CBIR-ResNet-ConvNeXt-Tiny

Department of Computer Science
Texas State University

Kevin Morales
eps60@txstate.edu

Abstract

This project presents a Content-Based Image Retrieval (CBIR) system capable of retrieving visually similar images from a large dataset based solely on image content. Using the Caltech-256 dataset, the system extracts four types of visual features: classical descriptors (HSV color histograms and Local Binary Patterns) and deep visual embeddings generated from pretrained ResNet-50 and ConvNeXt-Tiny convolutional neural networks. A cosine similarity–based ranking method is used to retrieve the top-K most similar images to a given query. The system is evaluated quantitatively using Precision@5 and Precision@10 across 200 random queries, and qualitatively through visual success and failure cases, including tests on external images not present in the dataset. Results show that deep features significantly outperform classical handcrafted descriptors, with ConvNeXt-Tiny providing the strongest overall retrieval performance. The project demonstrates how modern CNN embeddings enable robust and scalable CBIR systems and highlights common failure modes and opportunities for improvement.

1 Introduction

Content-Based Image Retrieval (CBIR) refers to the task of finding images in a database that are visually similar to a given query image. Unlike traditional search methods that rely on filenames, tags, or textual metadata, CBIR systems analyze the visual content of images directly, allowing retrieval based on color, texture, shape, or high-level semantic features. As digital image collections continue to grow, efficient and accurate visual search has become increasingly important for applications such as online shopping, medical imaging, surveillance, digital asset management, and multimedia search engines.

Despite its usefulness, CBIR remains a challenging problem. Images of the same object category can vary widely in lighting, pose, background clutter, scale, and viewpoint. Conversely, images from different categories may share similar colors or textures, making them difficult to distinguish based on simple descriptors. These challenges have motivated the development of feature extraction techniques ranging from classical handcrafted representations to modern deep learning–based embeddings.

The goal of this project is to design and evaluate a CBIR system that incorporates both classical and deep visual features. Classical descriptors, such as HSV color histograms and Local Binary Patterns (LBP), provide baseline representations based on low-level image properties. Deep neural networks, including ResNet-50 and ConvNeXt-Tiny, generate high-dimensional embeddings that capture more complex and abstract visual information. By comparing these feature types on the Caltech-256 dataset, the project aims to explore their strengths, limitations, and retrieval performance.

In addition to quantitative evaluation using Precision@K, the system is tested qualitatively through visualizations of retrieval results and failure cases. External query images not included in the dataset are also examined to assess generalization to real-world inputs. Overall, this project demonstrates the advantages of deep feature embeddings for CBIR and highlights important considerations for building effective visual search systems.

2 Methodology

The proposed Content-Based Image Retrieval (CBIR) system retrieves visually similar images from a large database based exclusively on image content. Given a query image, the system extracts a visual feature representation, compares it against a database of precomputed feature vectors, and ranks all images using a similarity metric. The top-K most similar images are then returned as the retrieval result. This pipeline allows efficient and scalable visual search without relying on textual metadata or manual annotations during retrieval.

The overall workflow of the system consists of the following stages: image preprocessing, feature extraction, feature storage, similarity computation, and result ranking. Classical handcrafted features and deep convolutional neural network (CNN) features are both implemented and compared to analyze their retrieval performance.

2.1 Dataset

All experiments are conducted using the Caltech-256 dataset, which contains 30,607 images distributed across 256 object categories. Each category contains at least 80 images and represents a wide variety of everyday objects, including animals, vehicles, household items, tools, food, and musical instruments. The dataset exhibits substantial variation in background clutter, lighting conditions, viewing angles, and object scale, making it well suited for evaluating CBIR systems under realistic conditions.

For testing generalization beyond the dataset, a small collection of external query images containing objects not explicitly represented in Caltech-256 was also used. These images were processed using the same feature extraction and retrieval pipeline as the dataset images.

2.2 Image Preprocessing

All images are resized to 224 × 224 pixels to ensure compatibility with deep neural network backbones. Pixel intensities are normalized using the standard mean and standard deviation values from the ImageNet dataset. For deep feature extraction, images are converted into PyTorch tensors and processed in batches. For classical feature extraction, images are converted to appropriate color spaces or grayscale formats as required.

2.3 Classical Feature Extraction

Two handcrafted feature descriptors are used as classical baselines: color histograms and Local Binary Patterns (LBP).

