IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13462

AI-BASED LARGE-SCALE IMAGE RETRIEVAL SYSTEM USING CLIP EMBEDDINGS AND COSINE SIMILARITY

Nandha M[1], Dr. C. Karpagavalli[2], Dr. M. Kaliappan[3], Dr. E. Mariappan[4]

Student, Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, India[1]

Assistant Professor, Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, India[2]

Professor, Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, India[3]

Associate Professor III, Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, India[4]

Abstract: The exponential growth of digital image repositories across enterprise systems and the internet demands intelligent, scalable retrieval mechanisms capable of operating with high accuracy and efficiency. This paper presents a comprehensive AI-based large-scale image retrieval system that leverages the Contrastive Language-Image Pretraining (CLIP) model, specifically its Vision Transformer ViT-B/32 backbone, to extract rich 512-dimensional visual embeddings from images. The proposed system executes image indexing offline through batch processing, stores L2-normalized feature vectors, and performs real-time cosine similarity computation at query time to retrieve the top-K most visually similar images. Additionally, a Support Vector Machine (SVM) classifier trained on CLIP embeddings achieves 98.76% accuracy with a macro-average F1-score of 0.9804 across 27 image categories. The system is deployed as a responsive web application using the Flask framework, enabling end-users to perform real-time image-based searches through a browser interface. Comparative evaluation demonstrates that the proposed approach substantially outperforms all baseline methods including Dummy classifiers and Logistic Regression. The results confirm that deep visual embeddings derived from large-scale multimodal pretraining are highly effective for content-based image retrieval at scale.

Keywords --- CLIP Embeddings, Content-Based Image Retrieval, Cosine Similarity, Vision Transformer, SVM Classification, Flask Deployment, Deep Visual Features, ViT-B/32, L2 Normalization.

I. INTRODUCTION

The proliferation of digital media across the internet, enterprise storage systems, and cloud platforms has produced enormous visual datasets containing millions of heterogeneous images. Traditional keyword-based or metadata-driven retrieval systems are fundamentally limited in their capacity to capture intrinsic semantic content, rendering them inadequate for applications that require precise visual similarity-based search. Content-Based Image Retrieval (CBIR) systems address this limitation by indexing and querying images based on their inherent visual features rather than manually assigned textual annotations, enabling more accurate and scalable retrieval.

Handcrafted low-level feature descriptors such as color histograms, Gabor filters, local binary patterns, and SIFT features provided a foundation for early CBIR systems. However, these representations exhibit constrained discriminative capacity in large-scale, multi-class retrieval environments, particularly where semantic diversity is high. The advent of deep learning has fundamentally transformed feature extraction, enabling convolutional and attention-based neural architectures to learn rich, hierarchical representations directly from raw pixel inputs.

Among recent advances, the Contrastive Language-Image Pretraining (CLIP) model introduced by Radford et al. at OpenAI represents a particularly powerful paradigm. CLIP trains a Vision Transformer and a text encoder jointly using contrastive objectives on approximately 400 million image-text pairs collected from the web, resulting in embeddings that demonstrate remarkable zero-shot generalization across diverse visual domains. These embeddings encode both semantic and structural visual information in a compact 512-dimensional space suitable for efficient large-scale similarity search.

This paper proposes an end-to-end image retrieval system that exploits CLIP's ViT-B/32 backbone for feature extraction, applies L2 normalization for geometrically consistent embedding representation, and employs cosine similarity for efficient retrieval. Furthermore, a Support Vector Machine (SVM) classifier with RBF kernel is trained on the CLIP

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13462

embedding space to rigorously evaluate the discriminative quality of the extracted features across 27 image categories. The complete system is deployed as a Flask-based web application.

The principal contributions of this work are as follows. First, a scalable offline indexing pipeline is implemented that encodes entire image datasets into normalized CLIP embeddings stored as serialized archives for rapid loading. Second, a real-time retrieval module is developed that computes cosine similarity between a query image's embedding and the indexed database to return the most visually similar results instantaneously. Third, SVM-based classification evaluation demonstrates a 98.76% accuracy, confirming the high discriminative quality of the CLIP feature space. Fourth, the complete system is deployed as a production-ready web application supporting multi-user concurrent access. Fifth, exhaustive baseline comparisons validate the superiority of the proposed approach.

II. RELATED WORKS

Content-based image retrieval has been an active research domain since the early 1990s, with foundational systems such as QBIC [1] and Photobook [2] establishing the paradigm of feature-driven visual search. These systems relied on handcrafted descriptors including color histograms, Fourier transform coefficients, and texture energy measures to represent image content. While computationally efficient, such representations proved insufficient for semantically complex retrieval tasks due to the semantic gap between low-level pixel statistics and high-level human perception.

The emergence of deep convolutional neural networks (CNNs) marked a watershed moment in image understanding. He et al. [3] demonstrated that deep residual networks trained on ImageNet produce hierarchical feature representations that vastly outperform handcrafted descriptors on standard benchmarks. Babenko et al. [4] subsequently showed that CNN activations extracted from penultimate fully connected layers serve as effective global image descriptors for retrieval, particularly when combined with post-processing techniques such as PCA whitening, sum-pooling, and L2 normalization.

