Pencil Tip to Wardrobe Drip: Sketch to Image Apparel Generation

Amrit Khera
Georgia Tech
akhera30@gatech.edu

Mini Jain
Georgia Tech
mjain335@gatech.edu

Puru Malhotra
Georgia Tech
pmalhotra37@gatech.edu

Rachit Chadha
Georgia Tech
rchadha33@gatech.edu

Abstract

In the ever-evolving landscape of fashion design, transitioning sketches into actual garments remains a complex and labor-intensive process. Our project, "Pencil Tip to Wardrobe Drip: Sketch to Image Apparel Generation," seeks to streamline this procedure by developing a system that converts fashion sketches, enhanced by textual descriptions, into realistic images of apparel. This method combines deep learning techniques to interpret and synthesize visual and textual data, thereby providing a tool for rapid prototyping and iterative design modifications. We aim to develop and analyze various multi-modal approaches that enable designers to visualize end products quickly and adjust designs effortlessly, which can significantly accelerate the development cycle in the fashion industry.

1. Introduction/Background/Motivation

We tried to create a system that turns simple sketches of clothes, along with descriptions, into realistic images of apparel. Our goal was to make a tool that helps fashion designers quickly see how their designs might look as real clothes without having to actually make the clothes. This process typically requires extensive collaboration and time, involving multiple iterations from sketch to product. Our automated system can generate high-fidelity images from sketches supplemented with descriptive text and could help them easily change and refine their designs, speeding up the process of developing new fashion items.

Usually, the fashion sketches get translated to actual garments through manual interpretation of the sketches, choosing the right type of fabric, and actually making the garment—either a prototype or the real garment. This is a process consuming lots of time, resources and involvement of designers, illustrators, and pattern makers. The major limitations of this conventional approach include expensive prototyping production costs, slow turnaround from design to production, and inflexibility in making rapid changes based on provided feedback. This can really be a bottleneck in creativity and efficiency, especially for young designers in the industry or the ones experimenting with newfangled designs. Moreover, it is important to note that there is no existing work on the particular fashion apparel generation guided through text and sketch. Hence, the problem is novel.

If successful, this will deliver to designers a powerful tool in the fast prototyping and iteration of clothes designs. This might dramatically decrease the amount of time and, by extension, the amount of money it takes to translate new fashion ideas into the market because the program could quickly convert simple sketches into realistic, detailed pictures. In addition, this may enhance creative liberty as the designer can experiment very wildly, much more than ever before, with a realization of the effect in real-time, hence shortening the general cycle time for developing new designs in the fashion industry. This would be greatly beneficial to the smaller design firms and even individual designers who may not have the capacity to do very detailed physical prototyping.

In the data collection stage of our project, we utilized the FEIDEGGER Dataset, a multimodal corpus of fashion images coupled with textual descriptions. The dataset consists of 8732 high-resolution images, each depicting a dress from the available on the Zalando shop against a white-background along with five textual annotations in German. We wrote a python script to automate the processing of datasets that are to be stored in JSON files. The script read from the FEIDEGGER dataset the URLs and descriptions of the images. The images were downloaded from the given URL, stored in a folder and all the images were passed through a canny edge detection to get a corresponding sketch image. After this, the instructions, which were primarily given in German were converted to English using the Google translate API. This was done to ensure the language consistency of the dataset. To map the images to the right description, we created a JSON with Instruction and Image name keys. This preprocessing step was automated in a way that allowed the rest of our data to be easily prepared in the context of the following deep learning stages of our project.

Our project builds on existing methods in the image generation and manipulation domains. Several works have treated the problem of producing realistic images from sketches, most notably in the context of faces and generic objects. For example, the Pix2pix framework has demonstrated great potential for transforming sketches into photorealistic images using conditional adversarial networks. Also, the fashion industry has tools such as VITON and CP-VTON, which have been developed so that they can allow virtual try-on of clothes by mapping clothes onto posed human images and combine aspects of shape and texture understanding.

