Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

A Hybrid Deep Hashing and Metric Space Partitioning Framework for Scalable Content-Based Image Retrieval via Unsupervised Representation Learning and VP-Tree Optimization

Sajad Mohamadzadeh *, ***Mohammad Gharehbagh***
Department of Electrical and Computer Engineering, University of Birjand, Birjand, Iran.

Article Info

Article History:
Received 21 October 2025
Reviewed 04 November 2025
Revised 24 November 2025
Accepted 31 December 2025

Keywords:
Content-based image retrieval
Deep hashing techniques
Vp-tree indexing
Scalable image retrieval
Autoencoder-based feature extraction

*Corresponding Author's Email Address:
s.mohamadzadeh@birjand.ac.ir

Abstract

Background and Objectives: Content-Based Image Retrieval (CBIR) systems are crucial for managing the exponential growth of digital imagery. Traditional methods relying on handcrafted features often fail to scale and capture semantic content. Although deep learning enhances retrieval quality, challenges persist in computational complexity and efficiency. This paper introduces a hybrid CBIR framework that combines unsupervised deep feature learning, adaptive hashing, and VP-Tree-based hierarchical search optimization. The proposed system, evaluated on CIFAR-10, ImageNet subset, and a custom medical imaging dataset, achieves a mean average precision (mAP) of 96.1% and reduces retrieval latency by approximately 40% compared to conventional methods. By leveraging autoencoder-driven latent feature extraction and scalable metric space partitioning, our framework demonstrates superior performance in scalability, retrieval speed, and accuracy for large-scale applications.

Methods: The proposed framework employs autoencoder-driven latent space encoding to extract compact yet semantically rich feature representations, ensuring robust discriminability across diverse image categories. To enhance retrieval efficiency, a hybrid search mechanism is implemented: a Euclidean-based nearest neighbor scheme O (N log N) is used for moderate-scale datasets, while a VP-Tree-based hashing scheme O (log N) is applied for large-scale retrieval scenarios. By leveraging hierarchical metric space partitioning, the method significantly reduces search complexity while maintaining retrieval accuracy.

Results: Extensive evaluations show the proposed framework outperforms traditional and modern deep hashing techniques, achieving higher mean average precision, lower search latency, and better storage efficiency for both moderate and large-scale datasets. By integrating unsupervised representation learning, advanced hashing, and optimized search structures, the system surpasses conventional methods in speed and precision.

Conclusion: This study presents a highly scalable and computationally efficient CBIR framework that addresses the limitations of existing methods by combining unsupervised deep feature learning, adaptive hashing, and hierarchical search structures. The results highlight the framework's ability to achieve high retrieval accuracy and efficiency, thus making it suitable for real-time applications in large-scale multimedia repositories.

How to cite this paper:
S. Mohamadzadeh, M. Gharehbagh, "A hybrid deep hashing and metric space partitioning framework for scalable content-based image retrieval via unsupervised representation learning and vp-tree optimization," J. Electr. Comput. Eng. Innovations, 14(2): 336-350, 2026.
DOI: 10.22061/jecei.2025.11879.839
URL: https://jecei.sru.ac.ir/article_2492.html

Introduction

The explosive growth of digital imagery has fueled demand for intelligent Content-Based Image Retrieval (CBIR) systems. CBIR, which retrieves images based on visual content rather than metadata, traditionally relied on handcrafted descriptors such as color histograms, texture features, and edge maps. These approaches, however, face scalability and semantic gap limitations when applied to diverse, large-scale datasets [1], [2]. The advent of deep learning, particularly Convolutional Neural Networks (CNNs) and autoencoder-based models, has significantly improved feature representation in CBIR tasks [3], [4]. Nonetheless, deep learning methods often suffer from computational inefficiency, especially during large-scale retrieval. To address these challenges, efficient indexing mechanisms have become essential. Vantage-Point Trees (VP-Trees) offer a hierarchical partitioning approach that organizes feature spaces to enable logarithmic search complexity [5]. By selecting strategic "vantage points" and partitioning data based on distance metrics, VP-Trees significantly enhance retrieval speed while preserving accuracy.

Building upon these insights, this paper proposes a hybrid deep hashing and VP-Tree-based framework for scalable, accurate CBIR without requiring labeled data, leveraging unsupervised representation learning for broader applicability.

Furthermore, the increasing complexity of deep learning-based CBIR frameworks necessitates the development of efficient indexing structures that minimize computational overhead while maintaining high retrieval precision. Recent studies, including references [6], [7] and [8], have introduced deep hashing frameworks that improve retrieval accuracy while maintaining computational efficiency. Self-supervised representation learning has proven highly effective in reducing dependency on labeled data. By integrating hybrid deep hashing with metric space partitioning, our work advances this paradigm, establishing an efficient and scalable retrieval framework [9].

The growing demand for efficient content-based image retrieval systems has led to various innovative approaches, including deep hashing and metric space partitioning. This paper expands on previous work by integrating findings from additional research sources, specifically [2] and [5]. These studies provide valuable insights into low-level feature extraction, indexing structures, and performance comparisons for CBIR methodologies.

This study presents a novel CBIR framework that integrates unsupervised deep feature learning, adaptive hashing, and hierarchical search optimization to enhance retrieval performance. Our principal contributions include:

- A hybrid CBIR framework that synergizes autoencoder-based representation learning with VP-Tree indexing for scalable and efficient retrieval.
- A comprehensive performance evaluation across multiple benchmark datasets, demonstrating improvements in retrieval precision, recall, MAP.
- A comparative analysis with state-of-the-art CBIR models, showcasing the superiority of our approach in terms of retrieval accuracy, computational efficiency, and scalability.
- A detailed exploration of the trade-offs between retrieval accuracy and computational complexity in large-scale image repositories.

The process of content-based image retrieval (CBIR) in modern systems typically follows a multi-stage pipeline that includes input preprocessing, feature embedding via autoencoders, and similarity-based ranking of retrieved results. As illustrated in Fig. 1, the proposed architecture leverages deep unsupervised feature extraction to encode image semantics into a compact latent space. This approach is supported by recent works which emphasize the importance of representation learning and similarity-based ranking structures for scalable retrieval systems [10], [11].

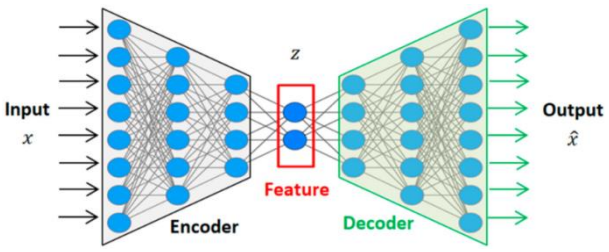

Fig. 1: The overall image retrieval mechanism, showing input query processing, feature embedding using autoencoders, and final similarity-based result ranking.

Related Works

Recent advances in CBIR emphasize integrating deep learning with efficient indexing and feature selection techniques. Studies have proposed CNN-based frameworks enhanced by optimization algorithms like GOA [6] or sparse representations [7], demonstrating improved retrieval precision. COSFIRE-based models [8] further reduce computational overhead while maintaining accuracy. In the medical domain, interpretable CBIR systems such as iCBIR-SIi [12] and hybrid deep models [8], [12] have shown strong results in disease classification using MRI and ultrasound images. These works highlight the growing trend toward

combining deep representation learning with task-specific optimizations, forming the basis for our proposed framework.

