Dress-UP: A Deep Unique, Personalized Fashion Recommender

Evelyn Choi
Stanford University
echoi1@stanford.edu

Poonam Sahoo
Stanford University
pnsahoo@stanford.edu

Ananya Vasireddy
Stanford University
ananyasv@stanford.edu

Abstract

As computer vision models continue to advance, fashion has become an increasingly relevant industry for these types of technologies. In fact, many popular clothing companies and e-commerce companies use such artificial intelligence models to recommend clothes to their users. In this project, we aimed to create a deep fashion recommender by extending CLIP [15], a multimodal model that takes in text and images as input and provides image outputs. We did so by investigating the effects of augmenting the architecture of the model, specifically adding a spatial transformer network (STN) and replacing the vision transformers in the original model's image encoder with Swin transformers. We also finetuned the model further using the Deep Fashion dataset, which consists of over 800,000 images of clothes with descriptive annotations [20]. After gathering our results, we then conducted user evaluations, in which we incorporated a generative image model's outputs (originally a deep convolutional generative adversarial network, but later replaced by Stability AI's Stable Diffusion 1.5 [18]) as an additional basis of comparison to our modified model's outputs. In the end, we found that further finetuniing Fashion-CLIP (a version of CLIP finetuned on fashion-related data [2]) and adding an STN beat our baselines. However, the user studies showed that many people preferred the generative Stable Diffusion model and Fashion-CLIP simply finetuned on the Deep Fashion dataset. In the future, extending upon this work could include using a more modern fashion dataset or larger dataset in general.

1. Introduction

Fashion is inherently personal and individualistic. As a result, using artificial intelligence (AI) models for customizable clothing recommendations is an exciting and increasingly popular method for e-commerce platforms such as Instagram Shopping to improve the experience of shoppers. In practice, user recommendation systems rely on owning a wealth of information about the user, at the expense of their privacy. As a result, the question that we faced was: is it possible to output individualized clothing recommendations to a user with limited user context? Our ultimate objective in this project is to create a deep fashion recommender that outputs relevant fashion images to a user with *just text and image input about their fashion tastes*. This problem is important because typical fashion recommenders often use traditional search methods reliant on large amounts of data. Instead, we aim to harness the power of recent, pretrained models such as CLIP [15] and FashionCLIP [2], a finetuned version of CLIP specialized towards fashion.

For our clothing recommender, our input at test time is a user's general clothing preferences in images and texts. We then attempt to provide image recommendations of clothing that fit the user's style. We do so using our own modified version of CLIP's image and text encoder, which outputs embeddings of the user's preferences. We then search through our fashion dataset and output images whose embeddings are most similar to those of user input. Finally, at the user evaluation stage, we supplement our outputs with generations from a generative image model, specifically Stability AI's Stable Diffusion 1.5. We initially planned to incorporate a deep convolutional generative adversarial network - or a DCGAN - instead, as highlighted in our Methods section, but Stable Diffusion outputted better images for our use case. This way, we are able to evaluate the results of our model, which will only output images from the datasets it has been trained and finetuned on, in comparison to a generative image model.

Since collecting large amounts of data for proper user evaluations is costly, we first choose the models to implement and train/finetune them. Then, we collect user preference input and run evaluations afterwards.

2. Related Work

There are several related papers in machine learning (ML), human-computer interaction, and interdisciplinary spaces focused on fashion recommendation systems and improving vision models for identifying and categorizing fashion in particular. In order to understand the state of the field, we read through survey papers on fashion recommendation systems such as Chen et al.'s survey of AI in fashion [1] and Deldjoo et al.'s survey of different fashion recommender systems [3]. In terms of vision models trained on text-image fashion datasets, many people have had success with finetuning state of the art vision models on fashion datasets. For example models such as FashionCLIP [2] and ARMANI [25] are finetuned for the fashion domain using CLIP [15] and MaskCLIP [5], respectively, and their strengths are in classification on their respective datasets but do not generalize as well, which we discuss later. Many fashion recommenders are focused more on the user data side of recommendations, a perspective best suited for e-commerce companies with large amounts of data about their customers. For example, when H&M hosted a Fashion Recommendation competition on Kaggle, the first place winner used a gradient boosting method and collaborative filtering to match similar customer profiles [11]. However, we wanted to tackle this problem with a multimodal computer vision focus instead, simplifying the amount of data inputted by the user. Fashion is an especially nuanced area because there are typical elements of computer vision problems, such as identifying different items of clothing, but there is also a more subtle aspect of identifying different styles or aesthetics. DeepStyle learns different aesthetics rather than focusing on categories of items by using a user-item feedback matrix [12]. Yet a different approach by Ramesh et al. is an outfit recommender based on specific event contexts and that recommends outfits for different scenarios in a user's life [17]. From the HCI perspective, Vaccaro et al. present an in-depth user study of people's preferences as it comes to fashion and receiving style recommendations [21]. One of their key findings is the importance of communicating about fashion via both images and text, which motivates our multimodal approach. Finally, Kang et al tackle the difficult task of fashion recommendation by incorporating visual features into the objective function of the generative model that they train [9]. They use GANs to surface recommendations to the user. While these papers show that there are numerous ways to approach a fashion recommendation problem, we knew that we couldn't rely on traditional "recommendation system" approaches such as collaborative filtering due to the small-scale user study we intended to run. We focus more on improving upon and finetuning existing fashion-related vision models and modernize our approach to fashion recommendation with stable diffusion as opposed to GANs.

