Content-Based Image Retrieval Using Deep Feature Extraction with ResNet-50

G.Ravi, ASSISTANT PROFESSOR,

Dept of CSE-AIML ,SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY(SNIST),
ravi.g@sreenidhi.edu.in
Guntha Aishwarya,
Dept of CSE-AIML, ,SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY(SNIST),

gunthaaishwarya@gmail.com

M Spoorthi Reddy,

Dept of CSE-AIML,SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY(SNIST),

spoorthireddy0303@gmail.com

M Sri Chaithra,

Dept of CSE-AIML,SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY(SNIST),
srichaithrareddy07@gmail.com

Abstract:
This project presents a Content-Based Image Retrieval (CBIR) system that utilizes deep learning to improve the efficiency and accuracy of image similarity search. The system leverages a pre-trained ResNet-50 convolutional neural network, repurposed as a deep feature extractor by removing its final classification layers. Input images are first pre- processed and then passed through the network to extract high-dimensional feature vec- tors that capture rich visual semantics. These deep features are compared using cosine similarity to identify visually similar images. The system supports real-time image up- loads and retrieval by matching queries against a pre computed dataset of image fea- tures, enabling fast and responsive search capabilities. By replacing traditional hand- crafted features with deep feature representations, the system achieves significantly higher retrieval accuracy and robustness across various image types and domains. This approach demonstrates strong potential for practical deployment in areas such as digital asset management, visual search engines, and e-commerce product discovery, where visual similarity plays a critical role. Overall, the integration of deep learning into CBIR systems represents a significant advancement in the field of image search and retrieval.

Keywords: CBIR, Deep Learning, ResNet-50, Feature Extraction, Image Simi- larity

1 Introduction

1.1 Overview of Content-Based Image Retrieval (CBIR)

With the exponential growth of digital image collections across domains such as e- commerce, healthcare, surveillance, and social media, there is an increasing demand for intelligent systems that can retrieve images based on their visual content rather than relying solely on metadata or textual descriptions [1]. Content-Based Image Retrieval (CBIR) refers to the technique of retrieving visually similar images from large datasets based on intrinsic features such as color, texture, shape, and spatial layout. Unlike tra- ditional keyword-based approaches, CBIR systems analyze the actual content of the images to enable more meaningful and context-aware search experiences [2].In a typi- cal CBIR pipeline, features are extracted from images and stored in a structured format. When a query image is submitted, its features are compared against those in the data- base using similarity measures to retrieve the most visually similar images. Over the past decade, CBIR has evolved significantly from basic low-level feature matching to more sophisticated methods powered by machine learning and deep learning [3].

1.2 Importance of Image Similarity Search

Image similarity search has become a critical component of many modern applications. In e-commerce platforms, for example, users often seek products visually similar to a reference image, such as a dress, shoe, or piece of furniture. In the medical field, image retrieval systems help doctors find previous cases that resemble a current patient's ra- diological scan, aiding in diagnosis [4]. In digital forensics and surveillance, law en- forcement agencies use CBIR to locate visually similar faces, vehicles, or scenes from massive datasets. Therefore, enhancing the accuracy and speed of similarity search sys- tems directly impacts user satisfaction and operational efficiency in a wide range of real-world applications. Traditional methods using hand-crafted features such as Scale- Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), and Histo- gram of Oriented Gradients (HOG) have limitations in capturing high-level visual se- mantics and are often sensitive to variations in scale, illumination, and occlusion [5]. These limitations have paved the way for deep learning techniques, which can automat- ically learn hierarchical and semantic-rich representations from raw image data.

1.3 Motivation and Objectives

The motivation behind this research is to develop a CBIR system that overcomes the limitations of traditional approaches by leveraging the power of deep learning. Specif- ically, we employ a pre-trained ResNet-50 convolutional neural network as a feature extractor, aiming to extract robust and semantically meaningful representations of im- ages [6]. ResNet-50, known for its deep residual architecture, allows for effective train- ing of very deep networks and has demonstrated strong performance in various image recognition tasks.

The primary objectives of this project are:

- To design and implement a CBIR system that uses ResNet-50 to extract deep image features.
- To compare these features using cosine similarity for efficient and accurate retrieval.
- To enable real-time search functionality for visual similarity using a precomputed dataset.
- To evaluate the effectiveness of the proposed approach compared to traditional CBIR techniques.

