Finding the Fit: Vision-Language Models for Clothing Retrieval

Upamanyu Dass-Vattam
Stanford University
udvattam@stanford.edu

Henry Palmer
Stanford University
hpalmer@stanford.edu

Abstract

Fashion inspiration is increasingly shaped by online visual media, yet participating in aesthetic-driven fashion communities remains financially inaccessible for many due to the high cost of designer clothing. We propose a computer vision system that identifies affordable alternatives to high-end fashion by learning multimodal embeddings of clothing images and text descriptions. Using the DeepFashion-MultiModal dataset, we train and evaluate several embedding models (linear, CNN/LSTM, and fine-tuned ResNet/DistilBERT) alongside a multi-label classification pipeline to predict shape, fabric, and color attributes. We explore both cosine similarity and Goodall distance to assess embedding quality and test two recommendation strategies: one based on nearest-neighbor search in the learned embedding space, and another using classifier-predicted attributes. Expert human raters evaluated the system's recommendations, showing a strong preference for the embedding-based approach. Our results suggest that multimodal representations can successfully capture nuanced style features, offering a scalable solution for democratizing access to fashion trends.

1. Introduction

The digital revolution has transformed the fashion landscape, with social media platforms catalyzing unprecedented growth in trend visibility and consumption. Fashion communities flourish across platforms like Instagram, TikTok, and Pinterest, creating vibrant micro-communities centered around specific aesthetics and designers. However, genuine participation in these spaces often faces a significant financial barrier: designer and luxury items remain prohibitively expensive for most consumers, even as their cultural visibility reaches an all-time high.

In response, demand for affordable fashion has surged, shaping both commercial platforms and social conversations in the fashion world. Long-standing secondhand clothing marketplaces like Poshmark, Depop, and Vinted have reached new levels of popularity, while niche platforms like Grailed have carved out spaces for avant-garde resale. An entire genre of content has emerged around secondhand shopping, especially among Gen Z creators, who vlog thrift hauls, resell curated finds, and offer styling tips rooted in affordability. These trends are influenced not just by economic conditions, but also by a strong sense of nostalgia, with styles like Y2K and '70s flared pants seeing renewed attention. Despite decades-long advances in visual search technology, such as Google reverse image search or Pinterest, most existing systems are limited in scope. These platforms are often not fashion-specific and return images from runway shows or old listings for clothes no longer available for purchase. When they do return viable fashion alternatives, they tend to be either in the same luxury tier, which is inaccessible to the consumer, or fast fashion alternatives like Shein that are poor quality and environmentally damaging.

Machine learning has become an increasingly important tool in the fashion industry. Computer vision models are used for trend forecasting, recommendation systems personalize shopping experiences, and visual search tools help users find similar products from photos. While these technologies improve convenience, few of them meaningfully address the affordability gap between aspirational fashion and realistic access. Visual search might show similar silhouettes, but the recommendations are often within the same price bracket or from brands outside a user's budget.

Our project goal is to build a system that identifies budget-friendly alternatives with similar aesthetic properties. We separate this problem into two tasks: building a system to identify alternatives, and deploying the system on budget-friendly, second-hand websites. Based on the scope of CS231N and the timescale of the project, we decided to focus on the first task.

We approach this problem not just as technologists, but as fashion enthusiasts who believe that style shouldn't be limited by price point. As Gen Z reshapes the fashion economy through secondhand markets, upcycling, and online communities, there is growing demand for tools that reflect how people actually engage with clothes. By bridging fashion inspiration and affordability, we hope to make trend partici-

pation more inclusive and reflective of how people actually discover and wear clothes today.

2. Related Work

Recent advances in multimodal representation learning have significantly shaped the landscape of fashion retrieval systems, enabling both practical applications and futuristic tools. For example, a study by Park and colleagues [1] focused on improving instance-level visual search within outfit images using convolutional neural networks (CNNs) and a combination of categorical and instance-level loss functions. Their goal was to retrieve exact product matches under varying poses and lighting conditions. In contrast, our system will emphasize stylistic similarity rather than exact duplication, using attribute-based evaluation to capture specific visual details like sleeve length or neckline.

More recent transformer-based approaches such as FashionVLP [2] expand fashion retrieval into interactive scnearios. FashionVLP enables users to iteratively refine their search using natural language, combiding image embeddings with user input in a unified transformer framework. While this study optimized retrieval accuracy for commercial, dialog-based applications, our system will use a static query interface with a focus on visual and textual similarity.