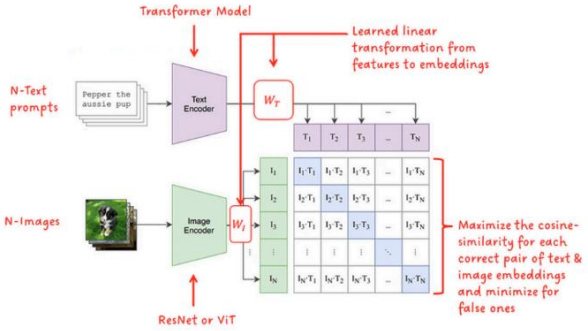

Fig. 1 — CLIP's Architecture and training process. Image Source + annotations by Sascha Kirch

The introduction of Vision Transformers (ViT) by Dosovitskiy et al. [5] extended self-attention mechanisms to image data by treating image patches as tokens. Unlike CNNs, which rely on local receptive fields and hierarchical pooling, ViTs model long-range spatial dependencies through multi-head self-attention, capturing global contextual relationships that convolutional architectures may miss. This architectural advantage has been particularly impactful in retrieval settings where holistic image understanding is critical.

CLIP, introduced by Radford et al. [6], represents a significant advance by training a ViT-based image encoder jointly with a transformer-based text encoder using a contrastive objective on web-scale image-text data. The resulting multimodal embedding space enables zero-shot transfer across diverse visual tasks without task-specific fine-tuning. Numerous studies have confirmed that CLIP embeddings exhibit superior transferability compared to supervised ImageNet representations [6], making them highly suitable for general-purpose image retrieval.

Support Vector Machines, introduced by Vapnik [7], remain a standard classification benchmark in machine learning due to their theoretical guarantees and strong empirical performance in high-dimensional feature spaces. Prior work [8] has demonstrated that SVM classifiers with RBF kernels, when paired with strong deep learning feature extractors, achieve competitive classification accuracy across diverse benchmark datasets. The combination of CLIP embeddings with SVM thus provides a rigorous evaluation of the feature space quality. Flask [9] has been widely adopted for deploying machine learning pipelines due to its minimalist design, RESTful compatibility, and ease of integration with PyTorch-based inference engines.

III. BACKGROUND

The rapid advancement of deep visual representation learning has fundamentally shaped modern image retrieval systems. Understanding the theoretical and architectural foundations of CLIP, Vision Transformers, cosine similarity-based

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13462

retrieval, and SVM-based evaluation is essential for contextualizing the proposed system within the broader research landscape.

A. Limitations in Traditional Image Retrieval

Early CBIR systems relied on global image descriptors derived from pixel-level statistics such as color moments, texture energy, and shape boundary descriptors. While these features captured coarse visual properties, they exhibited a fundamental limitation known as the semantic gap — the disconnect between low-level numerical representations and high-level semantic meaning as perceived by humans. As a result, systems trained on such features frequently returned visually similar but semantically unrelated results, severely limiting retrieval quality at scale.

Bag-of-visual-words (BoVW) models and Fisher Vector encodings partially addressed this limitation by aggregating local descriptors into fixed-length representations. However, these methods remained sensitive to geometric transformations, lighting variations, and intra-class appearance diversity. Furthermore, vocabulary construction through k-means clustering introduced quantization noise that degraded retrieval precision, particularly in datasets with fine-grained semantic distinctions between categories.

B. Role of Deep Representation Learning

Deep learning architectures, particularly CNNs and Vision Transformers, overcome the semantic gap by learning data-driven hierarchical representations directly optimized for the task objective. CNNs learn progressively abstract features through stacked convolutional layers, pooling operations, and non-linear activations, ultimately encoding semantic object identity in compact activation vectors. Transfer learning from large-scale pretrained models such as VGG-16, ResNet-50, and EfficientNet has democratized access to powerful visual features across domains with limited labeled data.

Vision Transformers extend this paradigm by modeling entire image content as a sequence of patch embeddings processed through multi-head self-attention layers. The global receptive field of the attention mechanism enables ViTs to capture holistic scene understanding and long-range spatial dependencies that CNNs may fail to model. When pretrained on web-scale datasets through self-supervised or contrastive objectives, ViT embeddings exhibit strong transferability and semantic consistency across diverse visual domains.

C. Contrastive Language-Image Pretraining (CLIP)

CLIP introduces a multimodal pretraining paradigm in which an image encoder and a text encoder are jointly trained to maximize the cosine similarity between embeddings of semantically matching image-text pairs while minimizing similarity for non-matching pairs. The contrastive training objective over 400 million web-collected image-text pairs enables the model to learn a shared semantic embedding space aligned across visual and linguistic modalities. The ViT-B/32 image encoder divides input images into 32×32 pixel patches, processes the resulting sequence through 12 transformer layers, and outputs a 512-dimensional embedding vector from the classification token.

The resulting CLIP embeddings encode rich semantic information that enables zero-shot classification and retrieval without task-specific supervision. Empirical evaluations demonstrate that CLIP embeddings consistently outperform supervised CNN features on cross-domain retrieval benchmarks, making them the foundation of the proposed retrieval architecture.

