Content-Based Image Retrieval (CBIR) Systems for Fashion Product Images Dataset

Ceren Arkaç*, Barış Aygen†, and Sena Yapsu‡
*Data Science, Sabancı University, İstanbul, Türkiye
Email: cerenarkac@sabanciuniv.edu
†Computer Science, Sabancı University, İstanbul, Türkiye
Email: hbaris@sabanciuniv.edu
‡Data Science, Sabancı University, İstanbul, Türkiye
Email: ysena@sabanciuniv.edu

***Abstract*—This study presents a Content-Based Image Retrieval (CBIR) system designed for fashion product image datasets, addressing the limitations of metadata-based search methods. The proposed CBIR framework leverages classical feature extraction techniques, including color histograms, Fourier descriptors, and Local Binary Patterns (LBP), separately from deep learning-based feature extraction using a pre-trained ConvNeXt model. The system retrieves visually similar images by comparing extracted numerical representations, reducing reliance on manual metadata tagging. The evaluation was conducted using a modified Precision@5 and Normalized Discounted Cumulative Gain (NDCG@5), with weighted adjustments applied to both image features and metadata labels. The results demonstrate that combining classical methods with optimized weights enhances image retrieval accuracy, without the inclusion of deep learning techniques while maintaining computational efficiency. This hybrid approach offers a scalable solution for personalized fashion product recommendations, improving the quality of visual search results on digital platforms.**

I. Introduction

Advancements in imaging and communication technologies have significantly increased the volume of digital images, creating challenges to efficiently locate specific images within large datasets. This surge in digital image storage demands more effective search tools, making image retrieval systems increasingly essential. Content-Based Image Retrieval (CBIR) systems have emerged as a powerful solution as they compare features such as color, shape, and texture between query and database images to identify visually similar content. By focusing on extracted feature vectors rather than raw pixel data, CBIR systems reduce computational costs while enabling faster and more accurate image comparisons using compact and semantically rich representations.[1].

Modern CBIR techniques are mainly categorized into classical feature extraction methods, which rely on manually extracted features, and deep learning approaches, which use neural networks to automatically learn rich image representations. Classical methods, such as color histograms, Fourier descriptors, and Local Binary Patterns (LBP), focus on feature extraction methods that are explicitly designed and engineered by humans based on domain knowledge and mathematical formulations. The single feature is difficult to fully characterize the image information. Therefore, researchers often combine different features to increase the correctness of image retrieval [2].On the other hand, deep learning methods mostly use pre-trained Convolutional Neural Networks (CNNs) to learn complex high-level features from the data. As highlighted by Ahmed[3], pre-trained CNNs have demonstrated significant performance improvements in CBIR tasks, offering robust feature extraction capabilities that can generalize across diverse datasets.

The arrival of Convolutional Neural Networks (CNNs), using local features, for instance-level retrieval, has become more significant. Global features such as color, texture, and shape, local features capture the details of an image, provide more complex features to work which presents similarity to human perception[4]. This project aims to explore both classical and CNN-based approaches for CBIR in the context of fashion product image retrieval, delivering visually accurate and efficient image recommendations.

To effectively evaluate an image retrieval system, it is crucial to consider the specific task and requirements, ensuring the selection of appropriate evaluation metrics for an accurate assessment[4]. Our model employs Precision and Normalized Discounted Cumulative Gain (NDCG) metrics to assess performance and guide tuning. It incorporates data labels with designated weights, reflecting their importance in the calculation of both Precision and NDCG scores. The weighted scores are then combined and normalized for a comprehensive evaluation.

II. Problem Description

Despite advancements in image retrieval systems, the fashion industry faces unique challenges in developing personalized recommendation engines. Fashion products are highly visual, with user preferences often shaped by subtle visual features such as patterns, textures, and colors. Traditional systems relying on manual metadata tagging struggle with inconsistency, subjectivity, and scalability, which can lead to less effective recommendations.

This project addresses the need for a more reliable, automated image retrieval system by focusing on Content-Based Image Retrieval (CBIR) methods. CBIR systems compare image features directly, reducing the reliance on

manually labeled data. However, the complexity of fashion datasets—characterized by diverse styles, categories, and varying product representation—introduces challenges in balancing accuracy and efficiency.

Our goal is to develop a CBIR system adapted to fashion image retrieval that minimizes manual annotation while enhancing recommendation quality. By integrating classical feature extraction methods, such as color sub-blocks and Local Binary Patterns (LBP), with deep learning approaches like ConvNeXt, the project aims to create a balanced and effective solution for delivering visually accurate product recommendations in a scalable and efficient manner.

III. Methods

A. Dataset Description and Preprocessing

The dataset used in this project consists of 44441 product images accompanied by metadata in CSV format. Each row in the metadata corresponds to an individual product in the image dataset, identified by a unique product ID. The product ID also matches the filename of the associated image. The dataset offers two versions, differing only in image resolution. Initially, we worked with the lower-resolution dataset (60x80 pixels); however, after delivering the progress report, we transitioned to the higher-resolution dataset to improve image quality for analysis. We resized the images to 224x224 pixels to ensure compatibility with ConvNeXt V2 model. Descriptions of each attribute in the metadata are provided in the table below.

TABLE I
Metadata Features in the Dataset

Attribute	Description	Data Type
id	A unique identifier for each product.	Numeric
gender	The target audience of the product, such as Men, Women, Unisex, Girls, and Boys.	String
masterCategory	The high-level category of the product, such as Apparel or Accessories.	String
subCategory	A more detailed categorization under the master category, such as Topwear, Bottomwear, or Shoes.	String
articleType	The specific type of a product, such as T-Shirts, Shirts, or Handbags.	String
baseColour	The color of the product.	String
season	The season for which the product is designed, such as Summer or Fall.	String
year	The year that the product was released. The range is between 2007–2019.	Numeric
usage	The intended usage, such as Casual, Sports, or Formal.	String
productDisplayName	A descriptive name of the product, often including the brand name.	String

The initial data preprocessing involved removing products with an unusual number of attributes from both the metadata and image datasets. Additionally, only the intersecting product IDs between the metadata and image file names were retained, and the datasets were sorted by ID. This process resulted in a total of 44,419 unique products. To address the imbalance in class distributions, necessary adjustments were made to ensure it would not negatively impact retrieval performance. The details of the evaluation and further preprocessing steps will be explained *Evaluation*.

B. Representation of Image Content

The content of an image can be defined as its distinct features. Those distinct features are mainly the visual features of an image, which are color, shape, texture, and interest points. A CBIR process comprises three stages: feature selection, extraction, and representation. Feature selection should be based on the characteristics of the image dataset itself. For example, for a fabrics dataset, one should consider color and texture features of the images since they will provide more information than shape features would do. In the following lines, we will share background information for three essential features of images, color, shape, and interest point descriptors upon the research and experiments we conducted.

1) Shape: Shape feature extraction is a critical step in many image analysis and object recognition tasks. In this project, several methods for extracting and comparing shape features were implemented and evaluated, referencing established approaches from literature. The goal was to identify the most effective method for capturing and analyzing shape properties in the given dataset. In the article of Patel, Desai and Bhavsar, there are three shape methods proposed [5]. First method is Fourier descriptors which are derived by applying the Fourier Transform to the boundary of an object's shape. This method transforms the spatial domain boundary representation into a frequency domain representation, capturing essential shape details in the Fourier coefficients. By retaining only the low-frequency components, Fourier descriptors provide a compact and robust representation that is invariant to translation, scaling, and rotation. Fast Fourier Transform (FFT) enhances computational efficiency, making this method particularly useful for automated shape analysis. The Second method is geometric moments which are a form of weighted average calculated as a function of the intensity of image pixels. They are widely used in fields such as image processing and object segmentation. Geometric moments describe properties such as the overall strength, orientation, and center of gravity of an object, offering a holistic perspective of its shape. After image segmentation, GM can effectively describe an object's characteristics, providing meaningful features for comparison. The last method proposed is Algebraic Moment Invariants which are derived from the eigenvalues of matrices computed using the central moments of an object. This method considers the initial primary moments and formulates moment invariants that are robust to transformations such as translation and scaling. AMI has been noted for its variable performance depending on the configuration of the object's outline. When applied effectively, it can provide a distinctive and compact representation of an object's shape. In this project, AMI was implemented and found to perform exceptionally well in distinguishing between shapes, yielding the best results among the methods tested.