3. Methods

3.1. Baseline and Vanilla Finetuned Model

Our baseline model is FashionCLIP (FCLIP), which is a variant of CLIP that is finetuned on a different fashion dataset, or the Deep Fashion dataset (iterated on more in section 4). CLIP is a ML framework that is trained on text-image pairs [15]. The objective during training is to minimize the distance between embeddings of matching text-image pairs and maximize the distance between non-matching pairs. More specifically, the loss function $\mathcal{L}_{CLIP}$ is equivalent to $\frac{1}{2}\left(\mathcal{L}_{\text{text-to-image}} + \mathcal{L}_{\text{image-to-text}}\right)$, where the first loss term ensures that the correct image is ranked highest in similarity over the incorrect images for a given text, and the second loss term does the same but instead ranks similar texts to a given image. CLIP uses separate image and text encoders. FCLIP in particular uses CLIP's ViT-B/32 Transformer architecture as an image encoder and uses a masked self-attention Transformer as a text encoder, and was trained on a fashion dataset of over 800,000 images [2]. Other baselines from the milestone that we ran but ultimately performed worse than FCLIP include CLIP (ViT-B/32 image encoder), CLIP (ResNet image encoder), and SigLIP [24].

Our first improvement on the baseline was finetuning FCLIP on the DeepFashion dataset without any changes to model architecture. We did not freeze any layers.

3.2. DCGANs and Stable Diffusion

When choosing a generative image model to compare the results of our modified model to, we initially settled on a deep convolutional generative adversarial network (DCGAN). DCGANs are a subclass of generative adversarial networks, a type of model that has two major components: a discriminator, which attempts to distinguish between real and fake images, and a generator, which attempts to output images that are able to fool the discriminator by becoming more "realistic" [6]. However, DCGANs are unique in the fact that they incorporate convolutional neural networks into their architectures [16].

In the context of our project, we implemented and trained a DCGAN on the Deep Fashion dataset. Before accomplishing this, however, we first trained a simple DCGAN on black and white images of clothes from the Fashion MNIST dataset to see if we could get stable results [23]. This model's architecture was adopted from pre-existing open source code [7] tailored towards the Fashion MNIST dataset, and the full structure can be seen in the Appendix in Figure 10 and Figure 11.

After achieving success with this model, we moved on to training a more complex DCGAN on our Deep Fashion dataset, meaning the new model needed to be capable of processing colored (RGB) images [10]. To develop the architecture for this model, we took inspiration from pre-existing open-source code involving a DCGAN tailored to generate images of hands. However, we modified this code by changing the data pre-processing, the details of which are described in Section 4. The full architecture of this model's discriminator and generator can be seen in Figure 1 and Figure 2.