The work of Moro et. al [3] proposes a more scalable solution to multimodal retrieval by decoupling image and text embeddings through a two-stage process: pretraining a vision-language transformer (ViT) and then applying deep metric learning (DML) to map each modality into a shared latent space. Their architecture avoids paired inference at test time and supports fast approximate k-nearest-neighbor search. Our project adopts a similar efficiency principle, using cosine similarity in a precomputed embedding space, but we incorporate interpretability through a classifier that outputs shape, fabric, and color attributes. This enables us to evaluate embedding quality not only by cosine distance, but also by how well the clusters align with meaningful visual categories.

Similarly, Chia et al. [4] introduced FashionCLIP, an adaptation of OpenAI's CLIP model [5] trained on fashion-specific image-text pairs to support zero-shot image retrieval and classification. Inspired by this work, we will use a contrastive loss function to align visual and textual modalities. However, while FashionCLIP was meant to learn a general-purpose fashion representation, we will focus on task-specific tuning for finding affordable alternatives.

Beyond technical papers, creative projects such as daydream.ing [6] have pushed the boundaries of fashion design with machine learning. Using diffusion models and latent text-image embeddings, daydream.ing generates novel fashion visuals from text-based prompts. While they are not a retrieval system, this company shares our interest in democratizing fashion and highlights how AI can reshape visual culture.

Overall, these works illustrate the spectrum of vision-language systems in fashion, from scalable architectures to generative artistry. We will attempt to unify some of these findings, such as contrastive embedding and multimodal retrieval, to enable affordable fashion discovery.

3. Data

We are using the DeepFashion-MultiModal dataset [7] to train and test our embedding models and classification models. This dataset includes 43,497 high-resolution human images with manual annotations of attributes for clothes' shapes, fabrics, and colors. The annotations are defined as follows, where 'NA' means the image does not contain the corresponding category:

Shape Annotations:

0. sleeve length: 0 sleeveless, 1 short-sleeve, 2 medium-sleeve, 3 long-sleeve, 4 not long-sleeve, 5 NA
1. lower clothing length: 0 three-point, 1 medium short, 2 three-quarter, 3 long, 4 NA
2. socks: 0 no, 1 socks, 2 leggings, 3 NA
3. hat: 0 no, 1 yes, 2 NA
4. glasses: 0 no, 1 eyeglasses, 2 sunglasses, 3 have a glasses in hand or clothes, 4 NA
5. neckwear: 0 no, 1 yes, 2 NA
6. wrist wearing: 0 no, 1 yes, 2 NA
7. ring: 0 no, 1 yes, 2 NA
8. waist accessories: 0 no, 1 belt, 2 have a clothing, 3 hidden, 4 NA
9. neckline: 0 V-shape, 1 square, 2 round, 3 standing, 4 lapel, 5 suspenders, 6 NA
10. outer clothing a cardigan?: 0 yes, 1 no, 2 NA
11. upper clothing covering navel: 0 no, 1 yes, 2 NA

Fabric Annotations:

0 denim, 1 cotton, 2 leather, 3 furry, 4 knitted, 5 chiffon, 6 other, 7 NA

Color Annotations:

0 floral, 1 graphic, 2 striped, 3 pure color, 4 lattice, 5 other, 6 color block, 7 NA

The dataset is sufficiently large that image augmentation is unnecessary. The images are scaled to 224x224 pixels and normalized. The textual captions are then padded to the max sequence length.

4. Methods

Recent advances in text-to-image modeling have enabled users to better find products from searches, but text queries leave a lot to be desired in a visual medium such as fashion. When virality is captured in an Instagram post or advertise-

ment, a user's description may lose details of the image or be biased by their own preferences. Therefore, we determined that using an image-based query would allow for a more objective comparison and search. However, the text description still provides useful information, so our final representation of an image is a multimodal representation of both an outfit's image and its textual description.

4.1. Embedding

To build these multi-modal representations, we first had to choose an embedding model to project image-text item pairs into a shared latent space.

4.1.1 Baseline: Resnet10/DistilBERT-mini

We started by using a pretrained Resnet model [8], Resnet10, to embed images, and a pretrained transformer model, Distilbert-mini [9], to embed textual descriptions. We then concatenated each of these embeddings to give us a 1280-dimensional vector for each of 30,448 image-text pairs in the training set.