Through experimentation with the above methods, Fourier Descriptors emerged as the most effective approach for shape extraction and comparison in the context of this project. Using Euclidean distance for similarity measurement, Fourier

Descriptors consistently outperformed AMI and Geometric Moments in terms of distinguishing similar shapes from the dataset. In addition to that, there is a work proposes the over performance of Fourier Descriptors over other methods [2].

2) Intereset Point Descriptors: The widely used methods in the extraction of interest point descriptors are mainly Scale Invariant Feature Transform (SIFT), Speed Up Robust Feature Transform (SURF), and ORB (Oriented FAST and Rotated BRIEF). SIFT is known for its accuracy and robustness under scale, rotation, and illumination changes. One of its drawbacks is its being computationally expensive. SURF offers faster processing by simplifying SIFT's operations, however, it still requires significant computational resources. On the other hand, ORB provides a lightweight and efficient alternative, combining the speed of FAST for keypoint detection with the simplicity of BRIEF descriptors[6]. In our experiments, we employed ORB in BoF (Bag of Features) method which is used for feature representation.

3) Texture: Texture is a fundamental visual attribute in image analysis, playing a pivotal role in Content-Based Image Retrieval (CBIR) systems. Various techniques have been developed to extract texture features, each tailored to specific applications and datasets. Different texture feature extraction methods offer distinct advantages depending on the application context. Gray-Level Co-occurrence Matrix (GLCM) captures spatial relationships between pixel intensities, providing statistical measures such as contrast, correlation, and entropy [7]. Gabor filters, known for their ability to capture spatial frequency information, are widely used in texture analysis for their robustness to orientation and scale variations. Wavelet transforms decompose images into frequency subbands, enabling multi-resolution texture analysis. MPEG-7 Texture Descriptors provide standardized, efficient representations for multimedia applications. Each of these methods has demonstrated effectiveness in specific domains [8].

Local Binary Patterns (LBP) is one of the most extensively used techniques for texture analysis due to its computational simplicity and robustness. The LBP operator compares the intensity of a central pixel to its surrounding neighbors, encoding local texture variations as binary patterns. These binary patterns are aggregated into histograms, forming compact texture descriptors.

LBP has been effectively applied in diverse CBIR systems. For instance, studies demonstrated its success in constructing feature vectors for ultrasound and capsule endoscopy image databases [9]. These studies optimized the LBP-based indexing process using clustering techniques, achieving competitive retrieval performance. Similarly, integrating LBP with shape and color features has yielded impressive results on benchmark datasets like COREL and CIFAR-10, showcasing its adaptability to various domains [10].

To address limitations in invariance to rotation and scaling, several LBP variants have been proposed. Multi-Block LBP, for example, computes average intensities over blocks before applying the operator, while Center Symmetric LBP reduces computational complexity by focusing on diagonally opposite pixel pairs. These adaptations enhance the versatility of LBP in different retrieval scenarios [8].

In our project, LBP was chosen as the primary texture descriptor for CBIR due to its efficiency in capturing local texture variations, a critical factor for fashion products. The discriminative power of LBP allows effective differentiation between intricate fabric patterns, aligning well with the objectives of visual similarity retrieval. Experimental evaluations further validated its suitability, achieving competitive performance compared to shape and color-based descriptors.

4) Color: Feature extraction for color of images were studied extensively for years. We revised several papers and compared the successes of different methods on our dataset. We will go through of each method in this part of our report. In their review, Srivastava et al.[4] reference the introduction of color moments by Stricker and Orengo (1995), highlighting its statistical approach to capturing color distributions. The first order-moment, mean, points out the average color. The second-order moment, variance, indicates the spread of the color distribution, while the third-order moment, skewness, represents the asymmetry of the color distribution. We extended this approach a bit further and implemented the approach followed by Ahmed[3]. This approach extracts 18 color features based on six color moments for HSV channels. Although this method is good at detecting colors, it poorly performs when we choose a query image with a simple color-based pattern like striped t-shirts. Another common approach for detecting color features of an image is computing color histograms. Color histograms, which was firstly revealed by Swain and Ballard, are based on generating three distinct histograms, each corresponding to one of the primary colors: red, green, and blue. To represent the color distribution, the method quantizes the colors into discrete bins. Increasing the number of bins enhances the histogram's ability to differentiate between images [4]. However, during our experiments, where color histograms were computed for several images using different bin sizes, the most accurate matches (evaluated visually) were observed when the bin size was set to 8 for all channels. In contrast, using larger bin sizes often resulted in poorer outcomes. A notable advantage of using color histograms is their invariance to small changes in scale and rotation. However, due to lack of any spatial information in color histograms, two distinct images can share even the same histogram. When we compared the performance of standard color histograms and color moments, we observed that they suggest similar images, while color moments gives slightly more relevant images. Calculating color distribution entropy is one way to extend the information contained in color histograms. It calculates the entropy of the color histogram, which is essentially a summary of how frequently different colors occur in the image. Unfortunately, it was inadequate even for retrieving fashion product images that had similar colors. All methods mentioned above provide global color properties of images while failing to provide any spatial information about an image. To solve this issue, several methods were proposed in the literature. One of the prevalent methods for color-spatial models is computing

color correlograms. Since it is a computationally expensive method, auto-correlograms are preferred over them, where only the main diagonal of the co-occurrence matrices is calculated and stored. Moreover, computation of color auto-correlograms is still not a computationally efficient approach. For this reason, we didn't prefer using auto correlograms in our project. Instead, we sought different approaches to include spatial information of colors in images. Srivastava discusses two alternative methods, namely the Color Coherence Vector (CCV) and the Color Feature Extraction of Sub-blocks. In CCV method, pixels are classified as coherent if they belong to a large connected component, and non coherent otherwise. In our experiments, we preferred to implement color feature extraction of sub-blocks methods, which is relatively easy to implement. The method divides an image into smaller sub-blocks and extracts color histograms from each block. The number of sub-blocks is determined by the selected grid. For example, a grid of size 4x4 forms 16 sub-blocks. We experimented with different grid sizes and found that 4x4 is a good choice for the images in our dataset. Among four methods (standard color histogram, 6 color moments for each 3 channel, color distribution entropy, color histogram of sub-blocks), the last method performed a far better than the other methods. It has kind of a power to recognize shape features. For instance, when the query image is a black cap, it retrieved a black cap for the first place, and a green cap to the second place. Also, for another query image which shows a t-shirt with a Puma print, it suggested a ball with Puma print for the fifth place. During the experiments for color feature, we used euclidean distance to compare feature vectors. The reason for our choice over cosine similarity relies on working euclidean distance faster than cosine similarity and the outcomes are at same degree of success.

C. Feature Extraction

In this project, we aimed to perform a Content-Based Image Retrieval (CBIR) task on a fashion products dataset by retrieving visually similar images through the computation of distances between their numerical representations. Feature extraction plays a critical role in this process, as it defines how images are transformed into numerical representations suitable for comparison. We employed two approaches for feature extraction: (1) classical methods of feature extraction from images, such as computing color histograms and Fourier descriptors, and (2) extracting image embeddings from ConvNeXt V2, which is a pre-trained convolutional neural network (CNN) introduced by Liu et al. [11], which employs a co-design strategy with masked autoencoders to optimize the performance and scalability of convolutional networks, offering state-of-the-art efficiency in feature extraction.

