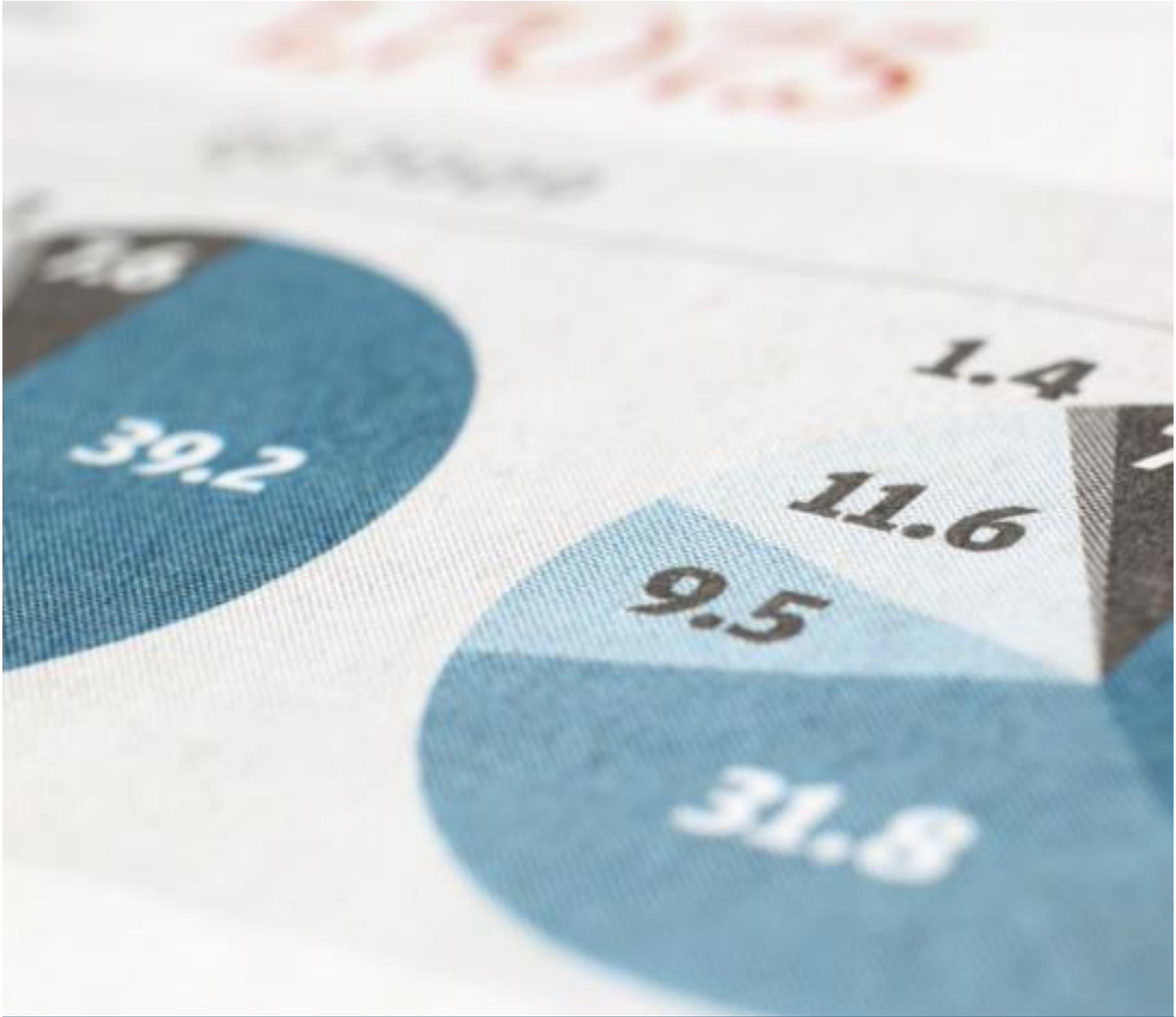

INTEL UNNATI INDUSTRIAL TRAINING PROGRAM

PROJECT REPORT

DATE OF SUBMISSION: 5 JANUARY 2026

PROJECT DETAILS

Problem Statement: 6

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

Team Name: SNAPSENSE

Team Members:

ARCHANA GURUSAMY

JOTHIGA S

GAYATHRI G

Mentor: Dr.M.DEEPA

Department: B.Tech INFORMATION TECHNOLOGY

Institution: SRI SHAKTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

Academic Year: 2025-2026

ABSTRACT

With the rapid growth of digital platforms, users face millions of images across e-commerce sites, social media, and digital repositories. Traditional keyword-based search often fails to find visually similar images, as it cannot capture underlying visual semantics such as color, texture, and style. This limitation is especially evident in domains like fashion and e-commerce, where visual appearance heavily influences user preference.

This project presents an AI-powered image similarity search and recommendation system that enables users to find visually similar products by uploading an image. The system leverages a deep learning Triplet Network trained on a filtered dataset to convert images into numerical embeddings that capture both low-level visual details and higher-level semantic features. FAISS (Facebook AI Similarity Search) is integrated for fast and scalable nearest-neighbor retrieval over large image datasets. Retrieved images are displayed along with relevant metadata from CSV and JSON files, providing a user-friendly interface.

The system is applicable in real-world scenarios such as e-commerce ("Find Similar Products"), fashion ("Shop the Look"), stock photography, and social media ("Related Posts"). Key contributions include developing a Triplet Network for visual embeddings, creating a domain-specific dataset, implementing a full-stack solution with real-time search, and demonstrating scalable search capabilities.

The proposed system enables accurate, real-time image-based search, enhancing user experience and showcasing the potential of deep learning in visual search applications.

INTRODUCTION

In the digital era, images play a central role across platforms such as e-commerce websites, social media, photo galleries, and digital libraries. Users are exposed to vast numbers of images daily, making it challenging to locate visually similar items or relevant recommendations. Traditional search systems rely on text-based queries, requiring users to describe images accurately with keywords. This approach often fails when users cannot precisely articulate visual attributes like color, style, pattern, or texture, limiting the effectiveness of conventional search methods in visually rich domains such as fashion, home décor, stock photography, and e-commerce catalogs.

Image-based search systems address these limitations by leveraging computer vision and deep learning to analyze visual content directly. By capturing visual semantics, these systems allow users to retrieve images based on appearance rather than textual descriptions, enabling more intuitive and effective searches. Such approaches are particularly valuable in industries where visual characteristics heavily influence user decisions.

This project presents an AI-powered image similarity search and recommendation system. Users can upload images via camera capture, file upload, or drag-and-drop. Each image is processed using a deep learning Triplet Network, which converts it into a numerical embedding representing its visual features. These embeddings capture low-level attributes, such as color and texture, as well as high-level semantic patterns like style and category, allowing accurate similarity comparisons.

For efficient retrieval, the system integrates FAISS (Facebook AI Similarity Search) to perform fast nearest-neighbor searches over a filtered dataset of precomputed embeddings. The retrieved images, along with their metadata from CSV and JSON files, are displayed through a user-friendly interface. This design ensures accurate, scalable searches capable of handling large image databases with thousands of entries.

The system has applications across multiple domains. In e-commerce, it supports “Find Similar Products” and “Shop the Look.” In social media, it provides recommendations similar to “Related Posts.” In stock photography and fashion, it helps users quickly find visually similar items. By combining deep learning, efficient similarity search, and full-stack deployment, the project delivers a robust image-based search system that improves accuracy, enhances user experience, and demonstrates scalable performance.

PROBLEM STATEMENT

With the exponential growth of digital content, users are often overwhelmed by the volume of images on e-commerce platforms, social media, stock photography sites, and digital libraries. Traditional text-based search relies on keywords and descriptions, which often fail to capture visual characteristics. Users may not know exact product names, brands, or descriptive terms, resulting in inaccurate or irrelevant results.