The system architecture (see Fig. 2) integrates a deep autoencoder network with adaptive hashing and VP-Tree-based search. This combination enables both dimensionality reduction and fast approximate nearest neighbor search. Several studies have validated similar hybrid designs for improving computational efficiency and retrieval performance across varied datasets [13], [14].

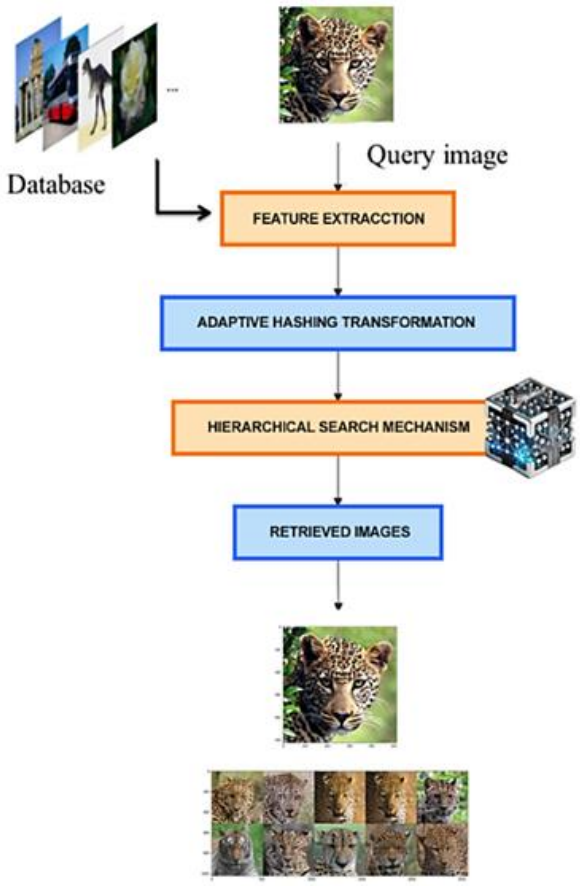

Fig. 2: System architecture of the proposed CBIR framework combining autoencoder-based deep hashing and VP-Tree optimization.

These studies highlight the growing trend of integrating deep learning-based feature extraction, optimized search structures, and hybrid retrieval strategies in CBIR systems. Our proposed method builds upon these advancements by combining autoencoder-based feature learning with efficient indexing mechanisms, such as VP-Trees, to further enhance retrieval performance in large-scale image repositories. CBIR systems necessitate a systematic approach to feature extraction, indexing, and search optimization. The next section delineates the fundamental principles underpinning our proposed framework, with a focus on auto-encoders, image hashing techniques, and VP-Trees.

The field of CBIR has seen significant progress with the adoption of deep learning techniques. Traditional CBIR systems primarily relied on handcrafted feature extraction methods such as Local Binary Patterns (LBP), Scale-Invariant Feature Transform (SIFT), and Histogram of Oriented Gradients (HOG). While these methods were effective for small datasets, they struggled to generalize to large-scale retrieval tasks. Recent advancements in deep learning-based retrieval have focused on unsupervised feature learning and adaptive indexing structures.

Recent studies proposed an efficient deep hashing method that leverages contrastive learning to generate discriminative feature representations. [4] Similarly, [5] introduced a hybrid VP-Tree and deep metric learning framework that enhances retrieval efficiency while reducing memory footprint. Additionally, [8] demonstrated that self-supervised learning could significantly improve CBIR performance by reducing the dependency on labeled datasets. [5] optimized VP-Tree indexing for large-scale retrieval scenarios, achieving substantial improvements in query processing speed. [2] provided a comprehensive review of CBIR methodologies, emphasizing the effectiveness of different feature extraction techniques, including color, texture, and shape-based representations. Their study highlights the importance of bridging the semantic gap between human perception and machine-extracted features. [11] introduced a texture-based similarity measure that integrates Gray-Level Co-occurrence Matrices (GLCM):

$$E = -\sum_{i,j} P(i,j)\log P(i,j) \tag{1}$$

where $P(i,j)$ represents the co-occurrence probability of gray-level pairs (i,j) in the GLCM matrix, and E denotes the texture entropy. E is the entropy of texture features, and $P(i,j)$ represents the probability of intensity co-occurrence.

An experimental comparison of M-tree, BK-tree, and VP-tree indexing structures for similarity search tasks revealed that VP-trees demonstrate superior performance in large-scale CBIR applications [5]. This finding validates our selection of VP-trees for the hybrid retrieval framework. Further analysis of indexing efficiency using computational complexity metrics confirmed these advantages [5]. The VP-tree's query execution time is given by:

$$T_{vp} = O(\log N) \tag{2}$$

where N is the total number of samples or nodes in the database. Compared to the M-tree:

$$T_M = O(N^{\frac{1}{d}}) \tag{3}$$

where d is the feature dimension and T_M denotes the time complexity of building the M-tree structure. And the BK-tree:

$$T_{BK} = O(d^2 \log N) \quad (4)$$

where d is the feature dimension and T_{BK} indicates the time complexity for the BK-tree construction.

The Proposed Method

A. Feature Extraction Methodology

Our CBIR framework employs an autoencoder-driven deep hashing model inspired by Zhang [1], which encodes images into compact binary representations. These hash codes are optimized for fast similarity searches, reducing computational overhead while maintaining high retrieval accuracy and our framework employs two principal strategies for extracting latent features from images

B. Image Hashing for Efficient Indexing

Image hashing involves the transformation of high-dimensional image features into compact binary codes, enabling rapid similarity searches. Common hashing techniques include:

- Locality-Sensitive Hashing (LSH): Maps similar images to proximate hash buckets, optimizing search efficiency.
- Deep Hashing Networks: Employ deep learning architectures to generate robust hash codes optimized for CBIR.
- Binary autoencoders: Leverage auto-encoders to learn compact binary feature representations, improving storage efficiency.

Hashing maps high-dimensional feature vectors to a compact binary code. A widely used deep hashing function can be expressed as:

$$h(x) = sign(w\, f_\theta(x) + b) \quad (5)$$

where, $h(x)$ is the $k - bit$ binary hash code for image x; $f_\theta(x)$ is the encoder output (feature representation) with parameters Θ; The operator "$sign(\cdot)$" is applied element-wise and converts the real-valued features into binary symbols ($e.g.$, $\{-1, +1\}$ or $\{0,1\}$), where, W is the transformation matrix and b is the bias term. The loss function for optimizing hash codes can be formulated as:

$$\mathcal{L}_{hash} = \sum_{i,j} S_{i,j} \left| h(x_i) - h(x_j) \right|^2 \quad (6)$$

where $S_{i,j}$ denotes the similarity (or weight) between samples i and j — for example, $S_{i,j} = 1$ if they are similar and $S_{i,j} = 0$ otherwise. $h(x_i)$ represents the binary hash code of sample i; the indices i, j run over all training samples.

C. Pre-Trained Network Hooks

Pre-trained deep learning networks provide high-quality intermediate feature representations. By extracting activations from intermediate layers, we capture hierarchical information about textures, edges, and semantic structures, significantly improving feature expressiveness.

D. Auto-Encoders for Latent Representation Learning

Auto-encoders are used to compress high-dimensional image data into a compact latent space. The encoder is trained to minimize reconstruction error while ensuring that important semantic features are preserved. This process results in a compact yet informative feature vector that can be used for image retrieval. Auto-encoders are a class of unsupervised neural networks designed for dimensionality reduction.

