Project Report

Akhil Agnihotri

1 Introduction

The goal of this challenge is to develop a generative machine learning model capable of generating clothing designs given a text input. These generated designs should be 'realistic and feasible' (similar to the dataset images) to produce. To this extent, below is a road-map and summary of the steps followed to achieve this goal. Additionally, we will also discuss some extensions of our approach should there be no time or resource constraints.

This report has two main parts - Stable Diffusion [Rombach et al., 2022] and CVAE, where the former refers to using Stable Diffusion and the latter refers to using a from-scratch Conditional Variational Auto-Encoder (CVAE) model for the task at hand. Stable Diffusion offers better performance of course, but the CVAE module is shown as a proof-of-concept approach that one could try if faced with a new task such as this. First we will discuss the Stable Diffusion approach.

2 Stable Diffusion

We make use of the Low-Rank Adaptation (LoRA) Hu et al. [2021] technique to finetune the pretrained stable-diffusion-v1-4 model. The key steps can be found under `$ROOT/stable-diffusion/README.md`, which includes information gathered from von Platen et al. [2022] and HuggingFace [2022], and also sample results. The main reason for using this architecture are below:

- Stable Diffusion is a technique that helps improve the stability and quality of image generation in Generative Adversarial Networks (GANs). It's designed to address issues like mode collapse and training instability.
- Stable Diffusion allows for smooth interpolation and manipulation of generated images by controlling the amount of noise introduced at each step. This feature helps in creating variations of images based on input prompts.
- While other models such as AttnGAN [Zhang et al., 2019] and DALL-E [Ramesh et al., 2021] might be more suitable for text-to-image generation task, we could not obtain sustainable and reproducable procedure for finetuning such large models.

Please refer to `$ROOT/stable-diffusion/README.md` file for detailed results. Appendix A.1 contains the model architecture summary.

3 CVAE

For the task of generating images based on text descriptions, a popular choice is to use a CVAE or a GAN-based model with a conditional setup. To this extent, a CVAE model is constructed to generate images `conditioned` on the text prompt. While there are more novel conditioning approaches, in this proof-of-concept, the conditioning is done by just appending the `(image,prompt)` into an `input` tuple for the model.

The key components while implementing this approach were:

- Creating a custom `MainDataLoader` class that works with the fashion dataset, and returns a `(image, prompt)` tuple as an input to the model. This included tokenizing the text using the `transformers` library, and transforming the images based on PyTorch transforms and a `config.json` configuration file.
- Creating a custom CVAE model architecture.
- Creating a custom `train.py` file for training the model

Please refer to `$ROOT/cvae/README.md` file for more details. Appendix A.2 contains the model architecture summary.

4 Possible Extensions

Due to time and resource constraints, the repository is not exhaustive. Some possible extentions are below:

4.1 ControlNetModel / BigGAN / StyleGAN

These additional models can be tested instead of Stable Diffusion. For example, using style transfer or attribute manipulation techniques that StyleGAN provides. This is beneficial since StyleGAN can be finetuned to transfer the style attributes from the given dataset images to generate new images. This can be built on top of the Stable Diffusion model that is previously discussed. This would result in generating a wide range of diverse images, where the user has fine-grained control over the generated images. The user could potentially manipulate specific attributes, such as the age or gender of the model as well.

4.2 User Feedback Loop

One possible extension is to include a feedback loop that integrates human feedback into model training. A few ways of doing so are:

- **Reinforcement Learning:** Use feedback as rewards to update the model's parameters. This approach is similar to the Reinforcement Learning with Human Feedback (RLHF) paradigm in large language models.
- **Fine-tuning:** Fine-tune the model on subsets of the data that receive specific feedback, emphasizing areas where improvement is needed.

A Appendix

A.1 stable-diffusion-v1-4 model configuration


```
    StableDiffusionPipeline {
  "_class_name": "StableDiffusionPipeline",
  "_diffusers_version": "0.21.2",
  "_name_or_path": "runwayml/stable-diffusion-v1-4",
  "feature_extractor": [
    "transformers",
    "CLIPImageProcessor"
  ],
  "requires_safety_checker": true,
  "safety_checker": [
    "stable_diffusion",
    "StableDiffusionSafetyChecker"
  ],
  "scheduler": [
    "diffusers",
    "PNDMScheduler"
  ],
  "text_encoder": [
    "transformers",
    "CLIPTextModel"
  ],
  "tokenizer": [
    "transformers",
    "CLIPTokenizer"
  ],
  "unet": [
    "diffusers",
    "UNet2DConditionModel"
  ],
  "vae": [
    "diffusers",
    "AutoencoderKL"
  ]
}
```


A.2 CVAE model architecture

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1                  [-1, 512]      25,228,416
            Linear-2                    [-1, 8]           4,104
            Linear-3                    [-1, 8]           4,104
            Linear-4                  [-1, 512]           4,608
```

```
            Linear-5                [-1, 49152]       25,210,368
================================================================
Total params: 50,473,248
Trainable params: 50,473,248
Non-trainable params: 0
----------------------------------------------------------------
```

References

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

HuggingFace. How to load image data. `https://huggingface.co/docs/datasets/v2.4.0/en/image_load`, 2022.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019.