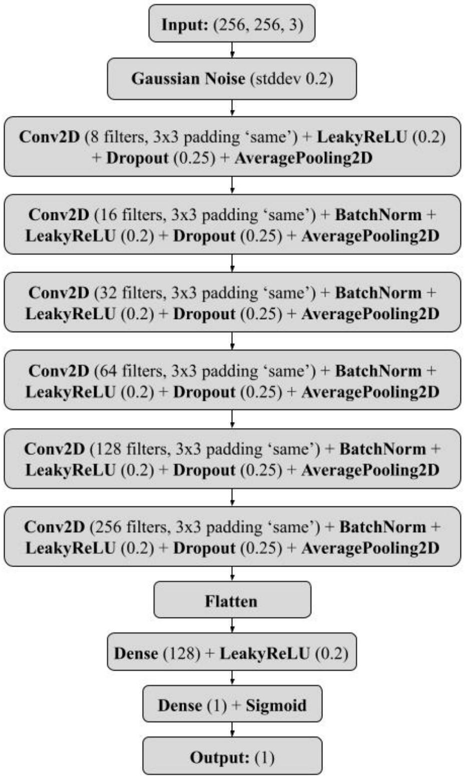

Input: (256, 256, 3)
Gaussian Noise (stddev 0.2)
Conv2D (8 filters, 3x3 padding 'same') + LeakyReLU (0.2) + Dropout (0.25) + AveragePooling2D
Conv2D (16 filters, 3x3 padding 'same') + BatchNorm + LeakyReLU (0.2) + Dropout (0.25) + AveragePooling2D
Conv2D (32 filters, 3x3 padding 'same') + BatchNorm + LeakyReLU (0.2) + Dropout (0.25) + AveragePooling2D
Conv2D (64 filters, 3x3 padding 'same') + BatchNorm + LeakyReLU (0.2) + Dropout (0.25) + AveragePooling2D
Conv2D (128 filters, 3x3 padding 'same') + BatchNorm + LeakyReLU (0.2) + Dropout (0.25) + AveragePooling2D
Conv2D (256 filters, 3x3 padding 'same') + BatchNorm + LeakyReLU (0.2) + Dropout (0.25) + AveragePooling2D
Flatten
Dense (128) + LeakyReLU (0.2)
Dense (1) + Sigmoid
Output: (1)

Figure 1. Discriminator architecture diagram of larger, RGB DCGAN.

In the end, however, we decided to use Stability AI's Stable Diffusion 1.5 instead of our DCGAN in the user evaluations due to better performance. Stable Diffusion is a text-to-image model that utilizes several denoising autoencoders to synthesize original images [18]. It is important to note that Stable Diffusion is a latent diffusion model, meaning the model is applied to a latent representation of images rather than raw pixel values. As a result, this model is much less expensive to train and use at test time than previous diffusion models.

3.3. Spatial Transformer Networks

To improve on the performance of the FashionCLIP model, we edited the architecture of the model, especially with regards to the image encoder. We implemented a version of the FashionCLIP model with an added spatial transformer network (STN) [8]. This involved using the fine-tuned weights from the FashionCLIP model, and training the new STN and FashionCLIP model end-to-end on the DeepFashion dataset. The STN has a localization network of 2 convolutional layers followed by a max pooling layer. It then extracts features to produce the affine transformation parameters with 2 fully connected layers. In addition, we initialize the STN at the identity transformation.

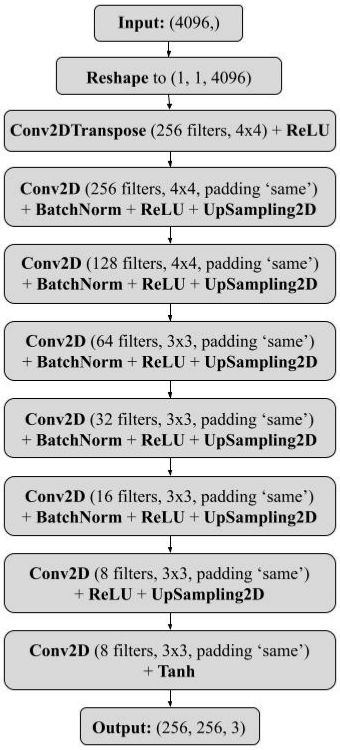

Input: (4096,)
Reshape to (1, 1, 4096)
Conv2DTranspose (256 filters, 4x4) + ReLU
Conv2D (256 filters, 4x4, padding 'same') + BatchNorm + ReLU + UpSampling2D
Conv2D (128 filters, 4x4, padding 'same') + BatchNorm + ReLU + UpSampling2D
Conv2D (64 filters, 3x3, padding 'same') + BatchNorm + ReLU + UpSampling2D
Conv2D (32 filters, 3x3, padding 'same') + BatchNorm + ReLU + UpSampling2D
Conv2D (16 filters, 3x3, padding 'same') + BatchNorm + ReLU + UpSampling2D
Conv2D (8 filters, 3x3, padding 'same') + ReLU + UpSampling2D
Conv2D (8 filters, 3x3, padding 'same') + Tanh
Output: (256, 256, 3)

Figure 2. Generator architecture diagram of larger, RGB DCGAN.

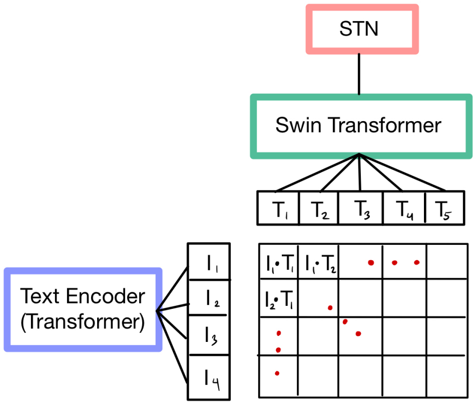


Figure 3. Diagram of STN+Swinn-FCLIP

Spatial Transformers allow a model to learn what spatial transformations to perform on the input images in order to improve a model's geometric invariance. This is useful for making models more robust to diverse image inputs. The transformer comprises of a localization network, that learns different transformation parameters that are to be applied to the input. Its grid generator then produces a grid representing which coordinates in the input image map to ones in the output image. The transformer then uses differentiable image sampling to produce the output. We incorporated an STN into the model since it allows the model to focuse more on relevant parts of the images. This is important for our dataset, as DeepFashion contains clothes on many different models in different poses. For this milestone, we experimented with placing an STN before the image encoder. We hoped this may allow the model to focus more on the content of the image rather than certain geometric or visual differences, such as the angle of the photo or rotation of clothing. For our STN we used a localization network that was a CNN with two layers, and a regression network consisting of two fully connected layers.

