Image-to-Image Retrieval for Fashion Items

Jaeeun Lee (2022149023)

1 Introduction

In this project, I explore the task of image-to-image retrieval for fashion items, where the goal is to find visually similar items given a query image. Recently, online shopping platforms are heavily relying on visual search and retrieving similar items, which makes this an important task. By retrieving similar fashion items, we can provide users with similar products they prefer, find alternatives to out-of-stock items, and enhance their shopping experience through visual search capabilities. Rather than relying on text-based queries, users can simply upload an image of a fashion item they like and easily get recommendations for similar products. To address this problem, I implemented an AI-based image retrieval pipeline by extracting feature embeddings and ranking images based on similarity and compared this with a naïve baseline simply using color histograms. By evaluating both methods using precision@5, recall@5, and hit@5, I compare how much improvement is gained by using the proposed AI pipeline.

2 Task Definition

This project tackles the task of image-to-image retrieval for fashion items. Given a query image, the goal is to retrieve visually similar images that belong to the same style group. This task is crucial for visual search and recommendation systems in modern e-commerce platforms. In particular, it is useful when users want to find cheaper alternatives, discover complementary items, or find alternatives for out-of-stock items. The input is a single query image of a fashion item and the output is a ranked list of top-k most similar images from the database, ordered by their similarity to the query image. The system is considered ”good” if it successfully retrieves related items to the query image, meaning they belong to the same style group. To evaluate retrieval quality, I measure whether relevant items appear within the top-k retrieved results using retrieval metrics such as Precision@5, Recall@5, and Hit@5.

3 Methods

This section describes the naïve baseline and the improved AI pipeline used for the image-to-image retrieval task.

3.1 Naïve Baseline

3.1.1 Method Description

The naïve baseline uses a heuristic method by representing each image by a simple RGB color historgram. Each image is resized to 224×224 pixels and converted into a normalized 3D color historgram over the RGB channels. For retrieval, I compute the cosine similarity between historgram vectors, and the top-k most similar images are return for each query image. Images belonging to the same style category as the query image are treated as ground-truth positives.

3.1.2 Why it is considered naïve

This approach is naïve because it only uses the low-level visual cues and does not use any semantic representations extracted from the images. Therefore, the resulting histogram vector only captures the global color distribution of the image, without considering the high-level visual patterns like spatial structure and semantic content.

3.1.3 Expected failure cases

Since the naïve baseline only capture the color distribution, images with similar colors may be incorrectly retrieved even though they are visually distinct items. Therefore, the naïve baseline is expected to fail in the following cases.

1. Fashion items with similar dominant colors but different garment types or styles (e.g., a blue shirt vs. blue jeans).
2. Items where color is not a strong indicator of style (e.g., patterned or multi-colored clothing).
3. Cases where semantic similarity depends on shape, texture, or fine-grained details rather than color alone.

3.2 AI Pipeline

3.2.1 Method Description

The proposed method uses a pre-trained ResNet-18 [1] model trained on the ImageNet dataset as a feature extractor. The final classification layer is removed, and the global average-pooled feature vector is used as the image embedding. Using the image embeddings, cosine similarity is computed between the given query image and all other image embeddings. Images are ranked in descending order of similarity, and the top-k most similar images are retrieved as the final result.

3.2.2 Pipeline Stages

The AI pipeline consists of the following stages.

1. Preprocessing: Each image is resized to 224×224 pixels and images are normalized using ImageNet mean and standard deviation. Also, images are converted to tensors suitable for the neural network.
2. Embedding or representation: The pre-trained ResNet-18 model trained on the ImageNet dataset is used as an embedding model. ResNet-18 is a convolutional neural network consisting of 18 layers, including convolutional layers and residual blocks with identity skip connections. The final fully connection layer is removed, and the output of the global average pooling layer is used as the image representation. This is used as a fixed-length 512-dimensional feature vector that represents each image. The extracted feature vectors are normalized using L2 normalization to compute the similarity using cosine similarity during retrieval.
3. Decision/ranking component: For a given query image, cosine similarity is computed between its embedding vector and all other image embeddings. The images with the top-k similarity scores are returned as the final retrieval results.

