ShopLens - AI Shopping Assistant

Team: **Insight Engineers**
Sayan Das - B2430035 **Raihan Uddin** - B2430070
Supervisor: **Br. Bhaswarachaitanya** (Tamal Maharaj)

April 30, 2025

Outline

Introduction

Background

- Fashion trends driven by **celebrities, influencers, and social media**.
- **Manual searching** for similar items is slow and frustrating.
- **Gap** between what users admire and what they can find online.

Motivation

- The industry is seeking **innovative ways to enhance user experience** and **boost sales conversions**.
- Enabling users to find products inspired by admired images offers a **transformative opportunity**.
- **Imagine:** Uploading a photo of a celebrity, automatically detecting fashion items, and retrieving similar products.
- This approach **eliminates tedious searches** and makes fashion shopping more **accessible, personalized, and enjoyable**.
- With mobile devices and advances in machine learning, the time is **ripe for an AI-based shopping assistant**.

Objectives and Scope

Objectives

- Develop an application to **assist users** in finding fashion items **easily and efficiently**.
- Implement an **AI-powered text-based search system** allowing **natural language queries**.
- Design and integrate an **image-based product search** for retrieving **visually similar fashion products**.
- Leverage **object detection** and **visual similarity algorithms** specialized for fashion items.

Objectives

- Develop an application to **assist users** in finding fashion items **easily and efficiently**.
- Implement an **AI-powered text-based search system** allowing **natural language queries**.
- Design and integrate an **image-based product search** for retrieving **visually similar fashion products**.
- Leverage **object detection** and **visual similarity algorithms** specialized for fashion items.

Objectives

- Develop an application to **assist users** in finding fashion items **easily and efficiently**.
- Implement an **AI-powered text-based search system** allowing **natural language queries**.
- Design and integrate an **image-based product search** for retrieving **visually similar fashion products**.
- Leverage **object detection** and **visual similarity algorithms** specialized for fashion items.

Objectives

- Develop an application to **assist users** in finding fashion items **easily and efficiently**.
- Implement an **AI-powered text-based search system** allowing **natural language queries**.
- Design and integrate an **image-based product search** for retrieving **visually similar fashion products**.
- Leverage **object detection** and **visual similarity algorithms** specialized for fashion items.

Scope

What This Project Does Cover

- **Text and image-based search** for fashion items.
- A unified chatbot interface for **natural language queries** and **image uploads**.
- Implementation of **object detection** to find the different types of clothings of a person is wearing and ability to select one of them for searching.

Scope

What This Project Does Cover

- **Text and image-based search** for fashion items.
- A unified chatbot interface for **natural language queries** and **image uploads**.
- Implementation of **object detection** to find the different types of clothings of a person is wearing and ability to select one of them for searching.

Scope

What This Project Does Cover

- **Text and image-based search** for fashion items.
- A unified chatbot interface for **natural language queries** and **image uploads**.
- Implementation of **object detection** to find the different types of clothings of a person is wearing and ability to select one of them for searching.

Scope

What This Project Does Not Cover

- Our system currently supports independent image-based and text-based retrieval. However, it cannot perform **multimodal queries** that combine both inputs.
- Our system does not yet incorporate user-specific preferences, browsing history, or style profiles,
- The current **Streamlit-based** prototype, while functional, may not scale optimally for very large product inventories or concurrent user loads without transitioning to a more robust backend infrastructure.

Scope

What This Project Does Not Cover

- Our system currently supports independent image-based and text-based retrieval. However, it cannot perform **multimodal queries** that combine both inputs.
- Our system does not yet incorporate user-specific preferences, browsing history, or style profiles,
- The current **Streamlit-based** prototype, while functional, may not scale optimally for very large product inventories or concurrent user loads without transitioning to a more robust backend infrastructure.

Scope

What This Project Does Not Cover

- Our system currently supports independent image-based and text-based retrieval. However, it cannot perform **multimodal queries** that combine both inputs.
- Our system does not yet incorporate user-specific preferences, browsing history, or style profiles,
- The current **Streamlit-based** prototype, while functional, may not scale optimally for very large product inventories or concurrent user loads without transitioning to a more robust backend infrastructure.

