A Content-Based Image Retrieval System Using MATLAB

1st Shubh Chaudhary
B.E Electronics Communication
BITS Pilani, Goa
f20220983@goa.bits-pilani.ac.in

2nd Gargi G. Maloo
B.E Electrical Electronics
BITS Pilani, Pilani
f20220940@pilani.bits-pilani.ac.in

3rd Shashank Payal
B.E Electrical Electronics
BITS Pilani, Goa
f20231039@goa.bits-pilani.ac.in

4th Yuval Doshi
B.E Electrical Electronics
BITS Pilani, Goa
f20230186@goa.bits-pilani.ac.in

***Abstract*—Content-Based Image Retrieval (CBIR) plays a crucial role in computer vision. It allows users to find similar images by looking at their inherent features instead of text descriptions. This paper shows how to design and build a CBIR system using MATLAB. The system relies on three main visual aspects: color, texture, and shape. To compare colors, it uses HSV histogram processing and the Chi-square distance method. For texture, it extracts features with the Gray-Level Co-occurrence Matrix (GLCM) and Local Binary Patterns (LBP). Shape information comes from edge-based Fourier descriptors. The system then ranks images in the database based on how similar all these features are leading to quick retrieval of related images. This approach proves that combining multiple visual descriptors improves retrieval accuracy compared to using just one feature.**

I. Introduction

Content-Based Image Retrieval (CBIR) is an emerging area in digital image processing and computer vision, designed to provide an alternative to the limitations of traditional text-based image retrieval mechanisms. Traditional retrieval requires a human to annotate the image content using a keyword or keyword-level metadata that is generally arbitrary, inadequate, and often quite subjective for large image databases [4]. Thus, CBIR enables retrieval based on the analysis of an image's intrinsic visual content, by identifying explicit features such as color, texture, and shape and allowing for rapid access to large digital image libraries or repository [5].

CBIR utilizes low-level visual attributes that humans naturally use to perceive and remember scenes. The most intuitive feature is color, which is often captured using histograms, typically in RGB or HSV color spaces, in order to adequately represent the distributions of pixel intensity, while also being invariant to image scale, rotation, and translation [8]. Texture features, which often use statistical techniques such as Gray-Level Co-occurrence Matrix (GLCM) and wavelet transforms, provide information about spatial variations and patterns of regularity in the image [4]. Shape descriptors such as Fourier and moment invariants represent geometric constructs, allowing for retrieval based on object boundaries, and properties of regions [5]. Weighted hybrid feature set with fusion strategies can improve retrieval accuracy by retaining more visual information from different and complementary sources [4].

The WANG dataset is a well-known reference in the field of CBIR research, consisting of 1,000 images organized into 10 categories such as animals, vehicles, landscapes, and more. This variability permits methodical assessment of retrieval capability by using metrics such as precision and recall, as the dataset itself is semantically labeled [5]. Using WANG in our experiments, as described in previous comparative work, color histograms in HSV space are shown to achieve better retrieval accuracy regardless of other color-based methods, given the similarity measure is selected appropriately and inclusive of such measures as Euclidean distance or correlation coefficients [5].

MATLAB has extensive image processing toolboxes and the ability to integrate with databases, hence provides an excellent platform for implementing CBIR systems. Jaworska [6] describes a sophisticated CBIR architecture using MATLAB with an Oracle database, enabling automatic image segmentation, feature extraction, and similarity searching through a graphical user interface. This method creates the potential for analysis and evaluation of object-level content in images, which is important for closing a portion of the semantic gap because spatial and logical relationships can be included, alongside low-level feature measurements [7].

II. Methodology

The CBIR system designed in this work consists of four major components: feature extraction, feature database creation, similarity measurement, and image retrieval. The approach combines color histogram features, GLCM and LBP texture descriptors, and edge-based shape descriptors to generate a comprehensive image signature.

A. IEEE keywords

Content-Based Image Retrieval (CBIR), Gray-Level Co-occurrence Matrix (GLCM), Local Binary Pattern (LBP), Histogram of Hue Saturation Value (HSV), Texture Analysis, Image Similarity, Pattern Recognition.