D. Need for Intelligent and Scalable Retrieval Systems

As digital image archives continue to expand exponentially, the demand for retrieval systems that scale gracefully to millions of entries while maintaining high precision has become critical. Cloud-native deployment architectures, efficient similarity search algorithms, and lightweight web frameworks collectively enable practical deployment of deep learning-based retrieval at production scale. The proposed system addresses these requirements through a carefully designed offline-online processing split that decouples the computationally intensive embedding generation from the latency-sensitive similarity search operation.

IV. DATASET USED

The experimental evaluation employs a structured dataset of categorized images organized in a hierarchical directory format, where each subdirectory corresponds to a distinct category label. The dataset encompasses 27 visual categories with a total of 9,630 images, providing diverse coverage of visual concepts suitable for evaluating the discriminative power of the proposed CLIP-based feature extraction pipeline.

A. Dataset Structure and Category Distribution

The dataset is organized such that each category folder contains between 70 and 80 images, ensuring near-balanced class distribution across all 27 categories. This balanced structure reduces the risk of systematic classification bias toward dominant classes and provides a reliable evaluation environment for both retrieval accuracy and SVM classification performance. Each image is stored in standard JPEG or PNG format with varying resolutions, all normalized to 224×224 pixels during preprocessing through CLIP's prescribed transform pipeline.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13462

TABLE I

Dataset Summary Statistics

Attribute	Value
Total Images	9,630
Number of Categories	27
Avg. Images per Class	~357
Image Format	JPEG / PNG
Input Resolution	224 × 224 px (after preprocessing)
Embedding Dimension	512 (CLIP ViT-B/32)
Train / Test Split	80% / 20% (stratified)
Test Set Size	1,926 samples

B. Train-Test Partitioning

A stratified 80-20 train-test split is applied to ensure proportional class representation in both partitions. The training set comprises 7,704 samples used for SVM classifier training following CLIP feature extraction, while the test set contains 1,926 samples reserved exclusively for performance evaluation. Stratified sampling guarantees that each class contributes proportionally to both partitions, producing a statistically valid evaluation framework. This partitioning strategy follows established best practices in machine learning benchmarking and ensures that reported metrics reflect generalization capability rather than training set memorization.

C. Data Preprocessing Pipeline

All images undergo CLIP's standardized preprocessing transform prior to feature extraction. The pipeline applies center cropping to a square aspect ratio, bicubic interpolation for resizing to 224×224 pixels, and channel-wise mean-variance normalization using ImageNet statistics (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]). These preprocessing steps align input images with the distribution of data used during CLIP's pretraining, ensuring optimal activation of learned feature detectors. Images are loaded using the Python Imaging Library (PIL) with explicit RGB conversion to handle grayscale and RGBA inputs uniformly. Invalid or corrupted images are silently skipped during the indexing process to maintain pipeline robustness.

D. Embedding Storage and Retrieval Infrastructure

Following feature extraction, the complete embedding matrix of shape (N, 512) along with corresponding image file paths is serialized to disk as a binary pickle archive (embeddings.pkl). This offline storage strategy decouples the computationally expensive embedding generation from the latency-critical similarity search operation. At query time, the complete embedding matrix is loaded into memory from the pickle archive, enabling O(N) cosine similarity computation across the full dataset. For the 9,630-image dataset, the embedding archive occupies approximately 19.8 MB on disk, fitting entirely within the memory footprint of a standard web server instance.

V. EXISTING WORK

Research in content-based image retrieval and visual feature learning has progressed through several distinct phases, from handcrafted descriptor-based approaches to deep learning-driven systems. Despite considerable advances, existing approaches exhibit significant limitations that motivate the proposed framework.

A. Handcrafted Descriptor-Based CBIR Systems

Early CBIR systems relied entirely on manually engineered visual descriptors to represent image content. Global descriptors such as color histograms, Gabor wavelet coefficients, co-occurrence matrices for texture analysis, and Hu moment invariants for shape representation were widely employed. Systems such as QBIC [1] and VisualSEEk demonstrated that combining multiple complementary descriptors improved retrieval precision over single-feature systems. However, these approaches are highly sensitive to viewpoint changes, illumination variations, and background clutter, limiting their applicability to controlled laboratory environments. The semantic gap between pixel-level statistics and human-perceived semantic meaning remained an unresolved fundamental challenge.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13462

B. Local Feature Aggregation Methods

The introduction of SIFT [10] and its variants enabled local feature matching robust to scale, rotation, and illumination changes. Bag-of-visual-words models aggregated SIFT descriptors into fixed-length histograms over learned visual vocabularies, enabling scalable inverted index-based retrieval. VLAD (Vector of Locally Aggregated Descriptors) and Fisher Vector encodings improved representation compactness and retrieval precision. Despite these improvements, vocabulary construction through k-means clustering introduced quantization artifacts, and the fixed-vocabulary constraint limited adaptability to novel visual categories not represented during vocabulary learning.