Dataset Description

Source - Fashionpedia Dataset

Hugging Face

Link - https://huggingface.co/datasets/detection-datasets/fashionpedia

This dataset was constructed by fashion experts, containing **46,781 images** with **342,182 bounding boxes**. It is used for fine-tuning the YOLOS model for object detection tasks within the fashion domain.

Dataset Statistics

General Information

Total Images	46,781
Total BBoxes	342,182

Data Split

Training Set	45,623 images
Validation Set	1,158 images

Each sample contains:

Field	Description
image_id	Unique image identifier
image	RGB image
width, height	Image dimensions
objects	Detected objects metadata

Object Metadata:

- **bbox_id**, **category**, **bbox** (Pascal VOC format), **area**

Dataset Statistics

46 Fashion Categories

- shirt, blouse
- top, t-shirt, sweatshirt
- sweater
- cardigan
- jacket
- vest
- pants
- shorts
- skirt
- coat
- dress
- jumpsuit
- cape
- glasses
- hat
- headband, hair accessory
- tie
- glove
- watch
- belt
- leg warmer
- tights, stockings
- sock
- shoe
- bag, wallet
- scarf
- umbrella
- hood
- collar
- lapel
- epaulette
- sleeve
- pocket
- neckline
- buckle
- zipper
- applique
- bead
- bow
- flower
- fringe
- ribbon
- rivet
- ruffle
- sequin
- tassel

Source - Fashion Product Images Dataset

Link - https://www.kaggle.com/datasets/paramaggarwal/fashion-product-images-dataset

It contains a total of **1,05,542 fashion products**, almost each with a image and associated metadata.

Dataset Statistics

Field Name	Description
article_id	Unique product identifier
prod_name	Product name
product_type_name	Broad category of the product
colour_group_name	Primary color grouping
department_name	Department like Menswear, Ladieswear
section_name	Section within department
garment_group_name	Garment group classification
detail_desc	Detailed product description

Data Preprocessing

Fashionopedia Dataset

Removing Invalid Bounding Boxes

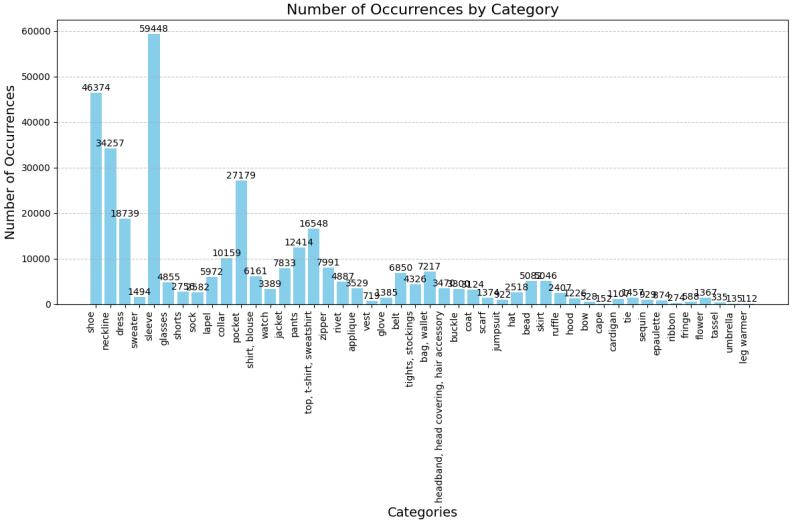
Problem: Some bounding boxes have zero width or height.
Solution:

- Detect and remove invalid bounding boxes.
- Retain only boxes with positive area, along with their IDs, categories, and area values.

Significance: Invalid bounding boxes introduce noise and hurt model training. Cleaning at this stage ensures better stability and learning quality.

Result: Train and validation datasets maintain **45,623** and **1,158** images, respectively, but with only valid boxes.

Visualizing Class Occurrences



Data Augmentation

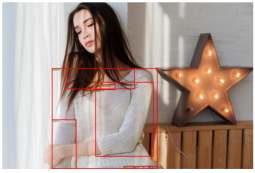
We used Albumentations Library to do some data augmentation on the dataset.

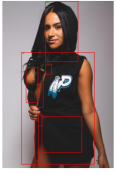
Significance: Data augmentation boosts diversity, reduces overfitting, and improves generalization across different object appearances and conditions.