2.3.1 Color Histograms

Color features are extracted using HSV color histograms to capture global color distribution while maintaining robustness to lighting variations. Each image is converted from RGB to HSV color space. Histograms are computed with fixed binning across each channel and normalized to ensure scale invariance. The resulting histogram produces a **96-dimensional feature vector** for each image, representing its global color composition.

2.3.2 Local Binary Patterns

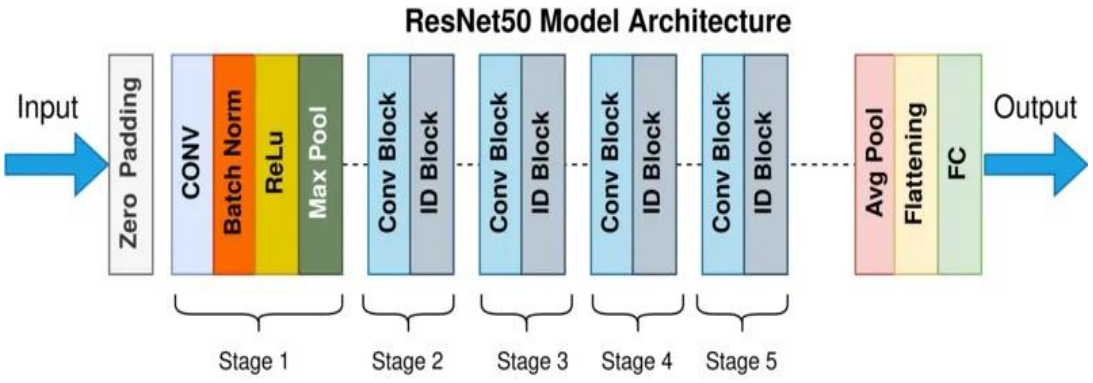
Texture features are extracted using the Local Binary Pattern (LBP) operator. Each grayscale image is processed with a neighborhood of 24 sampling points and a radius of 3 pixels. Uniform LBP encoding is used to reduce the number of possible patterns while preserving texture information. A normalized histogram of LBP codes is then computed, yielding a 26-dimensional feature vector that characterizes the local texture patterns of the image.

2.4 Deep Feature Extraction

In addition to classical features, deep feature embeddings are extracted using two pretrained convolutional neural network architectures: ResNet-50 and ConvNeXt-Tiny. Both models are pretrained on the ImageNet dataset and used strictly as fixed feature extractors.

