Project Report on

Problem Statement 6

Deep Learning Project: AI-Powered Image Similarity Search and Recommendation System

College Name

Nutan College of Engineering and research
Talegaon Dabhade, Pune - 410507

Submitted By :

1) Samarth Suresh Ghare

2) Yogesh Sominath Kardile

3) Harshad Mahesh Kshirsagar

Under The Guidence of

Prof. Dipika Paranjape

Intel Technology India Private Limited

2025-26

Table of Contents

Sr. no.	Contents	Page no.
1	Introduction	3
2	Problem statement	4
3	Objectives	4
4	System Architecture	5
5	Dataset	7
6	Model Design	8
7	Training Strategy	10
8	Similarity Search	11
9	Evaluation	12
10	Web Application	13
11	Technologies Used	14
12	Conclusion	15
13	Future Enhancements	16
14	Applications	17

1. Introduction

The rapid expansion of e-commerce platforms and digital fashion marketplaces has resulted in massive online product catalogs, providing consumers with a wide range of choices. However, discovering visually similar products remains a major challenge, as traditional search systems rely heavily on textual metadata such as product names, tags, and descriptions. These methods often fail to represent important visual characteristics like color, texture, pattern, and style, leading to poor search relevance and increased effort for users.

To address these limitations, image-based similarity search has emerged as an effective alternative. By leveraging deep learning techniques, visual search systems can understand and compare images at a semantic level, allowing users to upload a reference image and retrieve visually similar items from large-scale datasets. This approach aligns naturally with user behavior in fashion discovery, where visual appearance plays a crucial role in product selection.

This project presents an AI Fashion Recommendation Search Engine, an end-to-end deep learning powered system designed for large-scale fashion datasets. It utilizes a Triplet Network with a ResNet50 backbone to learn discriminative image embeddings and employs FAISS for fast and scalable similarity search with GPU acceleration. The system is deployed using Streamlit for an interactive user interface and integrates Google Drive as a CDN for efficient image hosting, offering a practical and production-ready solution for visual product discovery.

The system is deployed using Streamlit, providing a simple and interactive web interface for image-based search. Large-scale image hosting is managed through Google Drive as a CDN, allowing efficient access to over 47,000 fashion images. Additionally, a Recall@K evaluation pipeline is included to quantitatively assess retrieval performance. Overall, the project demonstrates a scalable and production-ready solution for visual product discovery in modern e-commerce and fashion recommendation systems.

2. Problem Statement

With the rapid expansion of e-commerce platforms, especially in the fashion domain, online product catalogs have grown to include thousands or even millions of items. Existing search and recommendation systems primarily rely on textual information such as product names, categories, tags, and user-provided descriptions. However, fashion products are highly visual in nature, and critical attributes such as color combinations, fabric texture, patterns, design cuts, and overall style cannot be accurately or consistently expressed using text. This limitation often results in irrelevant search results and a poor user experience.

Users frequently encounter situations where they like a fashion item seen in an image—on social media, advertisements, or other websites—but are unable to find visually similar products using keyword-based searches. Even minor variations in naming conventions or incomplete metadata can prevent relevant items from being retrieved. This creates a significant gap between user intent and system output, reducing customer satisfaction and potentially impacting sales for e-commerce platforms.

Therefore, there is a strong need for an intelligent, image-based search system that allows users to find fashion products based purely on visual similarity. Such a system must be capable of learning meaningful visual representations from images, efficiently searching large-scale datasets in real time, and delivering accurate results through an interactive web interface. Addressing these challenges is essential to improve visual product discovery, enhance user engagement, and support scalable deployment in modern e-commerce environments.

3. Objectives

The primary objective of this project is to design and implement an AI-based fashion image similarity search system that enables users to find visually similar products by uploading an image. The system aims to learn robust visual embeddings using deep learning techniques, perform efficient similarity search using FAISS, and scale effectively to large fashion datasets containing over 50,000 images. Additionally, the project focuses on supporting real-time web deployment through an interactive and user-friendly interface.

