COVID-19 Search Engine - Information Retrieval Course Project

Matteo Farè ID: 989345

University of Milan, Department of Computer Science.

Contributing authors: matteo.fare1@studenti.unimi.it;

Abstract

Content-Based Image Retrieval (CBIR) holds significant promise in medical imaging, enhancing image analysis and retrieval. This review examines a CBIR system using VGG16, MobileNet, and ResNet50 for feature extraction to retrieve the top-N similar chest CT scans based on cosine similarity. The models are compared on processing time, precision, recall, F1 score, and Discounted Cumulative Gain (DCG).

1 Introduction

Content-Based Image Retrieval (CBIR) systems are designed to retrieve relevant images from large datasets based on the visual content of the images rather than metadata or keywords. These systems analyze various features such as color, texture, and shape to find images that are visually similar to a given query image.

In the context of the medical field, CBIR systems can help manage and utilize the large amounts of imaging data generated daily. Even if the real applications are still scares, as described in [1] and [2], these systems can aid healthcare professionals by providing a quicker access to similar cases, supporting diagnosis, treatment planning, and medical research[1].

This study focuses on implementing a CBIR system for retrieving chest CT scans of COVID-19 positive cases and healthy patients. The system uses three deep learning models, VGG16, MobileNet, and ResNet50, implemented through the Keras framework. These models are well-known for their effectiveness in image recognition tasks and are adapted here for features extraction.

[1]How CBIR Supports Radiologists in Clinical Routine: https://www.youtube.com/watch?v=DIs6JBUGHlE

2 Research Question and Methodology

2.1 Research Question

The primary goal of this project is to evaluate the effectiveness and efficiency of the adopted models in a CBIR system for retrieving similar chest CT scans. The research aims to determine which model provides better performance in terms of computational time and classification metrics.

2.2 Proposed Approach

The CBIR system retrieves the top N most similar images based on cosine similarity of the extracted features. The steps involved include:

1. **Dataset Preparation**: Collect and preprocess a dataset of chest CT scans, including COVID-19 positive and healthy images.
2. **Feature Extraction**: Utilize VGG16, MobileNet, and ResNet50 models to extract features from the images.
3. **Similarity Measurement**: Calculate cosine similarity between the feature vectors of the query image and the dataset images.
4. **Relevance Feedback**: Apply Relevance Feedback through Rocchio's Algorithm to improve retrieval performance.
5. **Performance Evaluation**: Assess the models based on processing time, precision, recall, F1 score, and DCG.

2.3 Formal Problem Definition

Given a query image I_q and a database of images $D = \{I_1, I_2, ..., I_n\}$, the problem is to find the top-N images $\{I_{r1}, I_{r2}, ..., I_{rN}\}$ from D that are most similar to I_q based on cosine similarity of their feature vectors.

3 Experimental Results

3.1 Dataset

The employed dataset [3] comprises chest CT scans from COVID-19 positive patients and healthy individuals. The dataset is pre-processed to ensure the integrity and quality of the data removing any corrupted files. In the context of this study, no instances of corrupt files were identified.

It's worth noting that while there is a slight imbalance in the dataset, it is not significant enough to negatively impact the performance of the models. For a more detailed understanding and visual representation of the dataset, refer to Appendix A: Figure A1.

After completing the quality assurance phase, the dataset was divided into training and test sets, with the latter consisting of 20% of the total dataset.

3.2 Evaluation Metrics

The performance of the CBIR system is evaluated using the following metrics:

- **Processing Time**: The time taken to extract features from the dataset.
- **Precision**: The ratio of relevant images retrieved to the total images retrieved, calculated as:

$$Precision = \frac{TP}{TP + FP}$$

  where:

  – TP represents True Positives (relevant images retrieved),
  – FP represents False Positives (irrelevant images retrieved).

- **Recall**: The ratio of relevant images retrieved to the total relevant images, calculated as:

$$Recall = \frac{TP}{TP + FN}$$

  where:

  – FN represents False Negatives (relevant images not retrieved).

- **F1 Score**: The harmonic mean of precision and recall, calculated as:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

- **Discounted Cumulative Gain (DCG)**: A measure of ranking quality that evaluates the usefulness, or gain, of an image based on its position in the result list. This metric helps in understanding how well the retrieved images are ranked, giving higher importance to relevant images appearing at the top of the list. The DCG is calculated as:

$$DCG = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

  where:

  – p is the position of the result in the list,
  – rel_i is the relevance score of the result at position i.

3.3 Experimental Methodology

The project involves three deep learning models, VGG16, MobileNet, and ResNet50, employed for extracting features from the dataset. The next step is to measure the cosine similarity to retrieve the top-N similar images. then, after the initial retrieval and evaluation, Rocchio's Algorithm is applied to refine the query representation based on relevant and irrelevant feedback.

