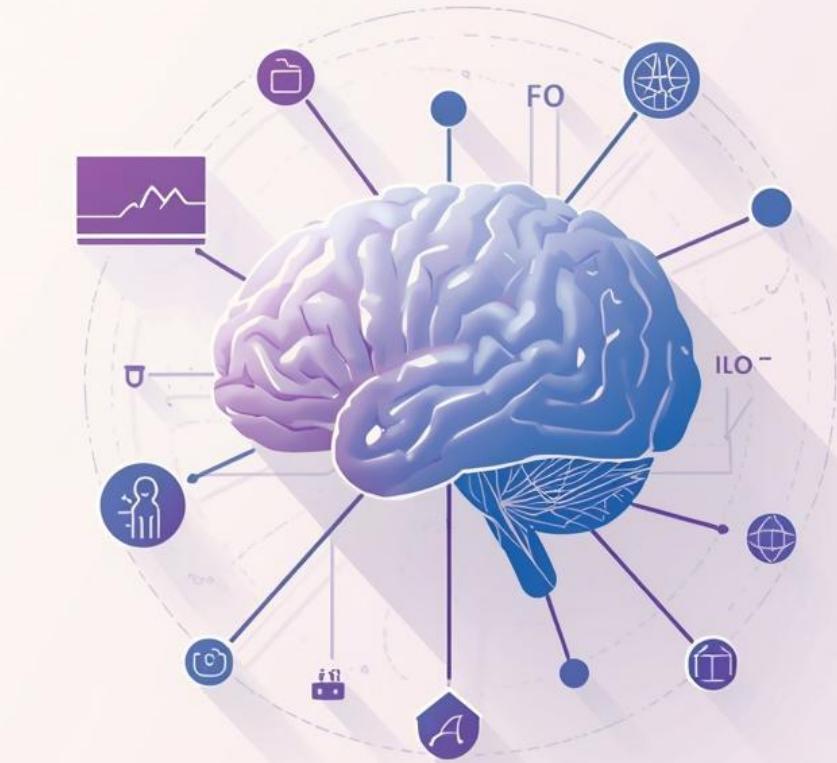


AI-Powered Image Similarity Search Engine

Group name: CodeNiti

By Yash Anokar & Ojas Kamlekar

Intel® Unnati Industrial
Training Program 2024.



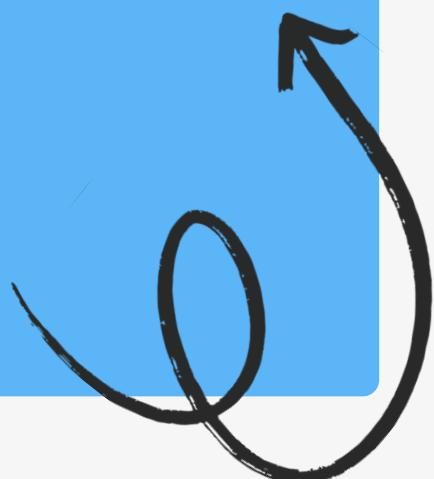
Introduction to Image Search

Project Objective:

- Learn visual patterns and semantic relationships between images
- Recommend similar images based on visual content alone
- Handle large-scale databases efficiently (1000+ images)
- Provide real-time responses for user queries

Technical Challenge:

Build a deep learning model using Triplet Networks that learns to map visually similar images closer together in an embedding space, enabling fast and accurate similarity search.



