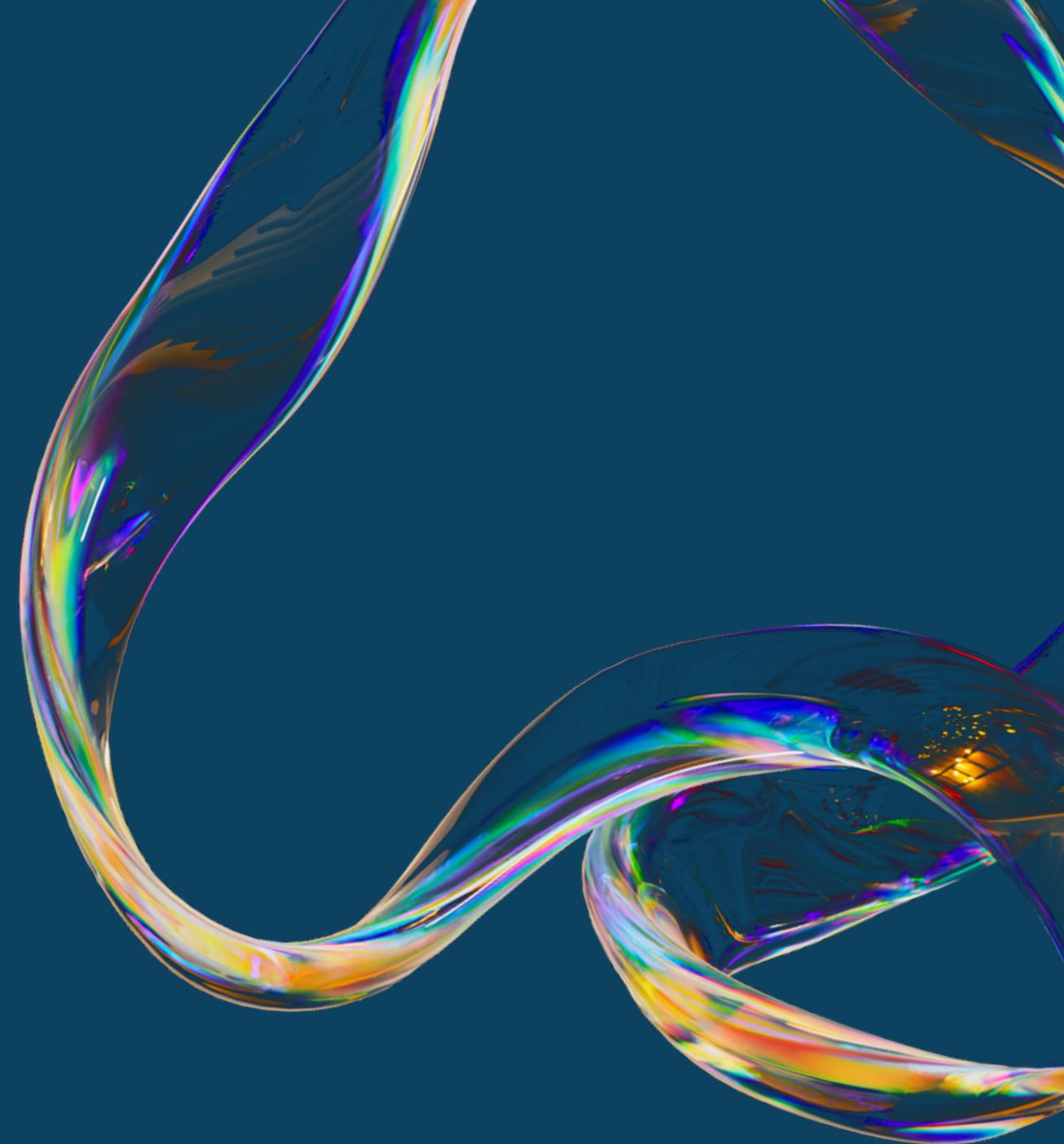




DEEP STYLE EMBEDDING: FASHION COMPATIBILITY AND VISUAL RETRIEVAL SYSTEM

Present by Le Huynh Thuy Vy

AGENDA

- 
- 1 Problem Statement & Solution Overview
 - 2 Architecture & Technical Foundation
 - 3 Feature 1: Visual Search
 - 4 Feature 2: Outfit Compatibility
Recommendation
 - 5 System Deployment & Operation
 - 6 Results, Limitations & Future Development

OVERVIEW

SOLUTION OVERVIEW

Deep Style Embedding: Fashion Compatibility and Visual Retrieval System

Applying Deep Learning to extract image features, which are then converted into embedding vectors. Recommendations are generated based on the similarity between these vectors.

- ResNet50 Embedding for similar product search.
- Siamese Embedding for image-based outfit recommendations.