4. System Architecture

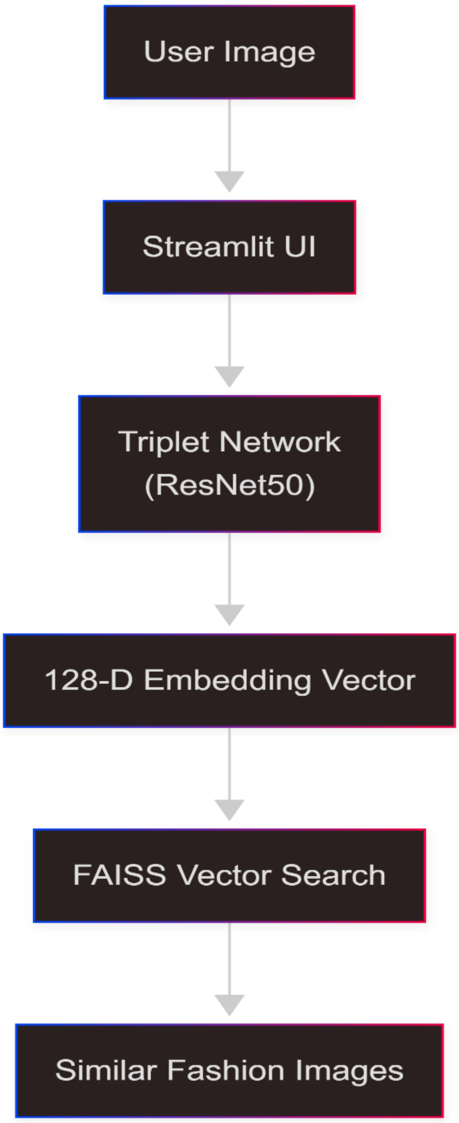

fig. 01 :- System Architecture

The proposed AI Fashion Recommendation Search Engine follows a modular and scalable architecture designed to efficiently process user images and retrieve visually similar fashion products in real time.

1. User Image Input

The workflow begins when a user uploads a fashion image through the application interface. This image can be a product photo, screenshot, or any fashion-related visual reference. The uploaded image serves as the query input for the similarity search system.

2. Streamlit User Interface

The uploaded image is first handled by the Streamlit-based web interface, which acts as the presentation layer of the system. Streamlit manages user interactions, image uploads, and result visualization. It also performs basic preprocessing steps such as image resizing and normalization before forwarding the image to the deep learning model.

3. Triplet Network (ResNet50 Backbone)

The preprocessed image is passed to the Triplet Network, which uses ResNet50 as the backbone architecture. This deep learning model has been trained using triplet loss with batch-hard mining to learn discriminative visual features. The network transforms the input image into a compact representation while preserving semantic visual similarities such as color, texture, shape, and style.

4. 128-Dimensional Embedding Vector

The Triplet Network outputs a 128-dimensional embedding vector for the input image. This vector represents the image in a high-dimensional feature space where visually similar fashion items are located closer together, and dissimilar items are positioned farther apart. These embeddings enable accurate and meaningful similarity comparisons.

5. FAISS Vector Search Engine

The generated embedding vector is then passed to the FAISS (Facebook AI Similarity Search) engine, which performs fast nearest-neighbor search against a pre-built index containing embeddings of all fashion images in the dataset. FAISS enables efficient similarity search even for large-scale datasets (50K+ images) and supports GPU acceleration for real-time performance.

6. Similar Fashion Image Retrieval

Based on cosine similarity (or distance metrics), FAISS retrieves the top-K most similar embedding vectors. The corresponding fashion images are fetched from the dataset (hosted via Google Drive CDN) and returned to the Streamlit interface, where they are displayed as similar fashion image recommendations to the user

5. Dataset

The dataset used in this project consists of **46,994 fashion images** collected to support large-scale image similarity learning and retrieval. The dataset is designed to represent a diverse range of fashion products commonly found on modern e-commerce platforms. It includes multiple fashion categories such as Dresses, T-shirts, Women's Fashion, and Mixed Fashion, ensuring broad visual diversity in terms of clothing styles, colors, textures, patterns, and designs. This diversity is essential for training a robust deep learning model capable of capturing fine-grained visual similarities across different fashion items.

All images are stored and managed using Google Drive, which serves as a scalable and cost-effective storage solution. Instead of downloading and storing the dataset locally, the project utilizes Google Drive CDN links to stream images dynamically during training, indexing, and inference. This approach significantly reduces local storage requirements and allows the system to scale efficiently without performance degradation. Each image is associated with a unique CDN URL, enabling fast and reliable access across different system components.

The dataset is organized in a structured manner, with images grouped according to their respective categories. This organization supports effective training of the Triplet Network, where visually similar items are sampled as positive pairs and dissimilar items as negative pairs. The presence of multiple categories and intra-category variations helps the model learn discriminative embeddings that accurately represent subtle visual differences such as fabric type, design patterns, and color combinations.

