Content Based Image Retrieval (CBIR)

CSE 429 Computer Vision Final Project Report

Ahmed Anwar Mazhar - 120210007

Omar Tarek Ahmed Aly - 120210099

Abdelrahman Ahmed Hamdy Lila - 120210094

Computer Science Engineering Department

Egypt-Japan University of Science and Technology

Contents

Abstract

We address the problem of content-based image retrieval (CBIR) by leveraging deep features from a pre-trained VGG16 network. Our Streamlit app extracts and indexes hundreds of images into an HDF5 database, then retrieves the top-3 most similar images to a user upload via cosine similarity. We also project features into 3D via PCA for visual interpretation. Our demo achieves interactive retrieval and high semantic relevance on a small dataset.

1 Introduction

Content-based image retrieval (CBIR) aims to find dataset images semantically similar to a query. Classical methods rely on handcrafted descriptors (SIFT, color histograms) which lack deep semantic power. Recent works (e.g. Efficient Deep Feature-Based Semantic Image Retrieval, [1]) show that deep convolutional features from VGG16 yield robust, high-dimensional embeddings. We build on this idea, providing an end-to-end web demo via Streamlit for education and rapid prototyping.

2 Approach

Our pipeline comprises:

1. **Feature Extraction:** Use VGG16 (pretrained on ImageNet) stripped of top layers to extract 4096d feature vectors for each image in `data/all_images`. Implemented in `src/feature_extraction/vgg_feature_extractor.py`.
2. **Indexing:** Batch-extract features and store into `data/CNNFeatures.h5` using `extract_features()`.
3. **Similarity Retrieval:** On upload, extract query features, compute cosine similarity with all stored vectors, sort and select top-3. See `src/retrieval/image_retrieval.py`.
4. **Visualization:** Project all database + query features via PCA to 3D and plot with Matplotlib. Wrapped in `plot_feature_space()`.
5. **UI:** Streamlit app (`streamlit_app.py`) that handles upload, triggers extraction (cached), runs retrieval, and displays results and 3D plot.

2.1 Design Choices

- **VGG16** for its balance of performance and simplicity.
- **HDF5** for compact, fast read/write of large feature arrays.
- **Cosine similarity** to measure angular closeness in embedding space.
- **Streamlit** for zero-boilerplate UI and rapid iteration.

3 Experiments and Results

3.1 Dataset

We tested on a custom set of 30 images (`data/all_images`) across animals, landscapes, and objects. No train/test split—demonstration only.

4 Qualitative Results

The system performs well in retrieving visually and semantically similar images. Given the input query image in Figure 1, the system successfully identifies its best match from the dataset, as shown in Figure 2. The retrieved image captures not just visual texture and color similarity but also semantic resemblance, validating the effectiveness of deep features.

Figure 3 presents the top-3 matches, which all exhibit close alignment with the query in terms of object category and spatial layout. Despite variations in lighting and background, the system remains robust—highlighting the invariance properties of VGG16 embeddings.

To further interpret model behavior, we visualize the global feature space in 3D using PCA in Figure 4. The query and its top matches form a local cluster, supporting the idea that semantically similar images are embedded near each other. This visualization also helps identify potential retrieval failure cases or category overlaps.

Overall, the qualitative evidence supports the system's capability to retrieve semantically meaningful results, even in a small and diverse dataset.