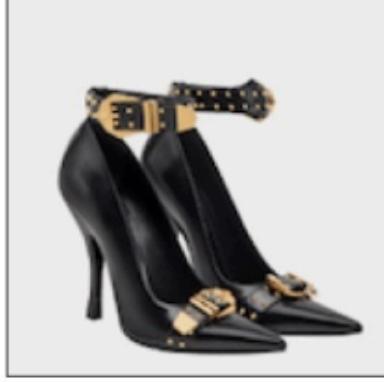


SOLUTION OVERVIEW

Deep Style Embedding: Fashion Compatibility and Visual Retrieval System

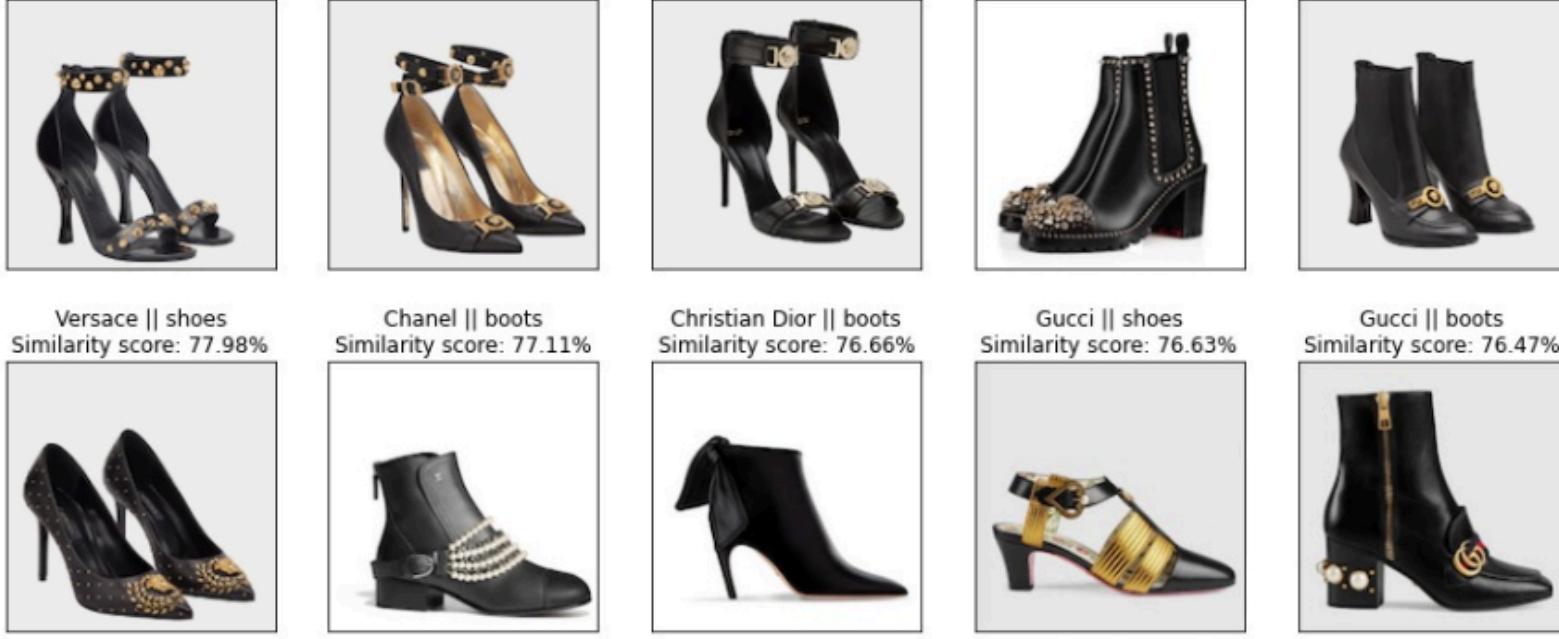
Similar Product Recommendations

Versace || shoes



==== Recommended products: =====

Product	Similarity score
Versace shoes	85.16%
Versace shoes	84.49%
Versace shoes	80.16%
Christian Louboutin boots	79.00%
Versace boots	78.17%
Versace shoes	77.98%
Chanel boots	77.11%
Christian Dior boots	76.66%
Gucci shoes	76.63%
Gucci boots	76.47%



Outfit Recommendations



POLYVORE OUTFIT DATASET

- A fashion dataset collected from Polyvore.com, where users create and share outfits.
- Purpose: Used for research in fashion compatibility and outfit recommendation.
- Scale: ~251,000 product images and ~68,000 outfits (each outfit contains 2–8 items)

Directory / File Name	Description	Use Case
disjoint/	Contains train/validation/test data with non-overlapping items (disjoint split).	For training & testing the model's ability to make general outfit recommendations.
nondisjoint/	Train/test data where items can overlap between sets.	Used for faster training and for testing inset recommendations.
images/	Repository of fashion images (251K product images).	For product visualization and image feature extraction (using ResNet/VGG).
polyvore_outfit_titles.json	A list of outfits and their associated item_ids.	To link individual items into complete outfits.
categories.csv	Categories and classifications for different product types.	For standardizing labels (category mapping) during training.

