Age Domain Translation using Diffusion Models

Dhruv Sharma DHRUV.SHARMA@FAU.DE
MSc. Data Science
Friedrich-Alexander-Universität
Erlangen-Nürnberg, Germany

Editor: Dhruv Sharma

Abstract

This paper describes the use of diffusion models for age manipulation given an input image and edit prompt. Diffusion models are or diffusion probabilistic models are Markov chains trained using variational inference. They can be applied to a variety of tasks, including image denoising, inpainting, super-resolution, and image generation. We present a method to train a pre-trained diffusion model on a custom dataset of human faces with different age groups for the task of age domain translation. Comparison between the images generated from trained model and pre-trained model shows how well the model varies facial features to manipulate the age group and experimental results demonstrate the performance of the model with different training hyperparameters and inference parameters. We also discuss how further improvements can be done to the model.

Keywords: Diffusion Models, Markov Chains, Variational Inference

1. Introduction

Diffusion models are generative models that work on two main principles, the forward noising process where the images are gradually corrupted with noise and the reverse denoising process in which the model learns to recover the images. But a disadvantage of diffusion models is that because of its repeated, sequential nature, the reverse denoising process is slow. In addition, these models work in pixel space, which becomes huge when generating high-resolution images. Therefore, they consume a high amount of memory. Instead of using the actual pixel space, latent diffusion can reduce the memory and compute complexity by applying the diffusion process over a lower dimensional latent space. This is the main difference between standard diffusion and latent diffusion models. The diffusion model pipeline includes a UNet, Variational Auto-Encoder, Scheduler, Tokenizer and a Text Encoder out of which, the UNet and the Auto-Encoder are trained. Images are conditioned on the text embeddings while the U-Net iteratively denoises the latent image representations. A scheduler algorithm computes a latent image representation using the output of the U-Net. After that, the latent image representation is decoded by the decoder part of the variational auto encoder.

2. Method

2.1 Setting up pre-trained CompVis diffusion model

For our task of age domain translation, we will utilize a pre-trained diffusion model capable of generating photo-realistic images given any text input. CompVis

> CompVis (2021) Stable-Diffusion [Source Code]
> https://github.com/CompVis/stable-diffusion

This paper uses CompVis/stable-diffusion-v1-4; hence, the correct model weights for version 1.4 should be downloaded. After setting-up the environment and weights directory, inference can be done easily by the following shell command:

```
python scripts/txt2img.py --prompt "a photograph of an astronaut riding a
horse" --plms
```

However, we need image to image translation for our use case.

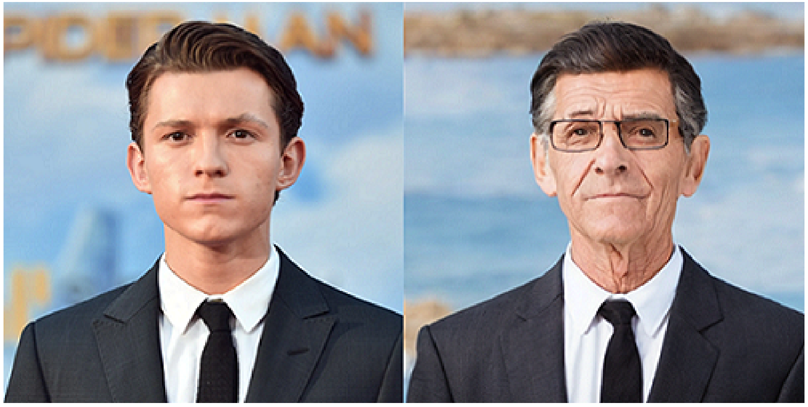
```
python scripts/img2img.py --prompt "A fantasy landscape, trending on
artstation" --init-img <path-to-img.jpg> --strength 0.8
```

2.2 Hugging Face

The above approach would work on machines with a powerful GPU (atleast 10 GB VRAM) but gives Cuda out of memory error for inference. Therefore, working on a local machine or remote server with enough VRAM is recommended. Furthermore, we would also need to modify the scripts to allow for training custom datasets. “Hugging Face” library provides stable diffusion pipelines and custom training methods suitable for our problem. In addition to that, the pre-trained CompVis model is also available via Hugging Face that can be used by specifying the model as “CompVis/stable-diffusion-v1-4” in the HuggingFace pipeline.