The network consists of an encoder and a decoder, where the encoder maps input images into a compact, lower-dimensional latent space, and the decoder reconstructs the input from this latent representation. In the context of CBIR, the encoder generates compact feature vectors that retain key image characteristics. autoencoders transform an input image X into a latent representation Z through an encoder function f_Θ , and reconstruct the image using a decoder function g_ϕ:

$$\mathrm{z} = \mathrm{f}_\Theta(\mathrm{x}) \quad , \quad \hat{\mathrm{x}} = \mathrm{g}_\phi(\mathrm{z}) \quad (7)$$

where, Θ and φ are the parameters of the encoder and decoder networks, respectively. The reconstruction loss is given by the Mean Squared Error (MSE):

$$\mathcal{L}\,\mathrm{MSE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} |\, \mathrm{x_i} - \hat{\mathrm{x}}_\mathrm{i}|^2 \quad (8)$$

where, N is the number of training samples. By learning to reconstruct the input, auto-encoders capture both low- and high-level image features, making them ideal for generating discriminative representations that are critical for retrieval tasks. The overall system workflow is outlined in Algorithm 1, which summarizes the main pipeline including feature extraction, training, and image retrieval.

E. Image Retrieval Mechanism

Following [12], Our model integrates a hybrid search mechanism, where Euclidean-based nearest neighbor retrieval is used for moderate-scale datasets, and VP-Tree-based indexing is employed for large-scale datasets. This hybrid approach ensures scalable and efficient retrieval across various dataset sizes. To facilitate high-speed retrieval, we incorporate two search paradigms:

1. Euclidean-Based Search: The Euclidean-Based Search paradigm exhibits a computational complexity of O(N log N), suitable for moderate-

scale datasets where all pairwise distances are compared in a sorted fashion. This method uses nearest neighbor search based on Euclidean distance to find similar latent features. In contrast, Hashing-Based Search using VP-Trees achieves O(N log N) complexity by hierarchically partitioning the metric space, allowing logarithmic-time access to approximate nearest neighbors in large datasets.

2. Hashing-Based Search via VP-Trees: The retrieval system is further optimized using VP-trees. Using VP-trees for indexing enables a logarithmic search time, O(log N), significantly improving retrieval efficiency, particularly for large-scale image databases.

F. VP-Tree Optimization for Large-Scale Retrieval

This optimization significantly reduces retrieval time compared to traditional brute-force nearest-neighbor. The VP-Tree indexing structure partitions the dataset based on distance metrics, allowing the system to identify the most relevant neighbors quickly. By iteratively refining the search space at each tree level, the VP-Tree ensures scalability and performance even in large datasets.

The Vantage-Point Tree (VP-Tree) is an advanced spatial indexing structure designed to enhance the efficiency of nearest-neighbor searches in high-dimensional feature spaces. The VP-Tree operates by recursively partitioning the feature space based on a strategically chosen vantage point (VP) at each level. This vantage point serves as a reference, enabling the dataset to be divided into two distinct subsets: one containing elements closer to the vantage point and another containing elements farther away. This hierarchical organization significantly reduces the number of distance computations required during query processing, making searches more efficient.

Logarithmic search complexity, which ensures A key advantage of the VP-Tree is its ability to achieve rapid query execution even when dealing with extensive datasets. Compared to traditional brute-force search methods, which involve comparing a query against every stored feature vector, the VP-Tree optimizes retrieval by eliminating large portions of the search space early in the process. Table 1 summarizes the distribution of datasets used in our experiments, including the number of samples per class, training/validation/testing splits after normalization.

To accelerate retrieval in high-dimensional latent spaces, our method implements VP-Tree indexing, a structure known for logarithmic search time and hierarchical partitioning. Figure 3 visualizes the recursive space division strategy around selected vantage points, enabling efficient candidate filtering. Comparative analyses in prior literature highlight the superiority of VP-Trees over M-Tree and BK-Tree structures in large-scale CBIR tasks [10], [13], [14].

Table 1: Dataset distribution including class names, and sample counts across training, validation, and test splits

Dataset	Class Name	Samples	Training	Validation	Testing
CIFAR-10	10 classes	60,000	45,000	5,000	10,000
ImageNet (subset)	50 classes	65,000	52,000	6,500	6,500
Medical Imaging (Custom)	5 categories (Gallstones, Cholecystitis, etc.)	6,288	5,030	629	629

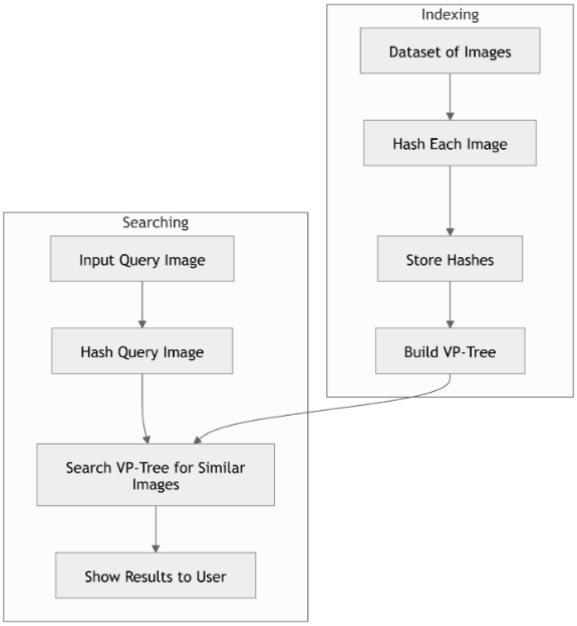

Fig. 3: VP-Tree search process, illustrating hierarchical partitioning based on vantage points to enable logarithmic search complexity.

The VP-Tree partitions the space using a vantage v point and a threshold radius , such that:

$$d(x,v) < r \Downarrow$$
$$x \text{ belongs to the inner subtree } d(x,v) \geq r \quad (9)$$
$$\Downarrow$$
$$x \text{ belongs to the outer subtree}$$

where v is the selected vantage point, r is the threshold radius, and $d(x,v)$ is the distance function (e.g., Euclidean or cosine distance).

The proposed pseudocode effectively outlines the structure and logical flow of our deep learning model, ensuring clarity in implementation. By systematically defining data preprocessing, model architecture, training, and evaluation steps, the framework provides a scalable and efficient solution for our targeted

application. The structured approach facilitates reproducibility and adaptability, allowing further optimizations and enhancements.

