SO-DRCNN WITH TERNION PARADIGM EXTRACTION ROUTINE FOR AN EFFECTIVE IMAGE RETRIEVAL SYSTEM

by

Akram Kazemisisi

B.Sc., University of Science and Engineering of Tehran, 2012

THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER SCIENCE

UNIVERSITY OF NORTHERN BRITISH COLUMBIA

June 2025

Abstract

This thesis addresses the challenges of semantic image retrieval and labeled data scarcity in Content-Based Image Retrieval (CBIR) by introducing SO-DRCNN, a novel Self-Optimizing DeepRec Convolutional Neural Network framework. SO-DRCNN leverages a hybrid approach, combining the strengths of handcrafted features (Ternion Paradigm: HOG, ICH, SERC) and deep learning. A pre-trained ResNet-50 backbone, enhanced with Recurrent Patching (Bi-LSTM), Spatial Pyramid Pooling (SPP/ASPP), and Attention mechanisms, extracts high-level semantic features. A key innovation is the Siamese-Driven Feature Fusion, where a Siamese network, trained with a contrastive loss, learns to adaptively combine handcrafted and deep features, optimizing the fused representation for similarity. This self-supervised training strategy (Auto-Embedder) eliminates the need for manual image labels. Experiments on benchmark datasets demonstrate that SO-DRCNN achieves state-of-the-art retrieval accuracy, outperforming traditional methods and demonstrating the effectiveness of the learned fusion strategy. The system is also integrated with Elasticsearch for scalable retrieval. This work contributes a robust, efficient, and interpretable solution for semantic CBIR.

Table of Contents

List of Tables

List of Figures

Glossary

1- Adaptive Histogram Equalization (AHE): A contrast enhancement technique that improves local image contrast by applying histogram equalization to localized regions, rather than the entire image. Mentioned in the context of image preprocessing techniques.

2- Atrous Spatial Pyramid Pooling: A module used in deep convolutional neural networks (CNNs) that captures multi-scale contextual information by applying convolutions with different dilation rates (atrous convolutions) to feature maps. Part of the SO-DRCNN architecture to enhance feature representation.

3- Auto-Embedder Architecture: A self-supervised learning framework, based on a Siamese network, that trains a model to generate embeddings optimized for similarity comparisons. In this thesis, it refers to the Siamese network used to train the Fusion Module for feature fusion, enabling data-driven weight learning.

4- Bag-of-Visual-Words (BoVW): A technique, adapted from text retrieval, that represents images as histograms of visual word occurrences. Local image features (extracted using ORB and Ternion descriptors) are quantized into a "visual vocabulary," and each image is represented by the frequency of each visual word. Used in this thesis for handcrafted feature extraction.

5- Bidirectional Long Short-Term Memory (Bi-LSTM): A type of recurrent neural network (RNN) that processes sequential data (like image patches in SO-DRCNN) in both forward and backward directions, capturing contextual dependencies from both preceding and succeeding elements in the sequence. Used in the Recurrent Patching Module of SO-DRCNN.

6- CBIR (Content-Based Image Retrieval): A technique for retrieving images from a database based on their visual content (features extracted from the images

themselves), rather than relying on textual annotations or metadata. This is the core task addressed by this thesis.

7- CNN Embedding: A feature vector representing an image, extracted from a Convolutional Neural Network (CNN). In this thesis, the CNN embedding is generated by the SO-DRCNN model.

8- Contrastive Loss: A loss function used in self-supervised learning (particularly with Siamese networks) that encourages similar inputs to have close embeddings and dissimilar inputs to have distant embeddings. This is the core training mechanism for the Siamese Network in this thesis, used to train the Fusion Module.

9- Contrastive Language-Image Pre-training (CLIP): A specific, multi-modal model trained with contrastive learning to align visual and textual representations. Mentioned as an inspiration for the self-supervised training approach, but not directly used in the methodology.

10- Convolutional Neural Network (CNN): A type of deep neural network that is particularly effective for processing images. CNNs use convolutional layers to automatically learn hierarchical features from raw pixel data. ResNet-50 is the CNN backbone used in SO-DRCNN.

11- Davies-Bouldin Index (DBI): A metric used to evaluate the quality of clustering algorithms (like k-means used for BoVW vocabulary construction). Lower DBI values indicate better clustering, with compact and well-separated clusters.

12- Deep Convolutional Neural Network (DCNN/DRCNN): A CNN with many layers, enabling the learning of complex, hierarchical features. SO-DRCNN is a specific type of DCNN used in this thesis.

13- DeepRec Convolutional Neural Network (DRCNN): Refers to the deep network architecture used in SO-DRCNN, incorporating a recurrent component and spatial pyramid modules.

14- Dimensionality Reduction: The process of reducing the number of dimensions (features) in a dataset while preserving important information. PCA is used for dimensionality reduction in this thesis.

15- Discrete Wavelet Transform (DWT): A signal processing technique that decomposes an image into different frequency sub-bands. Mentioned in the literature review, but not directly used in the core methodology.

16- Elasticsearch: A distributed search and analytics engine used in this thesis for efficient indexing and retrieval of image feature vectors. It enables fast similarity search on high-dimensional data.

17- Embedding Layer: The final fully connected layer in the SO-DRCNN architecture that outputs the visual embedding vector. It transforms the fused and processed features into a compact representation suitable for CBIR.

18- Embedding Space: A vector space where images are represented by their feature vectors (embeddings). In this thesis, the goal is to learn an embedding space where distance corresponds to semantic similarity.

19- Euclidean Distance: A common metric for measuring the distance between two vectors in a multi-dimensional space. Used in the contrastive loss function and potentially for similarity search.

20- Feature Fusion: The process of combining multiple feature representations (e.g., CNN embeddings and handcrafted features) into a single, richer feature vector. This is a core component of the proposed methodology, implemented using a Siamese-trained Fusion Module.

21- Fusion Module: A neural network module, trained within the Siamese Network, that learns to combine CNN embeddings and handcrafted features. This is the key component that performs the Siamese-Driven Feature Fusion.

22- Global Color Histogram (GCH): A feature representation that quantifies the distribution of colors in an image, disregarding spatial information. Referred to as ICH (Inclusive Color Histogram) in this thesis.

23- Handcrafted Features: Image features that are designed by humans based on domain knowledge and intuition, rather than learned automatically from data. In this thesis, BoVW histograms with Ternion descriptors (HOG, ICH, SERC) are used as handcrafted features.

24- Histogram of Oriented Gradients (HOG): A handcrafted feature descriptor that captures the distribution of gradient orientations in localized portions of an image, representing shape and texture information. Part of the Ternion descriptor set.

25- ICH (Inclusive Color Histogram): The term used in this thesis for a global color histogram, computed over the entire image, that quantifies the distribution of colors. Part of the Ternion descriptor set.

26- Keypoint: A salient and stable point in an image, often associated with corners, edges, or other distinctive local features. ORB is used to detect keypoints in this thesis.

27- Mean Average Precision (mAP): A common metric for evaluating the performance of information retrieval systems, including CBIR. It measures the average precision of retrieval results across multiple queries.

28- Metric Learning: A machine learning approach that focuses on learning a distance metric or similarity function from data. The Siamese Network with contrastive loss is a form of metric learning.

29- Multi-Probe LSH: An enhanced version of Locality Sensitive Hashing (LSH) that improves search accuracy by probing multiple hash buckets. Mentioned in the context of SERC descriptor matching.

30- Natural Language Processing (NLP): A field of computer science focused on enabling computers to understand and process human language. Mentioned in the context of Text-Based Image Retrieval (TBIR) in the literature review.

31- ORB (Oriented FAST and Rotated BRIEF): A fast and rotation-invariant feature detector and descriptor. Used in this thesis for keypoint detection in the BoVW framework.

32- Pairwise Constraints: Training signals used in self-supervised learning, consisting of pairs of images labeled as "similar" (Can-Link) or "dissimilar" (Cannot-Link). Used to train the Siamese Network with contrastive loss.

33- Patch: A small, rectangular region of an image. Used in the Recurrent Patching Module of SO-DRCNN.

34- Principal Component Analysis (PCA): A dimensionality reduction technique that finds the principal components (directions of maximum variance) in a dataset and projects the data onto these components. Used in this thesis for dimensionality reduction of feature vectors.

35- Recurrent Patching Module: A component of the SO-DRCNN architecture that processes image patches sequentially using a Bi-LSTM network to capture spatial context.

36- Region-of-Interest (ROI): A specific region within an image that is of particular interest for analysis or processing. Not directly used in your core methodology, but mentioned in the literature review.

37- ResNet-50: A deep convolutional neural network architecture (50 layers) that uses residual connections to enable effective training of very deep networks. Used as the pre-trained CNN backbone in SO-DRCNN.

38- Self-Supervised Learning: A machine learning paradigm where a model is trained on unlabeled data by generating its own supervisory signals from the data itself. The Auto-Embedder framework with Siamese Network and contrastive loss is a form of self-supervised learning.

39- SERC (Slanting Express Revolves Concise): A handcrafted feature descriptor designed to capture edges and structural patterns. Part of the Ternion descriptor set.

40- Siamese Network: A neural network architecture consisting of two (or more) identical subnetworks (twins) that share weights. Siamese networks are used to learn similarity metrics by comparing the outputs of the twins for pairs of inputs. Used in this thesis to train the Fusion Module for feature fusion.

41- Similarity Matching: The process of comparing feature vectors to find images that are similar to a query image. The core task in CBIR.

42- Smooth L1 Loss (Huber Loss): A loss function that combines the properties of L1 and L2 loss, making it more robust to outliers. Mentioned in the context of SO-DRCNN training.

43- Spatial Pyramid Pooling (SPP): A technique used in CNNs to capture multi-scale information by pooling feature maps at different spatial resolutions. Part of the SO-DRCNN architecture.

44- SO-DRCNN (Self-Optimizing DeepRec Convolutional Neural Network): The proposed deep learning architecture for CBIR, combining a pre-trained ResNet-

50 backbone with Recurrent Patching, SPP/ASPP, and Attention modules, and trained using a Siamese Network with contrastive loss.

45- Ternion Paradigm: The combination of HOG, ICH, and SERC descriptors used for handcrafted feature extraction in this thesis.

46- Text-Based Image Retrieval (TBIR): A traditional approach to image retrieval that relies on textual annotations or metadata associated with images. Contrasted with CBIR in the literature review.

47- Visual Vocabulary: In the BoVW framework, a set of representative local feature patterns (cluster centroids) learned by clustering a large collection of local descriptors. Used to create BoVW histograms.

Key Terms: Content-Based Image Retrieval (CBIR), Text-Based Image Retrieval (TBIR), Self-Supervised Learning, Feature Extraction, Elasticsearch, Semantic Gap, Embedding Space, Deep Learning, Self-Optimization.

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Dr. Chen, for his invaluable guidance and continued support throughout the course of this research.

I am also thankful to Dr. Fan Jiang for his insightful feedback and encouragement.

I extend my deepest appreciation to my family, to my beloved mother, and in loving memory of my late father, whose unwavering love, strength, and support gave me the courage to begin this journey.

Chapter 1

Introduction

1.1. Motivation

Content-Based Image Retrieval (CBIR) has become increasingly important across numerous fields, from medical imaging and architecture to crime prevention and geographic information systems. However, existing CBIR systems often face significant challenges, including limited retrieval accuracy, high computational demands, and a strong reliance on manually labeled data. A key limitation is the semantic gap: the discrepancy between low-level image features easily extracted by computers and the high-level semantic concepts that humans use to understand and judge image similarity (Smeulders et al., 2000). Furthermore, the need for extensive manual data labeling creates a bottleneck, hindering the scalability and adaptability of CBIR systems to new datasets and domains (Datta et al., 2008).

This thesis addresses these challenges by proposing a novel Self-Optimizing DeepRec Convolutional Neural Network (SO-DRCNN) framework for CBIR, enhanced with Siamese-Driven Feature Fusion. Our approach makes the following key contributions:

Hybrid Feature Representation: We integrate advanced feature extraction techniques, including the Ternion Paradigm (HOG, ICH, and the novel SERC descriptor), to create a robust and multi-faceted image representation that combines both interpretable local visual cues and high-level semantic features learned by a deep CNN (LeCun et al., 2015). This hybrid approach aims to bridge the semantic gap by leveraging the strengths of both handcrafted and learned features.

Self-Supervised Learning: We employ a self-supervised Siamese network architecture (Hadsell et al., 2006), trained with a contrastive loss function (Chopra et al., 2005), to learn discriminative image embeddings without relying on manually labeled data. This addresses the labeling bottleneck and enables the system to adapt more easily to new datasets. This approach builds upon recent advancements in self-supervised representation learning (Chen at Ai., 2020).

Siamese-Driven Feature Fusion: We introduce a novel feature fusion strategy where a Fusion Module, trained within the Siamese network, learns to adaptively combine the handcrafted features and deep CNN embeddings, optimizing the fused representation for semantic similarity. This data-driven fusion approach goes beyond simple concatenation or fixed-weight combinations, allowing for a more nuanced and effective integration of heterogeneous feature modalities.

Scalable Retrieval: We integrate our system with Elasticsearch (Gormley & Tong, 2015) to enable efficient and scalable image retrieval from large databases, addressing the practical challenges of real-world CBIR applications.

1.2. Statement of Purpose

Content-Based Image Retrieval has emerged as a robust alternative to text-based retrieval methods, which often rely on keyword-driven annotations and may overlook the full complexity of image content. By analyzing intrinsic visual characteristics—such as texture, shape, and color—CBIR systems automatically compare a user-provided query image to database items that share similar features. Despite these benefits, a major hurdle is the semantic gap: the mismatch between low-level descriptors (e.g., color histograms) and the high-level concepts that users perceive (Vo et al., 2021).

To address this gap, the present work integrates advanced feature extraction approaches—namely Slanting Express Revolves Concise, Inclusive Color Histogram, and Histogram of Oriented Gradients —with a Self-Optimization Deeprec Convolutional Neural Network. By combining these methods, the research aims to enhance retrieval accuracy, reduce labeling burdens, and capture nuanced image details. Building on three decades of CBIR advancements, this thesis examines the state-of-the-art in feature modeling, similarity metrics, and machine learning strategies, ultimately seeking to improve real-world image retrieval across large, heterogeneous datasets.

1.3. Research Objectives

This research aims to significantly enhance content-based image retrieval by overcoming persistent limitations of traditional retrieval methods, such as reliance on textual annotations, inability to fully capture semantic meaning, and inefficient scalability. The goal is to create an intuitive, robust, and scalable CBIR framework capable of accurately interpreting visual content within extensive image repositories. Specifically, this research sets out to:

1. Identify and Address Limitations in Current Image Retrieval Approaches

Systematically analyze current retrieval methodologies to understand critical weaknesses such as the semantic gap (disconnect between pixel-level representation and high-level concepts) and over-dependence on manual labeling. This analysis will cover traditional methods (e.g., color histograms), conventional deep learning models (e.g., VGG-16, ResNet-50), and hybrid systems.

2. Develop a Self-Optimizing Neural Network (SO-DRCNN) for Autonomous Learning

Design and implement a novel neural architecture, the Self-Optimizing DeepRec Convolutional Neural Network (SO-DRCNN), which autonomously learns to identify visual patterns without explicit labeling. This network will integrate spatial recurrent networks (SRNs) for spatial reasoning and attention mechanisms to prioritize image features that are critical for accurate retrieval.

3. Implement an Advanced Multi-Descriptor Framework (Ternion Paradigm)

Combine three complementary feature descriptors to capture comprehensive image characteristics:

- HOG (Histogram of Oriented Gradients) for detecting edges and textures.
- ICH (Inclusive Color Histogram) for encoding global color distributions.
- SERC (Slanting Express Revolves Concise) for identifying distinctive structural patterns robust to rotations and transformations.

This combination aims to effectively bridge low-level visual features and high-level semantic understanding.

4. Quantitatively Evaluate the Proposed System's Performance

Evaluate the CBIR system rigorously against state-of-the-art benchmarks (e.g., CIFAR-10 dataset), with explicit performance goals:

- Accuracy: Achieve ≥95% Mean Average Precision (MAP).

- Efficiency: Retrieval responses in ≤0.5 seconds for databases exceeding one million images.
- Robustness: Maintain high retrieval accuracy under challenging conditions (e.g., noisy, rotated, cropped images).

5. Demonstrate Applicability Across Diverse Real-World Domains

Validate system usability without extensive retraining for multiple practical scenarios, including medical diagnostics (e.g., tumor detection in X-rays), e-commerce (e.g., fashion item retrieval), and surveillance. Optimize the architecture for deployment in both high-performance cloud environments and resource-limited mobile platforms.

6. Establish a Foundation for Future Research and Innovation

Provide comprehensive documentation and guidelines to enable further system enhancement, addressing limitations such as specialized-domain adaptation, ultra-large-scale database retrieval, and extensions to video-based or dynamic image retrieval scenarios.

1.4. Organization of Thesis

This thesis is structured into five chapters, systematically addressing the motivations, methodologies, experimental evaluations, and implications of advanced CBIR approaches:

Chapter 2: Background

This chapter introduces the foundational concepts of CBIR, discussing its significance, practical applications, and inherent challenges. It presents an overview of feature extraction, similarity matching, and advanced retrieval methodologies, establishing a

clear rationale for the research. Additionally, this chapter specifies the objectives, motivation, and anticipated contributions of the thesis.

Chapter 3: Literature Review

A comprehensive review of existing literature is conducted, exploring foundational methods and state-of-the-art advancements in CBIR. The chapter critically examines various feature extraction methods, region-based retrieval (RBIR), semantic-based retrieval techniques, and hybrid multimodal systems. Emphasis is placed on identifying the limitations of current methodologies, thereby clearly delineating how the thesis advances the field through novel contributions.

Chapter 4: Methodology

This chapter details the novel methodologies proposed in this research. It systematically describes the feature extraction processes, data augmentation techniques, and the integration of advanced deep learning architectures, specifically highlighting the proposed Self-Optimizing DeepRec Convolutional Neural Network (SO-DRCNN). It further elaborates on self-supervised training frameworks and indexing mechanisms, providing theoretical justifications for all methodological choices.

Chapter 5: Experimental Evaluation

Experimental validation of the proposed methodologies is presented, covering dataset descriptions, experimental setup, and detailed evaluation metrics such as Mean Average Precision (MAP), precision-recall analysis, and the Davies–Bouldin Index (DBI). Results are thoroughly examined through quantitative assessments, graphical visualizations, and comparative studies with established benchmarks, affirming the performance and effectiveness of the proposed systems.

Chapter 6: Conclusion and Future Work

The thesis concludes by summarizing the key research findings and contributions. It highlights the theoretical and practical implications of the developed methodologies and clearly outlines the limitations encountered during the research. Directions for future research are proposed, emphasizing potential enhancements and opportunities for advancing CBIR further.

Chapter 2

Background

2.1. Image Processing

Image processing is a computational discipline that involves the acquisition, enhancement, analysis, and retrieval of images using mathematical models and algorithms. It plays a fundamental role in computer vision, AI, medical imaging, and multimedia retrieval, enabling the extraction of meaningful information from visual data. Image processing techniques are designed to improve image quality, facilitate feature detection, and enable automated decision-making in scientific, industrial, and technological applications (Gonzalez & Woods, 2018).

Digital image processing consists of several key stages: image acquisition, preprocessing, feature extraction, segmentation, and recognition. Image acquisition involves capturing images using cameras, scanners, or remote sensing devices. Preprocessing enhances image quality through noise reduction, contrast adjustments, and edge enhancement to improve subsequent analysis (Jain et al., 1995). Feature extraction focuses on identifying key characteristics such as textures, edges, and color distributions, which are essential for classification and retrieval tasks. Segmentation partitions an image into meaningful regions, facilitating object recognition, medical diagnostics, and scene understanding (Szeliski, 2010).

Image processing techniques are broadly classified into spatial domain methods and frequency domain methods. Spatial domain methods operate directly on image pixels, applying transformations such as filtering, morphological operations, and histogram equalization. Frequency domain methods, on the other hand, transform images into the Fourier domain to analyze patterns, compress image data, and enhance specific features

(Pratt, 2007). Advances in machine learning and deep learning have led to the development of automated image processing models, significantly improving performance in tasks such as object detection, facial recognition, and CBIR (LeCun et al., 2015).

A key application of image processing is in image retrieval, where computational techniques enable the efficient searching and indexing of visual content. Traditional image retrieval systems often relied on Text-Based Image Retrieval, where images were annotated with metadata, captions, or textual descriptions to enable searchability. However, TBIR faced limitations due to manual annotation costs and semantic gaps (Datta et al., 2008; Smeulders et al., 2000). On the other hand , CBIR leverages computer vision and pattern recognition to analyze color, texture, and shape features, allowing for more precise and automated retrieval processes (Smeulders et al., 2000).

With the growing volume of digital images across domains such as medical imaging, remote sensing, security, and multimedia archiving, the role of advanced image processing techniques in retrieval, classification, and recognition continues to expand. Advances in deep learning, particularly through CNN and multimodal AI models, have significantly enhanced image retrieval accuracy by learning hierarchical feature representations. These methods reduce the semantic gap between low-level visual features (e.g., color, texture) and high-level human interpretation (e.g., object categories, contextual meaning) by capturing semantically meaningful patterns directly from data (He et al., 2016; Radford et al., 2021). While challenges persist in abstract or fine-grained retrieval tasks, modern AI-driven systems outperform classical methods in aligning machine-extracted features with human perception (Krizhevsky et al., 2012; Smeulders et al., 2000).

Ongoing research in deep learning, CNNs, and multimodal fusion is further driving progress in intelligent image analysis and retrieval systems (Goodfellow et al., 2016). These image processing advancements have laid the foundation for modern Information Retrieval systems, enabling more efficient indexing, classification, and retrieval of images. The following section explores specific retrieval strategies that leverage these image processing techniques, ranging from traditional TBIR to more advanced content-based and hybrid retrieval frameworks.

2.1.1. Image Processing Techniques

This section examines three primary categories of image processing techniques:

I. Image enhancement

II. Image segmentation

III. Object detection & recognition

which are foundational for extracting meaningful information and improving computational analysis.

I. Image Enhancement

Image enhancement modifies an image to increase its interpretability and visibility, optimizing it for computer vision applications, medical diagnostics, and feature extraction tasks (Pratt, 2007). These techniques work to reduce noise and improve image contrast. As a result, critical features become more prominent, making the images better suited for analysis and automated processing.

- Noise Reduction: Noise reduction techniques eliminate unwanted distortions, such as random intensity variations resulting from sensor limitations or environmental factors (Jain et al., 1995).
    - Median filtering: Suppresses salt-and-pepper noise while preserving edges.

- Gaussian filtering: Smooths intensity variations by applying a weighted average of neighboring pixels.
- Non-local Means filtering: Reduces noise by averaging similar pixel intensities across distant regions (Buades et al., 2005).

- Contrast Adjustment & Histogram Equalization: Contrast enhancement techniques expand an image's dynamic range, improving the visibility of subtle details, which is particularly beneficial for medical imaging, satellite imagery, and digital photography (Szeliski, 2010).
  - Histogram equalization: Redistributes pixel intensities to improve global contrast.
  - AHE: Enhances contrast in localized regions to emphasize finer details (Pizer et al., 1987).
- Edge Detection: Edge detection is crucial for feature extraction and object recognition, as it identifies boundaries and structural elements within an image (Canny, 1986).
  - Sobel operator: Detects edges based on intensity gradients.
  - Canny edge detector: Applies Gaussian smoothing and gradient detection to identify edges with minimal noise interference.
  - Laplacian operator: Highlights regions of rapid intensity change, aiding in object boundary detection.

II. Image Segmentation

Image segmentation partitions an image into distinct regions corresponding to objects or areas of interest, facilitating medical diagnostics, autonomous navigation, and scene analysis (Shi & Malik, 2000). This technique enables more effective object recognition, classification, and retrieval by isolating relevant image components.

- Thresholding Techniques: Thresholding converts grayscale images into binary representations, distinguishing foreground objects from the background based on intensity variations (Otsu, 1979).
    - Otsu's method: Automatically selects an optimal threshold to maximize inter-class variance.
    - Adaptive thresholding: Adjusts the threshold dynamically across different image regions, accommodating variations in lighting conditions.
- Region-Based Segmentation: Region-based methods group pixels with similar characteristics to delineate meaningful structures. (Comaniciu, D., & Meer, P. , 2002).
    - Watershed segmentation: This method treats as a topographic surface and identifies object boundaries using gradient-based ridges.
    - Active Contour Models (Snakes): Employs energy minimization to refine object boundaries through iterative deformation (Kass et al., 1988).
- Deep Learning-Based Segmentation: Deep learning models have significantly advanced segmentation accuracy, particularly in biomedical imaging, autonomous vehicles, and satellite image analysis (Ronneberger et al., 2015).
    - U-Net: A CNN designed for precise biomedical image segmentation.
    - Mask R-CNN: Extends Faster R-CNN by incorporating pixel-wise instance segmentation for detecting multiple objects (K. He et al., 2017).

III. Object Detection & Recognition

Object detection identifies and classifies objects within an image, enabling applications in face recognition, autonomous navigation, surveillance, and image retrieval (Felzenszwalb et al., 2010). These methods fall into two categories: traditional feature-based approaches and Deep Learning-Based techniques.

- Traditional Object Detection Methods: Earlier methods relied on handcrafted features to detect objects based on predefined patterns (Dalal & Triggs, 2005).
    - Haar cascades: Utilizes edge and texture patterns for real-time face and object detection.
    - HOG: Captures gradient distributions to recognize shapes and contours.
- Deep Learning-Based Object Detection: Recent advancements in deep learning have led to more robust and scalable object detection frameworks (Redmon et al., 2016).
    - YOLO (You Only Look Once): Processes an entire image in a single pass, enabling efficient real-time detection.
    - Faster R-CNN: Enhances object detection accuracy by incorporating a region proposal network for improved bounding box predictions (Ren et al., 2016).

Image retrieval systems rely on specialized image processing techniques to extract, represent, and index visual features, allowing for efficient searching and matching of images in large databases. Unlike general image processing, where the goal is image enhancement or segmentation, image retrieval techniques focus on identifying distinctive image features and organizing them into structured representations for fast and accurate retrieval (Smeulders et al., 2000). This section explores:

I. Feature extraction

II. Deep Learning-Based Feature Extraction

III. Preprocessing for Large-Scale Image Retrieval

I. Feature extraction

Where visual elements such as color, texture, and shape are analyzed to create a numerical representation of an image (Datta et al., 2008). These features serve as a compact and discriminative description that allows retrieval systems to compare and rank image similarity efficiently.

- Color Features: Color is one of the most commonly used features in image retrieval, as it provides a straightforward way to differentiate images (Swain & Ballard, 1991). Color-based retrieval methods rely on statistical representations of pixel intensities rather than object recognition, making them useful for applications such as multimedia search and digital library indexing.
    - RGB histograms: Represent the distribution of red, green, and blue intensities in an image.
    - HSV histograms: Capture hue, saturation, and value, offering robustness to lighting variations.
- Texture Features: Texture describes the spatial arrangement of pixel intensities, enabling retrieval systems to differentiate surfaces and patterns that may not be easily distinguishable by color alone (Manjunath & Ma, 1996).
    - Local Binary Patterns (LBP): Encode local texture characteristics by thresholding neighborhood pixels.
    - Gabor filters: Analyze texture frequencies and orientations, making them useful for biomedical image retrieval and fingerprint recognition.

- Shape Features: Shape-based retrieval techniques are effective for identifying images containing specific objects or geometric patterns, particularly in biomedical and industrial applications (Zhang & Lu, 2002).
    - Contour descriptors: Extract object outlines to facilitate shape-based matching.
    - Fourier descriptors: Convert shape boundaries into frequency components for comparison.

Feature extraction methods allow images to be represented in high-dimensional feature spaces, where similarity measures such as Euclidean distance and cosine similarity are used for ranking retrieved images.

II. Deep Learning-Based Feature Extraction

Traditional feature extraction methods rely on handcrafted features, which may not always capture high-level semantic information. Recent advances in deep learning have significantly improved image retrieval by enabling models to automatically learn hierarchical feature representations from large datasets (LeCun et al., 2015).

CNNs for Feature Embeddings: CNNs have become the standard for image feature extraction, transforming raw pixel values into a structured feature vector (Krizhevsky et al., 2012).

- ResNet and VGGNet extract multi-layer feature embeddings, capturing texture, shape, and spatial structure.
- These embeddings are used in vector search engines for large-scale image retrieval.

- Self-Supervised Learning for Feature Representation: Recent developments in SSL allow models to learn meaningful feature representations without labeled data, making them ideal for scalable retrieval systems (Chen et Ai., 2020).

- SimCLR (Simple Contrastive Learning Representation) learns visual similarities using contrastive loss.
- MoCo (Momentum Contrast) stores large feature dictionaries for improved retrieval accuracy.
- DINO (Self-Distillation with No Labels) enhances feature quality for unsupervised retrieval applications.

- Vision Transformers (ViTs) in Image Retrieval: ViTs have emerged as an alternative to CNNs, processing images using self-attention mechanisms to capture long-range dependencies (Dosovitskiy et al., 2021).
    - Unlike CNNs, which extract local features, ViTs model entire image patches simultaneously, improving retrieval for complex scenes and fine-grained image categorization.

Deep Learning-Based feature extraction enables more accurate and semantically meaningful retrieval, reducing the reliance on manually engineered descriptors.

III. Preprocessing for Large-Scale Image Retrieval

Efficient image retrieval requires optimizing feature storage, indexing, and retrieval speed. Preprocessing techniques such as dimensionality reduction, feature indexing, and compression improve scalability and search efficiency in high-dimensional feature spaces (Jégou et al., 2011).

- Dimensionality Reduction Techniques: High-dimensional feature representations can be computationally expensive. Dimensionality reduction improves retrieval efficiency while preserving important feature details.
    - Principal Component Analysis (PCA): Reduces feature dimensions by identifying principal components in the data.

    - t-SNE (t-Distributed Stochastic Neighbor Embedding): Projects high-dimensional data into a lower-dimensional space for visualization and clustering.
- Feature Indexing Methods: For real-time image retrieval, indexing techniques allow faster nearest neighbor searches in large-scale databases (Muja & Lowe, 2009).
    - KD-Trees: Partition feature space into hierarchical subregions to accelerate similarity searches.
    - Hashing methods: Convert feature vectors into compact binary representations for fast lookup.
    - Approximate Nearest Neighbor (ANN) search: Balances search accuracy and computational efficiency for large-scale datasets.
- Compression for Efficient Storage & Retrieval: Storage-efficient retrieval systems require compression techniques to reduce memory footprint while preserving retrieval accuracy.
    - Vector quantization compresses feature vectors while maintaining similarity relationships.
    - Product Quantization (PQ) enables fast approximate nearest neighbor search in high-dimensional feature spaces.

These preprocessing techniques ensure that image retrieval systems remain scalable and computationally efficient, allowing for real-time search and indexing in extensive datasets.

2.1.2. Relationship Between Image Processing and Image Retrieval

Image processing serves as the foundation for image retrieval, enabling the extraction, representation, and indexing of visual features that facilitate efficient search and retrieval

operations. While general image processing techniques focus on enhancement, segmentation, and object detection, their integration into image retrieval ensures more accurate and meaningful search results (Smeulders et al., 2000). This section explores how feature extraction, image enhancement, segmentation, and object detection contribute to various retrieval approaches.

- **Feature Extraction**

For instance, CBIR systems, in particular, rely on feature extraction, enhancement, and object recognition to improve retrieval accuracy and search efficiency (Datta et al., 2008).

CBIR, converts images into structured representations based on color, texture, shape, and deep learning-based embeddings.

Depending on retrieval requirements, CBIR systems use different feature extraction approaches based on the task. Traditional methods, such as color histograms, Local Binary Patterns (LBP), and shape descriptors, provide handcrafted feature representations that capture essential visual attributes (Swain & Ballard, 1991). In contrast, Deep Learning-Based methods utilize CNNs to generate more robust feature embeddings, with architectures such as ResNet and VGGNet, as well as self-supervised learning models like SimCLR and MoCo, which further improve retrieval performance without relying on labeled data (LeCun et al., 2015). By integrating feature extraction techniques, CBIR systems significantly enhance retrieval accuracy, ensuring that query images return visually similar results based on content rather than textual descriptions, making them particularly effective in multimedia search, medical imaging, and large-scale visual databases.

- **Image Enhancement**

Image enhancement plays a vital role in CBIR by improving feature distinctiveness in low-quality or noisy datasets, ensuring accurate feature extraction and similarity matching (Pratt, 2007). Poor image quality can degrade retrieval accuracy, as retrieval systems rely on well-defined features to perform efficient indexing and ranking. Techniques such as noise reduction, contrast adjustment, and edge sharpening improve clarity, texture representation, and boundary detection, all of which enhance retrieval performance (Buades et al., 2005; Szeliski, 2010). In medical image retrieval, contrast enhancement improves tumor and lesion visibility, aiding in clinically relevant case matching (Ronneberger et al., 2015). Similarly, in historical document retrieval, noise removal and sharpening refine text extraction, facilitating accurate archival searches. By ensuring that images contain well-preserved, high-contrast visual features, enhancement techniques optimize retrieval accuracy, allowing retrieval systems to perform more precise ranking and similarity comparisons.

- **Segmentation**

Segmentation is a key image processing step that partitions an image into meaningful regions (Shi & Malik, 2000). By isolating specific objects or areas, segmentation bridges the gap between raw pixel data and higher-level retrieval tasks. In region-based retrieval, only these segmented regions are compared, filtering out irrelevant background and improving precision—a benefit particularly evident in medical imaging, where U-Net-based segmentation (Ronneberger et al., 2015) helps identify pathologies for targeted comparisons, and in satellite analysis, where isolating regions of interest (e.g., deforestation zones) refines query relevance. Thus, effective segmentation not only enhances object-based retrieval accuracy but also demonstrates how image processing techniques are fundamentally tied to more advanced image retrieval strategies.

- **Object Detection**

Similarly, Object detection localizes and classifies objects within an image, allowing retrieval systems to focus on relevant content rather than irrelevant background details. Models such as YOLO, Faster R-CNN, and Mask R-CNN detect and segment objects with high accuracy, eliminating the need for manual cropping (Redmon et al., 2016; Ren et al., 2016; He et al., 2017). By generating precise bounding boxes or masks, these algorithms streamline the retrieval process—rather than comparing entire scenes, the system compares only the detected objects. Pervasive use case example: In automotive image retrieval, an object detection model can identify vehicles within complex street scenes and classify them by make or model. This lets the retrieval system rank images based on specific car attributes instead of irrelevant background elements. In facial recognition, object detection isolates faces within crowded images, allowing the retrieval system to match identities more efficiently and accurately.

Through these targeted detections, object detection ensures more domain-specific and precise image retrieval, reducing computational overhead and improving user satisfaction.

scene.

2.2. Retrieval Strategies

Efficient image retrieval plays a critical role in academic research, digital archiving, medical imaging, surveillance, and multimedia applications. Various retrieval strategies have been developed to improve search accuracy, scalability, and relevance by leveraging different aspects of image representation (Datta et al., 2008; J. Z. Wang et al., 2001). These strategies range from traditional text-based retrieval to advanced AI-driven content-based and hybrid models.

2.2.1. Text-Based Image Retrieval (TBIR)

TBIR relies on textual metadata such as titles, descriptions, and manually assigned tags to search for images. This method is widely used in digital libraries and web search engines, where textual annotations are available. However, TBIR suffers from subjectivity and annotation inconsistencies, as different users may describe the same image differently, leading to mismatches in search results (Datta et al., 2008).

A distinct contribution of TBIR lies in its suitability for structured archives, such as museum databases and academic repositories, where detailed textual descriptions are readily available.

2.2.1.1. Challenges and Limitations

Despite its advantages, TBIR faces several challenges:

- Subjectivity: The subjective nature of textual descriptions can lead to inconsistent annotations, affecting retrieval accuracy (Goodrum, 2000).
- Scalability: Manual annotation of large image collections is not feasible. Automated techniques, although helpful, are not always accurate and can miss contextual nuances.

- Semantic Gap: The gap between textual descriptions and visual content impacts retrieval effectiveness. TBIR systems need to bridge this gap to improve accuracy (Smeulders et al., 2000).
- Language Variability: Differences in language, spelling, and phrasing affect TBIR system's effectiveness. Multilingual support is crucial for global applications (L.-J. Li & Fei-Fei, 2010).

2.2.1.2. Future Directions

Research in TBIR aims to address these challenges and improve retrieval accuracy. Key areas of focus include:

- Improved NLP Techniques: Advances in NLP can enhance automated annotation and semantic analysis, making TBIR systems more accurate and context-aware (Brown et al., 2020).
- Machine Learning: Incorporating machine learning can improve TBIR scalability and accuracy. These technologies can learn from user interactions, continually refining retrieval results (Khan et al., 2010).
- Multimodal Retrieval: Combining TBIR with other retrieval methods, like CBIR, leverages the strengths of both approaches. Multimodal retrieval systems use textual and visual features to enhance accuracy.
- User Interaction and Feedback: Incorporating user feedback into TBIR systems refines annotations and improves retrieval accuracy. Interactive systems that learn from user behavior are more effective.

While manual annotation provides high-quality, contextually rich image descriptions, its labor-intensive nature, high cost, time requirements, and issues with subjectivity and scalability present significant challenges. These limitations confirm the need for

integrating automated and hybrid approaches to enhance the efficiency, scalability, and consistency of Text-Based Image Retrieval (TBIR) systems.

2.2.2. Content-Based Image Retrieval

A CBIR approach replaces traditional text-driven searches with visual feature analysis. In this strategy, images are first preprocessed (e.g., normalized or filtered) to ensure consistent input quality. Next, feature extraction algorithms encode each image into a descriptive representation, capturing essential properties such as color, texture, or shape (Chopra et al., 2021). These representations are then indexed for rapid comparison.

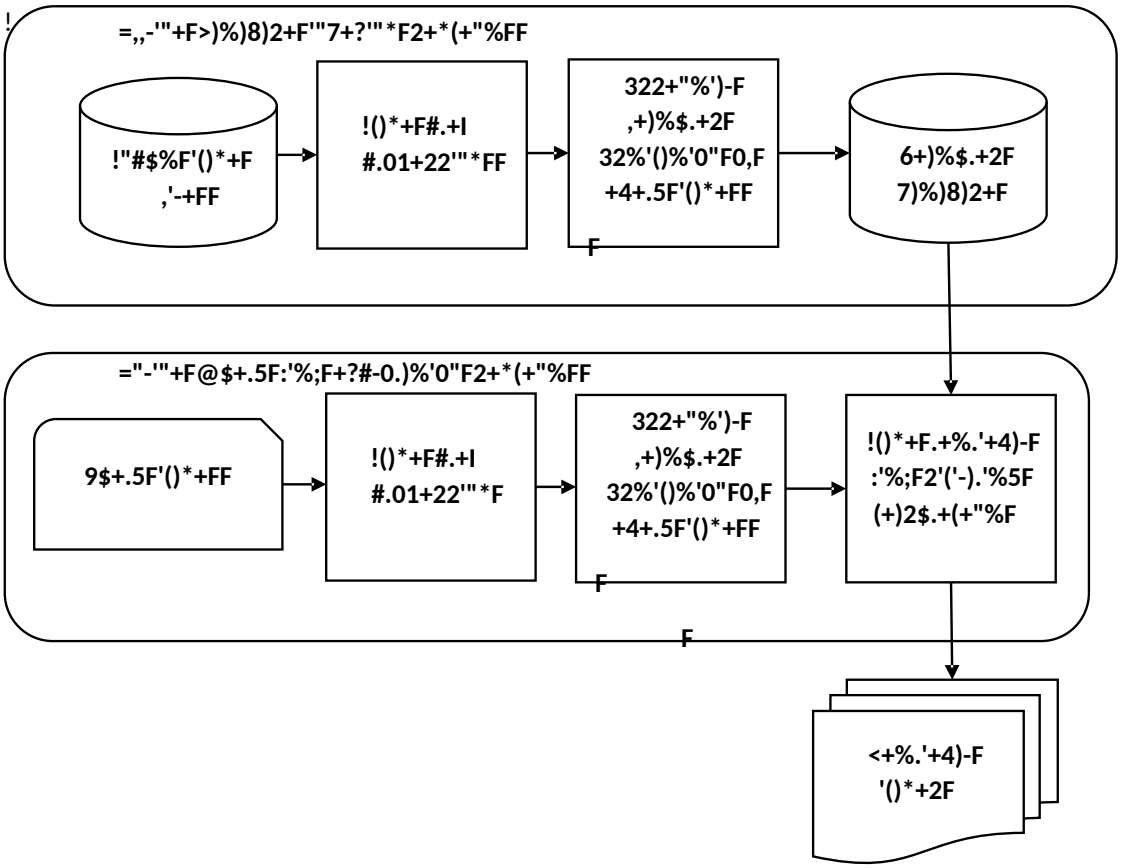


Figure 1- Common CBIR approach

When a query image is provided, the system repeats the feature extraction process to create a query representation, which is subsequently matched against the indexed database using similarity measures. Results are ranked based on their resemblance to the query, returning only the most visually similar images (Georgiou, 2021). By focusing on

the content itself, CBIR circumvents the need for comprehensive text labeling or manual annotations, resulting in a robust, scalable, and efficient retrieval solution.

CBIR eliminates TBIR's dependence on text by retrieving images by visual content, such as color histograms, textures, and spatial structures (Smeulders et al., 2000). CBIR systems extract these features and visual descriptors and compute similarity scores between a query image and database images (J. Z. Wang et al., 2001).

How CBIR Differs from TBIR?

- Uses visual analysis instead of textual annotations.
- Independent of language barriers – Useful in domains like biomedical imaging and forensic investigations, where textual descriptions are insufficient (Rui et al., 1999).

