Ain Shams University
Faculty of Engineering
Computer and Systems Engineering Department

Multimedia Project Report,

Content-based Image Retrieval System (CBIR)
Content-based Video Retrieval System (CVIR)

Group members:

name	code
Mostafa Mohamed Badawi Farhat	16X0127
Ahmed Hamed Ahmed Abdallah	16W0002
Ahmed Mohamed Fahmy Aly	15X0015
Salah Soliman El-Sayed Soliman	15T0510

Table of Contents

Abstract

Multimedia content analysis is applied in different real-world computer vision applications, and digital images constitute a major part of multimedia data. In last few years, the complexity of multimedia contents, especially the images, has grown exponentially, and on daily basis, more than millions of images are uploaded at different archives such as Twitter, Facebook, and Instagram. Most of the search engines retrieve images on the basis of traditional text-based approaches that rely on captions and metadata. In the last two decades, extensive research is reported for content-based image retrieval (CBIR), image classification, and analysis.

Content-based image retrieval, also known as query by image content (QBI C) and content-based visual information retrieval (CBVIR), is the application of compute r vision techniques to the image retrieval problem, that is, the problem of searching for digital image s in large database s.

Content based image retrieval, in the last few years has received a wide attention. Content Based Image Retrieval (CBIR) basically is a technique to perform retrieval of the images from a large database which are similar to image given as query. CBIR is closer to human semantics, in the context of image retrieval process. CBIR technique has its application in different domains such as crime prevention, medical images, weather forecasting, surveillance, historical research and remote sensing. Here content refers to the visual information of images such as texture, shape and color. Contents of image are richer in information for an efficient retrieval in comparison to text based image retrieval.

In order to build this system, we use a 3D color histogram in the HSV color space. A 3D HSV color descriptor will ask a given image how many pixels have a Hue value that fall into bin #1 AND how many pixels have a Saturation value that fall into bin #1 AND how many pixels have a Value intensity that fall into bin #1. The number of pixels that meet these requirements are then tabulated. This process is repeated for each combination of bins. Once, we have our image descriptor, we extract features from each image in our dataset. The process of extracting features and storing them on persistent storage is commonly called “indexing”. We represent the feature vectors that represent and quantify the image into a csv file. We then compare these features for similarity. This method will take two parameters, the queryFeatures extracted from the query image (i.e. the image we’ll be submitting to our CBIR system and asking for similar images to), and limit which is the maximum number of results to return. To compare images we have utilized the chi-squared distance, a popular choice when comparing discrete probability distributions and return necessary results.

Problem statement and Introduction

Due to recent development in technology, there is an increase in the usage of digital cameras, smartphone, and Internet. The shared and stored multimedia data are growing, and to search or to retrieve a relevant image from an archive is a challenging research problem. The fundamental need of any image retrieval model is to search and arrange the images that are in a visual semantic relationship with the query given by the user. Most of the search engines on the Internet retrieve the images on the basis of text-based approaches that require captions as input. The user submits a query by entering some text or keywords that are matched with the keywords that are placed in the archive. The output is generated on the basis of matching in keywords, and this process can retrieve the images that are not relevant. The difference in human visual perception and manual labeling/annotation is the main reason for generating the output that is irrelevant.

It is near to impossible to apply the concept of manual labeling to existing large size image archives that contain millions of images. The second approach for image retrieval and analysis is to apply an automatic image annotation system that can label image on the basis of image contents. The approaches based on automatic image annotation are dependent on how accurate a system is in detecting color, edges, texture, spatial layout, and shape-related information.

Objective: To build a system that can excerpt visual content of an image inevitably, like color, shape or texture. Our goal here is to build an image search engine. Given our dataset of photos, we want to make this dataset “search-able” by creating a “more like this” functionality — this will be a “search by example” image search engine.

Implementation

Programming Language: Python

Packages: numpy, cv2, imutils, argparse, glob, csv.

Dataset: INRIA ftp://ftp.inrialpes.fr/pub/lear/douze/data/jpg1.tar.gz [1]

CBIR

Content-Based Image Retrieval System can be boiled down into 4 distinct steps:

Defining image descriptor: At this phase we decide what aspect of the image we want to describe. Are you interested in the color of the image? The shape of an object in the image? Or do you want to characterize texture?

Indexing the dataset: Now that you have your image descriptor defined, your job is to apply this image descriptor to each image in your dataset, extract features from these images, and write the features to storage (ex. CSV file, RDBMS, Redis, etc.) so that they can be later compared for similarity.

