IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Content-Based Image Retrieval System Using Machine Learning

Vinnakota Krishna Indu, Maddala Charmila, Akkanaboina Swaroop, Chella Lakshmi Sucharitha

B. Tech Students, Department of CSE (Artificial Intelligence and Machine Learning), Dadi Institute of Engineering and Technology, NH-16, Anakapalle, Visakhapatnam-531002, A.P

Mr. A. Venkateswara Rao -Department of CSE (Artificial Intelligence-Machine Learning) Diet, Anakapalli, India.

Assistant Professor: Mrs.P. Swapna, Department of CSE (Artificial Intelligence-Machine Learning) Diet, Anakapalli, India.

Abstract

The exponential growth of digital image data necessitates advanced systems for efficient retrieval based on visual content. This project proposes a Content-Based Image Retrieval (CBIR) system that leverages machine learning techniques to bridge the gap between low-level image features and high-level user intent. The system extracts color histograms and deep learning-based features using pre-trained models like ResNet50 and VGG16 , combining them into hybrid feature vectors for enhanced representation. To ensure scalability, FAISS (Facebook AI Similarity Search) is employed for indexing and rapid similarity comparisons using metrics like cosine similarity and Euclidean distance . Machine learning classifiers such as SVM (80% accuracy) and KNN (60% accuracy) are trained on color histogram features, while a fine-tuned ResNet50 model (97.5% accuracy) enhances deep feature extraction. The system processes user queries in real time, delivering results through an intuitive interface and addressing challenges like computational efficiency and semantic understanding. Evaluated on datasets like Corel 1K, Vistex, and Faces, the system demonstrates superior performance in retrieving visually similar images, with applications in e-commerce, medical imaging, and forensic analysis. By integrating traditional and modern techniques, this CBIR system sets a benchmark for accuracy, scalability, and user-centric design in image retrieval.

Keywords— Content-Based Image Retrieval (CBIR), ResNet50, FAISS, SVM, KNN, color histograms, feature extraction, cosine similarity, Euclidean distance, hybrid features, real-time retrieval.

Introduction

In the digital age, the exponential growth of image data has created a pressing need for efficient and accurate image retrieval systems. Traditional methods of managing large image databases, which rely heavily on manual metadata tagging, are no longer sufficient to meet the demands of modern applications. This project introduces a Content-Based Image Retrieval (CBIR) System Using Machine Learning , designed to address these challenges by enabling searches based on visual content rather than textual metadata. By leveraging advanced machine learning techniques, this system extracts and analyzes features such as color, texture, shape, and spatial relationships, providing users with precise and relevant image retrieval results. The integration of deep learning models like Convolutional Neural Networks (CNNs) ensures that the system can handle complex image datasets with high accuracy and scalability.

Dataset:

The dataset used in this Content-Based Image Retrieval (CBIR) project consists of 1,000 images organized into 8 distinct categories :

Dinosaurs
Flowers
Beaches
Buses
Horses
Mountain and Snow
Elephants
Food

The dataset is split into training and testing sets using an 80-20 ratio :

Training Set : 800 images (100 images per category).
Testing Set : 200 images (25 images per category)..

Dataset Description:

The dataset is designed to evaluate the CBIR system's ability to retrieve visually similar images based on color, texture, shape, and deep learning features . Key details include:

Image Preprocessing :

Resizing : All images are resized to 224x224 pixels to align with the input requirements of CNN models like ResNet50 and VGG16 .

Normalization : Pixel values are scaled to the range [0, 1] to ensure consistency and improve model convergence.

Data Augmentation : Techniques like rotation , flipping , and brightness adjustments are applied to the training set to enhance diversity and prevent overfitting.

Feature Extraction :

Color Histograms : Extracted in the HSV color space with 8x8x8 bins to capture global color distribution.

CNN Embeddings : Features are extracted from pre-trained models (ResNet50 , VGG16) to capture high-level semantic patterns.

Benchmark Datasets :

Public datasets like Corel 1K , CIFAR-10 , and PlantVillage are referenced in the literature review for comparative analysis.

These datasets are used to validate the system's scalability and performance (e.g., SVM achieved 80% accuracy on Corel 1K).