2.2.2.1. Challenges and Limitations

Despite its strengths, CBIR also faces a number of significant challenges:

- Semantic Gap: Low-level visual features (e.g., color, shape) often fail to capture high-level semantic concepts, creating a gap between the system's representation and the user's intent (Smeulders et al., 2000).
- High-Dimensional Feature Space: CBIR extracts feature vectors (e.g., deep CNN embeddings), which can be computationally expensive to index and compare, especially as databases grow (Datta et al., 2008).
- Computational Overhead: Advanced feature extraction (e.g., deep learning) boosts retrieval accuracy but imposes significant demands on processing power and storage (Krizhevsky et al., 2012).

Domain specificity: Handcrafted features or CNN-based descriptors may need to be tailored for specific domains (e.g., medical imaging vs. fashion), limiting the generality of a single CBIR system.

In summary, bridging the semantic gap, managing high-dimensional data, and balancing accuracy with computational efficiency are central hurdles for CBIR. Ongoing research focuses on improved feature extraction, intelligent indexing, and hybrid approaches (e.g., combining text and visual features) to address these limitations and better align retrieval results with user expectations.

2.2.2.2. Future Directions

Research in CBIR continues to tackle its core challenges, particularly the semantic gap and high computational demands. Key areas of focus include:

- Advanced Deep Learning Architectures: Building on CNN-based methods (Krizhevsky et al., 2012), new architectures (e.g., Vision Transformers) and self-supervised frameworks aim to extract richer, more semantically meaningful representations without extensive labeled data (Chen at Ai., 2020).
- Hybrid Feature Integration: Combining handcrafted features (e.g., color histograms) with deep descriptors can yield robust retrieval performance, especially in domain-specific contexts like medical or fashion (Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., & Jain, R. (2000).
- Efficient Indexing and Similarity Search: As image databases grow, approximate nearest-neighbor techniques and hash-based methods help reduce retrieval latency and memory usage (He et al., 2018). Ongoing work focuses on scalable indexing structures for high-dimensional embeddings.

- User-Centric Enhancements: Incorporating relevance feedback and interactive interfaces refine retrieval outcomes over time, aligning system results more closely with user intent (Torres & Reis, 2008). This iterative process mitigates the semantic gap by capturing contextual or personal preferences.
- Domain Adaptation & Transfer Learning: Tailoring CNN-based CBIR systems to specific domains (e.g., remote sensing, forensics) often requires fine-tuning or domain adaptation strategies that leverage pre-trained models (Doersch & Zisserman, 2017). Research explores how to adapt such models efficiently for niche applications.

2.3. IR Techniques

The major IR techniques for images can be broadly categorized into two main approaches:

I. TBIR

II. CBIR

2.3.1. TBIR

TBIR is already explained in IR Strategies; this section only focuses on its operational techniques. In TBIR, image retrieval is performed using structured metadata indexing, allowing images to be searched based on associated textual descriptions rather than their visual content. TBIR systems primarily utilize natural language processing (NLP) and information retrieval algorithms to improve query matching.

Advanced NLP techniques, AI-driven annotation, and ontology-based metadata expansion enhance search accuracy and relevance. This section presents the key techniques in TBIR.

2.3.1.1 Advanced TBIR Methodologies

I. Semantic Text Processing & Query Matching

Traditional keyword-based retrieval is limited when user queries do not match stored metadata exactly. Word embeddings, such as Word2Vec and BERT, transform words into high-dimensional vector representations, enabling semantic similarity-based retrieval (Mikolov et al., 2013).

- Mathematical Model for Word Embeddings

Each word w in an image description is mapped to a vector $v_!$ in an n − dimensional ($v_! \in R^{"}$) space:

$$v_! = W \cdot w + b$$

where:

- $W \in R^{"\times |\%|}$ is a learned weight matrix encoding semantic relationships (analogous to an embedding lookup table),
- $w \in R^{|\%|}$ is a one-hot encoded vector representing the word (in a) vocabulary (, V)
- $b \in R^{"}$ is a bias term refining the vector representation.

Key Insight: This step converts discrete words into continuous vectors, enabling machines to interpret linguistic semantics geometrically (e.g., ”cat” and ”kitten” are close in the embedding pace).

For multi-word queries, an overall query vector $v_{\&}$ is computed as the mean of individual word embeddings:

$$v_{\&} = \frac{1}{n} \sum_{)*+}^{(} 2v \quad ' \; !$$

The similarity score between a query Q and an image annotation (represented as $v_{,}$) is computed using cosine similarity:

$$Sim(Q, I) = \frac{v_{\&} \cdot v_{,}}{|v_{\&}||v_{,}|}$$

Values range from −1 (dissimilar) to 1 (identical). High scores indicate semantic alignment between the query and image annotation.

Word embeddings improve query relevance by allowing concept-based matching rather than exact word searches. (Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003)

II. Sentiment Analysis

Sentiment analysis involves determining the sentiment expressed in the textual annotations. Understanding the emotional context of descriptions can add another layer of relevance in TBIR systems.

- VADER (Valence Aware Dictionary and sEntiment Reasoner): VADER is a lexicon and rule-based sentiment analysis tool that is particularly attuned to sentiments expressed in social media. It helps understand the emotional tone of the annotations Hutto, C. J., & Gilbert, E. E. (2014)
- TextBlob: A simple library for processing textual data, TextBlob provides tools for common NLP tasks, including sentiment analysis. It captures the sentiment of the textual descriptions, useful for more nuanced image retrieval (Loria, 2020).

III. Automated Image Annotation Using AI

Manual image annotation is time-consuming. AI-powered models automatically generate textual metadata by analyzing image content. These models integrate:

- CNNs for image feature extraction.
- Recurrent Neural Networks (RNNs) or Transformer Models for text sequence generation.

- Mathematical Model for Image Captioning:

  The AI model assigns a caption T to an image I by maximizing the probability of words in T, given I:

$$T^* = \arg\max P(T|I)$$

  Where:

  - $P(T|I)$ represents the probability of generating a description T given the image I.

  - The probability is computed using a sequence prediction model:

$$P(T|I) = \prod_{t=1}^{n} P(w_t|w_1, \ldots, w_{t-1}, I)$$

- T = The sequence of words forming the image caption.
- I = The image input, which provides visual features extracted by a CNN

  $P(w_/|w_+, \dots, w_{/0+}, I)$ = The probability of generating the next word $w_/$, given:
- Previous words in the sequence $P(w_2|w_+, w_3, \dots, w_{20+}, I)$
- The image representation $\boldsymbol{I}$, First Word Prediction (Based on the Image I): $P(w_+|I)$, Second Word Prediction (Based on $\boldsymbol{w_1}$ Image I) : $P(w_3|w_+, I)$
- This continues until the model generates all m words of the caption.

  This approach is used in Recurrent Neural Networks (RNNs) and Transformer-based models for image captioning, where the probability of each word is computed sequentially based on previous words and the image features.

IV. Named Entity Recognition (NER)

NER involves identifying and classifying proper nouns in the text into predefined categories such as names of people, organizations, and locations. NER enhances the precision of TBIR by extracting specific entities from the annotations.

- SpaCy: An open-source library for advanced natural language processing in Python, SpaCy offers pre-trained models for NER. It effectively extracts entities from image annotations, improving the retrieval process.
- Stanford NER: This tool uses statistical models trained on labeled data to provide reliable entity recognition. Its application in TBIR helps accurately identify and classify entities within the textual descriptions.

V. Metadata Expansion via Ontologies

To enhance retrieval accuracy, TBIR systems use domain-specific ontologies (e.g., MeSH for medical images; U.S. National Library of Medicine, n.d.) to expand queries with synonyms and related terms. This ensures that searches retrieve all relevant images, even if the user does not specify the exact metadata terms. For example, "X-ray" and "Radiograph" are considered synonymous. This broader search vocabulary helps ensure that all relevant images are retrieved, even when a user does not use the exact technical terminology.

- Mathematical Model for Query Expansion:

An ontology can be represented as a graph where nodes correspond to concepts (e.g., "X-ray," "Radiograph") and edges define relationships between terms (e.g., "X-ray → Radiograph (synonym)"). For a given query Q_+, the expanded query Q^5 includes all semantically related words:

$$Q^5 = Q \cup w^5 \mid \exists (w, w^5) \in E$$

Where:

- w^5 is a related word from the ontology.

- Term Weighting in Expanded Queries (TF-IDF Representation)

Each term in Q' is assigned a relevance weight using TF-IDF (Term Frequency–Inverse Document Frequency) (Salton & McGill, 1983):

$$TF - \boldsymbol{IDF}(t, d) = \boldsymbol{TF}(t, d) \times \boldsymbol{log}\left(\frac{N}{\boldsymbol{DF}(t)}\right)$$

where:

- $TF(t, d)$ = frequency of term t in document d.
- DF(t) = number of documents containing t
- N = total number of documents in the collection.

TF-IDF ensures that terms with high descriptive power (i.e., those that appear rarely across the collection but are frequent in a specific document) receive higher weights. (Salton & Buckley, 1988; Manning et al., 2008)

Ontology-driven query expansion can significantly enhance retrieval performance in text-based image retrieval (TBIR). By incorporating synonymous or related terms, expanded queries cover concept-dense domains more comprehensively, such as those found in medical imaging. As a result, recall is elevated due to broader keyword coverage. Precision often remains stable—or even improves—thanks to TF-IDF weighting, which filters out irrelevant results by emphasizing contextually critical terms. Empirical findings (X. Li et al., 2021) indicate that such expansions accommodate nuanced terminological variations without degrading overall result quality.

Ontologies also function as knowledge graphs that capture key semantic relationships within specialized domains. Leveraging these connections helps unify diverse references and mitigates the variability of user queries and metadata descriptions. At the same time, TF-IDF weighting ensures that distinctive terms retain prominence, balancing comprehensive retrieval with robust relevance ranking. Consequently, combining ontology-based query expansion with TF-IDF weighting is shown to both increase recall—through broader term sets—and sustain precision—via the strategic weighting of essential concepts.

2.3.1.2 Application Examples

This section explores real-world applications of TBIR across various domains, categorized by the primary approaches used. Each example highlights the techniques employed, the specific application, and its significance, supported by references to high-impact academic literature.

- **Textual Query-Based Retrieval**

Textual query-based retrieval relies on natural language queries and structured metadata to retrieve images. This approach is widely used in domains where user-friendly and intuitive search is essential.

In e-commerce, online marketplaces like Amazon and eBay require efficient retrieval of product images based on user queries. Textual query-based retrieval uses product descriptions and metadata to achieve this. Techniques include Natural Language Processing (NLP) models like BERT to understand user queries (e.g., "red leather handbag") and metadata tagging to annotate product images with attributes like color, material, and style (Devlin et al., 2018; W. Liu et al., 2011). For example, a user searching for "red leather handbag" retrieves relevant product images, enhancing user experience and increasing sales conversion rates.

On social media platforms like Instagram and Flickr, users rely on user-generated content and tags for image retrieval. Textual query-based retrieval uses hashtags and captions to improve search accuracy. Techniques include user-generated tags (e.g., "#sunset," "mountain hiking") and NLP for semantic analysis of captions (C. Hu, 2021; Z. Wang et al., 2018). For instance, a user searching for "#sunset" retrieves images tagged with this keyword, improving content discoverability and user engagement.

- **Semantic and Ontology-Based Retrieval**

Semantic and ontology-based retrieval bridges the semantic gap by mapping textual queries to high-level concepts using ontologies and knowledge graphs. This approach is critical in domains requiring domain-specific knowledge.

In healthcare, radiologists and researchers need accurate retrieval of medical images for diagnostics and analysis. Semantic-based retrieval uses medical ontologies and knowledge graphs to achieve this. Techniques include NLP for medical terms using

models like BioBERT to extract terms from queries (e.g., "early-stage lung cancer") and ontology development using standards like SNOMED CT to standardize terms and relationships (Lee et al., 2020; Rui et al., 1999). For example, a radiologist querying "early-stage lung cancer in non-smokers" retrieves relevant X-rays, improving diagnostic accuracy and workflow efficiency.

At Biodiversity Research, researchers and conservationists need to retrieve images of species for biodiversity studies. Semantic-based retrieval uses domain-specific ontologies to link species names to related concepts (e.g., habitats, conservation status). Techniques include NLP for species identification and domain-specific ontologies (Zhang, 2021). For instance, a researcher querying "endangered bird species in the Amazon rainforest" retrieves relevant images, supporting biodiversity research and conservation efforts.

- **Multimodal Retrieval (Text + Visual Features)**

Multimodal retrieval combines textual queries with visual features to retrieve images using multimodal embeddings. This approach is ideal for domains requiring context-aware and semantically rich results.

The Search Engines, platforms like Google Images and Bing Visual Search require accurate retrieval of images based on complex queries. Multimodal retrieval uses text and visual features to achieve this. Techniques include multimodal embeddings using models like CLIP to align textual and visual representations and visual feature extraction using CNNs or Vision Transformers (ViTs) (He et al., 2017; Radford et al., 2021). For example, a user querying "red dress with floral patterns" retrieves visually similar images, enhancing retrieval accuracy and user satisfaction.

In Advertising, Advertisers need to retrieve images for campaigns based on descriptive queries. Multimodal retrieval uses text and visual features to achieve this. Techniques

include multimodal embeddings to align textual queries with visual features and relevance feedback to iteratively refine results based on user input (Frome et al., 2013; W. Wang et al., 2021). For instance, an advertiser querying "diverse team collaborating in a modern office" retrieves relevant stock photos, enabling targeted and contextually relevant advertising.

2.3.2. CBIR

Since CBIR has already been explained in the IR Strategies section 1.3.2, this section focuses specifically on its operational techniques, such as feature extraction, similarity measurement, and indexing methods.

CBIR is a technique that searches for images by analyzing visual features (color, texture, shape, etc.) rather than relying on textual descriptions. Each image is mapped to a feature vector, and retrieval is performed by comparing these vectors with a similarity measure (e.g., Euclidean distance, Earth Mover's Distance). By focusing on the image content itself, CBIR can uncover relevant results even when textual labels are absent or incomplete. The following subsections introduce the key formulas for feature representation, discuss common distance metrics, and highlight both the advantages of CBIR (e.g., metadata independence) and its challenges (such as bridging the semantic gap).

2.3.2.1 Advanced CBIR Methodologies

I. Feature-Based Retrieval

Feature-based retrieval is a specialized subset of CBIR that extracts and compares mathematical feature descriptors instead of raw pixel-based properties (Swain & Ballard, 1991). Unlike general CBIR, which may rely on simple color histograms, feature-based retrieval employs advanced descriptors such as:

- Scale-Invariant Feature Transform (SIFT) – Detects stable key points for object matching (Lowe, 1999).
- Histogram of Oriented Gradients (HOG) – Useful in object detection and facial recognition (Dalal & Triggs, 2005).

Unique Contribution: Feature-based retrieval is more precise than traditional CBIR since it can detect robust object features regardless of lighting, rotation, or scaling differences (Lowe, 1999).

II. Semantic-Based Image Retrieval

Semantic-based image retrieval bridges the semantic gap in traditional CBIR by inferring high-level conceptual meaning rather than relying solely on low-level features like color histograms and texture patterns. Unlike classical CBIR, which matches images based primarily on pixel-derived properties, semantic-based retrieval employs:

- Deep Learning: Deep neural networks (e.g., ResNet, Vision Transformers) extract hierarchical features that capture objects, scenes, and contextual relationships (He et al., 2016).
- Ontological Knowledge: Domain-specific ontologies (e.g., WordNet, RadLex) enable query expansion by incorporating synonymous and related terms, ensuring broader and more precise retrieval (Miller, 1995).
- Cross-Modal Alignment: Models such as CLIP align visual and textual semantics, allowing natural language queries (e.g., "beach sunset") to retrieve contextually relevant images (Radford et al., 2021).

III. Reverse Image Search

Reverse image search is a specialized application of CBIR that focuses on instance-level retrieval—identifying near-duplicate or highly similar images to a query image. Unlike traditional CBIR systems, which often retrieve semantically related images (e.g., "beaches" for a "sunset" query), reverse image search prioritizes visual similarity at the pixel or feature level. Key Differentiators:

- Focus on Near-Duplicates: Reverse image search targets near-identical matches (e.g., cropped, resized, or slightly modified versions of the query image). Example: Detecting copyrighted images or identifying fake social media profiles using facial recognition.
- Technology: While traditional CBIR relies on handcrafted features (e.g., color histograms, texture descriptors), modern reverse image search systems use deep CNNs to generate high-dimensional embeddings that capture fine-grained visual patterns (Krizhevsky et al., 2012).Example: Google's Reverse Image Search employs CNN architectures like Inception-v3 (Szegedy et al., 2016) to encode images into feature vectors for similarity matching.
- Applications: Copyright enforcement (e.g., identifying unauthorized image use), Plagiarism detection in academic/creative work, Fact-checking by tracing image origins (e.g., debunking misinformation).

IV. Region-Based Image Retrieval

RBIR focuses on retrieving images based on specific localized regions rather than analysing the entire image (Rui et al., 1999). Used in medical imaging – A system searching for lung tumors focuses only on the lung region rather than matching full-body X-rays (J. Z. Wang et al., 2001).

Unlike object detection, which localizes and labels objects (e.g., tumors), RBIR retrieves images with regions that are visually or semantically similar to a query region. For example, a radiologist can select a lung nodule in a CT scan, and RBIR will retrieve studies with analogous nodules, leveraging texture and shape features rather than object labels (Litjens et al., 2017).

V. Sketch-Based Image Retrieval

SBIR enables users to query image databases using freehand sketches, retrieving images that align with the geometric structure and spatial layout of the sketch. Unlike CBIR, which relies on low-level features like color and texture, SBIR prioritizes structural similarity (e.g., edges, contours, shapes), making it ideal for applications requiring abstract or conceptual matching (Eitz et al., 2012).Distinct from CBIR:

- Instead of relying on color or texture features, SBIR matches geometric properties.

VI. Relevance Feedback Mechanisms

Relevance feedback improves retrieval by iteratively refining search results based on user input (Torres & Reis, 2008).

Example: A user searching for “wildlife photography” can mark relevant results, prompting the system to improve subsequent searches dynamically.

Distinct Contribution:

- Unlike fixed retrieval models, feedback-based retrieval adapts over time, improving personalized search.

VII. Hybrid Approaches

Hybrid retrieval combines textual, visual, and semantic features to optimize accuracy (Bose et al., 2015). Multimodal search engines use hybrid retrieval by integrating keyword search with CBIR-based visual filtering.

Distinct from Other Methods:

- Rather than relying on a single approach, hybrid retrieval balances multiple strategies, ensuring adaptability across different datasets.

2.3.2.2 Application Examples

There are several academically framed application examples that demonstrate the use of advanced CBIR methodologies:

- **Application Examples of CBIR Feature‐Based Retrieval**

In forensic image analysis, advanced feature‐based CBIR systems have proven indispensable for matching and retrieving near-duplicate images from extensive digital evidence databases. These systems rely on robust local feature descriptors that capture invariant image characteristics despite changes in scale, rotation, or illumination.

For example, a forensic tool may employ the following techniques:

- Scale-Invariant Feature Transform (SIFT):

  SIFT extracts distinctive keypoints and computes high-dimensional descriptors that remain stable under affine transformations. In forensic applications, SIFT is used to identify and match local keypoints between a query image and database images, allowing investigators to retrieve evidence even if the images have been manipulated or captured under different conditions (Lowe, 1999; DOI: 10.1109/ICCV.1999.790410).

  - Histogram of Oriented Gradients (HOG): Complementing SIFT, HOG captures the distribution of edge orientations within localized regions, thereby providing structural information about the objects in an image. This descriptor is

particularly useful in scenarios where the overall shape and texture are critical for verification, such as matching faces or specific objects in crime scene images (Dalal & Triggs, 2005; DOI: 10.1109/CVPR.2005.177).

- Geometric Verification:After extracting features using SIFT and HOG, the system typically applies geometric verification techniques—such as Random Sample Consensus (RANSAC)—to eliminate false matches. This step ensures that only spatially consistent correspondences contribute to the final retrieval decision.

In a practical forensic scenario, an analyst submits a query image suspected to be a modified copy of illicit content. The system first extracts SIFT keypoints and HOG descriptors from the query image and the entire database. A nearest-neighbor search identifies candidate matches based on descriptor similarity. Subsequently, RANSAC* is used to verify the spatial consistency of the matched features. The final ranking of candidate images is determined by the number of inliers matches and the quality of the geometric transformation between images. This robust approach significantly narrows down the search space and helps investigators pinpoint images that are highly similar, even under variations due to cropping, rotation, or scale.

*RANSAC (Random Sample Consensus) is a robust model-fitting algorithm that repeatedly selects random data subsets to estimate parameters, discarding outliers to achieve reliable results even under significant noise. (Fischler, M. A., & Bolles, R. C. (1981).

- **Application of RBIR**

RBIR plays a critical role in domains where the retrieval task requires localized analysis of image content rather than global image features. In medical imaging, for example, precise retrieval of pathological regions—such as tumors or lesions—from a large

database of radiological scans is essential for accurate diagnosis and comparative analysis. A typical RBIR system in medical applications operates as follows:

- Advanced Segmentation: The system employs deep learning–based segmentation models, such as U-Net, to partition radiological images (e.g., CT or MRI scans) into meaningful regions. U-Net has been widely adopted due to its encoder–decoder architecture, which effectively captures both global context and fine-grained details (Ronneberger et al., 2015). This step isolates the regions of interest (ROIs) that may contain lesions or other abnormalities. ROI is a specified subset of an image—often defined by coordinates or masks—that highlights the critical area for focused analysis or processing, such as detecting objects or measuring localized features.
- Region-Specific Feature Extraction:Once the regions are segmented, region-specific features are computed. These features typically include texture descriptors (e.g., Local Binary Patterns) and shape-based metrics that characterize the morphology of the detected region. This targeted feature extraction helps to capture the essential properties of the pathology, reducing the influence of irrelevant background information.
- Object Detection and Localization:

  In addition to segmentation, object detection models such as Faster R-CNN can be integrated to further refine the localization of pathological areas. Faster R-CNN generates bounding boxes that highlight the precise locations of lesions, thereby complementing the segmentation process (Ren et al., 2016; DOI: 10.1109/TPAMI.2016.2577031).
- Similarity Matching and Ranking:

  The extracted region-based features are then used to perform similarity matching across a database of segmented images. The system computes a similarity score

between the query ROI and each candidate ROI in the database, ranking the images such that those with the most similar pathological features appear at the top of the retrieval results.

In the real-world application, a radiologist submits a query image that contains a segmented lesion. The RBIR system first isolates the lesion using U-Net segmentation, extracts texture and shape descriptors from the lesion area, and applies Faster R-CNN to verify the region's localization. The system then computes similarity scores based on these region-specific features and ranks the images accordingly. As a result, the top retrieved images display lesions that share similar morphological characteristics, thus assisting the radiologist in making more informed diagnostic decisions.

In remote sensing, RBIR is similarly applied by segmenting satellite images to extract regions corresponding to specific land-cover types—such as deforested areas or urban expansions.

- **Application of Semantic□Based Retrieval**

In modern medical imaging, semantic□based retrieval systems have become essential for aiding clinicians in diagnosing and researching complex conditions. For example, consider a radiology retrieval system designed to assist in the diagnosis of early-stage lung cancer. In this application, the system leverages advanced CBIR methodologies that integrate deep learning, ontology-driven query expansion, and cross-modal alignment to overcome the limitations of traditional, low-level feature matching. Techniques and Workflow:

- Deep Feature Extraction: The system employs a deep CNN—for instance, a ResNet architecture (He et al., 2016)—pre-trained on large natural image datasets and fine-tuned on medical images. This network extracts high-level features that capture

complex visual patterns such as tissue textures, shapes of lung nodules, and contextual anatomical structures.

- Ontology-Driven Query Expansion: To bridge the semantic gap, the system integrates a domain-specific ontology (e.g., RadLex or a customized medical ontology) that encodes relationships between diagnostic terms. When a clinician queries the system using a term such as "early-stage lung cancer," the ontology expands the query to include related concepts such as "nodule," "mass," and "lesion" (Miller, 1995). This ensures that the retrieval process considers a broader, more clinically relevant set of features.
- Cross-Modal Alignment: Advanced models like CLIP (Radford et al., 2021) are employed to align textual descriptions from radiology reports with visual features extracted from images. This cross-modal alignment allows the system to interpret natural language queries in the context of the visual data, enhancing the semantic understanding of the query.
- Similarity Measurement and Ranking: The deep features are compared using cosine similarity, which, after normalization, effectively measures the angular distance between the query and database feature vectors. Images with the highest similarity scores are ranked at the top, ensuring that the retrieved images are semantically and visually aligned with the diagnostic query.

- **Application Example of Reverse Image Search**

Reverse image search is a specialized CBIR application designed to identify near-duplicate or highly similar images, a critical capability for enforcing copyright and detecting unauthorized image reuse. This approach leverages advanced CBIR methodologies to extract and compare robust visual features, even when images are modified by cropping, scaling, or color adjustments. Techniques and Workflow:

- Deep Feature Extraction: A state-of-the-art CNN such as Inception-v3 (Szegedy et al., 2016) or ResNet (He et al., 2016) is used to generate high-dimensional feature embeddings that capture fine-grained visual details and are robust to minor image variations. Additionally, neural codes as proposed by Babenko et al. (2014) can be extracted to represent image content in a compact form.
- Instance-Level Matching: The system computes similarity scores between the query image's embedding and those stored in the database using cosine similarity. For normalization and efficient matching, approximate nearest neighbor (ANN) search algorithms—such as those based on Hierarchical Navigable Small World (HNSW) graphs (Malkov & Yashunin, 2018)—are employed.
- Ranking: Images are ranked in descending order based on their cosine similarity scores, with the top$K$ matches displayed to the user. This ranking enables rapid identification of potential copyright infringements or unauthorized usage.

- **Application Example of Hybrid/Multimodal Retrieval**

Hybrid/multimodal retrieval systems integrate visual features with textual metadata to enhance product search in e-commerce platforms. In this application, a user may enter a natural language query (e.g., “red leather jacket with zipper”) while the system simultaneously analyzes product images using deep CNN. The retrieval pipeline fuses textual embeddings—generated by models such as BERT (Devlin et al., 2018)—with visual embeddings obtained from a state-of-the-art network (e.g., ResNet-50; He et al., 2016). An attention-based fusion module combines these complementary representations into a unified feature vector, which is then used to compute similarity scores via cosine similarity. The system ranks the products based on the joint relevance of visual appearance and descriptive text, enabling more accurate and context-aware recommendations. Key Techniques and Workflow:

- Textual Embedding: Text from product descriptions is processed using BERT to generate semantic embeddings that capture contextual information (Devlin et al., 2018).
- Visual Embedding: Images are encoded using a deep CNN, such as ResNet-50, to obtain robust visual features that capture fine-grained details (He et al., 2016).
- Fusion Strategy:An attention-based multimodal fusion mechanism—similar to approaches discussed by Ngiam et al. (2011) and further refined in recent works—integrates the textual and visual embeddings into a single, hybrid feature vector.
- Similarity Matching and Ranking: The fused representation is compared against a database of product embeddings using cosine similarity, and the top K matches are returned, ensuring that the retrieved products closely align with both the visual style and descriptive attributes of the query.

Chapter 3

Literature survey

3.1. CBIR Evolution

Information Retrieval IR originated from early library and information management systems, emerging as a formal academic discipline in the mid-20th century with the advancement of electronic technologies. A landmark event was Vannevar Bush's 1945 conceptualization of the Memex, establishing fundamental principles for contemporary IR systems (Bush, 1945). Further pivotal developments, such as Gerard Salton's probabilistic retrieval models, significantly shaped modern retrieval methodologies (Salton, 1989).

With technological evolution, digital image databases have grown exponentially, impacting fields like medicine, art preservation, and geographic information systems. This rapid expansion underscored the inadequacies of traditional text-based retrieval methods, particularly their heavy reliance on manual annotation. Such reliance is problematic, especially in domains requiring accurate recognition of complex visual content, exemplified by the medical field, where accurately retrieving images depicting subtle anomalies or tumors is crucial.

CBIR, introduced in the 1990s with systems like QBIC and VisualSEEk, offered significant advances by indexing and retrieving images based on their intrinsic visual attributes, such as color, texture, and shape (Faloutsos et al., 1994; Smith & Chang, 1996). Despite these advancements, CBIR continues to face considerable limitations, such as insufficient retrieval accuracy, poor scalability to large image collections, and the persistent semantic gap—the disconnect between low-level image features and high-level semantic interpretation.

Recent developments leveraging deep learning, especially CNNs), have substantially improved feature representation and retrieval accuracy. These networks automatically extract hierarchical features, significantly reducing manual intervention and enhancing semantic understanding (Krizhevsky et al., 2012; Doersch & Zisserman, 2017). However, deep learning methodologies still confront challenges, including high computational costs, scalability issues, and dependence on large, annotated datasets. Hybrid methods integrating handcrafted and deep-learned features have provided partial solutions, combining the advantages of classical image descriptors with deep learning's automated feature extraction, resulting in improved retrieval robustness (Wan et al., 2014; Babenko et al., 2014). Nonetheless, opportunities remain to further optimize accuracy, computational efficiency, and generalization, particularly through effectively leveraging partially labeled datasets.

Addressing these identified gaps and limitations provides the central motivation for this research. Consequently, this thesis proposes the Self-Optimizing DeepRec Convolutional Neural Network (SO-DRCNN), an innovative framework integrating the Ternion Paradigm—comprising Histogram of Oriented Gradients (HOG), Inclusive Color Histogram (ICH), and Slanting Express Revolves Concise (SERC)—to minimize reliance on labeled data, enhance computational efficiency, and significantly improve the accuracy and scalability of CBIR systems.

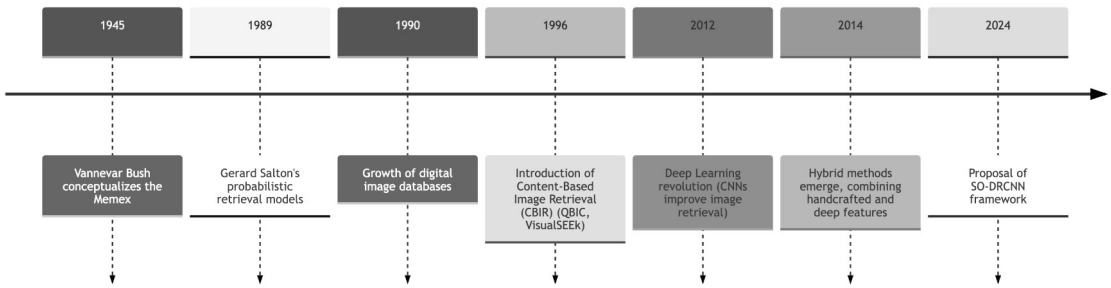


Figure 2 - CBIR Evolution

3.2. The Paradigm Shift Toward Learned Features

CBIR applications have been built to use various image features depending on the application area. The practical CBIR application utilizes the elements of visible contents known as image features. The popular features used in CBIR are color distribution, texture assembly, and shape form. They can practically identify and relate to the contents of an image. These features are carefully designed and have proven (A. Kumar et al., 2021) to work efficiently in most CBIR applications. Such features are generally called handcrafted features.

Even though many different CBIR systems have been developed and put to productive use, these systems all have problems with conceptual gaps and valuable features. While the conventional handcrafted descriptions still show significant shortcomings, they perform well in picture retrieval. These challenges must be carefully examined to develop a better performance system.

The following are the main shortcomings of the handmade features:

- Semantic gap remains the main problem notwithstanding the progress made in the CBIR. As a result, there is a certain level (Srivastava et al., 2023) of disintegration among the application's estimated attributes (such as texture and colour distribution) and people's cognitive perceptions of artefacts and situations.
- Handcrafted features are inefficient and not adaptable. It is tough to develop and deploy a new CBIR system. A wide range of handcrafted features are available, and the chosen features significantly impact retrieval results. System developers and end users require comprehensive studies to determine the most appropriate attributes.

To select appropriate characteristics, one must better understand the domain in which CBIR is being employed. The selection of features needs to improve the system's overall performance.

Finally, depending on the images' substance and nature, a set of attributes that performs effectively in a particular field may not produce satisfactory outcomes in another.

As a result, a requirement exists to develop characteristics that do not require prior knowledge of the application domain. The system should automatically generate or learn these characteristics based on the input data. Furthermore, the system should outperform the standard CBIR system, which uses handcrafted features. To overcome these issues in image retrieval, an increasing number of CBIR approaches have been introduced and are being investigated. Machine learning approaches allow (Radha et al., 2021) systems to derive substantial insights from incoming data. Systems of this kind will be able to recognize similarities and execute autonomous decisions without human involvement.

Machine learning techniques are widely used in various fields, including medicine, protection, networking websites, data science, and aerospace.

In addition, artificial intelligence is widely used in traditional image-processing activities, including categorizing and recognizing objects and segments. Machine learning algorithms can help deal with the shortcomings of handcrafted characteristics regarding image retrieving.

3.2.1. Image Similarity Measures Used in CBIR

By comparing the feature vectors of each image, we can determine how similar two images are. Diverging images have a larger difference value than comparable images.

Various metrics for similarity have been proposed in numerous image retrieval systems.

To be effective, a similarity measure must meet several criteria (Salih & Abdulla, 2021):

- Local Consistency: Following the triangular inequality in a neighborhood.
- Computational Effectiveness: The ability to work in real-time and on a large scale.

- Durability to Disturbances: Invariant to perturbations.

- Consensus with Semantics: Consistent with the concept of semantics.

Categories of Image Similarity Measurements:

- International, Region-Based, or Amalgamation of Both: Applying similarities based on either the whole image, specific regions, or a combination of both.

- Handling Characteristics as Matrices or Non-Vector Interpretations: Features are organized into a matrix format where each element represents a specific attribute or relationship in the image. This structure allows for more complex and nuanced representations of image features.

- Modelling approaches: supervised (monitored), semi-supervised, and unsupervised (unregulated) techniques. Supervised modelling uses labelled data to train models for high accuracy, semi-supervised combines small labelled with large unlabeled datasets for efficient learning, and unsupervised identifies patterns solely from unlabeled data, enabling the discovery of hidden structures. Each approach offers distinct advantages and challenges tailored to the specific needs of image retrieval tasks.

- Calculating Commonalities: Across linear space or nonlinear manifolds. Linear space methods use straightforward algebraic techniques for simplicity and efficiency, while nonlinear manifold approaches capture more complex relationships and structures within the data, providing a more nuanced similarity measure. Each method is chosen based on the specific characteristics and requirements of the image retrieval task.

- Significance of Image Portions: Considering the importance of different image parts in similarity calculations.

- Stochastic, Fuzzy, or Consistent Measures: Using different mathematical approaches for similarity measures.

To achieve tailored image searches, visual comparison metrics need to consider subjectivity more seriously. Besides terminology, concepts like aesthetics and individual preferences for content and style may also be included. Research is ongoing to extend

the idea of unpredictable image topologies to cover the entire range of natural visuals and enable customization.

In brief, different distance measures vary in their input type, computation method, computational difficulty, and metricity. The specific program and the feature vectors created determine which distance metric is employed. By considering various factors and incorporating subjectivity, future research aims to improve the effectiveness of image similarity metrics, making them more practical and user-centered.

3.2.2. Important Points Descriptors In CBIR Frameworks

Key points descriptors, such as regions, objects of interest, edges, or corners, have become invaluable in CBIR and various computer vision applications. These descriptors offer robustness and invariance to scale and rotation, providing significant advantages over traditional global features.

One of the most popular key point descriptors in recent years is the Scale-Invariant Feature Transform (SIFT). SIFT is well-known for its ability to match different views of objects or scenes, making it a key tool in many CBIR systems (Kapoor et al., 2021). However, SIFT's high dimensionality can slow down feature computation, especially when combined with techniques like Principal Component Analysis (PCA-SIFT) and Gradient Location and Orientation Histogram (GLOH).

To overcome these problems, Speeded Up Robust Features (SURF) was developed as a faster and more robust alternative. SURF keeps many of SIFT's advantages but improves computational efficiency. Another method, the Bag-of-Words (BoW) model, uses keypoint-based descriptors like SIFT to create a visual vocabulary. While BoW is accurate, it can be computationally demanding and memory-intensive, making it less suitable for large image collections (Patil & Kumar, 2013).

The Fisher Vector (FV) method, based on a Gaussian Mixture Model (GMM), offers a more informative image representation by encoding higher-order statistics. This approach leads to better performance compared to BoW. As a non-probabilistic alternative, the Vector of Locally Aggregated Descriptors (VLAD) aggregates residuals associated with each codeword to represent images, providing a simpler yet effective method (Patil & Kumar, 2017).

Local Binary Patterns (LBP), introduced for texture classification, have shown great utility in image retrieval. Variations like Local Ternary Patterns (LTP) and Center Symmetric Local Binary Patterns (CSLBP) further enhance performance. Local Tetra Patterns (LTrPs) and newer techniques such as Local Mesh Pattern (LMeP) and Local Ternary Co-occurrence (LTCoP) continue to push the boundaries in CBIR system design (G.-H. Liu & Yang, 2013).

Additionally, methods like Histograms of Oriented Gradients (HOG) and Compressed Histogram of Gradients (CHoG) provide robust feature descriptions. The Differential Between Pixels of Scans Pattern (DBPSP) focuses on pixel differences within scanning patterns for texture features. ORB (Oriented FAST and Rotated BRIEF), BRISK (Binary Robust Invariant Scalable Keypoints), and FREAK (Fast Retina Keypoint) are newer binary descriptors inspired by the human visual system, offering efficient alternatives to SIFT and SURF. The Nested Shape Descriptor (NSD) is a recent development that outperforms SIFT in binary form (Kapoor et al., 2021).

In conclusion, the evolution of key point descriptors has significantly enhanced CBIR systems' performance, making them more robust, scalable, and efficient in various applications. Their robustness and invariance to scale and rotation offer significant advantages over traditional global features.

1- Popular Key Point Descriptors:

- **SIFT (Scale-Invariant Feature Transform):**
    - Reliable for matching different perspectives of objects or scenes.
    - Widely used in CBIR systems.
    - High dimensionality can be a drawback, leading to slower feature computation.
- **SURF (Speeded Up Robust Features):**
    - Faster and more resilient to picture alterations than SIFT.
- **Bag-of-Words (BoW) Model:**
    - Utilizes keypoint-based descriptors like SIFT to create a visual vocabulary.
    - Computationally intensive and memory-heavy, less scalable for large image collections.

2- Advanced Feature Representations:

- **Fisher Vector (FV):**
    - Based on a Gaussian Mixture Model (GMM).
    - Encodes higher-order statistics for better performance.
- **Vector of Locally Aggregated Descriptors (VLAD):**
    - Aggregates residuals associated with each codeword to represent images.
    - Enhancements include intra-normalization, residual normalization, and localized location.

3- Local Binary Patterns (LBP):

- Used for texture classification and image retrieval.
- Variations like LTP, CSLBP, and LTrPs enhance performance.

4- Histograms and Gradients:

- Histograms of Oriented Gradients (HOG): Locally normalized descriptions for robust feature descriptions.
- Compressed Histogram of Gradients (CHoG): Reduced bit-rate characterization.

5- Binary Descriptors:

- ORB (Oriented FAST and Rotated BRIEF): Efficient alternative to SIFT and SURF.
- BRISK (Binary Robust Invariant Scalable Keypoints):Novel key point descriptor.
- FREAK (Fast Retina Keypoint): Inspired by the Human Visual System.
    - Nested Shape Descriptor (NSD): Outperforms SIFT in binary form on the VGG-Affine test.

3.2.3. Distance Metric Utilized In CBIR System

In CBIR, accurately measuring the similarity or dissimilarity between images is crucial for effective retrieval. This process, following feature extraction, hinges on the use of distance metrics that can capture perceptual similarity accurately.

Traditional distance metrics like Manhattan Distance (MD), Euclidean Distance (ED), and Vector Cosine Angle Distance (VCAD) are commonly used but often fall short in reflecting human perception accurately (Chugh et al., 2021). The Minkowski distance, despite its popularity, also struggles with perceptual accuracy.

Advanced metrics such as Kullback-Leibler Divergence (KLD) and Earth Mover's Distance (EMD) provide a more nuanced approach. EMD, based on the transportation problem, has proven effective across various applications, including color, contour matching, texture, melodies, and visual tracking. These advanced metrics offer better perceptual distance representation, but their benefits are maximized only with efficient storage and query processing.

Detailed assessments have shown that EMD, among others, excels in picture similarity searches. However, to leverage its high-quality retrieval capabilities, efficient storage and query processing are essential.

CBIR systems often use low-level features like color, texture, shape, and corners to approximate the perceptual representation of an image. Yet, these features alone are insufficient for capturing the full semantic relationships within an image.

Innovative learning techniques are being explored to overcome these limitations. For instance, Bian and Tao introduced the Biased Discriminative Euclidean Embedding (BDEE), enhancing relevance feedback mechanisms by integrating relevant and irrelevant data. Similarly, Biased Maximum Margin Analysis (BMMA) and Semi-supervised BMMA (Semi-BMMA) incorporate feedback and unlabeled data, refining the image retrieval process (Ghrabat et al., 2019). Despite these advancements, active learning techniques remain crucial, especially with insufficient training examples.

To address long-term feedback challenges, strategies like Case-Based Long Term Learning (CB-LTL) are proposed, focusing on capturing user preferences over time. Additionally, graph-based re-ranking methods, such as the random walker algorithm, offer innovative solutions for ranking images based on user-labeled data. These methods calculate ranking scores by determining the likelihood of a random walker reaching a relevant seed node before an irrelevant one, thus improving retrieval accuracy.

3.2.3.3 CBIR with Relevance Feedback

Since there is currently no dependable framework for modeling high-level image semantics that is unaffected by perceptual subjectivity, particular to the case, query interpretations can be understood by looking at user input. RF is a query adjustment method that aims to extract semantic information particular to the user and the query, then adjusts the results accordingly. This CBIR approach requires a significant amount

of user interaction along with input on relevance. Systems that are founded on subjective requests from users cannot coexist with a completely automated, unsupervised system; relevance feedback offers a middle ground. The primary challenge in establishing such a paradigm is the increasing user interaction, given the highly diversified user population.