During preprocessing, images are resized and normalized to match the input requirements of the ResNet50 backbone network. The dataset is then used to generate embeddings for all images, which are stored and indexed using FAISS for efficient similarity search. The embeddings and their corresponding CDN image paths are maintained separately, allowing fast retrieval of images during inference without redundant data storage.

Overall, this dataset plays a crucial role in enabling high-accuracy visual similarity search. Its large scale, category diversity, and CDN-based streaming design make it well-suited for real-time, production-level deployment in fashion recommendation and e-commerce search applications.

6. Model Design

Component	Description
Backbone	ResNet50
Network Type	Triplet Network
Embedding Size	128
Loss Function	Triplet Margin Loss
Mining Strategy	Batch Hard Mining

Table No. 01 :- Model Design and its Components

The core of the AI Fashion Recommendation Search Engine is a deep learning model designed to learn meaningful and discriminative visual representations of fashion images. The model architecture and training strategy are carefully chosen to capture fine-grained visual similarities required for accurate fashion image retrieval.

Backbone Architecture – ResNet50

The model uses ResNet50 as its backbone network due to its strong feature extraction capabilities and proven performance in image recognition tasks. ResNet50 employs residual connections, which help in training deep networks by mitigating the vanishing gradient problem. This allows the model to learn rich hierarchical visual features such as edges, textures, shapes, and complex patterns that are essential for understanding fashion images.

Network Type – Triplet Network

A Triplet Network architecture is employed to learn similarity-based embeddings rather than performing traditional classification. The network processes three images simultaneously: an anchor image, a positive image (visually similar to the anchor), and a negative image (visually dissimilar). All three images are passed through the same ResNet50 backbone with shared weights, ensuring consistent feature extraction. This architecture enables the model to learn relative similarity relationships between images.

Embedding Size – 128 Dimensions

The output of the Triplet Network is a 128-dimensional embedding vector for each image. This compact representation balances descriptive power and computational efficiency. A lowerdimensional embedding reduces memory usage and accelerates similarity search while still preserving essential visual information required to distinguish between different fashion items.

Loss Function – Triplet Margin Loss

The model is trained using Triplet Margin Loss, which encourages the distance between the anchor and positive embeddings to be smaller than the distance between the anchor and negative embeddings by a defined margin. This loss function directly optimizes the embedding space for similarity search, ensuring that visually similar fashion items are clustered together while dissimilar items are pushed farther apart.

Mining Strategy – Batch Hard Mining

To further enhance training effectiveness, Batch Hard Mining is used to select the most challenging positive and negative samples within each training batch. By focusing on hard examples—where the model is most likely to make mistakes—the training process becomes more efficient and leads to faster convergence and improved retrieval accuracy.

7. Training Strategy

The training strategy of the proposed system is based on triplet learning, a powerful metric learning approach widely used for image similarity and retrieval tasks. Unlike traditional classification-based training, triplet learning focuses on learning a meaningful embedding space where visually similar images are positioned closer together, while visually dissimilar images are placed farther apart. This strategy is particularly suitable for fashion image similarity, where fine-grained visual differences play a critical role.

During training, the model processes images in the form of triplets, consisting of an anchor image (a), a positive image (p) that is visually similar to the anchor, and a negative image (n) that is visually different. All three images are passed through the same Triplet Network with shared weights to generate their respective embeddings. The goal of the training process is to ensure that the distance between the anchor and positive embeddings is smaller than the distance between the anchor and negative embeddings by a predefined margin.

The learning objective is enforced using the Triplet Margin Loss function, which is defined as

$$\mathbf{L=max(d(a,p)-d(a,n)+margin,0)}$$

where $d(a,p)$represents the distance between the anchor and positive embeddings, $d(a,n)$represents the distance between the anchor and negative embeddings, and *margin* is a hyperparameter that defines the minimum required separation between positive and negative pairs. If the constraint is already satisfied, the loss becomes zero, preventing unnecessary updates and stabilizing training.

To further improve learning efficiency, the training strategy incorporates Batch Hard Mining, which selects the hardest positive and negative samples within each training batch. Hard positives are those that are farthest from the anchor, while hard negatives are those closest to the anchor. Focusing on these challenging examples helps the model learn more discriminative features, accelerates convergence, and improves retrieval accuracy.