Label Encoding :

Class labels are encoded using LabelEncoder to ensure compatibility with machine learning models like SVM and KNN .

Applications :

The dataset's diversity supports real-world applications in e-commerce , medical imaging , and forensic analysis .

This structured dataset ensures the CBIR system can handle both small-scale and large-scale image retrieval tasks efficiently.

Methodology

Motivation:

The methodology of your Content-Based Image Retrieval (CBIR) system is driven by the need to address critical limitations in traditional image retrieval systems and leverage modern machine learning (ML) techniques for improved accuracy, scalability, and real-time performance. Here's a breakdown of the motivation behind each component of your methodology:

1. Feature Extraction with Hybrid Techniques
Traditional CBIR systems rely on manual or low-level features (e.g., color histograms, texture descriptors), which often fail to capture the semantic meaning of images. By integrating deep learning models (ResNet50, VGG16) with traditional methods, your system bridges the semantic gap. For example:

CNNs (ResNet50) automatically extract high-level features (e.g., object shapes, patterns) that align with human perception.
Color histograms and texture descriptors ensure compatibility with simpler datasets and provide baseline performance.
This hybrid approach balances computational efficiency with descriptive power, ensuring robustness across diverse image types.
2. Similarity Metrics (Cosine Similarity, Euclidean Distance)
Traditional systems often struggle with accurate similarity measurement due to reliance on rigid metrics. Your methodology uses cosine similarity and Euclidean distance to:

Handle high-dimensional feature vectors (e.g., CNN embeddings) effectively.
Provide flexibility in matching images based on orientation (cosine similarity) or absolute differences (Euclidean distance).
These metrics are chosen for their proven effectiveness in real-world applications like medical imaging and e-commerce.
3. Clustering (K-Means, Hierarchical Clustering)
Large datasets demand efficient organization to reduce retrieval time. K-Means clustering groups images into clusters based on feature similarity, narrowing the search space. Hierarchical clustering adds multi-level organization for nuanced retrieval. These techniques:

Improve scalability by avoiding exhaustive pairwise comparisons.
Enhance user experience by enabling faster results for large databases.
4. FAISS for Indexing
Storing and retrieving millions of high-dimensional feature vectors is computationally intensive. FAISS (Facebook AI Similarity Search) is chosen for its ability to:

Perform approximate nearest neighbor (ANN) searches efficiently.
Handle large-scale datasets with minimal latency.
This ensures real-time performance, even as the database grows.
5. Data Augmentation and Preprocessing
Real-world images vary in resolution, lighting, and orientation. Techniques like resizing to 224x224 pixels , normalization , and data augmentation (rotation, flipping) ensure:

Consistency in input data for CNN models.
Robustness against overfitting and variations in image quality.
6. Model Training (SVM, KNN, Fine-Tuned ResNet50)

SVM/KNN : Used for baseline classification with color histograms to validate simpler workflows.
ResNet50 : Fine-tuned to capture domain-specific features (e.g., medical or agricultural images).
This combination ensures the system works for both small-scale and specialized use cases.

Literature Review:

1. Traditional Approaches:

Early CBIR systems relied on handcrafted features like SIFT, SURF, and HOG for image retrieval. These methods effectively captured specific aspects of image content but struggled with generalizing to large and diverse datasets.

2. Machine Learning Integration:

Supervised learning techniques like Support Vector Machines (SVMs) and k-Nearest Neighbors (k-NN) were introduced to classify and retrieve images based on extracted features. However, their reliance on feature engineering limited their adaptability to complex datasets.