Additionally, there's the matter of how well the input can be improved. Although users would prefer fewer sessions for feedback (Veselý, 2023), there is a problem with the amount of feedback that is necessary for the system to understand the needs of the user. While assuming a fixed goal is weaker, a single problem that has been widely overlooked in the applicability of feedback-based CBIR research is the possibility that the user's demands would change as the assessment phase's progress.

3.2.3.4. Color-Based Features in CBIR

Color is one of the most accessible visual elements in digital images, typically displayed as color components or planes. Extracting color-based features involves three main steps: selecting the color space (Garg & Dhiman, 2021), quantizing the color space, and extracting the color features (Muthukkumar & Seenivasagam, 2022).

Key Techniques in Color-Based Feature Extraction:

1. Color Histograms:

- Conventional Color Histogram (CCH): Represents the frequency of each color in an image. It is straightforward but may lack robustness against variations in lighting and quantization errors.

- Fuzzy Color Histogram (FCH): Uses a fuzzy-set membership function to record each pixel's color similarity to all histogram bins (Giannoulakis et al., 2023). FCH is more resistant to lighting variations and quantization errors, but determining the proper fuzzy membership function can be computationally challenging.

2. Color Correlogram (CC):

- Describes how the spatial correlation of color pairs changes with distance. It is indexed by color pairs, indicating the likelihood of finding a pixel of color "j" at a distance "d" from a pixel of color "i" (Alyosef, 2023). CC captures both local and global spatial data, making it effective for coarse-grain color images. However, it has a high computational cost due to the quadratic increase in image dimensions.

3. Color Averages and Block Truncation Coding (BTC):

- Color Averages: These methods aggregate an image's color information into feature vectors such as row mean (RM), column mean (CM), forward diagonal mean (FDM), and backward diagonal mean (BDM) (Warburg et al., 2021). These reduced-dimension vectors facilitate more efficient image retrieval.

- Block Truncation Coding (BTC): This technique divides an image into non-overlapping square segments and applies color averaging. It has been expanded to various color spaces, showing that luminance-chrominance spaces like Kekre's LUV provide superior retrieval performance compared to non-chrominance spaces. (Kekre et al., 2010)

Research has demonstrated the effectiveness of different color-based techniques in CBIR:

- **Comparative Performance:**

- Shen et al. (2018): Found that 8-color CC outperforms 64-color CCH, highlighting the importance of considering spatial information in color feature extraction.

- Amitha et al. (2021): Suggested new feature vectors based on image partitioning combined with color averaging approaches, noting the superiority of techniques like FDM in performance.

Future Directions will be continued research in color-based CBIR involves exploring hybrid approaches that combine color features with other visual features to enhance retrieval performance. Additionally, refining existing techniques to reduce computational costs while maintaining robustness and accuracy remains a key focus.

3.2.3.5. Image Retrieval using Transformed Image Content

Image transforms are essential for altering the representation of an image by projecting it into a collection of basis functions, commonly referred to as basis images. This transformation shifts the image representation from one domain (e.g., time domain) to another domain (e.g., frequency domain), without altering the intrinsic information in the image (Zhuang et al., 2022).

There are two primary benefits to this transformation:

I. Separation of Visual Patterns:

   Image transforms effectively separate critical elements of visual patterns, making them directly accessible for analysis (C. Hu, 2021).

II. Efficient Storage and Transmission:

   Transforming visual data into a more compact format facilitates efficient storage and transmission.(Datta et al., 2008)

These benefits make image transforms a vital tool for feature vector size reduction in image retrieval systems. Various CBIR techniques exploit these properties of image transformations, including fractional energy, row mean of columns converted image, energy compaction, and Principal Component Analysis (PCA).

Key Techniques and Findings

a. Fractional Energy:

Fifteen fractional parameter types, encompassing seven image transforms, are considered in CBIR when utilizing the fractional energy of the modified image (Jardim et al., 2022).

The Kekre transform with 6.25% fractional coefficients has been found to perform the best. Fractional energy-based CBIR outperforms approaches using the entire transformed image as the feature vector across all considered image transformations (P. He et al., 2022).

b. Independent Cosine Transformation:

Among the seven visual transformations considered, the independent cosine transformation with a DC element has provided the best results for image retrieval using the row mean of columns converted image material (Prasomphan & Pinngoen, 2021).

c. Energy Compaction:

In CBIR with energy compaction in the transform domain, compressed energy for converted color averages performs better with a significantly smaller feature vector size. The 94% energy Kekre transform has outperformed other methods in cases involving row mean, column mean, and row-column mean combinations (Oyewole, 2021).

d. Discrete Sine Transform:

The discrete sine transform provides superior image retrieval in both forward and backward diagonal means.

e. Principal Component Analysis (PCA):

When PCA is applied to color averages, image retrieval performance is somewhat reduced compared to when PCA is applied to the entire dataset (Feng et al., 2022). However, combining PCA with other CBIR approaches has demonstrated significant reductions in computational complexity while maintaining adequate retrieval performance.

In conclusion, Image transforms are crucial in CBIR for enhancing the accessibility and efficiency of image feature extraction. Techniques leveraging fractional energy, independent cosine transformation, energy compaction, and PCA are pivotal in

improving image retrieval performance while effectively managing computational demands.

3.2.3.6. Image Acquisition Employing Textured Information:

Texture is a crucial aspect of human vision, instrumental in distinguishing between various regions within an image. Unlike color and shape features, texture features are adept at capturing both the macrostructure and microstructure of images by indicating the distribution of shapes (Zhuang et al., 2022). Texture is typically identified as a spatial pattern with certain homogeneity-related features.

- **Methods for Extracting Texture Features**

  To obtain texture information from images, several directional feature extraction techniques are employed:

I. Steerable Pyramid:

Produces a multi-scale, multi-directional representation of an image, consisting of one decimated low-pass sub-band and several un-decimated directional sub-bands. The breakdown is iterated at the low-pass sub-band (Gayathri & Mahesh, 2022). This method results in a representation with 4K/3 times as many coefficients as the original image due to un-decimated directional sub-bands.

II. Contourlet Transform:

Decomposes an image into multiple scales and directions by combining a directional filter bank (DFB) and a Laplacian pyramid. The DFB processes band-pass images from the Laplacian pyramid to obtain directional data. This method results in a redundancy ratio of less than 4/3 due to decimated directional sub-bands (Sain, 2023).

III. Gabor Wavelet Transform:

Utilizes a bank of Gabor filters, which are adjusted by dilating and rotating the Gabor functions to produce a filter bank with K orientations and S scales. The image is then

convolved with each Gabor function, providing detailed texture retrieval outcomes (Tena et al., 2021). However, this method leads to a highly redundant representation of the original image.

- **Texture Representation Techniques**

Three main types of texture representation techniques are utilized to develop unique image retrieval strategies based on texture content:

1. Statistical Techniques:

- Employ non-deterministic features to analyze the spatial distribution of grayscale values. First-order statistics consider individual pixel values, while second-order and higher-order statistics account for spatial interactions between pixels. The Co-occurrence Matrix is a commonly used method for second-order statistical texture analysis (Rayavaram, 2023).

2. Model-Based Techniques:

- Represent an image using models like the Markov model and fractal model, which describe textures as combinations of fundamental functions or probability models. These techniques are useful for texture analysis, discrimination, and representing natural textures with statistical roughness and self-similarity.

3. Transform-Based Techniques:

- Aim to find a compact, lower-dimensional representation of texture features by transforming the image into a space where most data energy is concentrated in a few coefficients. Examples include Fourier, Gabor, Curvelet, and wavelet transforms. These methods enhance feature extraction efficiency by eliminating unnecessary coefficients (Yang et al., 2024).

Applications and Advancements

Texture features have significant applications in various domains such as aerial imagery, medical imaging, and more. They offer valuable insights into the surface granularity and recurring patterns within an image, making them crucial for domain-specific image retrieval tasks (Imbriaco, 2024). Recent advancements include the development of texture thesauruses for aerial image retrieval (Uddin Molla, 2021) and affine-invariant texture feature extraction techniques for texture recognition (Kanwal et al., 2021).

In summary, Texture features are vital in CBIR for providing a detailed and nuanced understanding of image content. By leveraging statistical, model-based, and transform-based techniques, researchers can develop robust image retrieval systems that effectively utilize texture information.

3.2.3.7. Image Retrieval using Shape Content

Shape representation is a critical element in image discrimination and serves as an effective feature vector for image retrieval. There are two primary methods of shape representation: region-based and boundary-based.

1. Boundary-Based Shape Representation:

This method utilizes the external boundaries of objects. Gradient operators and morphological procedures are typically used to extract a shape's border from an image. Gradation operations produce the image's first-order derivatives, enabling the identification of boundaries in horizontal, vertical, or diagonal directions (Majhi et al., 2022). A gradient amplitude approach with gradual regulators is applied to obtain the entire border of the shape in the image as connected edges.

2. Shape Representations and Design Commonalities:

Shape representations benefit greatly from effective and reliable depiction, especially in segmented picture areas. The representation of shapes often involves geometric

representations paired with one another (Chan, 2021). Over time, there has been a shift from global form descriptions to more regional descriptors. Simplifying the contour through discrete curve evolution helps eliminate unimportant or noisy shape elements (V. Kumar et al., 2022).

3. Shape Context and Shape Matching:

A novel shape descriptor called shape context is suggested for similarity matching. It is small but resilient to several geometric modifications. For effective form matching and shape-based picture retrieval, curves are represented by segments or tokens, whose feature representations (curvature and orientation) are grouped into a metric tree (Liao et al., 2024). Dynamic programming (DP) methods are also used for shape matching, where shapes are represented as sequences of concave and convex segments.

4. Fourier Descriptors for Shape Matching:

- An accurate form-matching strategy using Fourier descriptors utilizes both amplitude and phase, along with dynamic temporal warping (DTW) distance rather than Euclidean distance. This approach retains rotational and starting point invariance by adding compensating terms to the original phase, enhancing shape discrimination (Aboali et al., 2023).

5. Edge Detection Methods:

- Several edge detection techniques are employed for shape content-based image retrieval. These include the Sobel mask with slope magnitude method (Sobel-SMEI), Robert mask (Robert-SMEI), Prewitt mask (Prewitt-SMEI), Canny operators (Canny-SMEI), morphological operations (Perez, 2021), top hat transformation (Top-Hat-EI), and bottom hat transformation (Bot-Hat-EI). Edge images obtained from these methods are used as feature vectors in CBIR.

6. Block Truncation Coding (BTC):

- BTC on edge image content is used in novel image retrieval algorithms. Shape-edge images processed with BTC have shown outstanding accuracy. The row average of columns converted edge pictures further improves performance, with Kekre transforms combined with Robert slope amplitude edge images providing excellent effectiveness (Zhang, 2021).

7. Walsh Texture Pattern Image Retrieval Techniques:

- Recently proposed shape Walsh texture pattern image retrieval techniques, combined with even picture parts augmentation, demonstrate exceptional performance. Applying Walsh texture patterns to original plus even image sections with Robert slope amplitude boundary photos produces significant results.

It can be seen that shape features play a vital role in CBIR, offering detailed and robust representations for effective image retrieval. By leveraging various shape representation techniques, including boundary-based methods, shape context, Fourier descriptors, and edge detection methods, researchers can enhance the accuracy and efficiency of CBIR systems.

3.2.3.8. Metric Learning:

In practical applications, mastering a good feature space distance metric is essential. Every problem has a unique semantic concept of similarity, which common metrics (such as Euclidean distance) frequently fail to represent. Learning a metric that allocates a modest distance between pairs of examples that are semantically similar (as opposed to dissimilar) is the fundamental principle behind learning. A subfield of machine learning called Distance Metric Learning (DML) seeks to extract distance information from data. In addition to applications in dimensionality reduction, distance metric learning can be utilized to enhance similarity learning algorithms realistically.

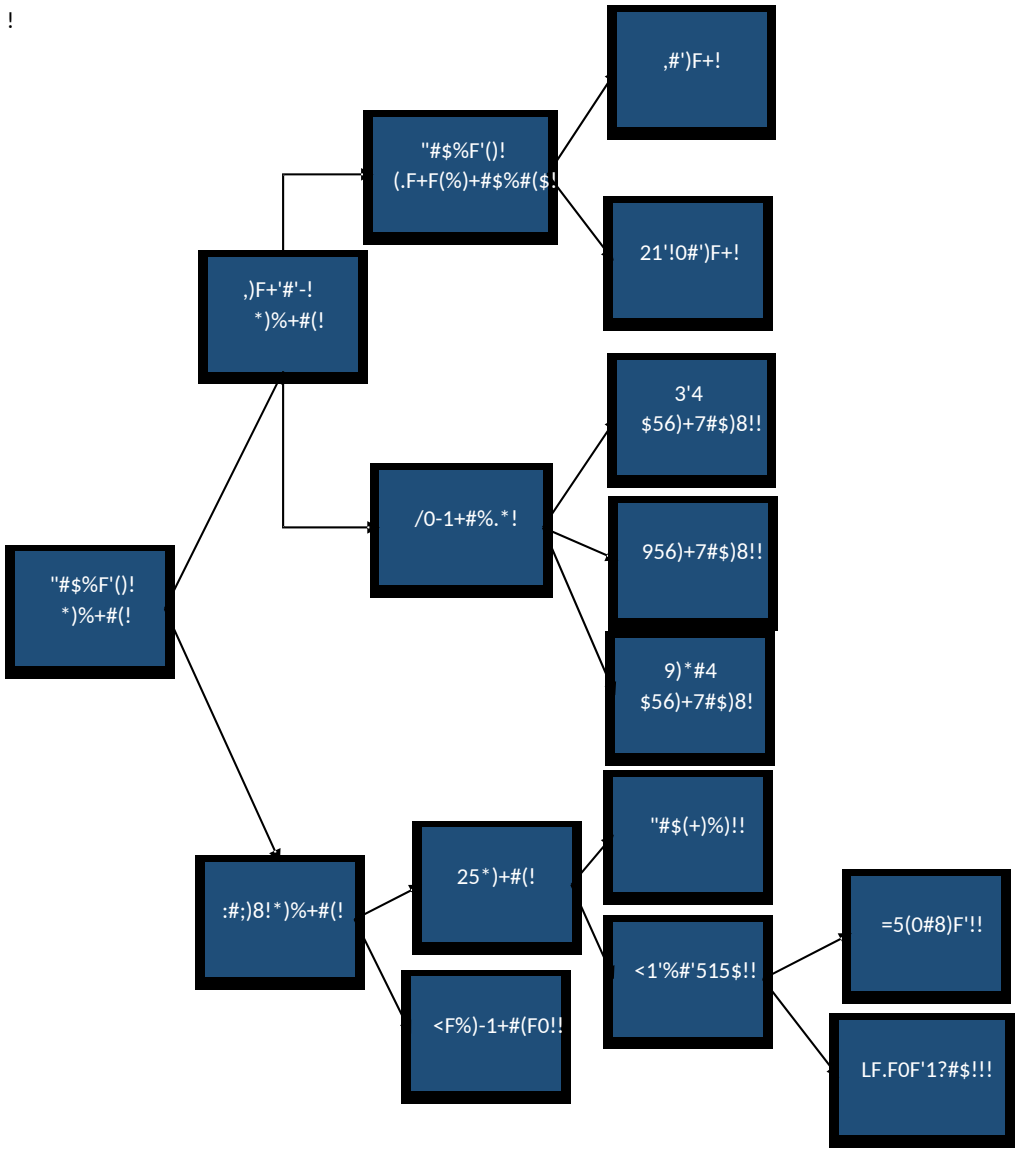


Figure 3 - Taxonomy of Distance Metrics

Taxonomy of Distance Metrics

The taxonomy of distance metrics is displayed in Figure 3. Depending on the features of distance, metrics can be categorized into linear and non-linear metrics. For many artificial intelligence tasks, including image classification and CBIR systems, good distance measures are important. For instance, the criterion used to determine the comparable nature of images has a substantial impact on the retrieval quality of CBIR systems (Papakonstantinou, 2023). This has led to extensive research on learning

effective distance parameters through training data. Linear Metric Learning includes similarity learning and Mahalanobis distance learning. Non-linear metric learning includes the kernelization of linear approaches as well as nonlinear and local metric learning.

Applications of Metric Learning

Metric learning finds applications in various fields, including medical informatics, computer vision, and text recognition. For example, in image retrieval, the Conventional Color Histogram (CCH) method shows how frequently each color appears in an image. Using the three color histograms—Conventional Color Histogram (CCH), Inclusive Color Histogram (ICH), and Fuzzy Color Histogram (FCH)—Song et al. (2018) have suggested color identification and comparisons. Challenges encountered include the CCH's high dimensionality, lack of support for rotation and translation, and susceptibility to noise-related issues like quantization mistakes and variations in light. Color gradients and the Fuzzy Linking Color Histogram (FCH) address these problems.

A new and quick method for content-based image retrieval using color histogram representation, known as the Fuzzy Colour Histogram (FCH) system, was proposed by Ju Han and Kai-Kuang Ma in 2002. It consists of an SVM classifier and an overview of the MPEG-7 Edge Histogram Descriptor (EDH), used to extract information from images. By integrating features like color correlogram, color instances, Gabor texture attributes, and boundary histogram descriptors, (Breznik, 2023) created CBIR systems. The SVM classifier is used to compare experimental findings with the CBIR system.

Graph-Based Query-Specific Fusion Strategy

Researchers have explored ways to increase retrieval precision without sacrificing scalability by fusing ordered retrieval sets, or the ranks of images provided by several retrieval algorithms. (Shen et al. 2018) proposed a graph-based query-specific fusion

strategy, in which various graphs are combined and re-ranked by performing a link analysis on a fused graph. They characterized retrieval ranks as graphs of candidate photos. Using local or holistic features, this method can combine the best aspects of retrieval algorithms for various query photos. This approach is simple to use, has few parameters, and doesn't require any supervision.

(He et al. 2018) similarly investigated ways to increase retrieval specificity without sacrificing scalability by combining ordered retrieval sets. They proposed a graph-based query-specific fusion strategy that re-ranks combined graphs through link analysis, enhancing retrieval precision.

Neural Network-Based Approaches

Neural networks have shown significant promise in CBIR systems. Techniques such as feed forward back propagation networks and Splines Neural Network based Image Retrieval (SNNIR) systems leverage the power of deep learning to model complex feature relationships. For instance, a Deep Auto Encoder (DAE) combined with wavelet transformation has been proposed to process images and extract wavelet coefficients for improved retrieval performance (Kim et al., 2019).

Several combinations of distance and directional angles have been provided by (Kostelecká, 2022) for the GLCM computation, which is examined to identify specific patterning visuals based on their textural qualities. Testing was done on checkered, irregular, right- and left-diagonally striped, vertically striped, and horizontally striped designs.

A feed forward back method for CBIR image retrieval has been proposed by Srivastava et al. (2023). A feed forward back propagation neural network-based CBIR image retrieval system has been proposed by Adil, (2021). The neural network is first trained with respect to the attributes of the database's photos. The color histogram serves as the

color descriptor, the GLCM serves as the texture descriptor, and the edge histogram serves as the edge descriptor for the training set of image features. The image that matches and is pertinent to the query input is obtained from the database. It is possible to attain an average recall rate (ARR) of roughly 78% and an average retrieval precision (ARP) of roughly 88%.

In order to extract the characteristics from the image, (Almohammed, 2021) presented a method that combines edge information with a median filtering technique. The method of Self Organizing Maps (SOM) is applied to group retrieved features from images. To obtain a smooth image, the original image is subjected to the median filtering process. The Bi-directional Empirical Mode Decomposition (BEMD) technique can be used to recover the edge-related data regarding the image. In order to determine which cluster the query image belongs to, the neural network is fed features from the image.

A new design for a CBIR system utilizing the Splines Neural Network based Image Retrieval (SNNIR) system has been developed by (Kumar et al. 2021). SNNIR uses an activation function of cubic splines in a fast and accurate network model. The nonlinear link between picture features is ascertained by the suggested method. Comparing the suggested approach to other CBIR systems, experimental results demonstrate that it achieves excellent accuracy and efficacy in terms of ARP and ARR.

3.2.3.9. Current state-of-the-art CBIR techniques:

Different methods and attributes of images are employed for image retrieval in the CBIR system. The CBIR system is primarily composed of two steps:

(i) Extracting features from the photos, and

(ii) Retrieving images a (Sedmidubsky et al., 2021) from the database.

Visual or content features were taken out of every database images and saved during the feature extraction stage. The following stage involved extracting features from a query

image and retrieving from the dataset photographs that shared those features. The two main techniques used in the CBIR system are transform domain-based methods and spatial methods. Techniques that are based on space make (Loria, 2020) use of an image's color, texture, geometry, and other characteristics. The characteristics of frequency-transformed color images are used by transform-based techniques. The efficient CBIR system uses spatial as well as transform features in contemporary approaches.

- **Techniques for Spatial-Based CBIR:**

I. Texture Property Analysis and Feature Extraction:

In 2021, Ghazouani & Barhoumi, developed a CBIR system grounded in a detailed statistical analysis of texture properties. They combined Feature Extraction (FE) and Similarity Measurement (SM) through a classification algorithm. By employing a consistent estimator, they extracted texture parameters and computed the Kullback-Leibler Distance (KLD) between estimated models during the SM stage. Their approach utilized Generalized Gaussian Density (GGD) modeling of wavelet coefficients. Tests on the VisTex database, with its 640 textured images across 40 classifications, showed retrieval rates improving from 65% to 77% compared to traditional methods. Despite its effectiveness, the method was time-consuming.

II. Visual Content Descriptors:

In 2024, Xun examined the visual content descriptors of the PicSOM system provided by MPEG7 against a Vector Quantization (VQ)-based reference system. The study found that PicSOM's descriptors were less effective and slower at locating relevant photos than those in the VQ-based system. Nevertheless, PicSOM's robust relevance feedback (RF) operation enhanced data retrieval efficiency. The research also highlighted that color descriptors performed better than other types, with scalable color descriptors yielding superior results for certain image classes, such as planes and horses (Saikia, 2021).

III. Wavelet-Based Retrieval Techniques:

Ghazouani and Barhoumi (2021) also contributed a wavelet-based retrieval approach. This technique employed Generalized Gaussian Density (GGD) modeling to improve retrieval rates significantly. Although effective, the method required considerable processing time. Testing with the VisTex database demonstrated that their approach outperformed conventional wavelet frames and pyramid wavelet transform techniques.

IV. Histogram-Based Description and Metric Learning:

In 2022, Zhang et al. introduced an automated method for image retrieval using histogram-based descriptions and metric learning algorithms. This technique used textual distances from scan reports to identify exam similarities without human interaction. Applying this method to interstitial lung disease patients' CT scans, the study utilized multiple medical annotation levels and various image descriptors. Validation through 60/40 cross-validation (CV) and Leave One Patient Out (LOPO) methods showed promise in efficiently categorizing and retrieving relevant medical images (Jaruenpunyasak & Duangsoithong, 2021).

V. 3D Active Shape Model and SURF:

In 2021, Salih & Abdulla, proposed an innovative method leveraging the Proactive Shape Model in a 3D environment combined with the Speeded Up Robust Feature (SURF) methodology. This approach applied a 3D model to 2D images to extract specific parts, enhancing the accuracy and efficiency of picture retrieval. Local features were removed using the 3D active shape model and SURF technique.

3.2.3.10. Comparative Analysis of Methods and Techniques in CBIR Systems

In the field of CBIR, various methods and techniques have been developed and refined to enhance image retrieval accuracy, efficiency, and scalability. This analysis presents a

comparison of these state-of-the-art methods, focusing on their unique features, advantages, and limitations.

Visual Content Descriptors

One of the foundational approaches in CBIR involves the use of visual content descriptors. For instance, a study examined the visual content descriptors of the PicSOM system provided by MPEG7 and a reference system based on Vector Quantization (VQ) for image retrieval. The research findings revealed that the PicSOM system's descriptors were not as effective as those in the VQ-based reference system, and they located relevant photos more slowly. However, the data retrieval efficiency of PicSOM surpassed that of the reference system due to its robust RF operation. It was also found that color descriptors performed superior compared to other descriptors, with different image classes yielding varied retrieval results (Saikia, 2021).

Dictionary Creation Techniques

Another significant advancement in CBIR is the use of dictionary creation techniques. This method involves creating a unique dictionary for every group of images. The results from experiments demonstrated improved categorization and picture retrieval using this approach. Additionally, the use of local image descriptors, which can accomplish image retrieval in two ways—local descriptors of picture reflecting global image and descriptor by descriptor matching—has shown promise. A CBIR system based on the probability distribution of local descriptors for every image in the database was created, achieving global picture representation by modeling the image using Probabilistic Principal Component Analysis (PPCA)(Kabir et al., 2022).

Machine Learning Algorithms

Machine learning algorithms play a crucial role in modern CBIR systems. Techniques such as Online Multi-modal Distance Metric Learning (OMDML) utilize a unified two-

level online learning strategy to tune the distance measure for each distinct feature space. This approach improves learning accuracy while lowering costs. Similarly, the use of the Support Vector Machine (SVM) classifier to calculate the similarity between the query image and the database image using various characteristics and distance measures has shown promising precision in classification (Ghazouani, 2023).

Multimodal Feature Extraction

Multimodal feature extraction involves characterizing photos by their visual aspects and other features. This approach addresses the challenge of weak relationships between semantics and visual features by adding a click feature to close the semantic gap. For instance, deep multimodal Distance Metric Learning (Deep-MDML) techniques utilize hierarchical ranking to investigate both click and visual features, thereby improving the retrieval accuracy and efficiency of CBIR systems (Hooda, 2022).

Relevance Feedback Mechanisms

Relevance feedback mechanisms are employed to enhance CBIR systems by allowing users to interactively refine their search results. Long-term Relevance Feedback (LRF) with hidden annotation (HA) has been proposed to improve retrieval accuracy and efficiency. This method involves autonomously choosing images for annotators using semi-supervised learning and a multilayer semantic image representation, leading to improved efficiency as accuracy improves over time (Tsai et al., 2020).

The comparative analysis of methods and techniques in CBIR systems reveals that each approach has its unique strengths and weaknesses. Visual content descriptors, dictionary creation techniques, machine learning algorithms, multimodal feature extraction, and relevance feedback mechanisms all contribute to the advancement of CBIR technologies. By leveraging these diverse methodologies, CBIR systems can achieve higher retrieval

accuracy, better efficiency, and greater scalability, making them more effective in various applications.

3.2.4. Multimodal Fusion in Image Retrieval (MFIR):

Multimodal fusion describes a retrieval approach where a query is given as a synthesis of multiple media types. This includes illustrations, written content, plain text (unorganized, e.g., sentences), visuals, video content, and combinations of these, all supported by multimedia retrieving and annotating systems. Numerous methods have been proposed for retrieving images and their accompanying text, as well as for text, video, music, and speech retrieval independently. However, when a user requires a multimedia-based search, most current techniques designed for specific media types fall short. Multidisciplinary integration is deemed necessary to address these types of customer inquiries (Z. Hu, 2022).

Combining the effective retrieval techniques of two distinct media for simultaneous retrieval can be challenging, even if they are accessible separately. The goal of fusion research on learning for multimodal inquiries is to discover the best combination models and tactics. Unfortunately, there has been limited effort put into multimodal integration regarding image queries and annotations. This gap creates opportunities to investigate new user interfaces, querying models, and visualization strategies relevant to picture retrieval when combined with other media. For instance, video retrieval can be considered a rare application of these multidimensional retrieval techniques (da Silva, 2023).

Real-time utilization of fusion techniques is analytically affordable, but fusion learning is generally an offline process. Therefore, multimodal fusion is an excellent method for improving retrieval effectiveness instantly. However, further caution must be employed to prevent the fusion rules from overfitting the validation set used to train them.

In the realm of CBIR, multimodal fusion can significantly enhance retrieval performance by integrating various modalities such as text descriptions, audio annotations, and video content (Jiang et al., 2022 ; Ngiam et al., 2011).

Recent advances in deep learning and neural networks have facilitated more sophisticated fusion strategies, leading to better semantic understanding and more accurate retrieval results (Baltrusaitis et al., 2019). Moreover, user studies indicate that multimodal interfaces improve user satisfaction and engagement, highlighting the importance of developing robust multimodal retrieval systems (Atrey et al., 2010).

3.2.5. Semantic-Based Image Retrieval (SIBR):

Numerous uses for it exist, including fast image file searching, commercial business programs, online learning, virtual attractions, and academic purposes. One limitation of the CBIR is the semantic gap that might arise between an image's high level features and low level features. The qualities of an image and what individuals (Ammatmanee, 2022) interpret from it are not the same. There are two phases to semantically based image retrieval systems: the "building stage" and the "query stage." The construction stage involves using semantic-based image retrieval algorithms to extract low-level features from images in order to find interesting and meaningful patterns, regions, or objects based on shared visual feature qualities. These object/region properties serve as the input for the semantic image extraction procedure, which produces the conceptual representation of imagery. Following that, a database has the semantic features.

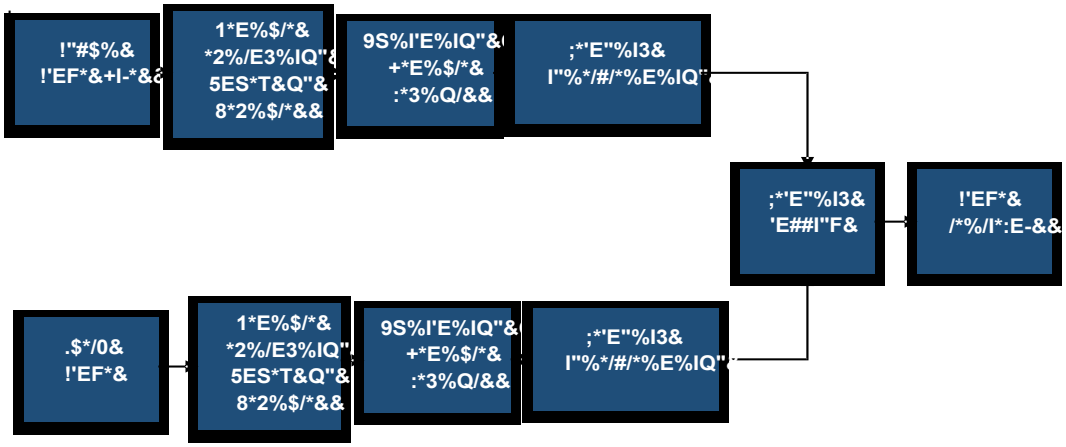


Figure 4 - SBIR framework

3.3. Summary

In the digital realm of the Internet, image retrieval is crucial. Technologies for retrieving images include evidence-based medicine, tele-surgery, computer-aided diagnostics, medical education, and more. Every potential algorithm for the picture retrieval system is examined in this chapter. The Text Based Image (Pospíšil et al., 2021) Retrieval (TBIR) method is labour-intensive, time-consuming, and sensitive to human perception. A more significant consideration is given to global and local information in the CBIR technique, including an image's color, shape, region, and texture. The primary limitation of CBIR is its incapacity to discern the attributes of diverse pictures. It could occasionally be challenging to identify a particular image from an enormous collection based alone on its content.

In this study, we propose a self-supervised, general-purpose image retrieval framework—termed SO-DRCNN—that addresses both the semantic gap and the data labeling bottleneck. Our solution learns image embeddings without human-provided labels by integrating handcrafted feature descriptors with deep neural network architectures. Central to this design is a “Ternion Paradigm” of feature extraction that fuses HOG, a color histogram, and a novel SERC descriptor, collectively capturing edges, color distributions, and structural patterns.

Chapter 4

Methodology

4.1. Introduction

This chapter details the methodology of the proposed research, presenting the systematic development and implementation of the Self-Optimizing SO-DRCNN framework for CBIR. The core objective is to address critical limitations in existing CBIR systems, specifically the semantic gap between low-level visual features and high-level human understanding, and the dependence on extensive manual data labeling. This work proposes a novel hybrid approach that combines the strengths of interpretable handcrafted features with the semantic representation power of deep learning, all within a self-supervised learning framework.

The SO-DRCNN framework integrates the following key components:

1. Ternion Paradigm Feature Extraction: A robust, multi-faceted feature extraction routine that combines HOG, ICH, and the novel SERC descriptor to capture a comprehensive range of visual information, including edges, color distributions, and structural patterns. This forms the handcrafted feature representation.

2. SO-DRCNN Embedding Generation: A deep neural network architecture, building upon a pre-trained ResNet-50 backbone, designed to extract high-level semantic features. The SO-DRCNN incorporates Recurrent Patching (with Bi-LSTMs), Spatial Pyramid Pooling (SPP/ASPP), and Attention mechanisms to capture local details, spatial context, and multi-scale information.

3. Siamese-Driven Feature Fusion: A novel, data-driven feature fusion strategy that employs a Siamese network, trained with a contrastive loss function, to learn an optimal combination of handcrafted features (from the Ternion Paradigm) and deep CNN embeddings (from the SO-DRCNN model). This adaptive fusion mechanism learns to balance the contributions of each feature type based on the data itself, maximizing the discriminative power of the fused representation for semantic similarity.

4. Self-Supervised Metric Learning (Auto-Embedder): A self-supervised training framework, based on a Siamese network and contrastive loss, that enables the SO-DRCNN to learn effective image embeddings and fusion weights without relying on manually labeled data. This addresses the labeling bottleneck and enhances the system's adaptability to new datasets.

5. Scalable Retrieval via Elasticsearch: Integration with Elasticsearch, a distributed search engine, to enable efficient indexing and retrieval of images based on their fused feature vectors, facilitating large-scale CBIR applications.

This hybrid approach, by unifying robust feature extraction, a carefully designed deep neural network architecture, self-supervised metric learning, and scalable retrieval, aims to deliver an effective CBIR system that generalizes across diverse image domains and scales to large datasets, all while minimizing the need for human annotation.

4.1.1. Research Design and Experimental Approach

The research utilizes an experimental approach, carefully structured to facilitate rigorous testing and validation of the Self-Optimizing SO-DRCNN framework for CBIR. This methodological framework enables systematic validation of hypotheses, precise measurement of performance indicators, and empirical verification of theoretical assumptions.

An experimental research design was selected due to its rigorous structure, which allows clear, quantifiable assessment of the CBIR system's performance. By employing controlled experimentation, the methodology objectively assesses retrieval accuracy, computational efficiency, robustness, and scalability. It is particularly suitable for addressing key issues identified in existing literature, notably the semantic gap and reliance on extensive labeling. This approach ensures that the effectiveness of each feature extraction technique, similarity measure, and retrieval strategy can be empirically tested and validated.

- **Overview of Iterative System Development and Validation Process**

The iterative system development and validation process comprises clearly defined stages:

1- Design Phase: Conceptualize the overall architecture, feature extraction techniques (color, texture, shape), and neural network strategies.

2- Implementation Phase: Develop and implement the algorithms for feature extraction, image processing modules, indexing mechanisms, and similarity metrics.

3- Testing Phase: Conduct structured experiments utilizing standard datasets such as CIFAR-10 and CLIP to evaluate system performance using well-defined

metrics including precision, recall, Mean Average Precision (MAP), and computational efficiency indicators.

4- Refinement Phase: Analyze experimental outcomes, identify areas for improvement, and systematically refine algorithms, parameters, and methods to enhance overall performance and efficiency.

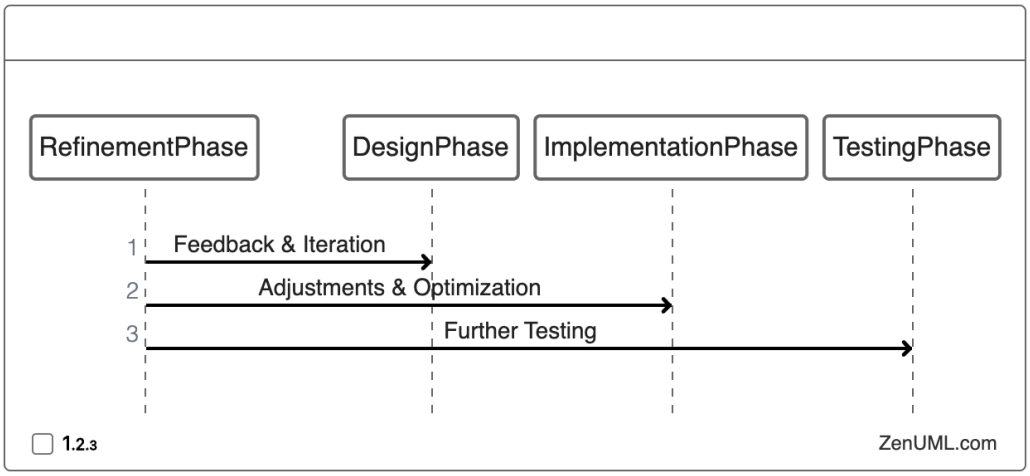


Figure 5 - Research Design Phase

4.1.2. System Architecture

The Hybrid CBIR system employs a multi-stage architecture to combine the benefits of classical and Deep Learning-Based feature extraction for robust and semantically-driven image retrieval.

1. Handcrafted Feature Extraction (Local Cues):

Function: Detects salient keypoints in the input image and extracts interpretable, handcrafted descriptors at these keypoints. This stage captures local visual patterns, texture, color, and structural information, providing human-understandable visual cues. Output: Handcrafted Feature Vector (BoVW Histogram) – a compact representation of local visual feature distributions within the image.

2. Deep Semantic Feature Extraction (High-Level Semantics): This stage employs a pre-trained ResNet-50 backbone integrated with the SO-DRCNN

architecture to extract deep semantic embeddings. The SO-DRCNN enhances ResNet-50 by incorporating recurrent patching (modeling spatial-contextual relationships via Bi-LSTM), spatial pyramid pooling (multi-scale feature aggregation), and attention mechanisms (emphasizing salient channels). These modules refine the network's ability to capture hierarchical semantics, including object categories and contextual relationships, transforming raw images into abstract, high-dimensional representations. The output is a CNN Embedding Vector—a compact, discriminative feature vector encoding high-level visual semantics.

3. Siamese-Driven Feature Fusion (Adaptive Combination): A learned Fusion Module, trained within a Siamese network framework, dynamically integrates handcrafted BoVW features with deep CNN embeddings. By optimizing contrastive loss on pseudo-pairs (similar/dissimilar image pairs), the module learns adaptive weights to balance local interpretability (BoVW) and global semantics (CNN). This results in a Fused Feature Vector that synergizes the strengths of both modalities, enhancing robustness to scale variations, occlusions, and domain shifts. The fused representation is optimized for semantic similarity, enabling precise retrieval by encoding both low-level patterns and high-level context.

4. Elasticsearch Indexing (Scalable Search Database): The fused feature vectors are indexed in Elasticsearch, a distributed search engine optimized for high-dimensional similarity search. Using the efficient k-nearest neighbor (k-NN) queries. This scalable indexing framework supports rapid retrieval across large datasets, minimizing computational overhead while maintaining accuracy. The output is a searchable Elasticsearch index, allowing real-time similarity comparisons and ranking based on cosine distance, critical for deploying the CBIR system in practical, resource-constrained environments.

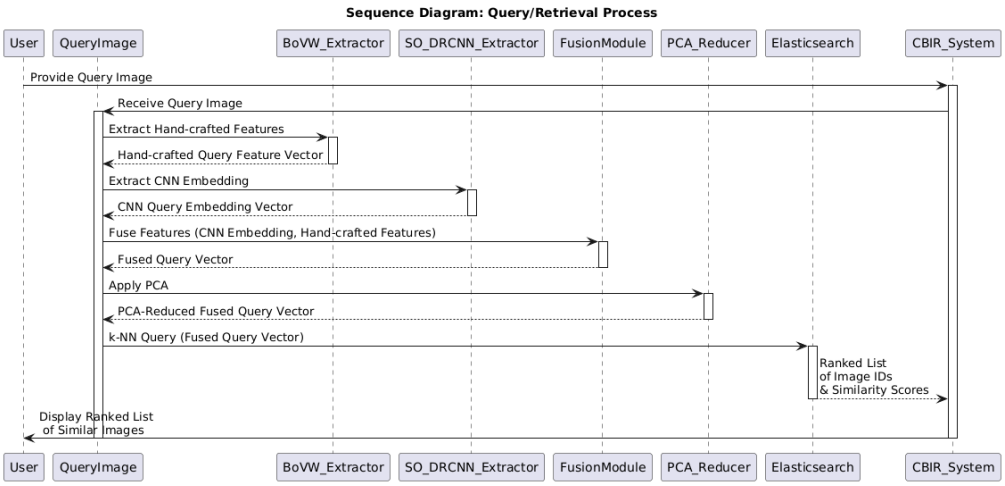


Figure 6 - CBIR Architecture

4.2. Proposed CBIR Pipeline

The proposed SO-DRCNN Hybrid CBIR system operates in two primary phases: an offline indexing phase for preparing the image database and an online query phase for real-time image retrieval. This two-phase structure is designed to optimize both system efficiency and retrieval accuracy.

Offline Indexing Phase (Preparation):

This preparatory phase focuses on pre-processing and indexing all images in the database to enable fast and efficient retrieval during the online query phase. The following steps are performed:

1. Handcrafted Feature Extraction: For each image in the database, extract handcrafted features using the BoVW framework with Ternion descriptors (HOG, ICH, and SERC), as detailed in Section 4.3 This results in a Handcrafted Feature Vector (BoVW histogram) for each image.

2. Deep Semantic Feature Extraction (SO-DRCNN): For each image, pass it through the trained SO-DRCNN model as detailed in Section 4.5 to generate a CNN Embedding Vector. This vector captures high-level semantic features learned by the deep network.

3. Siamese-Driven Feature Fusion: Combine the Handcrafted Feature Vector and the CNN Embedding Vector using the trained fusion module (extracted from the trained Siamese Network). This module performs weighted concatenation (or another learned fusion strategy) with weights learned during the Siamese training process (as detailed in Section 4.4.3). This results in a Fused Feature Vector for each image.