The classical methods of feature extraction employed in this project involve the computation of distinct features such as color, shape, texture and ORB descriptors to identify visually similar images within the dataset.

To capture color features, we utilized color histograms computed over sub-blocks of the images. This approach ensured a detailed representation of color distribution while maintaining spatial information about the colors. Shape features were extracted using Fourier descriptors, which allowed us to represent image boundaries in the frequency domain. This method provided robustness against transformations such as scaling, rotation, and translation, while retaining critical shape details. For texture features, we implemented Local Binary Patterns (LBP), which effectively captured local texture variations and patterns. Interest point descriptors were extracted using the Oriented FAST and Rotated BRIEF (ORB) method.

Initially, we used each of the feature extraction methods separately to represent the images. These features were then used to calculate cosine similarity between the query images and the images in the entire dataset. For each query image, we retrieved the top five most similar images based on the calculated similarities. The retrieval performances of each feature extraction method was evaluated visually by printing the retrieved results for randomly selected query images, highlighting the effectiveness of each feature in isolating visually similar images.

Subsequently, we combined the four features (color, shape, texture, and ORB descriptors) into a unified representation for each image. By calculating cosine similarity on the combined feature set, we again retrieved the top five similar images and evaluated the performance. This combination allowed us to leverage the strengths of each type of features for more robust similarity detection.

For the second approach, which involves extracting features from a pre-trained CNN, we first divided the dataset into training, validation, and test sets using a stratified splitting strategy to preserve the class distribution, with split ratios of 70%, 15%, and 15%, respectively. Feature extraction was performed on the training set using the pre-trained weights of ConvNeXt V2 where the pre-trained model weights derived are from training on the ImageNet-1K dataset using a Fully Convolutional Masked Autoencoder (FCMAE) under a self-supervised learning paradigm. Subsequently, the trained model was used to extract features from the validation and test sets independently, ensuring no data leakage. It is important to note that the validation set was reserved for potential hyperparameter tuning in the context of model training for a classification task. In fact, we trained XGBoost models to predict class labels for a given query image using features extracted from ConvNeXt V2, however, they performed poorly even with hyperparameter tuning. The model definition we preferred is ConvNeXt V2-Pico where the number of parameters is 9.1 million due to our limited resources. The image features are embedded into vectors of size 512.

D. Evaluation

To evaluate the image retrievals, we needed the ground-truth set, which is the set of images that are similar to a query image, for each image both in the retrieval database and the query set (test set). Due to the high number of images, we had to create the ground-truth sets using the metadata instead of curating them visually. The images for

the products whose attributes align with the attributes of the query image are identified as ground-truths for that image. However, this approach does not incorporate any order within the created set of ground-truth images. Therefore, we had to use a retrieval evaluation metric that does not take the order of the retrievals into account. Precision@K complies with the limitation we are faced with, leading us to choose it over well-known retrieval evaluation metrics such as ANMRR (average normalized modified retrieval rank) and mAP (mean average precision). In literature, CBIR systems are often evaluated by those well-known metrics since they assess both relevance and ranking quality. These metrics require the ground-truth images for a query to be ranked in order of similarity, where the most similar image is ranked first. Precision@K measures the proportion of relevant items in the top K items retrieved by the system. For example, to calculate Precision@5, one should determine how many items retrieved by the system are in the set of ground-truth images for the query image. For our project, the only feasible way to obtain the ground-truth images for a query images is filtering the metadata by the "articleType", "baseColour", "usage", and "gender" labels of the query image. We analyzed the unique values for those attributes and realized that there is a severe imbalance problem with all attributes determined, which reduces the size of the ground-truth sets for many query images to under 5. To address this issue, one of the strategies we followed is combining some similar classes into one class. For example, the "Clutches" and "Handbags" classes of "articleType" might be combined into "Handbags". Another strategy is not using the products belonging to rare classes of articleType, since articleType is the most important feature for the evaluation of a retrieval and it is the attribute having the highest class variability in the metadata. Another strategy is distributing the examples of the most frequent classes of articleType into other classes. For example, we observed that "Tops" are mostly tshirts for women, which drives us to label "Tshirts" for "Women" and "Girls" as "Tops". Due to high imbalance, some attribute combinations were barely existing. Therefore, we detected some attribute combinations and removed those products from the datasets if the number of occurrences are less than 6. For instance, "Heels" and "Men", "Sunglasses" and "Boys" pairs were appearing only once in the datasets, therefore, we removed those products. For "usage", we combined "Smart Casual" to "Casual" and "Travel" to "Sports" classes. To reduce the number of classes under "baseColour", we combined rare color into more common colors while taking our decisions by checking the images carefully. After those preprocessing steps, there were 20 lines having missing values for either "baseColour" or "usage". We filled those values manually by checking the images of the products. At the end, the metadata was comprised of 38092 products. The images dataset remained same, however, we excluded the images that are not matching the products in the metadata. To score the performance of retrieval, we created a test set of 5714 images which comprises the 15 percent of the preprocessed metadata, having the same articleType distribution. To ensure that each image in the test set has at least 5 similar images, we augmented 147 images so as to complete the number of ground-truth images to 5. In total, 786 product records were inserted into both the images dataset and the metadata. The applied augmentation methods include flipping images horizontally, rotating with random degrees, increasing contrast, adding slight color enhancements, and blur effect. The augmented images are recorded into metadata with the product ID indicating the augmentation technique applied to the original image. After the augmentation, we were able to ensure that each image in the test set includes at least 5 similar images.

Due to the binary nature of precision@K, the scores were low although the performance is better visually than the scores indicate. Therefore, we modified Precision@K for our project. Our project focuses on recommending visually similar photos with matching patterns and colors on social media feeds. The dataset contains valuable metadata such as Gender, Master Category, Subcategory, Article Type, Base Colour, and Usage, which are instrumental in evaluating the similarity of retrieved images to the query image. To align the evaluation with our project's objectives, we modified the precision metric to consider all these labels.

In our modified precision metric, each label category (e.g., Gender, Master Category, Subcategory, etc.) is assigned a weight reflecting its importance in the evaluation. A relevance score is computed for each retrieved image by comparing its labels with the query image's labels. If a label matches, its weight contributes to the score; otherwise, it contributes nothing. For example, if the query image and a retrieved image share the same Gender and Article Type, the sum of the weights for these labels is added to the relevance score. For the top K retrieved images, the weighted relevance scores are summed and averaged by K, measuring the average weighted relevance of the top K images retrieved by the system.

Although we normalize precision values, the minimum and maximum scores are already within the range of 0 to 1, so normalization does not change the results. To achieve better precision scores, we implemented two separate fine-tuning steps. First, we adjusted the weights of the label categories used in the precision metric. Initially, these weights were assigned equal values summing to 1. Similarly, the features used in similarity calculations, such as ORB, shape, texture, and color, were combined with equal weights summing to 1. This initial setup was used to run the model and collect baseline results.

Next, we fine-tuned the label weights using a subset of 20 images. By iterating with different weight configurations, we identified the optimal weights that yielded the best precision scores, as shown in Table II.

Using these optimized label weights, we recalculated similarity results and recorded the outcomes. In the second fine-tuning step, we adjusted the feature weights while keeping the optimized label weights constant. Using the same subset of 20 images, we iteratively fine-tuned the feature weights and identified the optimal configuration, as shown in Table III.

These optimized weights improved the model's ability to

TABLE II
LABEL WEIGHTS FOR PRECISION CALCULATIONS

Label	Weight
Gender	0.1
Master Category	0.1
Subcategory	0.3
Article Type	0.2
Base Colour	0.2
Usage	0.1

TABLE III
FEATURE WEIGHTS FOR COMBINED REPRESENTATION

Feature	Weight
Color	0.55
Texture	0.30
Shape	0.10
ORB Descriptors	0.05

retrieve visually similar images, ensuring consistency between the evaluation metrics and the quality of retrieved images.