Our approach wants to generalize these technologies to include textual description and sketch integration for images with more and accurate details. This is the dual-input strategy: inspired by advances in multimodal learning, where models like CLIP demonstrate promising results for understanding and generating content that accurately reflects combined textual and visual inputs. These include material texture characterization, which is necessary for the accurate representation of a model in the world of fashion and the chosen style.

2. Approach

To solve the problem of generating realistic fashion apparel images guided by sketch and text we apply pipeline and one-shot approaches. The pipeline approach consists of GANs first generating an image based solely on the sketch and then applying text guided manipulation of the image. On the other hand the one-shot approach delves into the domain of multi-modal Stable Diffusion models which generate the expected image taking both the sketch and text as input. We also generate baseline results using GANs and Stable Diffusion models.
We expect multimodal models both the pipeline and one shot approaches to perform better than the baseline GANs as adding additional modality enriches the model with more information and ensures the output is guided by a multi faceted approach. Moreover, we expect the Stable Diffusion based one shot model to perform better than the pipeline GAN approach as stable diffusion has shown to be more diverse than GANs and adding conditional control further increases the robustness of the model.

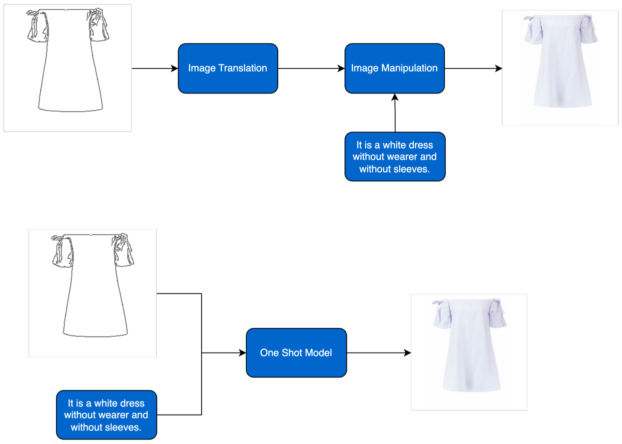


Image Translation
Image Manipulation
It is a white dress without wearer and without sleeves.
It is a white dress without wearer and without sleeves.
One Shot Model

Figure 1. Pipeline and One-Shot Approach

2.1. Pipeline Methods

The pipeline approach employs TediGAN [8], which is particularly suited for text-guided image generation. This model enhances the capability of GANs by incorporating textual descriptions along with sketches, providing a richer context and allowing for more detailed and specific generation based on the text input. By using TediGAN, we aim to achieve higher fidelity in the generated images that are more aligned with the design intentions expressed in the sketches and accompanying descriptions. This method should theoretically yield better results than image-only Pix2Pix [2] or CycleGAN [10] models by effectively utilizing the additional text input to guide the image synthesis process more precisely.

The first step of the TediGAN architecture is the inversion of a pretrained StyleGAN to train an image encoder that can map an unseen image to the latent space of a fixed StyleGAN model pretrained on a custom dataset. Once the inversion module is trained, given a real image, we can map it into the W space of StyleGAN. The next step is then training a text encoder that learns the associations and alignments between image and text for manipulating the image generated by the inverted StyleGAN according to the input text.

To use TediGAN for our specific task, we fine-tuned a pre-trained StyleGAN model from the official StyleGAN repository [3], making changes to incorporate our custom dataset and fine-tune the model. This was done keeping in mind the limited size of our custom fashion dataset as well as limited computational resources for training the StyleGAN on Colab and PACE. We fine-tuned the facial StyleGAN model on our dataset consisting of fashion images. We then inverted the StyleGAN model and trained the text encoder, using (Contrastive Language-Image Pre-Training) [4]. The model was adapted to better integrate and interpret the combined inputs, ensuring that the generated images re-