3.4. Swin Transformers

We also experimented with replacing the vision transformers in the image encoder with Swin transformers [13]. This is a recent architecture that implements a shifted window-based self-attention mechanism, where attention is repeatedly computed within shifted local windows. This reduces its computation complexity, as a normal vision transformer applies self-attention globally. This use of local context can make the Swin transformer more robust to image variations and allow it to better pay attention to important parts of the clothing. Its hierarchical structure also helps the model capture more important features of the image. We used a Swin transformer that was pretrained on ImageNet-1k, with a patch size of 4 and a window size of 7 [4] [22]. We chose this in hopes that a model with a large amount of general knowledge could more quickly learn and better adapt to our fashion dataset. A diagram of our STN+Swinn-FCLIP model (appending an STN and replacing the ViT in CLIP with a Swinn transformer) is in Figure 3.

4. Dataset and Features

While training the initial models, we use the Deep Fashion dataset [14], which contains over $800,000$ images of clothes annotated with category and descriptive attributes. See Figure 4 for an example image from the DeepFashion dataset. We specifically used a subset of the ”Category and Attribute Prediction Benchmark” section of the dataset with $14,000$ train images, $2,000$ validation images, and $4,000$ test images to train and finetune our models. We used this benchmark because of the rich metadata available about each image with regards to clothing type, material, and style. Since the dataset did not come with captions, we generated captions based on the annotated attributes. For example, a picture with the annotations ”solid”, ”long sleeve”, ”v-shaped neckline”, ”chiffon”, ”conventional” would yield the following caption: ”garment with solid pattern, long sleeve, v-shaped neckline, chiffon fabric, conventional fit.”

With regards to preprocessing, CLIP’s preprocessing includes reshaping the image, center cropping if necessary, and then normalizing each channel of the images before being passed through the model. For STN, the image is instead simply resized into a 224 by 224 pixel image before being passed into the model. For our DCGAN (the large RGB one), we resized our images to 256 by 256 pixel images and included horizontal flips to augment the dataset. The pre-existing model also allowed vertical flips, but we chose to omit them because images of clothes have a specific vertical, upright orientation. We also received user input from the participants in our study in the form of images. In order to preprocess user responses, we converted WEBM, PNG, and other image file types into JPEG files before going through the preprocessing described above so that embeddings could be generated successfully.

Figure 4. Example image from the DeepFashion dataset.

5. Experiments

5.1. Model Selection Experiments

For all models we used a learning rate of 1e-6. For the Swinn and STN augmented models, we experimented with learning rate values of 1e-3, 1e-4, 1e-5, and 1e-6. We did not have the compute resources for cross-validation, as we could not train many models until convergence. Instead, we found that the loss of our models only decreased with a learning rate as small as at least 1e-5 and 1e-6. We experimented with both and found that 1e-6 produced the best results for our models. We also used an AdamW optimizer (Adam optimizer with weight decay) as it is one of the best and most stable optimizers for learning, and is also used by the FCLIP model.

5.2. DCGANs

For the small black and white DCGAN we used 30 epochs. For the large, more complex RGB DCGAN, we used 427 epochs as that was what was feasible with our computational resources.

5.3. User Study

We conducted a user study of 20 students. Each participant gave text descriptions and uploaded $2-4$ images of clothing that matched their style. For the different CLIP models, we computed embeddings of user input and searched for images in the Deep Fashion dataset that were close to those embeddings to surface to the user. For the diffusion model, we took in user text input to generate new images. We asked the user to evaluate four models: the baseline FCLIP, the version of FCLIP we finetuned on DeepFashion, and the STN model trained on DeepFashion, as well as stable diffusion. For each model, we surfaced four recommendations to the user. We also asked the users to rate each model on a scale from $1-10$, rank the models in order of which recommendations they preferred, and give qualitative feedback on the model outputs.

6. Results

6.1. Model Selection

For each model, we generated image and text embeddings from the text-image pairs in our dataset and then computed the cosine similarity between the respective embeddings for each pair. We define the *text surfacing accuracy* to be the proportion of images for which the closest text embedding for that image corresponds to the actual text description of that image. For each image, we searched through all of the texts to find the closest text embedding for that given image. The *attribute surfacing accuracy* is the proportion of attributes in an image are accurately reflected in the ”nearest” text embedding.