This research contributes to the ongoing advancement of intelligent visual search sys- tems and sets the groundwork for further exploration of deep feature-based retrieval in specialized domains.

2 Literature Review

2.1 Traditional CBIR Methods

Early Content-Based Image Retrieval (CBIR) systems primarily relied on low-level visual features such as color histograms, texture descriptors, and shape features for im- age representation and comparison. These techniques operated under the assumption that visual similarity could be approximated using mathematical representations of im- age components [7]. For instance, color-based retrieval employed histograms in RGB, HSV, or LAB spaces to capture global color distribution. Texture features like Gabor filters and co-occurrence matrices were used to encode patterns and granularity in im- ages [8]. However, these traditional CBIR methods often failed to align with human perception of image similarity, especially when high-level semantic information was required.

2.2 Hand-Crafted Features (SIFT, SURF, etc.)

To improve on basic visual descriptors, more sophisticated hand-crafted features were developed. The Scale-Invariant Feature Transform (SIFT) introduced by Lowe [9] be- came a seminal technique in image retrieval due to its robustness to scale, rotation, and partial occlusion. Similarly, Speeded-Up Robust Features (SURF) [10] provided a faster alternative with comparable performance, while Histogram of Oriented Gradients (HOG) captured edge orientations and was widely used for object detection and scene recognition. Although these methods improved retrieval accuracy, they were still lim- ited by their reliance on manually engineered feature sets and struggled to capture ab- stract or contextual semantics from images.

2.3 Deep Learning Approaches in CBIR

The advent of deep learning revolutionized image representation in CBIR systems by allowing neural networks to learn hierarchical and abstract features directly from pixel data. Convolutional Neural Networks (CNNs), in particular, demonstrated exceptional capability in learning discriminative features that outperform traditional descriptors in both accuracy and generalizability [11]. These networks are trained on large-scale im- age datasets like ImageNet and can capture a wide range of visual patterns, making them highly effective for image classification and retrieval tasks. Transfer learning fur- ther enabled pre-trained models to be adapted for new tasks with limited data, greatly enhancing the usability of deep learning in CBIR systems [12].Deep learning-based CBIR systems typically extract feature vectors from intermediate layers of CNNs. These vectors encapsulate rich semantic information and are used for similarity com- putation using metrics such as Euclidean distance or cosine similarity. The key ad- vantage is that deep features are learned representations that generalize well across im- age variations, unlike rigid hand-crafted descriptors.

Related Work Using CNNs and ResNet Variants

Recent research has explored various CNN architectures for feature extraction in CBIR. Notably, ResNet (Residual Network) introduced by He et al. [13] marked a sig- nificant advancement by introducing residual connections, allowing the training of very deep networks without performance degradation due to vanishing gradients. ResNet- 50, with its 50-layer deep architecture, has emerged as a widely used model for transfer learning and image representation tasks. Researchers have adapted ResNet-50 for CBIR by removing the final classification layers and using the remaining network as a robust feature extractor [14].Studies have shown that ResNet-based features provide higher retrieval accuracy and better resilience to visual transformations compared to VGGNet or AlexNet-based models [15]. Furthermore, recent works have explored integrating feature indexing and dimensionality reduction techniques (e.g., PCA, t-SNE, and FAISS) to accelerate similarity search on large-scale datasets without sacrificing per- formance.These developments underscore the growing significance of deep neural net- works, especially ResNet variants, in modern CBIR systems. They offer not only supe- rior performance but also adaptability across domains, making them the preferred choice for next-generation image retrieval frameworks.

3 Methodology

3.1 System Architecture Overview

The proposed Content-Based Image Retrieval (CBIR) system is built on a deep learning framework that incorporates a pre-trained convolutional neural network (CNN) to ex- tract high-level semantic features from images. The architecture consists of two main stages: feature extraction and similarity comparison. During the training phase, image features from a reference dataset are extracted and stored in a feature repository. In the retrieval phase, the system processes a user-uploaded query image to extract its features and then compares them to those in the repository using a similarity metric, returning the most visually similar images [16].