4.1.2 Trained Embedding Models

Next, we trained 3 embedding model types: linear, 2D-CNN/LSTM, and fine-tuned Resnet10/DistilBERT-mini. We trained the first two models from scratch and fine-tuned the baseline as our third model. For each embedding model, we used an 70/20/10 train/validation/test split. We used the following loss function, inspired by CLIP [5]:

$$\mathcal{L} = -\frac{1}{2N}\sum_{i=1}^{2N}\log\left(\frac{\exp(S_{ii})}{\sum_{j=1}^{2N}\exp(S_{ij})}\right)$$

where:

- N is the number of image-text pairs in a batch.
- $\mathbf{Z} \in \mathbb{R}^{2N\times d}$ is the concatenated matrix of image and text embeddings: the first N rows are image embeddings and the next N rows are text embeddings.
- $\tau > 0$ is the temperature scaling factor.
- $\mathbf{S} = \frac{\mathbf{Z}\mathbf{Z}^\top}{\tau} \in \mathbb{R}^{2N\times 2N}$ is the similarity matrix computed from all pairwise dot products of the embeddings, scaled by τ.
- S_{ii} is the similarity between the i-th embedding and itself (its positive pair).
- $\sum_{j=1}^{2N}\exp(S_{ij})$ sums over all similarities for the i-th embedding, including both positives and negatives.

After training each embedding model, 667 training examples were discarded for not having textual descriptions, leaving us with a 64-dimensional embedding vector for each of 29781 image-text pairs in the training set.

4.2. Classification

All classification models are trained with a weighted cross-entropy loss to account for severe class imbalances in the labels. The models are trained with a 70/20/10 train/val/test split, and evaluated on loss and per class and overall accuracy.
The labels that make up the feature vector consist of 11 shape labels, from sleeve length to neckline to whether or not the navel is covered, a fabric label, and a color label.

4.2.1 Baseline

Our baseline model is a linear regression model. This model will be made using the SKlearn package, and evaluated based on accuracy and AUC for each class. A class based analysis will be performed on the results.

4.2.2 MLP

Next, we will use a multi-layer perceptron (MLP) [10] on generated multi-modal embeddings. The model consists of two blocks of linear layers with a dropout layer, finished with an multiheaded output layer, one head for each label to be predicted.

We decided to not experiment with a transformer model [11] because we precomputed fused multimodal embeddings for our models to use. The power of a transformer to learn the optimal representation of an input is therefore unnecessary.

4.3. Embedding Validation

4.3.1 k-NN

For each embedding vector in the test set, we ran the k-nearest neighbors algorithm [12] with a distance metric of cosine similarity, defined as follows:

$$\text{cos\_sim}(\mathbf{u},\mathbf{v}) = \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|\,\|\mathbf{v}\|} = \frac{\sum_{i=1}^{d} u_i v_i}{\sqrt{\sum_{i=1}^{d} u_i^2}\sqrt{\sum_{i=1}^{d} v_i^2}}$$

where:

- $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ are two d-dimensional vectors,
- $\mathbf{u}\cdot\mathbf{v}$ is their dot product,
- $\|\mathbf{u}\|$ and $\|\mathbf{v}\|$ are the Euclidean (L2) norms of $\mathbf{u}$ and $\mathbf{v}$, respectively.

We used this algorithm to find the 20 embeddings in the training set that were the most similar to each test embedding.

4.3.2 Goodall Distance

We wanted to incorporate the ground truth labels into evaluating the embeddings. We started by running k-nearest neighbors on the embeddings, clustering items of clothing based on their learned representation. Next, we retrieved the shape, fabric, and color labels for each image in a cluster and computed the pairwise Goodall distance [13] between each member.

The Goodall distance is for nominal categorical data, and it weights matches based on the rarity of a category value. Matches on rare values are considered more significant than matches on common ones, making it particularly useful for our multi-class label vectors with strong class imbalances.

$$d_G(\mathbf{x}, \mathbf{y}) = 1 - \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{x_i = y_i\}} \cdot \left(1 - P_i(x_i)^2\right)$$

Here, $P_i(x_i)$ denotes the empirical probability of observing category x_i in the i-th feature across the dataset. The distance decreases when matches occur on rare categories, better capturing the semantic structure of our discrete labels cosine distance.

Having a lower Goodall distance for a cluster shows that the articles of clothing in the cluster are more similar based on their descriptive features, thus a lower Goodall distance translates to a higher quality embedding.