Additionally, we employed Normalized Discounted Cumulative Gain (NDCG) to evaluate the retrieval system's effectiveness. Unlike simple precision metrics, NDCG considers both the relevance and the order of retrieved items, making it particularly suitable for ranking tasks. Difference from our precision metric is, its emphasize on the placement of highly relevant items in higher ranks. It decides the ranking by the relevance score of each retrieved result image. NDCG aligns with our goal of providing users with the most relevant fashion products at the top of the results. It measures the quality of a ranked list by considering the relevance of retrieved items while penalizing their positions in the ranking. The metric starts with the computation of Discounted Cumulative Gain (DCG), which assigns higher weights to relevant items appearing at higher ranks through a logarithmic discount factor. NDCG normalizes this score by dividing it by the Ideal DCG (IDCG), representing the best possible ranking. This normalization ensures the scores are bounded between 0 and 1, enabling meaningful comparisons across queries or systems.

E. Related Work

In the literature, content-based image retrieval tasks are generally carried out using domain-specific image datasets. Regarding "fashion product images" datasets, there are no journal papers that use our dataset. However, there are a few GitHub projects which are non-extensive and there is a preprint that was created for a course project. Among the GitHub projects, one aims at a classification task for articleType labels of query images, using transfer learning to train a classifier. This project handles imbalanced data by undersampling it. The second project trains a classifier for articletype. It handles the imbalanced dataset problem by data augmentation. The third project predicts masterCategory and subCategory given a query image, using a CNN.

The preprint created for a course project uses neural network architecture for a classification task using transfer learning with pre-trained models such as VGG19. Additionally, they use CNN-based and ResNet-based autoencoders for visual search. Data augmentation is used to handle imbalanced data. However, they do not use an evaluation metric to evaluate visual similarity search results. Therefore, our project studies on visual similarity search for fashion product images dataset, which do not have similarity search projects in the literature for this dataset. Moreover, the existing study does not include the evaluation results for the implementation of the similarity search numerically[12].

IV. RESULTS

A. Experimental Design and Questions

This study aims to address the following questions:

1) How effective are classical feature extraction methods (color, texture, shape, ORB descriptors) in retrieving visually similar images?
2) Does combining multiple feature types enhance retrieval accuracy compared to using them individually?
3) What impact does weighting features or labels have on improving retrieval results?
4) Can the classical methods compete with deep learning-based approaches in image similarity tasks?

The experiments focused on retrieving the top five visually similar images for query images. The performance was evaluated using modified Precision@5 and NDCG@5 metrics.

B. Observations and Results

1) Individual Feature Performance: Color histograms extracted from image sub-blocks using a 4x4 grid yielded the best performance among color-based approaches. Cosine similarity provided fast and reliable similarity measurements, with Precision@5 values demonstrating consistent relevance for images sharing dominant colors. Local Binary Patterns (LBP) effectively captured local texture variations, delivering consistent results for queries involving textured items such as striped t-shirts or patterned fabrics. For shape features, Fourier descriptors emerged as the most robust, offering invariance to scaling, rotation, and translation, while ORB descriptors provided complementary but less impactful information when used alone. Visual results of the similarity measurements can be seen in Fig 1-8.

2) Combined Feature Performance: Combining features significantly enhanced retrieval accuracy. First, feature vectors are combined by giving them equal weights. Then label weights are fine-tuned for the model. Precision calculation requires giving weights to labels, so that their correctness will affect the precision result in different ratios. Before fine-tuning, they are used at the same ratio but after fine-tuning with 20 images, best weights appeared as (gender: 0.1, masterCategory: 0.1, subCategory: 0.3, articleType: 0.2, baseColour: 0.2, usage: 0.1). Fine-tuning the label weights not only increased the precision and NDCG but also sometimes decreased them to make numbers more accurate in terms of visual similarity. Improvement in matching the query image attributes performance metric with fine-tuning the labels led better results, especially

in scenarios involving underrepresented categories. In addition, weights of features are optimized by fine-tuning the model with 20 random images from dataset with using best label weights. Algorithm claimed that (Color: 0.65, Texture: 0.30, Shape: 0.05) achieves the best results. As a result, it can be seen that combining features and fine-tuning weights improved the similarity prediction performance, make it more reliable to use scores to calculate system's retrieval accuracy. Using both precision and NDCG provided some insight analyzes about the order of the retrieval images.

3) Comparative Analysis with Deep Learning: While classical methods provided competitive results, integrating CNN embeddings into our framework posed several challenges. Despite the reputation of ConvNeXt V2 in literature, the embeddings we obtained from it did not exceed the retrieval performance of classical methods. Moreover, the performance has decreased significantly when both metrics are considered. Some sample retrieval results are shown in Fig. 15-17. This outcome may be attributed to factors such as the dimensionality of the image embeddings or potential misalignment between the features extracted by the pre-trained model and the specific characteristics of the dataset. On the other hand, classical methods excelled in scenarios involving simpler visual patterns or distinct shapes, showcasing their utility in specialized use cases.

C. Visual Results

Sample query images and their retrieved top-5 results are presented in Fig. 1-14. Precision and NDGC metrics are giving relevant results and also it can be visually seen that there are similar shape, color and texture patterns in the retrieved images. The results prove the combination of classical methods and weighting strategies in achieving accurate and meaningful image retrieval outcomes.

V. Discussion

The results of our Content-Based Image Retrieval (CBIR) system reveal significant insights into the performance of both classical and deep learning-based feature extraction methods. Classical techniques, such as color sub-blocks and texture-based features like Local Binary Patterns (LBP), performed well in most cases. Color histograms with sub-blocks not only captured the general color but also detected structural characteristics of the images. Texture-based features complemented color features in scenarios where color alone was insufficient to determine the image type, enhancing retrieval accuracy by capturing greater details. However, the results also indicated that Fourier descriptors and ORB features showed a bias towards certain categories (articleType), such as handbags, wallets, socks, and earrings. To mitigate this bias, the weights of these features were reduced, ensuring they did not disproportionately influence the retrieval results.

The fine-tuning of feature weights, with color contributing the most influence, proved effective in balancing the contribution of each feature type. However, shape descriptors such as Fourier descriptors contributed minimally, indicating limited relevance for the dataset used. Deep learning-based ConvNeXt V2 features resulted in worse results in term of accuracy scores. Additionally, also computationally expensive.

A key observation was that combining multiple features outperformed individual features, particularly after fine-tuning feature weights. This indicates the importance of integrating diverse visual characteristics for a more comprehensive representation of image content. Precision@5 and NDCG@5 metrics showed consistent improvement with optimized combinations, validating the effectiveness of the weighting strategy.

Furthermore, combining label with each of them having specific weights showed a better representation of accuracy. Having several different labels (keywords) for each image was a challenge in terms of evaluation but in the end it became an advantage to represent accuracy with numbers.

On the other hand, the poor performance of CNN features should be explored further and larger models should be tested.

Overall, the proposed CBIR system demonstrates a balance between accuracy and efficiency, making it suitable for scalable fashion product recommendations.

VI. Conclusions

This study focused on building and evaluating a Content-Based Image Retrieval (CBIR) system for fashion product images. We compared classical feature extraction methods, such as color histograms, with deep learning-based approaches using a pre-trained ConvNeXt model. The results showed that combining classical features with optimized weights improved retrieval accuracy. The weighted combination of features like color, texture, and shape performed better than using individual features alone.

The deep learning approach, while promising in theory, did not outperform the classical methods on this dataset. This might be due to a mismatch between the features learned by the pre-trained model and the specific needs of the dataset. Classical methods, on the other hand, were efficient and worked well, especially after fine-tuning feature and label weights.