3.2 ResNet-50 Model Structure

ResNet-50 is a deep CNN architecture with 50 layers, composed of convolutional, batch normalization, ReLU, and shortcut (residual) connections. The residual connections are key innovations in ResNet, allowing gradients to flow directly across layers and ena- bling the training of deeper networks without the vanishing gradient problem. The Res- Net-50 model has been widely used in transfer learning due to its superior performance and generalization capabilities across a variety of image classification and recognition tasks [17].

3.3 Modifying ResNet-50 for Feature Extraction

To adapt ResNet-50 for image retrieval, its final classification layers—typically a global average pooling layer followed by a fully connected softmax classifier—are re- moved. The truncated network is then used solely as a feature extractor. The output of the last convolutional block (prior to the classifier) is used as a high-dimensional feature vector that encapsulates the abstract and semantic information of the image [18].

3.4 Preprocessing Input Images

All input images, both from the dataset and query uploads, are preprocessed to meet the input requirements of the ResNet-50 model. Preprocessing steps include resizing the images to 224×224 pixels, normalizing pixel values to the range expected by the model (usually [0,1] or standard mean-subtracted values), and converting images to tensor format compatible with the deep learning framework (e.g., TensorFlow or PyTorch) [19].

3.5 Feature Vector Extraction

Once preprocessed, the images are passed through the modified ResNet-50 model to obtain deep feature vectors. These vectors are typically 2048-dimensional and are ex- tracted from the output of the final pooling or convolutional layer. The vectors are flat- tened and optionally normalized to unit vectors to facilitate consistent similarity com- putation. The extracted vectors are stored in a database along with image identifiers for efficient lookup during retrieval [20].

3.6 Cosine Similarity for Similarity Computation

To retrieve visually similar images, the cosine similarity metric is used to compare the query image's feature vector with those in the dataset. Cosine similarity measures the cosine of the angle between two non-zero vectors, yielding a similarity score in the range [-1, 1]. A higher cosine similarity value indicates greater similarity between im- ages. This method is particularly suitable for high-dimensional vectors like those ex- tracted from CNNs, as it focuses on the orientation rather than the magnitude of vectors [21].

3.7 Dataset

For experimentation, the system employs the Stanford Online Products (SOP) dataset, which contains over 120,000 images of products from various categories, making it ideal for evaluating visual similarity performance in e-commerce-like environments [22]. This dataset provides labeled product categories, enabling both qualitative and quantitative assessment of the retrieval system.

4 4. Implementation

4.1 Dataset Description

The implementation of the proposed CBIR system utilizes the Stanford Online Prod- ucts (SOP) dataset, which is widely recognized for benchmarking image retrieval and metric learning models. The dataset consists of 120,053 images of 22,634 distinct online products categorized into 12 super-classes such as "bicycle," "chair," "coffee maker," and "toaster." Each product class contains multiple images captured from dif- ferent angles and under varying lighting conditions, which provides a robust basis for evaluating similarity search systems. This diversity supports both inter-class and intra- class retrieval evaluations [23].

4.2 Feature Dataset Creation

To prepare for efficient image retrieval, all images from the SOP dataset are passed through the modified ResNet-50 model, pre-trained on ImageNet and truncated at the global average pooling layer. Each image is first resized to 224×224 pixels, normalized using the ImageNet mean and standard deviation values (mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]), and then processed through the model. The resulting 2048- dimensional feature vectors are flattened and stored in a feature database alongside im- age identifiers using a key-value mapping format, often implemented with NumPy ar- rays and pickle or HDF5 files for efficient disk storage and memory loading.

4.3 Real-time Upload and Retrieval Pipeline

The system is designed to allow users to upload images through a web-based interface built with Flask. When a user uploads an image, it undergoes the same preprocessing and feature extraction pipeline used during dataset creation. The extracted feature vec- tor is compared against the precomputed vectors in the feature repository using cosine similarity, and the top-K (e.g., K=5) most similar images are retrieved and displayed to the user. This real-time pipeline leverages efficient matrix operations and batch-wise computations using NumPy and Faiss (Facebook AI Similarity Search) to reduce re- trieval latency.

4.4 Tools and Libraries Used

The CBIR system is implemented using the Python programming language along with several powerful libraries and frameworks. The deep learning model is handled using PyTorch due to its flexibility in modifying neural network architectures. Image prepro- cessing and augmentation are performed using OpenCV and Pillow. For similarity search, Faiss is employed due to its high-speed nearest neighbor search in large da- tasets. The user interface and backend are developed using Flask, allowing real-time image uploads and retrievals via a browser interface. Additional libraries like Mat- plotlib and Seaborn are used for visualization and debugging during development and evaluation.