3. Deep Learning Approaches:

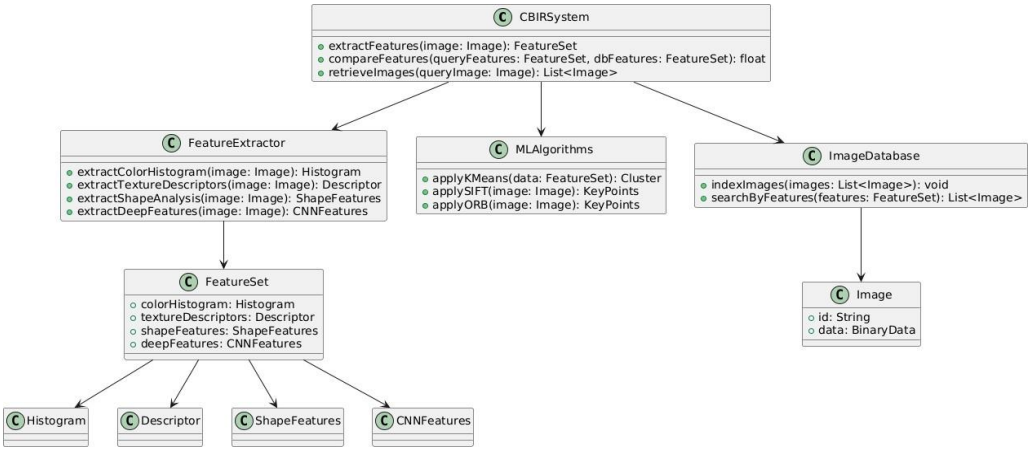
Recent advancements in Convolutional Neural Networks (CNNs) have transformed CBIR systems by automating feature extraction and learning high-level representations. Pre-trained models like ResNet, VGG, and Inception have demonstrated significant improvements in retrieval tasks. Integration of clustering techniques like K-means has further optimized retrieval by organizing features into meaningful groups.

Algorithms and implementation

Algorithms Used:

ResNet50 (Deep Learning Model):
Algorithm: ResNet50 is a convolutional neural network (CNN) architecture with residual connections that allow for deeper networks without vanishing gradients.
Purpose: It extracts high-level semantic features from images by leveraging its pre-trained weights on ImageNet.
Fine-Tuning:
Algorithm: Transfer learning is applied by fine-tuning the pre-trained ResNet50 model on the specific dataset.
Steps:

1. Freeze all layers initially to retain the pre-trained weights.
2. Add custom dense layers for classification.
3. Unfreeze the last few layers and fine-tune with reduced learning rate.



FAISS (Similarity Search):
Algorithm: FAISS uses an efficient indexing structure (e.g., `IndexFlatL2`) to perform fast nearest-neighbor searches in high-dimensional spaces.
Steps:
1. Normalize the feature vectors using L2 normalization.
2. Create an index using the training features.
3. Perform a search for the query image by computing the L2 distance to the indexed features. Siamese Network:
Algorithm: Siamese networks are used for learning similarity metrics between pairs of inputs. Steps:
1. Two input images are passed through a shared base network to extract their feature representations.
2. The absolute difference between the feature vectors is computed.
3. A binary classifier (sigmoid output) predicts whether the two images are similar or not.

Workflow of the CBIR System:

1. Training Phase:

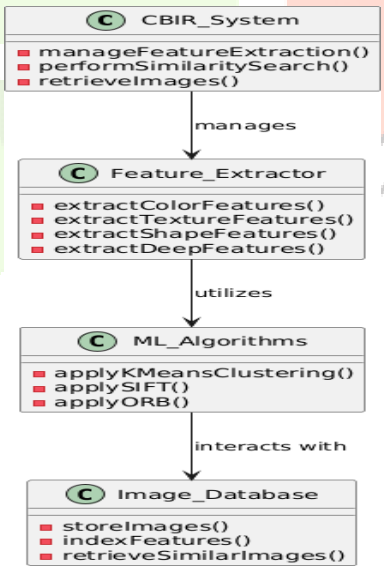
Images are preprocessed (resizing, normalization) and split into training data. Color histograms (RGB/HSV) capture color distribution, while ResNet50 extracts deep semantic features via transfer learning. The model is fine-tuned on the dataset to optimize classification accuracy. Combined color and ResNet50 features form hybrid embeddings, indexed using FAISS for efficient nearest-neighbor search.

2. Query Phase:

A query image undergoes identical preprocessing and feature extraction. Normalized features are compared against the FAISS index using metrics like cosine similarity, retrieving top-k visually similar images.

3. Siamese Network Training:

Pairs of images (similar: same class; dissimilar: different classes) are input to twin networks sharing weights. A contrastive or triplet loss function trains the network to minimize distance between similar pairs and maximize it for dissimilar ones, refining the embedding space for better similarity judgments. This enhances retrieval robustness beyond handcrafted/color features.