Overall, this training strategy enables the model to construct a highly structured embedding space optimized for similarity search. By combining triplet learning, margin-based loss optimization, and batch-hard mining, the system achieves robust and scalable performance suitable for large-scale fashion image retrieval applications.

8. Similarity Search

FAISS (Facebook AI Similarity Search) is used to perform fast nearest neighbor search on embedding vectors.

Index Type	IndexFlatIP
Similarity	Cosine Similarity
Hardware	GPU Accelerated

Table No. 02 :- Similarity Search

Similarity search is a critical component of the AI Fashion Recommendation Search Engine, responsible for retrieving visually similar fashion images efficiently from a large-scale dataset. Once the deep learning model generates fixed-length embedding vectors for all images, an efficient search mechanism is required to perform real-time nearest neighbor retrieval. For this purpose, the system uses FAISS (Facebook AI Similarity Search), a high-performance library developed by Facebook AI Research for large-scale vector similarity search.

The embedding vectors generated by the Triplet Network are indexed using IndexFlatIP, a FAISS index optimized for inner product similarity. Since all embedding vectors are L2-normalized, the inner product directly corresponds to cosine similarity, making it an effective and computationally efficient similarity metric. Cosine similarity measures the angular distance between vectors, which is well-suited for comparing learned image embeddings where magnitude is less important than direction.

To support fast retrieval over tens of thousands of fashion images, the FAISS index is built using GPU acceleration. By leveraging GPU parallelism, FAISS significantly reduces query latency and enables real-time similarity search even under high load. This allows the system to return top-K visually similar results almost instantly after a user uploads an image.

During inference, the query image embedding is computed by the trained Triplet Network and passed to the FAISS index. The index performs a nearest neighbor search to identify the most similar embedding vectors based on cosine similarity. The corresponding fashion images are then retrieved using their stored CDN paths and displayed to the user through the Streamlit interface.

9. Evaluation

The model was evaluated using Recall@K metric.

Metric Value

Recall@1 : 94.66%	Recall@5 : 98.90%

The performance of the proposed fashion image similarity system was evaluated using the Recall@K metric, which is widely used for assessing retrieval-based models. Recall@K measures the ability of the system to retrieve at least one relevant or visually similar item within the top *K* retrieved results. This metric is particularly suitable for image similarity and recommendation systems, where the objective is to ensure that correct matches appear among the top search results presented to the user.

In this project, the model achieved strong retrieval performance across different values of *K*. The Recall@1 score of 94.66% indicates that in most cases, the top-ranked result returned by the system is visually similar to the query image. This demonstrates the effectiveness of the learned embeddings in capturing fine-grained visual features. Furthermore, the Recall@5 score of 98.90% shows that nearly all relevant images are successfully retrieved within the top five results, highlighting the robustness and reliability of the similarity search pipeline.

The high recall values can be attributed to the use of a Triplet Network with batch-hard mining, which enables the model to learn highly discriminative embeddings. Additionally, the use of FAISS for similarity search ensures that nearest-neighbor retrieval is both accurate and efficient. These evaluation results confirm that the system performs well in large-scale fashion image retrieval scenarios and is suitable for real-world deployment in e-commerce and visual recommendation applications.

10. Web Application

A Streamlit web interface allows users to:

- Upload an image
- Get visually similar results
- View products directly from Google Drive CDN

The system includes a **web-based application developed using Streamlit**, which serves as the user interaction layer of the image similarity search engine. The web interface is designed to be simple, intuitive, and responsive, allowing users with minimal technical knowledge to easily interact with the system. Streamlit enables rapid deployment of machine learning applications and provides seamless integration with the backend inference pipeline.

Through the web application, users can **upload a fashion image** directly from their device. Once the image is uploaded, it is preprocessed and passed to the trained Triplet Network to generate a 128-dimensional embedding vector. This embedding is then used to perform similarity search through the FAISS index. The application retrieves the most visually similar fashion images and displays them to the user in real time.

The retrieved results are presented as image thumbnails, and each product image is served using **Google Drive CDN links**. This allows users to view high-quality images without requiring local storage or additional downloads. The integration of Streamlit with FAISS and Google Drive CDN ensures low latency, smooth user experience, and scalability. Overall, the web application provides an effective interface for real-time visual fashion search and demonstrates the practical usability of the system in real-world e-commerce scenarios.