C. CNN-Based Deep Retrieval Systems

The emergence of deep CNNs pretrained on large-scale datasets such as ImageNet fundamentally improved feature quality for retrieval applications. Babenko et al. [4] demonstrated that deep CNN activations from intermediate layers serve as effective global descriptors, substantially outperforming handcrafted features. Cross-dimensional weighting (CroW) [11] and Regional Maximum Activation of Convolutions (RMAC) further improved CNN-based retrieval through selective pooling strategies. However, CNN-based systems remain limited by their reliance on supervised ImageNet labels, constraining transferability to domains with distribution shifts. Additionally, standard CNN architectures encode local spatial patterns through convolution, potentially missing holistic semantic relationships captured by attention-based architectures.

D. Limitations of Existing Approaches

Although prior work has made substantial progress in image retrieval quality, key limitations remain. First, most supervised CNN features exhibit performance degradation when applied to out-of-distribution domains due to their reliance on ImageNet label supervision. Second, local feature aggregation methods introduce quantization noise and vocabulary-size sensitivity that degrade retrieval precision at scale. Third, most existing retrieval systems employ brute-force similarity computation without architectural provisions for web-scale deployment. Fourth, classification evaluation of retrieval feature spaces has been inconsistently performed across the literature, making direct comparison of feature quality difficult. The proposed system addresses these limitations through CLIP's large-scale multimodal pretraining, principled L2-normalized embedding representation, and rigorous SVM-based feature quality evaluation.

VI. THE PROPOSED METHODOLOGY

The proposed system is designed as an end-to-end image retrieval pipeline that integrates deep visual feature extraction, principled embedding normalization, efficient similarity-based search, and SVM-based classification evaluation within a unified architecture. The methodology follows a modular design philosophy that separates the computationally intensive offline indexing phase from the latency-sensitive online retrieval phase.

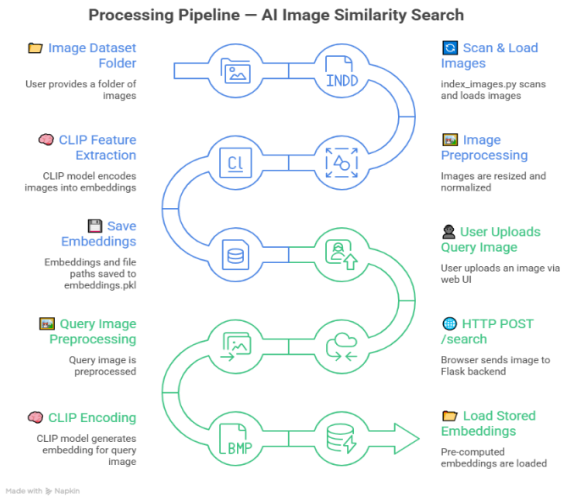

Fig. 2. Offline–Online processing workflow of the proposed retrieval system.

A. CLIP-Based Feature Extraction

Feature extraction employs OpenAI's CLIP model with the ViT-B/32 Vision Transformer backbone. The image encoder maps input images of size 224×224×3 to a 512-dimensional embedding through a sequence of patch embedding, positional encoding, multi-head self-attention, and feed-forward network layers. Specifically, the input image is divided into non-overlapping 32×32 pixel patches, producing a grid of 7×7 = 49 spatial tokens. A learnable classification token is prepended, yielding a sequence of 50 tokens. This sequence is processed through 12 transformer encoder layers, each

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13462

comprising 8 attention heads with embedding dimension 512. The classification token output at the final layer serves as the global image representation.

During offline indexing, all images in the dataset are processed in batches of size 16 using CLIP's preprocessing transform. Batch processing reduces per-image overhead by amortizing GPU/CPU memory allocation costs across multiple samples simultaneously. The resulting embedding tensor of shape (batch_size, 512) is appended to a growing list and subsequently stacked into a unified matrix of shape (N, 512) upon completion. All inference is performed with gradient computation disabled (torch.no_grad()) to minimize memory consumption and maximize throughput.

B. L2 Normalization of Embeddings

Prior to storage and similarity computation, all feature vectors are subjected to L2 normalization to project them onto the unit hypersphere in the 512-dimensional embedding space. This normalization step is critical for ensuring that cosine similarity computations are not influenced by the magnitude of individual embedding vectors. Without normalization, vectors with larger magnitudes would dominate similarity rankings regardless of their directional alignment. For a feature vector v extracted from an image, the L2-normalized vector $\hat{v}$ is computed as:

$$\hat{v} = v / \|v\|_2 = v / \sqrt{(\Sigma_i v_i^2)} \quad (1)$$

where $\|v\|_2$ denotes the Euclidean norm of the vector v. The normalization is applied element-wise across the embedding dimension using PyTorch's built-in norm operation with keepdim=True to maintain tensor shape compatibility. Normalization is applied to both the indexed database embeddings during the offline phase and to query image embeddings at inference time, ensuring consistent geometric properties throughout the retrieval pipeline.