Project Structure

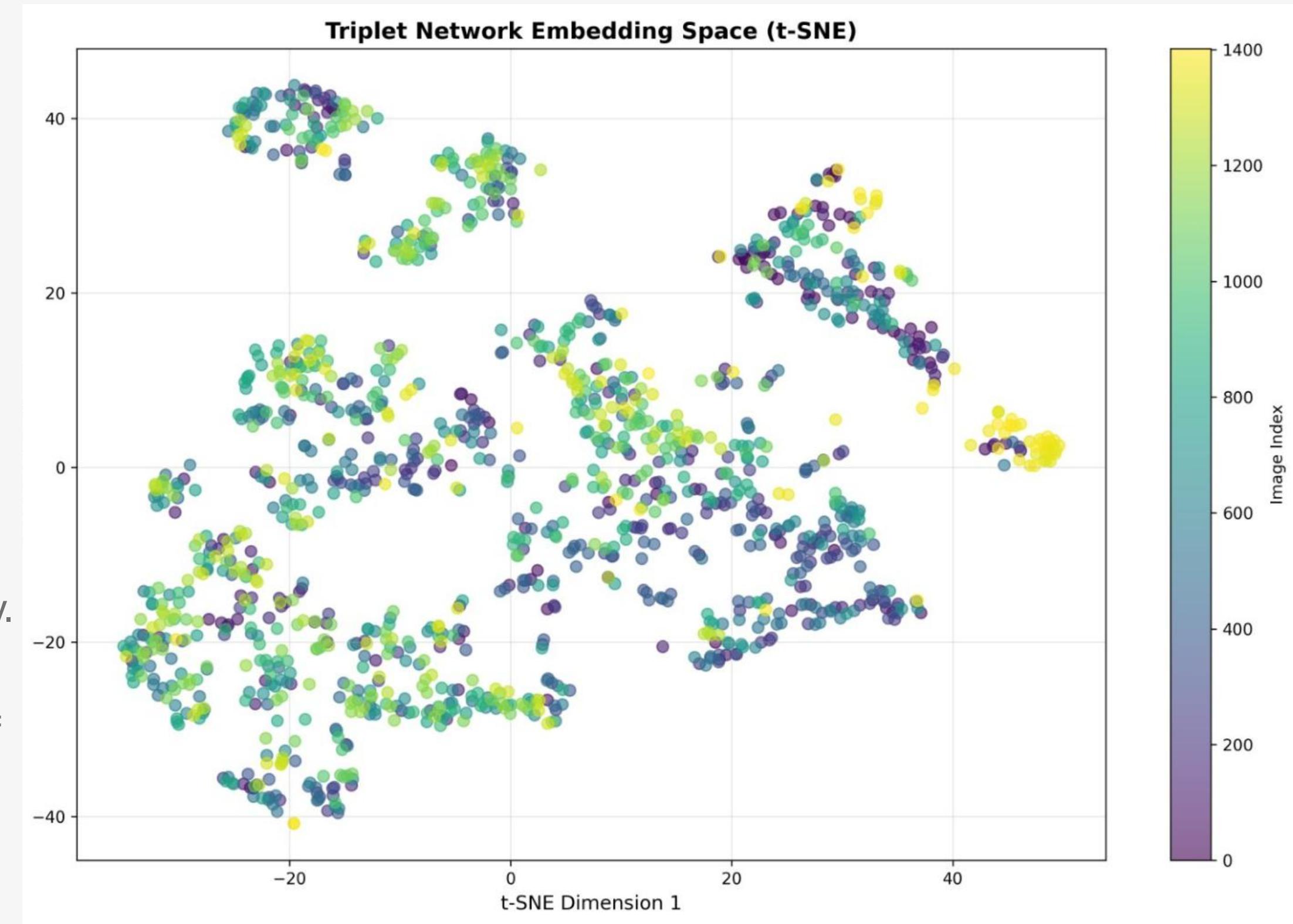
- triplet_data_generator.py – Loads images and creates anchor–positive–negative triplets for training.
- triplet_model.py – Defines the Triplet Network architecture used for learning image embeddings.
- train_triplet.py – Handles the training process of the Triplet Network.
- extract_features_triplet.py – Converts images into embedding vectors for similarity comparison.
- app_triplet.py – Flask-based backend server that exposes APIs for image similarity search.
- templates/ – Contains HTML files for the web interface.
- static/uploads/ – Stores images uploaded by users.
- static/dataset/ – Contains the training image dataset.
- checkpoints/ – Saves trained model weights during training.



Triplet Network Architecture Explained

Siamese neural network with three branches (anchor, positive, negative) sharing weights, enabling metric learning. Contains 5.1 million trainable parameters optimized for image similarity.

Implements margin-based metric learning with $L = \max(0, d(A,P) - d(A,N) + 0.2)$, ensuring similar images are mapped closer together.



