Fashion Image Recommendation Model for Natural Language Search Queries

1. Introduction

A common challenge in online clothing shopping is the difficulty of finding specific items through search queries. This issue arises because most search algorithms rely heavily on fashion-related keywords rather than natural language-based retrieval. For instance, searching for "long-sleeved shirt with black and white boxes" may not retrieve results like a checkered sweater, even though it aligns with the query's intent.

To address this limitation, this report proposes a fashion image recommendation model designed to handle natural language queries for image-text retrieval. The objective is to develop a model that enhances the shopping experience by accurately aligning search results with user queries and providing a user-friendly interface. This approach aims to improve search accuracy through better image-text alignment while presenting visual results that match users' expectations.

The proposed solution utilizes the DeepFashion-MultiModal dataset (Jiang et al., 2022) for training and testing. The CLIP (Contrastive Language-Image Pre-Training) model (Radford et al., 2021) was selected as the baseline for this study, and several methods were explored to enhance its performance.

2. Methodology

To implement the proposed model, image-text pairs were extracted from the DeepFashion-MultiModal dataset. The dataset was pre-processed and divided into training, validation, and testing subsets to facilitate the implementation process. For the training phase, contrastive loss was selected as the loss function, while recall@k was used as the evaluation metric to assess model performance.

The baseline performance of the pre-trained CLIP model was first evaluated by retrieving images corresponding to given text queries. Based on this evaluation, several methods were explored to enhance the model's performance. These included fine-tuning the CLIP model using the DeepFashion-MultiModal dataset and incorporating cross-modal attention into its architecture.

The effectiveness of these approaches was analyzed through experimental results, and the findings were used to identify limitations. Additionally, potential methods for further improvement were discussed to guide future work.

2.1. Dataset

DeepFashion-MultiModal (Jiang et al., 2022) is a human fashion dataset with 44,096 full-body human images and their corresponding textual descriptions. We use this dataset for training and testing the model.

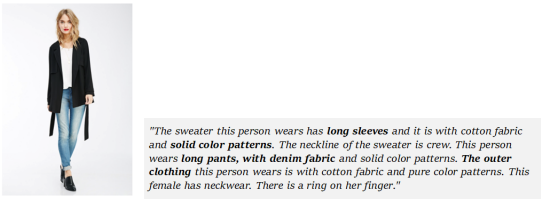

Figure 1: An example of an (image, text) pair from the DeepFashion-MultiModal dataset.

2.2. Experimental Setup

2.2.1. Data preprocessing

CLIP limits tokens to a maximum of 77 per text input. The queries in the DeepFashion-MultiModal dataset are often lengthy, as illustrated in *Figure 1*. Initially, we attempted to summarize queries using a pre-trained summarization model. However, this method proved time-consuming and inefficient, as most queries already fell within the token limit.

By inspecting the dataset using OpenAI's GPT-4 tokenizer, we found that most queries contained fewer than 77 tokens. For example, even longer queries such as the one shown in *Figure 1* contained 70 tokens. Although there may be slight differences in token counts between the tokenizer and CLIP's internal mechanism, we determined that significant length issues were unlikely.

Additionally, since this study focused on clothing, the truncated portions often contained non-essential details such as descriptions of jewelry (e.g., "neckwear" or "ring on her finger").

Consequently, we adopted truncation for efficiency without compromising model accuracy. After truncating the tokens, the data was split into train and test sets with a ratio of 8:2. Random seed was also fixed to ensure consistent results during testing.

2.2.2. Loss Function: Contrastive Loss

Contrastive loss is defined as the loss calculated as a function of the Euclidean distance between positive pairs and negative pairs. It is a loss function commonly used in learning cross-modal embeddings. The main purpose of

contrastive loss is to encourage pairs of matching images and text to have similar embeddings while pushing non-matching pairs further apart.

2.2.3. Evaluation Metric: Recall@K

The models were evaluated using Recall@K metrics, specifically Recall@1, Recall@5, and Recall@10, to measure the retrieval accuracy for each query.

$$Recall@K = \frac{\text{Number of Queries with Correct Match in Top-K}}{\text{Total Number of Queries}}$$

2.3. Baseline Model: CLIP

CLIP (Contrastive Language-Image Pre-Training) (Radford et al., 2021) is a neural network designed to process pairs of images and text. It employs a dual-encoder architecture comprising an image encoder and a text encoder, which are jointly trained using contrastive pre-training. This enables CLIP to align image and text representations within a shared embedding space. Notably, CLIP demonstrates the ability to predict the most relevant text for a given image without explicit optimization for the task, similar to the zero-shot capabilities observed in GPT-2 and GPT-3.