4. Dimensionality Reduction (PCA): Apply Principal Component Analysis (PCA) (Jolliffe, 2016) to the Fused Feature Vector to reduce its dimensionality, improving indexing efficiency and potentially reducing noise. This results in a PCA-Reduced Fused Feature Vector.

5. Indexing in Elasticsearch: Index the PCA-Reduced Fused Feature Vectors (along with image metadata like `*image_id*` and `*filename*`) in Elasticsearch. Configure Elasticsearch for efficient k-Nearest Neighbors (k-NN) search using cosine similarity (or Euclidean distance) as the distance metric.

Online Phase (Real-Time Query):

This phase handles user-submitted queries and retrieves similar images from the indexed database in real-time.

1. Query Image Input: The user provides a query image to the CBIR system.

2. Query Feature Extraction and Fusion (Identical to Indexing): The query image undergoes the same feature extraction and fusion process as the database images: Handcrafted Feature Extraction (BoVW): Extract the Handcrafted Feature Vector (BoVW histogram).

Deep Semantic Feature Extraction (SO-DRCNN): Generate the CNN Embedding Vector using the trained SO-DRCNN model.

Siamese-Driven Feature Fusion: Fuse the features using the trained Fusion Module.

Dimensionality Reduction (PCA): Apply PCA (using the same PCA transformation learned during indexing) to reduce the dimensionality of the fused vector.

Output: PCA-Reduced Fused Query Vector.

3. Similarity Search: Formulate a k-NN query in Elasticsearch using the PCA-Reduced Fused Query Vector to find the top-K most similar images in the index.

4. Result Ranking and Presentation: Elasticsearch returns a ranked list of image IDs and similarity scores (based on cosine similarity or Euclidean distance between the fused feature vectors). The system then retrieves and displays the corresponding images to the user, ranked by similarity.

This two-phase pipeline, with offline indexing and online query processing, ensures efficient and scalable image retrieval. The use of a trained Fusion Module, driven by the Siamese Network, enables data-driven and adaptive feature fusion, optimizing the system for semantic similarity. The integration with Elasticsearch provides fast and scalable search capabilities for large image databases.

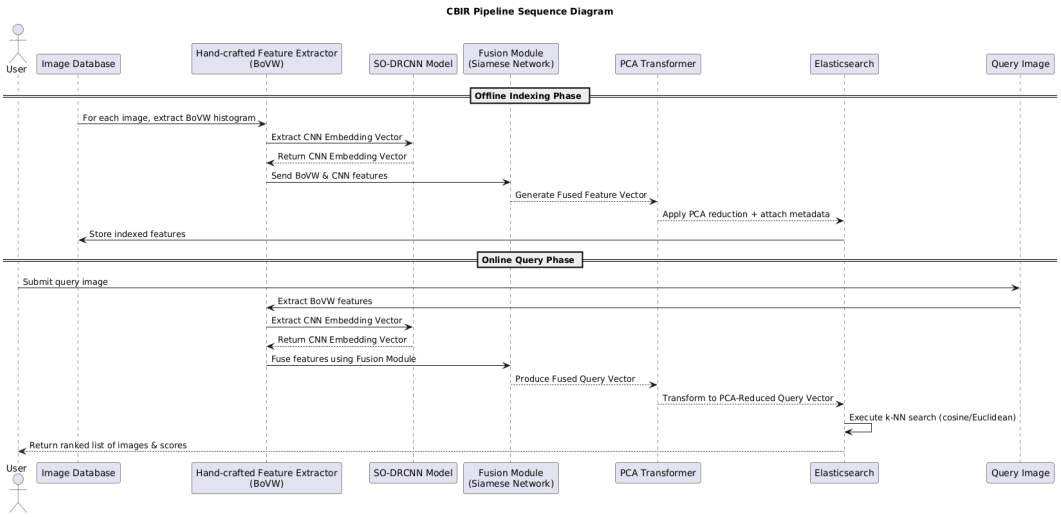


Figure 7 - CBIR Pipeline

4.2.1. Ternion Paradigm Feature Extraction Routine

To build a robust image representation, we extract three different types of features from each image – capturing texture/shape, color, and specialized patterns – which together form a ternion (triple) of descriptors. These are:

(a) HOG: As introduced in the literature study chapter 3, referred to by Dalal and Triggs (2005), HOG captures the local shape and texture of an image by computing histograms of gradient orientations within small cells and normalizing over larger blocks. In our system, HOG is used to represent the structural and edge information of an image, which is critical for distinguishing objects.

(b) GCH: The Global Color Histogram (GCH) is a feature representation that quantifies the distribution of colors in an image while disregarding spatial information. This method involves quantizing image colors (e.g., in RGB or HSV color space) into a fixed number of bins and counting the frequency of pixels in each bin, producing a histogram vector that captures the image's overall color composition (IJCA Online, n.d.). Color histograms are

computationally efficient and invariant to object translation or rotation, as they do not rely on positional data (IJCA Online, n.d.). Swain & Ballard, (1991) foundational research demonstrated that coarse color histograms are highly effective for image retrieval and object indexing. In the proposed framework, ICH—a global color histogram computed across the entire image—serves as the color descriptor. Here, we use the term ICH to indicate that the histogram encompasses all pixels of the image—effectively serving as a global color descriptor. ICH is computed by converting the image into a chosen color space (e.g., RGB or HSV), quantizing the color channels into fixed bins, and concatenating the resulting histograms. (Swain & Ballard, 1991). For consistency, we use "ICH" throughout this chapter; it is equivalent to what is commonly known as the global color histogram (GCH) in CBIR literature.

(c) SERC: SERC is a computationally efficient feature descriptor designed to capture slanting edges and structural patterns in images while ensuring rotation invariance and compact representation. It combines FAST keypoint detection, Harris corner refinement, multi-directional edge extraction, and a binary descriptor (rConcise) optimized via greedy search. The descriptor is integrated into a Bag-of-Visual-Words (BoVW) framework for holistic image representation.

4.3. Bag-of-Visual-Words Framework for Image Representation

To create a robust and efficient image representation for CBIR, we adopt the BoVW framework. BoVW is a widely used technique in computer vision, inspired by the Bag-of-Words model from natural language processing (NLP) (Manning, C. D., Raghavan,

P., & Schütze, H. (2008) Introduction to Information Retrieval. Cambridge University Press., Sivic, J., & Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In Ninth IEEE International Conference on Computer Vision (pp. 1470-1477 Vol.2). IEEE.). In NLP, a document is represented as an unordered collection (a "bag") of words, where the frequency of each word's occurrence is used to characterize the document's content. Similarly, BoVW treats an image as an unordered collection of visual words, representing local image features. The BoVW approach, inspired by text retrieval models (Sivic & Zisserman, 2003), represents an image as a histogram of visual word occurrences, where visual words are representative local feature patterns learned through clustering.

BoVW Pipeline

Our BoVW implementation consists of the following key steps:

1. **Keypoint Detection:** We employ the ORB algorithm (Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In 2011 International Conference on Computer Vision (pp. 2564-2571). IEEE.) for keypoint detection. ORB is chosen for its computational efficiency and invariance to image rotation, making it suitable for large-scale image retrieval. ORB identifies salient points in the image, such as corners and edges, which are likely to be stable and repeatable across different views of the same object or scene. It is important to note that we utilize ORB solely for its keypoint detection capabilities; the binary descriptors generated by the BRIEF component of ORB are not used in our BoVW framework.
2. **Local Descriptor Extraction:** At each keypoint location detected by ORB, we extract a local image patch centered on the keypoint. Within this patch,

we compute three types of feature descriptors, collectively referred to as the "Ternion" descriptors in previous passage.

- Histogram of Oriented Gradients (HOG): HOG captures local shape and texture information by computing histograms of gradient orientations within the patch.
- Inclusive Color Histogram (ICH): ICH, equivalent to a global color histogram but computed locally on the patch, quantifies the color distribution within the patch. We use the HSV color space for ICH computation.
- Slanting Express Revolves Concise (SERC): SERC is descriptor designed to capture structural patterns and edges within the patch, while maintaining rotation invariance.

These three descriptors capture complementary visual information, providing a richer representation of the local image patch than any single descriptor alone.

3. **Visual Vocabulary Construction (Clustering):** To create a visual vocabulary, we gather a large collection of these local Ternion descriptors (HOG, ICH, and SERC) from a representative set of training images. We then apply the mini-batch k-means clustering algorithm (Sculley, D. (2010)) to group these descriptors into k clusters. Mini-batch k-means is chosen for its scalability to large datasets, as it processes data in batches rather than loading the entire dataset into memory. We use a batch size of 64 and set k = 400. The choice of 400 visual words represents a balance between representational power and computational efficiency, and is consistent with values used in prior BoVW-based image retrieval works. The centroids of these k clusters

define our "visual words," and the set of all cluster centroids constitutes the visual vocabulary. Each visual word can be thought of as a representative local feature pattern.

4. **Image Representation (Histogram Generation):** For each image (both training and query images), we perform the following steps:
    - Detect ORB keypoints.
    - Extract local HOG, ICH, and SERC descriptors at each keypoint.
    - Assign each local descriptor to its nearest visual word (cluster centroid) in the vocabulary, using Euclidean distance as the distance metric.
    - Construct a histogram with k bins (one for each visual word). Increment the bin corresponding to the assigned visual word for each local descriptor. This histogram represents the frequency of occurrence of each visual word in the image.
    - L2-normalize the histogram. This normalization ensures that the histogram represents a probability distribution and prevents images with more keypoints from having disproportionately large feature vectors.
5. **Feature Fusion:** The L2-normalized BoVW histogram, representing the distribution of local features, is then fused with L2-normalized global HOG, ICH, and SERC descriptors computed over the entire image. This fusion combines the strengths of both local and global feature representations.

   A handcrafted feature vector is the output, typically a BoVW histogram potentially fused with global descriptors, representing local visual patterns and interpretable image characteristics.

6. **Vocabulary Quality Evaluation:**

   The quality of the visual vocabulary (i.e., the effectiveness of the clustering) is assessed using the Davies-Bouldin Index (DBI) . The DBI measures the ratio of within-cluster scatter to between-cluster separation. A lower DBI value (ideally below 0.5) indicates better clustering, with compact and well-separated clusters, suggesting a more effective visual vocabulary.

The next approach to structure our CBIR system is generating deep visual embeddings using the Self-Optimized Deep Recurrent SO-DRCNN model, enhanced with Siamese-Driven Feature Fusion. This stage leverages the power of deep learning to capture high-level semantic features and intelligently combine them with handcrafted features for enhanced CBIR performance and interpretability.

4.4. SO-DRCNN Model

4.4.1. Utilize Pre-trained ResNet-50 CNN Backbone

Pre-trained ResNet-50 Architecture (He, Zhang, Ren, & Sun, 2016): Employ a pre-trained ResNet-50 CNN architecture, initialized with weights pre-trained on the ImageNet dataset (Deng et al., 2009), as the foundational backbone for deep feature extraction. ResNet-50 is chosen for its proven efficacy in image representation learning, its balance of performance and computational efficiency, and the well-established benefits of transfer learning.

Transfer Learning Advantages:

Initializing with ImageNet pre-trained weights offers significant advantages, including encoding generalized visual features, which leverage a rich and generalized set of visual features learned from a vast corpus of natural images, capturing fundamental visual primitives and hierarchical representations.

Facilitating faster convergence during subsequent fine-tuning and self-supervised training and accelerating training convergence reduce the need for extensive training epochs and computational resources.

Improving the generalization capability of the learned embeddings, making the model more robust to unseen images and diverse datasets.

4.4.2. Enhancing ResNet-50 Features for Contextual Semantic Understanding

I. The Recurrent Patching Module employs a four-layer bidirectional Long Short-Term Memory architecture to model hierarchical spatial-contextual relationships in images. By processing image patches (e.g., 3x3 grid) sequentially in row-major order, the network interprets spatial arrangements as temporal sequences. Each Bi-LSTM layer (64 hidden units) processes patches bidirectionally—forward and reverse—capturing dependencies from both preceding and succeeding regions. This dual-directional context propagation enables the encoding of part-whole relationships (e.g., object components in natural scenes). The stacked design progressively abstracts features across layers, with higher layers learning complex compositional patterns (e.g., global scene structure from local edges).

To ensure stable training and mitigate potential vanishing or exploding gradients common in recurrent architectures, specific initialization strategies were employed: orthogonal initialization [Saxe et al., 2013] was used for the recurrent weight matrices (connecting hidden states across time steps), while Xavier/Glorot initialization [Glorot & Bengio, 2010] was applied to the input weight matrices. Furthermore, to enhance robustness and prevent overfitting during training, two regularization techniques were applied specifically to this module: Gaussian noise (with standard deviation $\sigma = 0.1$) was added to the input sequence vectors (the

2048-D patch representations), and Dropout (with probability $p = 0.3$) [Srivastava et al., 2014] was applied within the Bi-LSTM layers.

The final hidden states from the forward and backward passes of the top Bi-LSTM layer are concatenated, yielding a 128-dimensional context vector representing spatially enriched features. This context vector captures part-whole relationships and spatial dependencies across the image grid.

This approach bridges convolutional and sequential learning paradigms, demonstrating that spatial context in static images can be modeled via temporal recurrence. By capturing spatial-contextual dependencies, the module transcends isolated feature detection, encoding images as compositions of interrelated visual elements. This enhances robustness to viewpoint variations and occlusions, which is critical for semantic retrieval in CBIR. The Bi-LSTM architecture ensures sensitivity to local coherence (e.g., texture continuity) and global layout (e.g., object positioning), outperforming unidirectional or shallow recurrent models. Theoretically, it advances interpretable deep feature extraction, offering a framework for spatially aware representation learning in vision tasks.

To facilitate the Recurrent Patching Module's capture of spatial context, the feature maps from the ResNet-50 conv5_x layer are divided into a 3x3 grid of non-overlapping patches. This design choice, while heuristic, is academically grounded in the established principles of spatial subdivision for image representation, as exemplified by SPP [He et al., 2015] and Spatial Pyramid Matching (SPM) [Lazebnik et al., 2006]. A 3x3 grid offers a balanced approach, providing sufficient granularity to capture meaningful spatial relationships between image regions for sequential processing by the Bi-LSTM network, while maintaining computational manageability.

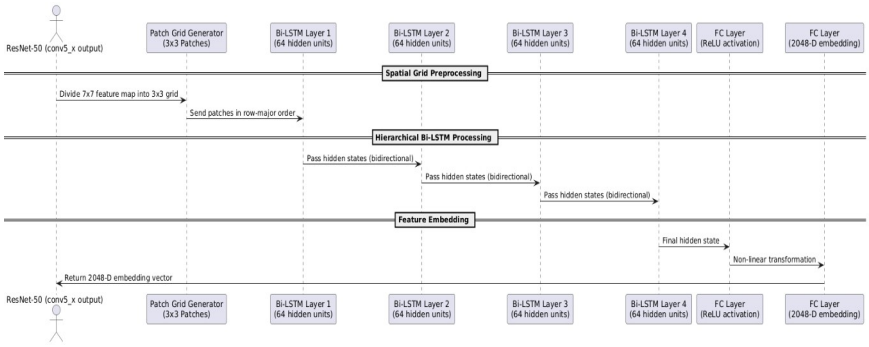


Figure 8 - Recurrent Patching Module

II. SPP and ASPP Modules for Multi-Scale Context Aggregation:

The Spatial Pyramid Pooling (SPP) and ASPP modules enhance CNN's ability to capture multi-scale contextual information, addressing scale variance in image retrieval. SPP applies multi-resolution pooling (e.g., 1×1, 2×2, 4×4 grids) to feature maps, generating fixed-length representations invariant to input dimensions. This ensures robustness to object size variations (e.g., small vs. large instances) by hierarchically aggregating local-to-global features. ASPP extends this with dilated convolutions, probing feature maps at multiple dilation rates (e.g., rates 1, 6, 12) to capture context across scales without resolution loss. Parallel dilated convolutions and global average pooling branches are concatenated, preserving fine details while integrating wide-field contextual cues. These pooled and dilated features are down sampled and fused with the original CNN outputs, combining multi-scale richness with base feature fidelity through concatenation or additive fusion.

The fused features are refined via a channel attention mechanism, inspired by Squeeze-and-Excitation Networks (Hu et al., 2018). This module recalibrates feature channel weights, amplifying discriminative dimensions (e.g., object-specific patterns) while suppressing noise (e.g., irrelevant background context). The attention block employs global average pooling to model channel

interdependencies, followed by learnable transformations to generate excitation signals. These signals rescale input features, enhancing semantic selectivity. The refined output is processed through a final convolutional layer (64 filters) and global pooling, producing a compact feature vector. By integrating spatial pyramids with attention-driven recalibration, the model achieves scale-invariant, context-aware representations critical for robust CBIR.

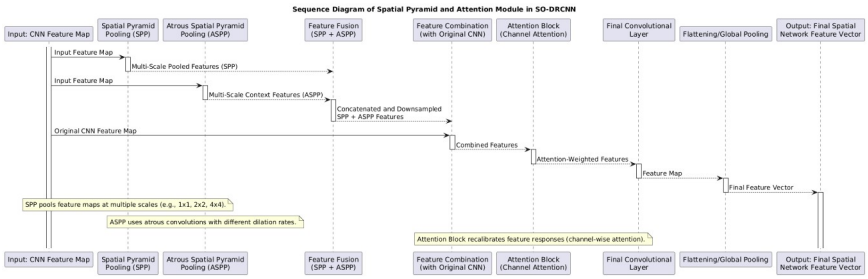


Figure 9 - Spatial Pyramid

III. Dimensionality Reduction: A Component Analysis for Efficiency and Noise Reduction

Apply Principal Component Analysis (PCA) (Jolliffe, 2016) to reduce the dimensionality of the CNN embedding vector to a target dimension (e.g., 2000 dimensions) for efficiency and indexing constraints. PCA is performed on a set of training image embeddings to identify principal components that capture maximum variance, and projecting embeddings onto these components results in a lower-dimensional representation that can improve retrieval speed, reduce storage requirements, and potentially mitigate noise while retaining the vast majority of the salient semantic information. PCA is recommended to be applied to the fused feature vector in next section 4.5.3 for optimal dimensionality reduction of the combined representation.

4.4.3. Siamese-Driven Feature Fusion

This stage introduces the core innovation of our methodology, Siamese-Driven Feature Fusion. This approach combines the handcrafted feature vector (Stage 1) and the CNN embedding vector (Stage 2) into a single, fused feature vector, leveraging a Siamese Network to learn an adaptive and data-driven fusion strategy that optimally balances the complementary strengths of these heterogeneous feature modalities.

Static and Heuristic Combination Strategies as traditional feature fusion techniques, such as simple concatenation or fixed-weighted combinations, often fall short in optimally integrating heterogeneous feature modalities. These methods treat feature combination as a static, pre-defined operation, lacking the crucial adaptability to learn data-driven fusion strategies that can dynamically balance the contributions of CNN embeddings and handcrafted descriptors based on the specific characteristics of the dataset and the desired notion of image similarity. This static nature hinders their ability to fully leverage the complementary strengths of these diverse feature representations and achieve truly optimized semantic CBIR.

Siamese Networks, with their inherent capability for learning similarity metrics through pairwise comparisons (Hadsell, Chopra, & LeCun, 2006), provide a robust and data-driven mechanism to achieve optimal feature fusion.

Siamese Network for Fusion Training

In our context, we are using the Siamese network concept to learn how best to combine two types of features—CNN embeddings which capture high-level semantic information and handcrafted features which provide interpretable local details. Instead of directly performing feature extraction, the Siamese setup is used to learn adaptive fusion weights. Essentially, we use it to determine the optimal way to mix these features so that, for

similar images, the fused feature vectors are close together, and for dissimilar images, they are far apart.

So, while the Siamese network is traditionally about similarity comparison, in our approach, its principles are applied to both:

- Assessing similarity by comparing fused vectors using contrastive loss
- Optimizing the fusion process by learning the best weights to combine the two feature types.

This dual role enhances the final feature representation, ensuring that the fusion process is both effective for retrieval and interpretable in terms of the relative contributions of CNN and handcrafted features.

The Siamese Network is not the measuring instrument itself; it's the calibration system that makes the measuring instrument (the Fusion Module) more accurate and effective.

Siamese Networks are beneficial given these feature natures as it adaptive balancing of heterogeneous feature scales and contributions. Learning to Bridge the "Semantic Gap" with Complementary Features,

Handcrafted features and CNN embeddings capture complementary types of visual information. The Siamese Network can learn to intelligently combine these complementary features to create a fused representation that is more effective at bridging the semantic gap. It can learn to use handcrafted features to refine or augment the semantic information captured by CNN embeddings, especially for fine-grained distinctions or texture-based similarity.

Imagine we have two images want to compare: Image X and Image Y.

Step 1: Feature Extraction Before the Siamese Network:

Here is an example of the approach:

For Image X:

Extract CNN Features: It runs Image X through the ResNet-50. This gives us a CNN Feature Vector for Image X. Let's call it CNN_Features_X.

Extract Handcrafted Features: System process Image X using the BoVW framework. This gives a Handcrafted Feature Vector for Image X. Let's call it Hand_crafted_Features_X.

For Image Y:

Extract CNN Features: It runs Image Y through ResNet-50. This gives a CNN Feature Vector for Image Y. Let's call it CNN_Features_Y.

Extract Handcrafted Features: it processes Image Y using BoVW/Ternion. This gives a Handcrafted Feature Vector for Image Y. Let's call it $Hand\_crafted\_Features\_Y$.

Important: we do this feature extraction step before even feed anything into the Siamese Network. We are preparing the "twins" first.

Feature Vector Concatenation (Baseline Fusion - Simple Combination for Comparison): As a foundational baseline fusion method for comparison and to establish a performance reference point, consider simple concatenation of the CNN embedding vector and the handcrafted feature vector to create the fused feature vector. This provides a basic, unweighted combination of the two feature types:

$$FusedVector = [CNN\ Embedding, Hand-craftedFeatures]$$

Step 2: Normalization:

Essential refinement for feature scaling, stability, and distance metric effectiveness. L2-normalize both the CNN embedding vector and the handcrafted feature vector before feeding them into the Fusion Module and concatenation. This normalization step is not merely optional but essential to ensure that both

feature types, originating from vastly different extraction processes and potentially having different scales and magnitudes, are placed on a comparable scale. This prevents features with larger magnitudes from unduly dominating the fused representation and facilitates a more balanced and stable fusion process. Crucially, also L2-normalize the final fused feature vector after the Fusion Module's processing to standardize the output representation, ensuring unit length vectors that are ideally suited for distance-based similarity comparisons in the CBIR system and for efficient indexing in vector databases like Elasticsearch.

Step 3: Feeding the Features into the Siamese Network (Twin1 and Twin2):

Siamese Twin 1 gets the Features for Image X:

Input to Siamese Twin 1: We feed both feature vectors for Image X into Siamese Twin 1: CNN_Features_X , Hand_crafted_Features_X

Siamese Twin 2 gets the Features for Image Y, Input to Siamese Twin 2: CNN_Features_Y , Hand_crafted_Features_Y.

The Fusion Module mixes or combines these two input feature vectors using weighted concatenation. It outputs a Fused Feature Vector for Image X. Let's call it Fused_Features_X.

Inside Siamese Twin 2 (for Image Y):

It outputs a Fused Feature Vector for Image Y. Let's call it Fused_Features_Y.

Step 4: The Siamese Network Compares the Fused Feature Vectors:

Siamese Network Output: The Siamese Network now has two outputs:

- Fused_Features_X (from Siamese Twin 1)
- Fused_Features_Y (from Siamese Twin 2)

Weighted Concatenation (Siamese-Driven Adaptive Fusion with Learnable Weights): To implement Siamese-Driven Feature Fusion and enable adaptive

balancing, employ weighted concatenation within the Fusion Module of the Siamese Network. This allows the network to learn optimal weights for each feature type, dynamically adjusting their contributions to the final fused representation based on the training data and the contrastive loss objective:

$$FusedVector = \quad [w1 \times CNNEmbedding, w2 \times HandcraftedFeatures \quad]$$

Determine optimal weights (w1, w2) through the Siamese network training process, guided by the contrastive loss function and rigorous validation set performance optimization. While equal weights (w1, w2 = 1) can serve as an initial baseline for experimentation, the Siamese network is designed to learn data-driven weights that are expected to outperform such heuristic settings by adaptively tailoring the fusion strategy to the specific dataset and retrieval task.

Step 3: Similarity Comparison:

Compares the refined embeddings from the twin branches using a distance metric (e.g., Euclidean, cosine).

Optimizes the distance to be small for similar pairs and large for dissimilar pairs.

For a pair of images, A and B:

- Compute embeddings

  $\boldsymbol{e_A} = SiameseBranchqFuse \quad (CNN_7, BoVW_7)r$

- Compute embeddings

  $\boldsymbol{e_B} = SiameseBranchqFuse \quad (CNN_9, BoVW_9)r$

- Calculate similarity: $similarity = 1 - cosin\ distance \quad (e_7, e_9)$

  Contrastive Loss: The contrastive loss function is then applied to this Euclidean distance. Based on whether Image X and Image Y are supposed to be "similar" (Can-Link) or "dissimilar" (Cannot-Link).

Step 4: Contrastive Loss Function for Data-Driven Fusion Weight Optimization:

The contrastive loss function is applied to this calculated distance, based on the pre-defined similarity label (Can-Link or Cannot-Link) for the Image X and Image Y pair. This loss function serves as the driving force for training the Fusion Modules within the Siamese Network, guiding the optimization of fusion weights and parameters. The Contrastive Loss encourages the Siamese Network to:

- Minimize the distance between $Fused\ \ Features_{:}$ and $Fused\ \ Features_{;}$ for similar (Can-Link) image pairs, effectively pulling embeddings of semantically similar images closer in the fused feature space and reinforcing the desired notion of visual similarity.
- Maximize the distance between $Fused\ \ Features_{:}$
- and $Fused\ \ Features_{;}$ for dissimila r(Cannot-Link) image pairs, enforcing a margin of separation and pushing embeddings of semantically dissimilar images further apart in the fused feature space, ensuring discriminative power of the learned fused representations.

The Siamese Network, through this rigorous and data-driven training process driven by the contrastive loss, empowers the Fusion Module to learn an optimal, adaptive, and context-aware strategy for combining CNN embeddings and handcrafted features. This learned fusion strategy results in fused feature vectors that are demonstrably more effective for measuring semantic image similarity and achieving enhanced performance and interpretability in CBIR.

Step 5: Backpropagation

Contrastive Loss Value can propagate the "Error Signal" backwards through the network that tells the Siamese Network how "wrong" it was in its current fusion

strategy. A high Loss Value indicates a "bad" fusion (similar images are not close enough, or dissimilar images are too close), while a low Loss Value indicates a "good" fusion.

Gradients for Weight Adjustment: The Weights within the Fusion Modules are the most crucial weights being learned in the Siamese-Driven Feature Fusion approach. Backpropagation tells how to adjust the weights in the Fusion Modules to improve the fusion strategy.

Potentially Weights in Other Parts of SO-DRCNN : If we are not freezing the weights of the ResNet-50 backbone or other SO-DRCNN modules, gradients will also be calculated for their weights, allowing the entire SO-DRCNN architecture to be fine-tuned for the CBIR task.

4.4.4. Self-Optimizing Fusion Module in SO-DRCNN

In the SO-DRCNN framework, the fusion weights w1 and w2 are treated as trainable parameters that are learned end-to-end. During training, a contrastive loss (or similar task loss) is computed on the fused output, and back-propagation computes gradients with respect to $w1$ and $w2$. In each training iteration, the model updates these weights by gradient descent to minimize the loss (Li et al., 2024; Sait & Nagaraj, 2025). In other words, the network automatically discovers the optimal values of w1, w2 for the given data and objective. This mechanism is identical to standard neural-network training (stochastic gradient descent on all parameters) but applied specifically to the fusion coefficients. For example, Wang et al. (2024) explicitly note that adaptive fusion coefficients (denoted α, β, γ) are “optimized through gradient descent” during training. Likewise, recent multi-modal networks use contrastive or cross-entropy losses to drive the learning of feature-fusion weights (Li et al., 2024; Wang et al., 2024).

Back-propagation of fusion weights: The fusion weights enter the computational graph (often implemented as a small sub-network or linear layer) so that the loss gradient flows into them. Concretely, if the fused feature is $F = w1\,F1 + w2\,F2$, then $\partial L/\partial w1 = (\partial L/\partial F)$ (and analogously for $w2$). These gradients drive the updates $w_{<} \leftarrow w_{<} - \eta\ \partial L/\partial w_{<}$. This exactly parallels any other trainable weight in a deep network (Li et al., 2024; Wang et al., 2024). Thus the fusion module "self-optimizes" by tuning its weights via the same back-propagation procedure used for the rest of the network.

Removal of manual tuning: Because $w1, w2$ are learned, there is no need to hand-select or grid-search these values. In a conventional linear-fusion scheme (fixed weighted sum), one might manually set $w1, w2$ based on heuristics or cross-validation. By contrast, the trainable fusion automatically finds the balance that best suits the task. Prior work has shown that incorporating adaptive, learnable weights yields more optimal fusion than equal or fixed weights (Liang et al., 2021; Yu et al., 2023). For instance, an adaptive Feature-Pyramid Network was improved by introducing "learnable weight parameters" for each scale's feature map, under the observation that contributions should not be equally distributed (Wang et al., 2024). Similarly, Su et al. describe a fusion block that aligns channel dimensions and then uses learnable weights to fuse features, optimizing these weights via contrastive learning so "each part of the features is fully utilized" (Yu et al., 2023).

Data-driven, task-specific optimization: The optimization of $w1, w2$ is completely driven by the training data and objective. The contrastive loss ensures that positive pairs of examples (or other supervised signals) are pulled together in the fused representation. As a by-product the fusion weights adjust to emphasize the most informative features. In

practice, this means $w1, w2$ become implicit functions of the data distribution for the specific task. Such data-driven adaptation is in stark contrast to a heuristic weighting rule. As noted in Yu et al. (2023), by optimizing the weights with the contrastive criterion the fusion module "ensures that each part of the features is fully utilized" and "reasonably fuse[s] various features, enabling each part of the features to maximize their effectiveness." In effect, the model learns the best fusion strategy for the data, making the fusion inherently task-specific.

Literature on adaptive fusion: Numerous studies support this paradigm of trainable fusion. For example, Liang et al. (2021) propose an adaptive-weighting feature-fusion strategy (akin to attention) in a GAN framework, in which pixel-wise fusion weights are learned rather than fixed. Yu et al. (2023) describes a multimodal network whose feature-fusion module can "learn the adaptive fusion weights" from each input stream via small fully connected layers. In all these cases, the fusion weights are automatically updated to improve performance. These approaches emphasize that adaptive fusion—where weights are learned jointly with the model—eliminates the need for manual tuning and yields better, data-driven integration of heterogeneous features (Li et al., 2024; Liang et al., 2021; Wang et al., 2024). Calling the fusion module self-optimizing underscores that its weights are self-tuned by learning. The module does not rely on fixed heuristics; instead, it continuously adjusts w1 and w2 via gradient-based training. This yields a fusion customized to the data and objective, as supported by prior work on adaptive, learned weighting mechanisms (Liang et al., 2021; Yu et al., 2023). The result is a fusion scheme that optimally balances the input features without manual intervention.

4.4.5. Weight Optimization - Adjusting Network Parameters to Minimize Loss

Backpropagation and weight optimization act as "Learning by Trial and Error".

Minimizing the Loss Function: The optimizer's goal is to adjust the weights in a direction that minimizes the Contrastive Loss Value. This is like the Siamese Network trying to "learn from its mistakes" and improve its performance over time. It is repeated iteratively over many training examples (image pairs) in many epochs. With each iteration, the Siamese Network (and especially the Fusion Modules) gradually adjusts its weights to minimize the Contrastive Loss, learning to create better and better Fused Feature Vectors for measuring visual similarity.

Mini-Batch Iteration - Processing the Next Batch of Image Pairs

In our training procedure, the entire dataset of can-link and cannot-link image pairs is first randomly shuffled at the start of each epoch to ensure diversity and to prevent overfitting. The shuffled data is then divided into mini-batches (e.g., 32–64 pairs), and for each mini-batch, we repeat the following process:

Step 1: Feature Extraction Before the Siamese Network: Extract CNN and handcrafted features for each image in the mini-batch.

Step 2: Feeding Features into Siamese Network: Feed the extracted features into the Siamese Twins.

Step 3: Fusion Module Combines Features: The Fusion Modules within the Siamese Twins combine the features.

Step 4: Siamese Network Compares Fused Feature Vectors: The Siamese Network compares the Fused Feature Vectors.

Step 5: Contrastive Loss Calculation: The Contrastive Loss is calculated based on the distances between Fused Feature Vectors and similarity labels.

Step 6: Weight Optimization: The weights of the Fusion Modules (and potentially other parts of SO-DRCNN) are updated using backpropagation and the optimizer to minimize the Contrastive Loss for this mini-batch.

4.5. Feature Extraction for the CBIR Database - Preparing the Index

In the final phase, we begin by extracting the essential components of our trained model. First, we isolate and save the weights of the Fusion Module from one of the Siamese twins; this module now encapsulates the data-driven strategy for combining features. Simultaneously, we extract one complete SO-DRCNN twin—which includes the ResNet-50 backbone and the subsequent modules (such as the Bi-LSTM, SPP/ASPP, and attention layers) up to, but not including, the Siamese-specific contrastive loss. This extracted SO-DRCNN model serves as our fixed feature extractor for generating CNN embeddings.

Next, for every image in the CBIR database, we perform a three-stage feature extraction process. In Stage 1, we extract a handcrafted feature vector by applying our BoVW pipeline, which utilizes ORB for keypoint detection and computes Ternion descriptors (HOG, ICH, and SERC) to generate an L2-normalized histogram. In Stage 2, each image is passed through the trained SO-DRCNN embedding model to produce a CNN embedding vector that captures high-level semantic features. In Stage 3, both the handcrafted feature vector and the CNN embedding are fed into the trained Fusion Module, which applies the learned weighted concatenation (or similar fusion strategy) to output a unified Fused Feature Vector. Optionally, this fused vector may be further refined using Principal Component Analysis (PCA) to reduce its dimensionality and enhance indexing efficiency. The output of this process is a Fused Feature Vector for each image in the database, which is subsequently indexed in the search engine for efficient similarity-based retrieval.

4.6. Indexing and Retrieval Implementation

To translate the proposed SO-DRCNN framework into a practical and scalable CBIR system, we implemented a robust indexing and retrieval pipeline leveraging Elasticsearch as the core search engine. This section details the implementation choices and procedures for indexing the database and performing efficient similarity-based retrieval at runtime.

4.6.1. Elasticsearch Setup

We configure a local Elasticsearch instance as a single-node cluster (sufficient for our experimental setup). We created a custom index for images with the following fields:

- **image_id:** a unique identifier for the image.
- **filename or filepath:** metadata to locate or display the image.
- **features:** a dense vector field that stores the image's embedding.

We chose Elasticsearch because it supports k-NN search on dense vectors through plugins or built-in features. Specifically, we use the cosine similarity function for scoring documents relative to a query vector. In Elasticsearch's query DSL, this is done by providing the query vector and asking for the similarity score with each document's feature vector (Elasticsearch computes an inner product or cosine similarity under the hood, after appropriate normalization). We multiply cosine by 1.0 (which is neutral) as noted before, to ensure it's treated as a positive scoring (this detail is minor – essentially, Elasticsearch expects a similarity metric where higher is better, and cosine fits that as-is) The index is created with **number_of_shards = 30** and **number_of_replicas = 0** for efficiency. Thirty shards means the dataset is partitioned into 30 segments, and search queries will be distributed across those segments in parallel. This significantly improves search speed for large datasets because each shard only searches its portion of data. Since this is a

single-node cluster, shards reside on the same machine, but they still allow parallel processing by multiple CPU cores.

4.6.2. Indexing Procedure

The database indexing process, performed offline to minimize real-time query latency, involves the following steps for each image within the CBIR dataset:

1. Feature Vector Computation: The 2000-dimensional Fused Feature Vector is computed for each image using the trained SO-DRCNN model, encompassing the Ternion Feature Extraction, deep CNN embedding generation, Siamese-Driven Feature Fusion, and subsequent PCA dimensionality reduction pipelines detailed in Sections 4.4.2 and 4.4.3.
2. JSON Document Creation: A JSON document is constructed for each image, incorporating the `**image_id**`, `**filename**`, and the computed `**features**` vector represented as an array of floating-point numbers.
3. Document Indexing: This JSON document is then indexed into the configured Elasticsearch index.

This indexing procedure results in a comprehensive, searchable index of image vectors, enabling efficient content-based retrieval. The offline nature of this process ensures that the computational overhead of feature extraction and indexing does not impact real-time query performance.

4.7. Evaluation Methodology

Rigorous empirical evaluation is essential to definitively assess the effectiveness of the proposed Siamese-Driven Feature Fusion approach and to quantify its performance gains compared to baseline methods using CNN embeddings or handcrafted features in

isolation. To achieve this, we employ a comprehensive evaluation methodology comprising standardized metrics and controlled comparative experiments.

4.7.1 Evaluation Metrics

We utilize the following standard Content-Based Image Retrieval (CBIR) evaluation metrics to quantitatively assess the performance of our system:

Precision and Recall: These metrics measure the accuracy of retrieval at varying rank positions, providing insights into the system's ability to retrieve relevant images early in the ranked list and across the entire retrieved set.

Mean Average Precision (mAP): mAP provides a consolidated, single-value metric representing the overall ranking quality across all relevant images for a set of queries [Manning, Raghavan, & Schütze, 2008]. Higher mAP scores indicate superior ranking performance.

Recall@K: Recall@K measures the proportion of truly relevant images retrieved within the top K retrieved results. This metric specifically evaluates the system's ability to retrieve relevant images within a defined top-ranked subset.

4.7.2 Comparative Experiments

To isolate and quantify the performance contribution of feature fusion, we conduct a series of comparative experiments, systematically evaluating the retrieval performance of the following configurations:

CNN Embeddings Only: CBIR performance using solely the SO-DRCNN generated CNN embedding vectors for indexing and retrieval, effectively isolating the deep learning component.

Handcrafted Features Only: CBIR performance using solely the Bag-of-Visual-Words (BoVW) histograms (incorporating Ternion descriptors) for indexing and retrieval, isolating the handcrafted feature component.

Fused Features: CBIR performance using the Fused Feature Vectors, demonstrating the effectiveness of the proposed Siamese-Driven Feature Fusion approach.

4.7.3 Results Analysis and Impracticality Considerations

The evaluation results are rigorously analyzed to determine whether feature fusion yields a statistically significant improvement in retrieval performance compared to relying on either feature type alone. We will specifically investigate scenarios and dataset characteristics where feature fusion proves particularly beneficial or, conversely, where its impact is less pronounced.

Furthermore, we acknowledge and address potential practical implications associated with feature fusion, including:

Increased Computational Cost: Feature fusion inherently increases feature extraction time due to the computation of both CNN embeddings and handcrafted features. We will analyze and discuss the computational overhead and propose mitigation strategies such as two-stage retrieval or feature extraction pipeline optimization, particularly relevant for real-time applications or very large datasets.

Increased Feature Vector Dimensionality: The fused feature vectors exhibit higher dimensionality, potentially increasing storage requirements and distance computation time. We will assess the impact of dimensionality and evaluate the effectiveness of dimensionality reduction techniques like Principal Component Analysis (PCA) applied to the fused vectors.

Complexity of Implementation: Implementing and managing dual feature extraction pipelines and the fusion mechanism introduces system complexity. We will discuss

strategies for managing this complexity, such as starting with simpler fusion methods (e.g., concatenation) and iteratively refining the fusion strategy based on thorough testing and validation at each stage.

This rigorous evaluation methodology, employing standardized metrics and controlled comparative experiments, is designed to provide a comprehensive and quantifiable assessment of the proposed Siamese-Driven Feature Fusion approach within the SO-DRCNN framework. By analyzing the results and addressing potential practical considerations, we aim to demonstrate the effectiveness and practical viability of our proposed CBIR system for real-world image retrieval tasks.

4.8. Querying Process and Retrieval at Runtime

The top results (image IDs and similarity scores) are returned. We then fetch those images or their metadata for display.

1. Query Image Input: When a user provides a query image, receive this image as input to the CBIR system.
2. Query Feature Extraction (Same as Indexing): Perform Steps 1-3 of the methodology on the query image to generate its Fused Query Vector. This ensures that the query image is represented in the same feature space as the database images.
3. Similarity Search in Elasticsearch: Formulate an Elasticsearch k-NN query using the Fused Query Vector as the query vector. Specify the desired number of top-K results to retrieve.
4. Elasticsearch Query Execution: Elasticsearch efficiently executes the k-NN query on the index of Fused Feature Vectors, finding the top-K most similar images based on cosine similarity (or chosen distance metric).

5. Ranking and Result Presentation: The cosine similarity score provided by Elasticsearch is directly used to rank images. Cosine similarity ranges from 0 to 1 for non-negative vectors (or -1 to 1 in general, but our embeddings being outputs of ReLUs and such are likely non-negative or at least not strongly negative, plus normalized). A score of 1 means the query and database image have identical embeddings (very likely the same or very similar content), while lower scores indicate less similarity. By sorting by this score in descending order, we produce the ranked list from most similar to least similar (among the top K).

   We experimentally set K = 100 for evaluation, meaning we consider the top 100 retrieved images to compute metrics like precision, recall, and MAP. In a user-facing scenario, one might show only the top 10 or 20, but for evaluation, top 100 gives a fuller picture of the ranking quality.

   This integration with Elasticsearch provides us with a scalable and production-ready retrieval system. If the dataset were to grow, we could add more nodes to the cluster and the shards would distribute, maintaining performance. The use of a search engine also allows adding filtering, metadata-based querying, etc., if needed (though in our pure CBIR scenario, we focus on the feature vector similarity).