Data Explorer	
Version 1 (17.08 GB)	
▼ polyvore_outfits	

↳ disjoint

↳ images

↳ maryland_polyvore_hair

↳ nondisjoint

↳ README

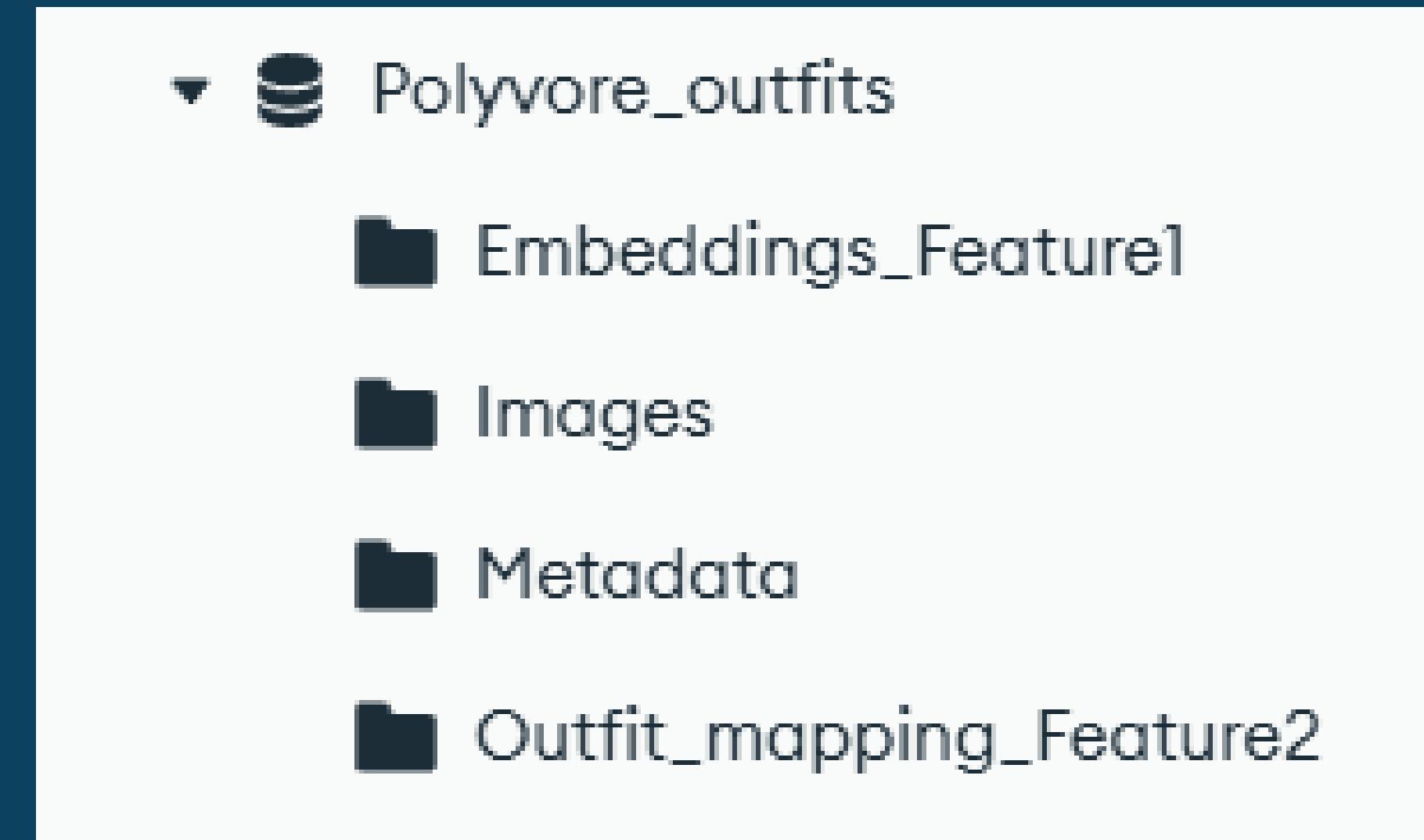
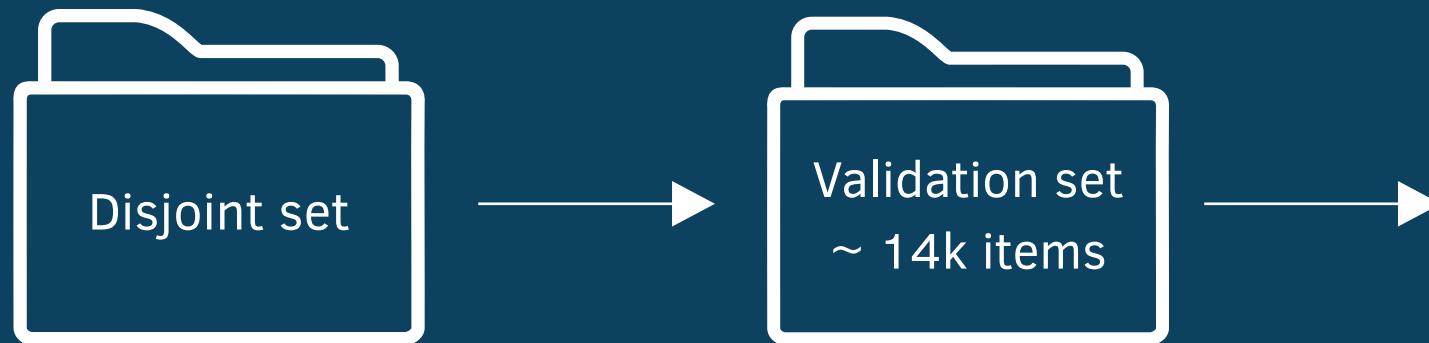
⌚ polyvore_item_inetadat

⌚ polyvore_outfit_titles.js

☰ categories.csv

MONGGODB ATLAS DATABASE

- Utilized MongoDB Atlas as the primary database.
- Stored data for 14,000 items from the disjoint validation set.
- The architecture consists of 4 collections with data structured in JSON format.
- Implemented Vector Search on the "Feature 1" vectors to enable efficient similarity queries.