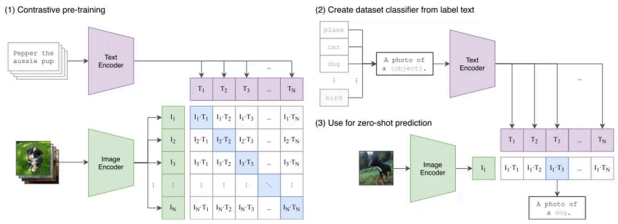

Figure 3: Architecture of CLIP

Several factors influenced the decision to use CLIP as the baseline model for this study. First, its ability to learn a shared embedding space between images and text makes it well-suited for multi-modal tasks such as text-to-image retrieval. Additionally, CLIP has demonstrated superior efficiency in zero-shot transfer compared to transformer-based language models, making it particularly advantageous for this application. Finally, its robustness to task shifts allows the model to generalize learned knowledge to new tasks, even when fine-tuned on smaller, task-specific datasets, such as the DeepFashion-MultiModal dataset used in this study.
Upon evaluating the Recall@k values of the pre-trained baseline model based on test data queries, the Recall@1, Recall@5, and Recall@10 are 0.0001, 0.0007, and 0.0020 respectively.

2.4. Problem Identification

The baseline CLIP model appears to have certain limitations in the context of fashion image retrieval tasks:

1. **Low Recall and Query-Specific Challenges**:
As shown in *Table 1*, the model seems capable of retrieving suitable images for simple queries such as "red dress," suggesting it performs reasonably well in text-to-image retrieval tasks. However, despite this, its recall values remain significantly low (e.g., Recall@1: 0.0001), indicating poor retrieval accuracy. Additionally, when tested with queries from the test dataset, the model retrieved visually similar images among the top results but failed to include the correct image within the top 50 results. This suggests that the model struggles with specific, nuanced fashion-related queries where multiple similar items exist.

Query	Top 5 Images Retrieved Based on Similarities to the Query Text (Baseline Model)
"a red dress"	
Test Set First Query (Refer to Figure 1 for details)	

Table 1. Top 5 Images Retrieved by the Baseline Model Based on Similarities to Query Texts

2. **Lack of Domain-Specific Training**:
CLIP was trained on a new dataset of 400 million (image, text) pairs collected from a variety of publicly available sources on the Internet (Radford et al., 2021). While this broad dataset enables generalization across various tasks, it lacks the domain-specific details necessary for high performance in specialized areas such as fashion image retrieval.

3. **Limited Cross-Modal Interaction**:
CLIP encodes images and text independently, which can result in weak alignment between the two modalities. This limitation seems to reduce the model's ability to effectively link the semantic relationships between text and image representations, which is critical for complex fashion-related queries.

2.5. Improved Model

2.5.1 Fine-Tuning with DeepFashion-MultiModal

As discussed earlier, the baseline CLIP model lacks specialization in fashion-specific domains, which limits its ability to accurately align textual descriptions with corresponding images. To address this limitation, we adopted fine-tuning with the DeepFashion-MultiModal dataset as the first improvement method.
The DeepFashion-MultiModal dataset provides detailed textual descriptions of fashion items, including attributes such as fabric types, patterns, and clothing styles. By fine-tuning CLIP on this dataset, the model gains a deeper understanding of fashion-related queries and improves its ability to handle domain-specific nuances.
Fine-tuning refines the model to focus on domain-specific features from the fashion dataset, addressing the baseline's limitations in handling natural language queries related to clothing.

2.5.2 Cross-Modal Attention Integration

A transformer-based cross-modal attention mechanism was implemented to enhance the interaction between image and text modalities. CLIP's dual-encoder structure processes images and text independently, which may lead to weak inter-modal relationships. To address this, cross-modal attention was introduced to merge the two modalities into a unified representation, capturing intricate dependencies. This mechanism employs a transformer encoder to process concatenated image and text embeddings, allowing the model to learn complex relationships and improve retrieval accuracy.