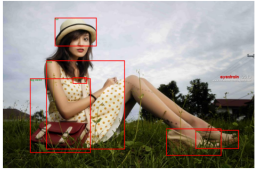
Training Dataset:

- **Resize and Pad:** 500×500 px size.
- **Horizontal Flip:** $p = 0.5$ for left-right invariance.
- **Brightness/Contrast Adjustments**.
- **Small Rotations and Scaling**.
- **Gaussian Blur and Noise**.

Before Augmentation







After Augmentation

Feature Extraction

PIL images need to be converted into numerical tensors for training.

Tool Used: YOLOS Feature Extractor (pretrained; applies normalization, standardization)

Significance: Feature extractors prepare raw images into consistent formats, allowing models to focus on learning patterns, not on raw inconsistencies.

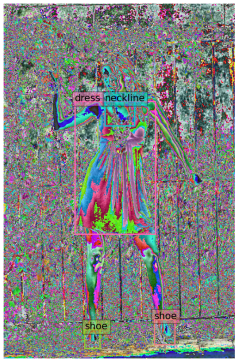


Image after YOLOS feature extraction

Data Preprocessing

Fashion Product Images Dataset

Fashion Product Images Dataset Cleaning

Preprocessing Steps:

1. Remove duplicate entries.
2. Exclude "Unknown" product groups.
3. Filter out descriptions longer than 40 words. Since FashionCLIP has a token limit of 77, this ensures safety margins.
4. Remove rare product types (less than 10 samples).
5. Assign random prices to each item.

Result: Dataset reduced from **105,542** to **37,704** entries.

Fashion Product Images Dataset Cleaning

Preprocessing Steps:

1. Remove duplicate entries.
2. Exclude "Unknown" product groups.
3. Filter out descriptions longer than 40 words. Since FashionCLIP has a token limit of 77, this ensures safety margins.
4. Remove rare product types (less than 10 samples).
5. Assign random prices to each item.

Result: Dataset reduced from **105,542** to **37,704** entries.

Fashion Product Images Dataset Cleaning

Preprocessing Steps:

1. Remove duplicate entries.
2. Exclude "Unknown" product groups.
3. Filter out descriptions longer than 40 words. Since FashionCLIP has a token limit of 77, this ensures safety margins.
4. Remove rare product types (less than 10 samples).
5. Assign random prices to each item.

Result: Dataset reduced from **105,542** to **37,704** entries.

Fashion Product Images Dataset Cleaning

Preprocessing Steps:

1. Remove duplicate entries.
2. Exclude "Unknown" product groups.
3. Filter out descriptions longer than 40 words. Since FashionCLIP has a token limit of 77, this ensures safety margins.
4. Remove rare product types (less than 10 samples).
5. Assign random prices to each item.

Result: Dataset reduced from **105,542** to **37,704** entries.

Fashion Product Images Dataset Cleaning

Preprocessing Steps:

1. Remove duplicate entries.
2. Exclude "Unknown" product groups.
3. Filter out descriptions longer than 40 words. Since FashionCLIP has a token limit of 77, this ensures safety margins.
4. Remove rare product types (less than 10 samples).
5. Assign random prices to each item.

Result: Dataset reduced from **105,542** to **37,704** entries.

Fashion Product Images Dataset Cleaning

Preprocessing Steps:

1. Remove duplicate entries.
2. Exclude "Unknown" product groups.
3. Filter out descriptions longer than 40 words. Since FashionCLIP has a token limit of 77, this ensures safety margins.
4. Remove rare product types (less than 10 samples).
5. Assign random prices to each item.

Result: Dataset reduced from **105,542** to **37,704** entries.

Methodology and Analysis

Tools and Libraries Used

- Python 3.11.11
- Jupyter Notebook
- VS Code
- Kaggle for Fine-Tuning
- datasets
- pandas
- numpy
- faiss
- lightning
- torch
- torchvision
- tensorflow
- opencv-python
- opencv-python-headless
- albumentations
- PIL (Pillow)
- fashion-clip
- streamlit
- sqlalchemy
- psycopg2-binary
- alembic
- pydantic
- python-dotenv
- google-generativeai
- ipykernel
- scikit-learn
- tqdm
- requests

