Mini-Project

Project Link: https://www.youtube.com/shorts/c_3uqdw4MFs
Github Link: https://github.com/22015680/Mini_Project_AI_for_Media/upload
Model Link: https://civitai.com/models/19265/alhaithamgenshinimpact

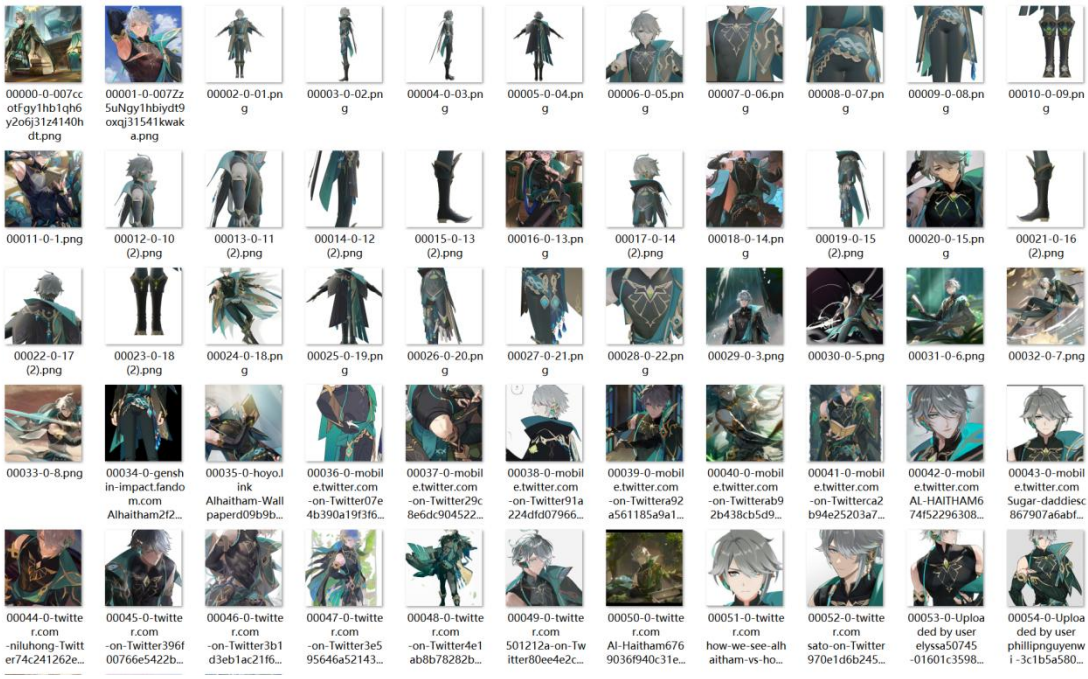
Aim of my project:

In this project, I aim to train an Alhaitham Lora model to generate an animation of the character performing various actions. The animation resembles the 2D anime style while maintaining the realistic 3D rendering of the character. My main objective is to leverage the power of MMD (MikuMikuDance), a popular 3D animation software, along with recent advances in generative modeling such as Stable Diffusion and ControlNet to produce realistic and diverse animations.

Methods used to implement project

To achieve this goal, I first collect a big dataset of Alhaitham. I used pindown and manual saving to extract 1895 images from pinterest/lofter/weibo. Then, I manually removed irrelevant, multi-person, transsexual, low quality, low resolution, real person, Chibi version, comics, black and white images, images lacking character features that differ too much from the official model, and selected images with similar painting style, selecting pictures with white background as much as possible. Next, I downloaded the official model made by Haizi Guan and imported it into blender for a multi-angle and detailed screenshot. After a series of selection, 58 images remained in the database.

In addition, I used dataset-tool to remove the duplicate images and used stable diffusion WebUI auto focal point crop size mode to crop the selected pictrues into 512*512 pixel size, because most of Lora base models were trained by 512*512 pixel size images. This operation ensured consistent image sizes, saved unnecessary training time when processing images and speeded up training.