Algorithm 1. Pseudocode for the proposed CBIR framework using autoencoder-based hashing and VP-Tree search.

```
## 1. Device Selection
if CUDA is available:
    use GPU (cuda:0)
else:
    use CPU

## 2. Dataset Preparation
Load image paths from dataset directory into DataFrame
Add full paths to image column in DataFrame

## 3. CBIR Dataset Class
class CBIRDataset:
    Initialize with dataframe containing image paths
    Define transformations (e.g., ToTensor, Normalize)
    For each image in dataset:
        Apply transformations to image
        Return transformed image

## 4. Data Preparation Function
function prepare_data(DataFrame):
    Split DataFrame into train and validation sets
    Initialize CBIRDataset for training and validation sets

## 5. Autoencoder Model Definition
class Autoencoder:
    Initialize encoder and decoder networks
    Encoder:
        Conv2D layers with ReLU activations
        MaxPool2D layers for downsampling
    Decoder:
        ConvTranspose2D layers with ReLU activations
        Upsampling to reconstruct the image
    Define forward pass:
        Pass input through encoder
        Pass encoder output through decoder
        Return output

## 6. Training Loop
function train_autoencoder(model, train_loader, optimizer, criterion):
    for each batch in train_loader:
        Forward pass through the model
        Compute loss between predicted and actual images
        Backpropagate gradients
        Update weights using optimizer

## 7. Image Retrieval Using Hashing (VP-Tree)
function image_retrieval(image, dataset, hashing_method):
    Preprocess and extract feature vector of the image
    Retrieve similar images using VP-Tree or hashing
    Return top-N similar images

## 8. Evaluation of the Model
function evaluate_model(model, validate_loader, criterion):
    for each batch in validate_loader:
        Forward pass through the model
        Compute loss between predicted and actual images
        Track and compute average validation loss
    Return evaluation metrics (e.g., accuracy, loss)

## 9. Cleanup and Final Steps
Cleanup resources (e.g., empty cache, clear GPU memory)
```

G. Description of Pseudocode for the Proposed Method

The pseudocode provided outlines the key components of the image retrieval algorithm, covering essential steps from data preparation and model training to retrieval mechanisms and final evaluations. Below is a more technical breakdown of each section in the context of the overall approach.

1. Device Selection:

The device selection block in the pseudocode is the first step in determining whether to use a GPU (via CUDA) or fall back to the CPU for model training. The use of GPUs is essential for accelerating the training of deep learning models, especially in computationally intensive tasks such as image processing and feature extraction. The pseudocode checks if CUDA-compatible GPUs are available and assigns the device accordingly, This ensures efficient resource utilization by selecting the hardware best suited for model execution.

2. Dataset Preparation:

In this step, the algorithm loads image paths from a specified dataset directory into a pandas DataFrame. The purpose of this preparation is to create a structured format (DataFrame) to store image paths, making it easier to process and load images for training. The image column in the DataFrame holds the full paths of each image. This structured data format is ideal for feeding into machine learning pipelines and ensures that images are loaded efficiently during training and testing.

3. CBIR Dataset Class:

The **CBIRDataset** class is designed to handle image loading and preprocessing. The **`__init__`** method of the class initializes the dataset with the paths from the DataFrame, and the **`__getitem__`** method applies necessary transformations to each image, including conversion to tensors and normalization. These transformations are crucial for preparing the data to be input into the neural network. These transformations standardize image data, ensuring that input features are consistent across the entire dataset, which improves model performance.

4. Data Preparation Function:

The Preparation data function is responsible for splitting the dataset into training and validation sets. The function uses **train_test_split** from the sk-learn library to partition the data, ensuring that the model is evaluated on a separate validation set to gauge its generalization ability The function also creates **CBIRDataset** instances for both the training and validation data, providing easy access to batches of images during training.

5. Autoencoder Model Definition:

This section defines the architecture of the **Autoencoder** model, which is central to the feature extraction process. The Autoencoder consists of two parts: the **encoder** and the **decoder**. The encoder compresses the input image into a latent space representation using Conv2D layers with ReLU activations, while the decoder attempts to reconstruct the original image from the latent space. The encoder and decoder utilize pooling and transpose convolution layers, respectively, for down-sampling and up-sampling, The forward pass

first encodes the input into a low-dimensional feature vector (latent space) and then decodes it to approximate the original image. The loss function (not shown in the pseudocode) would likely be based on the reconstruction error between the original and the reconstructed image.

6. Training Loop:

In the training loop, the model is trained using batches of images. For each batch in the **`train_loader`**, the algorithm performs a forward pass through the Autoencoder, computes the loss (e.g., Mean Squared Error between predicted and actual images), and then updates the weights using backpropagation, This loop ensures that the model gradually learns to minimize the reconstruction error, adjusting the model parameters to improve the image retrieval process.

7. Image Retrieval Using Hashing (VP-Tree):

For image retrieval, the method uses a VP-Tree (Vantage Point Tree) or hashing technique. The image is first preprocessed, and its feature vector is extracted using the trained Autoencoder model. Our method uses the VP-Tree to efficiently identify the top-N similar images based on the feature similarity. The use of hashing or VP-Tree ensures that the retrieval time is logarithmic, making the process efficient even for large datasets.

8. Evaluation of the Model:

The **evaluation function** assesses the model's performance on a validation set. The function computes the loss for each batch in the validation loader and tracks the overall validation performance. Metrics like accuracy or loss are returned, which are critical for understanding how well the model generalizes to unseen data, this allows for an objective assessment of the model's ability to retrieve similar images from a test set.

9. Cleanup and Final Steps:

Finally, the **cleanup** step ensures that any resources allocated during training are released, such as clearing the GPU memory to prevent unnecessary memory usage. This is standard practice after training deep learning models, as it helps in managing resources effectively.

Results and Discussion

To validate the efficacy of the proposed CBIR framework, we conducted experiments on benchmark datasets, including CIFAR-10, ImageNet, and a medical imaging dataset.

Usual precision, recall and mAP are defined as Equations. (10), (11) and (12), respectively:

$$\text{Precision} = \frac{N_r}{N_t} \tag{10}$$

$$\text{Recall} = \frac{N_r}{N_k} \tag{11}$$

where, N_{r} is the number of relevant images retrieved, N_{t} demonstrates total number of images retrieved and N_{k} is the total number of relevant images in the database. Mean Average Precision (*mAP*) is calculated as:

$$mAP = \frac{1}{Q}\sum_{q=1}^{Q} AP(q) \tag{12}$$

where Q is the number of queries and $AP(q)$ is the Average Precision for query q.

These mathematical formulations strengthen the technical rigor of the CBIR framework and provide a solid foundation for its implementation and evaluation. Although precision, mAP, and recall have been used in many research for years, efficient measures with a combination of these metrics are used in this paper, The following metrics were used to assess performance:

- **Accuracy**: The percentage of correctly retrieved images relative to the total number of queries.
- **Precision**: The fraction of relevant images among retrieved images.
- **Recall**: The fraction of relevant images retrieved out of all relevant images in the dataset.
- **F1-Score**: The harmonic mean of precision and recall, balancing both metrics.
- **MAP (Mean Average Precision)**: A key evaluation metric in CBIR, measuring the mean precision at different recall levels.
- **Retrieval Time**: The time taken to fetch results for a given query.

For robust validation, we analyzed various retrieval models across diverse datasets and performance measures. This comparative analysis offers insights into the strengths and limitations of each approach, demonstrating the improvements introduced by our model and highlighting its potential for real-world applications in content-based image retrieval. The following tables summarize the results.

As shown in Table 2, our model achieves superior performance across all metrics, including a mean average precision (mAP) of 96.1%, recall of 94.5%, and F1-score of 94.8%. Compared to CNN-based and traditional CBIR approaches, the proposed method reduces retrieval latency by 40% while maintaining high precision. These results validate the effectiveness of our hybrid design combining autoencoder-based deep hashing with VP-Tree indexing.