> Hugging Face (2023) Text-guided image-to-image
> https://huggingface.co/docs/diffusers/using-diffusers/img2img

Figure 1: Sample inference using CompVis model via Hugging Face

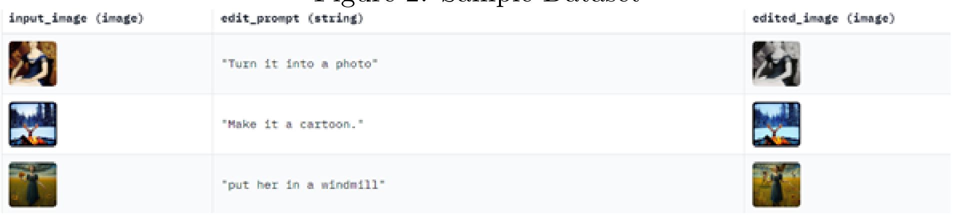


prompt - ”make this male look 50-59 years old”

2.3 Training the model on custom dataset - InstructPix2Pix

InstructPix2Pix is a method to fine-tune text-conditioned diffusion models such that they can follow an edit instruction for an input image. Models fine-tuned using this method take the following as inputs:

Figure 2: Sample Dataset



input_image (image)	edit_prompt (string)	edited_image (image)
	"Turn it into a photo"	
	"Make it a cartoon."	
	"put her in a windmill"	

Do not run accelerate config default command specified in the documentation after setting-up the InstructPix2Pix environment. Doing so will cause errors while training as the flag --mixed_precision=fp16 will fail to load the model on the GPU. Use the accelerate config command (no default) and enter your machine's configuration appropriately. Alternatively, modify the line mixed_precision: 'no' to mixed_precision: fp16 in the default_config.yaml file located at `C:\Users\user_name\.cache\huggingface\accelerate`

Hugging Face (2023) InstructPix2Pix Training
https://huggingface.co/docs/diffusers/training/instructpix2pix

2.4 Dataset

The dataset used to fine-tune the pre-trained model is the FFHQ-Aging-Dataset of human faces with 10 age group classes. Most often, the default download method mentioned in the documentation does not work due to google drive quota limits. Hence, the alternate method using PyDrive should be used. It is important to modify the get_ffhq_aging.sh script and both --pydrive and --cmd_auth flags when working on a remote server.
Command ./get_ffhq_aging.sh may throw a permission error which can be solved by modifying the file permissions in Linux. If the script is crashing abruptly while downloading the dataset, use **ulimit −n 100000** command to change the value of max open files to 100,000.

royorel (2020) FFHQ-Aging-Dataset [Source Code]
https://github.com/royorel/FFHQ-Aging-Dataset

We need to construct a dataset in the format required by InstructPix2Pix with 3 columns i.e. input_image, edit_prompt and output_image. In our approach, we constructed the dataset columns as follows:
output_image – original image from FFHQ-Aging-Dataset
edit_prompt – "make this {gender} look {age_group} years old"
input_image – noised version of the output_image Prafulla Dhariwa (2021)