Samples in the Dataset

When using the WebUI to crop and resize pictures, I still used deepbooru for caption, which was essential for tagging the pictures before training. In the tags of each picture, the character poses and features tags should be removed because it can prevent overfitting by leading such contents included in the base model, to Lora. Therefore, I deleted relevant contents in the tags of each picture. Afterward, I studied the training information of other models, and tried to train my own model. I have trained three models. The training information is below:

Name	Image Count	Repeats	Total Images
5_ALHAITHAM	58	5	290
(Total)	58	5	290

Training parameters — copy to clipboard


```
{
  ss_batch_size_per_device: "1",
  ss_bucket_info: "{"buckets": {"0": {"resolution":
  [512, 512], "count": 290}}, "mean_img_ar_error": 0.0}",
  ss_bucket_no_upscale: "True",
  ss_cache_latents: "True",
  ss_caption_dropout_every_n_epochs: "None",
  ss_caption_dropout_rate: "0",
  ss_caption_tag_dropout_rate: "0",
  ss_clip_skip: "2",
  ss_color_aug: "False",
  ss_enable_bucket: "True",
  ss_epoch: "4",
  ss_face_crop_aug_range: "None",
  ss_flip_aug: "False",
  ss_full_fp16: "False",
  ss_gradient_accumulation_steps: "1",
  ss_gradient_checkpointing: "False",
  ss_keep_tokens: "None",
  ss_learning_rate: "0.0001",
  ss_lowram: "False",
  ss_lr_scheduler: "cosine_with_restarts",
  ss_lr_warmup_steps: "0",
  ss_max_bucket_reso: "1024",
  ss_max_grad_norm: "1.0",
  ss_max_token_length: "225",
  ss_max_train_steps: "1160",
  ss_min_bucket_reso: "256",
  ss_mixed_precision: "fp16",
  ss_network_alpha: "32.0",
  ss_network_dim: "64",
  ss_network_module: "networks.lora",
  ss_new_sd_model_hash:
  "3b26c9c497c923a07ab8d55f2921cf44749535e4b0c890c5c37968e4c
  ss_noise_offset: "0.1",
  ss_num_batches_per_epoch: "290",
  ss_num_epochs: "4",
  ss_num_reg_images: "0",
  ss_num_train_images: "290",
  ss_optimizer: "lion_pytorch.lion_pytorch.Lion",
  ss_output_name: "last",
  ss_prior_loss_weight: "1.0",
  ss_random_crop: "False",
  ss_reg_dataset_dirs: "{}",
  ss_resolution: "(512, 512)",
  ss_sd_model_hash: "210559c6",
  ss_sd_model_name: "model.ckpt",
  ss_sd_scripts_commit_hash:
  "17960c880f56592f0e04c87370d556da8596f7f4",
  ss_seed: "1337",
  ss_session_id: "4135859620",
  ss_shuffle_caption: "True",
  ss_text_encoder_lr: "1e-05",
  ss_total_batch_size: "1",
  ss_training_comment: "None",
  ss_training_finished_at: "1678222991.9683623",
  ss_training_started_at: "1678222336.9036703",
  ss_unet_lr: "0.0001",
  ss_v2: "False"
}
```


Name	Image Count	Repeats	Total Images
6_ALHAITHAM	58	6	348
(Total)	58	6	348

Training parameters — copy to clipboard


```
{
  ss_batch_size_per_device: "1",
  ss_bucket_info: "{"buckets": {"0": {"resolution":
  [512, 512], "count": 348}}, "mean_img_ar_error": 0.0}",
  ss_bucket_no_upscale: "True",
  ss_cache_latents: "True",
  ss_caption_dropout_every_n_epochs: "None",
  ss_caption_dropout_rate: "0",
  ss_caption_tag_dropout_rate: "0",
  ss_clip_skip: "2",
  ss_color_aug: "False",
  ss_enable_bucket: "True",
  ss_epoch: "4",
  ss_face_crop_aug_range: "None",
  ss_flip_aug: "False",
  ss_full_fp16: "False",
  ss_gradient_accumulation_steps: "1",
  ss_gradient_checkpointing: "False",
  ss_keep_tokens: "None",
  ss_learning_rate: "0.0001",
  ss_lowram: "False",
  ss_lr_scheduler: "cosine_with_restarts",
  ss_lr_warmup_steps: "0",
  ss_max_bucket_reso: "1024",
  ss_max_grad_norm: "1.0",
  ss_max_token_length: "225",
  ss_max_train_steps: "1392",
  ss_min_bucket_reso: "256",
  ss_mixed_precision: "fp16",
  ss_network_alpha: "64.0",
  ss_network_dim: "128",
  ss_network_module: "networks.lora",
  ss_new_sd_model_hash:
  "adc3f4dff1f5e4958c28dca99f266ca8b5f9472fc08e7130acb6612cef39
  ss_noise_offset: "None",
  ss_num_batches_per_epoch: "348",
  ss_num_epochs: "4",
  ss_num_reg_images: "0",
  ss_num_train_images: "348",
  ss_optimizer: "bitsandbytes.optim.adamw.AdamW8bit",
  ss_output_name: "last",
  ss_prior_loss_weight: "1.0",
  ss_random_crop: "False",
  ss_reg_dataset_dirs: "{}",
  ss_resolution: "(512, 512)",
  ss_sd_model_hash: "28bb755e",
  ss_sd_model_name: "orangemix3.ckpt",
  ss_sd_scripts_commit_hash:
  "17960c880f56592f0e04c87370d556da8596f7f4",
  ss_seed: "None",
  ss_session_id: "995112495",
  ss_shuffle_caption: "True",
  ss_text_encoder_lr: "1e-05",
  ss_total_batch_size: "1",
  ss_training_comment: "None",
  ss_training_finished_at: "1678248061.3510406",
  ss_training_started_at: "1678247338.686199",
  ss_unet_lr: "0.0001",
  ss_v2: "False"
}
```

