Non-textual data extraction assignment - *implementation of a ‘toy’ CBIR*

Group 4

DELLA VALLE FRANCESCO
GONZALEZ-ALMANSA LAREDO CARLOS
SAVIANO GAETANO
JAIN ARTH AMIT

INFORMATION RETRIEVAL, EXTRACTION AND INTEGRATION
FEBRUARY 2024

CASE STUDY

In the contemporary digital landscape, the proliferation of image data demands sophisticated retrieval systems that can efficiently and accurately sift through vast repositories of visual content. Our project centers on the creation of a **Content-Based Image Retrieval (CBIR)** system—a technology that enables users to search for images based on their visual features rather than relying solely on textual annotations.The primary goal was to develop a CBIR system capable of retrieving images that closely match a given query image in terms of visual content.

State of the art - CBIR

The integration of CBIR systems offers several applications across numerous domains, capitalizing on their ability to analyze and retrieve images based on their visual content. The ways that CBIR can be utilized are:

- **Art Collections Management**: CBIR systems can assist art curators and historians in managing large art collections by facilitating the search for artworks based on visual similarities, styles, or attributes.
- **Medical Imaging Analysis**: In the field of medical imaging, CBIR systems aid healthcare professionals in diagnosing diseases, planning treatments, and conducting research.
- **Satellite Imagery Analysis**: CBIR systems play a crucial role in satellite imagery analysis for various applications, including urban planning, environmental monitoring, and disaster response.

Color space: color based image retrieval is the most basic and most important method for CBIR. Color features are the most intuitive and most obvious image features. A color histogram is the most used method to extract color features. A color histogram is a frequency statistic for different colors in a certain color space. The advantage is that it describes the global color distribution for images. It is especially suited for those images difficult to segment and neglect spatial locations. However, its drawback is that it cannot describe the local distribution of the image in color space and the spatial position of each color. It means that the color histogram cannot describe specific objects or things in the 4image. The color space needs to be divided into several small ranges in order to calculate the color histogram. Each interval is regarded as a bin. Thus, the color is quantized. The color histogram can be calculated through counting pixels where the colors fall into each interval. [3]
Descriptor generation: In order to find low level descriptors, systems extract vectors of features from images and use specific metrics for comparing them. The CBIR's approach is to use composed descriptors, where different descriptors are combined into a single feature vector and it is used to retrieve images, for example this feature vector can include descriptors of color, shape and texture. In this paradigm, it uses only one algorithm to index images and it is going to retrieve the most similar images to the query image based in relation to the unique feature vector. [4]
Distance measurement: content-based image retrieval calculates visual similarities between a query image and images in a database. Therefore, the retrieval result is not a single image but a number of images ranked by their similarities with the query image. The query image will be more similar to the database images if the distance is smaller. Different similarity measures will affect retrieval performances of an image retrieval system significantly, so it is important to find the best distance metric for the CBIR system. [5]
The technical proposition for the implemented CBIR consists of using a color space RGB approach, because of its simplicity and wide use, the implementations of both ‘smart’ Histogram’s and Harris’ descriptors and distances, the combination of the descriptors, and the additional functionalities of indexing and searching. The implementation is provided in the attachment to this documentation.

Project Objectives

- **CBIR System Development**: The core objective was to design and implement a CBIR system capable of analyzing and comparing images based on their visual features.
- **Implementation of Smart Histograms and distance**: Designing and implementing an intelligent histogram module capable of capturing and analyzing image descriptors choosen in a more nuanced and context-aware manner. The first descriptors the group decided is color.
- **Implementation of a second descriptor and distance**: The group choosed as second descriptors for the dataset the shape. The distance implemented is the Harris Distance.

- **Descriptors Combination:** Once extracted the features, the group focused on how to use the features combined together.
- **Implementation Additional CBIR processes**: Implementing searching and indexing methods for searching and storing images features.

METHODOLOGICAL APPROACH

The GitHub repository link for this project is:
https://github.com/GaetanoSaviano/Non-TextualDataExtractor

Dataset

Our dataset consists of 22 images of varying sizes and types, capturing diverse visual content including landscapes and urban scenes. To enhance the evaluation process, we've decided to expand our dataset by adding additional images. This augmentation aims to increase diversity and provide a more comprehensive testbed for evaluating our algorithms.
The link to the dataset repository is : https://imagedatabase.cs.washington.edu/groundtruth/italy/

CBIR System Development

Given the characteristics of our problem (abstractly finding visually similar images from a collection) our system should focus on visual characteristics, features of the images that we perceive by the vision. Because of that, we chose to focus our design into taking into account mainly two factors, the colors present in the image and the shapes the image presents.
During the following points the decisions made on each of this aspects.