Implementation Details:

Data Loading and Preprocessing:
Dataset Structure: The dataset is organized into training and testing directories (`train path` and `test path`) with subdirectories for each class/category.
Image Loading: Images are loaded using OpenCV (`cv2.imread`) and resized to a fixed size of `(224, 224)` pixels to ensure uniformity for input into the ResNet50 model.
Label Encoding: The categorical labels (e.g., class names) are encoded into numerical format using `Label Encoder`.

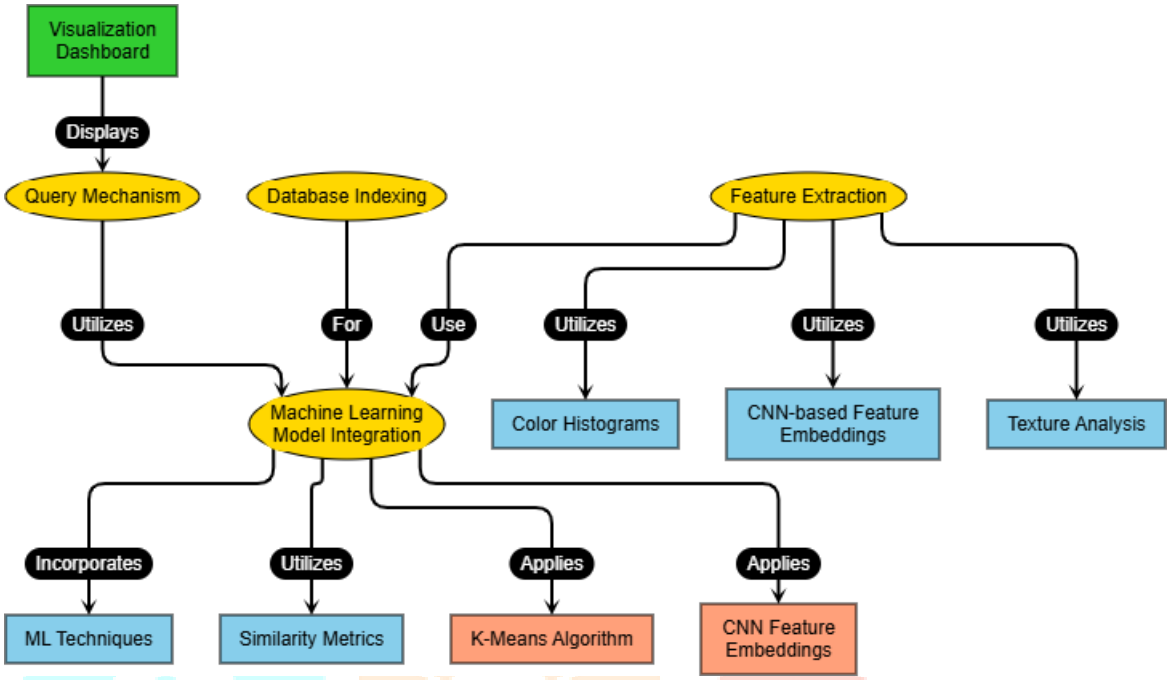
Feature Extraction:
Color Features: Histogram-based color features are extracted from the images. This captures low-level visual information such as color distribution.
ResNet50 Features: A pre-trained ResNet50 model (trained on ImageNet) is used to extract high-level semantic features from the images.

Fine-Tuning ResNet50:
Freezing Layers: Initially, all layers of the ResNet50 model are frozen to prevent changes This ensures that the pre-trained weights remain intact while training the new layers.
Fine-Tuning: After initial training, the last 10 layers of ResNet50 are unfrozen, and the model is fine- tuned with a lower learning rate (`1e-5`) to adapt the pre-trained weights to the specific dataset
.
Augmentation:
Data Augmentation: The `ImageDataGenerator` is used to apply transformations like rotation, flipping, zooming, and shearing to the training images. This increases the diversity of the training data and improves generalization.

Similarity Search with FAISS:
FAISS Index: The FAISS library is used to create an efficient index for similarity search. The combined features (color + ResNet50) are normalized using L2 normalization to ensure consistent scaling.
Query Processing: For a given query image, the same feature extraction pipeline is applied. The query features are then compared to the indexed features using the L2 distance metric to retrieve the most similar images.