Embedding Space and Similarity Metrics

Generates compact, L2-normalized 128-dimensional feature vectors for efficient storage and fast similarity computation

Mixed Precision (BF16) Optimization

Accelerates model training and inference by 2-3x, significantly boosting performance without compromising accuracy.

Custom CNN Training

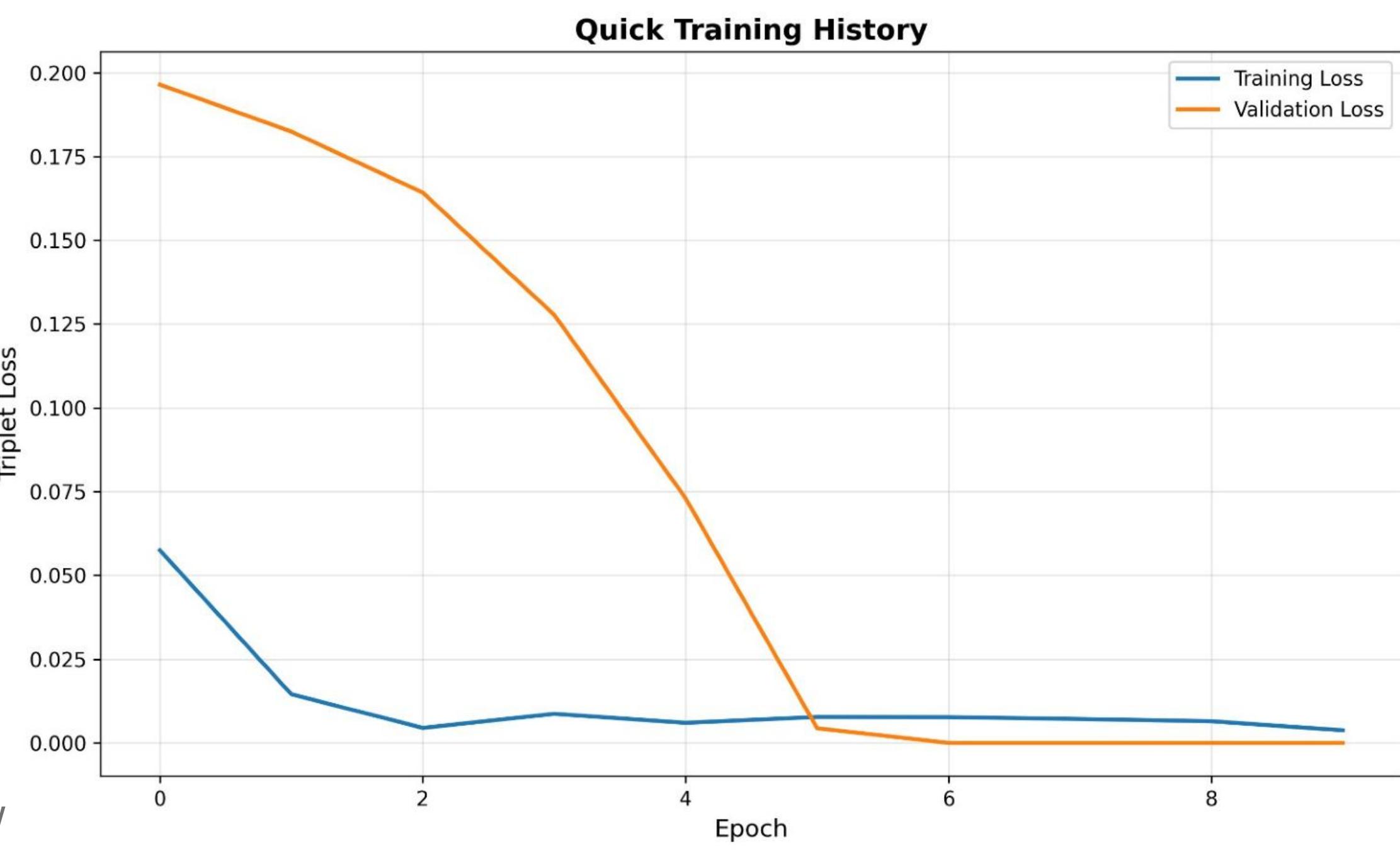
A 4-layer convolutional architecture with batch normalization and dropout learns hierarchical visual features from low-level edges to high-level semantic patterns.