Model	Cos Simil.	Text SFA	Attr. SFA
CLIP* (ViTB-32)	0.276	0.22	0.589
FCLIP*	0.272	0.074	0.672
Finetuned FCLIP	**0.288**	0.**338**	**0.819**
STN-FCLIP	0.284	0.142	0.707
SWIN-FCLIP	0.160	0.058	0.584
STN+SWIN-FCLIP	0.164	0.047	0.577

Table 1. Metric Results (SFA is Surfacing Accuracy). Baseline Models are starred, and the best scores are bolded.

We found that vanilla finetuned FCLIP and STN+FCLIP worked the best out of all of our modified models. To determine this, we took into account the three column metrics listed in Table 6.1. As a result, we included these two models in the user study. The significant improvement of our finetuned FCLIP model over our baselines across all metrics shows that our additional data from the DeepFashion dataset has helped the model adapt better to the new data. Adding an STN seems to improve performance across all metrics over our baseline, but still does not produce as high results as vanilla finetuned FCLIP. This may be because our DeepFashion dataset that we train on may be too small and not visually diverse enough to fully utilize the benefits of an STN. For example, in comparison, FCLIP was finetuned on over 800,000 fashion images. In addition, since the base pretrained CLIP model already uses a Transformer based architecture, and it is already trained on fashion, it may already encode the spatial features in a way that does not vary as much depending on the alignment of the image. Thus adding an STN could reap minimal benefits. It may be interesting to see if the STN can be better utilized if we trained the model with it from scratch. However, we did not have the compute resources for this.

We also found that Swin-FCLIP and STN+Swin-FCLIP seemed to have the lowest evaluation metric values. Again, this is most likely due to overfitting on the DeepFashion dataset, as we retrained complex architectures on a smaller dataset. In addition, due to compute constraints, we used a Swin transformer that was pretrained on ImageNet-1k [4], hoping that this general knowledge would help the model adapt better to our fashion images. However, this general knowledge may not have translated well to our fashion dataset, as it may look at different visual features that are not as significant for our fashion images. For example, ImageNet-1k has around 1,000 different classes and over a million images, and thus it is possible that our large pretrained Swin transformer was too complex for our dataset and overfit it to.

In addition, the Swin transformers are much different in architecture than the vision transformers used in CLIP. Thus, we may have better results if we train the Swinn-FCLIP and STN+Swinn-FCLIP on a larger fashion dataset and for longer, from scratch. This may produce better results than using a pretrained Swinn model and finetuned weights for the text encoder. Unfortunately we did not have the compute resources for this.

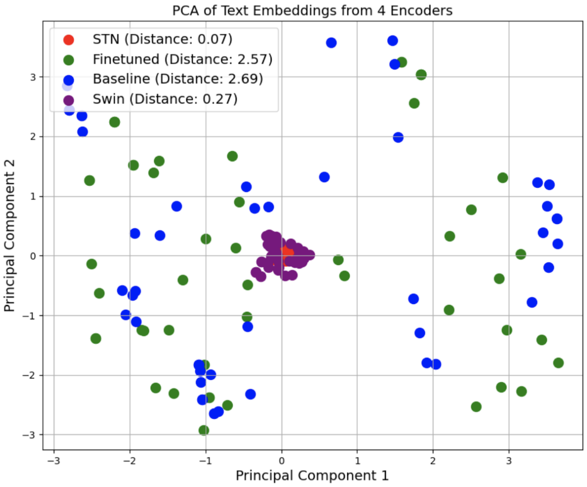


Figure 5. PCA of texts embeddings from different model encoders.

For finetuned FCLIP, STN-FCLIP, FCLIP (our baseline), and Swin-FCLIP, we examined the image and text embeddings of these models on the first 50 datapoints in our test set using PCA in 6.1 and 6.1. We can see in the graph that the STN-FCLIP model has the smallest mean embedding radius for both text and image, and its embeddings are much more closely clustered, especially in comparison to the baseline FCLIP and finetuned FCLIP. This may be caused by the STN model overfitting on the small, training dataset encoding less variance in its embeddings, as it only pays attention to a narrow amount of spatial features. In addition, the Swin-FCLIP model also demonstrates this tight clustering, most likely due to overfitting on the dataset. This makes sense as qualitatively, Swin-FCLIP and STN-FCLIP were more likely to confuse similar text embeddings with visually similar but different components, than finetuned FCLIP. For example, they would mistakenly believe an image of a cotton mini-length striped dress had the most similar embedding to the text, ”chiffon mini-length striped dress,” instead of ”cotton mini-length striped dress.” In general, Swin-FCLIP and STN-FCLIP had more difficulty identifying fabrics than finetuned FCLIP, which may be due to it being more subtle to visually recognize.