Implementation of Smart Histograms and distance

The first descriptor, the histograms, representation of the colors present in its pixels are calculated in our code with the `cv2.calcHist` function, which returns a normalized histogram for every channel of the image (in our case, because of using RGB, for red, green, and blue channels) so that different size images can be compared utilizing it.
Once we have the descriptor, we had to choose a distance to compare instances and get the most similar ones in the color dimension. For us, the paper of N.D. Gagunashvili [1] about chi-squared distance called our attention, and seeing that this measure was also implemented in the `cv2.compareHist` function we opted for using it, in our case without the weighting part.

Implementation of a second descriptor and distance

As aforementioned the second descriptor should be one representing the shapes the image is constructed by. There's a lot of literature on shape detection as it is probably the main way of recognition in our vision and we opted for the harris descriptor as multiple studies suggest to use.
Harris descriptor searches for corner points, this means points where lines change direction tessellating the image into the different shapes and elements forming the whole. For this our image is transformed into a GrayScale pixel collection, retrieving the information of all three channels, and is passed through the `cv2.cornerHarris` method.
To compare this new descriptor we opted for, in divided regions of the images, compare these harris descriptors treating them as vectors and calculating the euclidean distance.

Descriptors combination

Measures to check the similarities between these the descriptors of two images have been mentioned, but we should take both of these similarities into account because both of them form the way we perceive the images. For that we read the distance between the query image and the database images in both descriptors as a vector. This means we convert the similarities into a two dimensional vectorial space in which every axis is the distance of the image with the query image (so, the query image represents the point 0,0). We thought about

implementing other methods to join these two, like similarity multiplication explained in *Combining minutiae descriptors for fingerprint matching by Feng*[2], but this one seemed for us the more natural to us.
One detail is to make this possible, given we observed the two distances mentioned before having very different ranges, is to normalize the distance arrays (the arrays containing the distances from the query image to every of the database images) before combining them.

Implementation Additional CBIR processes

Our database can maybe not be too large but, in order to make the system scalable we don't store images themselves but the descriptors of every image (identified by its unique name) into a *.npy* file which can be easily loaded when starting the program. New images could be added, for that it just need to calculate its descriptors and store them in the file by utilizing the flag -add *filepath*, there would be no need to recalculate them all (if necessary, can also be done with the flag -rd).This *sim_search(img, db)* function performs image similarity search using a descriptor database. It consisted of: calculating color histograms for each color channel in the input image (img) and storing them in a dictionary *'hist_dict'*; appling the Harris corner detection algorithm to the grayscale version of the input image and stores the results in *'imgharr'*; calculating two types of distances (based on color histograms and Harris descriptors) for each item in the database (*'db'*) and saves the results; normalizing the calculated distance vectors; using an utility function *'find_min_values()'* to find the images in the database with the smallest distances compared to the query image. It returns a list of file names and their similarities. In summary, this function effectively implements an image similarity search process using color and Harris descriptors, aligned with the concept of Content-Based Image Retrieval (CBIR).

RESULTS

Performance Metrics

We employed a similarity metric based on the Euclidean distance to calculate the five most similar images to the query image. This approach provided us with valuable insights into the resemblance between images within our dataset. However, it's important to note that while the results are discernible, there remains potential for enhancement.

Future Directions

Future iterations of our system could benefit from refining the similarity metric or exploring alternative approaches to achieve more precise and robust image retrieval outcomes. Despite the current limitations, these initial findings serve as a solid foundation for further refinement and optimization, ultimately contributing to the continual advancement of our CBIR system.
Some of the future directions that can be implemented are :

- **Integration of Deep Learning Techniques:** Incorporating deep learning techniques, such as convolutional neural networks (CNNs), could significantly enhance the performance and robustness of the CBIR system. By leveraging CNNs for feature extraction and representation learning, the system can automatically learn discriminative features from images, leading to more accurate and semantically meaningful similarity assessments.
- **Multimodal Fusion:** Exploring multimodal fusion approaches to integrate information from multiple modalities, such as text, audio, and video, could further enrich the CBIR system's capabilities. By combining visual features with complementary data sources, such as textual descriptions or audio annotations, the system can provide more comprehensive and contextually relevant image retrieval results, catering to a wider range of user needs and preferences.

References

[1] Gagunashvili, N. D. (2010). Chi-square tests for comparing weighted histograms. In Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment (Vol. 614, Issue 2, pp. 287–296). Elsevier BV. https://doi.org/10.1016/j.nima.2009.12.037
[2] Feng, J. (2008). Combining minutiae descriptors for fingerprint matching. In Pattern Recognition (Vol. 41, Issue 1, pp. 342–352). Elsevier BV. https://doi.org/10.1016/j.patcog.2007.04.016
[3] Jun Yue, Zhenbo Li, Lu Liu, Zetian Fu. (2011). Content-based image retrieval using color and texture fused features
[4] Raquel E.Patino-Escarcina, Jose Alfredo F. Costa. (2007). Content Based Image Retrieval using a Descriptors Hierarchy
[5] Sanjay Patil,Sanjay Talbar. (2010). Content Based Image Retrieval Using Various Distance Metrics