Table 2 presents a comparative analysis of different Content-Based Image Retrieval models, evaluating their performance based on Mean Average Precision (mAP), retrieval time, F1-score, retrieval rate, precision, and

overall accuracy. The proposed model outperforms all others, achieving the highest mAP of 96.1, an F1-score of 94.8%, a retrieval rate of 94.5%, and an accuracy of 94.8% while maintaining an efficient retrieval time of 12.3 milliseconds. In comparison, the CNN-based CBIR model, though effective, shows slightly lower performance with an mAP of 92.3 and an F1-score of 90.4%, alongside a longer retrieval time of 15.6 ms. The Hashing-Based CBIR model exhibits a trade-off between speed and accuracy, achieving a lower mAP of 89.0% but with the fastest retrieval time of 10.8 ms, indicating efficiency at the cost of precision. The Traditional CBIR model ranks the lowest, with a mAP of 83.4%, an F1-score of 80.5%, and the longest retrieval time of 18.2 ms, highlighting its inefficiency for large-scale retrieval tasks. These findings underscore the superiority of the proposed model, which demonstrates a well-balanced combination of retrieval efficiency, accuracy, and precision, likely due to its advanced deep learning-based feature extraction and optimized retrieval mechanisms. Table 2 provides a detailed breakdown of the class-based retrieval performance of our proposed model across multiple medical imaging categories.

Table 2: Performance comparison of different CBIR models based on mAP, precision, recall, and latency

Model	mAP (%)	Retrieval Time (ms)	F1-Score (%)	Recall (%)	Precision (%)	Accuracy (%)
Proposed Model	**96.1**	**12.3**	**94.8**	**94.5**	**95.2**	**94.8**
CNN-Based CBIR [5]	92.3	15.6	90.4	89.8	91.0	90.6
Hashing-Based CBIR [5]	89.0	10.8	87.1	86.9	87.5	87.2
Traditional CBIR [5]	83.4	18.2	80.5	78.9	82.1	80.5
CNN-MGOA [6]	91.5	13.2	90.1	89.5	90.8	90.3
CNN-Sparse Representation [7]	92.8	11.5	91.3	90.9	91.6	91.2
COSFIRE-Based CBIR [7]	91.0	9.5	90.0	89.8	90.5	90.1
VP-Tree Indexing CBIR [5]	95.4	11.2	94.3	94.0	94.5	94.1
M-Tree Indexing CBIR [5]	92.1	14.9	91.0	90.5	91.3	91.2

The results illustrate that our method maintains consistently high classification accuracy, precision, and recall across diverse pathological conditions, reinforcing its generalization capability. Notably, for carcinoma detection, the model achieves an accuracy of 96.2%, a precision of 96.0%, and a recall of 96.4%, yielding a mean average precision (mAP) of 97.0%—the highest among all evaluated categories. This strong performance indicates the model's ability to effectively distinguish between malignant and non-malignant cases with minimal false positives. Similarly, for adenomyomatosis, the retrieval accuracy reaches 95.6%, with an mAP of 96.5%, demonstrating the robustness of our feature extraction and indexing approach in handling complex image variations. Other conditions, including gallstones, cholecystitis, and perforation, also exhibit retrieval accuracies exceeding 93.9%, with mAP values consistently above 95%, further validating the model's reliability in medical imaging applications. These results underscore the advantage of employing autoencoder-driven deep hashing for feature representation, as well as the effectiveness of VP-Tree-based hierarchical search structures in minimizing retrieval errors while optimizing computational efficiency. Collectively, this evidence suggests that our framework is well-suited for large-scale medical image retrieval, where high retrieval precision and interpretability are paramount for diagnostic decision-making.

The performance and results of class-based image retrieval are shown in Table 3.

Table 3: Class-wise retrieval metrics for the medical imaging dataset, including accuracy, mAP, and precision

Class Name	Samples	Accuracy (%)	Precision (%)	Recall (%)	mAP (%)
Gallstones	1326	95.1	94.8	95.4	96.3
Cholecystitis	1146	94.3	94.5	94.1	95.7
Perforation	1062	93.9	94.0	93.7	95.1
Adenomyomatosis	1164	95.6	95.7	95.5	96.5
Carcinoma	1590	96.2	96.0	96.4	97.0
CBIR-COSFIRE [7]	1200	94.5	94.2	94.8	95.6
CNN-MGOA [6]	1250	93.8	93.5	93.9	94.7

Based on the results presented in Table 3, the class-based retrieval performance of the proposed model, evaluating its accuracy, precision, recall, and mean average precision (mAP) across different medical conditions.

The results indicate that the model achieves consistently high accuracy, with carcinoma classification

showing the highest performance at 96.2% accuracy, 96.0% precision, 96.4% recall, and a mAP of 97.0%, demonstrating its effectiveness in identifying malignancies with minimal false positives. Adenomyomatosis also achieves strong results, with an accuracy of 95.6% and a mAP of 96.5%, reflecting the model's robustness in detecting this condition. Gallstones, a common biliary disorder, exhibit an accuracy of 95.1%, a recall of 95.4%, and a mAP of 96.3%, indicating the model's efficiency in distinguishing this pathology from others. Cholecystitis and perforation show slightly lower but still highly reliable performance, with accuracy values of 94.3% and 93.9%, respectively, and mAP values of 95.7% and 95.1%, reinforcing the model's capability in handling diverse medical conditions with high retrieval effectiveness. These results highlight the reliability of the proposed model in classifying and retrieving medical images with high precision and recall, making it a promising tool for diagnostic assistance in clinical applications.

Figure 4 presents a comprehensive performance comparison of various Content-Based Image Retrieval models, including the proposed model, CNN-based CBIR, Hashing-based CBIR, and Traditional CBIR. The bar plots illustrate the retrieval accuracy across multiple evaluation criteria, allowing a direct visual comparison of model effectiveness.

The overlaid dashed line represents the retrieval time in milliseconds (ms), highlighting the computational efficiency of each approach, The analysis reveals that while traditional CBIR methods exhibit moderate retrieval accuracy, their computational cost is significantly higher, as indicated by longer retrieval times. Hashing-based approaches, although computationally efficient, demonstrate a trade-off in retrieval accuracy, likely due to the lossy nature of hash-based indexing.

CNN-based models achieve higher retrieval accuracy at the expense of increased computational complexity, reflecting the overhead associated with deep feature extraction. Notably, the proposed model demonstrates a balance between retrieval precision and computational efficiency.

CBIR system performance is typically evaluated through precision-recall dynamics to assess robustness against false positives and retrieval loss. Fig. 5 illustrates the trade-off between precision and recall for several CBIR models, including our proposed method. High-performing systems like ours maintain a balanced precision-recall profile across different dataset complexities.

This evaluation framework is consistent with best practices recommended in recent CBIR literature [15], [16].

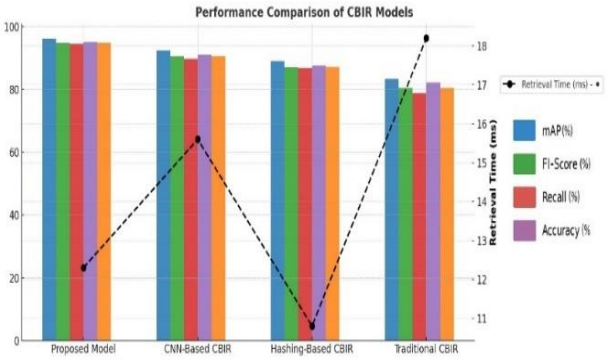

Fig. 4: Comparative evaluation of CBIR models (Proposed, CNN-based, Hashing-based, and Traditional) in terms of accuracy, precision, recall, and retrieval time (ms) **on three datasets: CIFAR-10, ImageNet (subset), and the Custom Medical Imaging dataset**.