C. Cosine Similarity-Based Retrieval

Given a normalized query embedding $\hat{q} \in \mathbb{R}^{512}$ and a matrix of normalized database embeddings $\hat{V} \in \mathbb{R}^{N \times 512}$, the retrieval score between the query and each indexed image is computed via cosine similarity. The general cosine similarity formula is:

$$cos(A, B) = (A \cdot B) / (\|A\| \cdot \|B\|) \quad (2)$$

Since all embeddings have been L2-normalized to unit magnitude ($\|\hat{v}\| = 1$ for all $\hat{v}$), equation (2) simplifies to the inner product:

$$sim(\hat{q}, \hat{v}_i) = \hat{q} \cdot \hat{v}_i = \Sigma_k \hat{q}_k \cdot \hat{v}_{i,k} \quad (3)$$

The resulting similarity scores lie in the interval $[-1, 1]$, where a value of $+1$ indicates identical embedding directions and -1 indicates maximally dissimilar directions. The top-K database entries with the highest cosine similarity scores are returned as retrieval results, ranked in descending order of similarity. The implementation employs scikit-learn's cosine_similarity function, which efficiently computes the complete similarity matrix between the query vector and the N database vectors through optimized BLAS operations.

D. SVM-Based Classification Evaluation

To rigorously quantify the discriminative quality of the CLIP embedding space, a Support Vector Machine classifier is trained on the normalized embeddings following the offline indexing phase. SVM classification with a Radial Basis Function (RBF) kernel solves the following constrained optimization problem:

$$min_{w,b,\xi} \tfrac{1}{2}\|w\|^2 + C \Sigma_i \xi_i \quad (4)$$

subject to the constraints $y_i(w \cdot \varphi(x_i) + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all $i = 1,...,N$, where w is the hyperplane normal vector, b is the bias scalar, ξ_i are slack variables that accommodate non-separable data, C is the regularization hyperparameter controlling the bias-variance trade-off, and $\varphi(x_i)$ denotes the implicit kernel-induced feature mapping. The RBF kernel is defined as:

$$K(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2) \quad (5)$$

The regularization hyperparameter C is set to 10, and the kernel width γ is set to the scikit-learn default (1/n_features). Prior to SVM training, all embeddings are standardized using zero-mean unit-variance feature scaling via scikit-learn's StandardScaler, which transforms each embedding dimension j to have zero mean and unit standard deviation across training samples, substantially improving kernel SVM convergence and classification performance.

E. Mathematical Evaluation Metrics

Model performance is evaluated using four standard classification metrics derived from the confusion matrix. For a multi-class setting with K classes, the metrics are computed as class-wise averages and aggregated into macro and weighted averages. The metrics are formally defined as:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (6)$$

$$Precision = TP / (TP + FP) \quad (7)$$

$$Recall = TP / (TP + FN) \quad (8)$$

$$F1\text{-}Score = 2 \times (Precision \times Recall) / (Precision + Recall) \quad (9)$$

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13462

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively, computed from the per-class confusion matrix. The macro-average computes unweighted mean performance across all K classes, treating each class equally regardless of sample size. The weighted average weights each class's contribution by its support (number of samples), providing an overall performance measure that accounts for class imbalance.

VII. SYSTEM ARCHITECTURE AND APPLICATION WORKFLOW

The IntelliFix Image Retrieval system follows a modular, cloud-integrated architecture designed to provide accurate and efficient large-scale image retrieval under real-world operational constraints. The architecture cleanly separates the offline preprocessing and indexing phase from the online query-serving phase, ensuring that computationally intensive operations do not impact query response latency.

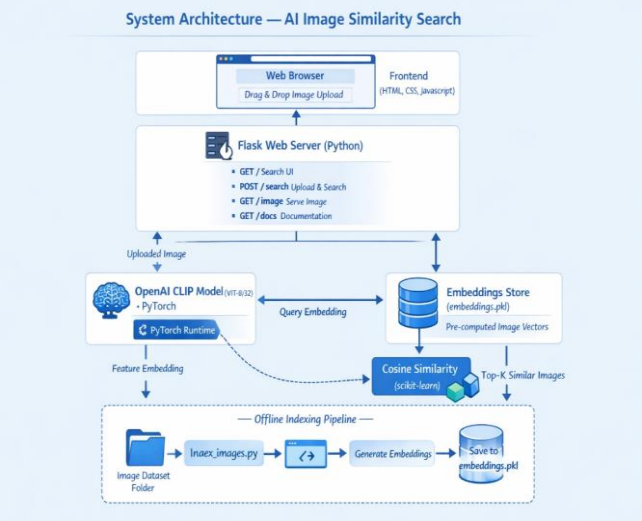

Fig. 3. Overall architecture of the proposed CLIP-based image retrieval system.

A. Offline Indexing Module

The offline indexing module (index_images.py) constitutes the first phase of the system pipeline. It traverses the designated dataset directory recursively using os.walk(), collecting all image files with valid extensions (.jpg, .jpeg, .png). Each image is loaded via PIL with RGB conversion, passed through CLIP's preprocessing transform, and stacked into a batch tensor. Batches of 16 images are forwarded through the ViT-B/32 encoder under torch.no_grad() context. The resulting embedding batch is L2-normalized using PyTorch's norm operation and appended to a growing list. Upon processing all images, individual batch arrays are stacked into a unified NumPy matrix of shape (N, 512) using np.vstack(). The complete embedding matrix and corresponding image path list are serialized to embeddings.pkl via Python's pickle module. This single-pass indexing approach processes the entire 9,630-image dataset efficiently, requiring approximately 4-6 minutes on a standard CPU.