OVERALL ARCHITECTURE

The system consists of four main modules:

IMAGE ENCODING MODULE (RESNET 50)



Extracts visual features from images.

COMPATIBILITY MODULE (SIAMESE NETWORK)



Learns the compatibility relationships between products.

DATABASE (MONGODB ATLAS)



Stores data and enables fast similarity search on vectors

API GATEWAY (FAST API & MICROSERVICES)



Manages communication between the front-end and back-end and facilitates system scaling.

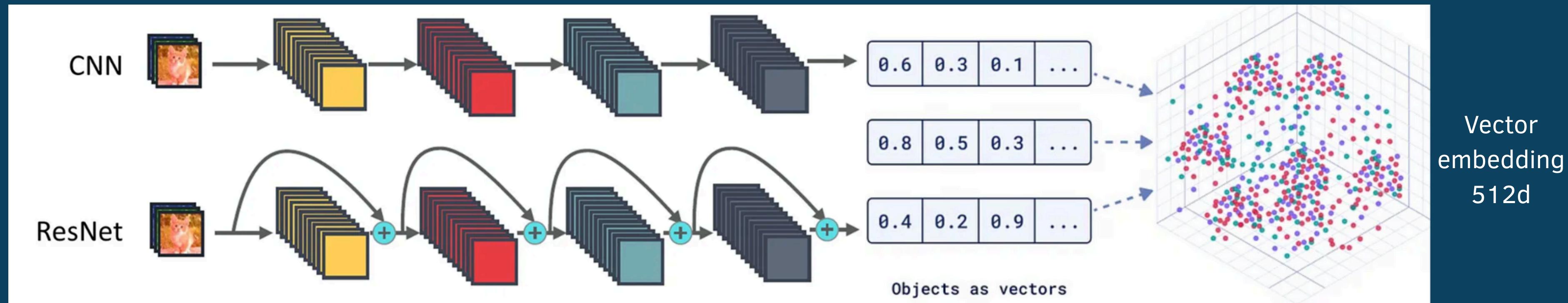
THE ROLE OF CNN (RESNET-50) IN GENERATING ITEM EMBEDDINGS

CNN

A specialized neural network for image processing. It learns visual features by applying filters across an image, detecting everything from simple edges to complex patterns.

ResNet-50 (Pre-trained Model)

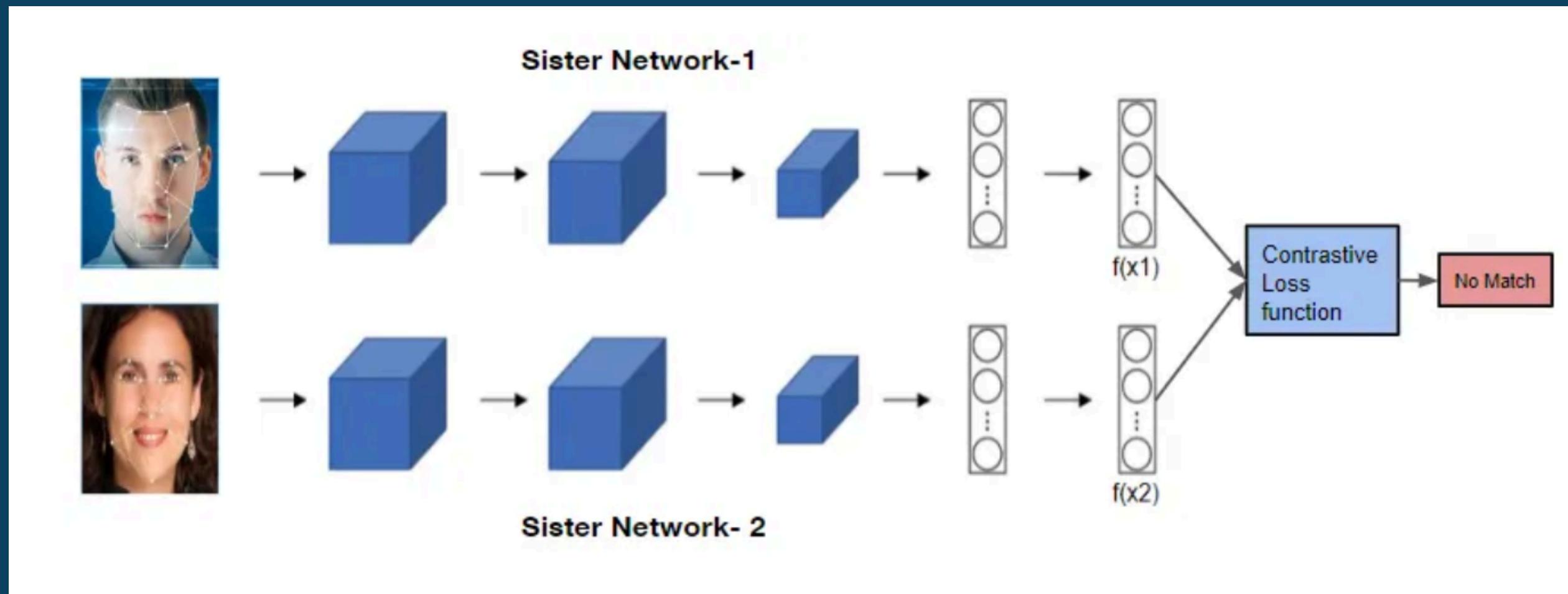
A deep (50-layer) CNN pre-trained on the large-scale ImageNet dataset. We use it as a powerful backbone and fine-tune it on our fashion data.