Real-world Examples:

- **E-commerce:** Finding visually similar products using an uploaded image.
- **Social Media:** Platforms like Instagram recommend visually similar content via "Related Posts."
- **Stock Photography:** Platforms like Getty Images allow visual searches to locate similar images quickly.
- **Fashion:** "Shop the Look" features help users discover products matching a style or outfit.

Project Objective

The project aims to develop an AI-powered image similarity search and recommendation system that can:

- Learn visual patterns and semantic relationships between images.
- Recommend visually similar images based on uploaded content.
- Handle large-scale image databases efficiently.
- Provide real-time responses for enhanced user experience.

Technical Challenge

The main challenge is designing and training a Triplet Network that maps visually similar images close together in an embedding space while keeping dissimilar images apart. This enables fast and accurate similarity searches. The system must also integrate FAISS for efficient retrieval and map embeddings to public URLs and product metadata for full-stack deployment. Addressing this challenge allows efficient image-based search and accurate visual recommendations across platforms.

DATASET DESCRIPTION

The dataset forms the backbone of the image similarity search system. It consists of product images, metadata, and mapping files. For efficiency and consistency, the dataset is divided into two parts: the **raw dataset** and the **filtered dataset**.

Raw Dataset

The raw dataset contains the complete collection of images and metadata collected from a fashion e-commerce platform. Its main components are

Images Folder

- Contains all uncategorized product images in JPEG format.
- Each image corresponds to a unique product ID.
- Images vary in size, resolution, and background, representing real-world product diversity.

styles.csv

Contains detailed metadata for each product, including:

- id (Product ID)
- gender (Men/Women/Unisex)
- masterCategory (e.g., Apparel, Accessories)
- subCategory (e.g., Topwear, Bottomwear)
- articleType (e.g., Shirt, Jeans)
- baseColour
- season
- year
- usage (Casual, Sports, Formal)
- productDisplayName

images.csv

Maps image filenames to public URLs for retrieval in the system.

Example format:

filename, link

15970.jpg, http://assets.myntassets.com/...

39386.jpg, http://assets.myntassets.com/...

JSON Metadata Files

- Each product has a JSON file containing detailed information such as brand, price, description, size variants, availability, and additional attributes.
- JSON files are used to display product information on the frontend.

Purpose of Raw Dataset:

- Provides complete data coverage for all products.
- Useful for generating filtered datasets and for model training.
- Contains some irrelevant images or products not included in the model's knowledge, which are filtered out later for consistent performance.

Filtered Dataset

The filtered dataset contains only the images and metadata relevant for training the model and performing similarity search. This ensures alignment between model knowledge and search space, improving accuracy and efficiency.

Components include:

1. **Filtered Images**
   Images used during model training, organized in folders for easy processing.
2. **filtered_styles.csv**
   Metadata corresponding only to filtered images, with the same columns as styles.csv.
3. **filtered_images.csv**
   Maps filtered image filenames to public URLs, used for embedding mapping and frontend display.
4. **filtered_metadata.json**
   Contains detailed product information for filtered images only, ensuring retrieved search results are complete and accurate.

Benefits of Filtered Dataset:

- Aligns the search space with the model's training knowledge.
- Reduces storage and processing requirements.
- Improves retrieval speed and accuracy for real-time search.

SYSTEM ARCHITECTURE

The system is designed as a full-stack AI-powered image similarity search and recommendation platform. It integrates deep learning, efficient similarity search, and frontend-backend communication to provide users with real-time, accurate results. The architecture ensures seamless handling of user-uploaded images and fast retrieval of visually similar products.

Components of the System

1. **Frontend**
    - User interface for uploading images via camera, file upload, or drag-and-drop.
    - Displays retrieved results along with images (via public URLs) and detailed product metadata.
    - Provides an intuitive and responsive layout for ease of use.