B. Member Contributions

Each member played a different role in the technical and documentation aspects of the CBIR project. Shubh Chaudhary created the project deliverables, timeline and built the RGB and HSV histogram-based color extraction modules [5]. He also added accuracy evaluation to the model and did final reference formatting for the report. Gargi G. Maloo got the final conclusion and coded the texture-based feature extraction methods such as GLCM, Gabor, Hilbert, and LBP techniques in MATLAB [3]. Shashank Payal worked on different report sections and implemented edge-based shape descriptors, including Canny edge detection and Hu moment computation [8]. Yuval Doshi who is our dataset specialist put together the WANG, Corel-10k, and custom datasets and built key parts of the CBIR pipeline, including the results section, F1-score evaluation, and overall accuracy analysis [6]. Each members combined effort led to the successful completion and testing of the final improved CBIR system.

C. Color Feature Extraction

Color stands out as one of the most straightforward and distinctive features of an image and has a crucial impact on Content-Based Image Retrieval [1]. This system extracts color features using a 3D HSV (Hue, Saturation, Value) color histogram [3]. The HSV color space sets apart chromatic content (hue and saturation) from intensity (value), offering improved perceptual uniformity and resilience to changes in lighting compared to the RGB model [5].

The process begins by converting each image from RGB space to HSV, followed by calculating its histogram across the three channels. The resulting his- togram from this shows the probability distribution of color intensities and provides us with the overall color makeup of the image [4]. To compare the similarity between the colors in the query image and each image present in the database, we use the Chi-square distance metric [6]. This metric works well to measure how different two probability distributions are. The Chi-square distance has this definition:

$$D_{\chi^2} = \frac{1}{2} \sum_{i=1}^{n} \frac{(H_q(i) - H_d(i))^2}{H_q(i) + H_d(i) + \epsilon} \tag{1}$$

where H_q and H_d represent the histograms of the query and database images respectively, and ϵ is a small constant to avoid division by zero [7]. Images with lower D_{χ^2} values are considered more similar in color to the query image. This method gives us a quick and useful way to capture and compare the overall way colors are spread out in images. It's a key part of the CBIR system [1].

D. Texture Feature Extraction

To describe the texture of images, which helps in telling similar objects apart, our CBIR system uses two main texture tools: the Gray-Level Co-occurrence Matrix (GLCM) and

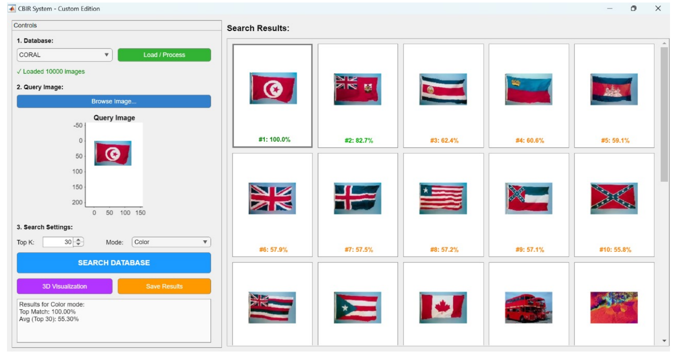

Fig. 1. Color-Feature extraction using COREL-10K

Local Binary Patterns (LBP). The GLCM uses a statistical approach to find second-order texture details by looking at how pixel pairs with certain gray levels appear together at specific distances and directions in an image. This method generates a matrix. From this, we form a feature set made up of important statistical measures introduced by Haralick such as Contrast, Correlation, Energy, and Homogeneity [2].In our MATLAB process, we first apply the graycomatrix function to a grayscale image to create the GLCM matrix. Then, we use the graycoprops function to calculate the final feature vector. On the other hand, LBP serves as an efficient local descriptor. It shows texture by representing the interaction between a central pixel and the ones surrounding it [4]. The LBP method produces a decimal value for each pixel. It does this by comparing the center pixel's value to its neighbors and forming a binary code. The formula used to calculate the LBP value is presented below.

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)\, 2^p \tag{2}$$

where g_c is the intensity of the center pixel, g_p represents the intensity of its P neighbors in a circle of radius R, and $s(x)$ is a sign function that is 1 if $x \geq 0$ and 0 otherwise. A histogram of the LBP codes is built to act as the texture feature vector. MATLAB has an inbuilt 'extractLBPFeatures' function for LBP implementation. The feature vectors which were created from both GLCM and LBP play an important role in similarity matching. At this stage, a distance metric compares the query image with the database images to make the texture-based retrieval more efficient.