Multi-Feature Extraction Pipeline

Combines deep learning features (40%), color histograms (35%), texture (15%), and shape descriptors (10%) for comprehensive visual understanding.

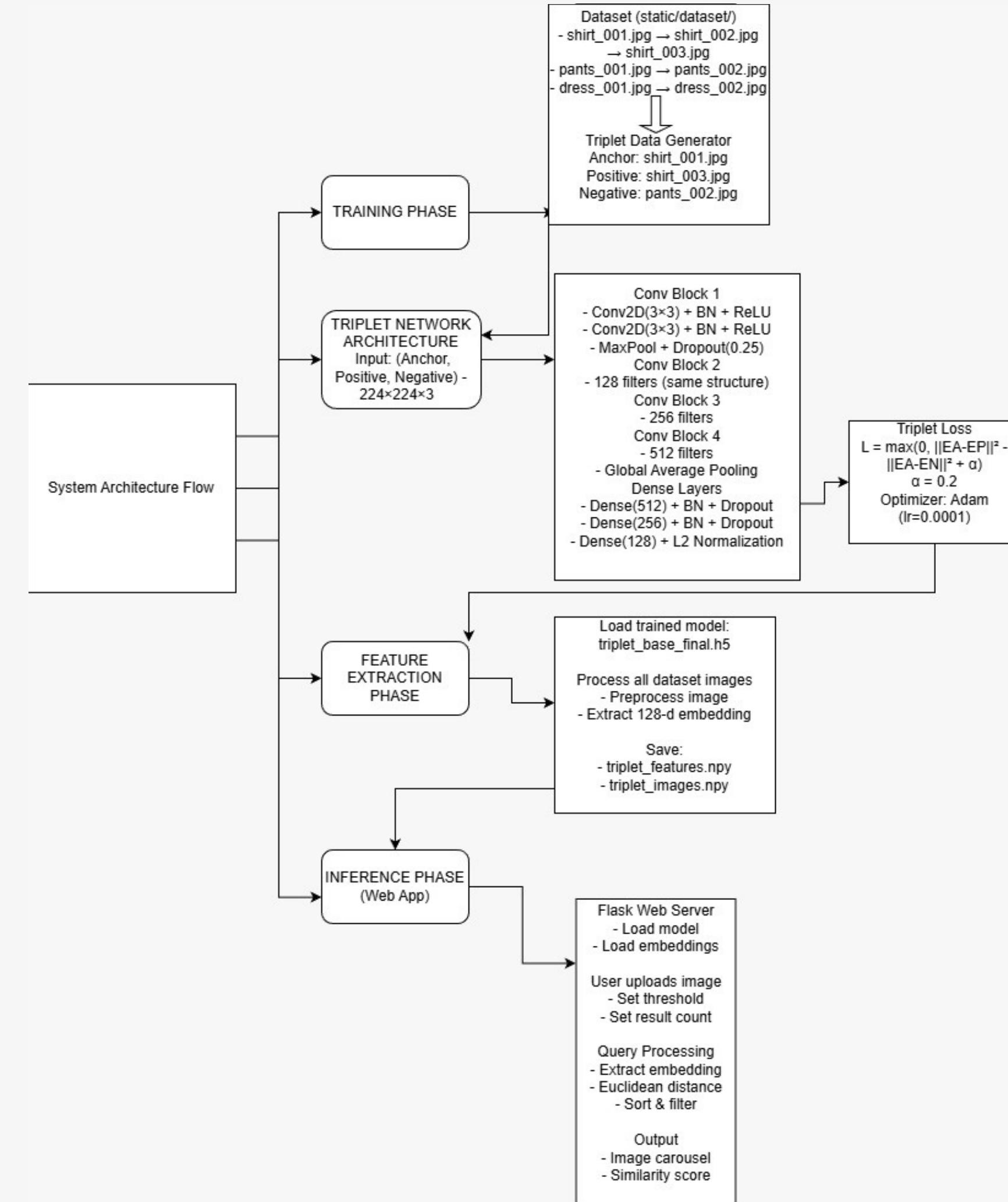
Training History and Model Improvement

The training process involved monitoring **loss curves** and accuracy metrics, demonstrating how iterative training enhances the quality of embeddings, progressively clustering similar images closer in the embedding space.