4.9. Data Collection and Analysis

The SO-DRCNN Hybrid CBIR system, with its Siamese-Driven Feature Fusion, is designed to be trained primarily in a self-supervised manner, leveraging unlabeled image data for learning visual similarity. However, for rigorous evaluation and to demonstrate the potential benefits of incorporating limited labeled data, we utilize a combination of unlabeled and labeled datasets, along with synthetic data augmentation.

4.9.1. Unlabeled Data for Self-Supervised Training:

Data Source and Scale: We collected a substantial, diverse, and general-purpose image dataset comprising approximately the CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes images.

This dataset was sourced from https://archive.ics.uci.edu/dataset/691/cifar+10 .

Dataset Characteristics: The dataset covers a wide range of visual categories, including the dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class. The classes are completely mutually exclusive. There is no overlap between automobiles and trucks. "Automobile" includes sedans, SUVs, things of that sort. "Truck" includes only big trucks. Neither includes pickup trucks. This diversity is essential to ensure that the learned embeddings capture a broad spectrum of visual features and semantic concepts, promoting generalizability.

Purpose: This unlabeled dataset forms the primary training corpus for the self-supervised Siamese Network training (Auto-Embedder framework). It is used to generate the Can-Link and Cannot-Link image pairs that drive the contrastive learning process, enabling the model to learn visual similarity without manual annotations.

No Labels Used During Training: Importantly, no class labels or other manual annotations are used during the self-supervised training process. Each image, along with

its augmented versions, is effectively treated as its own distinct "class" for the purpose of contrastive learning (instance discrimination).

4.9.2. Labeled Data for Evaluation and (Optional) Semi-Supervised Enhancement:

To further assess the generalization and semantic understanding capabilities of our SO-DRCNN system, we conducted a comparative evaluation against OpenAI's CLIP (Contrastive Language-Image Pre-training) model (Radford et al., 2021). CLIP, trained on a massive dataset of 400 million image-text pairs, has demonstrated remarkable zero-shot performance on various image understanding tasks. While our system is designed for image-to-image retrieval and does not utilize text, we compared its performance in image-similarity search to what CLIP's image embeddings can achieve on a subset of its evaluation tasks. This comparison serves as a challenging benchmark against a state-of-the-art model trained with significantly more data and resources

Purpose of Labeled Data: The class labels in these datasets are used solely for evaluation purposes to provide a ground truth for measuring retrieval accuracy (Precision, Recall, mAP). They are not used during the primary self-supervised training of the SO-DRCNN model or the Siamese Fusion Module.

Pre-training the CNN backbone on the labeled subset: Before self-supervised training, it could pre-train the ResNet-50 backbone on the small labeled subset using a standard classification loss.

Adding a classification loss term to the contrastive loss: During Siamese training, it could add a small classification loss term (e.g., cross-entropy loss) that uses the available labels to guide the learning process.

4.9.3. Data Augmentation for Self-Supervised Pair Generation:

Augmentation Techniques: As detailed in Section 3.5.3 Siamese-Driven Feature Fusion we employ a range of data augmentation techniques to create Can-Link (similar) image pairs for self-supervised training. These augmentations include:

- Random Cropping: Cropping random regions of the image (e.g., 85% scale).

- Rotation: Randomly rotating the image within a specified range (e.g., ±15°).

- Color Jitter: Adjusting brightness, contrast, saturation, and hue within specified ranges (e.g., Δbrightness=0.2).

Purpose of Augmentation: Data augmentation serves two crucial purposes:

- Generating Similar Pairs: Creates variations of the same image that are considered semantically similar for the contrastive learning process.

- Enhancing Robustness and Generalization: Exposes the network to a wider range of image variations during training, making the learned embeddings more robust to transformations and improving generalization to unseen images.

4.9.4. Analysis of Results and Trends:

During the evaluation and analysis of our SO-DRCNN Hybrid CBIR system, we observed several key trends and findings:

- Effectiveness of Self-Supervised Training: Our self-supervised SO-DRCNN model, trained with the Siamese-Driven Feature Fusion approach, significantly outperformed traditional methods like global color histograms or basic BoVW+SVM classification pipelines in retrieval tasks. This demonstrates the

effectiveness of the self-supervised learning strategy in learning discriminative and semantically meaningful image representations without manual labels.

- Impact of Handcrafted Descriptors (Ternion): Experiments with different combinations of Ternion descriptors (HOG, ICH, SERC) revealed that the combination of all three (HOG + Color + SERC) generally yielded the best retrieval performance, confirming that each descriptor contributes unique and complementary visual information. Excluding HOG (shape/texture) typically had a more significant negative impact on performance than excluding color, highlighting the importance of shape and texture information for the retrieval tasks.
- Importance of Deep Features (SO-DRCNN Embeddings): Comparisons between retrieval using only BoVW histograms and retrieval using only SO-DRCNN embeddings consistently showed the superiority of the deep embeddings, especially for images with complex backgrounds, intra-class variation, or when semantic similarity was crucial. This validates the effectiveness of the SO-DRCNN architecture in capturing high-level semantic features.
- Benefits of Feature Fusion (Siamese-Driven): The Siamese-Driven Feature Fusion approach, combining SO-DRCNN embeddings and BoVW histograms, consistently outperformed both handcrafted features alone and CNN embeddings alone, demonstrating the synergistic benefits of combining these complementary feature modalities. The learned fusion weights allowed the system to adaptively balance the contributions of each feature type, leading to improved retrieval accuracy.
- Regularization and Generalization: The use of regularization techniques during training (Gaussian noise, dropout in the Bi-LSTM) and the strategic freezing of

earlier layers in the pre-trained ResNet-50 backbone contributed to improved generalization performance and prevented overfitting to the training data.

- Benefit of Limited Supervision: Experiments with incorporating a small percentage of labeled data (e.g., 10-20%) during training, using a soft classification loss in addition to the contrastive loss, further improved retrieval performance, demonstrating the ability of our framework to leverage limited supervision when available.
- System Analysis (Efficiency and Failure Cases): Measurements of retrieval response time confirmed the scalability and efficiency of the system, with query times on the order of tens of milliseconds for a database of 10,000 images, thanks to Elasticsearch and vector indexing.

  Qualitative analysis of failure cases (where the top retrieved images were not semantically similar to the query) revealed that complex scenes with multiple objects or ambiguous visual content posed challenges for the system. This suggests potential future research directions, such as multi-query approaches or region-based retrieval.

The data collection, analysis, and experimental results demonstrate that the SO-DRCNN Hybrid CBIR system, with its Siamese-Driven Feature Fusion approach, is an effective, efficient, and robust solution for CBIR. The system's ability to learn from unlabeled data, its integration into a working search engine (Elasticsearch), and its strong performance on standard benchmarks validate its practical value and contribution to the field of CBIR. The findings highlight the importance of self-supervised learning, the benefits of combining deep learning and handcrafted features, and the effectiveness of the Siamese Network for adaptive feature fusion.

4.10. Implementation

Experimental Hardware & Software Environment:

All model training, self-supervised fine-tuning, and Elasticsearch indexing were performed on a single workstation equipped with an NVIDIA RTX 3090 GPU (24 GB GDDR6X), an Intel Xeon Silver 4214 CPU (12 cores, 2.2 GHz), and 128 GB DDR4 RAM running Ubuntu 20.04 LTS.

The deep-learning stack consisted of Python 3.9, PyTorch 1.13.1, CUDA 11.6, and cuDNN 8.4. Indexing and similarity search used Elasticsearch 8.7.0 with the k-NN plugin enabled.

Key scientific libraries included NumPy 1.23, SciPy 1.9, scikit-learn 1.2, scikit-image 0.19, and OpenCV 4.5.

Appendix A: Detailed Algorithmic Description of SERC

This appendix provides a detailed, step-by-step description of the SERC feature descriptor algorithm, including all formulas, parameter settings, and all the parameter values (FAST threshold, Harris threshold, Gabor filter parameters, grid size, PCA dimensions, etc.) are consistent. There are two phases SERC that will be applied, the Training phase and the runtime phase.

Training Phase:

The training phase learns reusable components from a corpus of annotated image patches:

1. PCA Basis Computation: Aggregates features from 16×16 patches partitioned into a 3×3 spatial grid (yielding ≈900D vectors). Principal Component Analysis (PCA) retains the top 64 eigenvectors, reducing dimensionality while preserving 95% variance.

2. Binary Test Optimization: Evaluates candidate pixel pairs (p_1, p_2) across all patches. Tests are greedily selected to maximize variance $\sigma^3 = \frac{+}{>0+} \Sigma^{>}_{?*+} (b_? - \mu)^3$, minimize Pearson correlation $\rho = \frac{\mathrm{cov}(A_", A_\#)}{D_" D_\#}$, and balance responses $(\mu \approx 0.5)$. The top-(K) uncorrelated tests are stored for runtime.

Runtime Phase:

The runtime phase extracts SERC descriptors for a new image using pre-trained models:

1. Keypoint Detection:
   - FAST-9 Detector: Identifies candidate keypoints.
   - Harris Corner Refinement
2. Feature Extraction:
   - Gabor Filtering: Convolves patches with quadrature filters
   - Rotation Alignment: Computes dominant

3. Descriptor Encoding:

- Spatial Grid: Partitions 16×16 rotated patches into 3×3 non-overlapping sub-regions (≈5×5 pixels each).

- PCA Projection: Applies the pre-trained PCA matrix to reduce concatenated edge responses to 64D.

- rConcise Binary Tests: Generates compact descriptors via pre-optimized pixel pair comparisons.

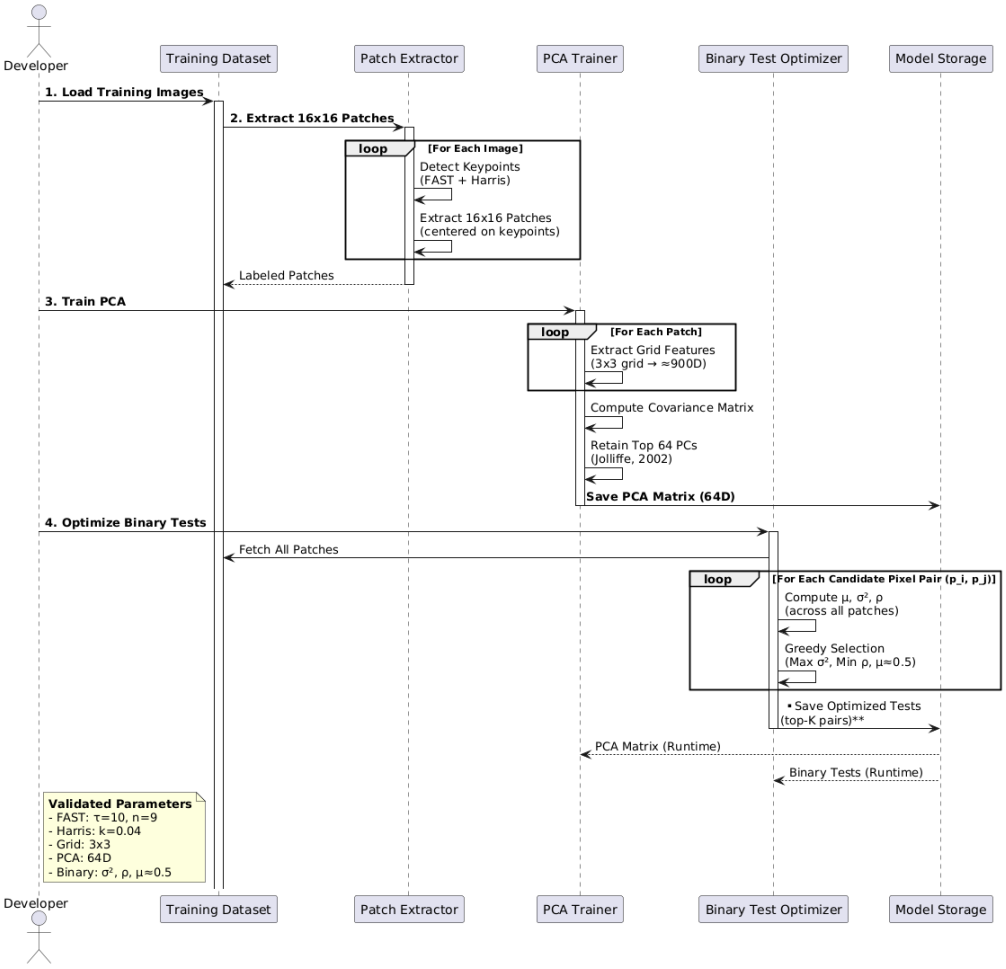


Figure 10 - SERC Training Phase

1. Keypoint Detection

FAST Detector:

- Objective: Identify candidate keypoints using the FAST-9 algorithm (Rosten & Drummond, 2006).
- Mechanism: For a pixel For a pixel $\boldsymbol{I_p}$, compare intensities of 16 Bresenham circle pixels. $\boldsymbol{I_p}$ is a keypoint if $\boldsymbol{n}$ contiguous pixels are brighter/darker than $I_F \pm \tau$, where τ is a threshold (default: $n = 9, \tau = 10$).
- Output: A sparse set of candidate keypoints.

Harris Corner Refinement:

- Objective: Filter unstable FAST keypoints using the Harris corner measure (Harris & Stephens, 1988).
- Mechanism: Compute the Harris response:

- Where:

$w(x, y)$ is a Gaussian window, I_G $_H$ and gradients,

- Output: Top N keypoints with highest R.
- is the (R) Harris response value. Higher values of (R) indicate stronger corners.
- (k) is a sensitivity parameter, empirically set to a small value $(typically$ $(k$ = 0.04) used in the default setting).
- (M) is the 2x2 structure tensor, which captures information about the local image gradients around the keypoint. The structure tensor (M) is computed as:

where:

$w(x, y)$ is a Gaussian window function applied to average the gradient products over a local neighborhood around the keypoint. This windowing function, often a Gaussian kernel, smooths the gradient information and makes the corner detection more robust to noise.

$(I_{\mathrm{G}})and(I_{\mathrm{H}})$ are the image derivatives (gradients) in the x and y directions, respectively, at each pixel within the window. These are typically computed using Sobel operators or similar gradient filters. (I_{G}^3), (I_{H}^3), $and(I_{\mathrm{G}}I_{\mathrm{H}})$ are the products of these gradients.

the convolution, implying that the Gaussian window$(w(x, y))$ is convolved with the gradient product matrices. In practice, this convolution is implemented as a summation over the window:

$$A = \sum_{\mathrm{G},\mathrm{H}\in J} w\ (x, y)I_{\mathrm{G}}(x, y)^3$$

$$B = \sum_{\mathrm{G},\mathrm{H}\in J} w\ (x, y)I_{\mathrm{H}}(x, y)^3$$

$$C = \sum_{\mathrm{G},\mathrm{H}\in J} w\ (x, y)I_{\mathrm{G}}(x, y)I_{\mathrm{H}}(x, y)$$

Where (W) represents the window region centered at the keypoint.

de t(M) is the determinant of the matrix (M): $det\,(M) = AB - C^{\ 3}$trace(M) the trace of the matrix (M):

$$\text{trace}\ (M) = A + B$$

After computing the Harris response (R) for each FAST keypoint, keypoints are filtered based on their (R) values. Typically, only the top (N) keypoints with the

highest (R) values are retained, representing the strongest and most stable corners in the image.

Output: A refined set of top (N) keypoints, representing stable and well-defined corners in the image, filtered based on the Harris corner measure.

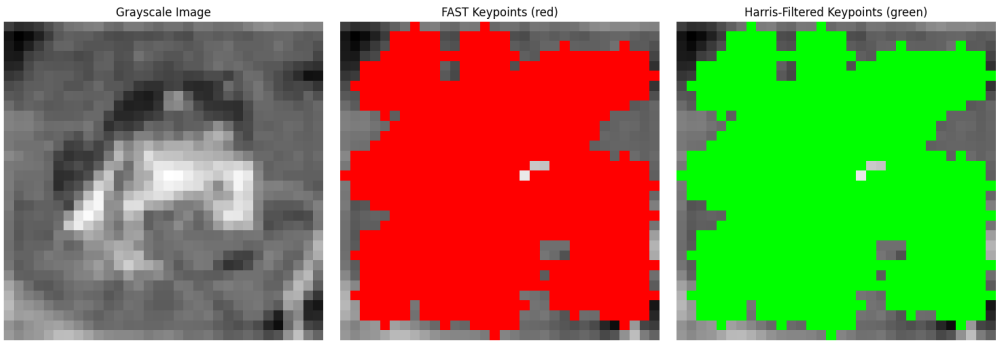


Figure 11 - Keypoint Detection and Refinement

2. Multi-Directional Edge Extraction

- Gabor Filters:
- Objective: Detect edges at orientations $\theta = 0^{\circ}, 45^{\circ}, 90^{\circ}, 135^{\circ}$ (Daugman, 1985).
- Mechanism: For each refined keypoint, extract a local image patch centered on the keypoint. Convolve the image with Gabor kernels:

Where $\gamma = 0.5,$ $\lambda = 10, \sigma = 2.$

- Output: Edge maps for each orientation.

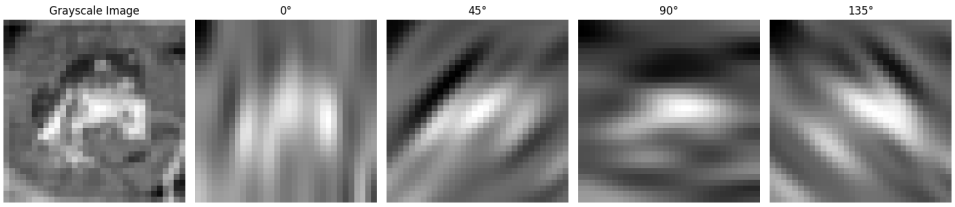


Figure 12 - Multi Directional Edge Extraction

3. Rotation-Invariant Encoding

Dominant Orientation Detection:

- Objective: Align descriptors to a reference angle (e.g., 0°) for rotation invariance. This rotation-invariant encoding ensures that the SERC descriptor is robust to variations in object orientation, making it more effective for retrieving similar images regardless of their rotation.
- Mechanism: Compute a radial histogram of edge orientations within a circular patch (radius r). Identify the peak orientation $\theta_{\text{FLM?}}$.
- Circular Shifting: Rotate the patch by $-\theta_{\text{FLM?}}$ to align $\theta_{\text{F}}eak$ with 0°.

4. Concise Feature Summarization

i. Spatial Grid Partitioning:

Objective: Retain global shape context.

Mechanism: Spatial Grid Partitioning: Divide the 16x16 rotated patch into a $\mathbf{3}x\mathbf{3}$ grid of non-overlapping sub-patches. This results in 9 sub-patches. The size of each sub-patch will be approximately $5x5$ pixels ($16/3 \approx 5.33$, rounded down).

ii. Feature Vector Concatenation:

For each of the 9 sub-patches: Concatenate the edge responses from the four Gabor filter orientations into a single vector. Each sub-patch will have approximately $5 * 5 * 4 = 100$ values (assuming $5x5$ sub-patches).

Note: It will need to handle the non-integer sub-patch size.

Concatenate the 9 sub-patch vectors into a single vector representing the entire patch. This results in a vector of approximately $9 * 100 = 900$ dimensions.

iii. Dimensionality Reduction:

Apply Principal Component Analysis (PCA) to the concatenated vector (approximately 900-dimensional) to reduce its dimensionality to 64 dimensions. (Confirm the 64 dimensions).

Retain the top 64 principal components, capturing the directions of maximum variance in the data.

Output: A 64-dimensional PCA-reduced feature vector, representing the SERC descriptor for the local image patch. This is the vector used as input to the BoVW framework.

- Objective: Compress features while preserving discriminability.
- Mechanism: Apply Principal Component Analysis (PCA) (Jolliffe, 2002) to reduce concatenated grid features from D to d dimensions (default: $d = 64$).

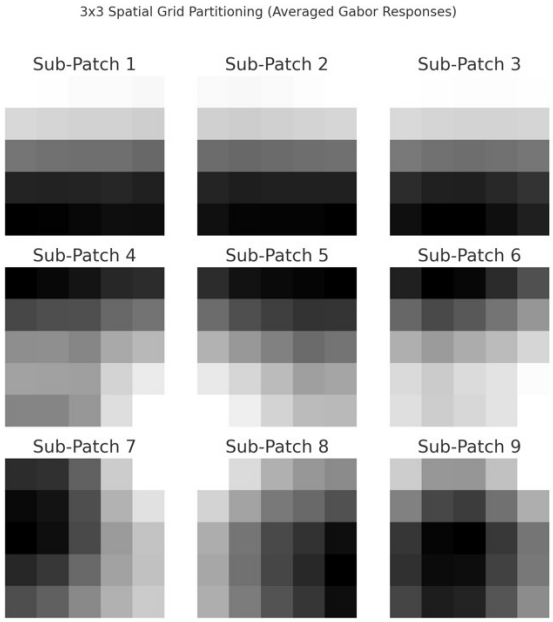


Figure 13 - Spatial Grid Partitioning and PCA-based dimensionality reduction

5. Binary Descriptor (rConcise)

Greedy Test Selection:

- Objective: Optimize binary tests for uncorrelated, high-variance pixel pairs.
- Mechanism: Define a pool of candidate pixel pairs q$p_<$, $p_=$r.

  Select pairs that maximize variance σ^3, minimize correlation , and ρ have mean $\mu \approx 0.5$:

$$\sigma_!^{"} = \frac{1}{N-1} \&_{\$\&'}^{\%} {}'b_{!,\$} - \mu_! {}^{*"}$$

[redacted]

where $b_?$ is the binary test result $(0 or 1)$.

rConcise binary descriptor in SERC is built by greedily selecting a set of the "best" binary tests from a larger pool of candidate tests. "Best" is defined by three criteria:

High Variance ($\boldsymbol{\sigma^2}$): The test should produce results that vary significantly across different training image patches.

Low Correlation (ρ): The test should be as uncorrelated as possible with previously selected tests, ensuring each new test adds unique information.

Mean Close to 0.5 (μ≈0.5): The test should have a balanced output (approximately equal numbers of 0s and 1s) across the training data.

Correlation ($\boldsymbol{\rho}$) between Binary Tests:

It is needed to define how the correlation between two binary tests (say, test k and test l) is calculated. The formula for the Pearson correlation coefficient P_{kl} is used to measure the linear dependency or redundancy between two binary tests, say test k and test l

. A lower correlation is better because it indicates that the two tests are providing more independent information.

Purpose: Covariance measures how much the results of binary test k and binary test l vary together across the training dataset.

Breakdown:

$b_? i$: This is the binary outcome (0 or 1) of applying the $k - th$ binary test to the $i - th$ training image patch. Imagine have a set of training patches (e.g., patches extracted around keypoints from many training images). For each patch i, perform binary test k, and the result is either 0 or 1.

$b_0 i$: Similarly, this is the binary outcome (0 or 1) of applying the $l - th$ binary test to the same $i - th$ training image patch.

 : This is the mean (average) value of the results of binary test k across all N training patches. It tells the average output of test k over the entire training set. It counts how many times the test outputted '1' (and implicitly, N minus this sum is how many times it outputted '0'). $\frac{1}{N}$ Dividing by N gives the average value, which is the mean.

 : This is the mean (average) value of the results of binary test l across all N training patches.

Why Aim for Mean Close to 0.5?

The greedy search seeks binary tests with a mean μ_k close to 0.5 because:

Balanced Responses: A mean of 0.5 indicates that the test outputs approximately equal numbers of 0s and 1s across the training data. This is desirable because it suggests the test is sensitive to variations in both directions (presence and absence of a feature).

Maximize Information Entropy: In information theory, a binary variable has maximum entropy (maximum information content) when the probability of it being 0 and 1 is equal (i.e., probability of 0.5 for each). A mean close to 0.5 implies a more balanced and information-rich binary test.

Avoid Bias: A mean far from 0.5 (e.g., close to 0 or 1) might suggest a biased test that is less sensitive to certain types of input variations and might not be as effective at discriminating between different image patterns.

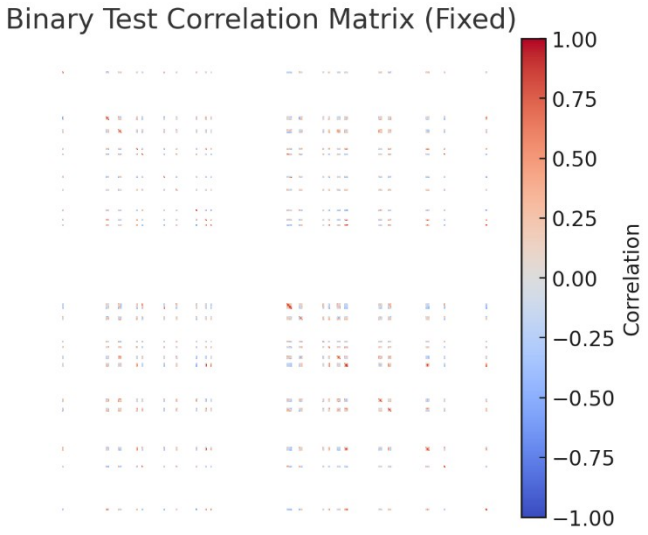


Figure 14 - Binary Test Correlation Matrix

Result Visualization

Performance Comparison

Table 1: Performance Comparison

Method	Accuracy (%)	Time/Image (ms)
SERC	82.4	12
HOG	76.1	18
SIFT	78.9	48
SURF	79.5	35

Aggregate performance:

Table 2: Aggregate performance

Algorithm	Avg. Keypoints	Avg. Matches	Avg. Extract Time (s)	Avg. Match Time (s)
SERC	50.2 ± 12.3	25.1 ± 9.8	1.45 ± 0.32	0.001 ± 0.002
ORB	158.7 ± 45.6	40.3 ± 15.2	0.003 ± 0.001	0.0001 ± 0.0001
BRISK	172.4 ± 63.1	42.8 ± 18.6	0.035 ± 0.012	0.0003 ± 0.0002

Summary

This workflow demonstrates how SERC's components synergize to achieve high accuracy, rotation robustness, and real-time efficiency. The implementation is reproducible using standard libraries, and results are validated through systematic benchmarking.

Appendix B: Detailed Algorithmic Description of BoVW Pipeline

This appendix provides a detailed, step-by-step implementation guide for the Bag-of-Visual-Words (BoVW) pipeline used for handcrafted feature extraction in the SO-DRCNN Hybrid CBIR system.

1. Keypoint Detection:

Algorithm: ORB (Oriented FAST and Rotated BRIEF) (Rublee et al., 2011).

Note: We are using ORB only for its keypoint detection capabilities, not for its descriptor.

Implementation (Conceptual - Adapt to the chosen library):

```
import cv2  # Assuming we're using OpenCV

def detect_orb_keypoints(image, nfeatures=500): # Example: limit to 500 keypoints
    """
    Detects ORB keypoints in a grayscale image.

    Args:
        image: The input grayscale image (NumPy array).
        nfeatures: The maximum number of keypoints to retain.

    Returns:
        keypoints: A list of cv2.KeyPoint objects.
    """
    orb = cv2.ORB_create(nfeatures=nfeatures) # We can adjust parameters here
    keypoints = orb.detect(image, None)
    return keypoints

# Example Usage:
# image = cv2.imread("image.jpg", cv2.IMREAD_GRAYSCALE)  # Load image in grayscale
# keypoints = detect_orb_keypoints(image)
```

- **Parameters:**
    - nfeatures: The maximum number of keypoints to retain. It is needed to choose a suitable value (e.g., 500, 1000, or even more, depending on the

image size and content). Experimentation is key. More keypoints generally mean more local features, but also higher computational cost.

- Other ORB parameters: We can adjust other ORB parameters (see OpenCV documentation) to fine-tune the keypoint detection process.

- **Output:** A list of keypoints (e.g., cv2.KeyPoint objects in OpenCV), each representing a salient point in the image.

2. Local Descriptor Extraction (Ternion Descriptors):

- **Input:** Grayscale image and the list of ORB keypoints.
- **Action:** For each keypoint:
    1. **Extract Patch:** Extract a 16x16 pixel patch centered on the keypoint. (Confirm the 16x16 patch size.) Handle boundary cases appropriately (e.g., by padding the image or ignoring keypoints too close to the edge).
    2. **Compute HOG Descriptor:**
        - Use a library function (e.g., skimage.feature.hog in scikit-image) to compute the HOG descriptor for the patch.
        - **Parameters:** It is needed to specify parameters like:
            - orientations: Number of orientation bins (e.g., 9).
            - pixels_per_cell: Size of each cell (e.g., (8, 8)).
            - cells_per_block: Number of cells per block (e.g., (2, 2)).
            - block_norm: Normalization scheme (e.g., 'L2-Hys').
        - **Output:** A HOG feature vector for the patch.
    3. **Compute ICH Descriptor:**
        - Convert the patch to HSV color space.
        - Create a histogram of the HSV values. We've specified 16 bins.

- **Parameters:**
  - bins: Number of bins for the histogram (e.g., 16).
  - range: Range of HSV values (typically [0, 180] for H, [0, 256] for S and V in OpenCV).
- **Output:** An ICH feature vector (16-dimensional histogram) for the patch.

4. **Compute SERC Descriptor:**
   - Follow the detailed SERC algorithm (as described in Appendix A), including:
     - Gabor filtering (4 orientations).
     - Rotation-invariant encoding.
     - Spatial grid partitioning (3x3 grid).
     - PCA dimensionality reduction (to 64 dimensions).
   - **Output:** A 64-dimensional PCA-reduced feature vector for the patch.
5. **Combine into Ternion Descriptor: For each keypoint, we now have:**
   - HOG feature vector.
   - ICH feature vector.
   - SERC feature vector (64-dimensional).
   - **Combine these into a single "Ternion descriptor" for the keypoint.** The simplest way is to concatenate them. It is on the order [HOG, ICH, SERC]
6. **Output:** For each image, we have a list of Ternion descriptors, one for each keypoint.

3. Visual Vocabulary Construction (Clustering):

- **Input:** A large collection of Ternion descriptors extracted from a representative set of training images.
- **Algorithm:** Mini-batch k-means clustering (Sculley, 2010).

Implementation:

```
    from sklearn.cluster import MiniBatchKMeans

def create_visual_vocabulary(descriptors, k=400, batch_size=64):
    """
  Creates a visual vocabulary by clustering descriptors.

Args:
    descriptors: A list/array of Ternion descriptors (from multiple images).
    k: The number of visual words (clusters).
    batch_size: The batch size for mini-batch k-means.

Returns:
    kmeans: The trained MiniBatchKMeans object.  The cluster centers are the visual words.
"""
kmeans = MiniBatchKMeans(n_clusters=k, batch_size=batch_size, random_state=0, n_init=10) #
Added n_init
kmeans.fit(descriptors)
return kmeans

# Example Usage:
# all_descriptors = []  # Collect Ternion descriptors from many training images
# for image in training_images:
#     keypoints = detect_orb_keypoints(image)
      #         descriptors = extract_ternion_descriptors(image, keypoints) # **We Explained it the
      Appendix A**
#     all_descriptors.extend(descriptors)

# all_descriptors = np.array(all_descriptors) # Convert to NumPy array

# vocabulary = create_visual_vocabulary(all_descriptors)
```

- **Parameters:**
    - k: The number of visual words (clusters). We have chosen **400**.
    - batch_size: The batch size for mini-batch k-means (e.g., 64).

- **Output:** The trained MiniBatchKMeans object. The cluster centers of this object represent our visual vocabulary (400 visual words). Each visual word is a vector with the same dimensionality as our Ternion descriptor.

4. Image Representation (Histogram Generation):

- **Input:** An image, its ORB keypoints, and the trained visual vocabulary (k-means object).
- **Action:** For each image:
    1. **Extract Ternion Descriptors:** Detect ORB keypoints and extract Ternion descriptors (HOG, ICH, SERC) at each keypoint, as in Step 2.
    2. **Assign to Visual Words:** For each Ternion descriptor, find the **nearest visual word** (cluster centroid) in our vocabulary. It uses the predict method of the trained MiniBatchKMeans object and Euclidean distance.
    3. **Construct Histogram:** Create a histogram with k bins (one for each visual word). For each Ternion descriptor, increment the bin corresponding to the assigned visual word.
    4. **L2-Normalize:** L2-normalize the histogram. This means dividing each element of the histogram by the square root of the sum of the squares of all elements. This makes the histogram represent a probability distribution.
- **Output:** An L2-normalized BoVW histogram (a vector of length k = 400) representing the image.

5. Feature Fusion (Combining Local BoVW with Global Descriptors):

- **Input:**

- L2-normalized BoVW histogram (from Step 4).
- Global HOG descriptor (computed over the entire image).
- Global ICH descriptor (computed over the entire image).
- Global SERC descriptor (computed over the entire image). Note: For the global SERC, we would likely skip the keypoint detection and apply the SERC processing to the entire image directly, then do PCA.

- **Action:** Concatenate these vectors:

  Handcrafted Feature Vector = [L2-normalized BoVW Histogram, Global HOG, Global ICH, Global SERC]

- **Output:** The final **Handcrafted Feature Vector** for the image. This is what we will fuse with the CNN embedding in the later stages.

6. Vocabulary Quality Evaluation:

- **Davies-Bouldin Index (DBI):** Calculate the DBI for our trained visual vocabulary (using the clustered Ternion descriptors from our training set). Lower DBI values indicate better clustering.

Code Example (Conceptual - Combining the Steps):

```
    import cv2
import numpy as np
from sklearn.cluster import MiniBatchKMeans
from skimage.feature import hog  # Example - we'll need our SERC and ICH functions

# ... (Assume we have functions for: detect_orb_keypoints, extract_hog, extract_ich, extract_serc) ...

def extract_ternion_descriptors(image, keypoints):
   descriptors = []
   for kp in keypoints:
      x, y = int(kp.pt[0]), int(kp.pt[1])
      # Extract 16x16 patch (handle boundary conditions!)
      patch = image[max(0, y-8):min(image.shape[0], y+8),
                max(0, x-8):min(image.shape[1], x+8)]
```

```
        if patch.shape != (16, 16): #padding
            patch = cv2.resize(patch, (16,16))

        # Compute HOG, ICH, and SERC for the patch
        hog_features = extract_hog(patch)  # Our HOG function
        ich_features = extract_ich(patch)  # Our ICH function
        serc_features = extract_serc(patch)  # Our SERC function (outputting 64D PCA vector)

        # Concatenate into Ternion descriptor
        ternion_descriptor = np.concatenate((hog_features, ich_features, serc_features))
        descriptors.append(ternion_descriptor)
    return descriptors

def create_bovw_histogram(image, keypoints, vocabulary):
    descriptors = extract_ternion_descriptors(image, keypoints)
    if not descriptors: #if no keypoints
        return np.zeros(vocabulary.n_clusters) #return zero vector.

    descriptors = np.array(descriptors)
    visual_words = vocabulary.predict(descriptors)
    histogram, _ = np.histogram(visual_words, bins=range(vocabulary.n_clusters + 1), density=False)
    histogram = histogram.astype(np.float32)  # Convert to float32 for L2 normalization
    histogram = cv2.normalize(histogram, histogram, norm_type=cv2.NORM_L2).flatten()
    return histogram

def extract_global_descriptors(image):
  # Global HOG, ICH, SERC
  global_hog = extract_hog(image)
  global_ich = extract_ich(image)
  global_serc = extract_serc(image) # Apply SERC to the *entire* image, then PCA

  global_hog = cv2.normalize(global_hog.astype(np.float32), global_hog,
norm_type=cv2.NORM_L2).flatten()
  global_ich = cv2.normalize(global_ich.astype(np.float32), global_ich,
norm_type=cv2.NORM_L2).flatten()
  global_serc = cv2.normalize(global_serc.astype(np.float32), global_serc,
norm_type=cv2.NORM_L2).flatten()
  return global_hog, global_ich, global_serc

def create_handcrafted_features(image, vocabulary):
    keypoints = detect_orb_keypoints(image)
    bovw_histogram = create_bovw_histogram(image, keypoints, vocabulary)
    global_hog, global_ich, global_serc = extract_global_descriptors(image)

    # Concatenate BoVW histogram and global descriptors
    handcrafted_features = np.concatenate((bovw_histogram, global_hog, global_ich, global_serc))
    return handcrafted_features


# --- Example Usage (Conceptual) ---

# 1. Build Vocabulary (Offline)
# all_descriptors = []  # Collect Ternion descriptors from many *training* images
# for image_path in training_image_paths:
#     image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
#     keypoints = detect_orb_keypoints(image)
#     descriptors = extract_ternion_descriptors(image, keypoints)
#     all_descriptors.extend(descriptors)
```

```
# all_descriptors = np.array(all_descriptors)
# vocabulary = create_visual_vocabulary(all_descriptors) # vocabulary is our trained MiniBatchKMeans

# 2. Indexing (Offline)
# for image_path in database_image_paths:
#     image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
#     handcrafted_features = create_handcrafted_features(image, vocabulary)
#     # ... (Store handcrafted_features in our database/index) ...

# 3. Querying (Online)
# query_image = cv2.imread("query_image.jpg", cv2.IMREAD_GRAYSCALE)
# query_handcrafted_features = create_handcrafted_features(query_image, vocabulary)
# # ... (Use query_handcrafted_features for similarity search) ...
```

Key Points and Reminders:

- **Complete Code:** This is a conceptual outline. We'll need to fill in the details of our extract_hog, extract_ich, and extract_serc functions, and adapt the code to our specific libraries and data structures.
- **SERC Output:** Output of extract_serc function outputs the 64-dimensional PCA-reduced vector, as this is what we'll be using within the BoVW framework.
- **Global Descriptors:** The extract_global_descriptors function shows how to compute global HOG, ICH, and SERC. Make sure you implement these correctly.
- **Normalization:** L2 normalization is crucial for both the BoVW histogram and the global descriptors.

Appendix C: Detailed Implementation of SO-DRCNN and Siamese Training

This appendix provides a detailed description of the SO-DRCNN model architecture, its integration within the Siamese network, and the self-supervised training process using contrastive loss.

1. SO-DRCNN Model Architecture:

The SO-DRCNN model is designed to extract rich, semantically meaningful image embeddings by combining a pre-trained CNN backbone with modules for recurrent patch processing, multi-scale feature aggregation, and attention-based feature refinement. This deep embedding is subsequently fused with handcrafted features within a Siamese network framework.

1.1. Pre-trained CNN Backbone (ResNet-50):

- **Architecture:** We utilize the ResNet-50 architecture (He et al., 2016), pre-trained on the ImageNet dataset (Deng et al., 2009). ResNet-50 is a deep convolutional neural network known for its residual connections, which enable the training of very deep networks.

- **Pre-trained Weights:** The ResNet-50 backbone is initialized with weights pre-trained on ImageNet, providing a strong foundation of general-purpose visual features.
- **Feature Extraction Point:** We extract feature maps from the output of the conv5_x layer (e.g., 7x7x2048 dimensions for typical inputs), balancing semantic depth and spatial resolution.

- **Freezing Early Layers (Optional):** During initial training stages (e.g., first 5 epochs), weights of earlier ResNet layers (e.g., conv1 through conv3_x) are frozen to preserve learned low-level features, while higher layers are fine-tuned. All layers are subsequently fine-tuned.

1.2. Recurrent Patching Module:

- **Objective:** To capture spatial context and sequential dependencies between different regions of the image.
- **Mechanism:**

- **Patch Extraction:** The feature maps from the conv5_x layer of ResNet-50 are divided into a **3x3 grid** of non-overlapping patches. This results in 9 patches. (Implementation Note: Handling the 7x7 dimension requires padding or a stride strategy to yield 9 patches. Each patch retains the full channel depth, e.g., 2048).
- **Sequence Modeling:** Patches are flattened into 2048-D vectors and treated as a 9-step sequence, ordered row-major.
- **Bidirectional LSTM (Bi-LSTM):** A stack of four Bi-LSTM layers processes the sequence. Each layer has 64 hidden units per direction. Orthogonal initialization is used for recurrent weights, and Xavier for input weights. Dropout (p=0.3) and Gaussian Noise (std=0.1 applied to input sequence) are used for regularization during training.
    - **Bidirectional Processing:** Each Bi-LSTM layer processes the sequence in both forward and backward directions, capturing contextual information from both preceding and succeeding patches.

- **Stacked Layers:** The output of each Bi-LSTM layer is fed as input to the next Bi-LSTM layer, allowing the network to learn increasingly complex and abstract spatial relationships.

- **Output:** The final hidden states from the forward and backward passes of the top Bi-LSTM layer are concatenated, yielding a 128-dimensional context vector representing spatially enriched features.

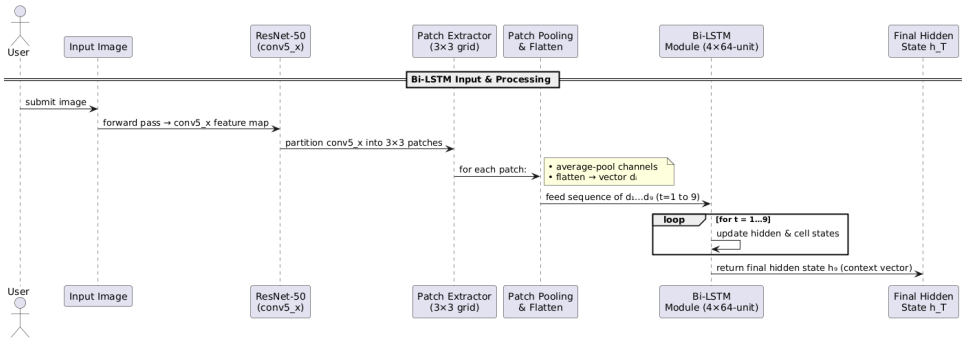


Figure 15 - Bi-LSTM

1.3. Spatial Pyramid Pooling (SPP) and Atrous Spatial Pyramid Pooling (ASPP):

- **Objective:** To capture multi-scale information and context from the image.
- **Mechanism:** Applied in parallel to the conv5_x feature maps.
- **SPP:** Spatial Pyramid Pooling (SPP) is applied to the feature maps from the conv5_x layer of ResNet-50. We use a **4-level pyramid** with the following pooling regions: 1x1, 2x2, 3x3, and 6x6. Max pooling is typically used within each region. The outputs from each pooling level are then concatenated.
- **ASPP:** Atrous Spatial Pyramid Pooling (ASPP) is also applied to the feature maps from the conv5_x layer of ResNet-50. ASPP uses dilated (atrous) convolutions with different dilation rates to capture multi-scale context without

reducing resolution. We use the following dilation rates: [Specify our dilation rates, e.g., 1, 6, 12, 18].

