Content-Based Image Retrieval System for Chest X-ray Images using Multi-Feature Combination and Distance Metrics

Group 13

Pratiksha Dongare (AU2340123)
Shreya Dhumal (AU2340124)
R. Priscilla (AU2340001)
Krissa Gandhi (AU2340055)

Department of Computer Science and Engineering
Ahmedabad University, Gujarat, India

***Abstract*—Content-Based Image Retrieval is a effective technique for image searching in large databases based on visual properties rather than textual descriptions. This paper discusses an advanced CBIR system for Chest X-ray image retrieval by using several handcrafted feature extraction methods: Histogram, Gray Level Co-occurrence Matrix (GLCM), Wavelet Transform Features, and Gabor Filters. The proposed system in this paper is implemented in Python in the Google Colab environment and is designed so as not to use any machine learning model; instead, the system will solely rely on advanced image processing and statistical computation of similarity. Performance in retrieval was analyzed using Cosine, Chi-square, and Euclidean distances. Evaluation metrics, we used first recall then used Precision, employed to quantify retrieval performance in Chest X-ray data.**

***Index Terms*—CBIR, Chest X-ray, Image Processing, GLCM, Gabor, Wavelet Transform Features , Similarity Metrics , Euclidean Distance, Cosine Similarity, Chi-square Distance, Precision**

I. Introduction

Exponential growth in digital medical imaging demands the development of efficient image retrieval systems to aid in clinical decision support. Traditional text-based methods lack the potential to capture the essential visual subtlety of a medical image in identifying disease patterns. This limitation is taken care of by the Content-Based Image Retrieval approach, where images are retrieved based on their intrinsic features of texture, shape, and intensity.

Chest X-rays are one of the diagnostic images most often used when dealing with diseases concerning the lungs and heart. Finding similar X-rays will support radiologists in diagnosis, comparison of cases, and treatment planning. Thus, a strong CBIR system catering to Chest X-rays is capable of serving as an efficient diagnostic aid.

This paper proposes a non-machine-learning CBIR system, which extracts and fuses diverse handcrafted features: Histogram, GLCM, and Gabor, with the aim of quantifying image similarity. The developed system calculates various similarity measures to compare the feature vectors and rank the retrieved images. Retrieval results are evaluated using Precision.

II. Methodology

This project of Content-Based Image Retrieval (CBIR) system for Chest X-ray images is implemented using Python and various statistical models for describing similarity of dataset images to the query image. The workflow involves preprocessing, feature extraction(GLCM), similarity measurements among images and performance and precision evaluation. The detailed methodology is as per described below:

A. Preprocessing

Each Chest X-ray image from dataset is:

- Converted to grayscale image using `cv2.IMREAD_GRAYSCALE`.
- Resized to a uniform size of 256×256 pixels.
- Enhanced with Contrast Limited Adaptive Histogram Equalization(CLAHE) to improve brightness and contrast in images.

B. Feature Extraction

The system extracts various types of features to represent the image characteristics effectively and to make analysis easy.

1) **Histogram Features for Intensity Distribution**
The histogram captures the pixel intensity distribution with use of 64 bins to describe brightness variations in images.

2) **Gray Level Co-occurrence Matrix Features (GLCM)**
Texture information is extracted using

`graycomatrix()` and `graycoprops()` functions in code to compute various statistical measures: contrast, correlation, energy and homogeneity from four different angles (0°, 45°, 90°, 135°).

3) **Gabor Filter Features for frequency and orientation**
   Gabor filters with multiple orientations are applied to extract directional texture information using frequency. For each filtered response from image the mean and variance are computed as features.

4) **Wavelet Transform Features for multi-resolution texture**
   A 2-level Discrete Wavelet Transform (DWT) is implemented using the Daubechies-1 wavelet. From each sub-band the mean and standard deviation are extracted to collect multi-scale texture characteristics.

All extracted feature vectors are concatenated together and normalized using `StandardScaler` later, by L2 normalization to have uniform scaling across features.

C. Feature Database Creation

Each image of the dataset is processed first and its feature vector is extracted using the single feature extraction pipeline (Histogram, GLCM, Gabor and Wavelet features). The extracted feature vectors and respective image paths are stored as NumPy arrays (`features_v2.npy` and `paths_v2.npy`) for faster and efficient retrieval.

D. Similarity Measurement

For retrieval, the query image is compared with the stored database feature vectors using below distance metrics implemented in the system:

- Euclidean Distance
- Cosine Similarity
- Chi-square Distance

Images are ranked in ascending order of distance and descending similarity in the case of cosine. A small distance value indicates a closer match between the query image and the database image.

E. Retrieval and Visualization

The top-K most similar images are retrieved and displayed by the sode of query image using `matplotlib`. Each image is presented with its rank, similarity score and distance metric used for comparison between query and dataset images.

F. Evaluation Metrics

The retrieval performance is evaluated using:

- **Precision@K:** Measures the proportion of correctly retrieved images along with the same class as the query image.