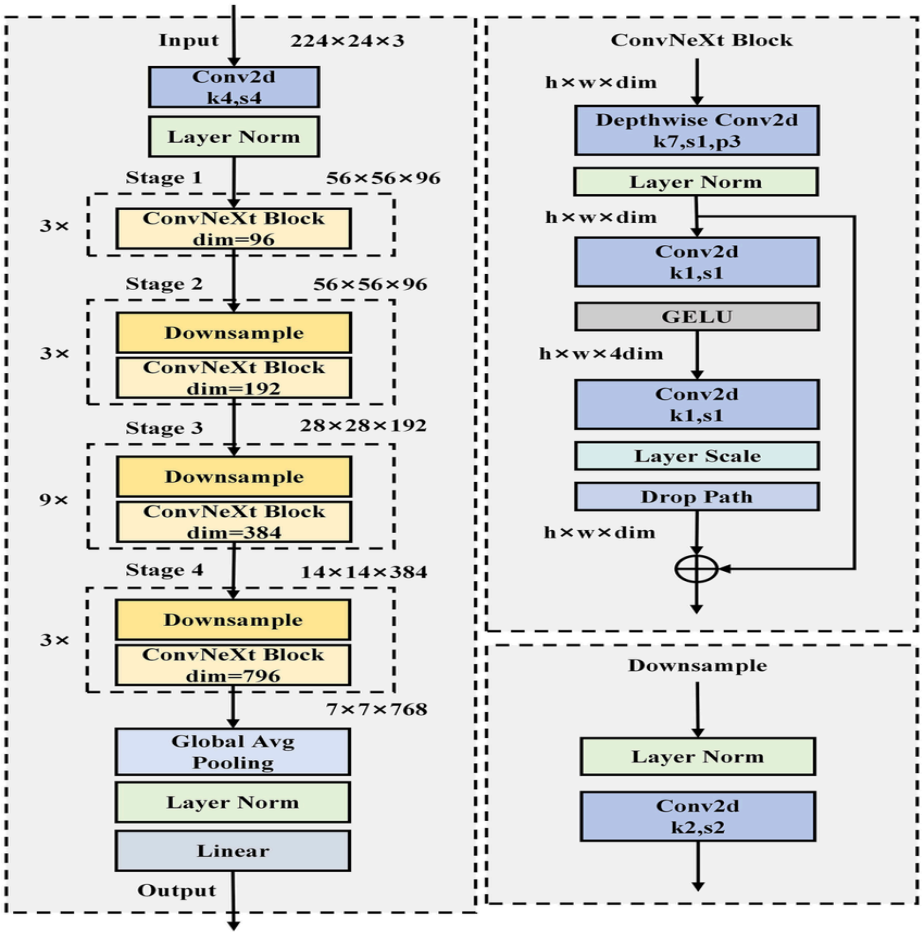
2.4.1 ResNet-50 Architecture

ResNet-50 is a deep residual network consisting of 50 layers with skip connections that allow efficient training of deep models. For feature extraction, the final fully connected classification layer is removed, and the output of the final convolutional block is used as the image embedding. This produces a 2048-dimensional feature vector per image that encodes high-level semantic information.



2.4.2 ConvNext-Tiny Architecture

ConvNeXt-Tiny is a modern convolutional neural network architecture designed using principles inspired by transformer models while retaining the efficiency of CNNs. Similar to ResNet-50, the final classification layer is removed to obtain deep visual embeddings. Each image is represented by a 768-dimensional feature vector. ConvNeXt-Tiny serves as a lightweight yet powerful modern backbone for image retrieval.



2.5 Feature Storage

To enable efficient retrieval and large-scale evaluation, all extracted features are stored using NumPy's binary array (.npy) format. Separate feature matrices are maintained for each feature type, including color histograms, LBP, ResNet-50 embeddings, and ConvNeXt-Tiny embeddings. Each matrix has dimensions N×DN \times DN×D, where N=30,607N = 30,607N=30,607 is the number of dataset images and DDD is the corresponding feature dimensionality.

Two additional arrays store the file paths and class labels of each image. This organization allows direct mapping between feature vectors, original images, and ground truth categories during retrieval and evaluation.

2.6 Evaluation Methodology

Retrieval performance is quantitatively evaluated using Precision@5 and Precision@10, which measure the proportion of correct matches in the top 5 and top 10 retrieved images, respectively. A retrieved image is considered correct if it belongs to the same category as the query image.

To obtain statistically reliable results, 200 random query images are sampled from the dataset. For each query, retrieval is performed using each feature representation, and Precision@5 and Precision@10 are computed. The final reported values represent the average performance across all 200 queries.

Qualitative evaluation is conducted using visual examples of successful retrievals, failure cases, and external query tests to further analyze system behavior and generalization performance.

3 Results

This section presents the experimental outcomes of the CBIR system using classical and deep feature extraction methods. Quantitative results are shown using Precision@5 and Precision@10 over 200 random query images, and qualitative results are demonstrated using retrieval figures for both successful and failure cases, including external query tests.

3.1 Quantitative Results

Table 1 summarizes the quantitative retrieval performance of all feature extraction methods. Deep features significantly outperform classical features, with ConvNeXt-Tiny achieving the highest Precision@5 and Precision@10 values.

Table:1 retrieval performance across feature extraction methods

Feature Type	Precision@5k	Precision@10
Color Histogram	0.0420	0.0375

LBP	0.0470	0.0415
ResNet-50	0.7590	0.7410
ConvNext-Tiny	0.8000	0.7825

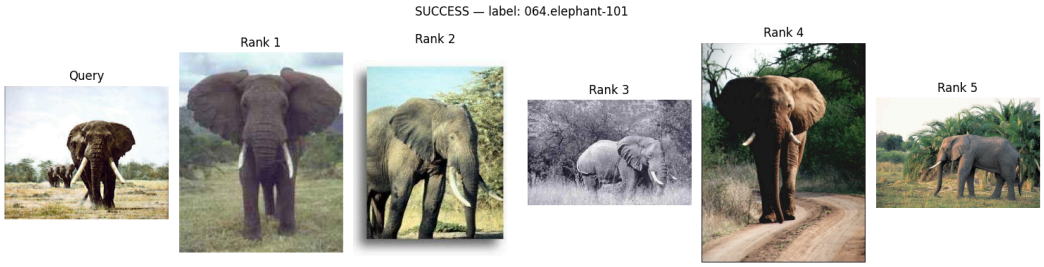
3.2 Qualitative Results

Qualitative retrieval results provide further insight into the strengths and limitations of the system. Figures 1–7 show examples of successful retrievals, failure cases, and external tests using real-world images not included in the Caltech-256 dataset.