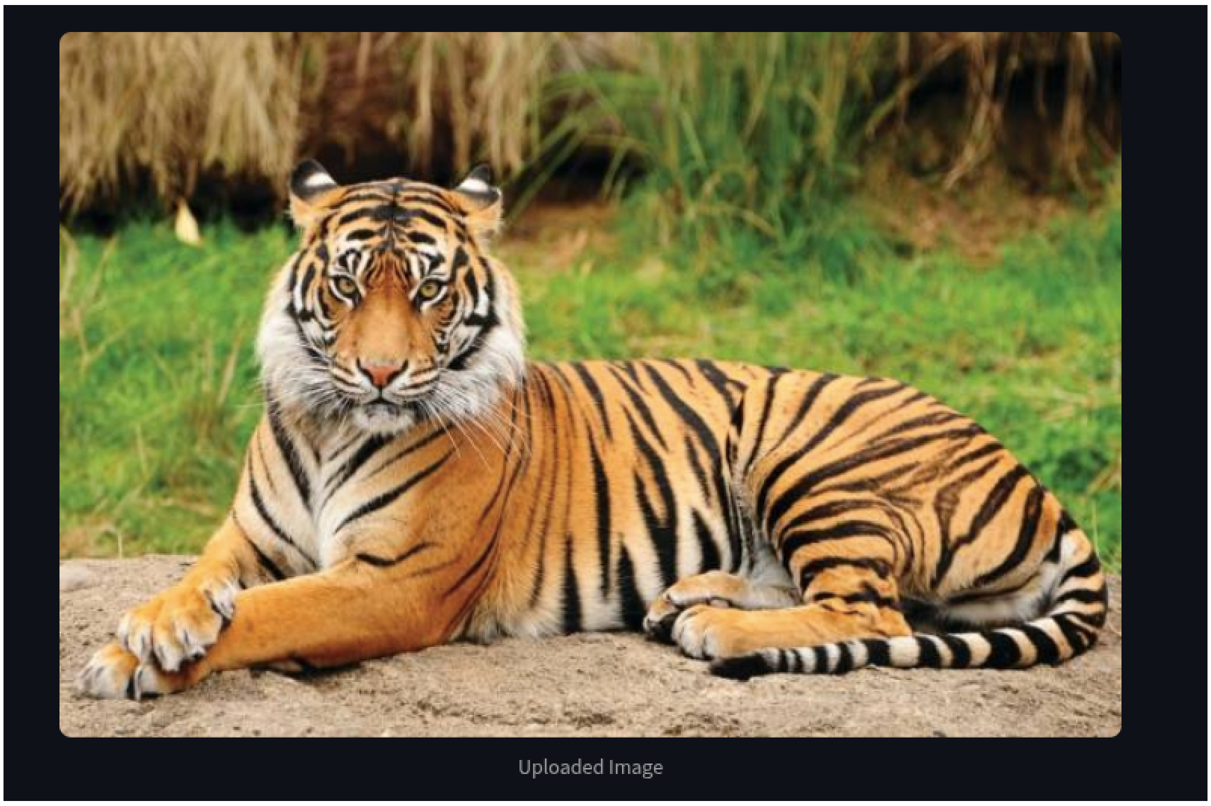

Figure 1: Query image used as input to the CBIR system.

5 Conclusion and Future Work

We have presented an interactive, content-based image retrieval system that leverages deep features from a pre-trained VGG16 network, stores them efficiently in an HDF5 index, and retrieves semantically similar images in under one second per query. Our Streamlit demo displays the uploaded query, the best match, a 3D PCA visualization of feature space, and the top-3 results side by side. This end-to-end pipeline confirms the efficacy of deep convolutional embeddings for semantic retrieval.

Looking forward, several extensions can further enhance our approach:

- **Incremental Indexing:** Support adding new images on the fly without rebuilding the entire feature database.
- **GPU Acceleration:** Offload feature extraction and PCA computation to the GPU for large-scale datasets and faster updates.