3.2.3 Design Choices and Justification

I chose ResNet-18 because it can extract strong image representations while maintaining a good balance representational power and computational efficiency. Since, ResNet-18 is widely used in computer vision research, it is a stable and reliable feature extractor suitable for this small-scale project. The model uses ImageNet pre-trained weights, which allows it to leverage rich visual priors learned from large-scale data. I removed the fully connected layer, which is unsuitable for retrieval, and use the global average-pooled feature vector to capture high-level information from the input

image. I used cosine similarity as the ranking metric because it is both computionally efficient and well-suited for comparing high-dimensional feature vectors. Feature vectors are L2-normalized so that the comparisons are not affected by differences in embedding magnitude for stable retrieval. Therefore, compared to the naïve baseline, the features extracted from ResNet-18 can capture higher-level visual concepts such as shape, texture, and object structure, which are important for distinguishing fashion styles.

4 Experiments

4.1 Datasets

In experiments, I use the *Fashion Product Images Dataset* from Kaggle. The original dataset consists of 44k high resolution fashion product images along with corresponding JSON files that describe product attributes. Since the dataset does not provide ground-truth labels for image-to-image retrieval, I use the rich metadata that can be used to define style-level similarity. To construct retrieval ground-truth, I group the images into style categories defined by the combination of **ArticleType, BaseColour, and Usage**. For instance, as shown in Fig. 1, a category such as `(wallets, yellow, casual)` represents a group of purple casual belts. Style groups containing fewer than 4 or more than 20 images were excluded, and images were organized into folders corresponding to these categories.

Figure 1: Dataset Visualization

After preprocessing and filtering, the final dataset I constructed includes **6,217 images** grouped into **697 style categories**. As shown in Fig. 2, each style category contains between **4 and 20 images**, with an average of **8.92 images per category**. Categories with fewer than 4 images were removed to ensure stable retrieval evaluation, and categories with more than 20 images were removed to avoid overly large groups dominating the metrics.

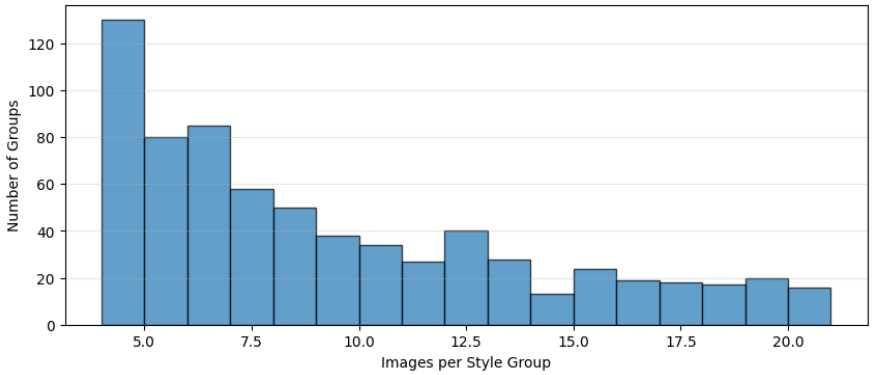

Figure 2: Distribution of Images per Style Group

Since this project does not involve training a model, dataset split is not required. Instead, I adopt a retrieval-oriented split where one image is randomly selected per style category (697 images) for

the query set , and all remaining images (5,520 images) for the gallery set. For each query image, images belonging to the same style category are treated as ground-truth positives, while all other images serve as negatives. This ensures that query images are not matched with themselves during evaluation.

4.2 Metrics

I evaluate quantitative results of the retrieval performance using three retrieval metrics that capture different aspects of retrieval quality.

- **Precision@5:** The proportion of relevant items among the top 5 retrieved results, reflecting retrieval accuracy
- **Recall@5:** The proportion of all relevant items in the dataset that appear in the top 5 results, reflecting retrieval completeness
- **Hit@5:** Whether at least one relevant item appears in the top 5 results, which aligns with user satisfaction in practical retrieval scenarios

Given that each style group contains an average of 8.92 images, I evaluate retrieval performance at $k = 5$ since top-5 evaluation is a balanced setting that is neither overly restrictive nor overly permissive.

4.3 Results

4.3.1 Quantitative Results

Table 1 and Fig. 3 compares the retrieval performance of the naïve color-histogram baseline and the ResNet-18–based AI pipeline. In case of precision@5, the AI pipeline achieves more than twice compared to the baseline indicating that the top-5 retrieval results are significantly more relevant. Recall@5 also increases considerably from 0.0516 to 0.1175, demonstrating that the model retrieves a larger proportion of relevant items overall. Hit@5 increases from 0.3305 to 0.5993, suggesting that the model returns at least one relevant item for a high proportion of queries. These results show that replacing heuristic-based features with visual embeddings extracted from pre-trained ResNet-18 leads to a substantial improvement in image-to-image retrieval performance.