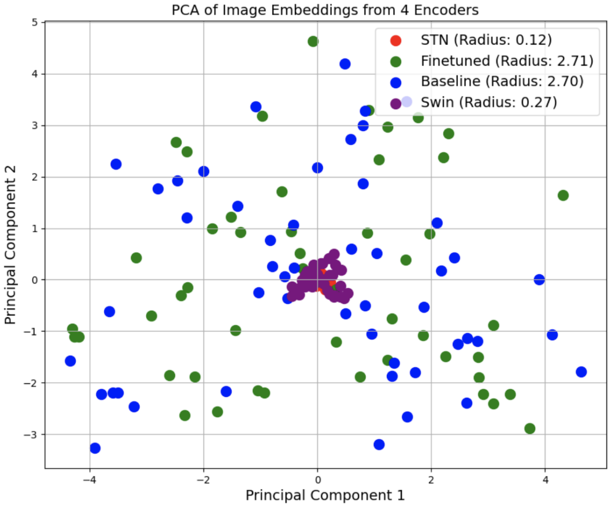


Figure 6. PCA of image embeddings from different model encoders.

6.2. DCGANs Results

The results of the first simple DCGAN were promising. After 30 epochs, the generated images were pixelated, but were recognizable as pieces of clothing (see 9). The second RGB DCGAN, however, did not produce images as clear or recognizable as the first DCGAN. As seen in 7, after 427 epochs, the images were blurry and somewhat contained the outlines of a model wearing clothes (which is the bulk of the Deep Fashion dataset). Nevertheless, the images were certainly not clear enough to be compared to our outputs in the user evaluation stage. This is likely due to the complexity of the images, which are 1) in color and 2) involve human models and faces. We knew that improving on this GAN would likely require training for even more epochs and potentially changing other parameters, which did not seem feasible with our computational resources and time constraints. As a result, we decided to switch to Stable Diffusion (version 1.5) instead, as it is open-source, fast, and produces much higher-quality outputs. An example of these outputs can be seen in Figure 8; evidently, these images are much less blurry than the outputs from the GAN and resemble photos of real clothes.

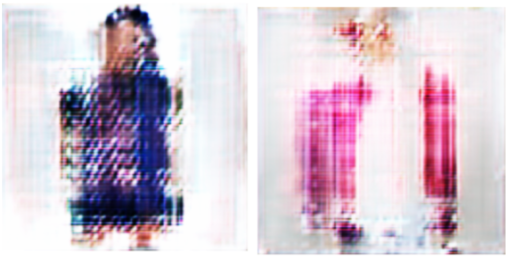


Figure 7. Outputs from larger, RGB DCGAN trained on Deep Fashion.

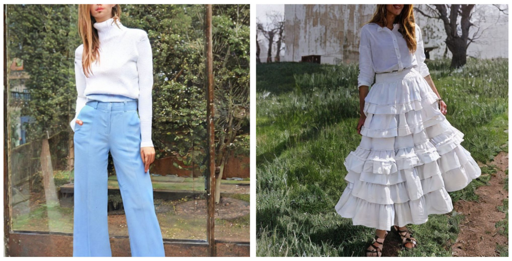


Figure 8. Outputs from state-of-the-art Stable Diffusion model.

6.3. User Study Results

A metric we used to evaluate each model based on user preferences is *Mean Average Precision@k* (MAP@k). This is a common metric for recommender systems and was used in H&M’s Fashion Recommendation Challenge [11].For a user, *Precision@k* can be defined as the proportion of top-k recommendations that are relevant. Then, *Average Precision@k* is the average of Precision@i for $i = 1, 2, \ldots, k$. Finally, MAP@k is the mean of Average Precision@k over all users. As we also asked users to rank the models and rate them out of 10, we calculated MAP@4, the average rank, and average rating for each model. The possible ranks were from 1 (best) to 4 (best), and the satisfaction scores were from 1 (worst) to 10 (best). The results are displayed in Figure 2.

Model	MAP@4	Avg. Rank	Avg. Rating
Baseline	0.417	2.63	5.53
Finetuned FCLIP	**0.559**	**1.89**	6.32
STN FCLIP	0.204	3.47	3.42
Stable Diffusion	0.558	2.0	**6.58**

Table 2. Results of our user evaluation for four models: baseline FCLIP, finetuned FCLIP, STN-modified FCLIP, and stable diffusion. Best scores are bolded

Overall, users were somewhat satisfied given the limited amount of information inputed to the recommendation system. Some of the feedback about the recommendations could have definitely been mitigated by soliciting more information from the users. For example, one user noted that he was recommended women's clothing, and he wasn't interested in those styles. Another user said that "often times the actual clothing items themselves were fine, but they wouldn't align with colors i liked." Considering that the best average rating for a model was 6.58 on a 10-point scale, there is still a lot of room to grow in this area. Both Finetuned FCLIP and Stable Diffusion outperformed baseline FCLIP on all metrics. However, STN FCLIP consistently was the worst model across all metrics. Stable Diffusion outperforms Finetuned FCLIP with regards to average ratings, but their MAP@4 scores are nearly identical and their ranking is worse.