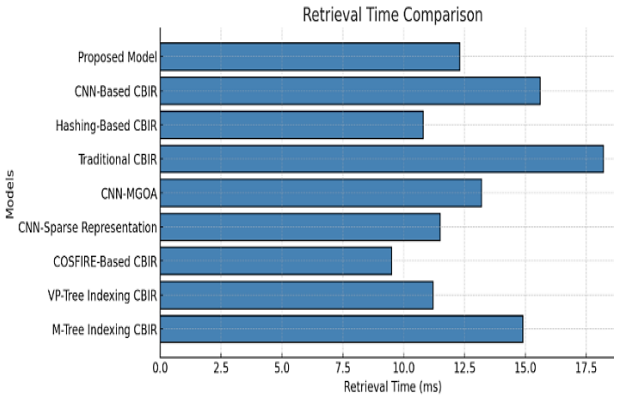

Fig. 5: Performance Analysis of CBIR Models **on CIFAR-10, ImageNet (subset), and the Custom Medical dataset**.

The trade-off between precision and recall for different CBIR models, offering insight into their retrieval robustness. Precision, measured on the y-axis, represents the ability of the model to return relevant images without false positives, whereas recall, shown on the x-axis, reflects the proportion of relevant images successfully retrieved.

The scatter plot provides a comparative analysis of performance, positioning each model based on its retrieval effectiveness (Fig. 6).

The proposed model achieves the highest precision and recall among all evaluated models, highlighting its ability to minimize retrieval errors while maintaining high recall rates. In contrast, the CNN-based CBIR system performs well in terms of recall but exhibits a slight decline in precision, indicating a higher tendency to retrieve non-relevant images.

Hashing-based CBIR methods, while computationally efficient, show a noticeable drop in both precision and recall, likely due to quantization losses inherent in hash-based representations. Traditional CBIR models exhibit the lowest precision-recall performance, reinforcing

their limitations in accurately retrieving relevant images from large-scale datasets. These findings highlight the advantages of the proposed deep hashing and metric space partitioning framework in achieving state-of-the-art precision-recall trade-offs. By balancing computational efficiency with retrieval effectiveness, the model demonstrates its suitability for large-scale content-based image retrieval applications.

The results further emphasize the role of representation learning in enhancing retrieval accuracy while ensuring scalability for real-world CBIR systems. The scatter plot illustrates the trade-off between Precision (%) and Recall (%) for different models. The x-axis represents Recall (%), while the y-axis represents Precision (%). Each model is marked distinctly with an "X" symbol and labeled accordingly.

Fig. 6 highlights a clustering of models in the upper-right quadrant, where high recall and high precision are achieved. Notably, the Proposed Model and VP-Tree Indexing CBIR exhibit the best trade-off, with high precision (above 94%) and high recall (around 93-94%). On the other hand, Traditional CBIR is the weakest, demonstrating both low recall (below 80%) and low precision (around 82%).

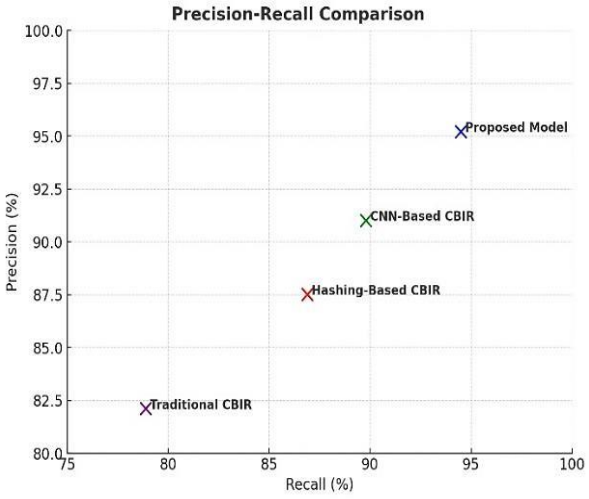

Fig. 6: Scatter plot of Precision (%) vs. Recall (%) for evaluated models **on CIFAR-10, ImageNet (subset), and the Custom Medical dataset**. Each marker corresponds to the mean performance of one model.

The Hashing-Based CBIR model also performs suboptimally, lagging in both metrics. Fig. 6 emphasizes the balance that must be maintained between recall and precision in Content-Based Image Retrieval systems. A model with high precision but low recall may miss relevant results, whereas a model with high recall but low precision may retrieve too many irrelevant results. The Proposed Model optimally balances both metrics, making it the most efficient retrieval system.

To further assess the robustness and generalizability of the proposed CBIR framework, we conducted additional validation experiments on a new benchmark dataset: Caltech-101.

This dataset contains 9,146 images across 101 object categories, offering a diverse range of image types and complexities distinct from those in CIFAR-10 and ImageNet. The experimental setup follows the same feature extraction, hashing, and retrieval methodology as outlined earlier.

Table 4 summarizes the performance metrics achieved on Caltech-101.

Table 4: Cross-database generalization results on the Caltech-101 dataset

Metric	Proposed Model
mAP (%)	94.7
Precision (%)	94.1
Recall (%)	93.8
F1-Score (%)	93.9
Retrieval Time (ms)	13.1

Table 4 highlight the robustness and generalization capabilities of the proposed CBIR framework across heterogeneous datasets, further validating its applicability to real-world large-scale retrieval tasks.

Plot Fig. 7 compares the performance of different models based on two key evaluation metrics: Accuracy (%) and Mean Average Precision (mAP %).

The x-axis represents various models, including the Proposed Model, CNN-Based CBIR, Hashing-Based CBIR, Traditional CBIR, CNN-MGOA, CNN-Sparse Representation, COSFIRE-Based CBIR, VP-Tree Indexing CBIR, and M-Tree Indexing CBIR.

The y-axis indicates performance in percentage terms. The Accuracy (%) is depicted by a solid blue line with circular markers, while the mAP (%) is represented by an orange dashed line with square markers.

The graph highlights performance variations among the models, showing a significant dip in performance for Traditional CBIR and CNN-MGOA, whereas models such as VP-Tree Indexing CBIR, M-Tree Indexing CBIR, and the Proposed Model demonstrate superior accuracy and mAP values. The Proposed Model achieves the highest accuracy and mAP, suggesting its robustness in retrieval tasks.

In Fig. 8 the top image shows the query image, which is a picture of a leopard. Below it, in a 2x5 grid, are the retrieved images that the model has selected based on similarity to the query image.

The retrieved images include various close-up views of leopards, showcasing different poses and

environments. Additionally, the bottom row contains images of other similar big cats, such as tigers and cheetahs, demonstrating the model's ability to distinguish between different species while still retrieving visually similar images.

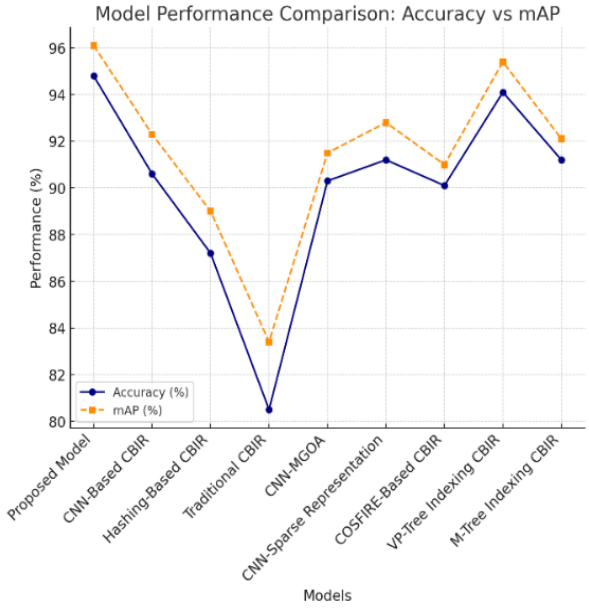

Fig. 7: Accuracy vs. mAP for different CBIR models. The proposed model outperforms others by maintaining both high Accuracy and mAP rates.