In the future, the system could be improved by training CNN models specifically for fashion data or combining classical and deep learning features. Also, one possible next step is to reduce the need for manual metadata tagging, which can be inconsistent and prone to mistakes. Machine learning models could be used to predict important attributes like color or category directly from product images. This would help reduce human errors, make data preparation faster, and improve the accuracy of search results for e-commerce platforms.

Appendix A
Contribution of Each Person

Throughout the project, we held regular meetings to stay updated on each other's progress. Ceren Arkaç primarily worked on data preprocessing, retrieval evaluation, color features, and transfer learning. Sena Yapsu focused on the literature review, selection and implementation of evaluation metrics, and also implementation of classical feature extraction

methods of shape and ORB. Barış Aygen contributed mainly to classical feature extraction, image retrieval and evaluation process. Overall, we collaboratively decided on the important milestones of the project as a team.

APPENDIX B
ADDITIONAL MATERIAL

A. Appendix Figures

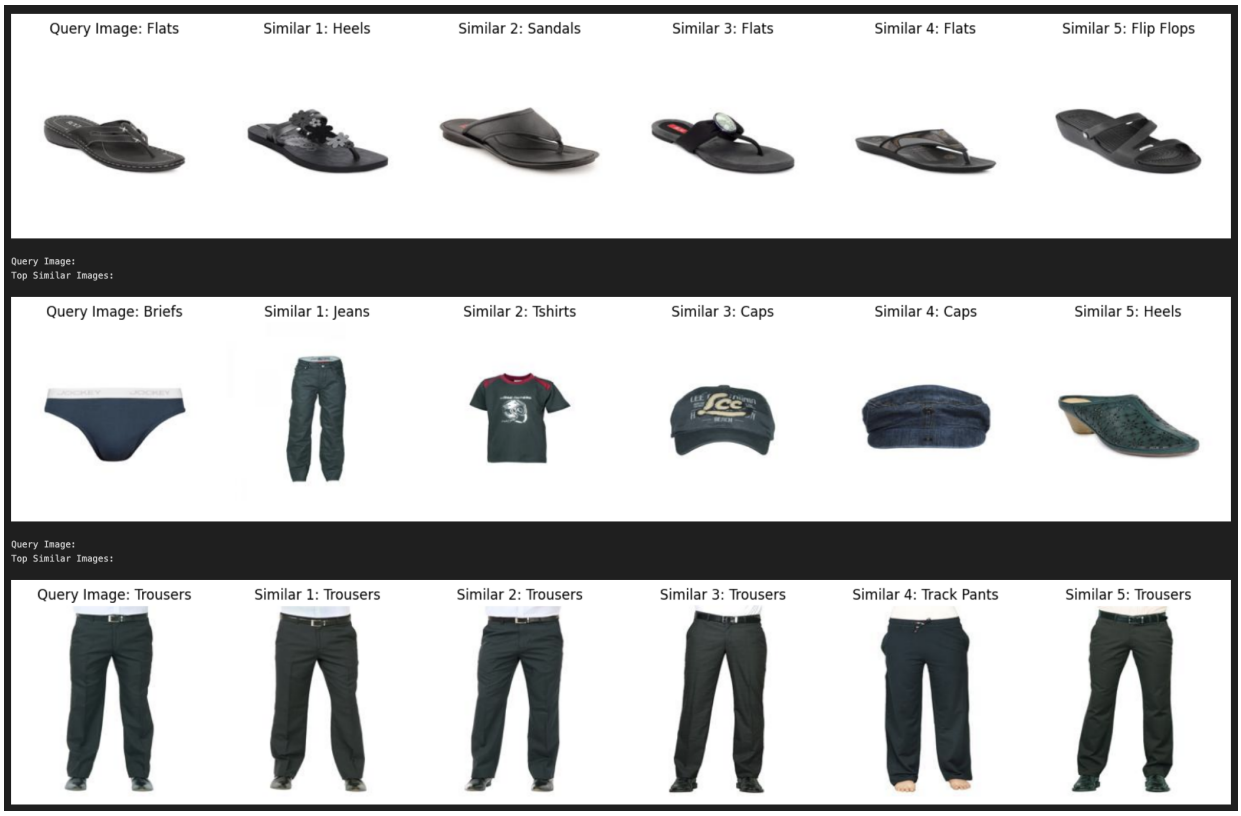

Fig. 1. Example retrieved images for a query using color features.

Fig. 2. Example retrieved images for a query using color features.

B. Appendix Figures

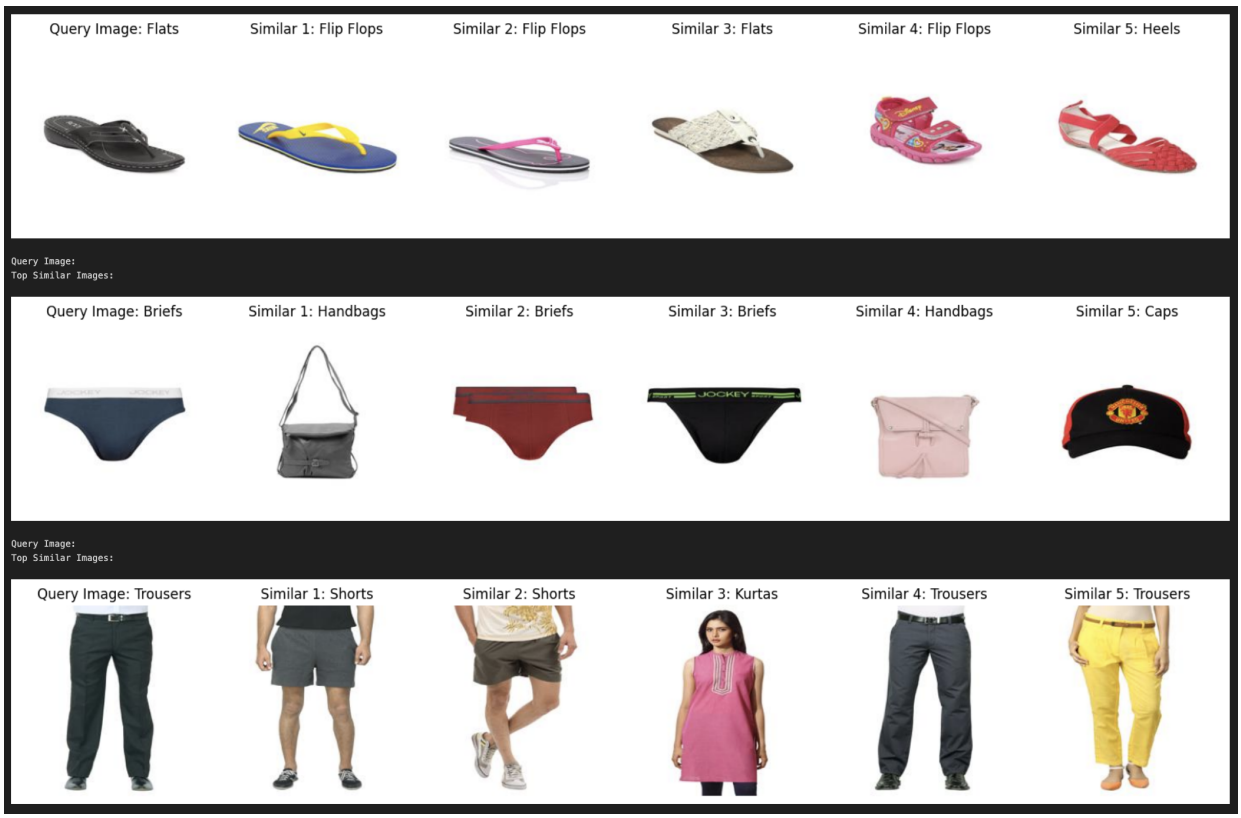

Fig. 3. Example retrieved images for a query using texture features.