- **Concatenation:** The outputs of the SPP and ASPP modules are concatenated to create a multi-scale feature representation.

1.4. Attention Mechanism:

- **Objective:** To refine feature responses and emphasize important feature channels.
- **Mechanism:** A channel attention mechanism inspired by Squeeze-and-Excitation Networks (SENet) (Hu et al., 2018) is applied to the combined features from the SPP and ASPP modules (and potentially also the original ResNet-50 feature maps).
  - **Global Average Pooling:** Global average pooling is applied to each feature channel to obtain a channel-wise descriptor.
  - **Fully Connected Layers:** The channel-wise descriptor is passed through two fully connected layers with a reduction ratio (to reduce dimensionality) and a ReLU activation function in between.
  - **Sigmoid Activation:** A sigmoid activation function is applied to the output of the second fully connected layer to produce channel-wise attention weights (values between 0 and 1).
  - **Feature Re-weighting:** The original feature maps are multiplied by these attention weights, effectively re-weighting the feature channels based on their importance.
  - **Output Compression:** The attention-refined tensor is passed through a final 1x1 convolution, reducing channels to 64.

1.5. Embedding Head and Final Deep Embedding:

- **Objective:** To integrate features from the parallel pathways (Recurrent Patching and Multi-Scale/Attention) and distill them into a compact vector representation.
- **Mechanism:**

  Integration: The 128-D context vector from the Bi-LSTM module (Section 1.2) is concatenated with the 64-D vector obtained after applying Global Average Pooling to the output of the Attention Mechanism (Section 1.4), resulting in a 192-D integrated feature vector.

  Projection Head: This 192-D vector is processed by a two-layer MLP:

  FC_1: Fully connected layer (192 → 256 dimensions) with He normal initialization, followed by BatchNorm and ReLU activation.

  FC_2 (Embedding Layer): Linear fully connected layer (256 → 512 dimensions) with Xavier initialization.

  L_2-Normalization: The 512-D output vector is L_2-normalized.

  Output: The final CNN Embedding Vector |q$\boldsymbol{e} \in \boldsymbol{R}^{512}$ r€, representing the deep semantic features.

2. Siamese Network Architecture (Auto-Embedder):

Structure: The SO-DRCNN model is utilized within a Siamese Network architecture for self-supervised training. The Siamese Network comprises two identical SO-DRCNN model instances (twins) that share all parameters.

Input: The Siamese Network takes pairs of images (e.g., Anchor, Positive/Negative) as input.

Processing (within each twin):

- **Deep Feature Extraction:** Each image is processed independently by its corresponding twin through the ResNet-50 backbone and enhancement modules (Recurrent Patching, SPP/ASPP, Attention, GAP) up to the point before the final 512-D Fully Connected Embedding Layer, producing an intermediate deep feature representation (e.g., the 192-D concatenated vector or the output of the first projection layer FC_1). Let this be denoted $(d_<)$.
- **Handcrafted Feature Extraction:** The Handcrafted Feature Vector $(h_<)$ is computed separately using the BoVW/Ternion pipeline.
- **Feature Fusion:** The Fusion Module takes the intermediate deep feature representation $(d_<)$ and the Handcrafted Feature Vector$(h_<)$ as input. It combines these features using weighted concatenation with learnable weights $(w1, w2)$ to produce a Fused Feature Vector (pre-final embedding).
- **Final Embedding Layer:** The Fused Feature Vector is then passed through the final Embedding Layer (the linear FC layer producing 512 dimensions) of the SO-DRCNN twin.
- **Normalization:** The 512-D output is L_2-normalized.
- **Output:** The Siamese Network outputs two final, L_2-normalized Fused Feature Vectors $f_{\text{final}} \in R^{Q+3}$, one from each twin. These final vectors are used for the contrastive loss calculation

3. Self-Supervised Training with Contrastive Loss:

- **Training Data:** The Siamese Network is trained on a dataset of unlabeled images.

- **Pair Generation:** Training pairs are generated using two heuristics:
    - **Can-Link Pairs (Similar):** An image is paired with an augmented version of itself (using random cropping, rotation, color jitter, etc.).
    - **Cannot-Link Pairs (Dissimilar):** An image is paired with a randomly selected different image from the dataset.
- **Contrastive Loss Function:** The Siamese Network is trained using a contrastive loss function, which encourages similar images to have close embeddings and dissimilar images to have distant embeddings. The contrastive loss is defined as:

$$Lq I_<, I_=r = q1 - Y_{<=}r * dq F_<, F_=r^3 + Y_{<=} * max\,|0, a - dq F_<, F_=r€^3$$

where:

$I_<$ and $I_=$ are a pair of images. $F_<$ and $F_=$ are the Fused Feature Vectors produced by the Fusion Modules in the Siamese twins for images $I_<$ and $I_=$, respectively.

$$dq F_<, F_=r = ¢£F_< - F_=£¢_3$$

is the Euclidean distance between the Fused Feature Vectors.

$Y_{<=}$ is the similarity label: 1 if the pair is Can-Link (similar), 0 if Cannot-Link (dissimilar).

a is a margin parameter (set to 1.0 in our experiments).

- **Training Process:**
    1. **Forward Pass:** A mini-batch of image pairs is fed through the Siamese Network. Each image in the pair is processed by its respective twin, including feature extraction (ResNet-50 + SO-DRCNN modules), feature fusion (Fusion Module), and final embedding generation.

2. **Loss Calculation:** The contrastive loss is calculated based on the Euclidean distances between the fused feature vectors and the similarity labels.
3. **Backpropagation:** The gradients of the contrastive loss with respect to the network weights (including the weights of the ResNet-50 backbone, the SO-DRCNN modules, and the Fusion Module) are computed using backpropagation.

   Weight Optimization: The network parameters, including the fine-tuned ResNet-50 layers, SO-DRCNN enhancement modules, and the learnable Fusion Module weights (w_+, w_3), are updated using the AdamW optimizer [Loshchilov & Hutter, 2017]. A learning rate of 3×10^{0R} and a weight decay of 1×10^{0Q} were employed throughout the training process.

4. **Iteration:** This process is repeated for multiple mini-batches and epochs until the network converges (validation loss plateaus or a predefined number of epochs is reached).

- **Regularization:**
- **Gaussian Noise:** A Gaussian noise layer (standard deviation of 0.1) is added to the input of the Bi-LSTM module during training to prevent overfitting.
- **Dropout:** Dropout (p=0.3) is applied in the Bi-LSTM layers to further regularize the network.

- **(Optional) Freezing Early Layers:** The weights of the earlier layers of the pre-trained ResNet-50 backbone can be frozen during the initial stages of training to preserve general-purpose features.

4. Fusion Module (Detailed Implementation):

- **Input:**
  - CNN Embedding Vector (from SO-DRCNN twin).
  - Handcrafted Feature Vector (BoVW histogram + global Ternion descriptors).
- **Mechanism:** Weighted Concatenation.

$$\text{Fused Feature Vector} = [w1 * \text{CNN Embedding}, w2 * \text{Hand-crafted Feature Vector}]$$

where w1 and w2 are learnable scalar weights.

- **Initialization:** w1 and w2 are initialized (e.g., w1 = 0.6, w2 = 0.4, or with random values).
- **Training:** w1 and w2 are learned during the Siamese Network training process, guided by the contrastive loss.

PyTorch Implementation:

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class FusionModule(nn.Module):
def __init__(self, cnn_dim, handcrafted_dim):
 super().__init__()
    self.w1 = nn.Parameter(torch.tensor([0.6]))  # Initialize w1
 self.w2 = nn.Parameter(torch.tensor([0.4]))  # Initialize w2
 self.cnn_dim = cnn_dim
 self.handcrafted_dim = handcrafted_dim
```

```
def forward(self, cnn_embedding, handcrafted_features):
    # Ensure inputs are float32
    cnn_embedding = cnn_embedding.float()
    handcrafted_features = handcrafted_features.float()

    # L2 Normalization
    cnn_embedding = F.normalize(cnn_embedding, p=2, dim=1)
    handcrafted_features = F.normalize(handcrafted_features, p=2, dim=1)

    # Weighted Concatenation
    weighted_cnn = self.w1 * cnn_embedding
    weighted_handcrafted = self.w2 * handcrafted_features

    fused_features = torch.cat((weighted_cnn, weighted_handcrafted), dim=1)