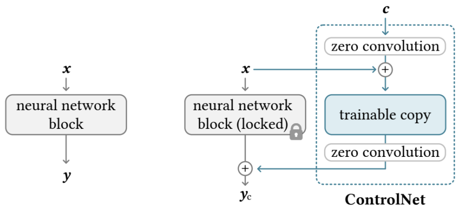


Figure 2. Modifying Neural Network Block

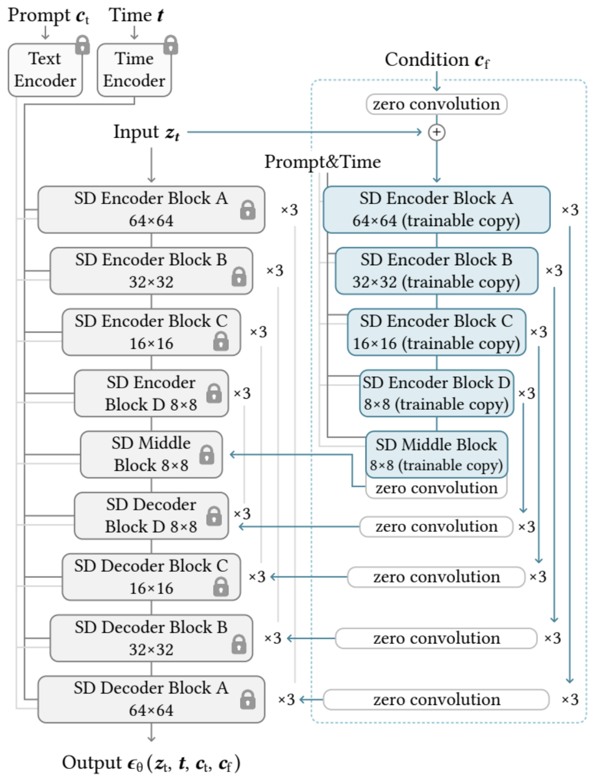


Figure 3. One Shot Approach: Controlling Stable Diffusion

flect both the visual and textual data accurately.

2.2. One Shot Methods

The one shot approach leverages a Stable Diffusion model known for its robustness in generating high-quality images from text descriptions [6]. We adapt this model using ControlNet to add multimodal input in terms of spatial conditioning to text to image diffusion models [9]. This setup is expected to surpass the capabilities of TediGAN and Pix2Pix by better integrating and balancing the visual and textual information, thus producing more accurate and detailed images. The incorporation of Stable Diffusion in our pipeline represents the most advanced application of multimodal input in our project, aiming to set a new standard for sketch-to-image generation in fashion design.

Stable Diffusion utilizes latent diffusion which performs the diffusion step efficiently in the lower dimensional latent image space. These text-to-image models achieve SOTA performance by incorporating text encoding models such as CLIP [5].

To add multimodal input in terms of sketch in addition to text we add additional conditions in the neural network block effectively creating a ControlNet as shown in Figure 2. This is done by creating a copy of the original weights which are locked and adding zero convolutions denoted by Z to add the conditions back into the architecture. The condition is passed into the copy which is trainable. This architecture preserves the original capabilities of the model as well as induces new controls robustly in the model. The new output of the model is given as:

$$y_c = F(x;\theta) + Z(F(x + Z(c;\theta_{z1});\theta_c);\theta_{z2}) \qquad (1)$$

Here, F denotes the original block, θ_c the cloned weights and θ_{z1}, θ_{z2} the weights of the zero convolution layers. The controllable stable diffusion architecture is shown in Figure 3. Here, the added control is in the form of the sketch of the fashion apparel.

We implement the changes to the stable diffusion models to incorporate the ControlNet architecture for training the model on our custom dataset leveraging the ControlNet codebase. We make changes to incorporate the control in the form of the sketch of the fashion apparel. Then we implemented training and performed hyper parameter tuning using a distributed training process described in experiments in section 3. Finally, we implemented the inference logic to generate images on the train and test set to perform evaluation using the implemented FID module and human evaluation.