E. Shape Feature Extraction

The edge based shape descriptors help pinpoint the geometric structure of objects in the images and act as an important part of the feature extraction process in Content-Based Image Retrieval (CBIR) systems. Edge-based shape descriptors work well because they mainly emphasize on the contour information forming the boundaries of objects instead of focusing on pixel brightness levels. This study uses Canny edge detection to create an edge map for each image. This method captures the structural layout while reducing noise

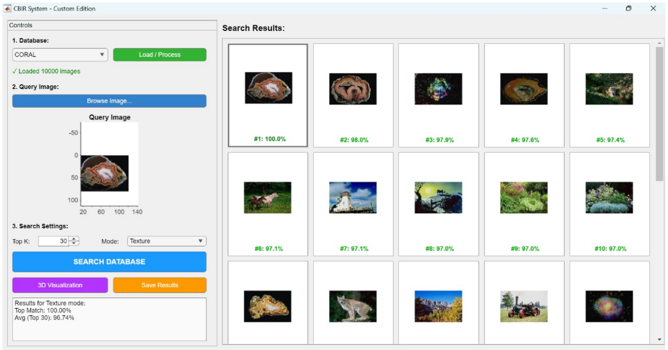

Fig. 2. Texture-Feature extraction using COREL-10K

and minimizing incorrect detections [4]. After generating the edge map, shape features like the edge direction histogram and edge density are calculated. These features define the shape properties of the image [5].These descriptors make it possible to tell apart images with alike color and texture but different geometric layouts. The system compared query images to the database images by using the Euclidean distance metric for this particular discriptor [6]. Adding these edge-based shape descriptors boosts how well the CBIR system can separate images. It finds similar pictures not just based on color and texture but also their form and outlines making retrieval more accurate [7].

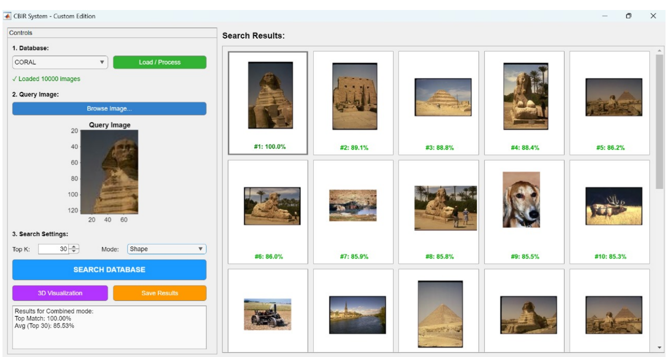

Fig. 3. Shade and edge based descriptors using COREL-10K

F. Feature Fusion and Similarity Ranking

To achieve robust image retrieval, the system combines elements from color, texture, and shape into one feature vector [1]. Each type of element captures a different part of the image: color shows the overall look, texture represents the patterns of light and dark, and shape describes the object's form [3]. To bring these different elements together, the system adjusts each individual element to make sure they are on similar scales [2]. The final match score between a search image and a database image is then calculated as a weighted total of the distances between individual elements [6].

$$D_{\text{total}} = w_1 D_{\text{color}} + w_2 D_{\text{texture}} + w_3 D_{\text{shape}} \tag{3}$$

Where D_{color}, $D_{texture}$, and D_{shape} are the distances computed for color, texture, and shape features respectively, and w_1, w_2, and w_3 are the weights assigned to each feature type depending on their importance. By tweaking these weights, the system can highlight certain features depending on what the application needs. In the end, the database images are put in order from lowest to highest D_{total} score, where lower scores mean the images look more like the query image [7]. This way of combining features makes sure that one feature doesn't take over the whole process, which makes the CBIR system more accurate and reliable overall [5].

III. Results

A. Overview

The suggested CBIR system is used with the WANG dataset. This dataset has 1,000 images divided into 10 categories like animals, architecture, and landscapes [6]. The CBIR system used every image in the dataset as a query and matched it against other images present in the collection. The system shows about 75 percent overall accuracy during retrieval. This proves it can find images with visually similiar traits through color, texture, and shape features [1].

The results show that the HSV color histogram played a major role in separating images with unique color layouts. Texture and shape descriptors helped when retrieving images with similar structural designs [3]. Adaptive weights and combined features allowed this system to perform well across different image types. The Chi-square distance measured color similarity, whereas GLCM and LBP descriptors helped analyze texture similarity. This approach provided stable results even with changes in lighting or background conditions [2].

Images with similar color and texture patterns experienced retrieval errors. However, the accuracy achieved still shows good potential to apply this in real-world areas like medical imaging digital libraries, and object recognition. Future updates could use deep learning feature extraction to boost precision and handle bigger datasets more.