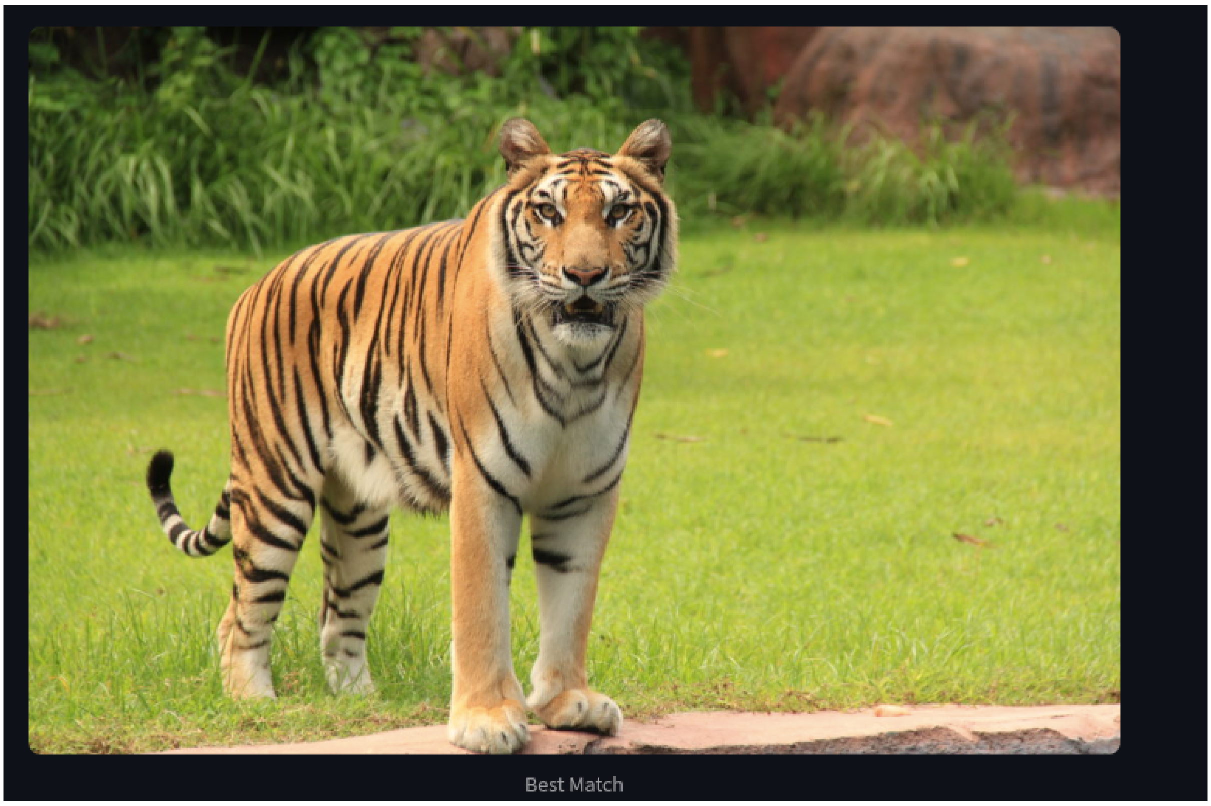

Figure 2: Best match retrieved from the database.

- **Alternative Embeddings:** Explore modern backbones (ResNet, EfficientNet) or fine-tuned metric-learning models to boost retrieval precision.
- **Advanced Similarity Metrics:** Incorporate learned distance metrics or re-ranking strategies (e.g., query expansion, diffusion) to refine results.
- **Enhanced User Interface:** Add filtering, dynamic sliders for the number of matches, and on-the-fly parameter tuning via Streamlit widgets.

These future directions aim to improve scalability, accuracy, and user experience, moving toward a production-ready CBIR solution.

6 Conclusion and Future Work

We have presented an interactive, content-based image retrieval system that leverages deep features from a pre-trained VGG16 network, stores them efficiently in an HDF5 index, and retrieves semantically similar images in under one second per query. Our Streamlit demo displays the uploaded query, the best match, a 3D PCA visualization of feature space, and the top-3 results side by side. This end-to-end pipeline confirms the efficacy of deep convolutional embeddings for semantic retrieval.

Looking forward, several extensions can further enhance our approach:

- **Incremental Indexing:** Support adding new images on the fly without rebuilding the entire feature database.
- **GPU Acceleration:** Offload feature extraction and PCA computation to the GPU for large-scale datasets and faster updates.
- **Alternative Embeddings:** Explore modern backbones (ResNet, EfficientNet) or fine-tuned metric-learning models to boost retrieval precision.