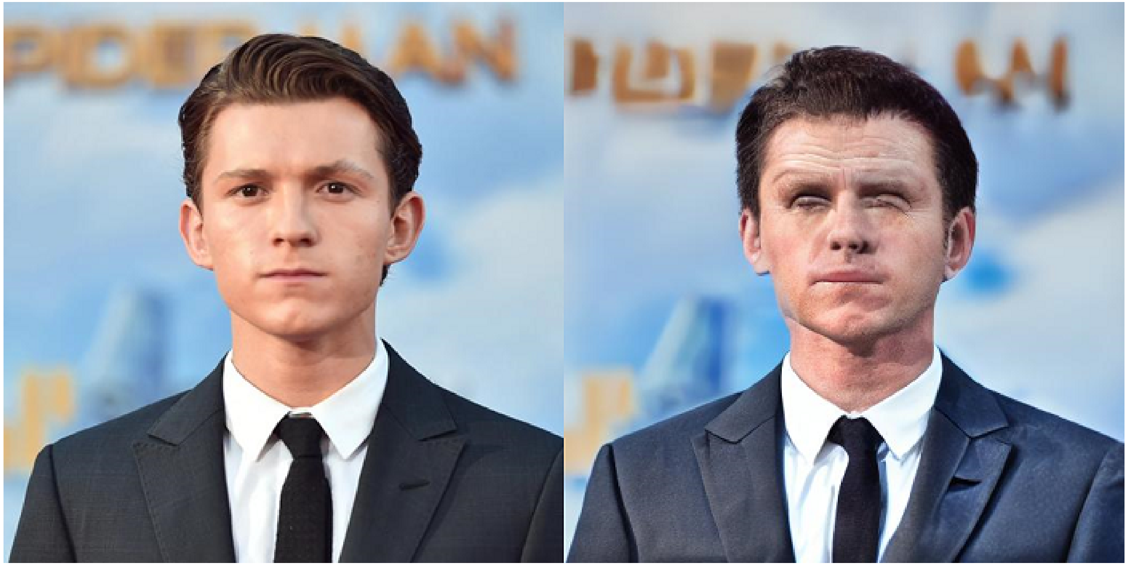
3. Experiments

3.1 Training Parameters

With all the components in place, we can begin training the CompVis model on our custom dataset. Many hyperparameters can be tuned during training but only the important ones are discusses below.

- learning_rate: the default value of 5e-05 gave good results. We tried a higher value of learning_rate but age manipulation results were poor.
- conditioning_dropout_prob: use a value of 0.05 as explained in the InstructPix2Pix paper Tim Brooks (2023). Refer to section 3.2.1
- train_batch_size: higher batch size leads to the leads to fewer variations on the input image during inference. 4, 8 or 16 are possible good values for batch size. Refer to the InstructPix2Pix documentation to find all possible training parameters.

Figure 3: Inference on Trained Model Before Tuning



prompt - ”make this male look 40-49 years old”

3.2 Inference Parameters

The quality of generated images from diffusion models depends not only on the training parameters but also on the inference parameters. A bad choice for inference parameters would produce substandard images even if the training parameters were tuned properly. Another point to remember is to stay within the recommended value ranges of inference parameters. Extreme values lead to nonsensical images or even random noise. guidance_scale, image_guidance_scale, num_inference_steps and negative_prompt are some of the important inference parameters. Refer to the InstructPix2Pix documentation for all inference parameters.

Figure 4: Varying guidance_scale

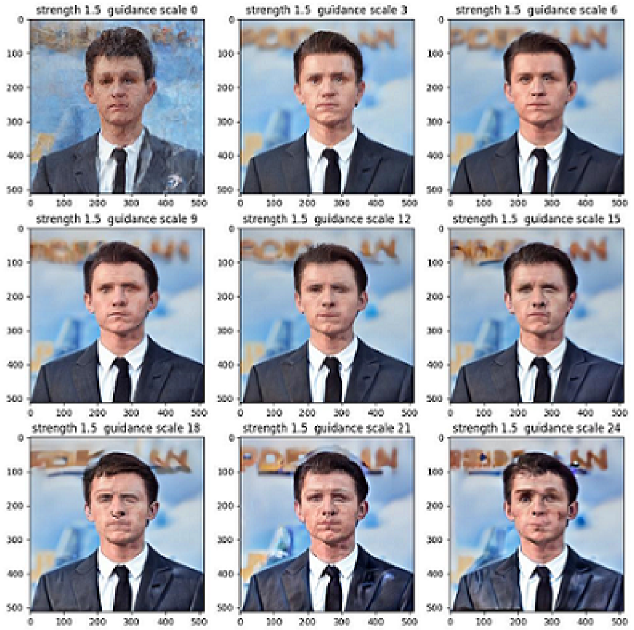
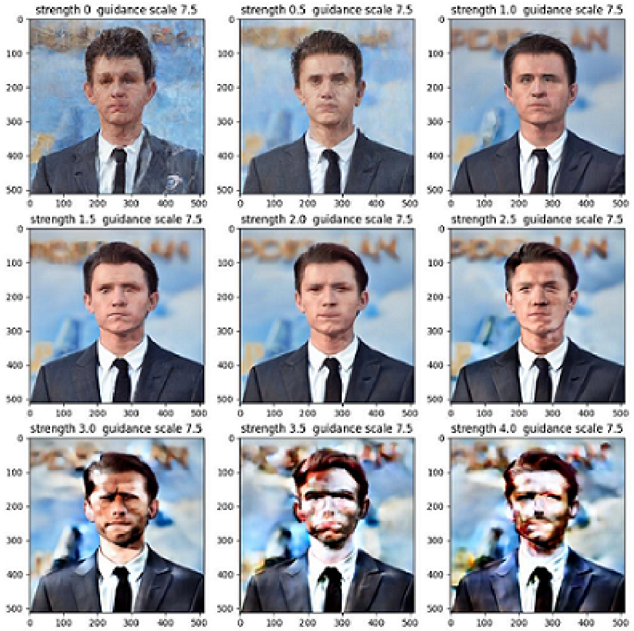

strength 1.5 guidance scale 0
strength 1.5 guidance scale 3
strength 1.5 guidance scale 6
strength 1.5 guidance scale 9
strength 1.5 guidance scale 12
strength 1.5 guidance scale 15
strength 1.5 guidance scale 18
strength 1.5 guidance scale 21
strength 1.5 guidance scale 24
0
100
200
300
400
500