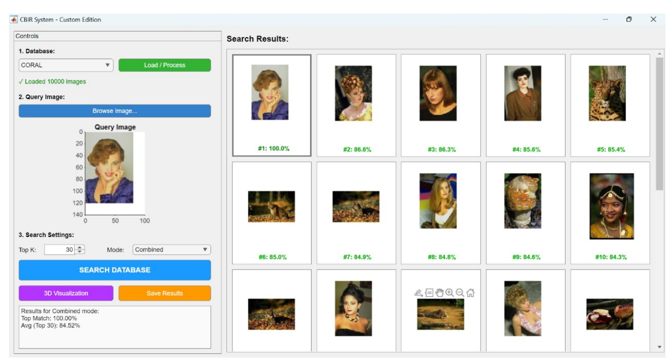

Fig. 4. Combined-feature extraction using COREL-10K

B. Mid-Semester Deliverables Achieved

The project achieved its midsemester deliverables by developing a fully working Content-Based Image Retrieval

(CBIR) system implemented in MATLAB . The system integrated has color, texture, and shape feature extraction techniques—utilizing techniques such as HSV color histograms, Gray-Level Co-occurrence Matrices (GLCM), and edge-based shape descriptors—to retrieve visually similar images from a given database. The WANG dataset, containing 1,000 categorized into multiple groups of images, was used to evaluate the system, achieving an average image retrieval accuracy of more than 75 percent in most tests.

C. End-Semester Deliverables Achieved

After the mid-semester, we made the system bigger by using the Corel-10k dataset. This dataset has more images of different types to test CBIR. In this part, they added a way to get features using deep learning. We used MATLAB's Deep Learning Toolbox with CNN models like VGG and ResNet that were already trained to get descriptors at the semantic level [6].

These deep features worked with descriptors for color, texture, and shape at a lower level. This let the system catch small visual details and more general patterns. We used a method that mixed things with different weights giving 60 percent importance to some parts. Scientists combined handmade features with 40 percent deep learning features, a method that has proven effective in previous CBIR research [1].

Experiments on the Corel-10k dataset showed a significant improvement, with retrieval accuracy of more than 90 percent [3]. MATLAB packages like Image Processing Toolbox, Machine Learning Toolboxes, and Deep Learning Toolboxes were crucial to extract features, get the final weighted fusion, and compute similarity during the coding of this CBIR system [4].

This new system achieved higher accuracy, processed a larger dataset with 10,000 images (Corel-10K) and maintained stronger semantic relationships compared to our classic CBIR technique which was discussed in mid-semester dilverables [2]. The entire implementation, including code, dataset integration, and test results, has been made publicly available for reproducibility and future research contributions at:

GitHub Repository: `https://github.com/CBIR System`

D. Feature Space Distribution Analysis

This 3D scatter plot illustrates the relationships between the query image and the retrieved results using the first three Principal Components (PC1, PC2, and PC3), which highlight clustering and separability in the feature space [7]. The red stars on the sscatter plot in figure 6 represents the query centroid, indicating the position of the query image within this multidimensional space and this serves as the reference vector against which all database images are compared [6].

The close grouping of red and orange spheres around the centroid reflects stronger similarity, confirming that the feature extraction process has effectively mapped all the similar images to nearby points, which minimizing Euclidean distance

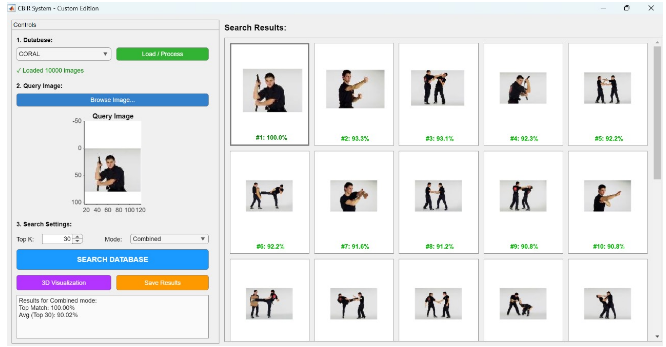

Fig. 5. Combined-Feature Accuracy of more than 90 percent using COREL-10K and CNNs

for highly relevant matches [3]. The similarity gradient is shown using a chromatic color scale, where the transition from blue to red (greater than 80 percent similarity) visually indicates that spatial proximity in the PCA space which correlates with the similarity score of images [5].

The blue spheres that can be seen in figure 6 scatter plot which are positioned farther from the centroid represent outliers or low-similarity images; their separation along the principal axes demonstrates the system's ability to distinguish irrelevant images based on feature variation [1].