The discrepancy between rank and score for Stable Diffusion versus Finetuned FCLIP can be explained by the variance of people's satisfaction scores. The variance for FCLIP was 0.55, whereas the variance in user scores for Stable Diffusion was more than double that, at 1.22. The polarizing nature of the Stable Diffusion model can be explained by some of the qualitative feedback: people noted that the AI-generated faces were "terrifying." However, a different user noted that "Models B and C [Finetuned FCLIP and Baseline] were relatively similar in that they included at least a few items that I might wear... Model D [Stable Diffusion] seemed much better for me because all of them were not only items that I would consider wearing, but that were *unique items* as opposed to commonplace styles." When taking into account the variance in user response, it seems that Finetuned FCLIP may be the best performing model overall given that it has the lowest average rank, is essentially tied for best MAP@4, and is less polarizing to users.

Finally, users also noted that the outputs from the FCLIP-variant models had "all very 2010's vibes." Considering that the outputs were surfaced from the DeepFashion dataset from 2016, it makes sense that users may consider some images to be out of style. So it is most likely the case that people prefer the stable diffusion model due to its more personalized flair, given that the Deep Fashion dataset is more generic and outdated. The user feedback gives us some understanding that personalized and specific recommendations were preferred. Trying to develop ways to mitigate variance amongst users is difficult, as people have different objectives and preferences. Further, satisfying preferences such as gendered clothing or color require collecting more information from users before surfacing recommendations.

7. Conclusion and Future Work

We sought to create a deep fashion recommender by improving on the multimodal model CLIP [15]. We did so by adding some state-of-the-art architectural augmentations, namely STNs and Swin Transformers. Additionally, since our recommender could only output images from the datasets it was trained and/or finetuned on, we aimed to incorporate a generative image model's outputs in our user evaluations as well to provide a basis of comparison.

We found that due to compute limitations (small train dataset and having to use pretrained weights), our architectural changes caused our new models to overfit to our small fashion dataset. However, we found that further finetuning the Fashion-CLIP model on our smaller dataset, and even adding an STN, allowed us to beat our baselines in our evaluation metrics. But in terms of user studies, the Stable Diffusion model and finetuned Fashion-CLIP were able to beat our baseline. Additionally, we initially planned to use a DCGAN as our generative image model, but we found that our outputs were poor in quality and that it was unrealistic to further improve upon them with our computational and time constraints. As a result, we decided to use the open-source Stable Diffusion model instead.

For future work, we'd like to try training the models that incorporate STNs and Swin transformers in their architecture from scratch on a larger dataset. We found before in our analysis that these models had worse results than the vanilla finetuned FCLIP since they overfit on the DeepFashion data. In addition, it would be useful to use newer datasets to train on or to use for recommendations, as some of the feedback from the user study was that some of the recommended clothes from our models were outdated styles from the 2010s.

8. Contributions and Acknowledgements

Evelyn implemented, tuned, and evaluated all proposed architectural changes, and analyzed results. Poonam ran all of the baselines, finetuned FCLIP, conducted the user survey (contacting participants, collecting responses, sending out recommendations, collecting feedback), analyzed the user study results, and developed the metrics for benchmarking the performance of CLIP models on this dataset.

Ananya worked on the simple, black and white DCGAN, the complex RGB DCGAN, and analyzing their results. She also attempted to further finetune the Stable Diffusion model, which is detailed in the Appendix.

9. Appendix

9.1. Black and White DCGANs

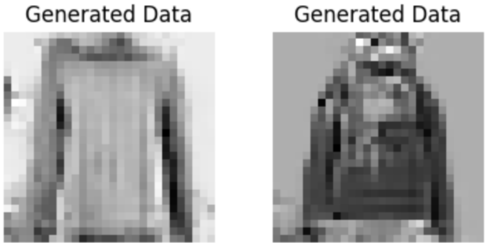


Figure 9. Outputs from simple, black and white DCGAN trained on Fashion MNIST.

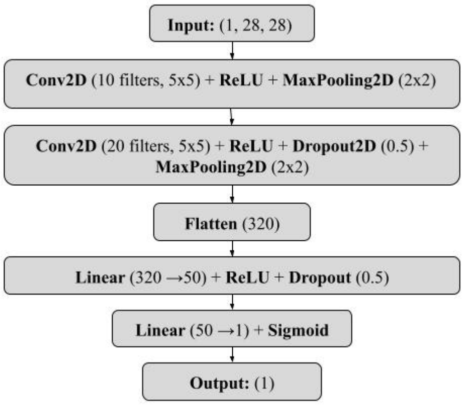
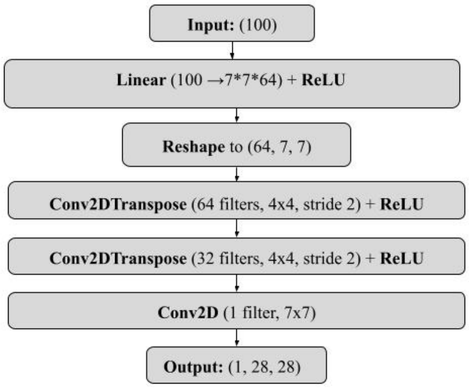