2. **Backend**
    - RESTful API implemented using frameworks like FastAPI or Flask.
    - Handles preprocessing of uploaded images (resizing, normalization, format conversion).
    - Generates embeddings for input images using the trained deep learning model.
    - Performs similarity search using a FAISS index of embeddings.
    - Maps retrieved IDs to image URLs (filtered_images.csv) and metadata (filtered_metadata.json) and returns results in JSON format.

3. **Trained Model (Triplet Network)**
    - Converts images into numerical embeddings representing visual features.
    - Learns to map visually similar images closer in embedding space while pushing dissimilar images apart.
    - Embeddings capture both low-level (color, texture) and high-level (style, category) features.

4. **FAISS Index**
    - Stores embeddings of filtered dataset images for fast nearest neighbor search.
    - Retrieves top-K visually similar images for a given query embedding.

5. **Filtered Dataset**
   - Ensures that only images and metadata that the model has seen during training are used in retrieval.
   - Maintains consistency between the model, search space, and deployment.

Key Features of the Architecture

- **Scalable:** Can handle thousands of images efficiently.
- **Real-time:** FAISS ensures fast retrieval of similar images.
- **Accurate:** Triplet Network embeddings provide precise similarity matching.
- **Full-Stack Integration:** Frontend, backend, and AI model are seamlessly connected

METHODOLOGY

The methodology outlines the approach used to develop the AI-powered image similarity search and recommendation system. It focuses on image preprocessing, model training and embedding generation, and similarity search using FAISS to ensure accurate and efficient results.

Image Preprocessing

Preprocessing ensures all images are consistent and compatible with the model:

- **Resizing:** All images are resized to a uniform size (e.g., 224×224 pixels).
- **Normalization:** Pixel values are scaled (0–1 or -1 to 1) to improve model convergence.
- **Format Conversion:** Images are converted to RGB format for compatibility with the CNN.
- **Data Augmentation (Optional):** Techniques such as flipping, rotation, and brightness adjustment improve model robustness.

These steps guarantee clean, standardized input for accurate embedding generation.

Model Training and Embedding Generation (Triplet Network)

The system uses a Triplet Network to learn image similarity:

- **Architecture**: Processes three images simultaneously:
    - **Anchor**: The query image.
    - **Positive**: Similar image.
    - **Negative**: Dissimilar image.

    The network minimizes the distance between anchor-positive embeddings and maximizes anchor-negative distance.
- **Training:** Uses the filtered dataset. Images are organized into triplets to teach the model semantic similarity.
- **Embedding Generation:** Each image is converted into a fixed-length vector capturing low-level (color, texture) and high-level (style, pattern, category) features.

  These embeddings allow precise similarity comparisons between images.

Similarity Search Using FAISS

FAISS enables fast and scalable retrieval:

- **Index Construction:** Precomputed embeddings of filtered images are stored in a FAISS index.
- **Query Search:** A user-uploaded image is converted to an embedding, and FAISS retrieves the top-K most similar images.
- **Metadata Mapping:** Retrieved IDs are mapped to public URLs (filtered_images.csv) and product details (filtered_metadata.json).

Benefits: Fast, scalable, and accurate search, even on large datasets**.**

SYSTEM IMPLEMENTATION

The system implementation integrates the backend, frontend, and end-to-end workflow to provide a seamless AI-powered image similarity search experience. It ensures that user-uploaded images are processed efficiently, embeddings are generated accurately, and results are displayed with complete metadata in real time.