3.4 Baseline System

To evaluate the performance of the CBIR system, a baseline reference using random retrieval is established. This approach retrieves images randomly from the dataset in response to a query, serving as a benchmark to compare against more advanced retrieval methods and assess their improvements.

3.5 VGG16 Model

The VGG16 architecture, introduced by the Visual Geometry Group at the University of Oxford [4], represents a significant advancement in the field of deep learning for image recognition. The model emphasizes depth through a straightforward architecture composed of 16 weight layers. The model uses small 3x3 convolution filters, a design choice that enhances its ability to learn intricate patterns while maintaining computational efficiency.

3.6 MobileNet Model

MobileNet [5] is a lightweight convolutional neural network architecture specifically crafted for efficient and resource-friendly image processing tasks. Its key innovation lies in the use of depthwise separable convolutions, a strategy aimed at achieving high performance with minimal computational overhead. This approach significantly reduces the number of parameters, making MobileNet suitable for deployment on devices with limited computational resources.

3.7 ResNet50 Model

ResNet50 [6], introduced by Microsoft Research, is a deep convolutional neural network that has significantly impacted the field of image recognition. The core innovation of ResNet50 lies in its use of residual learning, which allows the network to train much deeper models without the problems of vanishing or exploding gradients. This is achieved through shortcut connections that bypass one or more layers, enabling the gradient to flow directly through the network.

3.8 Customizing the models

In this study, all the models class are initialized with weights pre-trained on the ImageNet[2] dataset, which ensures that the convolutional layers are adept at recognizing a wide array of visual features. By customizing the models, removing their classification layers[3], and adding layers like *"GlobalAveragePooling2D"*, *"Flatten"*, and *"Dense"*, these implementations allow for the extraction of high-dimensional feature vectors that represent the input images in a form that is suitable for similarity comparison and image retrieval.

[2] ImageNet Dataset:https://www.image-net.org/about.php
[3] Usage example use as reference for features extraction: https://keras.io/api/applications/#usage-examples-for-image-classification-models

3.9 Feature Extraction Process

1. **Preprocessing**: Images are pre-processed, including resizing to match the input shape, converting to NumPy arrays, expanding dimensions for batch processing, and applying model-specific preprocessing.
2. **Feature Extraction**: Features are extracted from the images using the customized models.
3. **Normalization**: Extracted feature vectors are normalized to ensure consistency and facilitate comparison.

3.10 Top-N Similar Image Retrieval

1. **Cosine Similarity**: Cosine similarity is calculated between the features of the query image and features of all other images in the dataset.
2. **Sorting**: The calculated similarities are sorted, and indices of the closest images are retrieved.
3. **Top-N Retrieval**: The top-N similar images are selected based on the sorted similarity values. In this study, the value of N was set to the value of 100.

3.11 Plotting Similar Images

The *plot_similar_images* function visualizes the query image along with the first k results among the top-N similar images. To provide a concise and intuitive demonstration a small subset is randomly selected from the test dataset, providing a total of ten images that are equally split between the two classes.

The function displays the images in a grid-like structure, highlighting the similarity scores and corresponding class labels, and uses color coding to distinguish between images belonging to the same class as the query and those from the "opposite" class. This visual representation provides an intuitive understanding of the CBIR capabilities.

For a more detailed understanding and visual representation of the results, refer to Appendix A: Figure A2, Figure A4, and Figure A6.

3.12 Relevance Feedback with Rocchio's Algorithm

Rocchio's Algorithm (1) is a relevance feedback mechanism commonly used in information retrieval systems to refine search results. It adjusts the feature vectors of the query based on feedback provided by the user, typically in the form of relevant and irrelevant documents.

The algorithm works by moving the query vector closer to the centroids of the relevant documents and away from the centroids of the irrelevant documents in the feature space. This iterative process aims to enhance retrieval accuracy by iteratively refining the query representation.

In this study, Rocchio's Algorithm is employed to adapt the query representation based on feedback obtained from the initial retrieval results, with the goal of improving the relevance of the retrieved images.

$$q = \alpha q_0 + \beta \frac{1}{|D_r|} \sum_{d \in D_r} d - \gamma \frac{1}{|D_n|} \sum_{d \in D_n} d \tag{1}$$

where:

- q_0 is the original query vector,
- D_r is the set of relevant documents,
- D_n is the set of non-relevant documents,
- α, β, γ are constants to adjust the influence of the original query (set to *1*), relevant documents (set to *0.75*), and non-relevant documents (set to *0.15*), respectively.

For more detailed examples and insights into the relevance feedback process, refer to Appendix A: Figure A3, Figure A5, and Figure A7.

3.13 Results

- **Processing Time**: As shown in the Table A1, MobileNet exhibited notably faster processing times compared to ResNet50 and VGG16.
- **Evaluation Metrics**: As depicted in the Tables A2 to A8, all adopted models consistently outperform the baseline system in all parameters, improving further after the relevance feedback process. Among the models, ResNet50 exhibits the best overall performance, followed by MobileNet and VGG16.