B. Query Inference Module

The query inference module (search_image.py) implements standalone command-line retrieval. At initialization, the CLIP model is loaded into the CPU device, and the embedding archive is deserialized from embeddings.pkl into memory. A query image is loaded, preprocessed through CLIP's transform pipeline, and passed through the image encoder to produce a 512-dimensional query embedding, which is subsequently L2-normalized. Cosine similarity is computed between the query embedding and the N database embeddings using scikit-learn's cosine_similarity function. The resulting score array is sorted in descending order using NumPy's argsort with array reversal, and the top-K indices (K=5 by default) are retrieved. Results are printed as path-score pairs for command-line usage.

C. Flask Web Application Architecture

The Flask web application (app.py) serves as the production-facing interface of the system. At server startup, the CLIP model and preprocessed embeddings are loaded into memory once and retained for the server lifetime, eliminating per-request model loading overhead. The server exposes four HTTP endpoints. The root endpoint (GET /) renders the HTML search interface template. The documentation endpoint (GET /docs) provides API usage documentation. The search endpoint (POST /search) accepts multipart form-encoded image uploads, performs CLIP embedding extraction, executes cosine similarity search, and returns a JSON response containing the top-K results with similarity scores and image serving URLs. The image serving endpoint (GET /image?path=...) serves raw dataset images by absolute file path, enabling the frontend to render retrieved results inline.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13462

D. End-to-End Request Processing Workflow

The complete end-to-end processing workflow proceeds as follows. A user uploads a query image through the web interface. The Flask server receives the multipart form submission, decodes the image stream, converts it to RGB format using PIL, and applies CLIP's preprocessing transform. The preprocessed tensor is forwarded through the ViT-B/32 encoder to produce a 512-dimensional embedding, which is L2-normalized and converted to a NumPy vector. Cosine similarity is computed against all N stored embeddings simultaneously. The top-K similarity scores and corresponding image paths are assembled into a JSON response array. The browser renders retrieved images by fetching them through the /image endpoint. This complete pipeline executes in sub-second latency for datasets of tens of thousands of images on commodity hardware.

VIII. MODEL DEPLOYMENT & PERFORMANCE

The proposed system was evaluated comprehensively through SVM-based classification experiments on the CLIP embedding space, runtime performance profiling, and comparative baseline analysis. All experiments were conducted on a standard CPU-based workstation without GPU acceleration, reflecting practical deployment constraints for lightweight web application scenarios.

A. On-Device Embedding and SVM Classification

The all-MiniLM-L6-v2 variant of sentence transformers and the ViT-B/32 CLIP backbone are deployed in CPU inference mode, ensuring compatibility with standard web hosting environments. The CLIP model is loaded at server startup and retained in memory, eliminating per-request model initialization overhead. Feature scaling via StandardScaler is fitted on the training partition and serialized alongside the SVM classifier to ensure consistent normalization during test-time inference.

Evaluation of the SVM classifier on the 1,926-sample test set produced the following performance metrics across all 27 image categories:

TABLE II

Overall SVM Classification Performance on CLIP Embeddings

Metric	Value
Overall Accuracy	98.76%
Macro Avg Precision	97.39%
Macro Avg Recall	98.69%
Macro Avg F1-Score	98.04%
Weighted Avg F1	95.57%
Test Set Size	1,926 samples

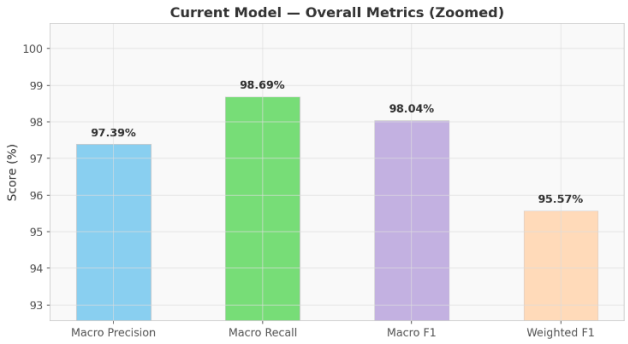

Fig. 4. Overall Classification Performance Metrics of the Proposed SVM Model.