YOLOS Model

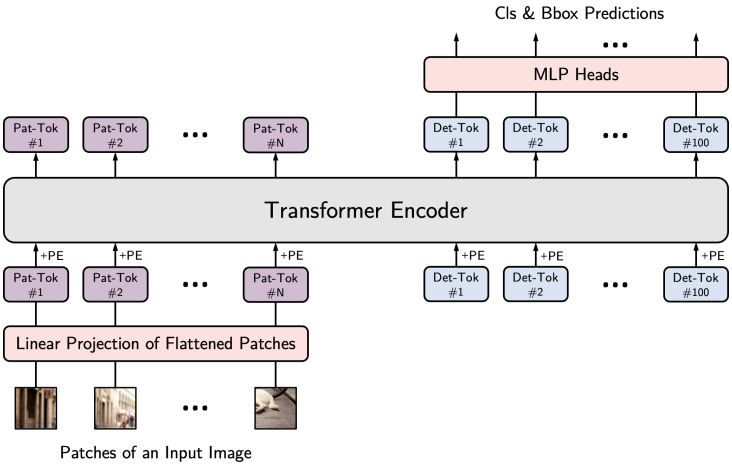
YOLOS (You Only Look One-level Series) is a Vision Transformer (ViT) trained using the DETR loss. Despite its simple design, a base-sized YOLOS model achieves an impressive **42 AP** (Average Precision) on the COCO 2017 validation dataset.

YOLOS-Small Details:

- Pre-trained for 200 epochs on ImageNet-1k.
- Fine-tuned for 150 epochs on COCO dataset.
- Achieves an AP of 36.1 on COCO 2017 validation.

We fine-tuned the YOLOS-Small model on the Fashionpedia dataset for object detection tasks. The model was trained using PyTorch Lightning's Trainer class for 1 epoch.

YOLOS Model Architecture



FashionCLIP Overview

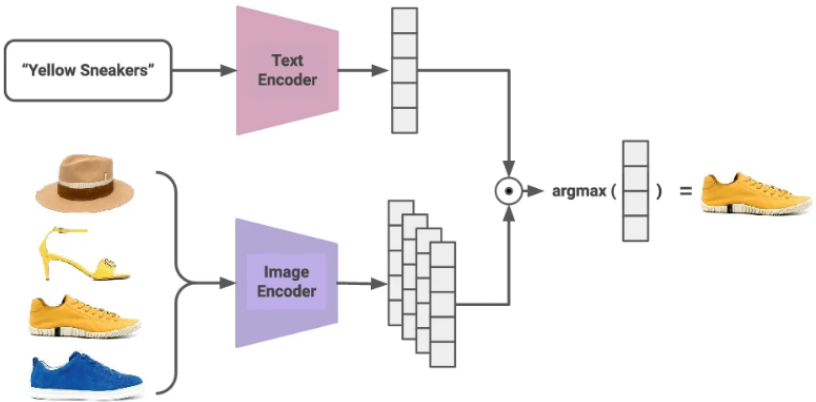
FashionCLIP is a domain-specific adaptation of the original CLIP model, fine-tuned on a large corpus of fashion-related images and textual descriptions.

It uses:

- **ViT-B/32 Transformer** as the image encoder.
- **Masked self-attention Transformer** as the text encoder.

Both encoders are trained jointly using a contrastive loss to maximize the similarity between corresponding (image, text) pairs.

FashionCLIP Overview



Shared Vector Space

FashionCLIP, like CLIP, creates a **shared vector space** for images and text.

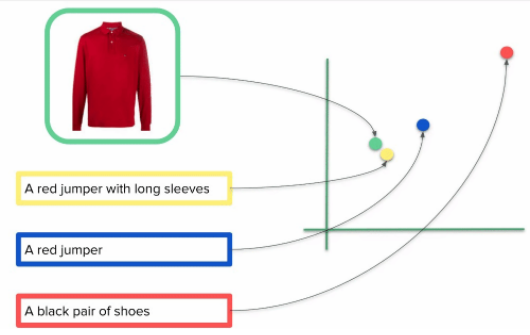
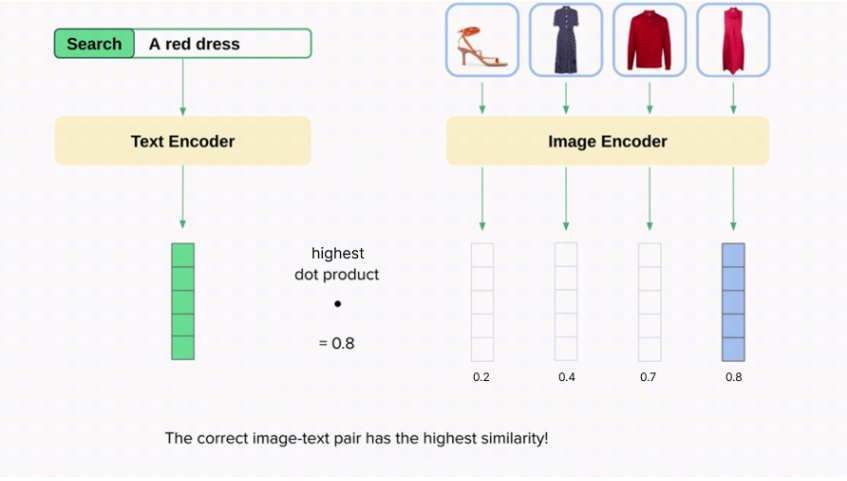