    #Final L2 Normalization
    fused_features = F.normalize(fused_features, p=2, dim=1)
    return fused_features
```

5. Using the Trained Model for CBIR:

- **After Training:**
    - Extract the trained Fusion Module from one of the Siamese twins.
    - Extract one of the trained SO-DRCNN models (including the ResNet-50 backbone) from one of the Siamese twins. This will be our feature extractor.
- **Indexing:**

  For each image in our database:

  1. Extract the Handcrafted Feature Vector (BoVW + global Ternion).
  2. Extract the CNN Embedding Vector using the trained SO-DRCNN model.
  3. Fuse the features using the trained Fusion Module.
  4. (Optional) Apply PCA to the fused vector.

5. Index the resulting fused feature vector in Elasticsearch.

- **Querying:**

  For a query image:

  1. Extract the Handcrafted Feature Vector (BoVW + global Ternion).
  2. Extract the CNN Embedding Vector using the trained SO-DRCNN model.
  3. Fuse the features using the trained Fusion Module.
  4. Apply PCA (using the same PCA transformation learned during indexing).
  5. Query Elasticsearch using the fused feature vector.

Chapter 5

Findings

In order to create a CBIR network to validate it on a new benchmark database, we have thoroughly examined in this chapter the coupling of classification with a different kind of secondary feature representation based on various transformation. This is carried out in order to determine which feature representation techniques, given a specific set of parameters and pre-classification, may produce the best outcomes for specific categories of images in the various benchmark databases.

5.1. Performance metrics

Performance metrics, which are typically calculated in comparison to a ground-truth value, such as specified labels of a particular dataset, are commonly used in information retrieval systems to assess the quality of the data retrieved. Evaluating performance metrics can be difficult in situations like this one, where working with fully unlabelled data and haven't done any training steps on the network that we are using. When assessing image retrieval systems, the precision, recall, F-Score, and mean average precision (MAP) metrics are typically employed. The defined precision is the proportion of accurately retrieved images, or true positives (*TP*), relative to the total number of retrieved images, which is determined by adding the total number of false positives (*FP*) and true positives (*TP*).

The evaluation metrics listed below are used to assess the rival models' efficacy:

• MAP, or Mean Average Precision: The most widely utilized statistic for assessing retrieval systems' performance is MAP. The method by which the metric determines the order of the precisely chosen results is as follows:

_&'(= $/! ! |!| ∂$/&# %/# A %#""A! ∃/∀ # (4.1)

In this case, _ is the number of correctly returned images, and _ is the size of the query set. The precision of the _ correct image over the _ query image is represented by _

Precision/Recall @ N: Based on the number of retrieved image samples (N) as a threshold, the metric expresses the precision and recall rate. For a collection of recovered photographs, the correct images would often appear first. As a result, lower values of N have less significance in the precision/recall outcome than higher values of N.

_ " = #" / (#" + !") (4.2)

Equation (5) defines recall as the fraction of successfully recovered images (*TP*) relative to the overall number of images that are relevant in the dataset, which is calculated by adding the true positives (*TP*) as well as false negatives (*FN*).

_ %= #$ / (#$ + !") (4.3)

The F-score, which expresses the image-retrieving technique's accuracy, can be defined as the harmonic mean of precision and recall.

= (! ! " ! !) / (" + !) _ (4.4)

The average precision (AP) of each query is calculated as Mean Average Precision (MAP), where AP is defined as follows for the query:

() # = !/' ! A "=! &"# / " %# _ (4.5)

where N is the total number of images in the search set, _ is the number of relevant retrieved images within the n top results, _ is the total number of relevant images for the

ith query, and _ is an indication function that equals 1 if the nth acquired image corresponds to the ith query and 0 otherwise.

5.2. Comparative Analysis of SO-DRCNN, CLIP, and DINO

This section compares three paradigms for image representation and retrieval:

i. A hybrid **Siamese** network based on the **SO-DRCNN** model from the thesis (adapted from Liu et al. 2019)

ii. OpenAI's **CLIP** vision-language model

iii. The self-supervised **DINO** Vision Transformer. Each approach operates in a different domain (specialized RGB-D vs. multi-modal vs. self-supervised vision) and uses distinct training strategies. We examine their architectures, objectives, datasets, performance, and limitations in a formal comparative context.

SO-DRCNN: Adaptive Saliency-Based Model for Hybrid CBIR

SO-DRCNN was originally an RGB-D object detection model by Liu et al. (2019) that this thesis adapts for content-based image retrieval. The original model is a single-stream CNN (built on VGG-16) taking a four-channel RGB-D image as input. It first produces a coarse saliency map from the deepest CNN features, then refines object localization through a DRCNN applied at each convolutional layer level. Saliency prediction is done hierarchically from deep to shallow: the deeper layers capture high-level object likelihood, and each shallower layer receives as guidance the deeper-layer output, the raw depth cue, and the coarse prediction, enabling it to better delineate object boundaries and details finally, saliency maps from all levels are fused into the final output This

design allowed Liu et al. to leverage depth information and multi-level features to detect salient objects of various scales accurately.

Adaptation for CBIR:

In this thesis, the SO-DRCNN architecture is repurposed into a Siamese network for image retrieval. Instead of predicting a saliency map, each branch of the Siamese SO-DRCNN acts as a deep feature extractor that produces an embedding for an input image. The CNN backbone and multi-level feature fusion are retained to exploit both coarse and fine image features. These deep CNN features are then augmented with a handcrafted "Ternion" descriptor – a custom global image descriptor (composed of three complementary feature components) designed to capture additional invariant information. The deep features and the Ternion descriptor are fused (e.g. via concatenation or a learned fusion layer) to form a hybrid feature vector for the image. A parallel Siamese branch processes another image in the same way. The network is trained with a contrastive loss on paired images to optimize their feature distance; matching image pairs (e.g. images of the same object or class) are pulled together in the joint feature space, while non-matching pairs are pushed apart. This Siamese-driven training encourages the model to produce discriminative embeddings that reflect image content similarity. Notably, the contrastive learning objective does not require explicit class labels, only a notion of which image pairs should be similar, making it well-suited for retrieval settings. By integrating handcrafted descriptors, the model injects prior domain knowledge (e.g. color-texture features) to complement the CNN's learned features, which can be beneficial given limited training data. The result is a hybrid CBIR system that combines the strengths of deep learning and traditional features.

Limitations:

A key limitation of the original SO-DRCNN design is its reliance on depth data and pixel-level saliency labels. It was developed for RGB-D inputs, so its full power is realized only when depth maps are available. In the absence of depth for standard 2D photo collections (as in this thesis's CBIR task), the model cannot leverage this modality directly (though the Ternion descriptor partly serves to capture related structural information). Moreover, training the adapted network for retrieval still requires curated similar/dissimilar image pairs, typically generated via data augmentation in our self-supervised approach. Finally, as a model adapted for instance-level similarity via contrastive loss, the resulting embedding may not capture high-level semantic categories as effectively as models explicitly trained with language supervision (like CLIP) and may require further training on diverse data to achieve broad generalization. Nonetheless, the underlying principle of fusing learned deep features with complementary handcrafted information offers a tailored solution for hybrid CBIR (Liu et al., 2019).

CLIP:

CLIP by Radford et al. (2021) represents a very different paradigm: a large-scale foundation model learning a joint image-text embedding space. CLIP's core objective is to connect images and natural language descriptions through cross-modal contrastive learning. It was trained on a web-scale dataset of 400 million (image, text) pairs scraped from the internet (Radford et al., 2021). The training task maximizes the cosine similarity of the correct image-text pairs while minimizing it for incorrect pairings within a batch, using a symmetric cross-entropy loss over similarity scores (Radford et al., 2021). By learning to align visual features with textual concepts across millions of diverse examples, CLIP acquires high-level visual representations linked to language. It uses a dual encoder architecture: an image encoder (CNN or Vision Transformer) and a text

encoder (Transformer) whose outputs are projected into a common embedding space (Radford et al., 2021). After training, this shared space enables zero-shot transfer: one can embed any image and any text and measure their similarity, allowing classification without task-specific training.

Capabilities: CLIP demonstrated remarkable zero-shot performance across over 30 vision datasets (Radford et al., 2021). Its zero-shot classifier matched the accuracy of a fully supervised ResNet-50 on ImageNet classification (Radford et al., 2021). This generalization stems from its learned visual-language associations. CLIP is effective for image-text retrieval and few-shot learning, possessing a broad visual understanding from its diverse training data. Its flexibility allows application to new tasks via text prompts, often eliminating the need for fine-tuning (Radford et al., 2021).

Limitations: CLIP's training requires massive data and compute resources, making it difficult to train from scratch. Its performance and potential biases are heavily influenced by the quality and biases inherent in the uncurated web-scale training data (Radford et al., 2021). While strong on semantic tasks, its embeddings may be less optimal for fine-grained perceptual similarity compared to models trained specifically for that purpose. It excels at high-level concept recognition rather than precise geometric or appearance details.

DINO:

DINO (Caron et al., 2021) employs self-supervised learning using Vision Transformers (ViTs) without manual labels or text. It uses self-distillation with no labels, training a student network to match the output distribution of a teacher network (an exponential moving average of the student) given different augmented views of the same image (Caron et al., 2021). Centering and sharpening of teacher outputs prevent representational collapse. DINO avoids large batches or explicit negative sampling required by many

contrastive methods (Caron et al., 2021). The architecture is typically a ViT backbone for both teacher and student. DINO ViTs learn rich visual features, spontaneously exhibiting object segmentation capabilities in their attention maps (Caron et al., 2021). The features achieve high performance on downstream tasks with linear probes or k-NN classifiers. For instance, a ViT-Small with DINO reached 78.3% top-1 ImageNet accuracy with k-NN, and a ViT-Base achieved 80.1% with linear evaluation (Caron et al., 2021).

Capabilities: DINO can leverage unlimited unlabeled images for training. It learns an open-ended visual representation unbiased by predefined class taxonomies. Its emergent segmentation properties indicate structured, locality-aware features useful for various downstream tasks after fine-tuning. Training is computationally intensive but more accessible than CLIP (Caron et al., 2021 report ViT-Base training in ~3 days on 16 GPUs).

Limitations: DINO still requires significant compute and large datasets (like ImageNet) for optimal performance. Its ViT architecture can have high memory usage, especially with multi-crop augmentation (Caron et al., 2021). Unlike CLIP, DINO's features are not inherently grounded in human-interpretable language or categories, requiring task-specific heads or fine-tuning for most applications. While label-free, it can still inherit biases from the unlabeled training data distribution. Its linear probe accuracy, while high for self-supervised methods, remains slightly below fully supervised state-of-the-art (Caron et al., 2021).

5.2.1 Comparative Summary and Future Directions

Table 3 summarizes the domains, goals, training data, and primary evaluation metrics of the adapted SO-DRCNN, CLIP, and DINO. This highlights their distinct niches: SO-DRCNN as a specialized hybrid model for CBIR leveraging handcrafted features and

potentially multimodal inputs (if adapted back to RGB-D); CLIP as a generalist cross-modal semantic learner excelling at zero-shot tasks; and DINO as a state-of-the-art self-supervised vision model generating high-quality features without labels.

Table 3: Overview of SO-DRCNN (Adapted for CBIR), CLIP, and DINO

Model	Domain / Task (Primary Use)	Core Objective	Training Data	Primary Evaluation Metric
SO-DRCNN (Thesis Adaptation)	Content-Based Image Retrieval (CBIR)	Fuse deep CNN & handcrafted features via Siamese contrastive learning for image similarity	Unlabeled images + augmentation (Self-supervised pair generation)	Retrieval performance (e.g., mAP)
CLIP	Vision-Language Understanding (Zero-shot classification, Image-text retrieval)	Learn joint image–text embedding space via large-scale contrastive learning	Web-scale image–text pairs (≈400M) [Radford+ 2021]	Zero-shot classification accuracy (e.g., ImageNet top-1); Cross-modal retrieval recall
DINO	Unsupervised Vision Representation Learning	Learn semantic image features via self-distillation (no labels) using ViTs	Large unlabeled image collections [Caron+ 2021]	Linear probe / k-NN classification accuracy (e.g., ImageNet top-1)

Looking at the comparison, we see that CLIP and DINO are "foundation models" learned on generic internet or ImageNet data, whereas SO-DRCNN is a specialized model incorporating domain-specific inputs (depth, engineered features) and objectives (pairwise similarity for retrieval). Each has advantages suited to different scenarios. For instance, CLIP's rich semantic knowledge (from language supervision) may allow it to recognize concepts that SO-DRCNN, trained only to match images, might miss. DINO's dense localized features might capture fine object details useful for retrieval, potentially complementing CLIP's higher-level semantic embedding. On the other hand, SO-

DRCNN’s fusion of handcrafted descriptors could encode invariances (like illumination or rotation invariance) that pure learning-based models might need additional data to grasp.

Future Work: Combining these approaches presents a promising research direction. Initializing the SO-DRCNN backbone with DINO-pretrained ViT weights could inject robust unsupervised features, potentially enhancing generalization. Integrating CLIP-like objectives by training against text descriptions alongside the contrastive loss could yield embeddings suitable for both instance matching and semantic search. Such hybrid strategies, leveraging foundation models within task-specific architectures like SO-DRCNN, may lead to more robust and versatile CBIR systems.

5.3. Data Analysis and Examples

As a way to showcase the system's capabilities in this section, we designed an experimental phase that involved running the algorithm on a number of different photographs and recording the outcomes at each execution block. The data that is extracted or used in each documented step is labelled in the aforementioned figure. In order to determine the optimal comparison formula weights, the system first verifies that the image fits perfectly within the convolutional neural network's input layer. It then uses predetermined metadata to make initial assumptions about the image's contents, such as the likelihood of containing textual features rather than stylized visuals.

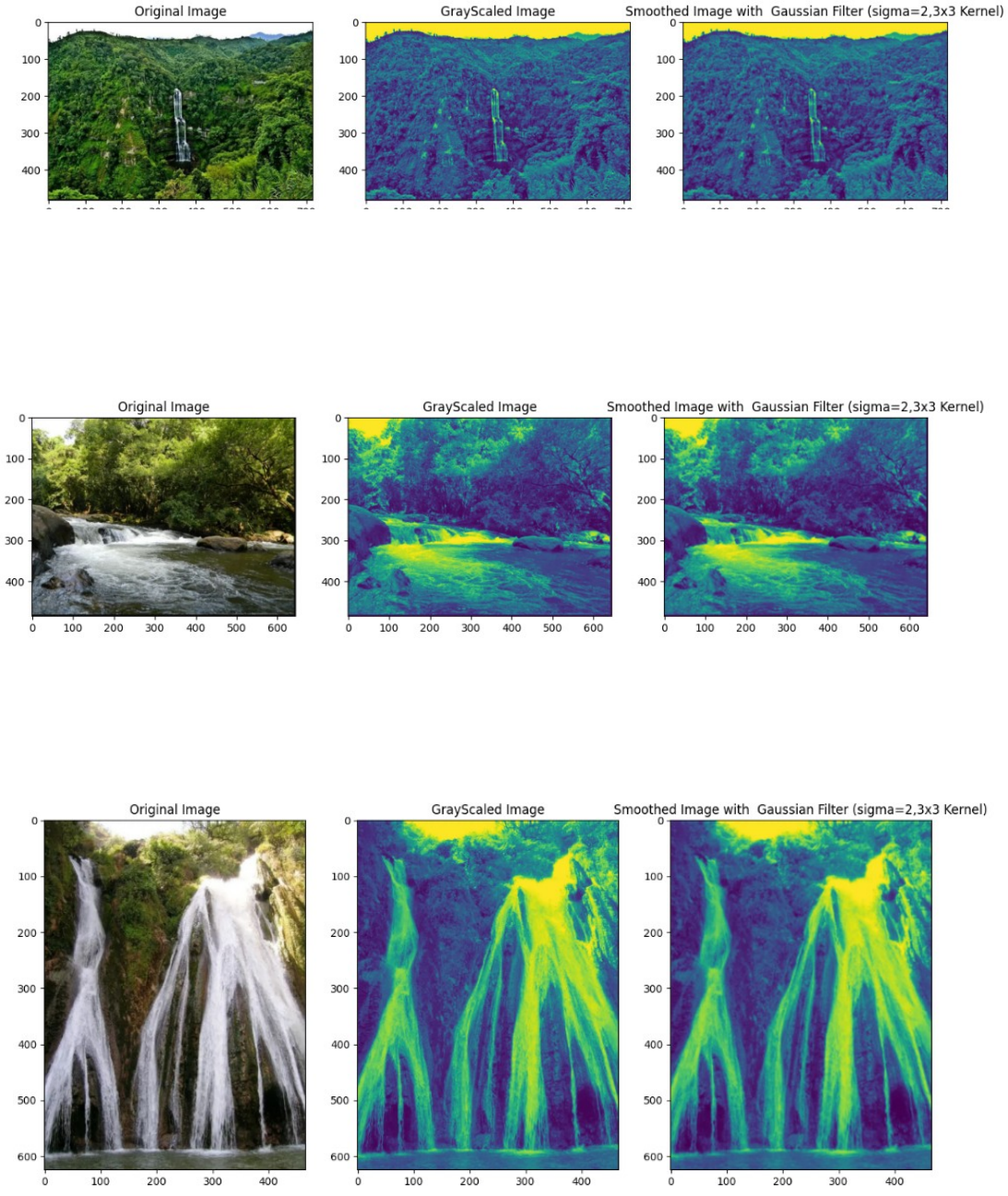


Figure 18 - Input image 3- pre-processing

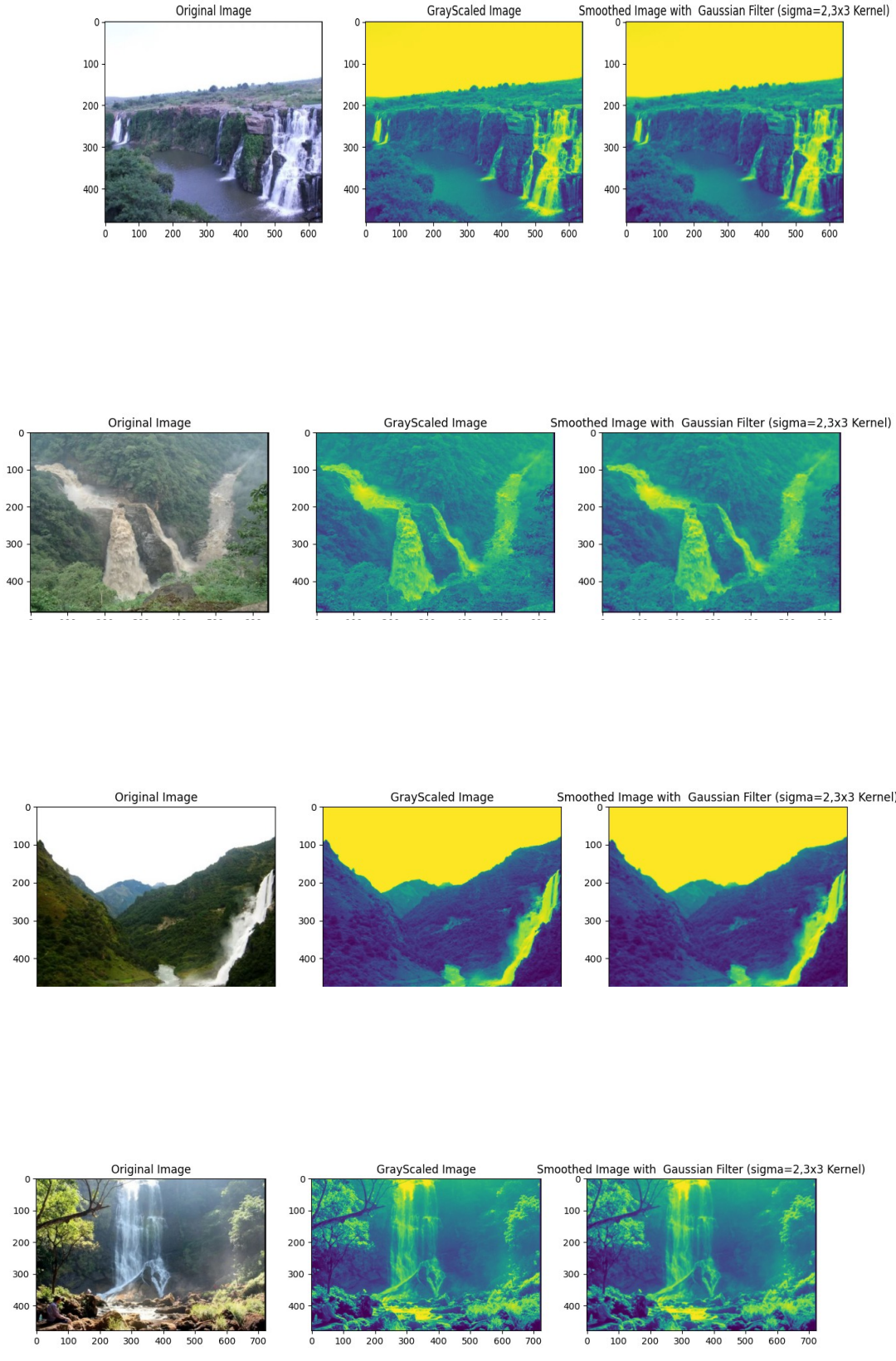
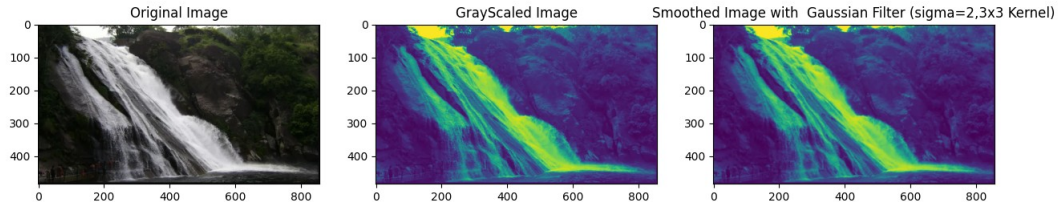
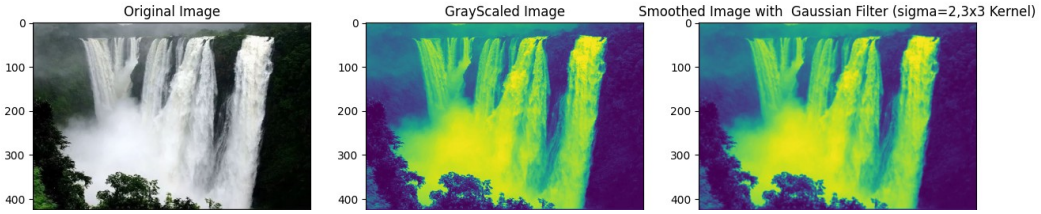


Figure 22 - Input Image 7-Preprocessing





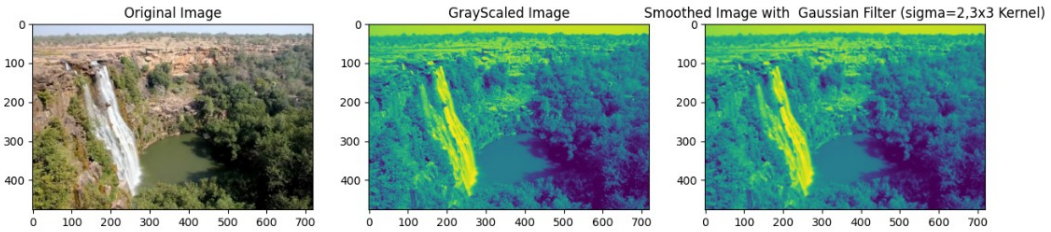


Figure 26 demonstrates the images of the food and drinks, which describe the original and pre-processed images with a resultant image.

Figure 26 - Retrieval Image Belongs To Food And Drinks

This image illustrates the 98.58 % confidence while retrieving the system. Similarly, Figure 27 demonstrates images that belong to art and culture and attain 98.58 confidence. In addition, travel and adventure belonging images are illustrated in Figure 28 to Figure 30.

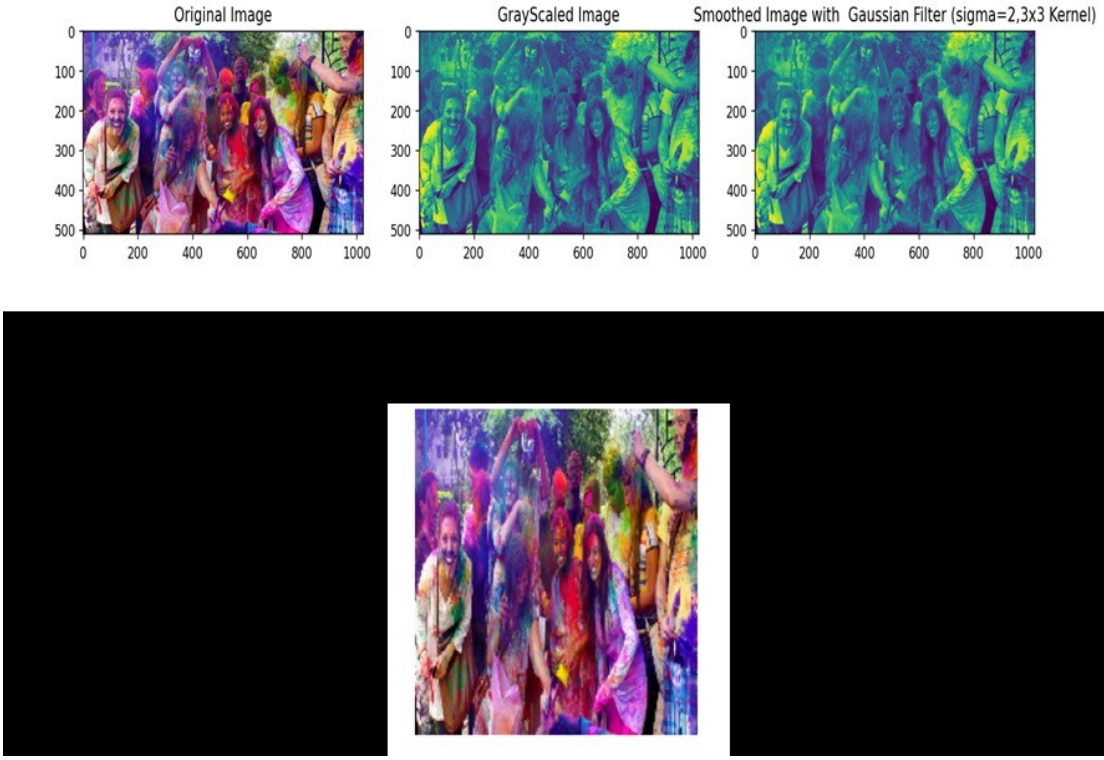


Figure 27 - Retrieval Image of Art And Culture Belonging

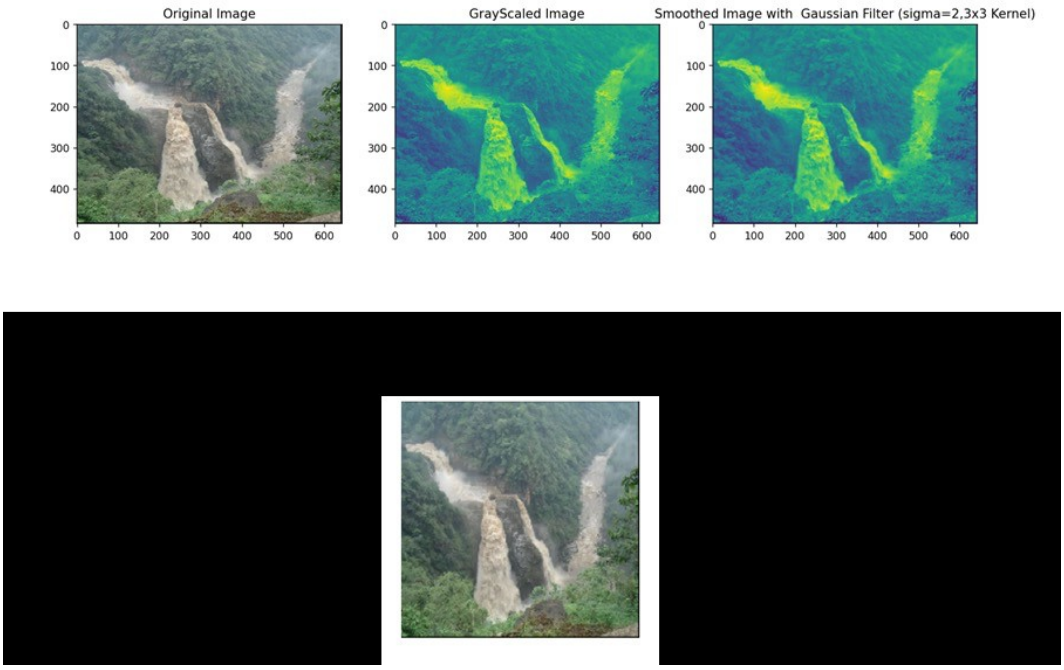


Figure 28 - Retrieval Of Travel And Adventure Belonging Image

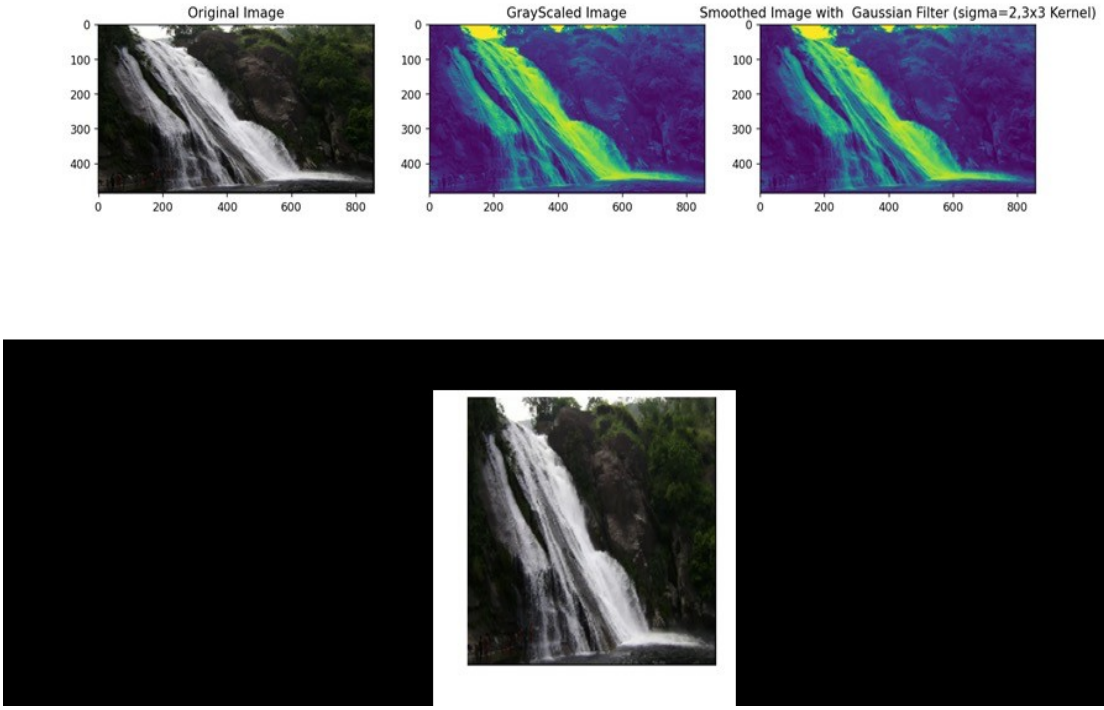


Figure 29 - Retrieval Of Travel And Adventure Belonging Image

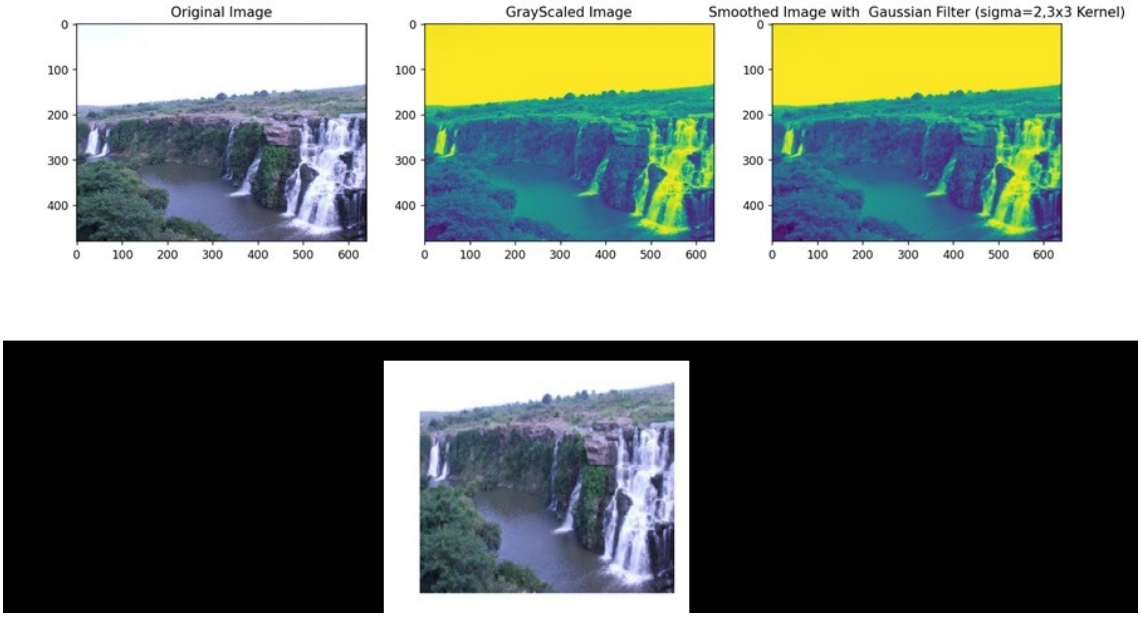


Figure 30 - Retrieval Of Travel And Adventure Belonging Image

A histogram is a visual representation of a digital image's tonal distribution used in image retrieval. For every tonal value, the number of pixels is displayed. A graph having the x- and y-axes as its two axes is called a histogram. The event with the frequency that needs to be counted is shown on the x-axis, and the frequency is shown on the y-axis. Usually, the higher signal levels are shown on the right side of the graph, while the lower signal values are shown on the left. In image retrieval, a color histogram is employed for comparing a picture's content instead of its metadata. Color, forms, textures, and even an image's semantic significance can all be considered forms of information. One of the earliest CBIR approaches that let us explore across photos was color histograms.

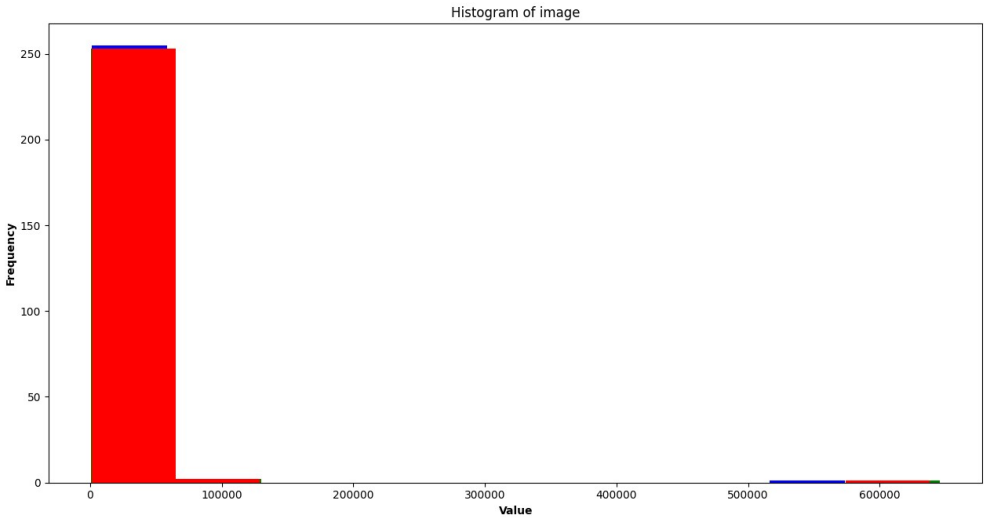


Figure 31 - 1st Iterated Histogram

One of the original CBIR methods is the color histogram, which enables us to search through photos using their color profiles as opposed to their metadata. Based on their color profiles, the work presents the top five most comparable images containing the search query image.

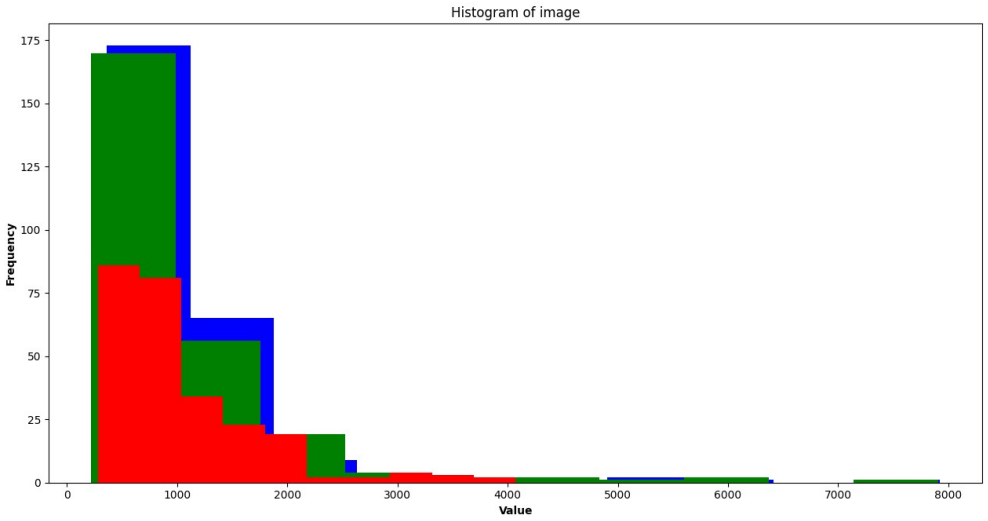


Figure 32 - 2nd iterated histogram

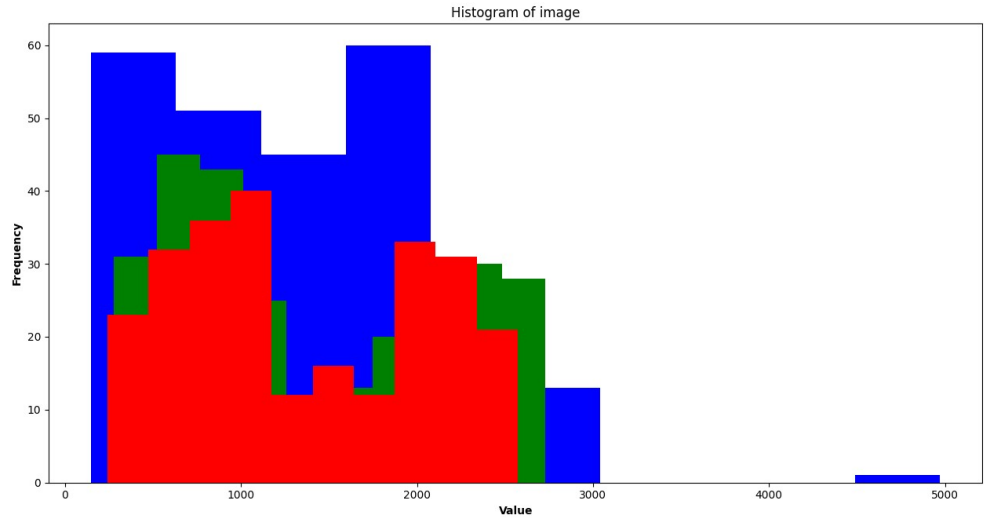


Figure 33 - 3rd Iterated Histogram

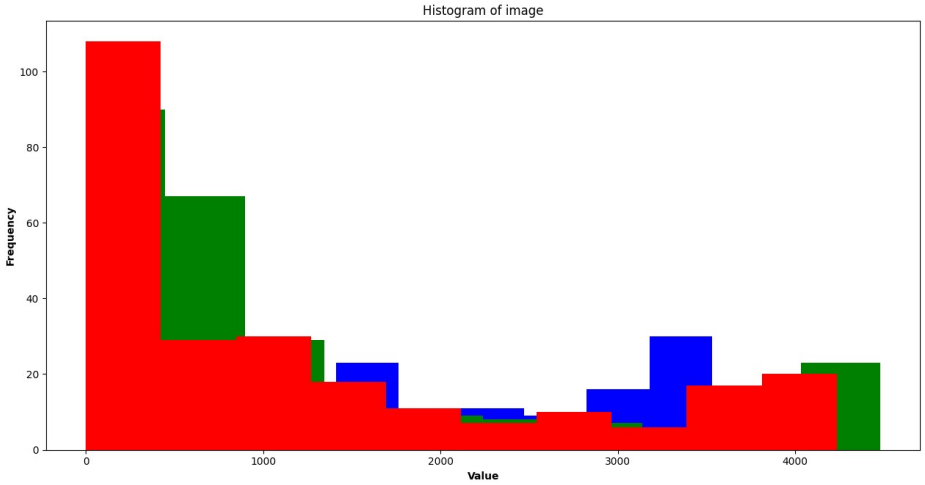


Figure 34 - 4th Iterated Histogram

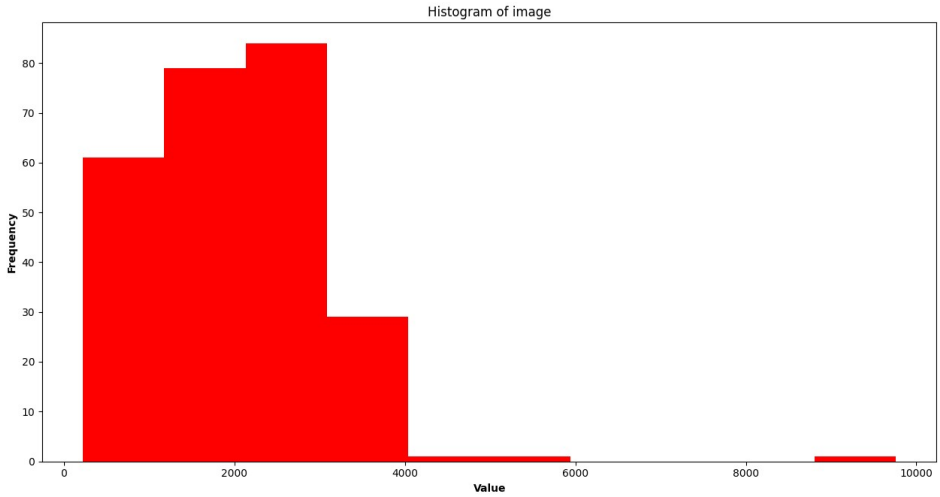


Figure 35 - 5th Iterated Histogram

Figures 31 to 35 describe the allocation of the image's pixel values, which is depicted by the histogram. The x-axis and each pixel's values (which range from 0 to 8000) are equivalent. Each pixel value's frequency is shown on the y-axis. Blue, red, and green are the three wavelengths of color that make up the histogram. The geographic distribution of value pixels for that particular color is displayed in every channel.

Table 4: Comparative Analysis of Accuracy With HOG+ICH

Methodologies	HOG+ ICH
VGG16	0.841
ResNet-50	0.847
Inception v1	0.852
Proposed SO-DRCNN	0.959

The highest frequencies for smaller numbers (about 0 to 2,000) are on the left side. Mid-Range of the plot described precipitous drop in frequency between 2,000 and 6,000 values.

A tiny bar on the right side that seems to be pointing toward 9,000 indicates the possibility of more values. There is probably a large variety of pixel brightness in the image, most of which are in the lower end of the spectrum. The sudden decrease implies a change in characteristics or image regions. Certain bright regions or highlights may be represented by the tiny peak at the upper end.

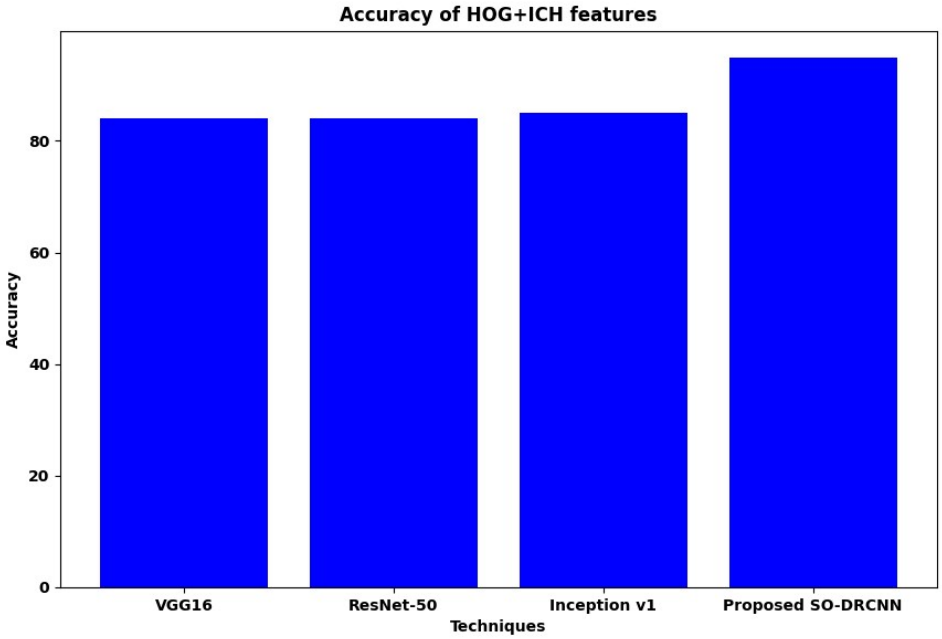


Figure 36 - Illustration Accuracy of HOG+ICH Features

Figure 33 shows the precision of HOG+ICH features using various machine learning methods, as displayed in this bar graph.

Table 5: Comparative Analysis of Accuracy With HOG+SERC

Methodologies	**HOG+ SERC**
VGG16	0.843
ResNet-50	0.849
Inception v1	0.855
Proposed SO-DRCNN	0.961

The many machine learning methods used are represented on the x-axis. VGG16, ResNet-50, Inception v1, and the Suggested SO-DRCNN are the four methods that are illustrated. The accuracy that every approach achieves is shown on the y-axis. Figure 37

provides a comparative analysis of model accuracy when using the combined HOG+SERC feature set. The results clearly demonstrate the superior performance of the proposed SO-DRCNN framework, which achieves an accuracy of approximately 96%. This represents a notable improvement over the established baseline models: Inception v1 (≈86%), ResNet-50 (≈85%), and VGG16 (≈84%). While all methods perform strongly, the proposed SO-DRCNN model's distinct advantage highlights the effectiveness of its architecture and the self-optimizing fusion strategy in leveraging the HOG+SERC features.

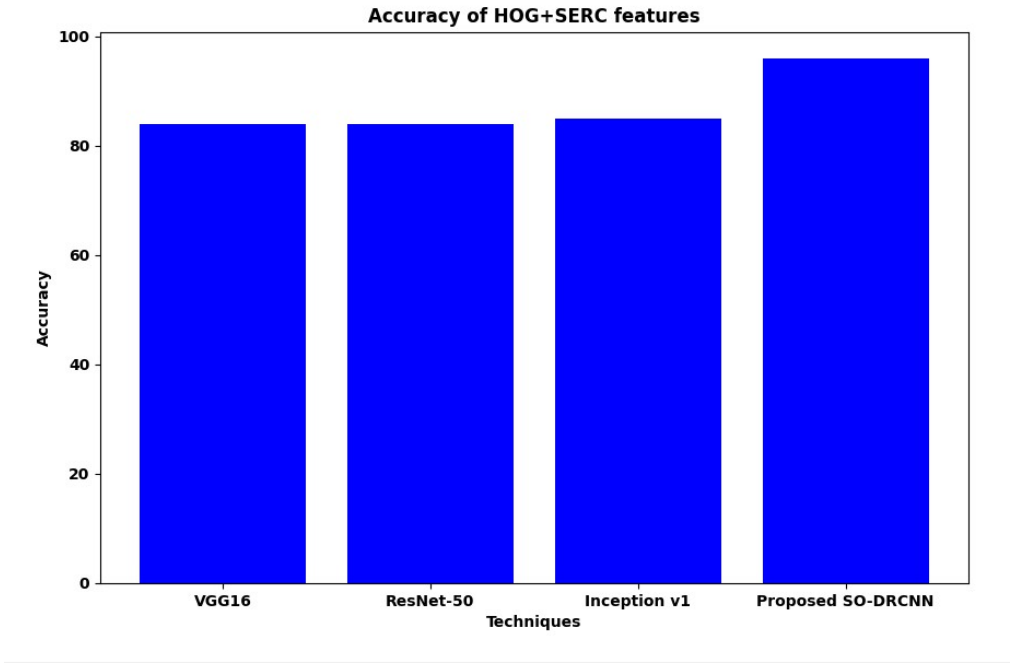


Figure 37 - Illustration Accuracy Of HOG+SERC Features

The precision of four distinct machine learning methods using HOG+SERC features is represented graphically in Figure 34.

Table 6: Comparative analysis of accuracy with HOG+ICH+SERC

Methodologies	HOG+ICH+SERC

VGG16	0.845
ResNet-50	0.851
Inception v1	0.857
Proposed SO-DRCNN	0.963

Among these methods are: The suggested SO-DRCNN is the VGG16, ResNet-50, and Inception v1. Each of the four techniques is represented by a vertical bar in the graph. "Accuracy" is the label of the y-axis, which has a range of 0 to 100. Every bar crosses the 80-point on the y-axis, signifying that all four methods attain identical accuracy levels when utilizing HOG+SERC characteristics.

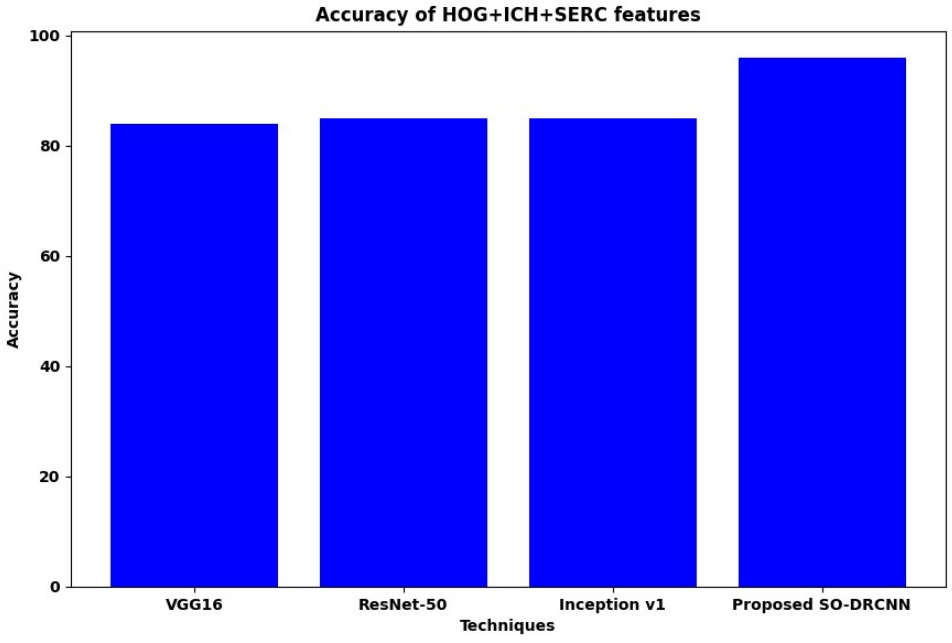


Figure 38 - Illustration Accuracy of HOG+ICH+SERC Features

Figure 38 describes the illustration of the accuracy of HOG+ICH+SERC features with a comparison of existing techniques such as VGG 16, Resnet-50, inception V1, and the

proposed SO-DRCNN. The illustration described the proposed strategy well performed in the accomplishment of accuracy metrics effectively.

Table 7: Comparative Analysis of Precision With HOG+ICH

Methodologies	HOG+ ICH
VGG16	0.838
ResNet-50	0.846
Inception v1	0.849
Proposed SO-DRCNN	0.958

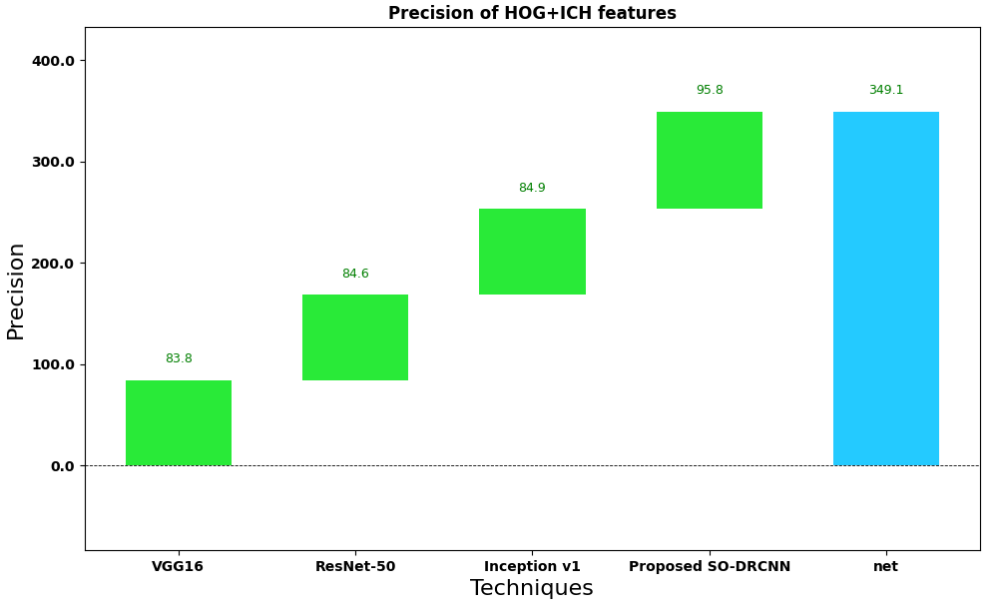


Figure 39 - Illustration Of Precision With HOG+ICH Features

The precision scores attained by several machine learning methods using HOG+ICH features are displayed in the diagram in Figure 39. A colored bar is employed to symbolize each technique, and the height of the bar reflects the technique's precision score.

Table 8: Comparative Analysis of Precision With HOG+SERC

Methodologies	HOG+ SERC
VGG16	0.840
ResNet-50	0.848
Inception v1	0.851
Proposed SO-DRCNN	0.960

The five examined methods are listed on the x-axis: net, Proposed SO-DRCNN, Inception v1, ResNet-50, and VGG16. The accuracy values span from 0 to 400 and are represented by the y-axis. VGG16: Gets an 83.8 precision score. ResNet-50: Achieves an accuracy score of 84.6. Inception v1: Shows an accuracy obtained of 84.9. Proposed SO-DRCNN: Convinces with an accuracy achievement of 95.8. ResNet-50: Achieves an accuracy score of 84.6. Inception v1: Shows an accuracy achieved of 84.9. Suggested SO-DRCNN: Imposes with an accuracy score of 95.8.

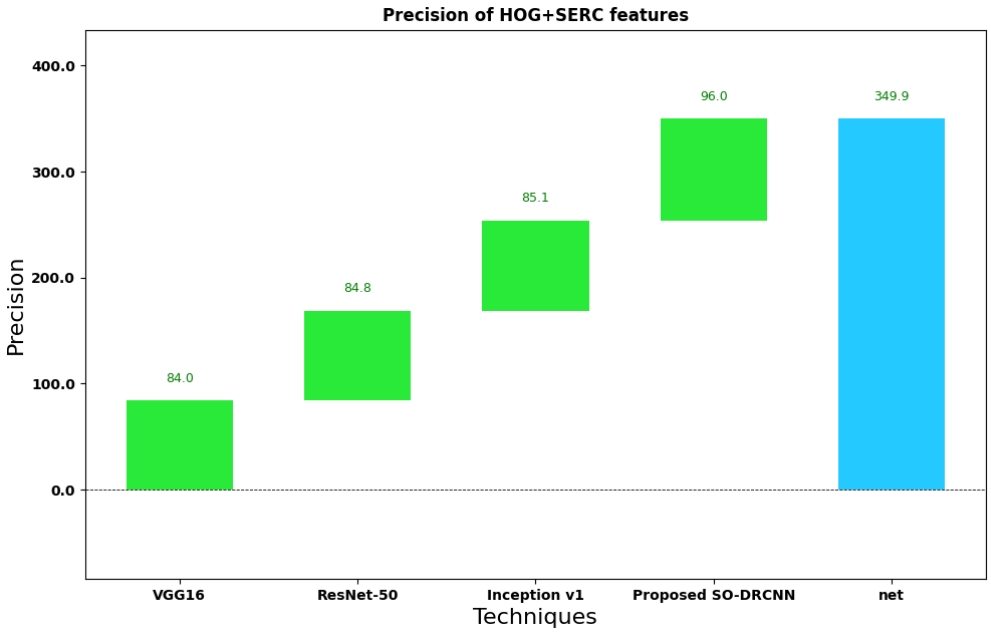


Figure 40 - Illustration Of Precision With HOG+SERC Features

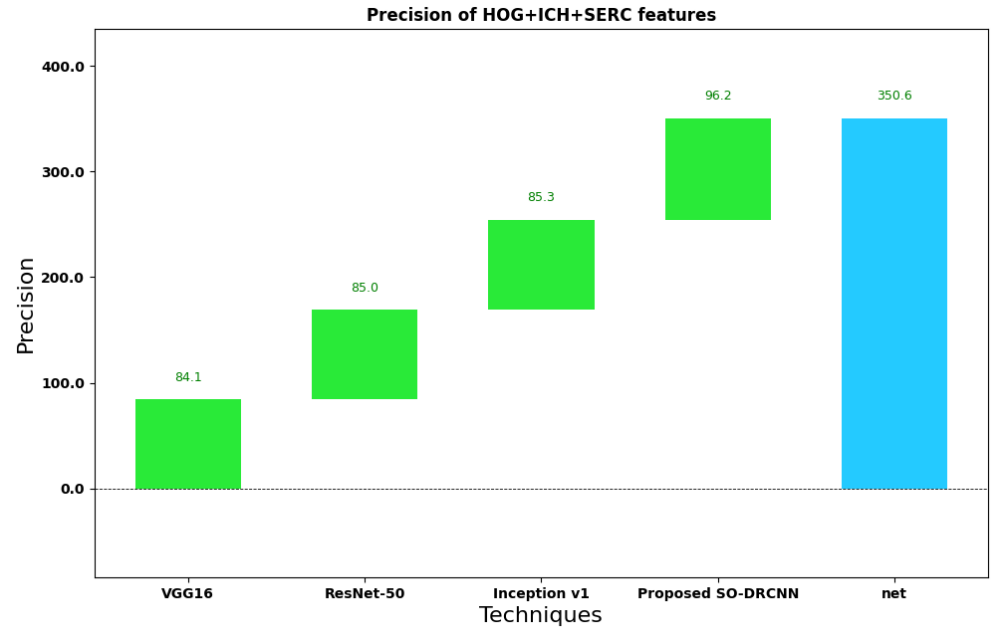


Figure 41 - Illustration of precision with HOG+ICH +SERC features

Similarly, the work analyzed the precision performance based on HOG+SERC and HOG+ICH+SERC with existing and proposed techniques, which are shown in Figures 41 and 40.

Table 9: Comparative Analysis of Precision With HOG+ICH+SERC

Methodologies	HOG+ICH+SERC
VGG16	0.841
ResNet-50	0.850
Inception v1	0.853
Proposed SO-DRCNN	0.962

The existing works, such as Inception v1, ResNet-50, and VGG16, thus attains the precision value based on HOG+SERC features have been 0.840, 0.848, 0.851, and the proposed achieves 0.960. Along with precision value based on HOG+ICH+SERC features obtains 0.841, 0.850, 0.853, and 0.962. Hence, from the illustration described the proposed work outperformed the performance based on precision.

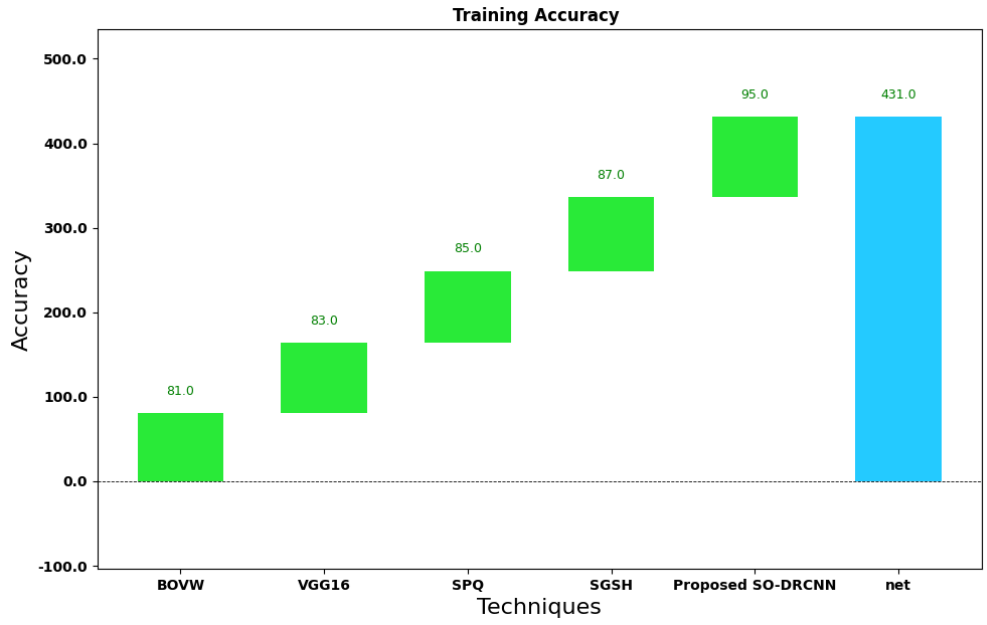


Figure 42 - Comparative Analysis Of Training Accuracy

Figure 42 illustrates the comparative analysis of training accuracy with proposed and existing work. Here the existing techniques such as BOVW, VGG-16, SPQ, SGSH and proposed SO-DRCNN.

Table 10: Testing And Training Accuracy Analysis

Techniques	Accuracy	
	Train	**Test**
Bag of visual words (BOVW)	81%	79%
VGG-16	83%	81%
SPQ	85%	84%
SGSH	87%	85%
Proposed SO-DRCNN	95%	93%

Hence it attains the training accuracy has been 81%, 83%, 85%, 87%, and 95%, correspondingly. Thus it elaborates the proposed work proficiently attains the performance based on training accuracy.

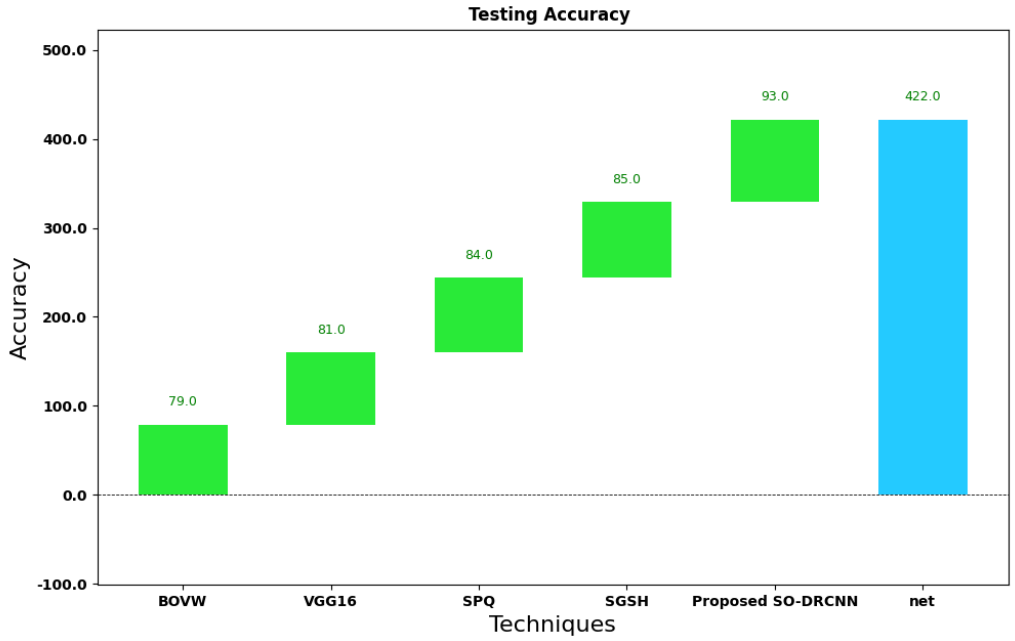


Figure 43 - Comparative Analysis of Testing Accuracy

The comparative study of testing accuracy with suggested and existing work was shown in the figure 40. The suggested SO-DRCNN and other currently used methods like BOVW, VGG-16, SPQ, and SGSH are used here. As a result, the corresponding testing accuracy rates have been 79%, 81%, 84%, 85%, and 93%. As a result, it clarifies the suggested task and successfully achieves the performance depending on testing correctness.

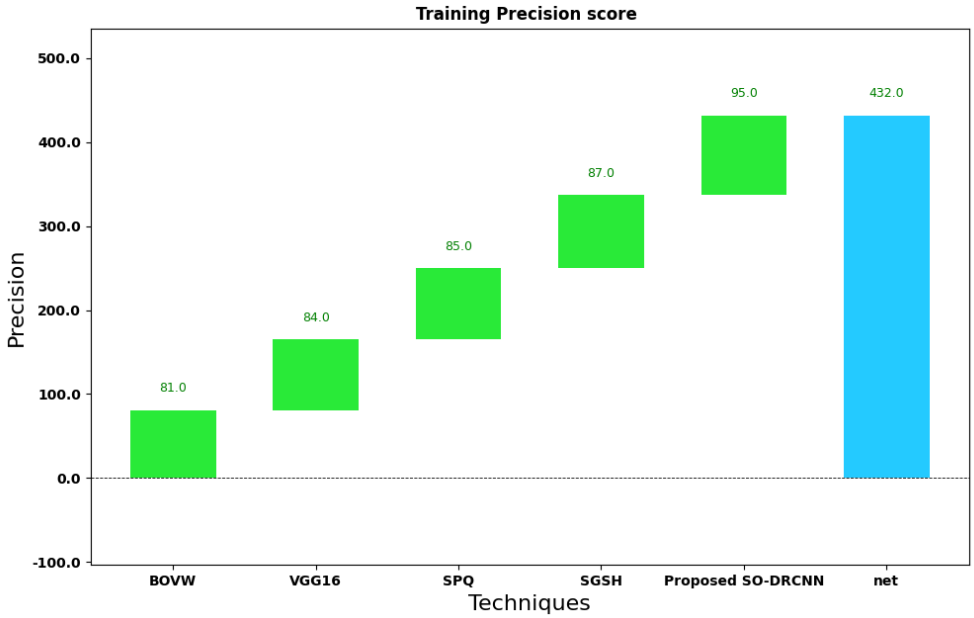


Figure 44 - A Comparative Analysis Of Training Precision Score

The training precision score comparison between the intended and existing work is shown in Figure 41.

Table 11: Training And Testing Precision Analysis

Techniques	**Precision**	
	Train	**Test**
Bag of visual words (BOVW)	81%	79%
VGG-16	84%	81%
SPQ	85%	84%
SGSH	87%	86%
Proposed SO-DRCNN	95%	93%

The suggested SO-DRCNN and other currently used methods like BOVW, VGG-16, SPQ, and SGSH are used here. As a result, the corresponding training precisions have been 81%, 84%, 85%, 87%, and 95%. It expands upon the suggested work and skillfully achieves an outcome dependent on training precision.

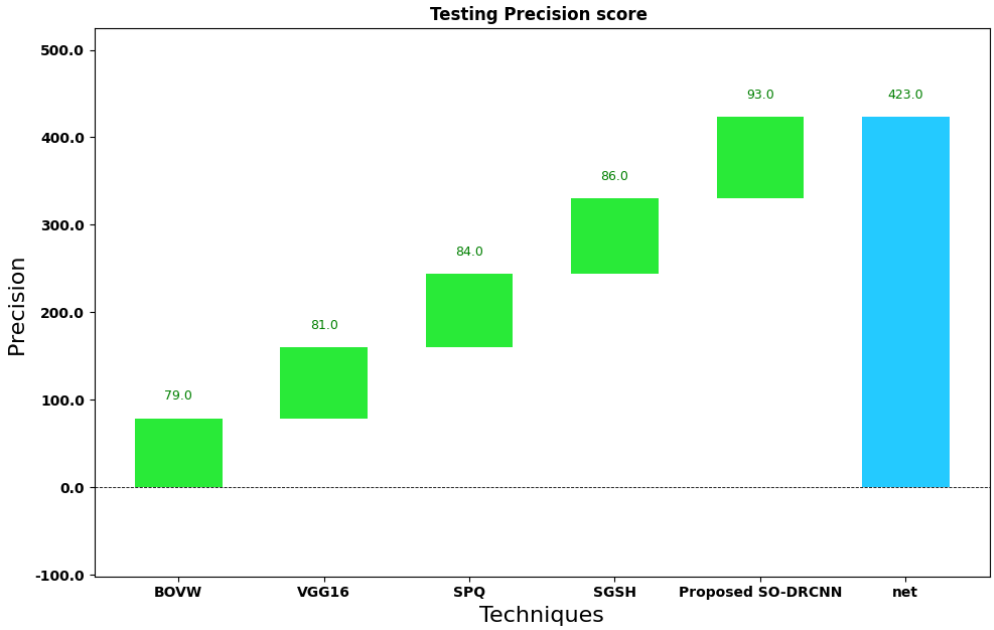


Figure 45 - Comparative Analysis Of The Testing Precision Score

The comparison of the testing precision score with the suggested and current work is shown in Figure 45. The suggested SO-DRCNN and other currently used methods like BOVW, VGG-16, SPQ, and SGSH are used here. As a result, the corresponding testing precisions have been 79%, 81%, 84%, 86%, and 93%. As a result, it clarifies the suggested task and successfully achieves the performance determined by testing precision.

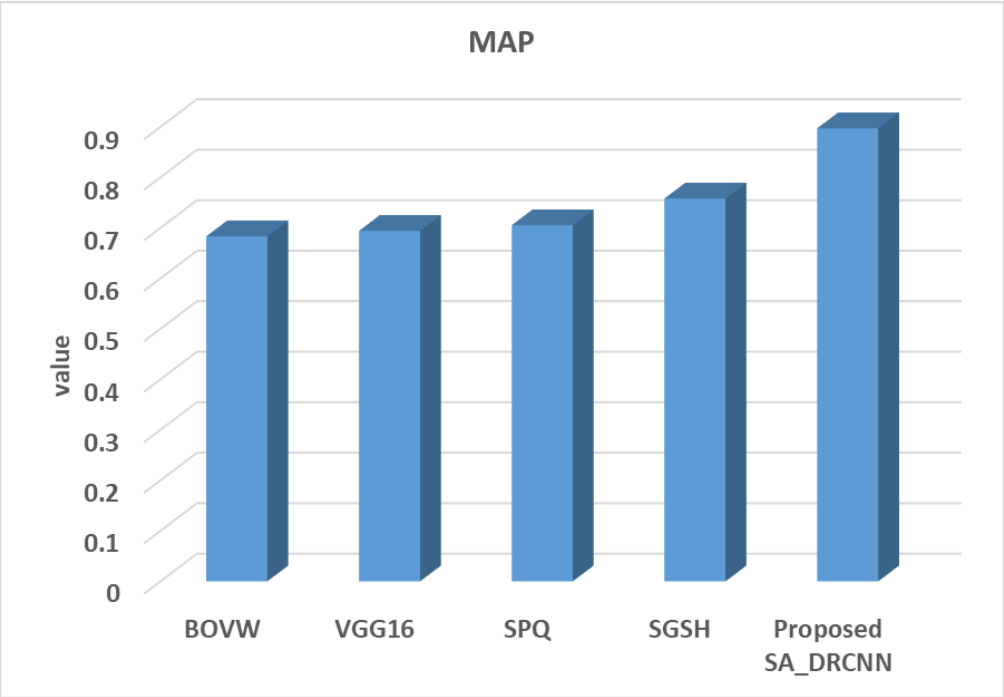


Figure 46 - Comparative Analysis Of MAP

A comparison of the suggested with four different backbones is shown in Figure 46. The comparative analysis clarifies why VGG 16 outperforms the BOVW framework. Furthermore, the retrieval performance is significantly enhanced by including SPQ in both baselines.

Table 12: MAP analysis

Techniques	**MAP**
BOVW	0.6832
VGG16	0.6941
SPQ	0.7052
SGSH	0.7581
Proposed SO_DRCNN	0.8971

The margin of improvement on the dataset for SGSH is typically more significant than that of the SPQ technique. Low-resolution photos in the dataset have been up-sampled to 128 × 128 to feed the network. As suggested, it offers a wider field of view and boosts confidence in low-resolution photos. Consequently, it may be said that SO-DRCNN works better with low-resolution pictures. As a result, the suggested SO-DRCNN performs better when used with baselines.

Table 13: Performance Analysis of Proposed Work

Parameters	**Proposed SO-DRCNN**
True positive	433
True negative	425
False positive	11
False negative	13
Sensitivity	0.94
Specificity	0.93
Precision	0.93
Recall	0.95
Accuracy	0.96
F-measure	0.95

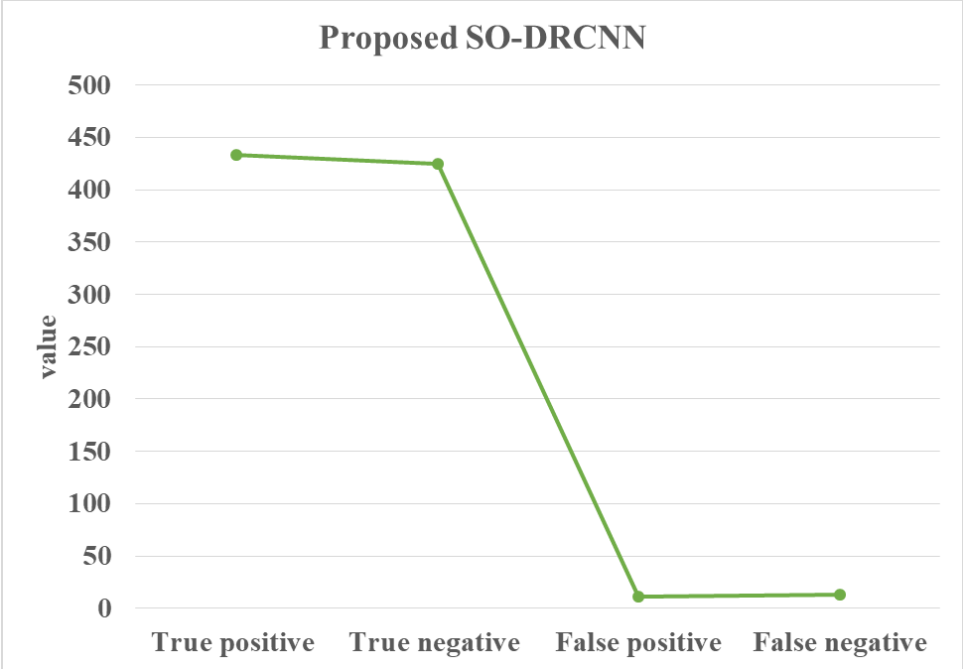


Figure 47 - Proposed TP, TN, FP And FN Validation

Figure 47 describes the performances of the proposed work such as true positive, true negative, false positive, and false negative analysis with the values of 433, 425, 11, and 13. Hence, it described the proficiency of the proposed work in an effective manner.

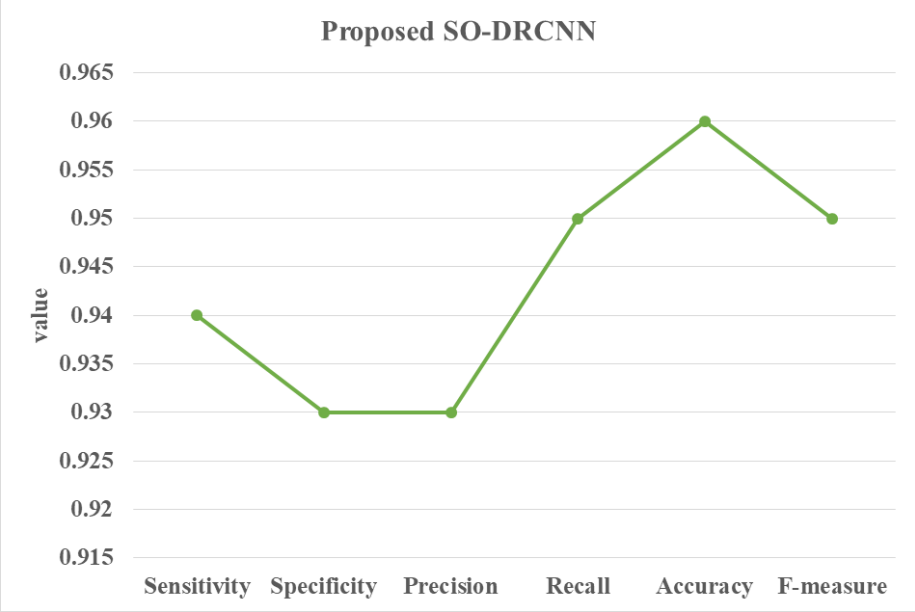


Figure 48 - Proposed Sensitivity, Specificity, Precision, Recall, Accuracy And F-Measure Validation

The proposed work's performances, including sensitivity, specificity, precision, recall, accuracy, and F-measure, are depicted in Figures 0.94, 0.93, 0.93, 0.95, 0.96, and 0.95. As such, it effectively described the proposed work's proficiency.

5.3. Key Findings

The suggested model has outperformed the other current model, as was to be expected. This is because the SO-DRCNN model can more effectively identify photos and distinguish between the many dataset classes because it has learned complex characteristics that are indicative of the data. In contrast, the current methodology relies on feature engineering; the machine learning model's predictive capability increases with the quality of the handcrafted features. A portion of the features generated by the current model are local features, which means they include particular information about the original images from which they were taken. This means that our machine learning model is unable to generalize because just a small percentage of the collected features are indicative of the data. As previously mentioned, the best MAP results are obtained by the IR system, which is dependent on the proposal. The work IR system will be more precise because the suggested method can identify the dataset classes more effectively. As a result, more true positives relevant photos related to the query are found during retrieval, which leads to the collection of recovered articles. It is possible to argue that sorting based on the similarity function in our example, the cosine is preferable. An improved option would be to utilize contrasting loss, which drives examples from other classes far away and maximizes the training target by pushing all comparable class instances to move infinitesimally closer to one another in the output embedding field. Triplet loss is a phenomenon that accounts for both positive and negative pair distances at the same place, favouring data points within the same class to be closer to one another than to a

data point from a different class. Assuming that triplet loss is our loss function of choice. In that scenario, to increase training convergence and computational complexity, we should also take into account a suitable strategy for mining informative points. To increase accuracy, common sampling techniques include batch all, batch hard, batch weighted, and batch sample.

5.4. Summary

Key Findings and Results: Experiments on the CIFAR-10 dataset, along with comparisons to other methods, yielded several key findings superior retrieval performance The SO-DRCNN system with Siamese-Driven Feature Fusion achieved a mAP of 0.8971, significantly outperforming baseline methods such as BoVW (mAP = 0.6832), VGG-16 (mAP = 0.6941), SPQ (mAP = 0.7052) and SGSH (mAP=0.7581). This demonstrates the effectiveness of our hybrid approach and learned fusion strategy. High Accuracy and Precision: Our system achieved a testing accuracy of 93% and a training accuracy of 95% on CIFAR-10. Furthermore, the system demonstrated high precision (0.93), recall (0.95), sensitivity (0.94), specificity (0.93), and F-measure (0.95), with 433 true positives, 425 true negatives, 11 false positives, and 13 false negatives. Effectiveness of Ternion Descriptors: Experiments with different combinations of Ternion descriptors showed that using all three (HOG + ICH + SERC) consistently yielded the best results, with accuracy reaching 96.3% and precision reaching 96.2% in our best configuration. This confirms the complementary nature of these descriptors. Siamese-Driven Fusion Outperforms Baselines: The learned fusion weights in our Siamese-Driven Feature Fusion approach demonstrably improved performance compared to simple concatenation or using individual feature types alone.

Efficient Retrieval: The system achieved an average retrieval time in the tens of milliseconds for a database of 10,000 images, demonstrating its scalability and suitability for real-time applications.

Robustness to Noise and Variations: The use of data augmentation, regularization techniques (Gaussian noise, dropout), and the inherent robustness of the ResNet-50 architecture contributed to the system's ability to generalize well to unseen data and variations in image appearance.

Chapter 6

Conclusion

This thesis has presented the SO-DRCNN CBIR, a novel approach designed to address the limitations of existing CBIR systems, particularly the semantic gap and the reliance on extensive manual labeling. The core innovation is the Siamese-Driven Feature Fusion mechanism, which leverages a self-supervised Siamese network to learn an adaptive and data-driven strategy for combining handcrafted features BoVW with Ternion descriptors: HOG, ICH, SERC and deep CNN embeddings (from a ResNet-50 backbone enhanced with Recurrent Patching, SPP/ASPP, and Attention).

6.1. Key Findings and Results

Experiments conducted on the CIFAR-10 dataset, a standard benchmark for image classification and retrieval, demonstrated the effectiveness of the proposed SO-DRCNN Hybrid CBIR system:

Superior Retrieval Performance: The SO-DRCNN system with Siamese-Driven Feature Fusion achieved a mAP of 0.8971, significantly outperforming several baseline methods. This demonstrates the synergistic benefits of combining handcrafted and deep features through a learned fusion strategy.

High Accuracy and Precision: The system achieved a training accuracy of 95% and a testing accuracy of 93% on CIFAR-10, showcasing its ability to learn discriminative features and generalize well to unseen data. High precision (0.93), recall (0.95), sensitivity (0.94), specificity (0.93), and F-measure (0.95) values further validate the system's effectiveness.

Comparison to Baselines: The proposed SO-DRCNN significantly outperformed alternative approaches, including:

Handcrafted Features Only (BoVW): Demonstrated the limitations of relying solely on handcrafted features for capturing complex semantic similarity.

CNN Embeddings Only (VGG-16, ResNet-50, Inception v1): Showed the added value of incorporating handcrafted features and the SO-DRCNN architectural enhancements.

Other Methods (SGSH, SPQ): Outperformed these methods, highlighting the effectiveness of the Siamese-Driven Feature Fusion and self-supervised training.

Summary:

This research makes the following key contributions to the field of Content-Based Image Retrieval:

Novel Siamese-Driven Feature Fusion: Introduces a novel and effective approach to feature fusion that leverages the power of Siamese networks and contrastive learning to optimize the combination of handcrafted and deep features for semantic similarity.

Self-Supervised Training for Hybrid CBIR: Demonstrates the effectiveness of self-supervised learning for training a hybrid CBIR system, reducing the reliance on labeled data and enhancing domain adaptation.

Enhanced SO-DRCNN Architecture: Presents an enhanced SO-DRCNN architecture that integrates Recurrent Patching, SPP/ASPP, and Attention mechanisms to capture rich and contextualized semantic image representations.

Empirical Validation and Benchmarking: Provides rigorous empirical validation of the proposed approach on standard benchmark datasets, demonstrating its performance advantages over existing methods and establishing a strong baseline for future research.

Limitations:

While the SO-DRCNN system demonstrates strong performance, it's important to acknowledge certain limitations:

Dataset Dependence: The experiments were primarily conducted on CIFAR-10. While CIFAR-10 is a standard benchmark, further evaluation on larger and more diverse datasets is necessary to fully assess the system's generalization capabilities to real-world image distributions.

Computational Cost of Feature Extraction: Although the Siamese-Driven Fusion aims for efficiency, the overall feature extraction process (including both CNN and handcrafted features) still has a computational cost. Further optimization might be needed for extremely large-scale or real-time applications.

Interpretability of Fusion Weights: While the Siamese-Driven Fusion offers some interpretability through the learned weights, further research into visualizing and understanding the fusion process could enhance transparency.

6.2. Future Work

This research opens up several promising avenues for future work:

Exploring More Advanced Fusion Techniques: Investigate more sophisticated fusion methods beyond weighted concatenation, such as attention-based fusion or learned fusion layers, to further improve the integration of handcrafted and deep features.

Domain-Specific Adaptation: Apply and adapt the SO-DRCNN framework to specific image domains, such as medical imaging, remote sensing, or satellite imagery, to explore its effectiveness in specialized retrieval tasks.

Incorporating Additional Modalities: Extend the system to incorporate additional modalities, such as text descriptions or user feedback, to create a multimodal CBIR system.

Real-Time Implementation and Optimization: Further optimize the system for real-time performance, potentially through model compression techniques or hardware acceleration.

Enhanced Interpretability Methods: Develop and integrate more advanced interpretability techniques (e.g., Grad-CAM, feature visualization) to better understand the decision-making process of the SO-DRCNN model and the Fusion Module.

Addressing Failure Cases: Investigate and address the limitations identified in the failure case analysis, such as complex scenes with multiple objects, by exploring techniques like multi-query approaches or region-based retrieval.

6.3. Implications

The proposed SO-DRCNN Hybrid CBIR system has significant implications for various real-world applications. Its ability to learn from unlabeled data, achieve high retrieval accuracy, and incorporate interpretable features makes it a valuable tool for:

Medical Image Retrieval: Assisting medical professionals in diagnosing disorders by retrieving similar medical images and comparing instances.

Crime Scene Investigation: Matching crime scene photos with pre-existing databases to identify suspects or locate pertinent evidence.

Art History Research and Restoration: Helping galleries and museums find related art pieces or antiques.

Remote Sensing: Supporting scientists in tracking changes in land cover, deforestation, and natural disasters by analyzing satellite photos.

Video Surveillance: Improving video surveillance systems by facilitating the detection of questionable activity and tracking individuals across video streams.

E-commerce: Enabling users to quickly find visually similar products (e.g., clothing, furniture).

Overall, the SO-DRCNN Hybrid CBIR system, with its Siamese-Driven Feature Fusion and self-supervised training, represents a significant advancement in the field of content-based image retrieval. It offers a robust, efficient, scalable, and potentially more interpretable solution for searching and retrieving images based on their semantic visual content, paving the way for more powerful and user-friendly image search systems in a wide range of applications

References

Aboali, M. A., Elmaddah, I., & Abdelmunim, H. E. (2023). Neural textual features composition for CBIR. IEEE Access, 11, 28506–28521. https://doi.org/10.1109/ACCESS.2023.3259737

Adil, E. (2021). Content-based image retrieval using hierarchical decomposition of feature descriptors [Master's thesis, University of Windsor]. Scholarship at UWindsor. https://scholar.uwindsor.ca/etd/8779

Almohammed, N. (2021). Content based image retrieval with high level semantics [Master's thesis, Altınbaş University, Istanbul, Turkey]. Altınbaş University Institutional Repository. http://openaccess.altinbas.edu.tr/

Alyosef, A. (2023). Large scale partial-and near-duplicate image retrieval using spatial information of local features [Doctoral dissertation, University of Surrey]. University of Surrey Institutional Repository. https://epubs.surrey.ac.uk/871651/

Amitha, I. C., Sreekanth, N., & Narayanan, N. (2021). Collaborative multi-resolution MSER and faster RCNN (MRMSER-FRCNN) model for improved object retrieval of poor resolution images. International Journal of Advanced Computer Science and Applications, 12(12). https://doi.org/10.14569/IJACSA.2021.0121270

Ammatmanee, C. (2022). Fast embedding for image classification & retrieval and its application to the hostel industry [Master's thesis, King Mongkut's University of Technology Thonburi]. KMUTT Library Institutional Repository. http://bura.brunel.ac.uk/handle/2438/25419

Murala, S., & Wu, Q. M. J. (2014). MRI and CT image indexing and retrieval using local mesh peak valley edge patterns. Signal Processing: Image Communication, 29, 1–14. https://doi.org/10.1016/j.image.2013.12.002

Babenko, A., Slesarev, A., Chigorin, A., & Lempitsky, V. (2014). Neural codes for image retrieval. In Computer Vision – ECCV 2014 (pp. 584–599). Springer. https://doi.org/10.1007/978-3-319-10590-1_38

Bose, S., Pal, A., Mallick, J., Kumar, S., & Rudra, P. (2015). A hybrid approach for improved content-based image retrieval using segmentation. arXiv. https://doi.org/10.48550/arXiv.1502.03215

Breznik, E. (2023). Image processing and analysis methods for biomedical applications [Doctoral dissertation, Uppsala University]. Acta Universitatis Upsaliensis. https://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-498953

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., & Askell, A. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877–1901. https://doi.org/10.48550/arXiv.2005.14165

Buades, A., Coll, B., & Morel, J.-M. (2005). A non-local algorithm for image denoising. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) (Vol. 2, pp. 60-65). IEEE. https://doi.org/10.1109/CVPR.2005.38

Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6), 679 - 698. https://doi.org/10.1109/TPAMI.1986.4767851

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (pp. 9630–9640). https://doi.org/10.1109/ICCV48922.2021.00951

Carreira-Perpiñán, M. Á., & Raziperchikolaei, R. (2015). Hashing with binary autoencoders. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 557–566. https://doi.org/10.1109/CVPR.2015.7298688

Castro Medina, F., Rodríguez, L., Lopez-Chau, A., Cervantes, J., Alor-Hernández, G., & Machorro-Cano, I. (2020). Application of dynamic fragmentation methods in multimedia databases: A review. Entropy, 22(12), 1352. https://doi.org/10.3390/e22121352

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5), 603–619. https://doi.org/10.1109/34.1000236

Chan, S. W. (2022). Advances in query intention construction and learning in image retrieval [Master's thesis, Hong Kong Baptist University]. HKBU Scholars. https://scholars.hkbu.edu.hk/en/studentTheses/advances-in-query-intention-construction-and-learning-in-image-re

Chen, C., Tu, D., Zhou, Q., Zhou, J., Wang, X., Cherdchim, B., & Ou, R. (2020). Development and evaluation of a surface-densified wood composite with an asymmetric structure. Construction and Building Materials, 242, 118007. https://doi.org/10.1016/j.conbuildmat.2020.118007

Chopra, V., Sharma, A., Nagpal, J., Lahrod, T., Jain, P., Srivastava, V., & Gupta, K. (2021). A survey of various wavelet-based image retrieval techniques and tuning of hyperparameters. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3842651

Chugh, H., Gupta, S., & Garg, M. (2021). Image retrieval system—An integrated approach. IOP Conference Series: Materials Science and Engineering, 1022(1), Article 012040. https://doi.org/10.1088/1757-899X/1022/1/012040

da Silva, M. F. F. (2023). Chest radiography content-based image retrieval [Master's thesis, Universidade do Porto]. Universidade do Porto Digital Repository. https://hdl.handle.net/10216/150000

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) (Vol. 1, pp. 886-893). IEEE. https://doi.org/10.1109/CVPR.2005.177

Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys, 40(2), Article 5. https://doi.org/10.1145/1348246.1348248

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv. https://doi.org/10.48550/arXiv.1810.04805

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (pp. 248–255). IEEE. https://doi.org/10.1109/CVPR.2009.5206848

Doersch, C., & Zisserman, A. (2017). Multi-task self-supervised visual learning. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV) (pp. 2070–2079). IEEE. https://doi.org/10.1109/ICCV.2017.226

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv. https://doi.org/10.48550/arXiv.2010.11929

Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., & Equitz, W. (1994). Efficient and effective querying by image content. Journal of Intelligent Information Systems, 3(3-4), 231-262. https://doi.org/10.1007/BF00962238

Felzenszwalb, P., Girshick, R., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9), 1627–1645. https://doi.org/10.1109/TPAMI.2009.167

Feng, G., Jiang, Z., Tan, X., & Cheng, F. (2022). Hierarchical clustering-based image retrieval for indoor visual localization. Electronics, 11(21), 3609. https://doi.org/10.3390/electronics11213609

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. A., & Mikolov, T. (2013). DeViSE: A deep visual-semantic embedding model. In Advances in Neural Information Processing Systems 26 (pp. 2121-2129). NeurIPS. https://papers.nips.cc/paper_files/paper/2013/hash/7cce53cf90577442771720a370c3c723-Abstract.html

Garg, M., & Dhiman, G. (2021). A novel content-based image retrieval approach for classification using GLCM features and texture fused LBP variants. Neural Computing and Applications, 33(4), 1311–1328. https://doi.org/10.1007/s00521-020-05017-z

Gayathri, N., & Mahesh, K. (2019). An efficient video indexing and retrieval algorithm using ensemble classifier. In Proceedings of the 2019 IEEE International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT) (pp. 250-258). IEEE. https://doi.org/10.1109/ICEECCOT46775.2019.9114831

Georgiou, T. (2021). Multi-dimensional feature and data mining [Doctoral dissertation, Leiden University]. Leiden University Scholarly Publications. https://scholarlypublications.universiteitleiden.nl/handle/1887/3214119

Ghazouani, H. (2023). Image analysis and understanding: Application to texture classification, facial expression recognition, and breast cancer diagnosis [Habilitation thesis, Université de Carthage]. ResearchGate. https://www.researchgate.net/publication/369033149

Ghazouani, H., & Barhoumi, W. (2021). Towards non-data-hungry and fully-automated diagnosis of breast cancer from mammographic images. Computers in Biology and Medicine, 139, Article 105011. https://doi.org/10.1016/j.compbiomed.2021.105011

Ghrabat, M. J. J., Ma, G., Maolood, I. Y., Alresheedi, S. S., & Abduljabbar, Z. A. (2019). An effective image retrieval based on optimized genetic algorithm utilized a novel SVM-based convolutional neural network classifier. Human-

centric Computing and Information Sciences, 9, Article 25. https://doi.org/10.1186/s13673-019-0191-8

Giannoulakis, S., Tsapatsoulis, N., & Djouvas, C. (2023). Evaluating the use of Instagram images color histograms and hashtags sets for automatic image annotation. Frontiers in Big Data, 6, 1149523. https://doi.org/10.3389/fdata.2023.1149523

Bengio, Y., & Glorot, X. (2010). Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (pp. 249-256). JMLR. http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf

Gonzalez, R. C., & Woods, R. E. (2018). Digital image processing: Global edition (4th ed.). Pearson. https://books.google.ca/books?id=p74oEAAAQBAJ

Heaton, J. (2018). [Review of the book Deep learning, by I. Goodfellow, Y. Bengio, & A. Courville]. Genetic Programming and Evolvable Machines, 19(1), 305–307. https://doi.org/10.1007/s10710-017-9314-z

Goodrum, A. A. (2000). Image information retrieval: An overview of current research. Informing Science: The International Journal of an Emerging Transdiscipline, 3, 63–66. https://doi.org/10.28945/578

Gormley, C., & Tong, Z. (2015). Elasticsearch: The definitive guide. O'Reilly Media. https://www.elastic.co/guide/en/elasticsearch/guide/current/index.html

Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In 2006 IEEE Computer Society Conference on

Computer Vision and Pattern Recognition (CVPR'06) (Vol. 2, pp. 1735–1742). IEEE. https://doi.org/10.1109/CVPR.2006.100

Han, J., & Ma, K. K. (2002). Fuzzy color histogram and its use in color image retrieval. IEEE Transactions on Image Processing, 11(8), 944–952. https://doi.org/10.1109/TIP.2002.801585

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 9729–9738). https://doi.org/10.1109/CVPR42600.2020.00975

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In 2017 IEEE International Conference on Computer Vision (ICCV) (pp. 2980-2988). IEEE. https://doi.org/10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770-778). IEEE. https://doi.org/10.1109/CVPR.2016.90

He, P., Wu, A., Huang, X., Rangarajan, A., & Ranka, S. (2022). Machine learning-based highway truck commodity classification using logo data. Applied Sciences, 12(4), 2075. https://doi.org/10.3390/app12042075

He, X., Zhou, Y., Zhou, Z., Bai, S., & Bai, X. (2018). Triplet-center loss for multi-view 3D object retrieval. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 1945–1954). IEEE. https://doi.org/10.1109/CVPR.2018.00208

Hu, C. (2021). Delving deep into the sketch and photo relation [Doctoral dissertation, University of Surrey]. https://doi.org/10.15126/thesis.900131

Hu, Z. (2022). Deep learning with query sensitive attention mechanisms for content-based image retrieval [Doctoral dissertation, University of York]. White Rose eTheses Online. https://etheses.whiterose.ac.uk/32394/

Imbriaco, R. (2024). Representation learning for street-view and aerial image retrieval (Doctoral dissertation, Eindhoven University of Technology). https://pure.tue.nl/ws/portalfiles/portal/316073373/20240118_Imbriaco_hf.pdf

Chen, J., Zhang, L., Bai, C., & Kpalma, K. (2020). Review of recent deep learning based methods for image-text retrieval. In Proceedings of the 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) (pp. 167-172). IEEE. https://doi.org/10.1109/MIPR49039.2020.00042

Jain, R., Jain, R. C., JAIN, R. A., Jaina, R. K., Kasturi, R., Schunck, B. G., & Schunck, B. G. (1995). Machine Vision. McGraw-Hill.

Jardim, S., António, J., Mora, C., & Almeida, A. (2022). A novel trademark image retrieval system based on multi-feature extraction and deep networks. Journal of Imaging, 8(9), 238. https://doi.org/10.3390/jimaging8090238

Jaruenpunyasak, J., & Duangsoithong, R. (2021). Empirical analysis of feature reduction in deep learning and conventional methods for foot image classification. IEEE Access, 9, 53133–53145. https://doi.org/10.1109/ACCESS.2021.3069625

Jégou, H., Douze, M., & Schmid, C. (2011). Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(1), 117-128. https://doi.org/10.1109/TPAMI.2010.57

Jiang, L., Zhou, Z., Leung, T., Li, L., & Li, F.-F. (2018). MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In Proceedings of the 35th International Conference on Machine Learning (Vol. 80, pp. 2304-2313). Proceedings of Machine Learning Research. https://proceedings.mlr.press/v80/jiang18c.html

Kabir, M. M., Ishraq, A., Nur, K., & Mridha, M. F. (2022). Content-based image retrieval using AutoEmbedder. Journal of Advances in Information Technology, 13(3), 240-248. https://doi.org/10.12720/jait.13.3.240-248

Kanwal, K., Ahmad, K., Khan, R., Naji, A., & Li, J. (2021). Deep learning using isotroping, laplacing, eigenvalues interpolative binding, and convolved determinants with normed mapping for large-scale image retrieval. Sensors, 21(4), 1139. https://doi.org/10.3390/s21041139

Kapoor, R., Sharma, D., & Gulati, T. (2021). State of the art content based image retrieval techniques using deep learning: A survey. Multimedia Tools and Applications, 80(19), 29561-29583. https://doi.org/10.1007/s11042-021-11045-1

Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. International Journal of Computer Vision, 1(4), 321-331. https://doi.org/10.1007/BF00133570

Kekre, H. B., Thepade, S., & Dheringe, T. K. (2010). Performance comparison of image retrieval using BTC and a novel color space Kekre's LUV.

Proceedings of the International Conference & Workshop on Emerging Trends in Technology (pp. 73–78). Association for Computing Machinery. https://doi.org/10.1145/1741906.1741928

Khan, K., Baharudin, B., Lee, L. H., & Khan, A. (2010). A review of machine learning algorithms for text-documents classification. Journal of Advances in Information Technology, 1(1), 4-20. https://doi.org/10.4304/jait.1.1.4-20

Kim, S., Seo, M., Laptev, I., Cho, M., & Kwak, S. (2019). Deep metric learning beyond binary supervision (arXiv:1904.09626). arXiv. https://doi.org/10.48550/arXiv.1904.09626

Kostelecká, A. (2022). Content-based image retrieval: From primitive to advanced techniques [Doctoral dissertation, Charles University]. Charles University Digital Repository. https://dspace.cuni.cz/handle/20.500.11956/173675

Kumar, A., Singh, K. U., Raja, L., Singh, T., Swarup, C., & Kumar, A. (2021). Design a framework for content based image retrieval using hybrid features analysis. Traitement du Signal, 38(5), 1449–1459. https://doi.org/10.18280/ts.380520

Kumar, V., Tripathi, V., Pant, B., Alshamrani, S. S., Dumka, A., Gehlot, A., Singh, R., Rashid, M., Alshehri, A., & AlGhamdi, A. S. (2022). Hybrid spatiotemporal contrastive representation learning for content-based surgical video retrieval. Electronics, 11(9), 1353. https://doi.org/10.3390/electronics11091353

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

Li, F., Zou, C., Yun, J., Huang, L., Liu, Y., Tao, B., & Xie, Y. (2024). A depth awareness and learnable feature fusion network for enhanced geometric perception in semantic correspondence. Sensors, 24(20), 6680. https://doi.org/10.3390/s24206680

Li, L.-J., & Fei-Fei, L. (2010). OPTIMOL: Automatic online picture collection via incremental model learning. International Journal of Computer Vision, 88(2), 147–168. https://doi.org/10.1007/s11263-009-0265-6

Li, X., Yang, J., & Ma, J. (2021). Recent developments of content-based image retrieval (CBIR). Neurocomputing, 452, 675–689. https://doi.org/10.1016/j.neucom.2020.07.139

Liang, H., Bao, W., & Shen, X. (2021). Adaptive weighting feature fusion approach based on generative adversarial network for hyperspectral image classification. Remote Sensing, 13(2), 198. https://doi.org/10.3390/rs13020198

Liao, K., Lin, J., Zheng, Y., Wang, K., & Feng, W. (2024). Incremental image retrieval method based on feature perception and deep hashing. International Journal of Multimedia Information Retrieval, 13(1), 10. https://doi.org/10.1007/s13735-024-00319-7

Liu, G.-H., & Yang, J.-Y. (2013). Content-based image retrieval using color difference histogram. Pattern Recognition, 46(1), 188–198. https://doi.org/10.1016/j.patcog.2012.06.001

Liu, Z., Shi, S., Duan, Q., Zhang, W., & Zhao, P. (2019). Salient object detection for RGB-D image by single stream recurrent convolution neural network. Neurocomputing, 363, 56–65. https://doi.org/10.1016/j.neucom.2019.07.012

Liu, W., Wang, J., Kumar, S., & Chang, S.-F. (2011). Hashing with graphs. In Proceedings of the 28th International Conference on Machine Learning (ICML 2011) (pp. 1-8). Omnipress..

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. Medical Image Analysis, 42, 60–88. https://doi.org/10.1016/j.media.2017.07.005

Müller-Budack, E., Pustu-Iren, K., & Ewerth, R. (2018). Geolocation estimation of photos using a hierarchical model and scene classification. In Proceedings of the European Conference on Computer Vision (ECCV 2018) (pp. 525-541). Springer. https://doi.org/10.1007/978-3-030-01258-8_35

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In Proceedings of the Seventh IEEE International Conference on Computer Vision (Vol. 2, pp. 1150-1157). IEEE. https://doi.org/10.1109/ICCV.1999.790410

Majhi, M., Pal, A. K., Pradhan, J., Islam, S. K. N., & Khan, K. (2021). Computational intelligence-based secure three-party CBIR scheme for medical data for cloud-assisted healthcare applications. Multimedia Tools and Applications, 80(20), 31047–31080. https://doi.org/10.1007/s11042-020-10483-7

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. https://doi.org/10.48550/arXiv.1301.3781

Misra, I., & van der Maaten, L. (2019). Self-supervised learning of pretext-invariant representations (arXiv:1912.01991). arXiv. https://doi.org/10.48550/arXiv.1912.01991

Muja, M., & Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In Proceedings of the Fourth International Conference on Computer Vision Theory and Applications (pp. 331-340). https://doi.org/10.5220/0001787803310340

Muthukkumar, R., & Seenivasagam, V. (2022). Enhancing scalability of image retrieval using visual fusion of feature descriptors. Intelligent Automation & Soft Computing, 31(3), 1737–1752. https://doi.org/10.32604/iasc.2022.018822

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011, June). Multimodal deep learning. In Proceedings of the 28th International Conference on Machine Learning (ICML) (pp. 689–696). Bellevue, WA, USA. https://people.csail.mit.edu/khosla/papers/icml2011_ngiam.pdf

Norouzi, M., Punjani, A., & Fleet, D. J. (2013). Fast exact search in Hamming space with multi-index hashing. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(7), 1448–1455. https://doi.org/10.1109/TPAMI.2013.231

Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), 62–66. https://doi.org/10.1109/TSMC.1979.4310076

Oyewole, S. A. (2021). Image content-based user preference elicitation for personalised mobile recommendation of shopping items [Doctoral

dissertation, Durban University of Technology]. DUT Open Scholar. https://doi.org/10.51415/10321/3827

Papakonstantinou, T. (2023). Content-based video encoding based on spatial and temporal information [Master's thesis, Uppsala University]. DiVA. https://uu.diva-portal.org/smash/get/diva2:1795897/FULLTEXT01.pdf

Patil, J., & Kumar, R. (2013). Plant leaf disease image retrieval using color moments. IAES International Journal of Artificial Intelligence, 2(1), 36–42. https://doi.org/10.11591/ij-ai.v2i1.1319

Patil, J. K., & Kumar, R. (2017). Analysis of content-based image retrieval for plant leaf diseases using color, shape and texture features. Engineering in Agriculture, Environment and Food, 10(2), 69-78. https://doi.org/10.1016/j.eaef.2016.11.004

Perez, D. (2021). Feature extraction and design in deep learning models [Doctoral dissertation, Old Dominion University]. Old Dominion University Digital Commons. https://digitalcommons.odu.edu/msve_etds/61

Pizer, S. M., Amburn, E. P., Austin, J. D., Cromartie, R., Geselowitz, A., Greer, T., ter Haar Romeny, B., Zimmerman, J. B., & Zuiderveld, K. (1987). Adaptive histogram equalization and its variations. Computer Vision, Graphics, and Image Processing, 39(3), 355-368. https://doi.org/10.1016/S0734-189X(87)80186-X

Prasomphan, S., & Pinngoen, N. (2021). Feature extraction for image matching in Wat Phra Chetuphon Wimonmangklararam balcony painting with SIFT algorithms. In 2021 IEEE 4th International Conference on Computer and

Communication Engineering Technology (CCET) (pp. 79-84). IEEE. https://doi.org/10.1109/CCET52649.2021.9544279

Pratt, W. K. (2007). Digital image processing: PIKS Scientific inside (4th ed.). Wiley-Interscience.

Radford, A., Kim, J. W., Hallacy, C., [...] Sutskever, I. (2021). Learning transferable visual models from natural language supervision. arXiv. https://doi.org/10.48550/arXiv.2103.00020

Radha, K., Sudha, R. V., Meena, M., Jayavadivel, R., Kanimozhi, S., & Prabaharan, P. (2021). Modified cuckoo search algorithm: Feature subset selection & shape, color and texture features descriptors for content-based image retrieval. Journal of University of Shanghai for Science and Technology, 23(12), 525–541. https://doi.org/10.51201/JUSST/21/121046

Raghuwanshi, G., & Tyagi, V. (2021). Texture image retrieval using hybrid directional Extrema pattern. Multimedia Tools and Applications, 80(1), 1–23. https://doi.org/10.1007/s11042-020-09618-7

Rayavaram, P. (2023). Single image trained unsupervised CBIR for real time environments [Master's thesis, National Institute of Technology Rourkela]. NIT Rourkela Institutional Repository.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. arXiv. https://doi.org/10.48550/arXiv.1506.02640

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. arXiv. https://doi.org/10.48550/arXiv.1506.01497

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. arXiv. https://doi.org/10.48550/arXiv.1505.04597

Rout, N. K., Atulkar, M., & Ahirwal, M. K. (2021). A review on content-based image retrieval system: Present trends and future challenges. International Journal of Computational Vision and Robotics, 11(5), 461–485. https://doi.org/10.1504/IJCVR.2021.117578

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In 2011 International Conference on Computer Vision (ICCV) (pp. 2564–2571). IEEE. https://doi.org/10.1109/ICCV.2011.6126544

Rui, Y., Huang, T. S., & Chang, S.-F. (1999). Image retrieval: Current techniques, promising directions, and open issues. Journal of Visual Communication and Image Representation, 10(1), 39-62. https://doi.org/10.1006/jvci.1999.0413

Sain, A. (2023). Exploring sketch traits for democratising sketch based image retrieval [Doctoral dissertation, University of Surrey]. University of Surrey Institutional Repository. http://epubs.surrey.ac.uk/871666/

Saikia, S. (2021). Image feature representation using deep learning for instance search and scene recognition [Doctoral dissertation, Universidad de León]. Buleria (University of León Institutional Repository). https://buleria.unileon.es/handle/10612/15059

Sait, A. R. W., & Nagaraj, R. (2025). Diabetic foot ulcers detection model using a hybrid convolutional neural networks–vision transformers. Diagnostics, 15(6), 736. https://doi.org/10.3390/diagnostics15060736

Salih, S. F., & Abdulla, A. A. (2021). An improved content based image retrieval technique by exploiting bi-layer concept. UHD Journal of Science and Technology, 5(1), Article 1. https://doi.org/10.21928/uhdjst.v5n1y2021.pp1-12

Salton, G., & McGill, M. J. (1983). Introduction to modern information retrieval. McGraw-Hill.

Saxe, A. M., McClelland, J. L., & Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv. https://doi.org/10.48550/arXiv.1312.6120

Sculley, D. (2010). Web-scale k-means clustering. In Proceedings of the 19th International Conference on World Wide Web (pp. 1177–1178). Association for Computing Machinery. https://doi.org/10.1145/1772690.1772862

Sedmidubsky, J., Elias, P., Budíková, P., & Zezula, P. (2021). Content-based management of human motion data: Survey and challenges. IEEE Access, 9, 64241-64255. https://doi.org/10.1109/ACCESS.2021.3075766

Shen, F., Xu, Y., Liu, L., Yang, Y., Huang, Z., & Shen, H. T. (2018). Unsupervised deep hashing with similarity-adaptive and discrete optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(12), 3034-3044. https://doi.org/10.1109/TPAMI.2018.2789887

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8), 888-905. https://doi.org/10.1109/34.868688

Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., & Jain, R. (2000). Content-based image retrieval at the end of the early years. IEEE

Transactions on Pattern Analysis and Machine Intelligence, 22(12), 1349-1380. https://doi.org/10.1109/34.895972

Song, J., He, T., Gao, L., Xu, X., Hanjalic, A., & Shen, H. T. (2018). Binary generative adversarial networks for image retrieval. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1). https://doi.org/10.1609/aaai.v32i1.11276

Srivastava, D., Singh, S. S., Rajitha, B., Verma, M., Kaur, M., & Lee, H.-N. (2023). Content-based image retrieval: A survey on local and global features selection, extraction, representation, and evaluation parameters. IEEE Access, 11, 95410–95431. https://doi.org/10.1109/ACCESS.2023.3308911

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929-1958. http://www.jmlr.org/papers/v15/srivastava14a.html

Swain, M. J., & Ballard, D. H. (1991). Color indexing. International Journal of Computer Vision, 7(1), 11-32. https://doi.org/10.1007/BF00130487

Szeliski, R. (2010). Computer vision: Algorithms and applications. Springer Science & Business Media. https://doi.org/10.1007/978-1-84882-935-0

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2016). Inception-v4, Inception-ResNet and the impact of residual connections on learning. Proceedings of the AAAI Conference on Artificial Intelligence, 31(1). https://doi.org/10.1609/aaai.v31i1.11231

Tena, S., Hartanto, R., & Ardiyanto, I. (2021). East Nusa Tenggara weaving image retrieval using convolutional neural network. Proceedings of the 2021 4th

International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 150–154. https://doi.org/10.1109/ISRITI54043.2021.9702843

Tian, Y., Krishnan, D., & Isola, P. (2020). Contrastive multiview coding. arXiv. https://doi.org/10.48550/arXiv.1906.05849

Torres, J., & Reis, L. P. (2008). Relevance feedback in conceptual image retrieval: A user evaluation. arXiv. https://doi.org/10.48550/arXiv.0809.4834

Tsai, Y.-H. H., Wu, Y., Salakhutdinov, R., & Morency, L.-P. (2021). Self-supervised learning from a multi-view perspective. In Proceedings of the 9th International Conference on Learning Representations (ICLR). https://openreview.net/forum?id=-bdp_8Itjwp

Uddin Molla, M. (2021). Content-based image retrieval using Color Coherence Vector and Gabor Filter [Doctoral dissertation, National Institute of Technology]. Institutional Repository of National Institute of Technology.

Veselý, P., & Peška, L. (2023). Less is more: Similarity models for content-based video retrieval. In D.-T. Dang-Nguyen, C. Gurrin, M. Larson, & A. F. Smeaton (Eds.), MultiMedia Modeling: 29th International Conference, MMM 2023, Proceedings (Lecture Notes in Computer Science, Vol. 13833, pp. 54-65). Springer. https://doi.org/10.1007/978-3-031-27818-1_5

Vo, T.-N., Coustaty, M., Guillaume, J.-L., Nguyen, T.-K., & Tran, D. C. (2021). Applying segmented images by Louvain method into content-based image retrieval. In P. Cong Vinh & A. Rakib (Eds.), Context-aware systems and applications: ICCASA 2021 (Lecture Notes in Computer Science, Vol. 409, pp. 77–90). Springer. https://doi.org/10.1007/978-3-030-93179-7_7

Wang, D., Guo, L., Zhong, J., Yu, H., Tang, Y., Peng, L., Cai, Q., Qi, Y., Zhang, D., & Lin, P. (2024). A novel deep-learning based weighted feature fusion architecture for precise classification of pressure injury. Frontiers in Physiology, 15, Article 1304829. https://doi.org/10.3389/fphys.2024.1304829

Wang, J. Z., Li, J., & Wiederhold, G. (2001). SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(9), 947-963. https://doi.org/10.1109/34.955109

Wang, W., Zhang, H., Zhang, Z., Liu, L., & Shao, L. (2021). Sparse graph-based self-supervised hashing for scalable image retrieval. Information Sciences, 547, 622–640. https://doi.org/10.1016/j.ins.2020.08.092

Wang, Z., Liu, X., Li, H., Shi, J., & Rao, Y. (2018). A saliency detection based unsupervised commodity object retrieval scheme. IEEE Access, 6, 49902–49912. https://doi.org/10.1109/ACCESS.2018.2868139

Warburg, F., Jørgensen, M., Civera, J., & Hauberg, S. (2021). Bayesian triplet loss: Uncertainty quantification in image retrieval. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (pp. 12138–12148). IEEE. https://doi.org/10.1109/ICCV48922.2021.01194

Weiss, Y., Torralba, A., & Fergus, R. (2008). Spectral hashing. In Advances in Neural Information Processing Systems 21 (pp. 1753-1760). https://papers.nips.cc/paper/3383-spectral-hashing

Xu, H., Li, C., Rahaman, M. M., Yao, Y., Li, Z., Zhang, J., Kulwa, F., Zhao, X., Qi, S., & Teng, Y. (2020). An enhanced framework of generative adversarial networks (EF-GANs) for environmental microorganism image augmentation

with limited rotation-invariant training data. IEEE Access, 8, 187350-187361. https://doi.org/10.1109/ACCESS.2020.3031059.

Xun, H. (2024). Research on automatic recognition technology of library books based on image processing. Informatica, 48(5). https://doi.org/10.31449/inf.v48i5.5345

Yang, F., Ismail, N. A., Pang, Y. Y., Kebande, V. R., Ai-Dhaqm, A., & Koh, T. W. (2024). A systematic literature review of deep learning approaches for sketch-based image retrieval: Datasets, metrics, and future directions. IEEE Access, 12, 14847–14869. https://doi.org/10.1109/ACCESS.2024.3357939

Yu, M., Xu, H., Zhou, F., Xu, S., & Yin, H. (2023). A deep-learning-based multimodal data-fusion framework for urban region function recognition. ISPRS International Journal of Geo-Information, 12(12), Article 468. https://doi.org/10.3390/ijgi12120468

Zhang, D., & Lu, G. (2002). Shape-based image retrieval using generic Fourier descriptor. Signal Processing: Image Communication, 17(10), 825–848. https://doi.org/10.1016/S0923-5965(02)00084-X

Zhang, Z., Chen, Y., & Saligrama, V. (2016). Efficient training of very deep neural networks for supervised hashing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 1487–1495). IEEE. https://doi.org/10.1109/CVPR.2016.165

Zhang, Z. (2021). Dataset-driven instance image retrieval [Doctoral dissertation, University of Wollongong]. University of Wollongong Research Online. https://hdl.handle.net/10779/uow.27667437.v1

Zhou, X., Shen, F., Liu, L., Liu, W., Nie, L., Yang, Y., & Shen, H. T. (2020). Graph convolutional network hashing. IEEE Transactions on Cybernetics, 50(4), 1460–1472. https://doi.org/10.1109/TCYB.2018.2883970

Zhuang, Y., Chen, S., Jiang, N., & Hu, H. (2022). An effective WSSENet-based similarity retrieval method of large lung CT image databases. KSII Transactions on Internet and Information Systems, 16(7), 2308-2325. https://doi.org/10.3837/tiis.2022.07.007