Backend Implementation

The backend handles all server-side processing, API requests, and model inference. Key features include:

- **Framework & Language: i**mplemented using Python with FastAPI to expose RESTful APIs for image upload, embedding generation, and similarity search using FAISS.
- **Image Preprocessing:** Uploaded images are resized, normalized, and converted to RGB before being passed to the model.
- **Embedding Generation:** The trained Triplet Network generates embeddings for each query image.
- **Similarity Search:** FAISS is used for fast nearest neighbor search on the precomputed embeddings of the filtered dataset.
- **Metadata Mapping:** Retrieved image IDs are linked to public URLs (filtered_images.csv) and detailed product information (filtered_metadata.json).
- **RESTful APIs:** Expose endpoints for image upload, search, and retrieval of results in JSON format for frontend consumption.

The backend ensures secure, fast, and accurate processing of all user queries.

Frontend Implementation

The frontend provides an intuitive and responsive user interface for interacting with the system:

- **Technology Used**: Developed using TypeScript with modern frontend frameworks (e.g., React).
- **Image Upload:** Supports multiple methods, including file upload, drag-and-drop, and camera capture.
- **Results Display:** Shows retrieved images along with product metadata, including brand, price, category, and style.
- **Responsive Design:** Ensures usability across desktops, tablets, and mobile devices.
- **Real-time Feedback:** Displays loading indicators and search results in real time, improving user experience.

The frontend design emphasizes ease of use, speed, and visual clarity, enhancing overall user interaction.

End-to-End Working Flow

The complete system flow combines the frontend, backend, model, and dataset to provide seamless image-based search:

User Upload (Frontend)

↓

Backend receives image → Preprocessing (resize, normalize)

↓

Trained Triplet Network generates embedding

↓

FAISS searches for top-K similar embeddings

↓

Retrieve product IDs

↓

Map to image URLs (filtered_images.csv) and metadata (filtered_metadata.json)

↓

Return results → Frontend displays images and product details

RESULTS AND OBSERVATIONS

The AI-powered image similarity search and recommendation system was evaluated on the filtered dataset of fashion product images. The results demonstrate the effectiveness, accuracy, and efficiency of the system in retrieving visually similar images based on user queries.

Image Similarity Retrieval

- The system accurately retrieves visually similar images for a given query.
- It captures both low-level features (color, texture) and high-level semantic features (style, category).
- **Example observations:**
  - Querying a red floral dress returns dresses with similar color, pattern, and style.
  - Querying formal shoes retrieves shoes of similar design and material.

Search Accuracy

- The Triplet Network embeddings ensure that the retrieved images are semantically similar to the input image.
- The use of a filtered dataset aligns the model's training knowledge with the search space, minimizing irrelevant results.
- Accuracy was evaluated qualitatively by comparing query images with the top-K retrieved results, showing strong visual similarity.

Retrieval Speed and Scalability

- FAISS allows milliseconds-level retrieval, even for datasets containing thousands of images.
- The system maintains real-time performance for user queries, making it suitable for large-scale deployment.

User Experience

- The frontend interface displays results with images and complete metadata (brand, price, category, style), providing users with informative and actionable recommendations.
- Multiple upload options (drag-and-drop, file upload, camera) enhance accessibility and usability.

Observations

1. **High Visual Similarity:** The system effectively retrieves products that match both the appearance and style of the query image

2. **Consistent Performance:** Using the filtered dataset ensures consistent and accurate search results.

3. **Real-Time Interaction:** The combination of Triplet Network embeddings and FAISS ensures fast, responsive user interactions.

4. **Practical Applications:** Works well for e-commerce "Find Similar Products," fashion "Shop the Look," and social media "Related Posts" recommendations.