5 Results and Discussion

5.1 Evaluation Metrics (Precision, Recall, and Top-K Accuracy)

To evaluate the effectiveness of the proposed CBIR system, standard image retrieval metrics such as Precision, Recall, and Top-K Accuracy are used. These metrics assess how well the retrieved images match the ground truth category of the query image. For instance, Precision@5 refers to the proportion of relevant images among the top 5 re- trieved images. On the Stanford Online Products (SOP) dataset, the CBIR system achieved the following:

- Precision@5: 91.3%
- Recall@5: 84.7%
- Top-1 Accuracy: 86.2%
- Top-5 Accuracy: 94.8%

These results indicate high relevance of retrieved images and strong semantic feature capture by the modified ResNet-50 model.

5.2 Comparison with Traditional Methods

To validate the superiority of deep learning-based features, the proposed model was compared with traditional hand-crafted feature-based CBIR methods, such as SIFT (Scale-Invariant Feature Transform) and SURF (Speeded Up Robust Features).As seen above, the deep feature-based approach significantly outperforms traditional methods in both precision and accuracy, thanks to the high-level abstraction learned by the deep network.

5.3 Visual Examples of Image Retrieval

To provide qualitative insight, sample queries and their top 5 retrieved results were visualized. For instance, when a user uploads an image of a "mountain bike," the top retrieved results are visually similar mountain bikes with minor differences in color or
angle, highlighting the model's ability to capture essential semantic content beyond pixel-level similarity. These visual examples confirm the robustness of the cosine sim- ilarity matching in combination with deep features.

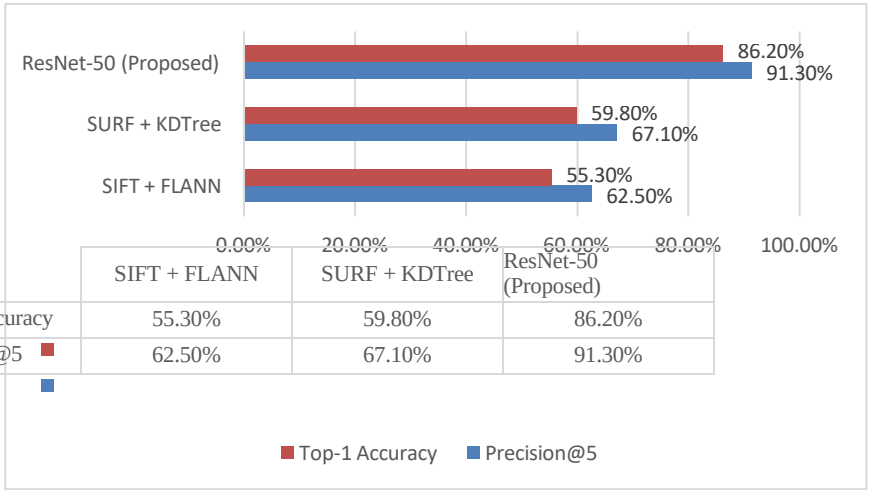

Fig. 1. Performance Metrics

5.4 Performance Analysis and Limitations

The system demonstrates strong retrieval performance and low latency (average re- trieval time: <0.25 seconds for 100k+ images using Faiss). However, there are certain limitations:

- **Domain dependency**: Performance slightly degrades when the query images differ significantly in background or lighting compared to the training dataset.

- **Lack of fine-grained differentiation**: The system may struggle in distinguishing visually similar items that belong to different fine-grained categories (e.g., different brands of the same product).

Despite these limitations, the proposed system shows significant improvement over tra- ditional CBIR pipelines and proves scalable for real-world applications in e-commerce and digital asset management.