4 Concluding Remarks

In conclusion, the evaluation of the CBIR system reveals that ResNet50 achieves the best result values among the adopted models. However, it's crucial to acknowledge the significance of MobileNet's performance, particularly considering its low processing time and near-comparable results to ResNet50. MobileNet's efficiency is particularly noteworthy due to its architecture designed for devices with low computational capability, such as smartphones. Therefore, while ResNet50 may excel in performance metrics, the efficiency and practicality of MobileNet make it a highly interesting choice in the regards of this study.

4.1 Future Work

Future research could explore integrating diverse deep learning architectures, such as EfficientNet or DenseNet, that could potentially offer better feature extraction capabilities. Additionally, it could explore different customizations for the models used, the application of hyperparameter tuning, and the implementation of clustering and unsupervised algorithms, such as k-means. Furthermore, the implementation of K-Nearest Neighbors (KNN) algorithm could be potentially interesting.

Appendix A Images and Tables

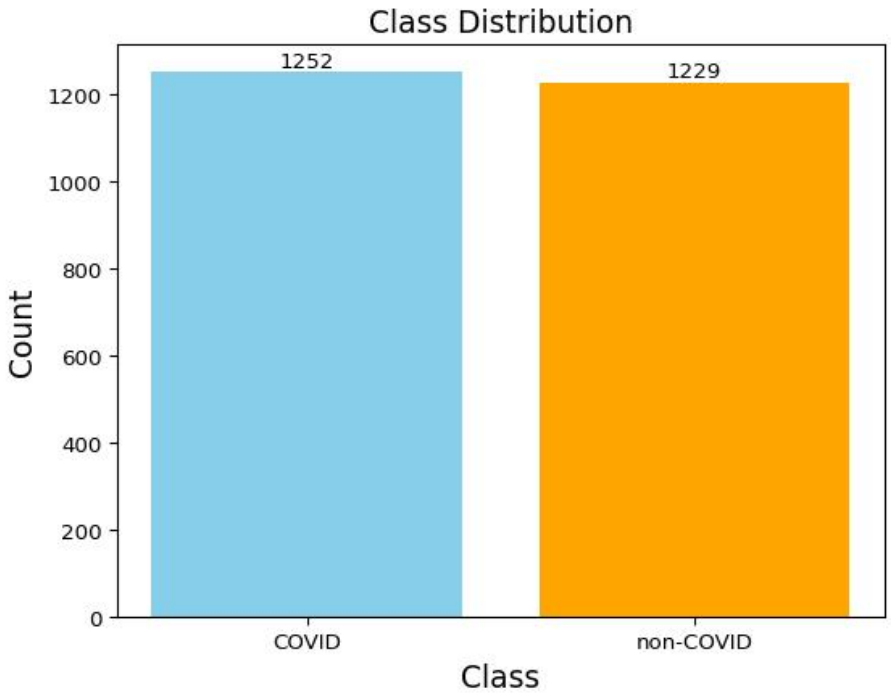

Fig. A1 Dataset Class Distribution

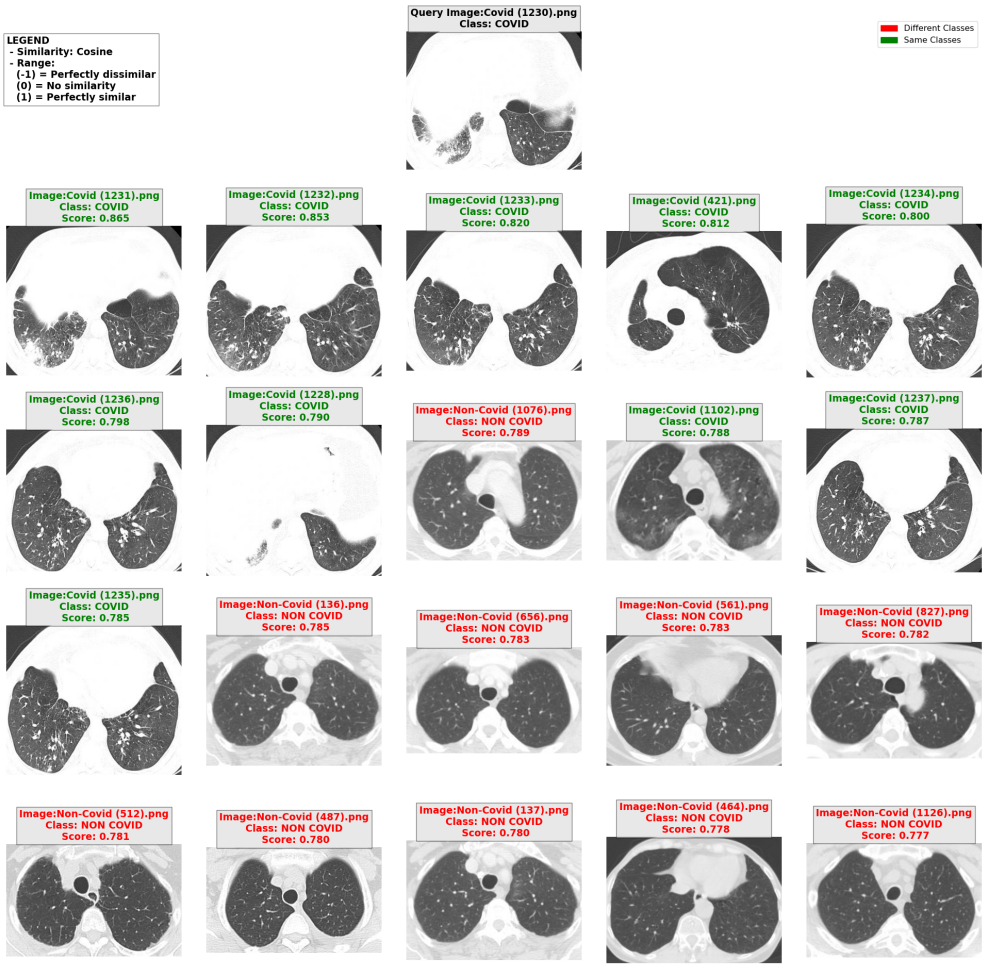

Fig. A2 MobileNet Example of Top-20 Similar Image Retrieved

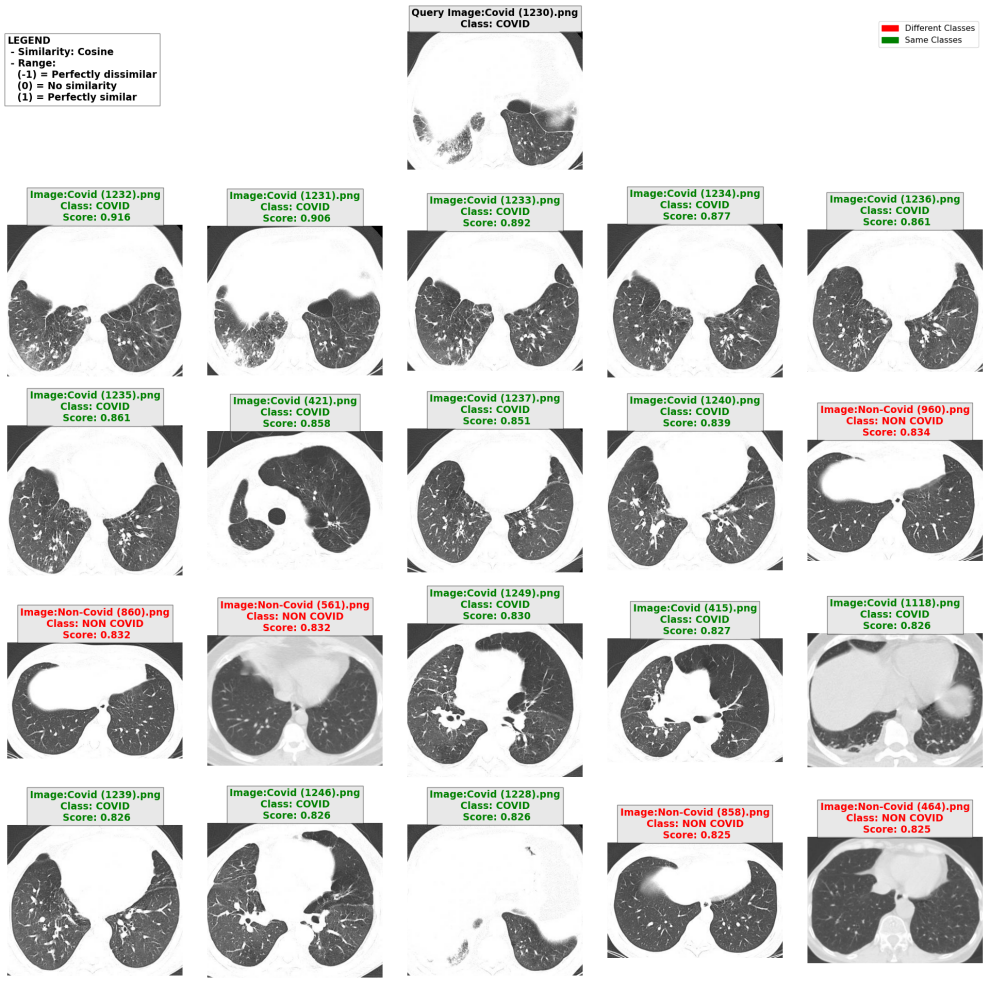

Fig. A3 MobileNet Example after Relevance Feedback

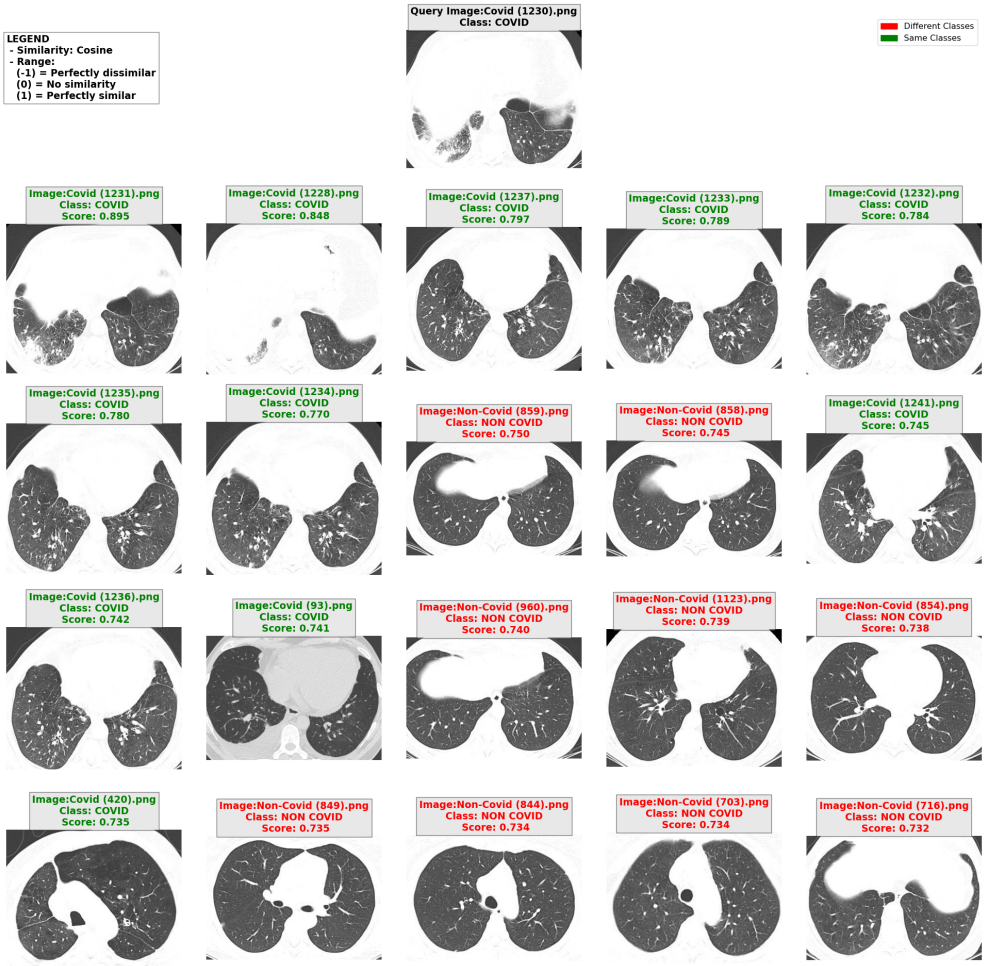

Fig. A4 ResNet50 Example of Top-20 Similar Image Retrieved

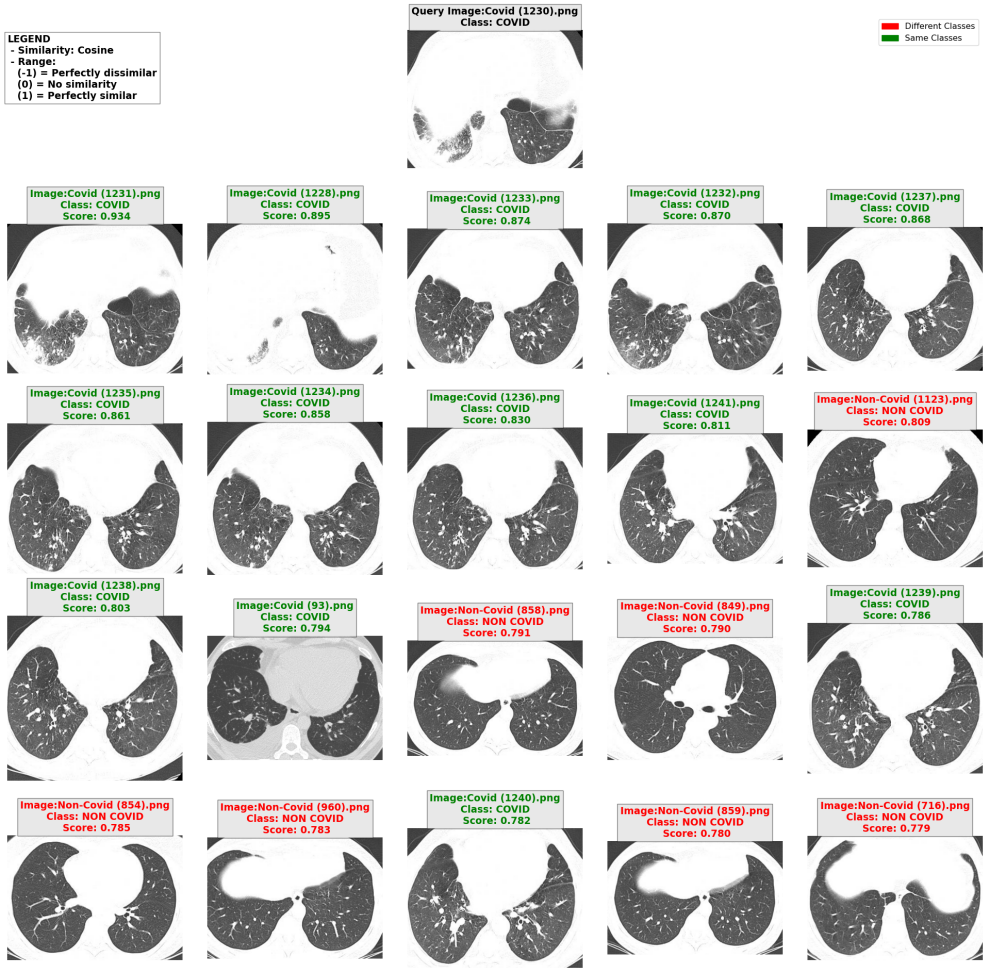

Fig. A5 ResNet50 Example after Relevance Feedback

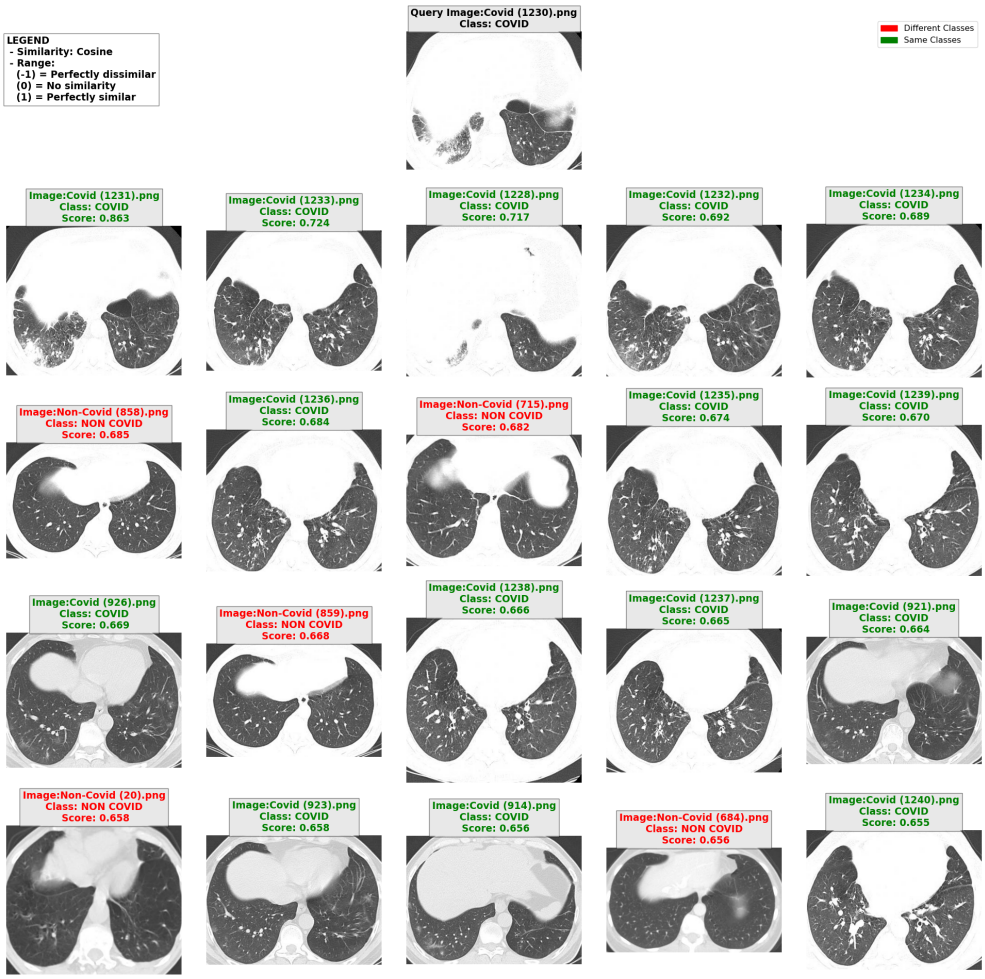