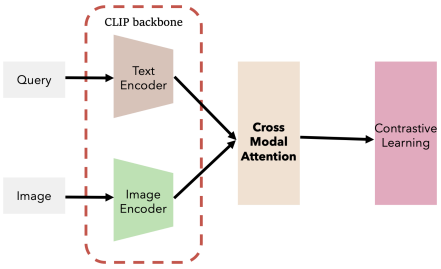

Figure 5: Cross-Modal Attention Mechanism Architecture

Technical Details:
First, image and text features are independently encoded using CLIP's encoders and normalized via L2 normalization to ensure consistency. A CLS token, initialized as zeros, is prepended to each modality's features to form distinct sequences. These sequences are then concatenated, creating a unified input representation for the transformer.
The transformer consists of six layers with eight attention heads, which learn both intra and inter-modal dependencies using multi-head attention mechanisms. Positional encodings are added to the concatenated input to maintain sequential relationships across embeddings. To initialize the transformer weights, the model was trained for one epoch on the DeepFashion-MultiModal dataset. The final CLS tokens for each modality are extracted from the Transformer's output and processed through a feed-forward network to generate the final representations. These embeddings are then compared using a contrastive loss function, which employs cosine similarity and cross-entropy loss in both image-to-text and text-to-image directions. Temperature scaling is applied to adjust the sharpness of similarity distributions.

2.6. Implementation Details

Common Setup (Baseline, Fine-Tuned, Cross-Modal Attention Models):
Architecture: CLIP ViT-B/32
Random Seed: 42 (set using torch.manual_seed(42))
Dataset Split: 80% training, 20% testing

Fine-Tuned and Cross-Modal Attention Models:
Learning Rate: 1×10^{-6}
Optimizer: AdamW
Temperature Parameter: 0.1
Epochs: 1

3. Experiments

3.1. Results

Model	Recall@1	Recall@5	Recall@10
Baseline (Pretrained CLIP)	0.0001	0.0007	0.0020
Fine-tuning with DeepFashion Dataset	0.0001	0.0011	0.0022
Adding Cross-Modal Attention	**0.0007**	**0.0018**	**0.0029**

Table 2. Comparison of Model Performance Across Different Retrieval Techniques

Table 2 presents the performance of the baseline model, the fine-tuned model, and the cross-modal attention-enhanced model.

3.2. Analysis

Among the proposed improvements, the addition of cross-modal attention yielded the highest Recall@K values for this fashion dataset. This suggests that integrating attention mechanisms effectively leveraged interactions between image and text modalities, enabling the model to better handle complex queries.
However, the recall values remain low overall. This can be attributed to the evaluation being conducted on the entire test dataset. For example, when the baseline model was tested with a smaller batch size of 32, the Recall@5 value reached as high as 0.5934. This suggests that reducing the evaluation scale within the dataset may lead to better recall performance.

4. Limitations and Future Work

4.1. Limitations

During the project, several limitations and challenges were encountered. Firstly, the number of training epochs was limited to just 1 or 2 due to hardware constraints. Initial attempts to train the model for 100 epochs using Elice resulted in GPU shortage errors, halting the training after 1 epoch. Training on Colab's CPU, requiring over 8 hours per epoch, was impractical.
Additionally, disk storage limitations in Elice (30 GB) were just sufficient to process the fashion dataset used but prevented experimentation with larger, more diverse datasets.

4.2. Future Work

To address these limitations and further improve the model's performance, the following improvements are proposed:
Expanding access to higher-performance GPUs would enable training for additional epochs, significantly enhancing model performance.
Lastly, instead of truncating long text queries, Segmenting and tokenizing long queries could preserve essential information while ensuring compatibility with model requirements.

5. Conclusion

The performance of the dual-encoder CLIP model for fashion text-image retrieval can be enhanced by incorporating cross-modal attention mechanism. This modification allowed better interaction between modalities, hence an overall improvement in the model's performance. Further improvements, as outlined in this report, should be explored to achieve more satisfactory results in future iterations.

References

[1] Jiang, Y., Yang, S., Qiu, H., Wu, W., Chen, C. L., & Liu, Z. (2022). DeepFashion-MultiModal. Retrieved October 13, 2024.

[2] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). *Learning Transferable Visual Models From Natural Language Supervision*. https://arxiv.org/abs/2103.00020

[3] *Contrastive Loss*. Contrastive Loss - an overview. (n.d.). https://www.sciencedirect.com/topics/computer-science/contrastive-loss

[4] Liu, H., Xu, S., Fu, J., Liu, Y., Xie, N., Wang, C.-C., Wang, B., & Sun, Y. (2021). CMA-CLIP: Cross-Modality Attention CLIP for Image-Text Classification.

[5] **OpenAI Tokenizer.** (n.d.). Tokenize text using OpenAI's Tokenizer tool. Retrieved from https://platform.openai.com/tokenizer