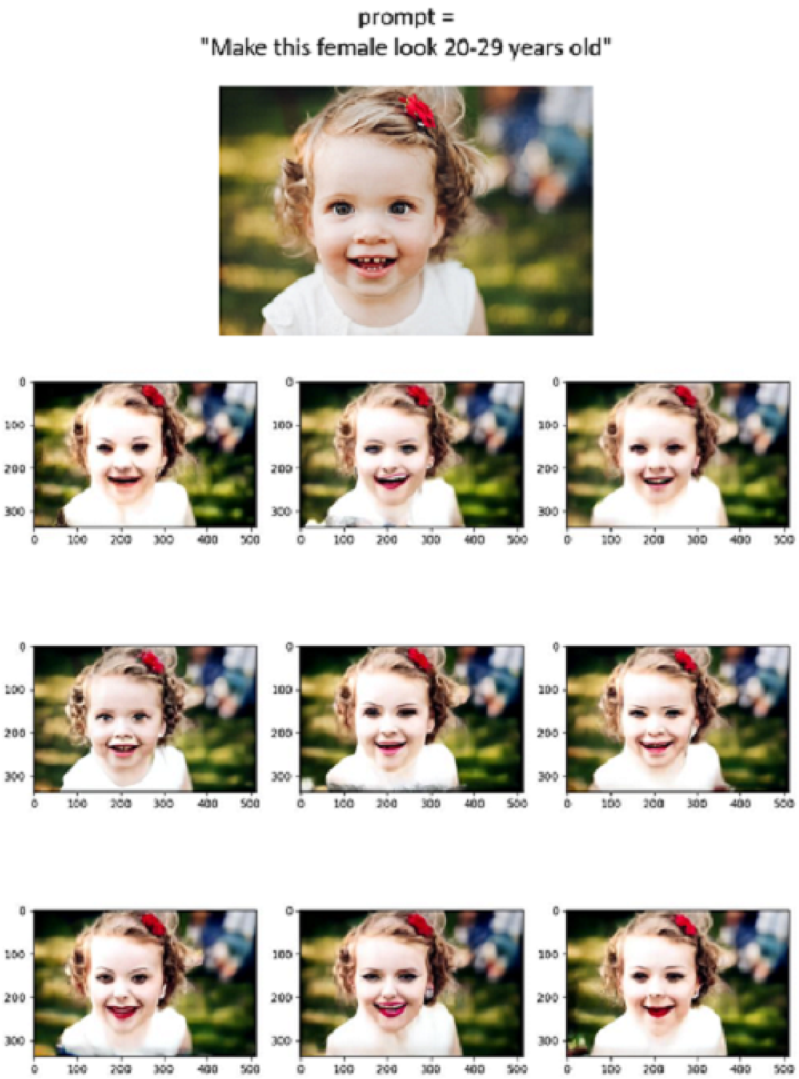
Figure 5: Varying image_guidance_scale


strength 0 guidance scale 7.5
strength 0.5 guidance scale 7.5
strength 1.0 guidance scale 7.5
strength 1.5 guidance scale 7.5
strength 2.0 guidance scale 7.5
strength 2.5 guidance scale 7.5
strength 3.0 guidance scale 7.5
strength 3.5 guidance scale 7.5
strength 4.0 guidance scale 7.5
0
100
200
300
400
500

3.3 After Fine-Tuning Parameters

Apart from the training and inference parameters, different schedulers can also be used in the diffusion model pipeline. DDIMScheduler, LMSDiscreteScheduler, or PNDMScheduler are the possible choices but no one scheduler gave significant better results than the other in our testing. The trained model is now ready for the task for age domain translation. Below are some sample images obtained after selecting suitable values for training and inference parameters.

Figure 6: Sample Images After Tuning



Comparison of generated images from our model to the pre-trained CompVis model.