Fig. A6 VGG16 Example of Top-20 Similar Image Retrieved

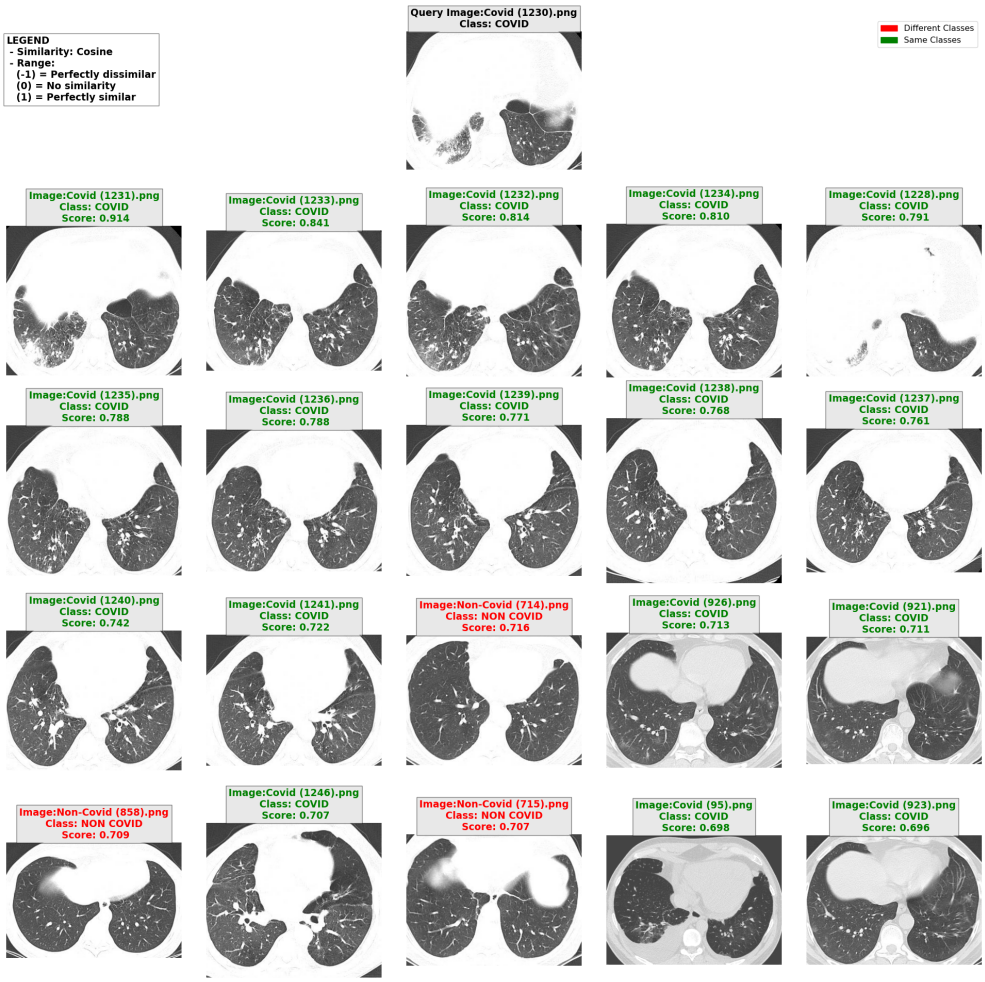

Fig. A7 VGG16 Example after Relevance Feedback

Model Name	Time (sec.)	Time (min.)
MobileNet	97.60	1.63
ResNet50	197.65	3.29
VGG16	358.45	5.97

Table A1 Time taken to extract features from the dataset

K	Avg. Precision	Avg. Recall	Avg. F1	Avg. DCG
5	0.4716	0.0475	0.0863	1.3948
10	0.4813	0.0971	0.1614	2.1765
20	0.4947	0.1994	0.2836	3.4454
30	0.4932	0.2979	0.3706	4.4858
40	0.4944	0.3981	0.4400	5.4464
50	0.4973	0.5004	0.4977	6.3659
60	0.4962	0.5994	0.5417	7.2098
70	0.4965	0.6997	0.5795	8.0321
80	0.4968	0.8000	0.6115	8.8295
90	0.4970	0.9005	0.6391	9.6049
100	0.4968	1.0000	0.6624	10.3565

Table A2 Baseline System Top-N Evaluation

K	Avg. Precision	Avg. Recall	Avg. F1	Avg. DCG
5	0.9002	0.0714	0.1309	2.7000
10	0.8445	0.1304	0.2225	3.9602
20	0.7934	0.2397	0.3609	5.8152
30	0.7636	0.3419	0.4621	7.3096
40	0.7443	0.4420	0.5421	8.6335
50	0.7291	0.5387	0.6057	9.8411
60	0.7164	0.6340	0.6577	10.9633
70	0.7054	0.7270	0.7005	12.0205
80	0.6964	0.8196	0.7370	13.0332
90	0.6882	0.9100	0.7676	13.9999
100	0.6809	1.0000	0.7940	14.9340