4.4. Recommendations

4.4.1 Embedding Based Recommendation

Given a user's image and description of an item, the best performing embedding model returns a representation. The reference source, in this case the DeepFashion-MultiModal dataset, has already been embedded. Then, we return to the user the top $k = 20$ images based on cosine similarity across the entire dataset.

4.4.2 Classification Based Recommendation

Given a user's image and description of an item, this will be converted into an embedding with the same embedding model as above. However, this will then be passed to the best performing classifier, which returns a prediction vector. This prediction will then be compared across the ground truth shape-pattern-fabric label vector for the dataset, and then we return to the user this top $k = 20$ images based on cosine similarity.

We use the classifier here with the hopes of predicting the features themselves, thus not only giving a higher degree of interpretability to the resulting recommendations, but with the hopes of learning additional distinguishing features between items.

4.4.3 Evaluation

As judging fashion recommendations is nearly impossible to do algorithmically, we enlisted the help of members of Stanford FashionX's executive board to rate 50 images recommended by each image by quality on a scale of 1-10. This represents a stand-in for an expert visual diagnostic. The three judges will each be provided a Google Form of 100 image-caption queries from the dataset, fifty of which will have recommendations from the embedding-based recommender and fifty of which will have recommendations from the classifier based recommender. The judges will be blind to the model origin. The ratings will then be collected in a Google Sheet and analyzed.

5. Experiments

5.1. Classification

To correct for class imbalances in many of the labels, we used a weighted cross-entropy loss that weights based on the class count versus overall count. We also attempted to perform a stratified sampling of the dataset so that each class was equally represented, but that left us without enough samples to train on.

5.1.1 Linear Regression Model

The linear regression model was trained with the default sci-kit learn parameters and no regularization.

Sleeve Length	Lower Clothing Length	Socks	Hat
0.333	0.315	0.261	0.279

Table 1: Per Label Accuracy (part 1)

Glasses	Neckwear	Wrist Wearing	Ring
0.211	0.247	0.178	0.235

Table 2: Per Label Accuracy (part 2)

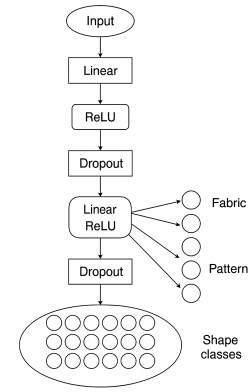
Waist Accessories	Neckline	Cardigan	Navel Covered
0.191	0.257	0.263	0.231

Table 3: Per Label Accuracy (part 3)

Color	Fabric Type	Overall
0.451	0.372	0.273

Table 4: Per Label Accuracy (part 4)

5.1.2 Multi Layered Perceptron



The model was trained for 10 epochs with cross-entropy loss and an AdamW optimizer [14] with a learning rate of 1e-4. These parameters were tuned manually by iterating on the model by hand.

Sleeve Length	Lower Clothing Length	Socks	Hat
0.7567	0.6648	0.7924	0.6683

Table 5: Per Label Accuracy (part 1)

Glasses	Neckwear	Wrist Wearing	Ring
0.7193	0.5509	0.5159	0.5586

Table 6: Per Label Accuracy (part 2)

Waist Accessories	Neckline	Cardigan	Navel Covered
0.4079	0.6930	0.6769	0.6086

Table 7: Per Label Accuracy (part 3)

Color	Fabric Type	Overall
0.4607	0.4567	0.6093

Table 8: Per Label Accuracy (part 4)

The MLP clearly outperforms the linear regression model on accuracy, and both of them outperform random chance. The MLP showed especially strong gains in categories like sleeve length, neckline, and navel coverage. However, performance remained lower for visually subtle or underrepresented labels, such as ring presence or waist accessories, likely due to persistent class imbalance and limited visual distinctiveness. Additionally, in many photos, only parts of the body were visible, with hands especially missing. This leads to a distinction between something being predicted as not visible because it is not possible to be visible and something being predicted as not visible because it is not there, which is not something our model accounts for. Only after viewing the recommendations based on the classified did we realize that the model only outputs three unique vectors of labels, meaning the MLP didn't actually learn the features. Instead, it simply learned to predict a generic and wide-ranging set of recommendations that minimizes loss. The only distinction it appears to be able to make is gender.