THE ROLE OF SIAMESE IN GENERATING ITEM EMBEDDINGS

A Siamese Network is a "twin" neural network architecture with two identical branches. Instead of classifying an input, it learns to measure the similarity between two inputs.

Logic: It learns to map similar (compatible) pairs to nearby points in an embedding space, and dissimilar pairs far apart.

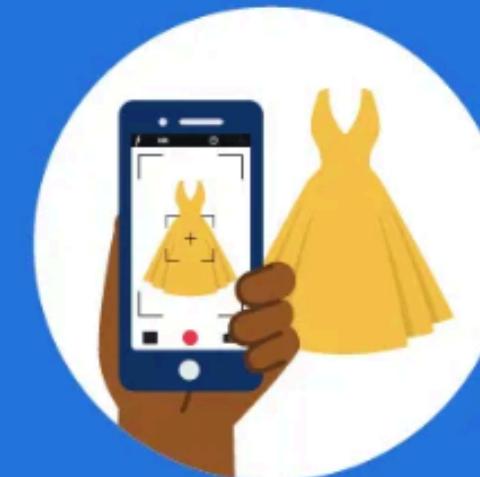


FEATURE 1: IMAGE SEARCH

WHAT IS THE IMAGE SEARCH FUNCTION?

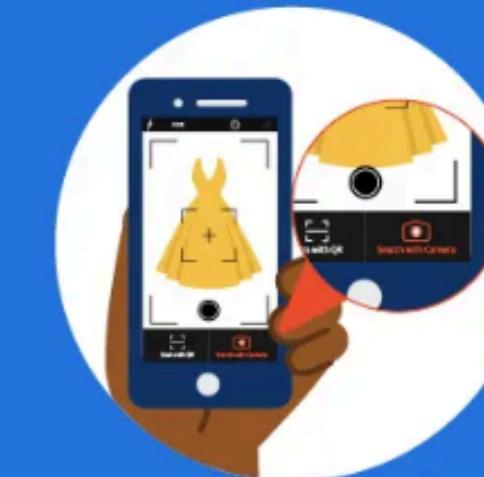
The Image Search function allows users to search and retrieve information from the photo they've taken, and directs them to a similar product on the app.

STEP 1



Take a photo

STEP 2



Use the Image Search
to identify the item

STEP 3



Make a purchase!