System Architecture Overview

The system architecture encompasses key components such as image upload, embedding extraction, similarity search, and API endpoints, ensuring seamless integration and efficient data flow within the image similarity recommendation system.



Backend Implementation Using FastAPI

The backend is designed with **Python FastAPI**, facilitating swift API responses for embedding retrieval and similarity queries, optimizing performance to handle large datasets efficiently and seamlessly integrate into the overall system architecture.



Embedding space Visualization

This visualization illustrates how images are represented as numerical vectors in an embedding space. Before training, image embeddings are randomly distributed, showing no clear structure. After training the Triplet Network, embeddings of visually similar images move closer together, forming well-defined clusters (e.g., shirts, pants, dresses), while dissimilar images are pushed farther apart. This organized structure enables efficient and accurate image similarity search based on visual features rather than textual labels.

Embedding Space Visualization

Before Training:
Random Distribution



After Training:
Organized Clusters

Shirts: • • •

• • •

Pants: • • •

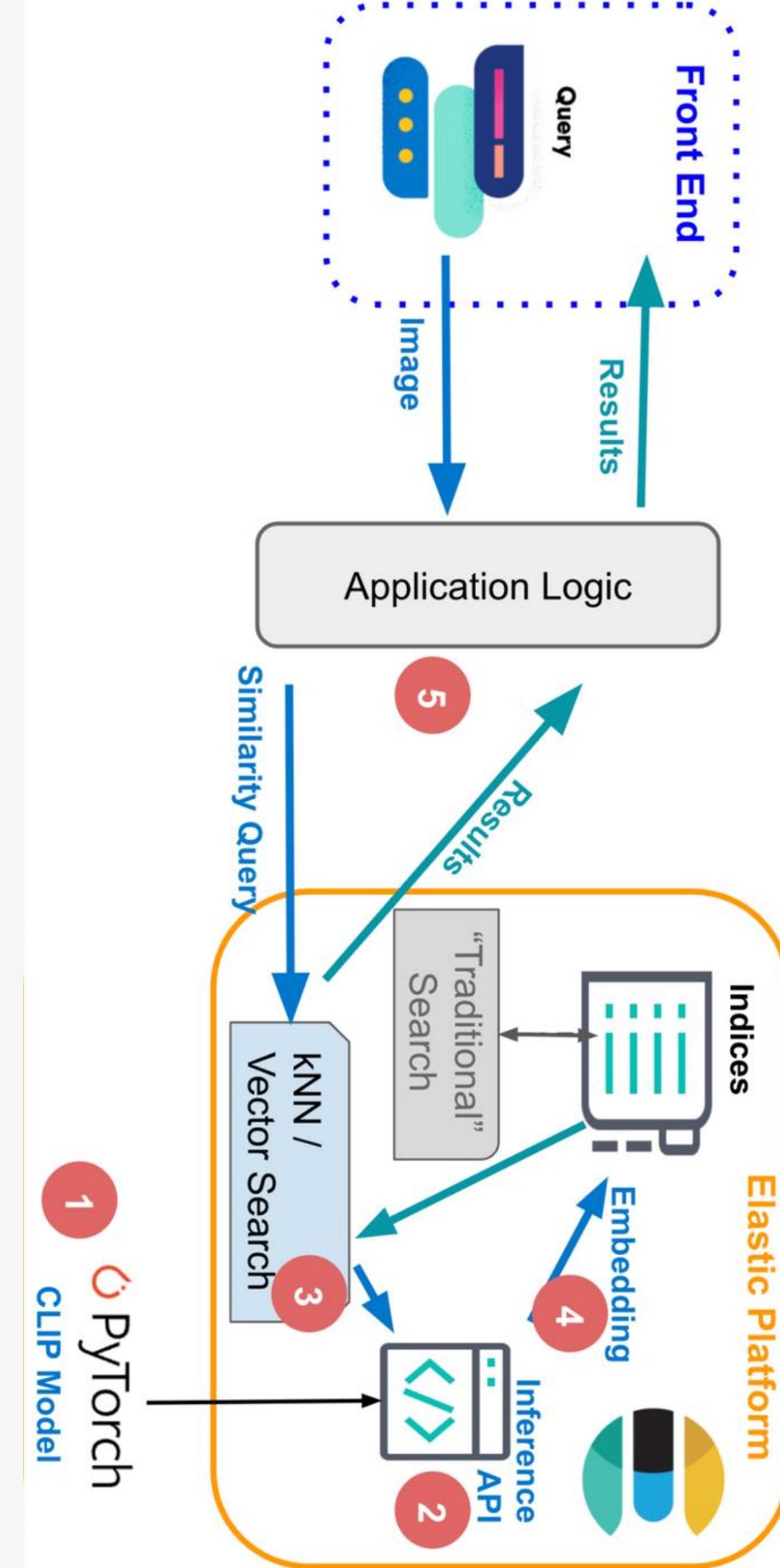
• • •

Dresses: • • •

• • •

User Interface & Advanced Features

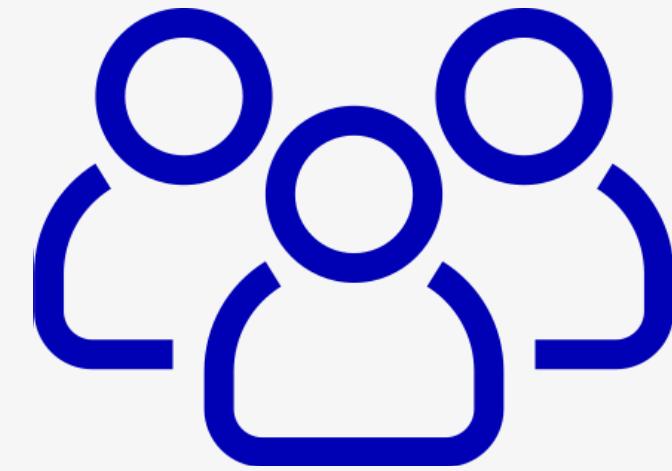
- Image upload-based visual search (no text dependency)
- Real-time similarity retrieval using embedding distance
- Interactive result carousel with ranked similarity scores
- REST-based backend–frontend communication
- Scalable architecture for large image datasets



Technologies & Tools Utilized

- **Deep Learning:** TensorFlow 2.13.0 with Keras API for neural network construction and training.
- **Computer Vision:** OpenCV 4.12.0 for image loading, preprocessing, and color conversion.
- **Numerical Computing:** NumPy 1.24.3 for efficient array operations and mathematical computations.