5.2. Embedding Validation

5.2.1 k-NN

For our first experiment on embedding validation, we computed the cosine similarity between each test embedding vector and its 20 most similar embedding vectors in the training set. Using these values, we averaged the cosine similarity across each test embedding's top-1, top-5, top-10, and top-20 nearest neighbors. Finally, we averaged these averages across every embedding in the test set, yielding the results seen in Figure 2.

Model	**Top-1**	**Top-5**	**Top-10**	**Top-20**
Zero-Shot ResNet/DistilBERT	0.936	0.928	0.924	0.920
Linear	0.925	0.899	0.881	0.859
Fine-tuned ResNet/DistilBERT	0.803	0.776	0.761	0.744
CNN/LSTM	0.995	0.991	0.989	0.985

Table 9: Avg. embedding similarity at different top-k accuracy levels.

Model	Avg. Cluster Goodall Distance
Zero-Shot ResNet/DistilBERT	0.7196
CNN/LSTM	0.6480
Linear	0.6600
Fine-tuned ResNet/DistilBERT	0.5990

Table 10: Average Goodall distance within clusters for different models

The withheld test set was embedded, then clustered using k-NN into 40 clusters so each cluster would on average have 100 samples in it. The labels for these clusters were then retrieved, and then the Goodall Distance was calculated.

5.2.2 Goodall Distance

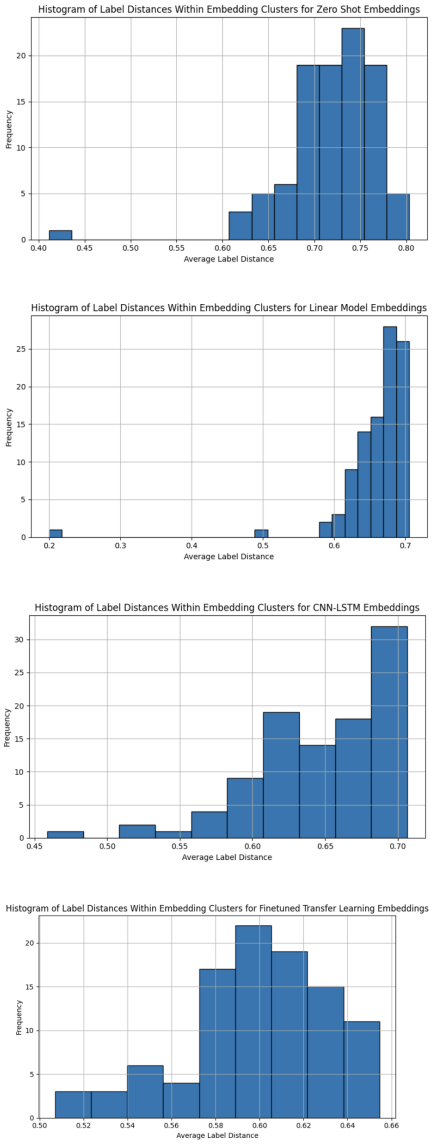


Figure 1: Distribution of average cluster distances by embedding model

As expected, the larger model size and powerful context of the fine-tuned transfer learning models led to the best performance, while the generic zero-shot model performed the poorest. We can also see that when training from scratch, using models that are better suited to their respective data in CNNs for images and LSTMs for text outperform the basic linear model.

Interestingly, the shape of the distribution changes when using a pretrained model versus training weights from scratch. We see that the pretrained models have a more normal distribution whereas when they are trained from

scratch, the models skew left. This is likely because the pretrained models come with an induced prior over the representation space because of their large context, leading to a spread-out embedding space. Thus, a more normal distribution emerges even with fine-tuning.

5.3. Model Recommendation

Figure 2: Example top 20 outfit alternative recommendations for a randomly selected input outfit with embedder-only method

Figure 3: Example top 20 outfit alternative recommendations for a randomly selected input outfit with classifier method

Method	Avg. Rating	Std. Dev.	Max	Min
Embedder	6.8	1.56	9	2
Classifier	2.2	0.44	5	1

Table 11: Summary statistics for method ratings.

The embedder-based recommender system was vastly preferred by the judges to the classification-based recommender system. One judge commented, "The recommendations seem to be based off a single characteristic and not based on the pieces themselves. Like if a shirt has a scoop neck and no sleeves and is orange, the model will recommend an image based on one of those characteristics."
The cause of the performance difference was that the classifier-based recommender system recommended the same top items for every query, meaning that the recommendation is not based off of the item at all, meaning the model isn't learning how to predict recommendation. In response to our judges' feedback, we looked at many recommendations, and found that the model only predicted three unique sets of clothing, and one of the sets dominated the majority of those predictions.