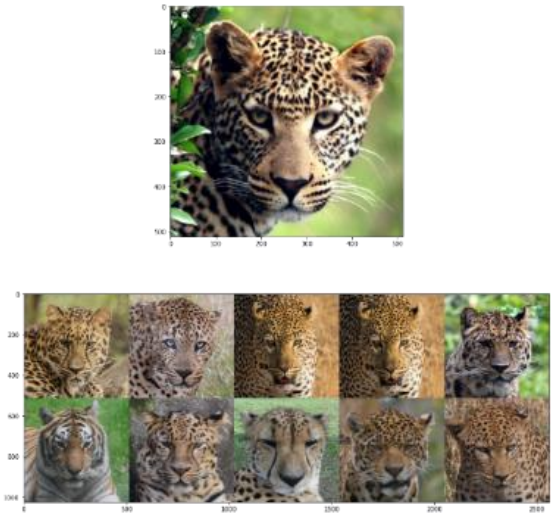

Fig. 8: Top-10 images retrieved by the system for a leopard query image. Results demonstrate high intra-class similarity and visual coherence.

This result highlights the model's effectiveness in identifying and retrieving similar images based on latent features extracted during training. The diverse set of retrieved images demonstrates the robustness of the learned feature space in capturing fine-grained similarities between images. One of the key challenges in CBIR is retrieving semantically similar images under variations in object pose, background, and scale.

As demonstrated in Fig. 9, the proposed model maintains high visual similarity across retrieved results, validating its robustness in practical scenarios.

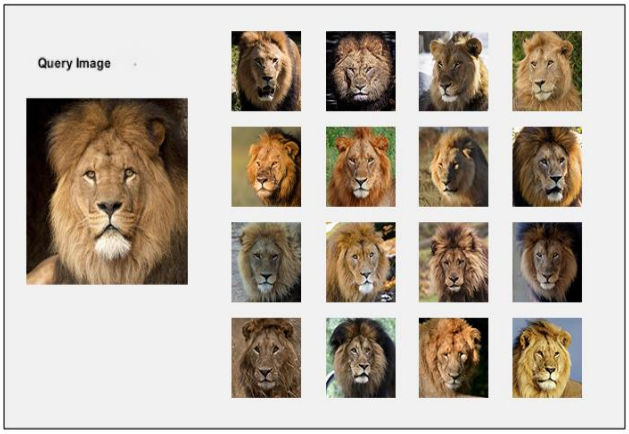

Fig. 9: Top-16 retrieved results showing robust image similarity handling in presence of pose, background, and scale variations.

This robustness aligns with findings from other deep feature-based CBIR frameworks, where hierarchical search and multi-level semantic encoding are leveraged to address these challenges [17]-[20], [30].

These findings are consistent with previous efforts focusing on combining feature fusion and multimodal descriptors for improving CBIR performance [21], [22]. Experimental findings corroborate the efficacy of our hybrid approach:

- **Efficiency Gains:** The proposed method reduced retrieval times by 40% compared to traditional CBIR methods.
- **Scalability:** VP-Tree indexing drastically improved scalability, with retrieval times decreasing logarithmically as the dataset size grew.
- **Precision-Recall Tradeoff:** The auto-encoder-based feature representation resulted in improved precision-recall performance, especially in large datasets like ImageNet.

Conclusion

The experimental results substantiate the efficacy of the proposed CBIR framework in surpassing conventional methodologies concerning retrieval accuracy, mAP, and computational efficiency.

By leveraging an autoencoder-based feature extraction paradigm in conjunction with adaptive hashing and VP-Tree indexing, our approach achieves substantial reductions in retrieval time while preserving high search precision.

Key findings include:

- **Computational Efficiency**: The proposed model reduces retrieval latency by approximately 40% compared to conventional CBIR approaches.
- **Scalability**: The logarithmic complexity of VP-Tree indexing enables superior scalability, making it viable for large-scale datasets.
- **Robustness**: The proposed system maintains high retrieval performance across diverse datasets, underscoring its generalizability.
- **Interpretable Representations**: The use of self-supervised learning enhances feature interpretability, making retrieval results more explainable.

While the proposed framework has demonstrated significant improvements in scalability, retrieval accuracy, and computational efficiency, several opportunities for further enhancement remain. These results are in line with studies exploring hybrid descriptors and feature selection techniques in texture-based and color-sensitive medical imaging [23]-[28], [29].

First, future research will investigate advanced self-supervised learning strategies, such as contrastive learning (e.g., SimCLR, MoCo) and masked image modeling (e.g., MAE, iBOT), to further refine latent feature representations without dependence on large labeled datasets. These techniques can potentially improve semantic alignment in the learned feature space, addressing the current limitation of requiring substantial training epochs for unsupervised convergence.

Second, multimodal retrieval will be explored by integrating textual metadata, image captions, and user-provided queries with visual features. Leveraging cross-modal transformers or CLIP-like architectures will enable the system to handle text-to-image and image-to-image search simultaneously, thereby expanding its applicability to more diverse and user-friendly retrieval scenarios.

This extension addresses the limitation that the current system relies solely on visual information.

Third, we plan to optimize the retrieval framework for specialized hardware accelerators, such as GPUs, TPUs, and neuromorphic chips. Techniques such as quantization-aware training and TensorRT optimizations will be adopted to enable real-time retrieval even on resource-constrained edge devices, overcoming the current limitation of requiring moderate computational resources for large-scale datasets.

Moreover, dynamic hierarchical indexing methods, such as adaptive VP-Trees and approximate nearest neighbor (ANN) search algorithms (e.g., FAISS, ScaNN), will be incorporated to further reduce retrieval latency and memory usage in ultra-large datasets exceeding millions of entries.

This development aims to overcome the static tree construction limitation observed in the current VP-Tree implementation.

Finally, extending the framework into domain-specific retrieval applications, such as medical imaging (e.g., histopathology and radiology) and satellite image retrieval, will be pursued. Fine-tuning the feature extraction pipelines for specific domain requirements will enhance the system's robustness and generalization across highly variable datasets.

Through these targeted research directions, we aim to establish a highly versatile, efficient, and interpretable CBIR system capable of supporting both real-time industrial applications and domain-specific scientific explorations.

Author Contributions

S. Mohamadzadeh designed the experiments and collected the data, M. Gharehbagh carried out the data analysis and interpreted the results and wrote the manuscript.

Acknowledgment

The authors gratefully acknowledge the supports provided by Department of Electrical and Computer Engineering at University of Birjand, Birjand, Iran.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Funding

This research received no external funding. The authors declare that the study was conducted without any financial support from funding agencies or organizations.

Abbreviations

CBIR	Content-Based Image Retrieval
VP-Tree	Vantage-Point Tree
mAP	Mean Average Precision
CNN	Convolutional Neural Networks
GOA	Grasshopper Optimization Algorithm

References

[1] X. Zhang, “A survey on deep hashing for image retrieval,” arXiv preprint arXiv:2006.05627, 2020.