2.3. Challenges

We anticipated challenges with the alignment of sketches and text descriptions, ensuring that the generated images accurately reflect the details conveyed in both input types. We also anticipated edge detection failing sometimes to capture the intricate details of the designs inside a dress. Additionally, we expected difficulties in model training, particularly in achieving convergence, given the complex and varied nature of fashion data. During the implementation, we faced issues with the quality of image generation, where initial outputs were not as detailed or accurate as required. The integration of text inputs with visual sketches did not initially produce the expected clarity in the design details. The initial models, particularly the early versions of our GAN implementation, did not perform adequately, necessitating several iterations of tuning and adjustments. It was only after refining our approach, enhancing our data preprocessing techniques, and integrating more sophisticated text-processing mechanisms that we began to see improvements.

3. Experimental Setup and Results

3.1. Loss Function

In the implementation of TediGAN, for training the image encoder for the inverted StyleGAN model we use image generation losses such as perceptual loss and LPIPS as we are using real images instead of the fake synthesized images.

$$\min_{\Theta_{D_v}} \mathcal{L}_{D_v} = \mathbb{E}\left[D_v\left(G\left(E_v(\mathbf{x})\right)\right)\right] - \mathbb{E}\left[D_v(\mathbf{x})\right] + \frac{\lambda_3}{2}\mathbb{E}\left[\|\nabla_{\mathbf{x}} D_v(\mathbf{x})\|_2^2\right] \tag{2}$$

To train the text encoder that learns the associations and alignments between image and text, we minimize the mean squared loss between the obtained image embedding and text embedding as given below.

$$\min_{\Theta_{E_l}} \mathcal{L}_{E_l} = \left\| \sum_{i=1}^{L} p_i \left(\mathbf{w}_i^v - \mathbf{w}_i^l\right) \right\|_2^2 \tag{3}$$

Stable diffusion models add noise to an input image z_0 iteratively to produce a noisy image z_t at time t. Image diffusion algorithms further learn a model to predict noise at time t given the conditional text c_t and task specific conditions c_f. The loss equation is given below which computes the difference between the original and predicted noise from the image diffusion model.

$$L = \mathbb{E}_{z_0,t,c_t,c_f,\epsilon\sim N(0,1)}\left[\|\epsilon - \epsilon_\theta(z_t,t,c_t,c_f)\|_2^2\right] \tag{4}$$

3.2. Evaluation Metric

Quantitative Comparison. The quality of generated or manipulated images is evaluated through Frechet Inception Distance (FID) [1]. The metric is usually employed in the context of GANs and computes the feature vector distances between generated and real images. The FID scores for the generated fashion apparel inform us about the performance of the models in terms of their ability to generate realistic and diverse apparel that are guided towards the expected image via text and sketch. Our primary focus of the metric is its ability to quantitatively define the generated output and its comparison with the expected output.

Qualititative Comparison. Apart from FID scores we also aim to employ basic Human evaluation as a measure of qualitatively ascertaining how good and appealing the generated images are and how well the model is able to make use of the sketch and text guidance to generate the expected output.