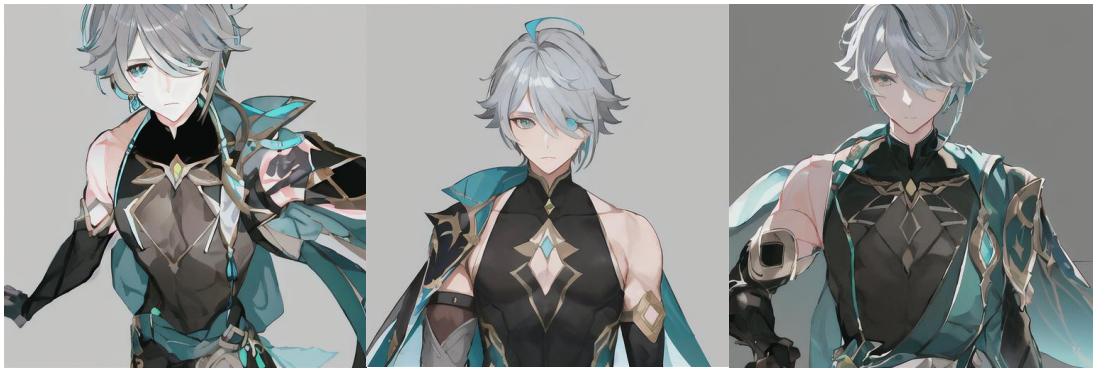

Name	Image Count	Repeats	Total Images
7_ALHAITHAM	58	7	406
(Total)	58	7	406

Training parameters — copy to clipboard


```
{
  ss_batch_size_per_device: "1",
  ss_bucket_info: "{"buckets": {"0": {"resolution":
  [512, 512], "count": 406}}, "mean_img_ar_error": 0.0}",
  ss_bucket_no_upscale: "True",
  ss_cache_latents: "True",
  ss_caption_dropout_every_n_epochs: "None",
  ss_caption_dropout_rate: "0",
  ss_caption_tag_dropout_rate: "0",
  ss_clip_skip: "2",
  ss_color_aug: "False",
  ss_enable_bucket: "True",
  ss_epoch: "4",
  ss_face_crop_aug_range: "None",
  ss_flip_aug: "False",
  ss_full_fp16: "False",
  ss_gradient_accumulation_steps: "1",
  ss_gradient_checkpointing: "False",
  ss_keep_tokens: "None",
  ss_learning_rate: "0.0001",
  ss_lowram: "False",
  ss_lr_scheduler: "cosine_with_restarts",
  ss_lr_warmup_steps: "0",
  ss_max_bucket_reso: "1024",
  ss_max_grad_norm: "1.0",
  ss_max_token_length: "None",
  ss_max_train_steps: "1624",
  ss_min_bucket_reso: "256",
  ss_mixed_precision: "fp16",
  ss_network_alpha: "128.0",
  ss_network_dim: "128",
  ss_network_module: "networks.lora",
  ss_new_sd_model_hash:
  "adc3f4dff1f5e4958c28dca99f266ca8b5f9472fc08e7130acb6612cef3
  ss_noise_offset: "0.1",
  ss_num_batches_per_epoch: "406",
  ss_num_epochs: "4",
  ss_num_reg_images: "0",
  ss_num_train_images: "406",
  ss_optimizer: "bitsandbytes.optim.adamw.AdamW8bit",
  ss_output_name: "alhaitham3",
  ss_prior_loss_weight: "1.0",
  ss_random_crop: "False",
  ss_reg_dataset_dirs: "{}",
  ss_resolution: "(512, 512)",
  ss_sd_model_hash: "28bb755e",
  ss_sd_model_name: "model2.ckpt",
  ss_sd_scripts_commit_hash:
  "17960c880f56592f0e04c87370d556da8596f7f4",
  ss_seed: "1337",
  ss_session_id: "4274931520",
  ss_shuffle_caption: "False",
  ss_text_encoder_lr: "5e-05",
  ss_total_batch_size: "1",
  ss_training_comment: "None",
  ss_training_finished_at: "1678673066.3077123",
  ss_training_started_at: "1678672266.5749204",
  ss_unet_lr: "0.0001",
  ss_v2: "False"
}
```