Fig. 4. Example retrieved images for a query using texture features.

Fig. 5. Example retrieved images for a query using shape features.

Fig. 6. Example retrieved images for a query using shape features.

Fig. 7. Example retrieved images for a query using orb features.

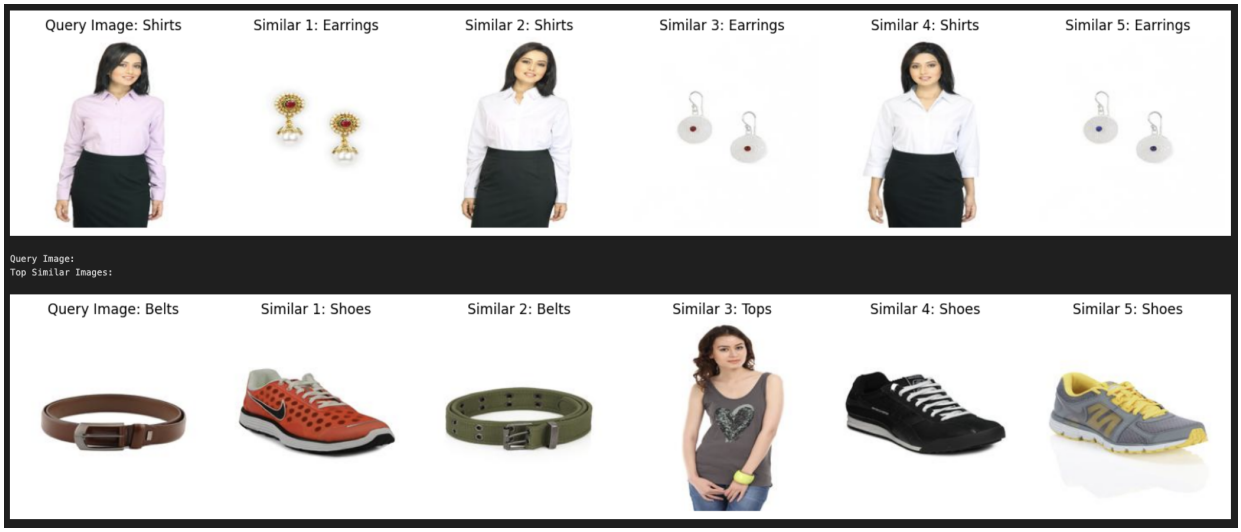

Fig. 8. Example retrieved images for a query using orb features.

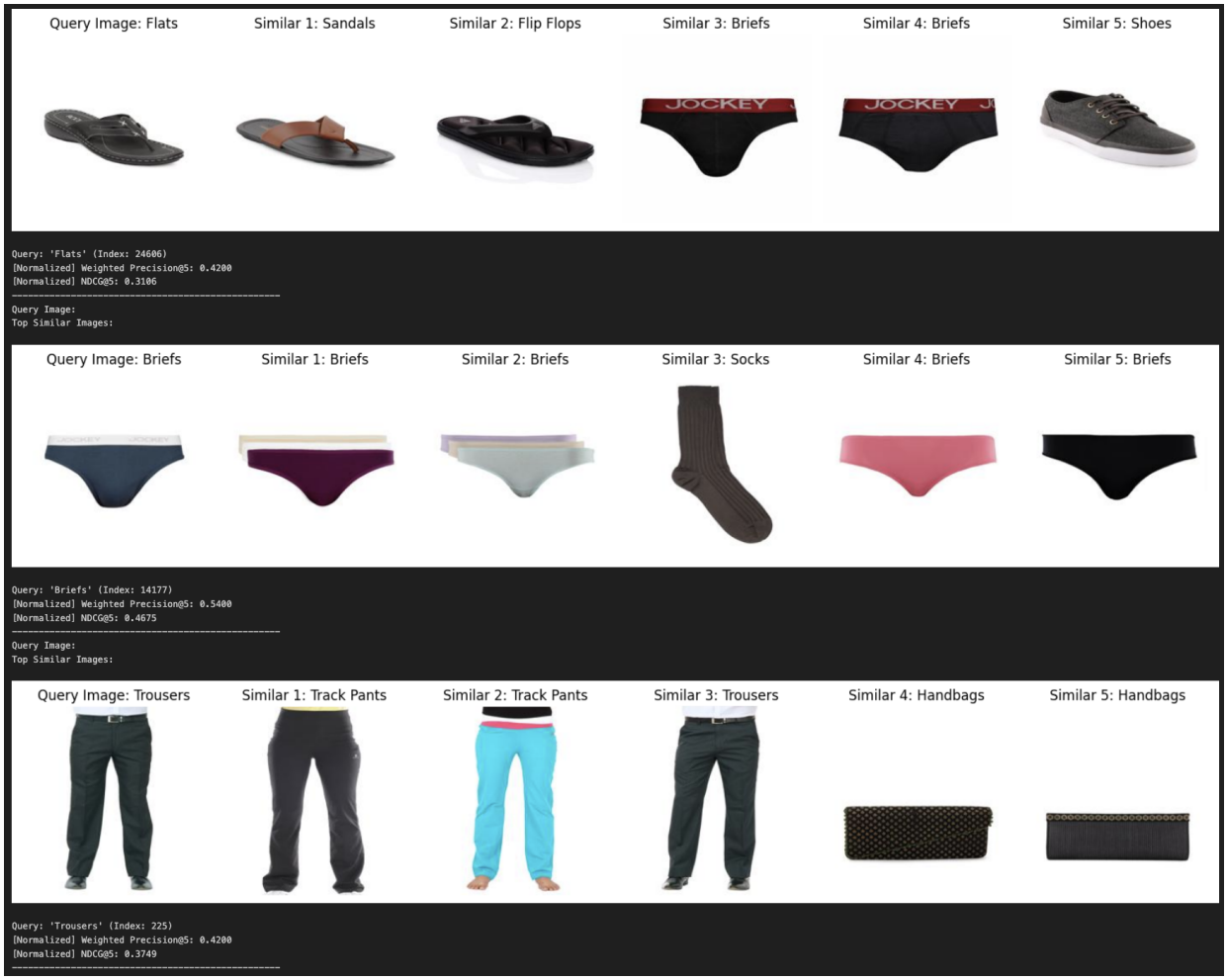

Fig. 9. Example retrieved images for a query using combined features.