3.2.1 Successful Retrieval Examples

Figure 1. Successful ResNet-50 retrieval

Figure 2. Successful ConvNext-Tiny retrieval



3.2.2 Failure Retrieval Examples

Figure 3. Failure case for ResNet-50

Figure 4. Failure case for ConvNext-Tiny

3.2.3 External Query Retrieval

External queries evaluate how well the system retrieves semantically similar results for images not included in the dataset. These examples include a clear object case, a cluttered/failure case, and an unseen-class example (microphone) using ConvNext-Tiny.

Figure 5. External query success example retrieved

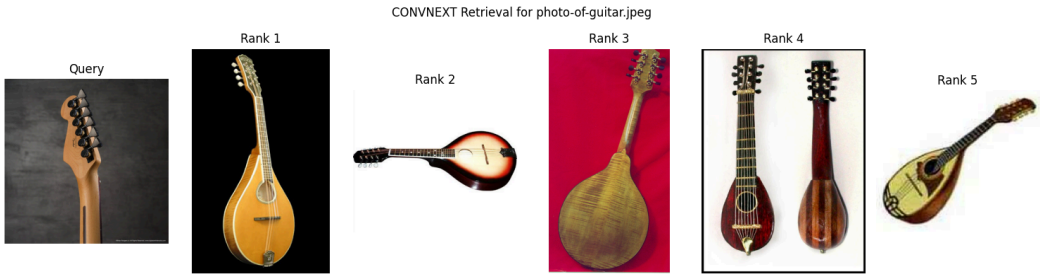


Figure 6. External query failure example

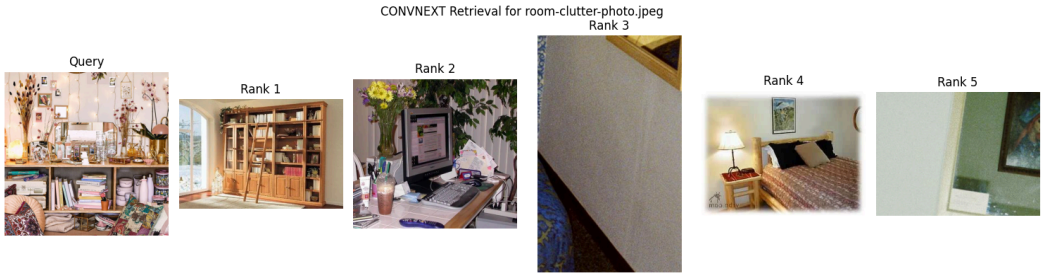


Figure 7. external query of an unseen object category (microphone)

4 Discussion

The results of this project show a clear distinction between classical and deep feature extraction methods for image retrieval. Classical descriptors such as color histograms and Local Binary Patterns performed poorly, as they capture only limited low-level information. In contrast, both ResNet-50 and ConvNeXt-Tiny achieved high retrieval accuracy, demonstrating the effectiveness of deep, learned representations for modeling visual similarity.

ConvNeXt-Tiny performed the best overall, which aligns with modern architectural trends favoring improved feature hierarchies and stronger texture–shape representations. The qualitative results show that both deep models retrieve correct matches when objects are clear, centered, and visually distinct. Failure cases typically occurred in images with cluttered backgrounds, small objects, or ambiguous class boundaries, which caused the models to prioritize visually similar patterns over the correct class.

External query evaluations added important insights. Clean external images often produced reasonable matches, indicating good generalization beyond the Caltech-256 dataset. Cluttered scenes, however, revealed limitations and highlighted the system's sensitivity to background dominance. The unseen-class query (microphone) was particularly informative: with no matching category available, the system retrieved objects with related shapes or textures, showing a form of semantic approximation rather than random retrieval.

5 Conclusion

This project implemented a complete CBIR system using both classical and deep feature extraction methods. The results demonstrate that deep models, especially ConvNeXt-Tiny, provide significantly stronger retrieval performance than handcrafted features. The system successfully retrieves semantically similar images in many cases and generalizes reasonably well to external queries.

External experiments also revealed the system's limitations, particularly with cluttered scenes and unseen object categories. Nevertheless, the retrieval behavior remained meaningful, often returning visually or semantically similar images even when an exact match was not available.

In conclusion, the project shows that modern CNN architectures are highly effective for content-based image retrieval. Future work may focus on fine-tuning, expanding the dataset, improving retrieval efficiency, or building a graphical interface to make the system more user-friendly and adaptable to real-world applications.