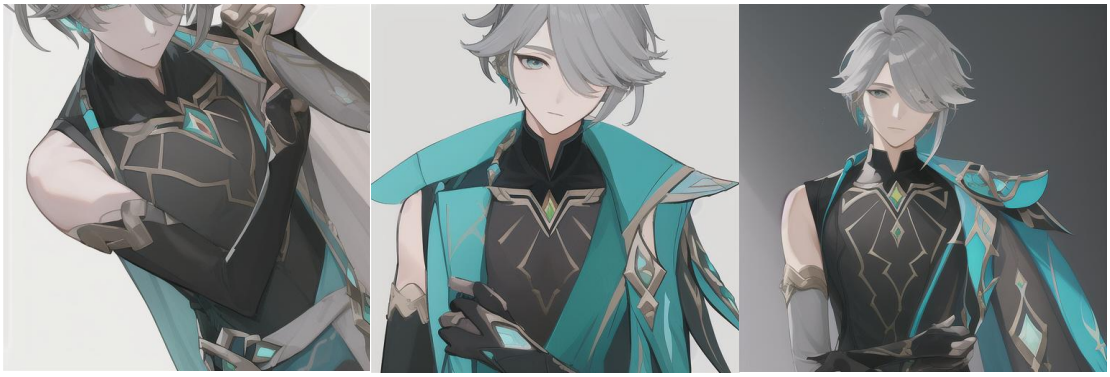

Model 1 Model 2 Model 3

I used the same prompts to test the results of these three model. It turned out to be below:



Examples of First Generation

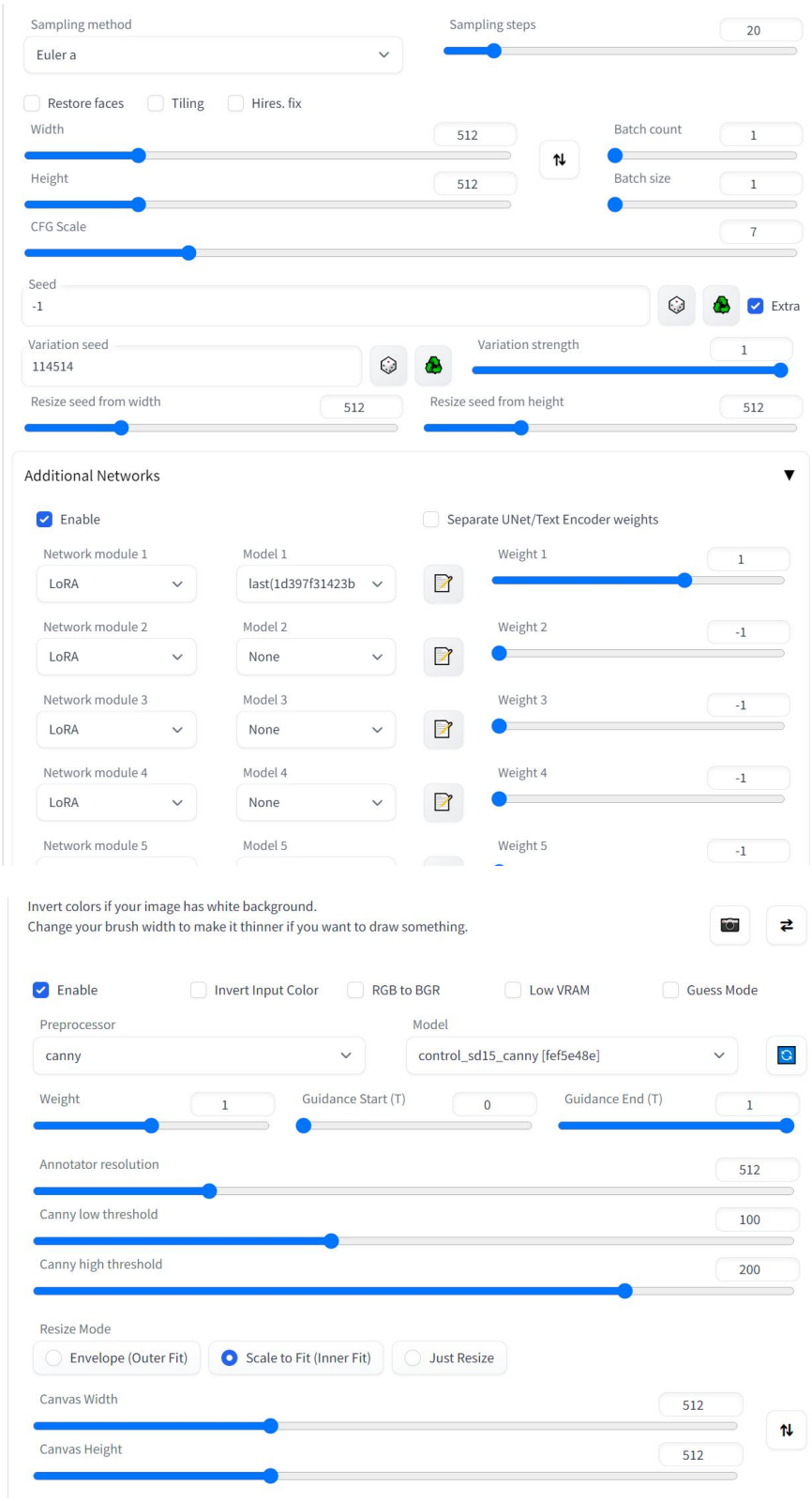
Examples of Second Generation



Examples of Third Generation

From the results, it can be seen that the second generation has the best results. Although in the third generation, I used a higher number of epochs and both network alpha and network dim were up to 128, overfitting occured in the images. With the same keywords input, using the second and third generation models respectively, the images generated by the third generation model often has only clothing details but no face, and the ACGN style line drawing is seriously obvious, so the second generation model wins. Obviously, the accuracy of the first generation is not enough and the resulting images have a clear bad anatomy.

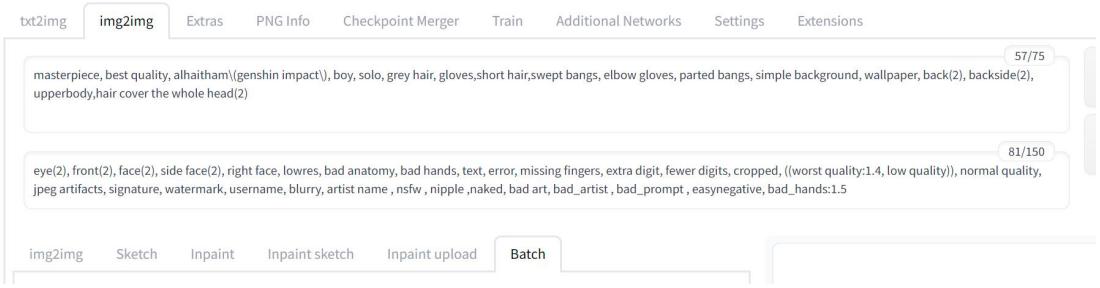
The next step after training Lora is the video output. This requires downloading the character model and the MMD motion file, then exporting a weakly rendered and lighted 3d video from the MMD. I used a video size of 512*512 pixel and a frame rate of 30fps. (I tried a 1024*1024 pixel video with 60fps and found that it took 24h to render, so I used a lower pixel and frame rate.) After that, I imported the video into PR to extract each frame. Afterwards, I rendered each image using the img2img function of the WebUI with Stable Diffusion on board. During the process I added my Lora model and useed the canny model from the ControlNet plugin for image edge detection. After several adjustments of the parameters, the final parameters resulted in the following:



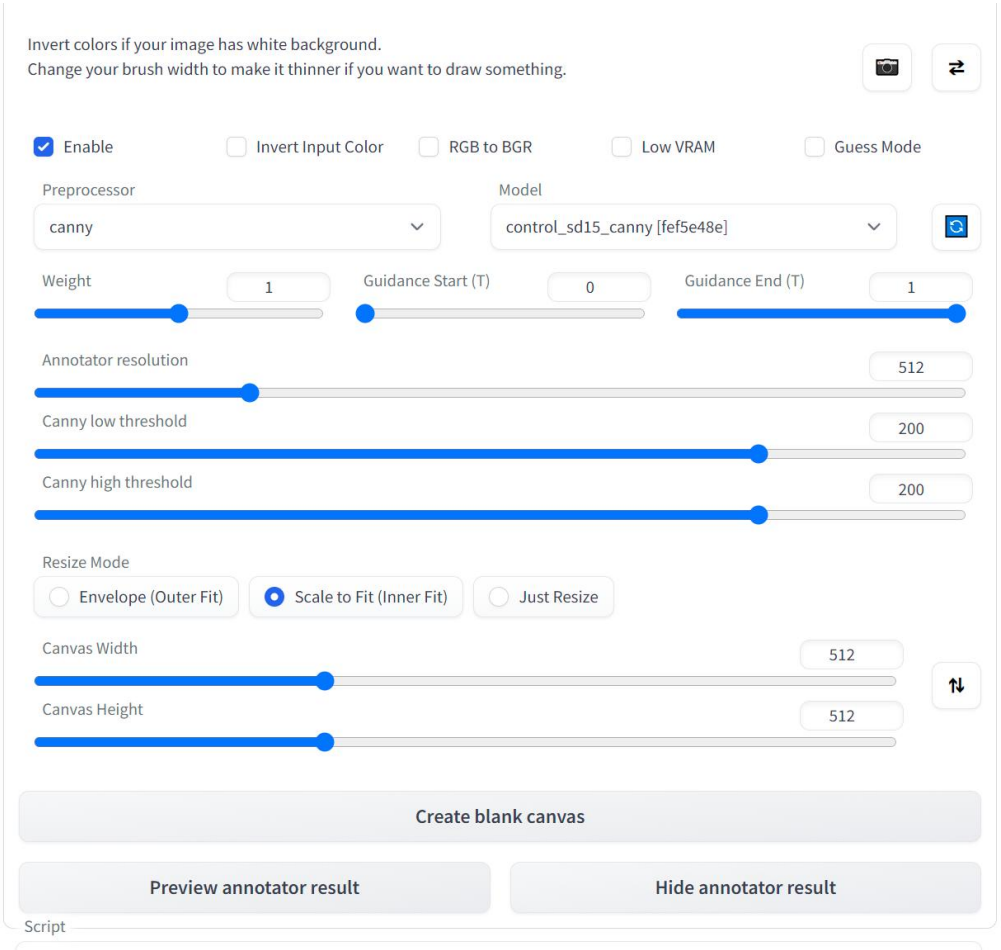
Parameters for AI Picture Generation

Finally, there is a check and adjustment of the images. Because the prompts were for generating the front of the character, there were many keywords about eyes and fringes, which caused the character backside images to appear as head-front creepy images when character spinning.

However, I solved the problem by extracting most of the images in error and resetting the prompts and relevant parameters for the backside of the character.



Backside Prompts



Backside ControlNet settings Changes

Brief evaluation and discussion of results

Overall, the project was successful in achieving its objectives and producing an impressive animation of Alhaitham performing various actions. The project utilized recent advances in generative modeling to achieve its goals and carefully selected a dataset to ensure consistency and prevent overfitting. The resulting animation is a testament to the power of modern generative modeling techniques and provides a sample for further research in this area. However,

the dataset used in the project is relatively small, which may limit the diversity of the generated animations. It would be interesting to see how the model performs with a larger and more diverse dataset.

References to related work cited in the report

Stable Diffusion Local Installation: https://www.youtube.com/watch?v=6MeJKnbv1ts
ControlNet Local Installation: https://www.youtube.com/watch?v=OxFcIv8Gq8o&t=260s
Lora Training: https://www.youtube.com/watch?v=9MT1n97ITaE
https://www.bilibili.com/read/cv22022392?spm_id_from=333.999.0.0
https://www.bilibili.com/video/BV1484y1H7p9/?spm_id_from=333.999.0.0&vd_source=8e11cb890d1d30a41ff0cb17eb13e755
https://www.bilibili.com/video/BV1fs4y1x7p2/?spm_id_from=333.999.0.0&vd_source=8e11cb890d1d30a41ff0cb17eb13e755
ControlNet+MMD Animation:
https://www.bilibili.com/video/BV1Jo4y1v7DR/?spm_id_from=333.999.0.0
Essential Machine Learning Parameters:
https://www.bilibili.com/video/BV1A8411775m/?spm_id_from=333.999.0.0&vd_source=8e11cb890d1d30a41ff0cb17eb13e755