FEATURE 1: IMAGE SEARCH

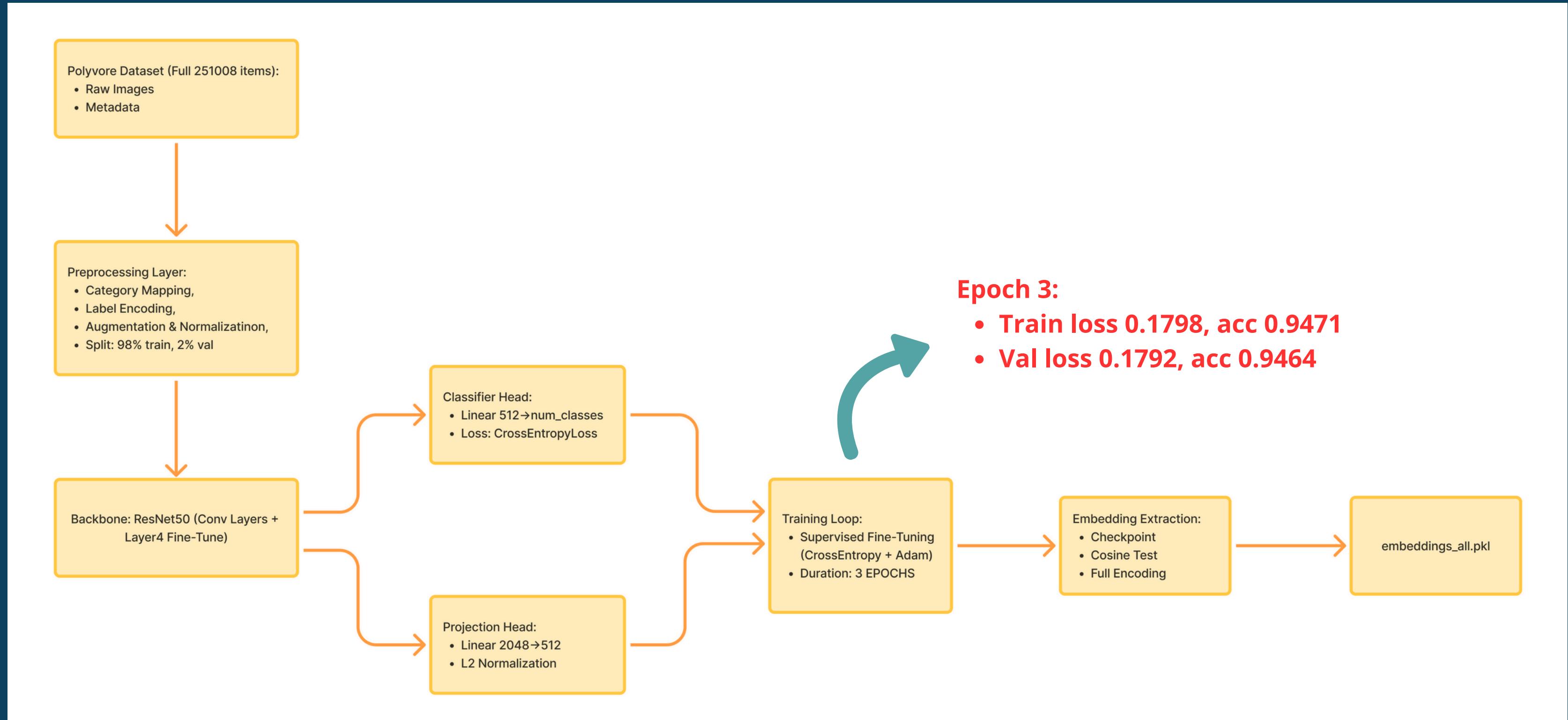


Figure: 512D Embedding Pipeline Architecture using Supervised Fine-Tuning

INTRA-CATEGORY STYLE DIFFERENCE (Category: hats)

QUERY: hats



Nearest 1 (Sim: 1.000)
Category: hats



Nearest 2 (Sim: 1.000)
Category: hats



Nearest 3 (Sim: 0.999)
Category: hats



Farthest 1 (Sim: -0.240)
Category: hats



Farthest 2 (Sim: -0.136)
Category: hats



Farthest 3 (Sim: -0.044)
Category: hats



3 FARDEST ITEMS IN THE ENTIRE EMBEDDING SPACE (Query: hats, ID: 201811447)

Farthest 1 (Sim: -0.452)
Category: jewellery



Farthest 2 (Sim: -0.439)
Category: shoes



Farthest 3 (Sim: -0.431)
Category: accessories



Figure: Comprehensive Vector Retrieval Test for Image Search (Item ID: 201811447)