Figure 7: Our Model

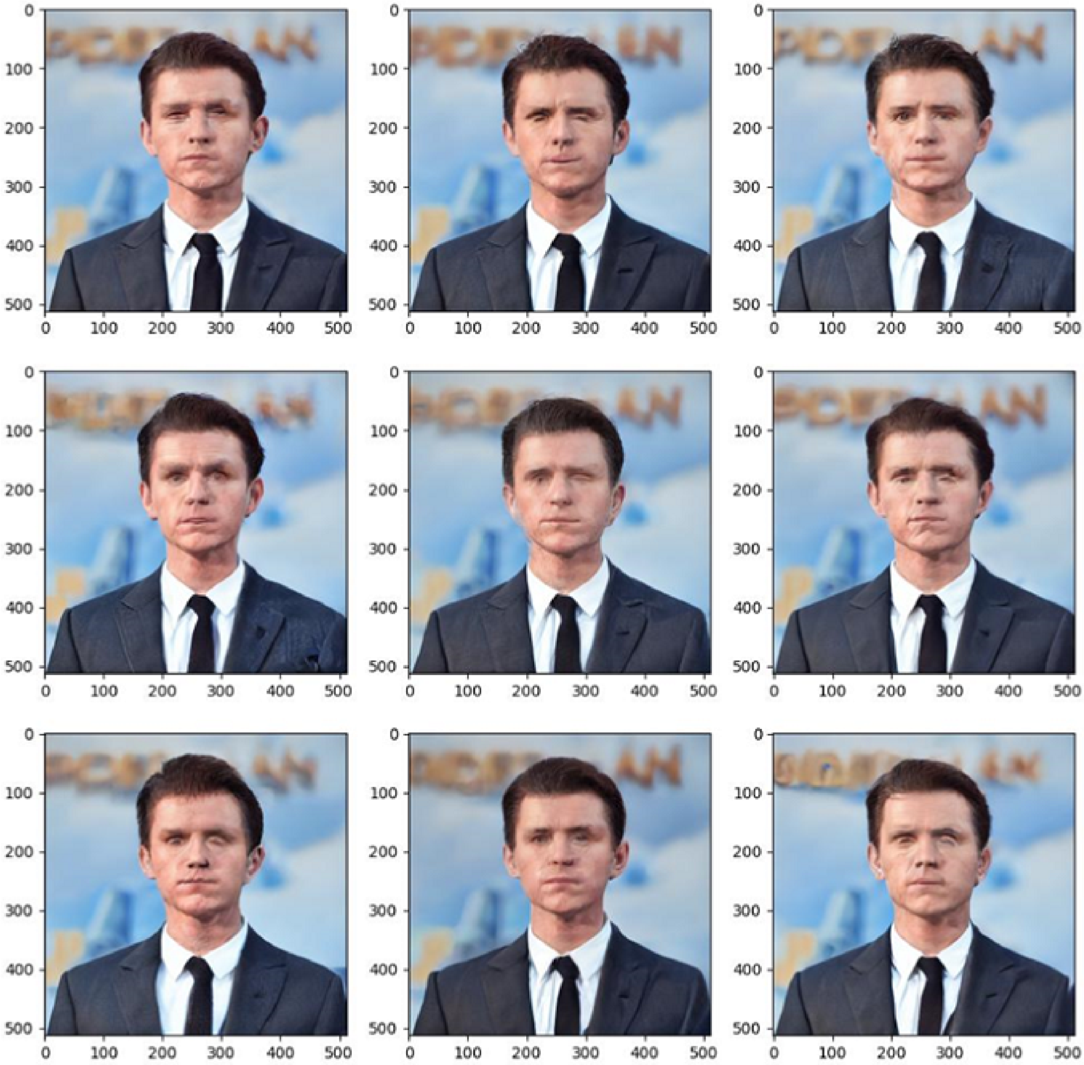
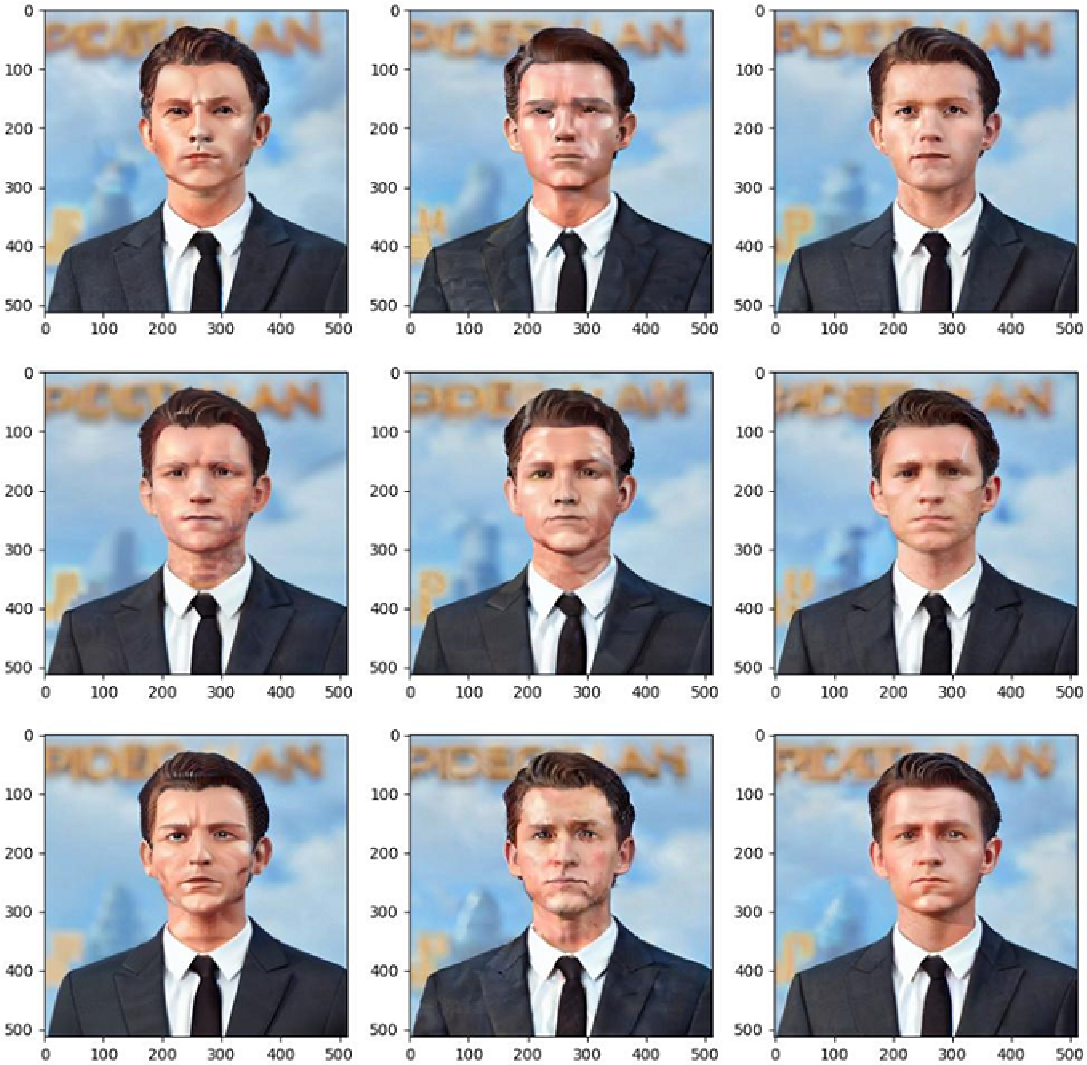