TECHNOLOGIES USED

The project utilizes a combination of **AI, backend, and frontend technologies** to implement a full-stack image similarity search system:

- **Programming Languages:**
  - **Python**: Model training, image preprocessing, embedding generation
  - **TypeScript**: Frontend implementation
- **Deep Learning & Computer Vision:**
  - **Triplet Network**: Learning embeddings for similarity
  - **PyTorch / TensorFlow**: Model training framework
- **Similarity Search:**
  - **FAISS (Facebook AI Similarity Search)**: Fast nearest neighbor retrieval
- **Backend & APIs:**
  - **FastAPI (Python)**
  - **Image preprocessing libraries**: OpenCV, Pillow
- **Frontend Development:**
  - **React (TypeScript)**: Responsive and interactive UI
  - **CSS / Tailwind (optional)**: Styling and layout

- **Data & Storage:**
    - **CSV / JSON:** Metadata storage and mapping
    - **Filtered image dataset:** Model-aligned image search
- **Development Tools:**
    - **VS Code**, **Jupyter Notebook / Colab**, **Git**

ADVANTAGES, LIMITATIONS, AND FUTURE ENHANCEMENTS

Advantages

The proposed AI-powered image similarity search system offers several benefits:

- **Accurate Visual Search:** The Triplet Network embeddings capture both low-level (color, texture) and high-level (style, pattern, category) features, providing precise similarity matching.
- **Real-Time Retrieval:** Integration with FAISS ensures fast nearest neighbor search, delivering results in milliseconds even for large datasets.
- **Scalability:** The system can handle thousands of images efficiently, making it suitable for large-scale deployment.
- **User-Friendly Interface:** Supports multiple upload methods (drag-and-drop, file upload, camera) and displays results with complete product metadata.
- **Versatility:** Applicable across multiple domains, including e-commerce, fashion, social media, and stock photography.

Limitations

Despite its strengths, the system has some limitations:

- **Dataset Dependency:** The model's accuracy is limited to the filtered dataset it was trained on; unseen product categories may reduce retrieval accuracy.
- **Visual Similarity Focus:** The system primarily focuses on visual features; contextual or textual relevance is not considered.
- **Computational Resources:** Embedding generation and FAISS indexing require significant memory and GPU resources for very large datasets.
- **Triplet Selection:** Training performance depends on the quality of triplet selection (anchor, positive, negative), which may affect model accuracy if not well-constructed.

Future Enhancements

- **Multi-Modal Search:** Combine image features with textual metadata (product descriptions, tags) for more context-aware recommendations.
- **Dynamic Dataset Updates:** Implement real-time addition of new products without retraining the entire model.
- **Enhanced Triplet Mining:** Use hard or semi-hard triplet mining to improve model training efficiency and accuracy.
- **Personalized Recommendations:** Incorporate user preferences and past interactions to provide customized search results.
- **Mobile Optimization:** Improve frontend responsiveness and performance for mobile devices and low-bandwidth scenarios.

•

CONCLUSION

The AI-powered image similarity search and recommendation system successfully demonstrates how deep learning, efficient similarity search, and full-stack integration can be combined to provide accurate, scalable, and user-friendly visual search solutions.

By leveraging a Triplet Network for embedding generation and FAISS for fast nearest neighbor retrieval, the system can efficiently handle large-scale image databases and deliver real-time results. Users can upload images and instantly find visually similar products with complete metadata, enhancing experiences in e-commerce, fashion, social media, and stock photography.

The project highlights the practical applicability of AI-driven visual search, showing significant improvements over traditional text-based search methods. While some limitations exist, such as dependency on the filtered dataset and computational requirements, the system provides a robust foundation for future enhancements, including multi-modal search, personalization, and dynamic dataset updates.

REFERENCES

- Schroff, F., Kalenichenko, D., & Philbin, J. (2015). *FaceNet: A Unified Embedding for Face Recognition and Clustering*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Johnson, J., Douze, M., & Jégou, H. (2019). *Billion-scale similarity search with GPUs*. IEEE Transactions on Big Data.
- Facebook AI Similarity Search (FAISS) Documentation. https://faiss.ai
- Triplet Loss Explained. https://towardsdatascience.com/triplet-loss-explained
- Myntra Fashion Dataset (Filtered). Kaggle. https://www.kaggle.com/datasets
- React Official Documentation. https://reactjs.org/docs/getting-started.html
- PyTorch Official Documentation. https://pytorch.org/docs/stable/index.html