Composed Image Retrieval System

Technical Documentation & Flowchart

A comprehensive guide to the fashion image retrieval system with color filtering capabilities

Table of Contents

1. System Overview

The Composed Image Retrieval (CIR) system is an advanced fashion recommendation platform that combines visual and textual information to find similar fashion items. Users upload a reference image and provide modification text (e.g., "make it greenish-yellow"), and the system returns the most similar items that match both the visual style and the requested modifications.

Key Features:

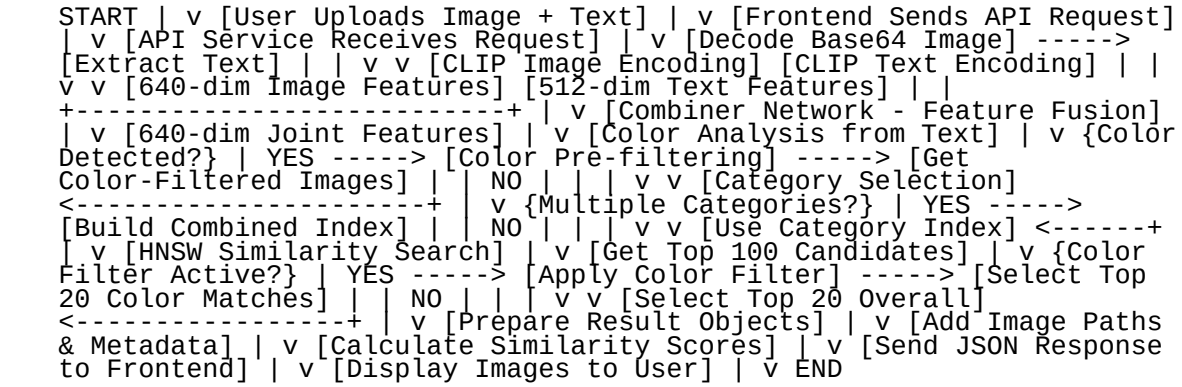
- Multi-modal search combining image and text
- Advanced color filtering with 80+ color combinations
- Category-specific optimization (Dress, Shirt, TopTee)
- Real-time search with 100-500ms response time
- Support for 37,189 fashion images

2. Architecture Components

Component	Description	Specifications
CLIP Model	Vision-language feature extraction	RN50x4 variant, 640-dim image features, 512-dim
Combiner Network	Feature fusion with attention	AttentionFusionCombiner, 640-dim joint features
HNSW Index	Fast similarity search	Hierarchical Navigable Small World, category-spe
Color Clustering	Color-based filtering	K-means clustering + KD-trees, adaptive radius
Flask API	RESTful service	CORS enabled, JSON responses, image serving
Frontend	User interface	HTML/JavaScript, drag-and-drop upload, real-time

3. Retrieval Process Flowchart

The following flowchart illustrates the complete retrieval process from user input to result display:



4. Detailed Process Steps

Step	Process	Output
1	Image Decoding	PIL Image object from base64
2	CLIP Preprocessing	Normalized tensor (224x224)
3	Image Feature Extraction	640-dimensional vector
4	Text Tokenization	CLIP tokenized text
5	Text Feature Extraction	512-dimensional vector
6	Feature Fusion	640-dimensional joint features
7	Color Analysis	Target color + radius
8	Category Selection	Search categories list
9	Index Selection	HNSW index for search
10	Similarity Search	Top 100 candidate indices
11	Color Filtering	Color-filtered image IDs
12	Result Preparation	Top 20 formatted results
13	Response Generation	JSON with metadata

5. Color Processing System

The color processing system analyzes modification text to extract color information and applies intelligent filtering to find images with matching colors.

Color Detection Methods:

1. Exact Color Matching: blue, red, green, etc.
2. Color Combinations: green-yellow, blue-green, etc.
3. Color Modifiers: light-blue, dark-red, etc.
4. Fashion Colors: turquoise, lavender, coral, etc.

Adaptive Radius System:

Query Type	Radius	Description
Combination	100	For 'ish', 'combination', 'mix' words
Light/Dark	80	For 'light', 'dark', 'pale' modifiers
Exact	60	For precise color matches

6. Performance Metrics

Metric	Value	Description
Total Images	37,189	Unique fashion items across all categories
Dress Images	11,643	Dress category items
Shirt Images	13,261	Shirt category items
TopTee Images	12,945	TopTee category items
Search Speed	100-500ms	Average response time per query
Color Support	80+	Supported colors and combinations
Result Count	20	Top results returned per query
Feature Dimensions	640	Joint feature vector size
Index Type	HNSW	Hierarchical Navigable Small World
Color Clusters	K-means	K-means clustering for color filtering

7. API Endpoints

Endpoint	Method	Description	Response
/health	GET	Health check	Status: healthy
/categories	GET	Get available categories	List of categories
/retrieve	POST	Main retrieval endpoint	JSON with results
/images/<id>	GET	Serve image files	Image file

Sample /retrieve Request:

```
{ "image": "...",
"modification_text": "make it greenish-yellow", "search_categories":
["dress", "shirt", "toptee"] }
```

8. Technical Specifications

Component	Technology	Version/Specs
Backend Framework	Flask	Python web framework
Deep Learning	PyTorch	Neural network framework
Vision Model	CLIP	RN50x4 variant
Search Index	HNSW	Hierarchical Navigable Small World
Color Clustering	scikit-learn	K-means + KD-trees
Image Processing	Pillow	PIL fork for image handling
Vector Operations	NumPy	Numerical computing
Frontend	HTML/JavaScript	Vanilla JS with drag-drop
API Format	JSON	RESTful API with CORS
Image Serving	Flask static	Direct file serving
Environment	Python venv	Isolated dependencies

Conclusion:

The Composed Image Retrieval System represents a sophisticated approach to fashion recommendation, combining state-of-the-art vision-language models with intelligent color processing and optimized search algorithms. The system provides fast, accurate, and user-friendly fashion discovery capabilities with support for complex color combinations and multi-category search.