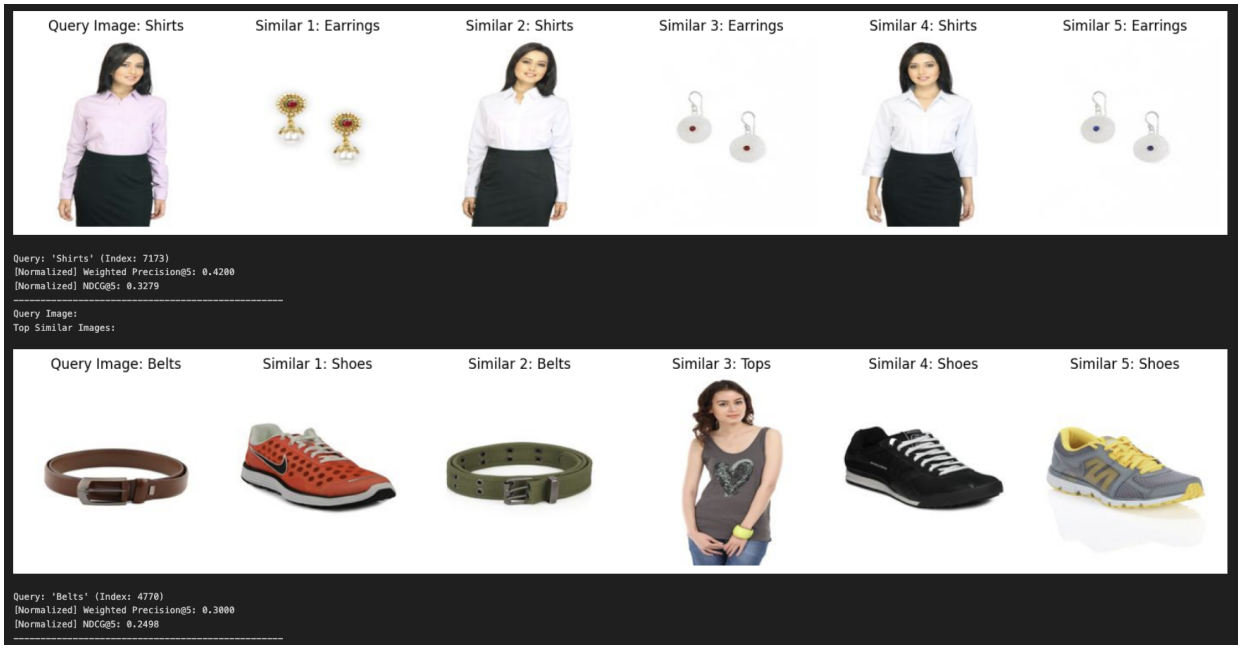

Fig. 10. Example retrieved images for a query using combined features.

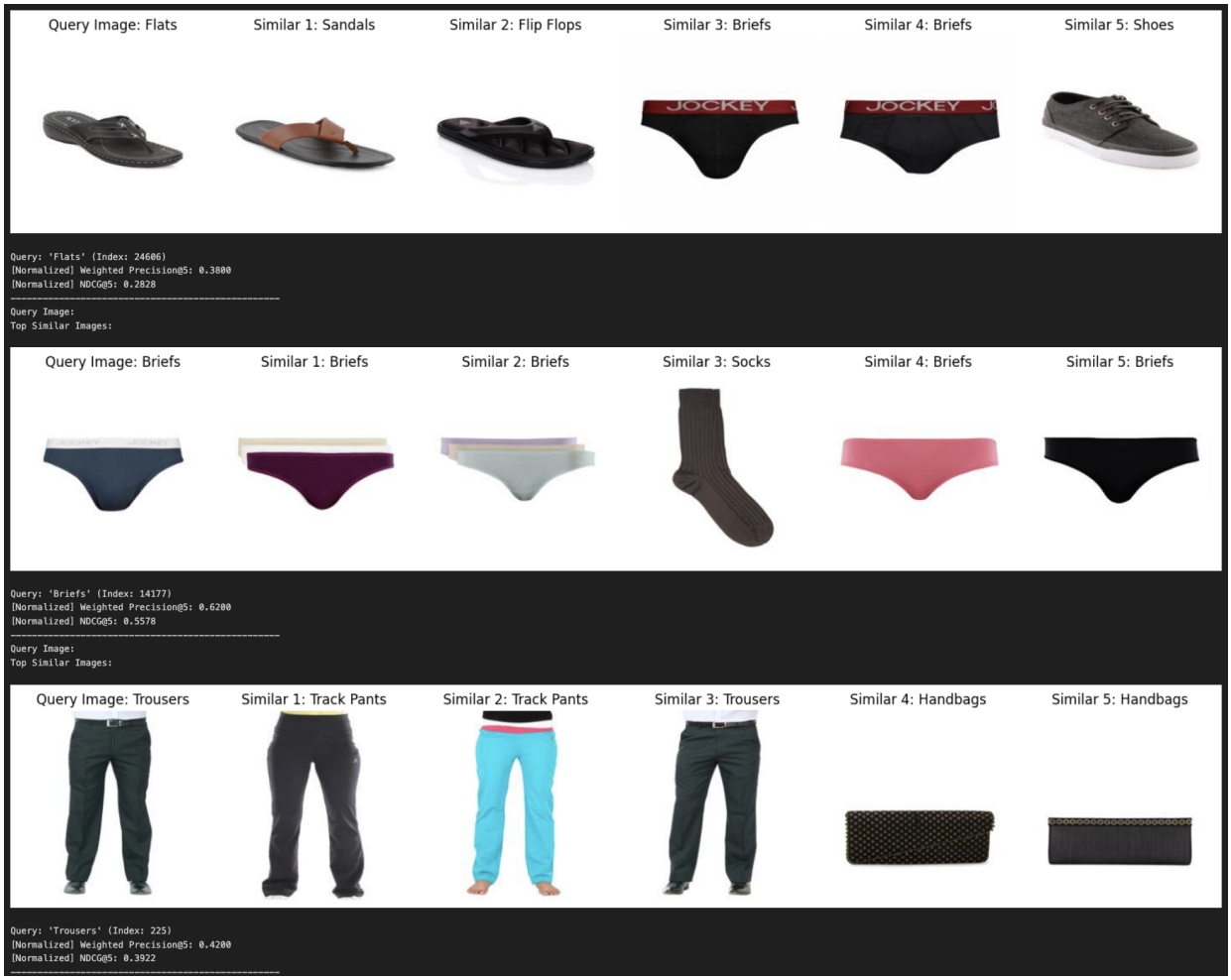

Fig. 11. Example retrieved images for a query using combined features (Label weights fine-tuned)

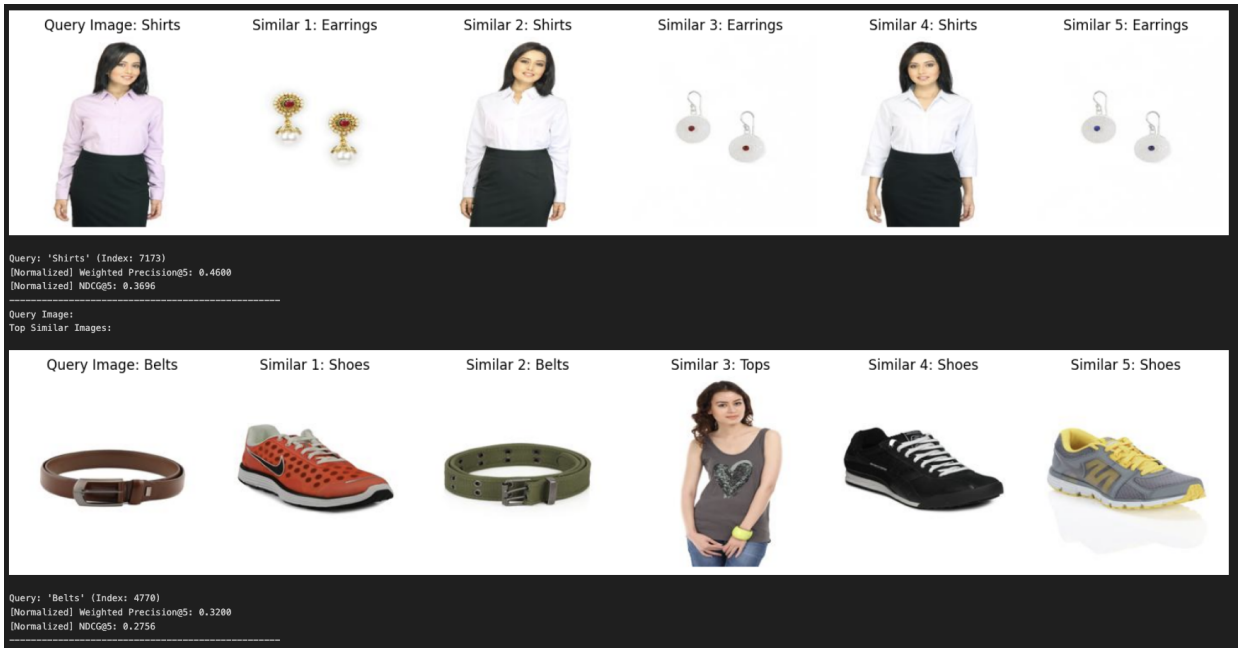

Fig. 12. Example retrieved images for a query using combined features (Label weights fine-tuned)