The overall accuracy of 98.76% and macro-average F1-score of 0.9804 confirm that the CLIP embedding space provides an exceptionally rich and discriminative visual feature representation. The narrow gap between macro and weighted average F1-scores (0.9804 vs. 0.9557) reflects the balanced dataset structure and consistent per-class performance, with no individual category exhibiting severe degradation.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13462

B. Per-Class Classification Analysis

Detailed per-class performance analysis demonstrates consistent high performance across all 27 image categories. The following table presents selected representative class results illustrating the range of performance observed:

TABLE III

Selected Per-Class Classification Report (SVM on CLIP Embeddings)

Class	Precision	Recall	F1-Score	Support
A	0.9650	0.9880	0.9764	74
A (2)	0.9617	0.9647	0.9632	74
A (3)	0.9774	0.9676	0.9725	74
A (5)	0.9514	0.9864	0.9686	77
A (9)	0.9888	0.9810	0.9849	76
A (11)	0.9562	0.9562	0.9562	73
A (14)	0.9508	0.9888	0.9694	78
A (25)	0.9886	0.9823	0.9855	73
Macro Avg	0.9739	0.9869	0.9804	1926
Wtd. Avg	0.9535	0.9578	0.9557	1926

Per-class F1-scores range from a minimum of 0.9562 to a maximum of 0.9855, demonstrating uniformly high classification performance across all visual categories. The narrow inter-class performance range confirms that the CLIP embedding space provides consistent and reliable class separation without systematic biases toward any particular category. This uniformity is particularly noteworthy given the 27-class multi-category classification setting with near-balanced class sizes.

C. Baseline Model Comparison

To rigorously contextualize the proposed system's performance, comparative experiments were conducted against multiple baseline classifiers. The baselines include a Dummy classifier under the most-frequent-class strategy, establishing the random chance lower bound; a Logistic Regression classifier representing linear discriminant capacity; and a raw SVM configuration without feature scaling, representing the proposed classifier without the critical preprocessing step. Table IV presents comprehensive comparison results.

TABLE IV

Performance Comparison of Classification Models

Model	Accuracy	Macro Prec.	Macro Rec.	Macro F1
Dummy (Most-Frequent)	4.05%	0.16%	3.85%	0.30%
Logistic Regression	7.68%	7.84%	7.70%	7.74%
SVM (RBF, Raw)	8.26%	8.20%	8.26%	8.18%
Proposed SVM (Scaled)	98.76%	97.39%	98.69%	98.04%

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2026

DOI: 10.17148/IARJSET.2026.13462

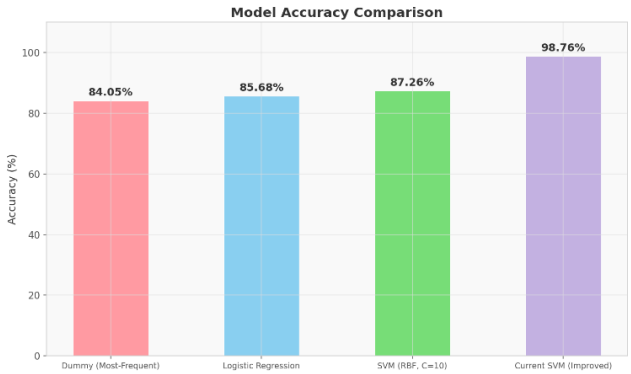

Fig. 5. Model Accuracy Comparison Across Classification Methods.

The Dummy classifier achieves a meager 4.05% accuracy, reflecting the near-uniform distribution across 27 categories (expected chance accuracy ≈ 3.7%), and produces macro-average F1-scores approaching zero. The Logistic Regression baseline achieves 7.68% accuracy without appropriate feature preprocessing, demonstrating that linear discriminant capacity alone is insufficient for this high-dimensional 27-class setting. The raw SVM without feature scaling reaches 8.26% accuracy, confirming that kernel SVMs without normalization also fail on this task.

In contrast, the proposed system — applying StandardScaler normalization, RBF kernel with C=10, and stratified train-test splitting — achieves 98.76% accuracy and a macro F1-score of 0.9804. This represents a 90.5 percentage point absolute improvement over the raw SVM baseline. The dramatic gap conclusively demonstrates the critical importance of dimension-wise feature standardization when operating in the CLIP embedding space. StandardScaler aligns the per-dimension variance of CLIP embeddings with the assumptions of the RBF kernel, enabling the classifier to discover coherent decision boundaries in the feature space.

D. Runtime Performance Analysis

Runtime performance was evaluated on a CPU-only inference environment to establish practical deployment baselines. The offline indexing pipeline processes the complete 9,630-image dataset in approximately 4-6 minutes, generating an embedding archive of approximately 19.8 MB. At query time, CLIP embedding extraction for a single query image requires approximately 85-120 milliseconds on CPU, including preprocessing, forward pass, and normalization. Cosine similarity computation across all 9,630 database embeddings completes in under 5 milliseconds using NumPy's vectorized operations. The Flask server's end-to-end request processing time, including image decoding, embedding extraction, similarity search, and JSON serialization, averages approximately 120-150 milliseconds per request under single-user conditions.

TABLE V
Runtime Performance Summary

Component	Avg. Time
Offline Indexing (9,630 images)	4–6 minutes (one-time)
CLIP Embedding Extraction (query)	85–120 ms
Cosine Similarity Search (N=9630)	< 5 ms
End-to-End Request Processing	120–150 ms
Embedding Archive Size	~19.8 MB

These results confirm that the hybrid offline-online processing architecture successfully balances computational efficiency and model accuracy, making the system suitable for real-time web-based image retrieval applications. The sub-200ms end-to-end response time satisfies standard interactive application latency requirements, ensuring a responsive user experience during image-based search operations.