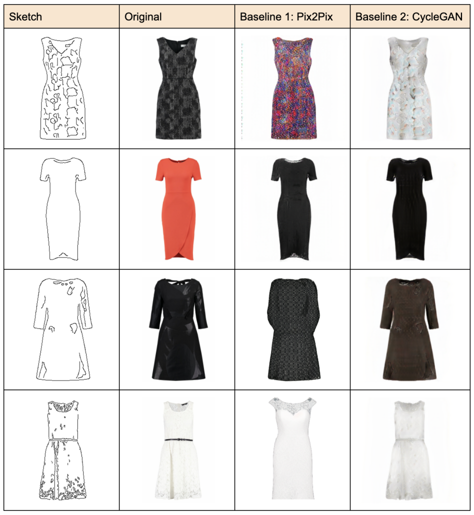


Figure 4. Sample baseline outputs on the FEIDEGGER dataset

3.3. Baselines

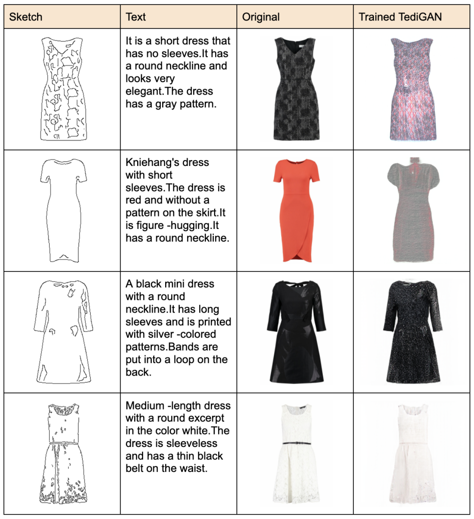
We use Pix2Pix and CycleGAN models, a type of conditional GAN well-known for image-to-image translation tasks as our baselines. These models take only the sketch (without any textual description) as input. They were chosen to establish a baseline for what can be achieved with visual data alone. This approach helps us understand the limitations and capabilities of using solely visual cues to generate detailed images. The Pix2Pix and CycleGAN models operate by training on pairs of images (in this case, a sketch and the corresponding real clothing item) and then attempting to generate a plausible output image from a new input sketch during inference.

We used the standard Pix2Pix framework but tailored the input preprocessing to handle sketches derived from the FEIDEGGER dataset, converting high-resolution fashion images into sketch formats using edge detection algorithms. The results for the same are shown in Figure 4.

3.4. GAN based pipeline approach

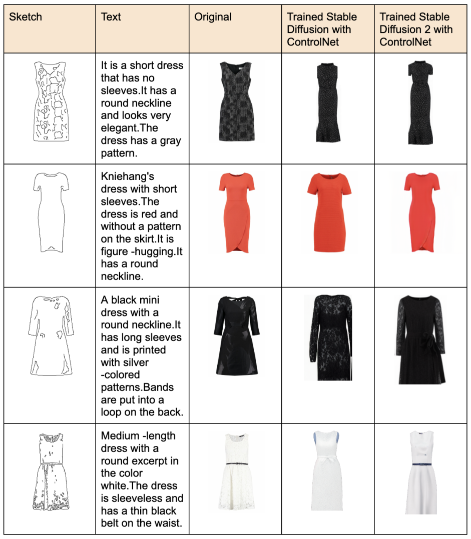
The TediGAN pipeline has 2 major components as outlined in Figure 1. The first component processes the input image through an inverted StyleGAN and generates a latent inverted representation of the image. The second component takes the inverted image along with the textual description, and uses representational embeddings for the image and text to manipulate the inverted image to incorporate the description and output the final result.

For our experiment, we tried out two approaches of training the StyleGAN. We first tried to train the StyleGAN from scratch using our fashion dataset, but we found that the model was not able to learn the style transfer from the sketch to the actual image due to limited number of images. We then tried to leverage a pre-trained StyleGAN model



Sketch	Text	Original	Trained TediGAN
	It is a short dress that has no sleeves.It has a round neckline and looks very elegant.The dress has a gray pattern.		
	Kniehang's dress with short sleeves.The dress is red and without a pattern on the skirt.It is figure -hugging.It has a round neckline.		
	A black mini dress with a round neckline.It has long sleeves and is printed with silver -colored patterns.Bands are put into a loop on the back.		
	Medium -length dress with a round excerpt in the color white.The dress is sleeveless and has a thin black belt on the waist.		