Siamese Network:
Base Network: A shared base network (ResNet50) is used to extract features from two input images. Distance Metric: The absolute difference between the feature vectors of the two images is computed using a `Lambda` layer.
Output Layer: A single neuron with a sigmoid activation function outputs a similarity score between 0 and 1, where 1 indicates identical images and 0 indicates dissimilar images.

Architecture



Future Scope

1. Advanced Feature Extraction:

- Multimodal Features: Incorporate texture, shape, and deep learning features from advanced CNNs (e.g., Efficient Net, Vision Transformers) alongside color and ResNet50.
- Hybrid Models: Combine textual metadata (e.g., captions, tags) with visual features for multimodal retrieval using NLP techniques (e.g., BERT, CLIP).

2. Enhanced Similarity Learning:

- Triplet Loss and Attention Mechanisms: Improve Siamese networks with triplet loss for finer-grained similarity and attention modules to focus on discriminative regions.
- Cross-Modal Contrastive Learning: Train models to align image features with textual or contextual embeddings for richer representations.

3. Scalability and Efficiency:

- Distributed Indexing: Explore distributed FAISS setups or hybrid indexing (e.g., HNSW + FAISS) for large-scale datasets.
- Model Optimization: Use quantization, pruning, or knowledge distillation to speed up inference and reduce resource usage for real-time applications.

4. User-Centric Enhancements:

- Relevance Feedback: Implement interactive refinement where users label retrieved results to iteratively improve search accuracy.
- Explainability: Integrate visualization tools (e.g., Grad-CAM) to highlight image regions influencing retrieval, enhancing transparency.

5. Deployment and Accessibility:

- Edge Computing: Deploy lightweight models on edge devices for low-latency, privacy-preserving retrieval.
- APIs and Interfaces: Develop user-friendly web/mobile interfaces with APIs for seamless integration into applications.

6. Cross-Domain Adaptability:
- Transfer Learning: Adapt pre-trained models to specialized domains (e.g., medical, fashion) via fine- tuning or domain adaptation techniques.
- Few-Shot Learning: Leverage meta-learning to handle niche datasets with limited labeled samples.

7. Ethical and Responsible AI:
- Bias Mitigation: Audit datasets and models for biases, incorporating fairness-aware algorithms.
- Privacy Preservation: Apply federated learning or differential privacy to protect sensitive image data.

8. Dynamic and Autonomous Systems:
- Continuous Learning: Automate retraining and FAISS index updates as new data arrives, ensuring up- to-date performance.
- Self-Supervised Learning: Reduce reliance on labeled data by pre-training on unlabeled datasets using contrastive or reconstruction tasks.

By addressing these areas, the CBIR system can evolve into a more robust, scalable, and user-friendly solution, adaptable to diverse real-world applications while prioritizing ethical and sustainability considerations.

Conclusion

This implementation combines deep learning (ResNet50), traditional feature extraction (color histograms), and efficient similarity search (FAISS) to build a robust CBIR system. The Siamese network further enhances the system's ability to learn similarity metrics between images. By leveraging these techniques, the system achieves high accuracy and efficiency in retrieving visually similar images

References

[1]. K. Pathoee, D. Rawat, A. Mishra, V. Arya, M. K. Rafsanjani, A. K. Gupta, A cloud-based predictive model for the detection of breast cancer, Int. J. Cloud Appl. Comput., 12 (2022), 1–12.

[2]. A. W. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Trans. Pattern Anal. Mach. Intell., 22 (2000), 1349–1380.

[3]. J. Zhou, X. Zhang, Y. Wang, An efficient CBIR system using color and texture fusion, J. Vis. Commun. Image Represent., 76 (2021), 103110.

[4]. N. Kumar, M. Saini, A. Arora, Convolutional neural networks for feature extraction in content- based image retrieval, Expert Syst. Appl., 183 (2021), 115-212.

[5].S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, IEEE Conf. Comput. Vis. Pattern Recognit., 2 (2006), 2169– 2178.

[6].F. Kamis, R. Ahmad, N. Rahim, A comprehensive study on CBIR using K-means clustering and CNN embeddings, Appl. Comput. Intell. Soft Comput., 2021 (2021), 1–12.

[7].D. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis., 60 (2004), 91–110.