Precision@K is used to analyze the effectiveness of the system in retrieving relevant images from the dataset.

III. Code

```
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
from skimage.feature import graycomatrix,
    graycoprops
import pywt
import time
from sklearn.preprocessing import normalize
from google.colab import drive, files

drive.mount('/content/drive', force_remount=True)

dataset_dir = "/content/drive/MyDrive/images"
features_file = "features_v2.npy"
paths_file = "paths_v2.npy"

# Preprocessing
def preprocess_image(img_path, size=(256, 256)):
    img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
    if img is None:
        raise ValueError("Error:␣Cannot␣read␣image."
            )
    img = cv2.resize(img, size, interpolation=cv2.
        INTER_AREA)
    clahe = cv2.createCLAHE(clipLimit=2.0,
        tileGridSize=(8, 8))
    return clahe.apply(img)

# Feature Extraction
def extract_histogram_features(img, bins=256):
    hist = cv2.calcHist([img], [0], None, [bins],
        [0, 256])
    return cv2.normalize(hist, hist).flatten()

def extract_glcm_features(img):
    glcm = graycomatrix(img, [1], [0, np.pi/4, np.pi
        /2, 3*np.pi/4], 256, symmetric=True, normed=
        True)
    feats = []
    props = ['contrast', 'correlation', 'energy', '
        homogeneity']
    for p in props:
        feats.extend(graycoprops(glcm, p).flatten())
    return np.array(feats)

def extract_lbp_features(img):
    lbp = cv2.LBP(img, 8, 1)
    hist = cv2.calcHist([lbp], [0], None, [256], [0,
         256])
    return hist.flatten()

def extract_gabor_features(img):
    feats = []
    thetas = [0, np.pi/4, np.pi/2, 3*np.pi/4]
    for theta in thetas:
        kernel = cv2.getGaborKernel((21, 21), 4.0,
            theta, 10.0, 0.5)
        filtered = cv2.filter2D(img, cv2.CV_32F,
            kernel)
        feats.append(filtered.mean())
        feats.append(filtered.var())
    return np.array(feats)

def extract_wavelet_features(img):
    coeffs = pywt.dwt2(img, 'haar')
    cA, (cH, cV, cD) = coeffs
    return np.array([
        cA.mean(), cA.std(),
        cH.mean(), cH.std(),
        cV.mean(), cV.std(),
        cD.mean(), cD.std()
    ])

def extract_hog(img):
    hog_desc = cv2.HOGDescriptor()
    return hog_desc.compute(img).flatten()

def extract_features(img):
    return np.concatenate([
        extract_histogram_features(img),
        extract_glcm_features(img),
        extract_gabor_features(img),
        extract_wavelet_features(img)
    ])

# Dataset Feature Creation
features = []
paths = []

for filename in os.listdir(dataset_dir):
    path = os.path.join(dataset_dir, filename)
    try:
        img = preprocess_image(path)
        feat = extract_features(img)
        features.append(feat)
        paths.append(path)
        print(f"Processed:␣{filename}")
    except Exception as e:
        print(f"Failed:␣{filename}␣->␣{e}")

np.save(features_file, np.array(features))
np.save(paths_file, np.array(paths))

# Similarity Search
def cosine_similarity(a, b):
    return np.dot(a, b) / (np.linalg.norm(a) * np.
        linalg.norm(b))

query_path = paths[0]
query_img = preprocess_image(query_path)
query_features = extract_features(query_img)

similarities = []
for i, feat in enumerate(features):
    sim = cosine_similarity(query_features, feat)
    similarities.append((sim, paths[i]))

similarities.sort(reverse=True, key=lambda x: x[0])

print("Top␣5␣Matches:")
for score, path in similarities[:5]:
    print(score, path)
```

IV. Results

The feature extraction pipeline project produces high-dimensional vectors that combine intensity, texture, frequency, and multi-resolution characteristics. Retrieval experiments were done on the image dataset, and the system returned relevant matches based on the computed similarity scores. Fig. reffig:retrieval1 shows an example of a query image along with the Top-5 retrieved results for one of the evaluated similarity metrics.