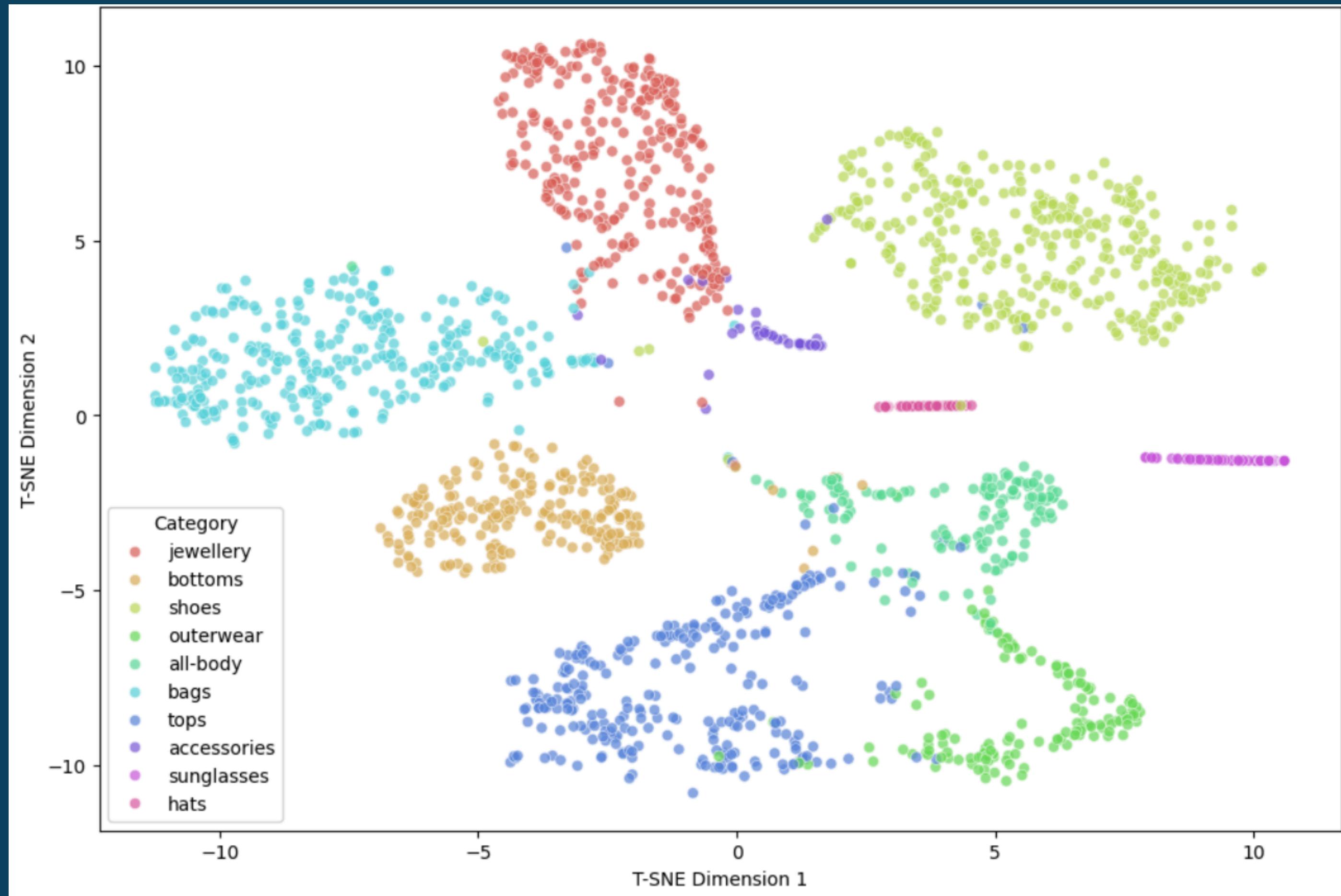


Figure: T-SNE Visualization of the 512D Embedding Space by Top 10 Categories (SAMPLE_SIZE = 2000)

FEATURE 2: OUTFIT COMPATIBILITY RECOMMENDATION



(a)



(b)