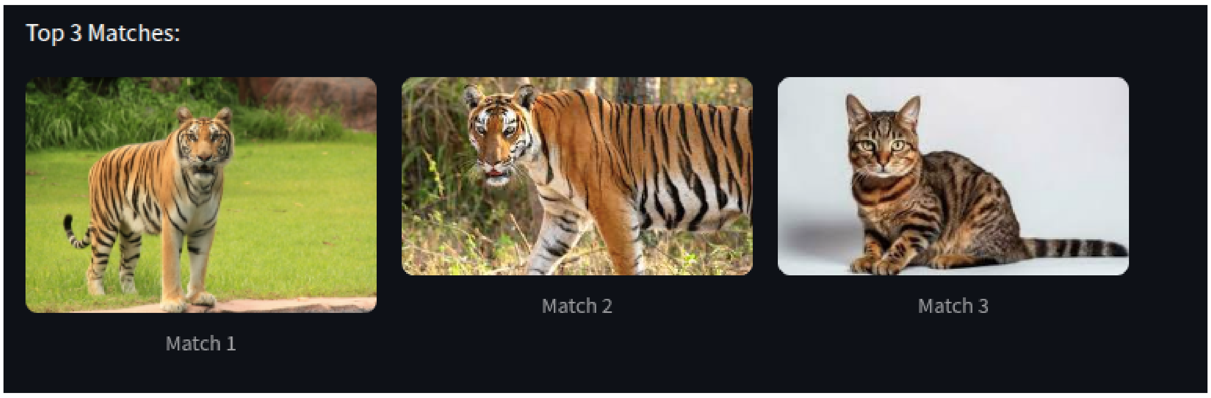

Figure 3: Top-3 gallery of most similar images.

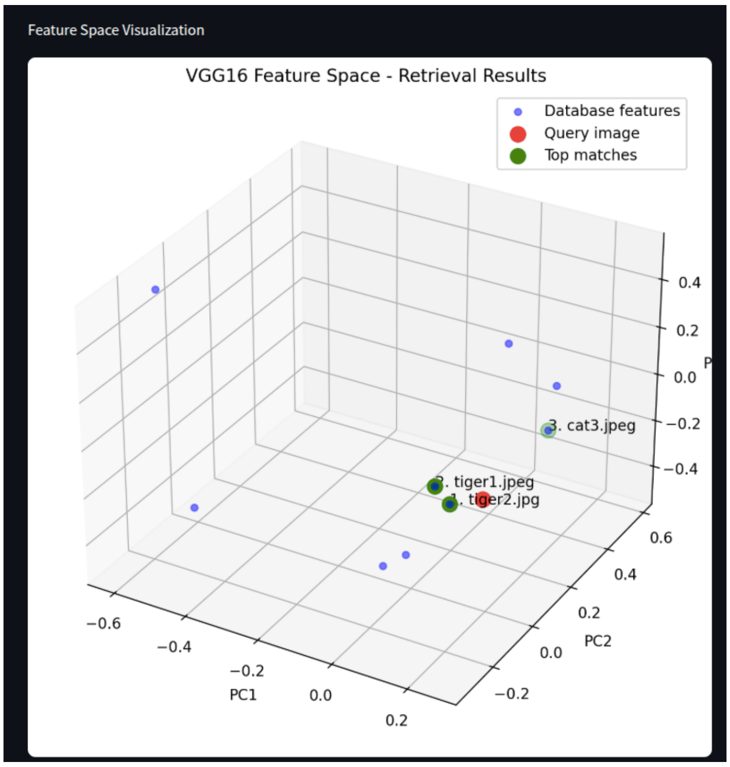

Figure 4: 3D PCA visualization of the feature space including query and top matches.

- **Advanced Similarity Metrics:** Incorporate learned distance metrics or re-ranking strategies (e.g., query expansion, diffusion) to refine results.
- **Enhanced User Interface:** Add filtering, dynamic sliders for the number of matches, and on-the-fly parameter tuning via Streamlit widgets.

References

[1] Author(s), “Efficient Deep Feature-Based Semantic Image Retrieval,” *Proceedings*, Year.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[3] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[4] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(1):117–128, 2011.

[5] J. Wan, D. Wang, S. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, 2014.

[6] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *European Conference on Computer Vision (ECCV)*, 2014.