, drastically reducing memory and computation requirements.
- **ControlNet:** Extends diffusion models with conditioning inputs (e.g., depth maps) for better control and structural consistency.

3. System Architecture

3.1. Prompt-to-Frame Breakdown (LLM Agent)

We utilize `mistralai/Mistral-7B-Instruct-v0.3` via the Hugging Face Inference API to convert user prompts into smooth, coherent animation frame contexts.

Prompt to LLM:

```
"You are a creative animation assistant. Break the given idea into a 5-step animation. Return the re
```

Sample Output:

```
["A ghibli-style dragon perches on a hill.",
 "The ghibli dragon glides over a medieval town...",
 ...]
```

3.2. Text-to-Image (First Frame)

We use the `StableDiffusionPipeline` from the Diffusers library with the base model `runwayml/stable-diffu`

```
pipe = StableDiffusionPipeline.from_pretrained(
    base_model, safety_checker=None, torch_dtype=torch.float16
)
pipe.unet.load_attn_procs(lora_weights_path)
pipe.to("cuda")
image = pipe(prompt=frame[0], num_inference_steps=50).images[0]
```

3.3. Image-to-Image (Subsequent Frames)

For temporal consistency, we use an image-to-image pipeline and optionally guide it with ControlNet:

```
refined = img2img_pipe(
    prompt=frame[i],
    image=prev_image,
    control_image=get_depth_map(prev_image),  # optional
    strength=0.15,
    guidance_scale=8.5,
    num_inference_steps=50
).images[0]
```

3.4. ControlNet for Depth Guidance

To preserve structure, we integrate ControlNet as follows:

```
controlnet = ControlNetModel.from_pretrained(
    "lllyasviel/sd-controlnet-depth", torch_dtype=torch.float16
)
img2img_pipe = StableDiffusionControlNetImg2ImgPipeline.from_pretrained(
    ..., controlnet=controlnet
)
```

4. Dataset and LoRA Fine-Tuning

4.1. Dataset Creation

We collected 50+ Ghibli-style images and captioned them using a combination of manual annotation and BLIP-based models. The dataset CSV follows:

```
file_name,caption
ghibli1.png,"A ghibli-style forest with warm light"
...
```

4.2. Uploading to Hugging Face

```
df = pd.read_csv("metadata.csv")
df["image"] = df["file_name"].apply(lambda fn: os.path.join(image_folder, fn))
ds = Dataset.from_pandas(df).cast_column("image", Image())
ds.push_to_hub("ibrahim7004/lora-ghibli-images")
```

4.3. LoRA Training

We use Hugging Face's training script with Accelerate:

```
accelerate launch train_text_to_image_lora.py \
  --pretrained_model_name_or_path="runwayml/stable-diffusion-v1-5" \
  --dataset_name="ibrahim7004/lora-ghibli-images" \
  --output_dir="./finetune_lora/ghibli" \
  --train_batch_size=1 --max_train_steps=3000
```

Final weights (`pytorch_lora_weights.safetensors`) were pushed to Hugging Face.

5. Output and Animation Assembly

We compile generated frames into animations:

GIF using Pillow:

```
frames[0].save("animation.gif", save_all=True,
               append_images=frames[1:], duration=100, loop=0)
```

MP4 using OpenCV: (alternative)

6. Conclusion and Future Work

This system demonstrates how LoRA-finetuned diffusion models can be used for controllable, stylized animation generation. Future enhancements include:

- Improving prompt-to-frame narrative logic with multi-turn LLM feedback
- Frame interpolation for smoother animation
- Expanding dataset for better generalization

This work provides a blueprint for accessible, stylized animation production using open-source AI tools.

Acknowledgments

This project used models and tools from Hugging Face, Stability AI, OpenChat, and the Diffusers library. We acknowledge the creative vision of Studio Ghibli as the visual inspiration behind the dataset and style.

References

[1] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). *High-Resolution Image Synthesis with Latent Diffusion Models*. arXiv. https://arxiv.org/abs/2112.10752

[2] Hu, E. J., Shen, Y., Wallis, P., et al. (2021). *LoRA: Low-Rank Adaptation of Large Language Models*. arXiv. https://arxiv.org/abs/2106.09685

[3] Zhang, L., & Agrawala, M. (2023). *Adding Conditional Control to Text-to-Image Diffusion Models*. arXiv. https://arxiv.org/abs/2302.05543

[4] Hugging Face Team. *Diffusers Documentation*. https://huggingface.co/docs/diffusers/index

[5] Hugging Face Team. *Accelerate Documentation*. https://huggingface.co/docs/accelerate/index

[6] Li, J., Li, D., Savarese, S., & Hoi, S. (2022). *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation*. arXiv. https://arxiv.org/abs/2201.12086

[7] Hugging Face Team. *Datasets Library Documentation*. https://huggingface.co/docs/datasets/index

[8] Mistral AI. *Mistral-7B-Instruct Model Card*. https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

[9] Clark, A., & Pillow Contributors. *Pillow Documentation*. https://pillow.readthedocs.io/en/stable/

[10] OpenCV.org Contributors. *OpenCV Python Documentation*. https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html

[11] Studio Ghibli. *Studio Ghibli Works – Visual Inspiration*. https://www.ghibli.jp/works/