FEATURE 2: OUTFIT COMPATIBILITY RECOMMENDATION

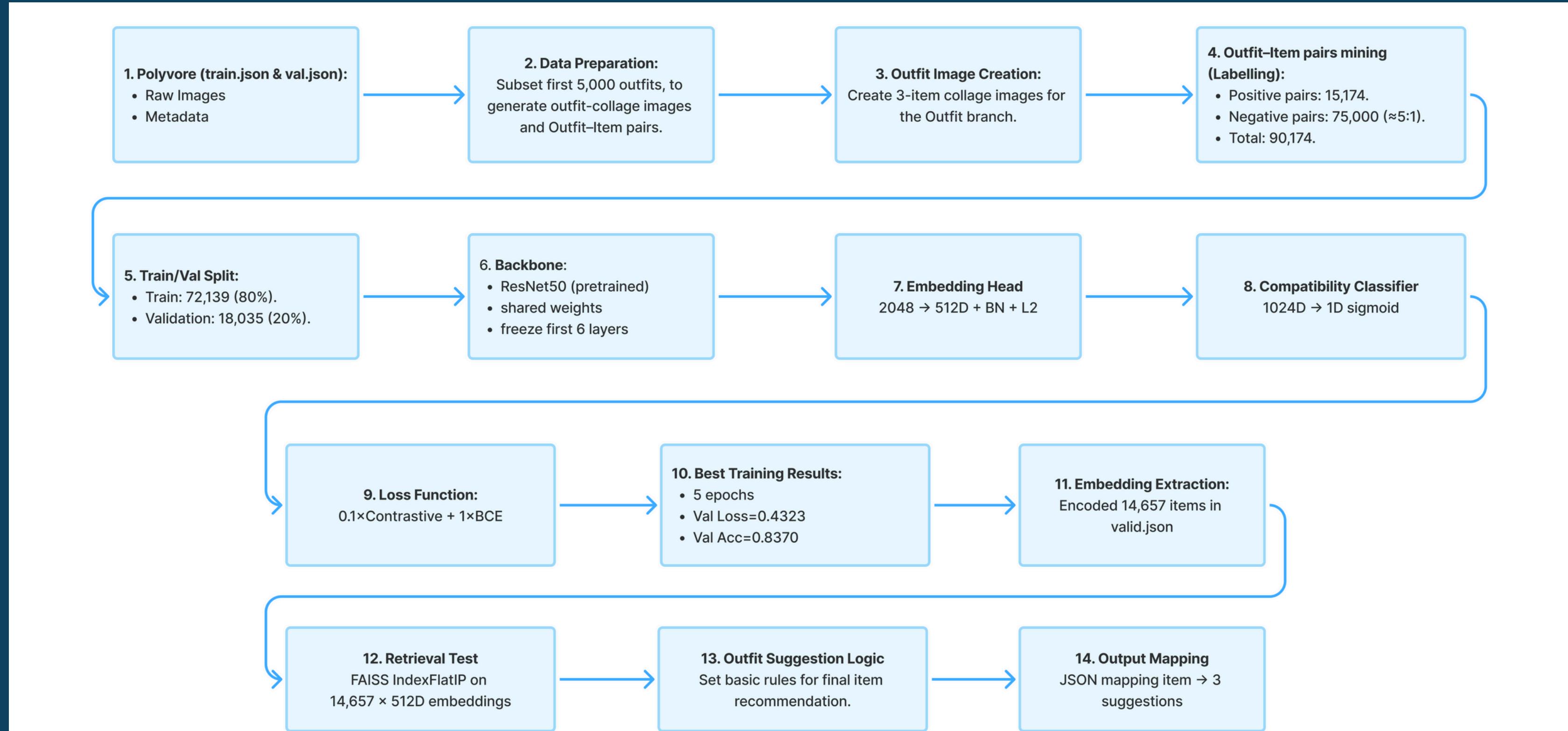


Figure: Siamese Network Pipeline for Fashion Compatibility Modeling

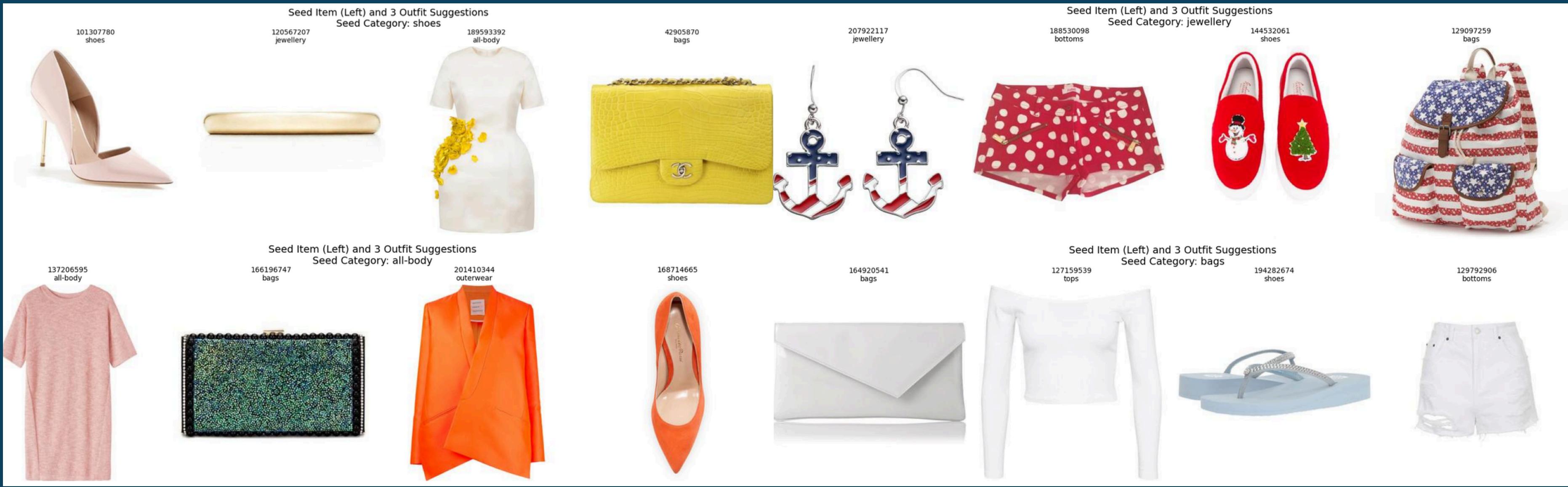


Figure: Visual Outfit Suggestion Grid based on Item Embedding Similarity

Loaded outfit mapping for 14657 items.

Seed Item ID: 117137688

Suggested Items: ['208401000', '185455419', '208594361']

Figure: Example of Full Retrieval Mapping Output (Seed Item ID: 117137688)

SYSTEM DEPLOYMENT & OPERATIONS

SYSTEM OVERVIEW - WEBSITE

BACKEND

- Web Framework: FastAPI
- Main Feature
 - Visual Similarity Search
 - Outfit Compatibility
- Database : MongoDB Atlas storage
 - Metadata
 - Image data
 - Vector embedding AI

GET	/healthz	Healthz
POST	/similar/{item_id}	Find similar items using MongoDB
POST	/search_image	Find similar items by image using MongoDB
POST	/compatible/{item_id}	Get compatible items using pre-computed mapping
GET	/image/{item_id}	Get product image by item_id

- Data Retrieval APIs (Return JSON)
 - POST /similar/{item_id}
 - POST /search_image
 - POST /compatible/{item_id}
- Utility APIs

Feature Visual Search

INPUT

SYSTEM

MONGODB

User uploads product images

Preprocess input images

Use the ResNet50 AI model to extract image features

Return the corresponding 512-dimensional embedding vector

API /search_image

Include a JSON file containing descriptions of the 10 suggested images

Vector Search

Search for the most similar vector – Cosine Similarity

Return the 10 closest vectors

Retrieve the descriptions of the 10 corresponding images

INPUT

SYSTEM

Receive the item_id selected by the user

Query MongoDB to retrieve the accompanying items

```
_id: ObjectId('6914a98066fd436a0827b502')  
item_id : "157649211"  
- rec_feature2 : Array (3)  
  0: "173534408"  
  1: "92665975"  
  2: "189275863"
```

Retrieve the description of the corresponding image

Compatibility Oufit

API /compatible

Provide a JSON file of the final items suggested to pair with the user-selected item

SYSTEM OVERVIEW

Front-end

What is React ?

- ◆ React is a **JavaScript library** for building component-based UIs that are **easy to reuse and manage**.
- ◆ React **optimizes performance** thanks to Virtual DOM, only updating the changed interface.
- ◆ React supports building modern web applications with rich routing, state management, and ecosystems.

React divides the source code into two main parts:

- ◆ **.jsx** → Contains the interface and logic of each component (UI + JavaScript).
- ◆ **.css** → Contains styles to format the interface (colors, layout, fonts, spacing...).

DEMO

[Demo video URL](#)

CONCLUSION

LIMITATION

- User interaction data is not captured to improve recommendations - the system currently does not record user behavior (clicks, product selections).
- Lack of user features: no login, history saving, or account-specific personalized recommendations.
- Data size limitations: The Polyvore dataset is limited in terms of brand diversity, materials, and camera angles, affecting generalizability.



THANK YOU

for your time and attention