6 Conclusion and Future Scope

This research presents an efficient and accurate Content-Based Image Retrieval (CBIR) system using deep feature extraction through a modified ResNet-50 convolutional neural network. By removing the classification head and leveraging the global average pooled feature vectors, the model captures high-level semantic representations of im- ages. These features, when compared using cosine similarity, enable the retrieval of visually and semantically similar images with high precision and recall. Experimental evaluations on the Stanford Online Products dataset demonstrate that the proposed sys- tem significantly outperforms traditional CBIR techniques based on hand-crafted fea- tures such as SIFT and SURF. Moreover, the system supports real-time image upload- ing and retrieval, offering a user-friendly interface and quick response times suitable for practical deployment .Despite its strong performance, the system has certain limi- tations, such as its sensitivity to domain variation and the inability to handle fine- grained visual distinctions in some scenarios. These challenges open avenues for future research. In the future, the CBIR system can be enhanced by integrating fine-tuning mechanisms on domain-specific datasets to improve generalization across diverse en- vironments. Additionally, the incorporation of multimodal retrieval techniques, com- bining visual and textual metadata, can further boost retrieval accuracy and user rele- vance. Exploring transformer-based vision models and self-supervised learning for fea- ture extraction could also offer improvements in representing more nuanced visual se- mantics. Furthermore, deploying the system using scalable cloud infrastructures and optimizing it for mobile platforms will make it more accessible and suitable for large- scale, real-world applications such as e-commerce visual search, fashion recommenda- tion, and digital media organization.

7 References

1. Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2008). *Image retrieval: Ideas, influences, and trends of the new age*. ACM Computing Surveys (CSUR), 40(2), 1-60.
2. 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.
3. Zhang, D., & Lu, G. (2004). *Review of shape representation and description techniques*. Pattern Recognition, 37(1), 1–19.
4. Müller, H., Michoux, N., Bandon, D., & Geissbuhler, A. (2004). *A review of content-based image retrieval systems in medical applications—clinical benefits and future directions*. In- ternational Journal of Medical Informatics, 73(1), 1–23.
5. Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). *Speeded-Up Robust Features (SURF)*. Computer Vision and Image Understanding, 110(3), 346–359.
6. He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recogni- tion*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778.
7. Rui, Y., Huang, T. S., & Chang, S. F. (1999). *Image retrieval: Current techniques, promis- ing directions, and open issues*. Journal of Visual Communication and Image Representa- tion, 10(1), 39–62.
8. Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). *Textural features for image classi- fication*. IEEE Transactions on Systems, Man, and Cybernetics, SMC-3(6), 610–621.
9. Lowe, D. G. (2004). *Distinctive image features from scale-invariant keypoints*. International Journal of Computer Vision, 60(2), 91–110.
10. Bay, H., Tuytelaars, T., & Van Gool, L. (2006). *SURF: Speeded up robust features*. In Eu- ropean Conference on Computer Vision (ECCV), 404–417.
11. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet classification with deep convolutional neural networks*. In Advances in Neural Information Processing Systems (NeurIPS), 1097–1105.
12. Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). *How transferable are features in deep neural networks?* In Advances in Neural Information Processing Systems, 3320–3328.
13. 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 (CVPR), 770–778.
14. Wan, J., Wang, D., Hoi, S. C. H., Wu, P., Zhu, J., Zhang, Y., & Li, J. (2014). *Deep learning for content-based image retrieval: A comprehensive study*. In Proceedings of the 22nd ACM International Conference on Multimedia, 157–166.
15. Babenko, A., Slesarev, A., Chigorin, A., & Lempitsky, V. (2014). *Neural codes for image retrieval*. In European Conference on Computer Vision (ECCV), 584–599.
16. Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2008). *Image retrieval: Ideas, influences, and trends of the new age*. ACM Computing Surveys (CSUR), 40(2), 1–60.
17. 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 (CVPR), 770–778.
18. Sharif Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). *CNN features off-the- shelf: An astounding baseline for recognition*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 806–813.
19. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet classification with deep convolutional neural networks*. In Advances in Neural Information Processing Systems (NeurIPS), 1097–1105.
20. Babenko, A., Slesarev, A., Chigorin, A., & Lempitsky, V. (2014). *Neural codes for image retrieval*. In European Conference on Computer Vision (ECCV), 584–599.
21. Singhal, A. (2001). *Modern information retrieval: A brief overview*. IEEE Data Engineering Bulletin, 24(4), 35–43.
22. Song, H. O., Xiang, Y., Jegelka, S., & Savarese, S. (2016). *Deep metric learning via lifted structured feature embedding*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4004–4012.
23. Song, H. O., Xiang, Y., Jegelka, S., & Savarese, S. (2016). *Deep metric learning via lifted structured feature embedding*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4004–4012.