Table 1: Retrieval Performance Comparison

Method	Precision@5	Recall@5	Hit@5
Baseline	0.0993	0.0516	0.3305
AI Pipeline	**0.2154**	**0.1175**	**0.5993**

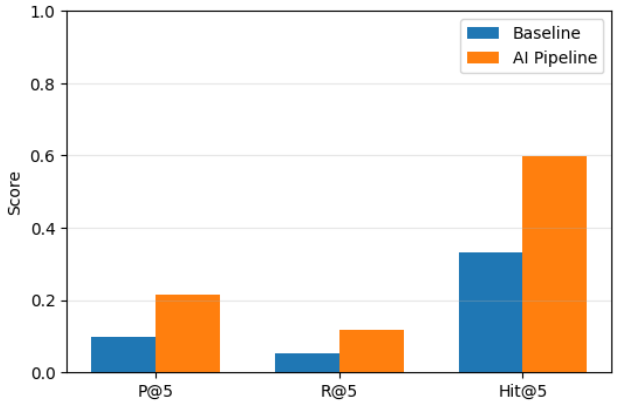

Figure 3: Retrieval Performance Comparison

4.3.2 Qualitative Results

Fig. 4 shows a failure case of the AI pipeline. For the query *backpacks_brown_casual*, the baseline retrieves at least one relevant backpack even though most results are incorrect. This suggests that for some cases global color cues can be strong enough to retrieve a correct item. Meanwhile, the AI pipeline retrieves backpack items correctly but with mismatched style-group attributes, leading to an overall failure. This indicates that the learned embedding can focus on category-level shape similarity while being less aligned with fine-grained style attributes encoded in the style group definition.

Figure 4: Case where the baseline outperforms the AI pipeline

Fig. 5 shows the opposite pattern from above. For the query in the *wallets_black_formal* style group, the baseline retrieves visually dark items such as bags, shoes, and apparel that share similar colors but are clearly different product categories and styles. In contrast, the AI pipeline retrieves multiple wallet items with similar structure and appearance, resulting in 4 correct hits within the top-5. This case demonstrates that learned embeddings provide more semantic and category-aware similarity than hand-crafted color features.

Figure 5: Case where the AI pipeline outperforms the baseline

Fig. 6 presents a challenging case for the query labeled *capris_navy_blue_casual* style group where neither method retrieves relevant items. The baseline retrieves items with dark colored items, including belts, footwear, bags, and accessories, revealing its failure to distinguish garment categories

beyond color distribution. The AI pipeline retrieves visually similar lower-body garments showing that the learned embedding captures shape and category-level similarity more effectively. However, the items are marked as incorrect because they do not exactly match the style defined by the metadata of the dataset. This is an limitation that comes from the ambiguity in the style definition.

Figure 6: Case where both pipelines failed

5 Reflection and Limitations

Overall, the ResNet-18-based AI pipeline worked better than the baseline by showing substantial improvements in precision@5, recall@5, and hit@5. This indicates that pre-trained visual features capture meaningful semantic similarity for image-to-image retrieval. The baseline was weak as predicted because it only encoded the color information, but sometimes succeed when color cues where highly information.

A major difficulty was that the model was never fine-tuned for image-to-image retrieval for fashion items. Since the ResNet-18 encoder only provides generic visual representations, it has no way to internalize how the dataset defines the style groups or which attributes should be prioritized. Therefore, the pipeline often retrieves items that look semantically reasonable, but fail under the dataset's specific grouping rules. Another important limitation is that the ground-truth positives were derived from metadata-based style groups (ArticleType + BaseColour + Usage), which do not always match human perception of visual similarity. Therefore, similar to the first limitation, the AI pipeline sometimes retrieve visually similar items that were still counted as incorrect.

Assuming the style groups of the dataset are correctly defined, the metrics provide an objective and reliable measure of retrieval quality. Precision@5, recall@5, and hit@5 successfully capture whether the model retrieves items that belong to the intended style group, making them well aligned with the task's formal definition of relevance. However, the metrics cannot account for perceptually similar items that fall outside the style labels based on the metadata, even when users would consider them reasonable matches.

With more time or compute, I would explore stronger representation models such as ViT-based encoders and apply contrastive fine-tuning to better align the embeddings with the pre-defined dataset's style group. I would also experiment with re-ranking strategies that combine visual similarity with metadata cues and refine the style grouping rules to reduce ambiguity and improve alignment between model outputs and evaluation metrics.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.