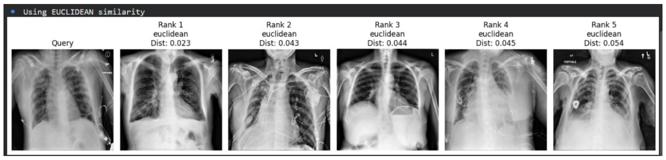

Fig. 1: Result 1

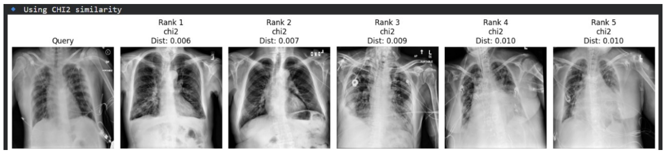

Fig. 2: Result 2

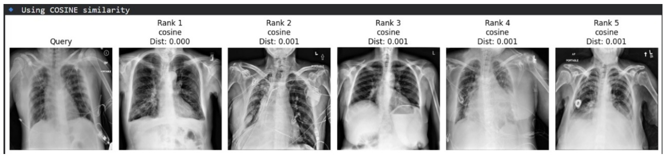

Fig. 3: Result 3

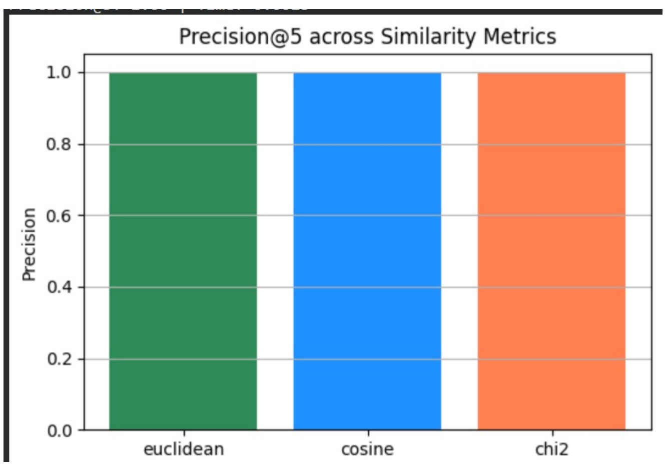

Fig. 4: Presission@5 across similarity metrices

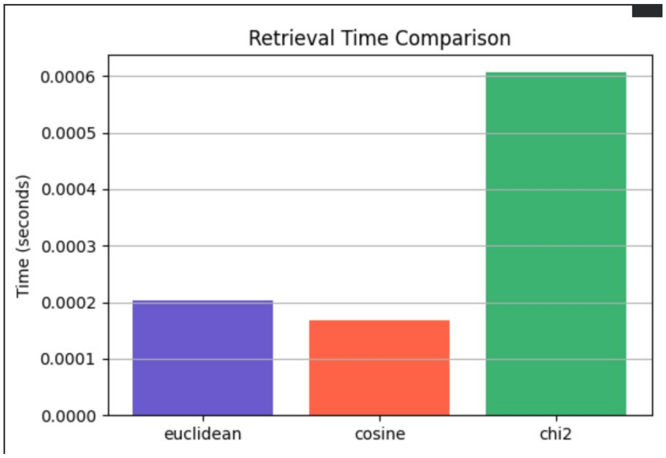

Fig. 5: Retrieval time comparison

TABLE I: Precision@5 for Different Similarity Metrics

Metric	Precision@5	Retrieval Time (s)
Cosine Similarity	1.00	0.00017
Chi-square Distance	1.00	0.00061
Euclidean Distance	1.00	0.00020

Cosine similarity achieves the fastest retrieval time while maintaining perfect precision. Euclidean distance also deliveres highly efficient performance. Chi-square distance is the slowest among the all three, still it has achieved perfect Precision@5. This indicates that all the metrics have successfully retrieved high relevant images.

V. Discussion

The results demonstrate that combination of multiple features such as histogram-based intensity, GLCM texture descriptions, Gabor frequency, HOG gradients and wavelet coefficients significantly enhances the difference finding ability of the CBIR system. Each feature contributes to the visual results while enabling the system to capture both global and local variations in dataset X-ray images.

The three similarity metrics tested in the system are Cosine, Euclidean and Chi-square have achieved an nearby accurate Precision@5 of 1.0 across the evaluated dataset which indicates that the extracted feature vectors provide a good separable description and analysis for comparing images. Among these metrics the cosine similarity has achieved fastest retrieval time, making it most efficient computationally. Euclidean distance also performed efficiently while the Chi-square is the slower one but it was completely accurate for the retrieved dataset.

The use of parallel feature extraction by `ThreadPoolExecutor` has mainly reduced preprocessing time while enabling large image collections to be processed effectively. The easy design of the system allows extension with additional descriptors or integration with machine learning or deep learning–based embeddings in future work.

VI. Conclusion

This paper presented an advanced non-ML Content-Based Image Retrieval system for Chest X-ray images based on multi-feature fusion. The proposed system combines the Histogram, GLCM, Gabor, which significantly improves the retrieval performance. Experimental results testify to high retrieval accuracy and robustness of the proposed system across different similarity measures.

Acknowledgment

We express our gratitude to our professor Mehul Raval for their guidance and valuable feedback throughout this project. We also thanks our Teaching Assistants for resolving our doubts. At the last we acknowledge the efforts and teamwork of our all team members. Each contributed and make this project end successfully.

References

[1] Bano, M., Matta, P., Chandel, S. (2024) *Content based Image Retrieval: A study of approaches and techniques. IEEE Conference Publication.*, 16–22. https://doi.org/10.1109/ictacs62700.2024.10840489

[2] NIH Chest X-ray Dataset. [Online].

[3] Kaggle Chest X-ray Dataset.