6. Conclusion

6.1. Recommendation Power

Based on our own qualitative analysis and our judges' verdicts, using the most similar embeddings to recommend similar clothing proves to have the best results. Despite attempts to correct the class imbalances, the model was unable to predict anything but a few discrete sets, suggesting that the classifier predicted a mix of generic clothing to minimize discrepancy in features. This suggests that the embeddings were able to capture the nuances of the clothing better than the vector of labels representing different features of clothing, meaning the embeddings were able to learn features of the clothing other than the ones represented in the labels. Although this was different from our expectations, in retrospect this makes sense, as the embeddings are able to give a complete representation of an item rather than boiling it down to just a few key indicators.

6.2. Similarity vs Quality

We expected the metrics for similarity and our stand in for quality in Gooddall distance to show similar results, instead there was a significant discrepancy. Instead, we observed a significant discrepancy between the two. Models that performed well on cosine similarity, like the zero-shot ResNet/DistilBERT, produced clusters that were worse when judged on Gooddall distance. Models like the zero-shot ResNet/DistilBERT are pretrained on large, general-purpose corpora and are not explicitly optimized to capture fine-grained fashion attributes like sleeve length or fabric type, especially because their layers were frozen. As a result, their embeddings likely group visually or semantically broad concepts together while sacrificing distance in their latent space.

6.3. Interpretability

A key motivation behind our classifier-based recommendation approach was to introduce interpretability into the

system. By predicting shape, fabric, and color attributes, we aimed to provide not just recommendations but also understandable explanations for why an item was recommended. An unfortunate consequence of using the higher-performance embeddings to drive recommendations is that we lose that interpretability. Especially in a highly personal and subjective domain like fashion, users are less likely to trust or rely on recommendations if they cannot understand why an item was recommended. Since our queries are multimodal, a user could theoretically tune the text portion of their query until they get better results, but without understanding the model's decision making processes, it becomes a game of guess-and-check. Therefore, when expanding this to an application, it would still be worthwhile revisiting a modeling-based approach from a user-focused perspective.

6.4. Future Works

Looking ahead, there are several promising directions to extend this work. One avenue is to integrate our recommendation engine into a real-world interface such as a browser extension or mobile app that scrapes fashion content and returns affordable lookalikes in real time from sites like Depop or Poshmark. Another is to enhance multimodal interaction by allowing users to refine search results filters based on predicted features like neckline, sleeve length, or fabric type. Additionally, we could incorporate reinforcement learning or preference-based re-ranking by collecting user interactions and feedback, allowing the system to adapt to individual tastes over time. Finally, expanding the dataset to include more stylistically diverse images and richer captions could further improve generalization and allow for recommendations across a broader spectrum of fashion genres.

References

[1] E. Park, S. J. Oh, and J. Kim, "Study on fashion image retrieval methods for efficient fashion visual search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[2] S. Goenka, M. Narasimhan, J. Weng, R. Mazumder, and D. Batra, "Fashionvlp: Vision language transformer for fashion retrieval with feedback," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17496–17506, 2022.

[3] G. Moro, S. Salvatori, and G. Frisoni, "Efficient text-image semantic search: A multi-modal vision-language approach for fashion retrieval," *Information Processing & Management*, vol. 60, no. 2, p. 103280, 2023.

[4] J. Y. Chia, T. Pham, M. Y. Tan, and Y. M. Teo, "Contrastive language and vision learning of general fashion concepts," *Scientific Reports*, vol. 12, p. 19737, 2022.

[5] 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," in *Proceedings of the 38th International Conference on Machine Learning*, 2021. arXiv:2103.00020.

[6] "daydream.ing." `https://daydream.ing/`, 2023. Creative project exploring generative fashion with AI.

[7] Y. Jiang, S. Yang, H. Qiu, W. Wu, C. C. Loy, and Z. Liu, "Text2human: Text-driven controllable human image generation," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–11, 2022.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 770–778, 2016.

[9] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[12] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[13] D. W. Goodall, "A new similarity index based on probability," *Biometrics*, vol. 22, no. 4, p. 882, 1966.

[14] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019.