Table A3 MobileNet Top-N Evaluation

K	Avg. Precision	Avg. Recall	Avg. F1	Avg. DCG
5	0.9038	0.0701	0.1286	2.7102
10	0.8545	0.1296	0.2213	3.9957
20	0.8048	0.2387	0.3605	5.8846
30	0.7768	0.3415	0.4636	7.4138
40	0.7572	0.4417	0.5447	8.7619
50	0.7423	0.5394	0.6100	9.9955
60	0.7282	0.6322	0.6613	11.1257
70	0.7174	0.7258	0.7054	12.2046
80	0.7090	0.8187	0.7433	13.2432
90	0.7007	0.9093	0.7746	14.2296
100	0.6943	1.0000	0.8028	15.1959

Table A4 ResNet50 Top-N Evaluation

K	Avg. Precision	Avg. Recall	Avg. F1	Avg. DCG
5	0.9095	0.0718	0.1322	2.7211
10	0.8608	0.1335	0.2288	4.0196
20	0.7932	0.2419	0.3656	5.8338
30	0.7596	0.3453	0.4674	7.3034
40	0.7358	0.4441	0.5447	8.5848
50	0.7174	0.5401	0.6058	9.7486
60	0.7038	0.6350	0.6562	10.8412
70	0.6922	0.7284	0.6978	11.8701
80	0.6811	0.8186	0.7311	12.8352
90	0.6729	0.9103	0.7611	13.7794
100	0.6655	1.0000	0.7862	14.6865

Table A5 VGG16 Top-N Evaluation

K	Avg. Precision	Avg. Recall	Avg. F1	Avg. DCG
5	0.982	0.077	0.141	2.910
10	0.912	0.138	0.236	4.258
20	0.844	0.247	0.374	6.197
30	0.808	0.349	0.477	7.760
40	0.787	0.450	0.560	9.158
50	0.767	0.545	0.623	10.397
60	0.752	0.639	0.676	11.561
70	0.740	0.732	0.720	12.663
80	0.727	0.821	0.755	13.686
90	0.718	0.911	0.787	14.690
100	0.710	1.000	0.814	15.651

Table A6 MobileNet Top-N Evaluation after Relevance Feedback

K	Avg. Precision	Avg. Recall	Avg. F1	Avg. DCG
5	0.986	0.077	0.140	2.918
10	0.926	0.139	0.237	4.303
20	0.865	0.251	0.380	6.311
30	0.827	0.354	0.484	7.904
40	0.800	0.452	0.564	9.294
50	0.779	0.547	0.628	10.552
60	0.763	0.640	0.681	11.729
70	0.750	0.731	0.725	12.831
80	0.738	0.822	0.762	13.882
90	0.728	0.911	0.793	14.889
100	0.720	1.000	0.821	15.867

Table A7 ResNet50 Top-N Evaluation after Relevance Feedback

K	Avg. Precision	Avg. Recall	Avg. F1	Avg. DCG
5	0.982	0.076	0.140	2.908
10	0.933	0.142	0.243	4.320
20	0.858	0.254	0.386	6.280
30	0.813	0.357	0.488	7.814
40	0.782	0.455	0.566	9.145
50	0.757	0.548	0.626	10.328
60	0.739	0.642	0.676	11.449
70	0.725	0.733	0.718	12.510
80	0.711	0.820	0.750	13.487
90	0.702	0.911	0.780	14.462
100	0.693	1.000	0.806	15.394

Table A8 VGG16 Top-N Evaluation after Relevance Feedback

References

[1] Kumar A, C.W.F.M.F.D. Kim J: Content-based medical image retrieval: a survey of applications to multidimensional and multimodality data. Journal of digital imaging **26**(6), 1025–1039 (2013) https://doi.org/10.1007/s10278-013-9619-2

[2] Röhrich, H.B.H.P.F.e.a. S.: Impact of a content-based image retrieval system on the interpretation of chest cts of patients with diffuse parenchymal lung disease. Eur Radiol **33**, 360–367 (2023) https://doi.org/10.1007/s00330-022-08973-3

[3] Soares, E., Angelov., P.: Sars-cov-2 ct-scan dataset [data set]. kaggle. (2020) https://doi.org/10.34740/KAGGLE/DSV/1199870

[4] Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition (2015). https://doi.org/10.48550/arXiv.1409.1556

[5] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications (2017). https://doi.org/10.48550/arXiv.1704.04861

[6] He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition (2015). https://doi.org/10.48550/arXiv.1512.03385