11. Technologies Used

Tool	Purpose
PyTorch	Model training
FAISS	Vector search
Streamlit	Web UI
Google Drive API	Dataset hosting
CUDA	GPU acceleration

Table No. 03 :- Technologies and tools used

PyTorch is used as the primary deep learning framework for model development and training. It provides flexibility in building custom neural network architectures such as the Triplet Network and supports efficient GPU-based training. PyTorch also simplifies experimentation, loss function implementation, and evaluation of deep learning models.

FAISS (Facebook AI Similarity Search) is employed for high-speed vector similarity search. It enables efficient indexing and nearest neighbor retrieval of high-dimensional embedding vectors, even for large datasets. FAISS supports GPU acceleration, which significantly reduces search latency and allows real-time similarity retrieval.

Streamlit is used to develop the web-based user interface. It allows rapid deployment of machine learning applications and provides a clean, interactive interface for uploading images and displaying similarity results. Streamlit bridges the gap between complex backend processing and user-friendly frontend interaction.

Google Drive API is utilized for dataset hosting and image delivery. By serving images through Google Drive CDN links, the system avoids local storage constraints while ensuring fast and reliable access to a large-scale image dataset. This approach also simplifies dataset management and scalability.

CUDA is leveraged to accelerate both model inference and FAISS similarity search operations. GPU acceleration significantly improves computational performance, enabling low-latency responses and making the system suitable for real-time, production-level deployment.

12. Conclusion

This project successfully demonstrates the design and implementation of an efficient and scalable image similarity search system tailored for fashion applications. By leveraging deep learning–based visual embeddings generated using a Triplet Network with a ResNet50 backbone, the system effectively captures fine-grained visual features such as color, texture, and style. The integration of FAISS for vector similarity search enables fast and accurate retrieval of visually similar fashion items, even when operating on large-scale datasets.

The use of GPU acceleration ensures real-time performance, making the system suitable for practical deployment in modern e-commerce environments. Additionally, the Streamlit-based web interface provides an intuitive and user-friendly platform that allows users to upload images and instantly receive relevant visual recommendations. Hosting images via Google Drive CDN further enhances scalability by reducing local storage requirements while maintaining fast image access.

Overall, the project highlights how deep learning and efficient similarity search techniques can overcome the limitations of traditional text-based search systems. The proposed solution improves product discovery, enhances user experience, and demonstrates strong potential for real-world adoption in fashion recommendation systems and visual search applications.

13. Future Enhancements

- **Hybrid Text and Image Search** – Combine keyword-based and image-based queries to improve search accuracy and flexibility.
- **Category-Based Filtering** – Enable filtering by category, gender, style, or clothing type to refine search results.
- **Mobile Application Integration** – Develop Android/iOS applications to allow users to search using smartphone cameras.
- **Cloud GPU Deployment** – Deploy the system on cloud-based GPU infrastructure for better scalability and high availability.
- **Multi-Modal Embeddings** – Learn joint embeddings from both images and text to enhance recommendation quality.
- **Personalized Recommendations** – Incorporate user preferences and browsing history for personalized fashion suggestions.
- **Incremental Index Updates** – Support real-time addition of new products without rebuilding the entire FAISS index.
- **Advanced FAISS Indexing** – Use optimized indexes (IVF, HNSW) for faster retrieval on extremely large datasets.
- **Explainable Recommendations** – Provide visual or textual explanations for why certain products are recommended.
- **Cross-Domain Search** – Extend the system to support other domains such as accessories, footwear, or lifestyle products.

14. Applications

- **E-commerce Search** – Enables users to find fashion products using images instead of text-based queries.
- **Recommendation Systems** – Provides visually similar product recommendations based on user preferences.
- **Visual Product Discovery** – Helps users discover new products through image-based exploration.
- **Duplicate Product Detection** – Identifies duplicate or near-duplicate images in large product catalogs.
- **Catalog Management** – Assists in organizing and clustering large fashion inventories based on visual similarity.
- **Trend Analysis** – Analyzes visual patterns and styles to identify emerging fashion trends.
- **Content Moderation** – Detects visually similar or restricted fashion items to enforce platform policies.
- **Brand Monitoring** – Helps brands track visually similar products across platforms for copyright and brand protection.
- **Cross-Selling and Up-Selling** – Recommends visually related items to increase average order value.
- **Second-Hand and Resale Platforms** – Enables image-based search for similar products in thrift and resale marketplaces.