Figure 8: Pre-trained Model



4. Conclusion

As seen from the experiments, our model is now capable of performing age domain translation on any input image of different age groups. It performs better than the CompVis model in terms realism. Images generated from the CompVis model are more cartoonistic with high variations from the input image even for low values of guidance_scale. For high values of guidance_scale, the CompVis model completely swaps the face of the input image, which is not the case in our model. As the model was trained on human faces, it only focuses on variations of the facial features and does not affect other parts of the image. However, around 30% of our model's samples have improper eye features. The performance can be improved by training a conditional diffusion model with only edit_prompt and output_image columns. Another possibility is to try building the custom dataset with different input_image column instead of noised images and train on that to see if there is any significant performance jump.

Acknowledgments

We would like to acknowledge support for this project from Friedrich-Alexander-Universität Erlangen-Nürnberg and the Department Artificial Intelligence in Biomedical Engineering.

Appendix A.

References

CompVis. Stable-diffusion [source code]. URL `https://github.com/CompVis/stable-diffusion`.

Alex Nichol Prafulla Dhariwa. Diffusion models beat gans on image synthesis. *OpenAI*, 2021.

Alexei A. Efros Tim Brooks, Aleksander Holynski. Instructpix2pix: Learning to Follow Image Editing Instructions. *University of California, Berkeley*, 2023.