- **Machine Learning:** Scikit-learn 1.3.2 for distance metrics, similarity calculations, and t-SNE visualization.
- **Web Framework:** Flask 3.0.0 and Werkzeug 3.0.6 for RESTful API, web interface, and file handling.
- **Visualization:** Matplotlib 3.7.1 for training plots, loss curves, and embedding space visualization.

Team Members and Contributions



Yash Anokar

Team Lead - Deep Learning & Model Development

- Designed and implemented Triplet Network architecture
- Developed and optimized Triplet Loss function
- Trained CNN models with hyperparameter tuning
- Prepared datasets and built triplet generation pipeline
- Implemented feature extraction and embedding storage
- Performed t-SNE visualization and embedding space analysis

Ojas Kamlekar

Full Stack Developer - Web Application & Deployment

- Developed Flask-based web application
- Implemented real-time image similarity search
- Designed interactive UI/UX with sliding image carousel
- Built RESTful backend APIs
- Integrated database with optimized queries
- Managed deployment and server configuration

Conclusion & Impact

Our project showcases the strategic application of Triplet Networks for visual similarity search. By integrating advanced features like triplet loss optimization, multi-feature extraction, and real-time search, our system provides a versatile and high-performance solution.

E-commerce &

Fashion

Product recommendation, visual search, style matching, and outfit suggestions with high accuracy.

AI Image Search

Digital Asset Management

Efficient image organization and duplicate detection.

Content Platforms

Enhanced visual content discovery and recommendation.

Our domain-specific training with triplet loss achieves 15-20% higher accuracy than transfer learning, handling over 1000 images with sub-second query times, demonstrating scalability and efficiency for real-world applications.

Embedding Similarity Search method



Search Through Images



Reduce Text Based Searching



Visual Searching