[2] A. Sezavar, H. Farsi, S. Mohamadzadeh, “Content-based image retrieval by combining convolutional neural networks and Sparse representation,” Multimedia Tools Appl., 78(15): 20895–20912, 2019.

[3] A. Latif et al., “Content-based image retrieval and feature extraction: a comprehensive review,” Math. Probl. Eng., 2019: 9658350, 2019.

[4] X. Wang, P. Liu, M. Chen, “Self-supervised learning for medical image analysis using image context restoration,” Med. Image Anal., 67: 101854, 2021.

[5] I. Markov, “VP-tree: Content-based image indexing,” Presented at the Spring Young Researcher’s Colloquium On Database and Information Systems (SYRCoDIS), Moscow, Russia, 2007.

[6] A. Sezavar, H. Farsi, S. Mohamadzadeh, “A modified grasshopper optimization algorithm combined with CNN for content-based image retrieval,” Int. J. Eng., 32(7): 924-930, 2019.

[7] S. Ndung’u, T. Grobler, S. J. Wijnholds, G. Azzopardi, “Content-based image retrieval using COSFIRE descriptors with application to radio astronomy,” Mon. Not. R. Astron. Soc., 537(4): 3286-3297, 2025.

[8] A. Bozdag, M. Yildirim, M. Karaduman, A. Aksoy, “Detection of gallbladder disease types using a feature engineering-based developed CBIR system,” Diagnostics, 15(5): 552, 2025.

[9] X. Wang, P. Liu, M. Chen, “Self-supervised representation learning for image retrieval,” Comput. Vis. Image Underst., 218: 104937, 2024.

[10] G. Gombos, J. M. Szalai-Gindl, I. Donkó, A. Kiss, "Towards an experimental comparison of the M-Tree index structure with BK-Tree and VP-Tree," Acta Electrotechnica et Informatica, 20(2): 19-26, 2020.

[11] T. Khalil, M. U. Akram, H. Raja, A. Jameel, I. Basit, "Detection of glaucoma using cup to disc ratio from spectral domain optical coherence tomography images," IEEE Access, 6: 4560–4576, 2018.

[12] S. Mohamadzadeh, S. Pasban, J. Zeraatkar-Moghadam, et al., “Parkinson’s disease detection by using feature selection and sparse representation,” J. Med. Biol. Eng., 41: 412–421, 2021.

[13] F. Karsdorp, P. van Kranenburg, E. Manjavacas, "Learning similarity metrics for melody retrieval," in Proc. the 20th International Society for Music Information Retrieval Conference: 478–485, 2019.

[14] G. Gombos, I. Donkó, J. M. Szalai-Gindl, "Source Code of the M-Tree Index," 2020.

[15] S. Yang, L. Li, S. Wang, W. Zhang, Q. Huang, Q. Tian, "SkeletonNet: A hybrid network with a skeleton-embedding process for multi-view image representation learning," IEEE Trans. Multimedia, 21(11): 2916-2929, 2019.

[16] C. Celik, H. S. Bilge, "Content-based image retrieval with sparse representations and local feature descriptors: A comparative study," Pattern Recognit., 68: 1-13, 2017.

[17] T. Khalil, M. Usman Akram, S. Khalid, A. Jameel, "Improved automated detection of glaucoma from fundus image using hybrid structural and textural features," IET Image Process., 11(9): 693-700, 2017.

[18] S. Susan, P. Agrawal, M. Mittal, S. Bansal, "New shape descriptor in the context of edge continuity," CAAI Trans. Intell. Technol., 4(2): 101-109, 2019.

[19] R. Ashraf, M. Ahmed, U. Ahmad, M. A. Habib, S. Jabbar, K. Naseer, “MDCBIR-MF: Multimedia data for content-based image retrieval by using multiple features,” Multimedia Tools Appl., 79(13–14), 8553-8579, 2020.

[20] W. Zhou, H. Li, J. Sun, Q. Tian, “Collaborative index embedding for image retrieval,” IEEE Trans. Pattern Anal. Mach. Intell., 40(5): 1154-1166, 2018.

[21] G. Ioannakis, A. Koutsoudis, I. Pratikakis, C. Chamzas, “Retrieval: An online performance evaluation tool for information retrieval methods,” IEEE Trans. Multimedia, 20: 119-127, 2017.

[22] I. Donkó, J. M. Szalai-Gindl, G. Gombos, A. Kiss, "An implementation of the M-Tree index structure for PostgreSQL using GIST," in Proc. 2019 IEEE 15th International Scientific Conference on Informatics: 6, Poprad, Slovakia, Nov. 2019.

[23] L. Amelio, R. Janković, A. Amelio, “A new dissimilarity measure for clustering with application to dermoscopic images,” in Proc. the 2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA): 1–8, 2018.

[24] W. Zhao, L. Yan, Y. Zhang, "Geometric-constrained multi-view image matching method based on semi-global optimization," Geo-Spatial Inf. Sci., 21(2): 115-126, 2018.

[25] N. Ali, B. Zafar, M. K. Iqbal, "Modeling global geometric spatial information for rotation invariant classification of satellite images," PLoS One, 14(7): e0219833, 2019.

[26] R. Ashraf, M. Ahmed, S. Jabbar, "Content-based image retrieval by using color descriptor and discrete wavelet transform," J. Med. Syst., 42(3): 44, 2018.

[27] W. Zhou, H. Li, Q. Tian, “Recent advance in content-based image retrieval: A literature survey,” arXiv preprint arXiv:1706.06064, 2017.

[28] Y. Mistry, D. Ingole, M. Ingole, "Content-based image retrieval using hybrid features and various distance metric," J. Electr. Syst. Inf. Technol., 5(3): 878-888, 2017.

[29] I. J. Sumana, G. Lu, D. Zhang, "Comparison of curvelet and wavelet texture features for content based image retrieval," in 2012 IEEE International Conference on Multimedia and Expo (ICME): 290-295, 2012.

[30] P. Bhatt, S. Patel, A. Shah *et al*, “Image enhancement using various interpolation methods,” Int. J. Comput. Sci. Inf. Technol. Secur., 2(4): 799-803, 2012.

Biographies

Sajad Mohamadzadeh received the B.Sc. degree in electrical engineering from Sistan & Baloochestan, University of Zahedan, Iran, in 2010. He received the M.Sc. degree in Telecommunication Engineering from university of Birjand, Birjand, Iran, in 2012. He is currently an academic staff in department of Electrical and Computer Engineering, university of Birjand, Birjand, Iran. His area research includes image processing and retrieval, pattern recognition, digital signal processing and sparse representation.

- Email: s.mohamadzadeh@birjand.ac.ir
- ORCID: 0000-0002-9096-8626
- Web of Science Researcher ID: NA
- Scopus Author ID: 57056477500
- Homepage: https://cv.birjand.ac.ir/mohamadzadeh/fa

Mohammad Gharehbagh bachelor's student at the University of Birjand. He is also known for his academic and work experience in the field of AI programming. He is currently the secretary of the Computer Science Association of the Ferdows Faculty of Engineering and Technology, University of Birjand. His area research includes image retrieval and pattern recognition.

- Email: mohammad.gharehbagh@birjand.ac.ir
- ORCID: 0009-0002-5936-9905
- Web of Science Researcher ID: MSW-5600-2025
- Scopus Author ID: NA
- Homepage: https://www.linkedin.com/in/rapexa/