Image and Text Encoders



Similarity Computation

The similarity between an image and a text embedding is computed using **cosine similarity**:

$$\text{Similarity}(\mathbf{I}, \mathbf{T}) = \frac{\mathbf{I} \cdot \mathbf{T}}{\|\mathbf{I}\| \|\mathbf{T}\|}$$

Where:

- $\mathbf{I}$ = Image embedding vector
- $\mathbf{T}$ = Text embedding vector
- $\cdot$ = Dot product
- $\|\mathbf{I}\|, \|\mathbf{T}\|$ = Vector norms (magnitudes)

Usage in Our Project

In our system:

- FashionCLIP generates embeddings for **inventory images** and **textual descriptions**.
- These embeddings are stored in a **FAISS database** for fast similarity search.

At query time:

- User inputs an **image** (e.g., a cropped garment detected by YOLOS) or **text** (e.g., "red summer dress").
- FashionCLIP embeds the query.
- Closest matches from the inventory are retrieved based on cosine similarity.

Evaluation Metric: Recall@6

For text-based retrieval, we evaluated the model using the **Recall@K** metric, which measures the proportion of relevant items retrieved in the top-K results.

Given:

- $T = \{t_1, t_2, \ldots, t_n\}$: Text embeddings
- $I = \{i_1, i_2, \ldots, i_n\}$: Image embeddings
- Similarity: $s(t_k, i_j) = t_k^\top i_j$

$$\text{correct} = \sum_{k=1}^{n} \mathbf{1}\{k \in \text{Top}_5(t_k)\} \qquad \text{Recall} = \frac{\text{correct}}{n}$$

We used **Recall@6** for evaluation and achieved a score of **0.68**, indicating that 68% of the time, the relevant item is found in the top 5 results.

Demo

Challenges and Limitations

Challenges

- Finding a good dataset that includes both images and meaningful textual descriptions was a major challenge.
- Initially, we used the **Amazon UK Products Dataset 2023 (2.2M products)** to seed our inventory.
- However, the fashion product descriptions in this dataset were often not meaningful.
- We then attempted to scrape Amazon data manually, but scraping was very slow and triggered repeated Cloudflare human checks (likely due to not using a proxy).
- Finally, we settled on the dataset mentioned earlier that offered high-quality image-text pairs.

Limitations

While our project achieved its core objectives, some limitations remain:

- **Small Object Retrieval Challenges**
- **Lack of Multimodal Retrieval**
- **Fixed Inventory Limitation**
- **Limited Personalization**
- **Streamlit Deployment Constraints**

Conclusion

Conclusion

In this project, we developed a functional fashion retrieval system capable of text-based and image-based search. Leveraging **YOLOS-small** for object detection and **FashionCLIP** for embedding generation, we created a shared vector space that enabled efficient retrieval using FAISS.
Our system achieved a promising **Recall@6 of 0.68**, demonstrating effective matching between user queries and inventory items.
Overall, this project establishes a strong foundation for building more advanced, scalable, and user-personalized fashion retrieval systems.

Future Work

Future extensions to enhance and scale the system include:

- **Web Integration**
- **Mobile Application**
- **Personalized Recommendations**
- **Multi-clothing Detection**
- **Multimodal Search**
- **Inventory Sync**
- **Social Sharing**
- **Virtual Try-On**

Thank You!