PROJECT ABSTRACT

Content-Based Image Retrieval using Color Histograms

A content-based image retrieval engine ranking photos by real color-histogram and edge features with chi-squared distance, evaluated on the Wang (Corel-1K) benchmark, with a live search demo, evaluation harness and viva kit. Delivered built-to-order with report, PPT and Q&A.

01. Title

Content-Based Image Retrieval using Color Histograms

02. Introduction / Background

Keyword search fails the moment images have no reliable tags — personal photo libraries, surveillance archives and product catalogues hold millions of untagged pictures. Content-based image retrieval (CBIR) describes each image with numbers computed from its own pixels and ranks a collection by visual similarity to a query image. This project implements the classical, fully transparent CBIR pipeline: a 2×2 spatial RGB histogram plus Sobel edge density forms a 257-dimensional descriptor per image, vectors are L2-normalized and indexed, and queries are answered by chi-squared distance ranking on the Wang (Corel-1K) benchmark.

03. Problem Statement

Students approaching image search usually reach for filename matching or a black-box embedding API, and cannot explain why any result was returned or measure retrieval quality. This project addresses that by building the engine from pixels upward: every histogram bin is inspectable, the distance metric is derived rather than assumed, and quality is measured with Precision@K and mean average precision on a labelled benchmark — with a live demo that exposes the feature vectors themselves.

04. Objectives

- Implement histogram and edge-density feature extraction with OpenCV/NumPy.
- Build the indexing and chi-squared ranking engine with a REST query API.
- Evaluate Precision@K and mAP on the Wang (Corel-1K) benchmark.
- Compare the classical 257-D descriptor against a 2048-D fused CNN variant.
- Build a live retrieval demo with a feature inspector and distance matrix.
- Deliver the complete student kit: engine, harness, demo, report, PPT and viva Q&A.

05. Proposed Methodology

The pipeline has four stages. (1) Features: each image is divided into a 2×2 grid, a 64-bin RGB histogram is computed per cell and concatenated (256-D), Sobel edge density is appended (257-D), and the vector is L2-normalized. (2) Index: all collection vectors are stored in memory with category labels. (3) Retrieval: a query image passes through identical extraction and the collection is ranked by chi-squared distance. (4) Evaluation: every benchmark image is replayed as a query, same-category hits count as relevant, and Precision@K plus mAP are computed; the fused variant concatenates a 2048-D CNN embedding before ranking and the gain is measured.

06. System Architecture / Working

The engine is a Python service (feature extraction, index, ranking) behind a Flask/FastAPI REST endpoint. The single-file web demo lets the user pick any photo as the query, shows the ranked results with similarity scores, exposes the exact 257-D vector in a feature inspector, and displays the pairwise chi-squared distance matrix over the demo collection. A system-report view documents the pipeline stages, the benchmark and the

design-target metrics.

07. Hardware / Software Requirements

- Python 3, NumPy, OpenCV, scikit-learn
- Flask/FastAPI (REST query API)
- Jupyter Notebook (evaluation harness)
- Wang image database (Corel-1K)
- HTML5 + JavaScript (live retrieval demo)
- Any modern laptop; no GPU required

08. Expected Outcomes

- A working CBIR engine with design-target Precision@10 ≥ 0.72 and mAP ≥ 0.65.
- A measured classical-vs-fused comparison on Corel-1K.
- A live query-by-example demo with feature inspector.
- A complete report, PPT and viva Q&A on retrieval theory and evaluation.

09. Applications

- Teaching feature engineering and distance metrics.
- Demonstrating information-retrieval evaluation (Precision@K, mAP).
- Prototype visual search for photo libraries and catalogues.
- Baseline for deep-learning retrieval comparisons.

10. Future Scope

- Large-scale indexing with approximate nearest neighbours.
- Relevance feedback to refine queries interactively.
- Fusion with text embeddings for hybrid search.
- GPU-accelerated extraction for million-image collections.