Input: (1, 28, 28)
Conv2D (10 filters, 5x5) + ReLU + MaxPooling2D (2x2)
Conv2D (20 filters, 5x5) + ReLU + Dropout2D (0.5) + MaxPooling2D (2x2)
Flatten (320)
Linear (320 →50) + ReLU + Dropout (0.5)
Linear (50 →1) + Sigmoid
Output: (1)

Figure 10. Discriminator architecture diagram of simple, black and white DCGAN.

9.2. Stable Diffusion

Although we were unsuccessful, we did attempt to finetune Stable Diffusion in two different ways: 1) finetune the model over a smaller portion (4,000 images) of the Deep Fashion dataset and 2) finetune the model via Google's DreamBooth [19] to teach the model a specific new concept (in our case, a person's face so that they would be able to see themselves wearing the clothes they request). Unfortunately, we were unable to get the first type of finetuning running, and although we were able to generate checkpoint files/execute the full finetuning script for the second type, it failed at test time when uploaded to HuggingFace.


Input: (100)
Linear (100 →7*7*64) + ReLU
Reshape to (64, 7, 7)
Conv2DTranspose (64 filters, 4x4, stride 2) + ReLU
Conv2DTranspose (32 filters, 4x4, stride 2) + ReLU
Conv2D (1 filter, 7x7)
Output: (1, 28, 28)

x

Figure 11. Generator architecture diagram of simple, black and white DCGAN.

References

[1] H.-J. Chen, H.-H. Shuai, and W.-H. Cheng. A survey of artificial intelligence in fashion. *IEEE Signal Processing Magazine*, 40(3):64–73, 2023.

[2] P. J. Chia, G. Attanasio, F. Bianchi, S. Terragni, A. R. Magalhães, D. Goncalves, C. Greco, and J. Tagliabue. Contrastive language and vision learning of general fashion concepts. *Scientific Reports*, 12(1), Nov 2022.

[3] Y. Deldjoo, F. Nazary, A. Ramisa, J. McAuley, G. Pellegrini, A. Bellogin, and T. D. Noia. A review of modern fashion recommender systems. *ACM Comput. Surv.*, 56(4), oct 2023.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[5] X. Dong, J. Bao, Y. Zheng, T. Zhang, D. Chen, H. Yang, M. Zeng, W. Zhang, L. Yuan, D. Chen, F. Wen, and N. Yu. Maskclip: Masked self-distillation advances contrastive language-image pretraining, 2023.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. 2014.

[7] R. Gupta. Data-aug-gan, Jan. 2024.

[8] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks, 2016.

[9] W.-C. Kang, C. Fang, Z. Wang, and J. McAuley. Visually-aware fashion recommendation and design with generative image models. *2017 IEEE International Conference on Data Mining (ICDM)*, pages 207–216, 2017.

[10] D. Klock. Dcgan256, 2020.

[11] C. G. Ling, ElizabethHMGroup, FridaRim, inversion, J. Ferrando, Maggie, neuraloverflow, and xlsrln. H&m personalized fashion recommendations, 2022.

[12] Q. Liu, S. Wu, and L. Wang. Deepstyle: Learning user preferences for visual recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page

841–844, New York, NY, USA, 2017. Association for Computing Machinery.

[13] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

[14] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[15] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021.

[16] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. 2015.

[17] N. Ramesh and T.-S. Moh. Outfit recommender system. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 903–910, 2018.

[18] R. Rombach, A. Blattman, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. 2022.

[19] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022.

[20] Y.-G. Shin, Y.-J. Yeo, M.-C. Sagong, S.-W. Ji, and S.-J. Ko. Deep fashion recommendation system with style feature decomposition. In *2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)*, pages 301–305, 2019.

[21] K. Vaccaro, T. Agarwalla, S. Shivakumar, and R. Kumar. Designing the future of personal fashion. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–11, New York, NY, USA, 2018. Association for Computing Machinery.

[22] R. Wightman. Pytorch image models. `https://github.com/huggingface/pytorch-image-models`, 2019.

[23] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[24] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training, 2023.

[25] X. Zhang, Y. Sha, M. C. Kampffmeyer, Z. Xie, Z. Jie, C. Huang, J. Peng, and X. Liang. Armani: Part-level garment-text alignment for unified cross-modal fashion design. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22. ACM, Oct. 2022.