Fig. 13. Example retrieved images for a query using combined features (Feature weights fine-tuned)

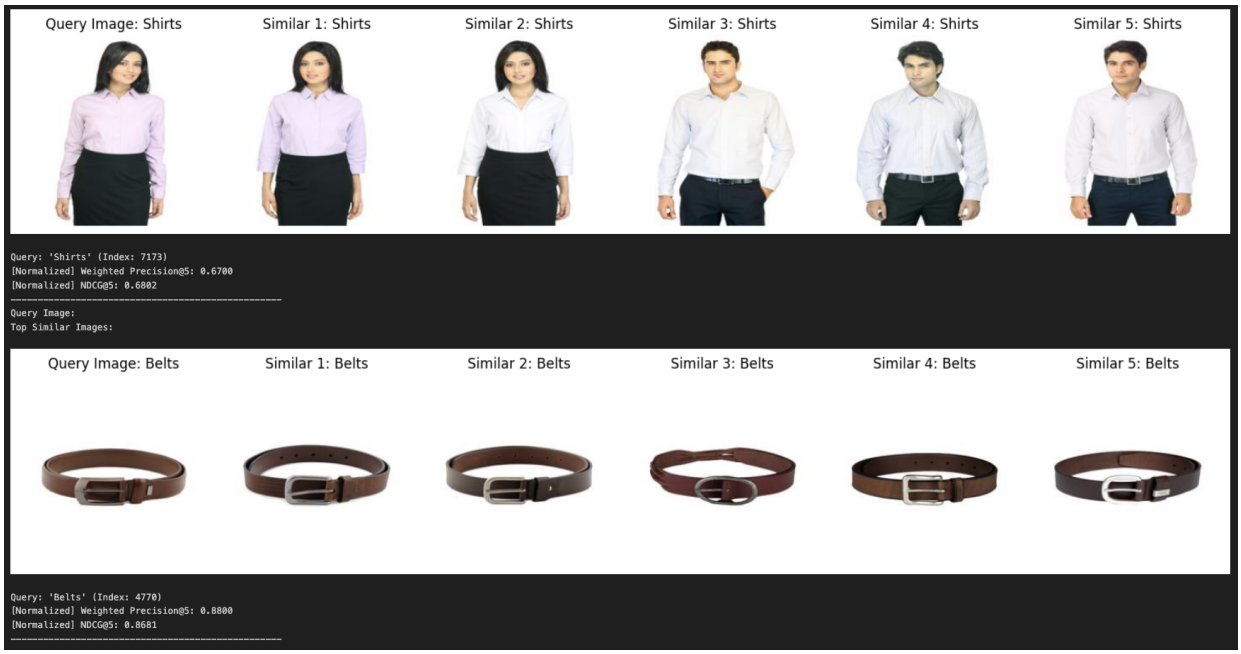

Fig. 14. Example retrieved images for a query using combined features (Feature weights fine-tuned)

Fig. 15. Example retrieved images for a query using features extracted from CNN

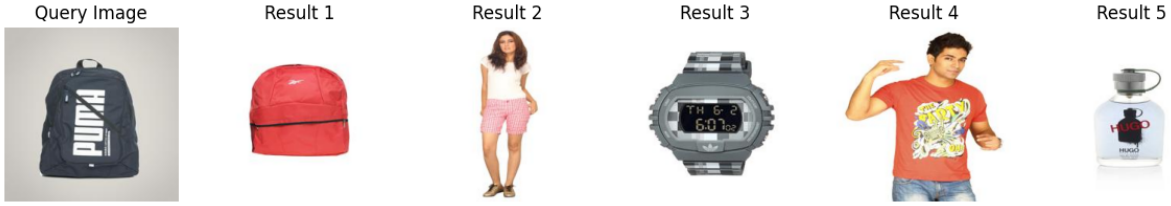

Fig. 16. Example retrieved images for a query using features extracted from CNN

Fig. 17. Example retrieved images for a query using features extracted from CNN

TABLE IV
PRECISION@5 AND NDCG@5 COMPARISON ACROSS METHODS FOR SAMPLE QUERY IMAGES WITH ALTERNATING ROW COLORS

Method	**Query Image 1**	**Query Image 2**	**Query Image 3**	**Query Image 4**	**Query Image 5**
Combined Precision	0.42	0.54	0.42	0.42	0.30
Combined NDCG	0.31	0.46	0.37	0.32	0.24
Label Tuned Precision	0.38	0.62	0.42	0.46	0.32
Label Tuned NDCG	0.28	0.55	0.39	0.36	0.27
Feature Tuned Precision	0.78	0.33	0.79	0.67	0.88
Feature Tuned NDCG	0.73	0.25	0.79	0.68	0.86

REFERENCES

[1] M. O. İncetas and R. U. Arslan, "Spiking neural network-based edge detection model for content-based image retrieval," *Signal, Image and Video Processing*, vol. 19, no. 1, 2024. [Online]. Available: https://doi.org/10.1007/s11760-024-03799-6

[2] G. Zhang, Z. M. Ma, Q. Tong, Y. He, and T. Zhao, "Shape feature extraction using fourier descriptors with brightness in content-based medical image retrieval," in *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2008, pp. 71–74.

[3] A. Ahmed, "Pre-trained cnns models for content based image retrieval," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, pp. 197–202, 2021. [Online]. Available: https://thesai.org/Publications/ViewPaper?Code=IJACSA&Issue=7&SerialNo=23&Volume=12

[4] S. Rani, G. Kasana, and S. Batra, "An efficient content-based image retrieval framework using separable cnns," *Cluster Computing*, vol. 28, no. 1, 2024. [Online]. Available: https://doi.org/10.1007/s10586-024-04731-w

[5] B. Patel, A. Desai, and R. Bhavsar, "A comprehensive review of shape feature extraction techniques," *2023 3rd International Conference on Computing and Communications (ICCC)*, pp. 1–6, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10037663

[6] D. Srivastava, S. S. Singh, B. Rajitha, M. Verma, M. Kaur, and H.-N. Lee, "Content-based image retrieval: A survey on local and global features selection, extraction, representation, and evaluation parameters," *IEEE Access*, vol. 11, pp. 95 410–95 427, 2023.

[7] K. L. Jain, "Texture based feature extraction methods for content based medical image retrieval systems," *PubMed*, 2014. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/25227014/

[8] Y. Patel *et al.*, "Content-based image retrieval: A survey on local and global features," *ResearchGate*, 2023. [Online]. Available: \url{https://www.researchgate.net/publication/373410194_Content-based_Image_Retrieval_A_Survey_on_Local_and_Global_Features_Selection_Extraction_Representation_and_Evaluation_Parameters}

[9] S. Iqbal, S. Ahmad, and H. Ullah, "Content based image retrieval using local binary pattern operator and data mining techniques," *PubMed*, 2015. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/25991105/

[10] M. Singh and D. Goyal, "An integrated approach for image retrieval using local binary pattern," *ResearchGate*, 2015. [Online]. Available: https://www.researchgate.net/publication/276099679_An_integrated_approach_for_image_retrieval_using_local_binary_pattern

[11] Z. Liu, Y. Lin, Z. Xie, Z. Zhang, Y. Cao, and H. Hu, "Convnext v2: Co-designing and scaling convnets with masked autoencoders," *arXiv preprint arXiv:2301.00808*, 2023. [Online]. Available: https://arxiv.org/abs/2301.00808

[12] I. Siddique, S. A. Javed, U. R. Khan, A. Munir, and H. Anwar, "A survey of fashion recommender systems for fashion merchandising," *arXiv preprint arXiv:2005.08170*, 2020, accessed: December 29, 2024. [Online]. Available: https://arxiv.org/pdf/2005.08170