Figure 5. Sample pipeline based GAN outputs on the FEIDEGGER dataset



Sketch	Text	Original	Trained Stable Diffusion with ControlNet	Trained Stable Diffusion 2 with ControlNet
	It is a short dress that has no sleeves.It has a round neckline and looks very elegant.The dress has a gray pattern.			
	Kniehang's dress with short sleeves.The dress is red and without a pattern on the skirt.It is figure -hugging.It has a round neckline.			
	A black mini dress with a round neckline.It has long sleeves and is printed with silver -colored patterns.Bands are put into a loop on the back.			
	Medium -length dress with a round excerpt in the color white.The dress is sleeveless and has a thin black belt on the waist.			

Figure 6. Sample one-shot diffusion outputs on the FEIDEGGER dataset

for facial images and fine-tuned it for our fashion dataset. This gave us better results in comparison. The fine-tuning was done for 200 iterations with the loss function mentioned above.

For the second component of the pipeline we used CLIP, which is a neural network trained on 400 million image and text pairs, as our text encoder. We minimized the loss of the encoder detailed above with the inverted image as the input and trained for 500 iterations. We evaluated our outputs using FID score as well as human evaluation. On trying out different combinations of learning rates and iterations, we found that these settings gave us the best final result.

We observed that TediGAN was able to perform better than image-only baseline GAN models like Pix2Pix and CycleGAN in reproducing specific details of the dress described in the text, such as color and motifs, which could not be inferred from the sketch. The results of using TediGAN for the FEIDEGGER dataset are detailed in Figure 5 and Table 1.

3.5. Stable Diffusion based one-shot apporach

The one shot model expects a multimodal input in terms of both the sketch and text to guide the generation as showing in Figure 1. We modify the Stable Diffusion text to image models using the controlnet architecture as described in Figure 3. This results in a model which not only generates diverse images but is also grounded in the input and robust. The control added is the sketch of the fashion apparel which is combined with the text encoded by the CLIP model.

For the experiments we try both Stable Diffusion and Stable Diffusion 2 models modified using the ControlNet architecture. Stable Diffusion 2 offers considerable improvements over its predecessor in terms of improved fidelity and quality of generated images [7]. Moreover Stable Diffusion 2 has enhanced capabilities in understanding complex prompts and generating images from them.

We train the modified diffusion models on our fashion dataset using the loss described in the previous section. We use the Adam optimizer to train the model on our dataset for 500 iterations over 4 GPUs using the Distributed Data Parallel strategy to parallelize the data and model across the GPUs to reduce training time.

The results are evaluated using human evaluation and FID scores covering qualitative and quantitative methods. We experiement with several hyperparameters specific to the generation process such as DDIM steps that is the number of steps in the iterative image generation process, number of training iterations, and the learning rate. In the end we found 20 steps, 500 iterations and 0.001 learning rate the best for our experiments.

We observe that the one shot stable diffusion based models perform better than the pipeline approach of TediGAN and the baselines of Pix2Pix and CycleGAN. Moreover, stable diffusion 2 based model achieves the best performance in terms of FID scores, image fidelity, and ability to preserve details of the apparel described through sketch and text such as color and motifs. The results on the FEIDEGGER dataset are present in Figure 5 and Table 1.

3.6. Discussion

Looking at the generated outputs of the baseline, pipeline and one-shot approach in Figures 3,4 and 5, we can see that the baseline Pix2Pix and CycleGAN models are able to fill out the sketch completely but are unable to capture the details of the garment such as the color and the motifs, as these details are only described in the text and not inferrable from

Model	FID Score
Pix2Pix	113.5396
CycleGAN	112.69592
TediGAN	101.90133
StableDiffusion ControlNet	96.69592
StableDiffusion 2 ControlNet	93.69592

Table 1. Results on the FEIDEGGER dataset

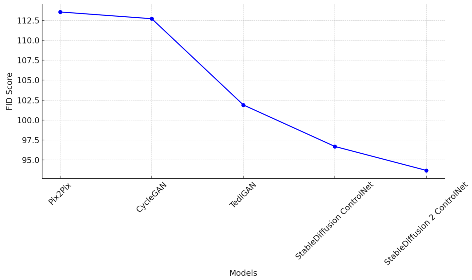


Figure 7. FID scores of various models on the FEIDEGGER datase

the black and white sketch. We can clearly observe this trend in Figure 7 where the baselines employing only sketch have much higher FID scores compared to both the pipeline TediGAN and the one-shot Stable Diffusion methods.