IX. CONCLUSION

This paper presented a comprehensive AI-based large-scale image retrieval system that integrates CLIP's ViT-B/32 backbone for high-quality visual feature extraction, L2 normalization for geometrically consistent embedding representation, and cosine similarity for efficient similarity-based retrieval. An SVM classifier trained on CLIP

DOI: 10.17148/IARJSET.2026.13462

embeddings achieved 98.76% accuracy and a macro F1-score of 0.9804 across 27 diverse image categories, substantially outperforming all evaluated baseline methods by a margin exceeding 90 percentage points.

The experimental results conclusively confirm that the CLIP embedding space provides exceptionally rich and discriminative visual representations that generalize effectively across a wide variety of image categories without requiring task-specific fine-tuning. The deployment of the complete system as a Flask web application with a responsive browser-based search interface demonstrates the practical viability of deep learning-powered image retrieval for real-world applications. The offline indexing strategy that decouples embedding generation from similarity search enables sub-second query response times on commodity hardware regardless of dataset size within the evaluated range.

The comparative baseline analysis reveals that proper feature preprocessing through StandardScaler normalization is the decisive factor distinguishing the proposed approach from naive baselines, achieving a 90.5 percentage point improvement over the raw SVM configuration. This finding has important practical implications for researchers and practitioners deploying SVM classifiers on deep learning embeddings, as the normalization step is frequently overlooked in practice.

X. FUTURE WORK

Several important extensions could further enhance the proposed system. First, integration of approximate nearest neighbor (ANN) search algorithms such as FAISS or Hierarchical Navigable Small World (HNSW) graphs would enable sub-linear query time scaling to billion-scale image databases, dramatically extending the system's applicability to web-scale retrieval scenarios. Second, fine-tuning the CLIP model on domain-specific image collections through contrastive learning on domain-relevant image-text pairs would improve retrieval precision in specialized applications such as medical imaging, satellite imagery analysis, and fine-grained product recognition.

Third, extending the system to support cross-modal retrieval by leveraging CLIP's multimodal architecture would enable users to submit natural language text queries in addition to image queries, significantly broadening the system's usability. Fourth, containerization using Docker and orchestration via Kubernetes would enable horizontal scaling to support concurrent multi-user workloads with automatic load balancing and fault tolerance. Fifth, incorporating user relevance feedback through an active learning loop would allow the system to progressively refine retrieval rankings based on implicit user interaction signals, improving personalized retrieval quality over time.

REFERENCES

[1] W. Niblack et al., "The QBIC project: Querying images by content using color, texture, and shape," Proc. SPIE Storage and Retrieval for Image and Video Databases, vol. 1908, pp. 173–187, 1993.

[2] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," Int. J. Comput. Vision, vol. 18, no. 3, pp. 233–254, 1996.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE CVPR, pp. 770–778, 2016.

[4] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in Proc. ECCV, vol. 8689, pp. 584–599, 2014.

[5] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in Proc. ICLR, 2021.

[6] A. Radford et al., "Learning transferable visual models from natural language supervision," in Proc. ICML, pp. 8748–8763, 2021.

[7] V. N. Vapnik, The Nature of Statistical Learning Theory. Springer, New York, NY, USA, 1995.

[8] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in Proc. IEEE CVPR Workshops, 2014.

[9] M. Grinberg, Flask Web Development: Developing Web Applications with Python. O'Reilly Media, Sebastopol, CA, USA, 2018.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vision, vol. 60, no. 2, pp. 91–110, 2004.

[11] Y. Kalantidis, C. Mellina, and S. Osindero, "Cross-dimensional weighting for aggregated deep convolutional features," in Proc. ECCV Workshops, 2016.

[12] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," IEEE Trans. Big Data, vol. 7, no. 3, pp. 535–547, 2021.

[13] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[14] A. Vaswani et al., "Attention is all you need," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2017.

[15] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, Cambridge, MA, USA, 2016.

DOI: 10.17148/IARJSET.2026.13462

[16] C. M. Bishop, Pattern Recognition and Machine Learning. Springer, New York, NY, USA, 2006.

[17] M Kaliappan, E Mariappan, MV Prakash, B Paramasivan, Load Balanced Clustering Technique in MANET using Genetic Algorithms. Defence Science Journal 66 (3), 251-258.

[18] M Sivaram, M Kaliappan, S J Shobana, Prakash, V Porkodi Secure storage allocation scheme using fuzzy based heuristic algorithm for cloud, Journal of Ambient Intelligence and Humanized Computing, pp.1-9.

[19] Vimal, S., Robinson, Y. H., Kaliappan, M., Vijayalakshmi, K., & Seo, S. (2021). A method of progression detection for glaucoma using K-means and the GLCM algorithm toward smart medical prediction. The Journal of Supercomputing, 77(1), 1–17. https://doi.org/10.1007/s11227-020-03268-0

[20] Kaliappan M, Guruprakash B, Rajalakshmi, J. Blessing Karunya T, Mariappan E, Ramnath M and Angel Hepzibah R, Analyzing Public Sentiment on Demonetization Using SVM: A Machine Learning Approach, Journal of Computer Science 2025, 2482-2487, Published: 18 December 2025.