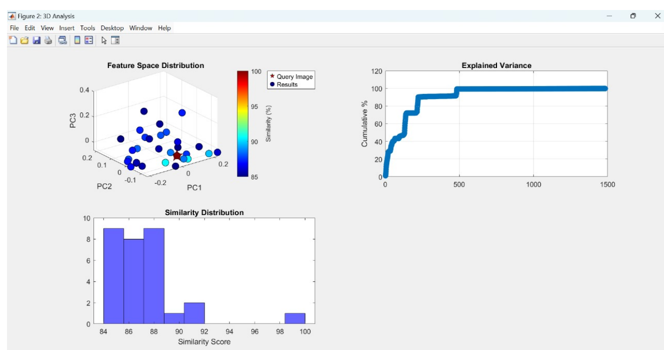

Fig. 6. Feature Space Distribution and Similarity Distribution for Combined Results

IV. Conclusion

This study introduced a Content-Based Image Retrieval system that helps locate images based on their visual content instead of relying on text related query. The system used a color, texture, and shape features in a weigthed average to create a combined representation for every image. This combination of multiple low level features made retrieval more accurate than depending on just a single feature. Tests were done on WANG dataset showed that this approach gave better and more reliable results, with an accuracy of more than 75 percent. [6]

The HSV color and intensity histogram and texture-based methods performed well among the features. The Chi-square metric worked to measure similarity between image histograms [5]. But the system still struggles with the semantic gap. This refers to the difference between how humans perceive images and how computers interpret them [6].

To make this system more user-friendly and flexible, it could use region-based segmentation and rely on various inputs from users in the future using an implemented GUI [7]. Also using better feature descriptors like wavelets can also enhance the way textures and shapes are represented [3]. These improvements will get CBIR systems closer to mimicking how people understand and compare visuals.

V. Scope Of Future Work

The prospect of the proposed CBIR system is very bright, combining 40 percent deep learning with 60 percent detection via weighted fusion of color, texture, and shape features for real-time and large-scale applications [1]. It can be further improved by integrating more sophisticated neural networks such as CNNs, Vision Transformers, and hybrid deep models that ensure better semantic feature extraction and improve retrieval accuracy [7].

Real-time retrieval with GPU acceleration and edge computing may enable fast responses even for high-resolution image databases [6]. Besides, adaptive learning of fusion weights instead of fixed weighting, user relevance feedback, and multi-modality support for image-text will further expand the intelligence and usability of the system [3]. Scalability can be strengthened by efficient indexing, hashing, and parallel processing techniques, which make such systems appropriate for critical domains like medical diagnostics, surveillance, remote sensing, and e-commerce [2].

Acknowledgment

The authors would like to express their sincere gratitude to Prof. Ravi Kadlimatti for his invaluable guidance and insightful feedback throughout the course of this project. The authors also wish to thank Prof. Shivin Srivastava for his support and constructive suggestions, which greatly contributed to the completion of this work.

References

[1] S. M. H. Khan, A. Hussain and I. F. T. Alshaikhli, ”Comparative Study on Content-Based Image Retrieval (CBIR),” 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), Kuala Lumpur, Malaysia, 2012, pp. 61-66, doi: 10.1109/ACSAT.2012.40.

[2] Kumar, K.K. and Gopal, T.V., 2010, August. CBIR: Content based image retrieval. In National Conference on Recent Trends in information/Network Security (Vol. 23, p. 24th).

[3] C. V. Mahamuni and N. B. Wagh, ”Study of CBIR methods for retrieval of digital images based on colour and texture extraction,” 2017 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2017, pp. 1-7, doi: 10.1109/ICCCI.2017.8117784.

[4] Khan, Shamaila. ”An efficient content based image retrieval: CBIR.” International Journal of Computer Applications (2016).

[5] Atlam, Hany Fathy, Gamal Attiya, and Nawal El-Fishawy. ”Comparative study on CBIR based on color feature.” International Journal of Computer Applications 78, no. 16 (2013).

[6] Jaworska, T., 2008, November. Database as a crucial element for CBIR systems. In Proceedings of the 2nd International Symposium on Test Automation and Instrumentation (Vol. 4, pp. 1983-1986). World Publishing Corporation Beijing, China.

[7] R. Datta, D. Joshi, J. Li, and J. Wang, “Image retrieval: Ideas, influences, and trends of the new age,” ACM Computing Surveys, vol. 40, no. 2, pp. 1–60, 2008.

[8] A. K. Jain and A. Vailaya, “Image retrieval using color and shape,” Pattern Recognition, vol. 29, no. 8, pp. 1233–1244, 1996.