In the pipeline-based TediGAN approach, we can see that the model can retain the correctness of the baseline GAN models, and starts to move towards better generated image in terms of the color and print. In the second example, we can see that the baseline generated a garment with the color black, whereas the TediGAN model's output color is closer to the original red color. However, in certain scenarios it still fails to capture the color correctly.

The One-shot stable diffusion method performs the best across all the sample inputs and is able to recreate the entire apparel, with the best color, details, and hues with the stable diffusion 2 based controlnet model performing the best in terms of both the quantitative FID scores and the qualitative human evaluation. However, we do note that in cases where the sketch does not capture all the details of the apparel, the output generated differs from the expected output. We believe that a larger and more accurate dataset would further improve results.

4. Experience

4.1. Contribution

The project was possible due to the contributions of all 4 team members. There were multiple stages of the project, from problem formulation to dataset creation, reviewing the existing literature, identifying the baselines, and implementing the approaches. This included the generation of our fashion dataset as described in section 1. Then implementing approaches and writing code in PyTorch to implement the pipeline and one shot approaches for training, tuning, inference and evaluation. In summary, all members of the team were involved in the implementation aspects and were actively sharing their code and results to the shared repository for others to review. The detailed major contributions of each team member can be found in Table 2.

4.2. Project Success

We were able to use multiple approaches to successfully generate realistic designs from sketches and descriptions of fashion images. The quality of the generation was good enough, but not exactly like the expected images. We attribute this to limited training data for our specific task and computational resources to train the model for longer periods. We also understood the inner workings of GAN and Diffusion models well enough to make improvements and train it for our own custom dataset. In this regard, we feel we succeeded.

4.3. Future Work

Looking ahead, there are several avenues for further development. One immediate area of focus could be the enhancement of the model's ability to handle even more diverse and complex designs extending beyond one-piece apparel. Developing a larger and more accurate dataset can also be one area of research which would improve the performance of the models. Additionally, exploring the integration of real-time feedback loops could allow for dynamic adjustments to the generated images based on iterative user inputs.

5. Work Division

Please add a section on the delegation of work among team members at the end of the report, in the form of a table and paragraph description. This and references do **NOT** count towards your page limit. An example has been provided in Table 2.

Student Name	Contributed Aspects	Details
Amrit Khera	Implementation and Analysis	Implemented and trained multiple variations of stable diffusion modified with ControlNet. Implemented the module for model evaluation using FID score, contributed to GAN experiments and performed hyperparameter tuning.
Mini Jain	Implementation and Analysis	Impelemented and trained the TediGAN image encoder and the text encoder. Implemented different image encoder models for the 1st component of TediGAN, analysed the results for the diffusion and GAN approaches.
Puru Malhotra	Data Creation and Implementation	Implemented the logic for dataset creation for this project. Identified and trained the baseline CycleGAN and Pix2Pix models. Implemented the text encoding component of TediGAN.
Rachit Chadha	Data Creation and Implementation	Implemented the logic for dataset creation for this project, utilised translation models and edge detection. Contributed to experiments and hyperparameter tuning for both GAN and Stable Diffusion Approaches.

Table 2. Contributions of team members.

References

[1] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Neural Information Processing Systems*, 2017. 4

[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2016. 2

[3] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021. 2

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 2

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 3

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 3

[7] 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. 5

[8] Weihao Xia, Yujiu Yang, Jing Xue, and Baoyuan Wu. Tedigan: Text-guided diverse face image generation and manipulation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2256–2265, 2021. 2

[9] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 3

[10] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017. 2

A. Project Code Repository

The Github repository for our final project is at: Pencil Tip to Wardrobe Drip