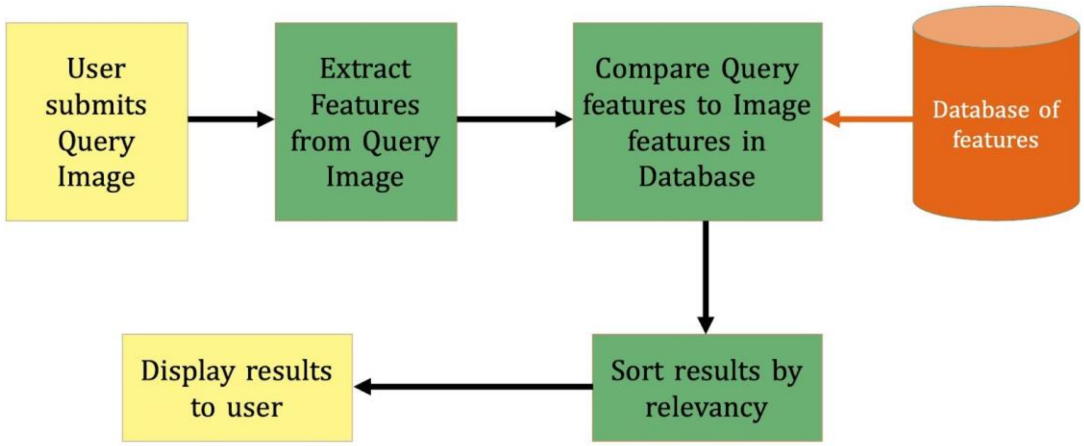
Defining the similarity metric: Now that we have a bunch of feature vectors. How do we compare them? Popular choices include the Euclidean distance, Cosine distance, and chi-squared distance, but the actual choice is highly dependent on (1) your dataset and (2) the types of features you extracted.

Searching: The final step is to perform an actual search. A user will submit a query image to your system (from an upload form or via a mobile app, for instance) and the job will be to (1) extract features from this query image and then (2) apply your similarity function to compare the query features to the features already indexed. From there, we simply return the most relevant results according to your similarity function.

Architecture and Workflow

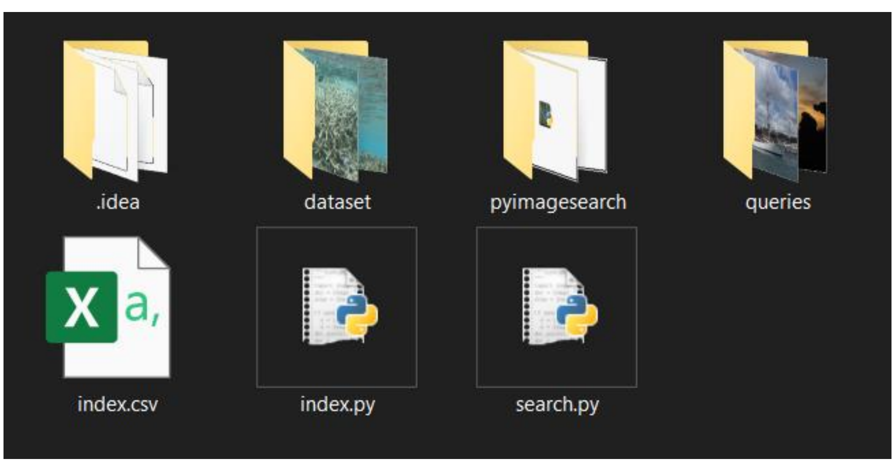
Feature Extraction:

Performing search on CBIR:

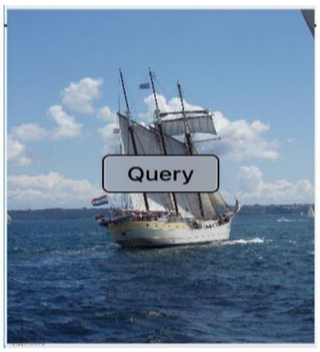


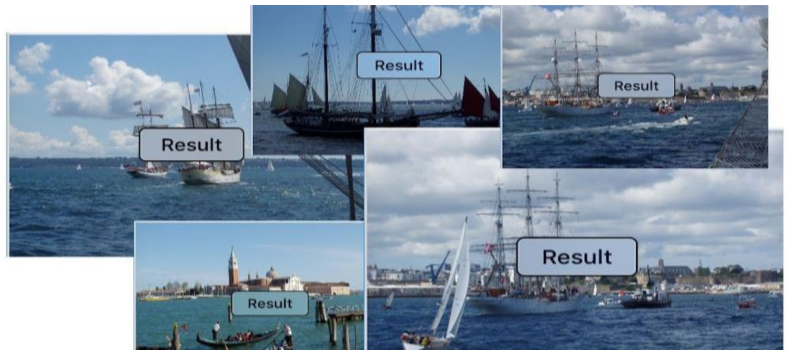
Execution steps:

- Run: python index.py --dataset dataset --index index.csv (Create indexing)
- After it is executed an index.csv file is generated
- Run: python search.py --index index.csv --query queries/image_name --result-path dataset
- The figure below shows the hierarchy of the folders. The image is downloaded from the dataset.
- The query images are stored in queries.
- The project folder contains the colordescriptor.py and searcher.py files.



Results





Conclusion

We utilized a color histogram to characterize the color distribution of the photos. Then, we indexed our dataset using our color descriptor, extracting color histograms from each of the images in the dataset. To compare images, we utilized the chi-squared distance. From there, we implemented the necessary logic to accept a query image and then return relevant results.

References

[1] Herve Jegou, Matthijs Douze and Cordelia Schmi, "Hamming Embedding and Weak geometry consistency for large scale image search" Proceedings of the 10th European conference on Computer vision, October, 2008

[2] https://courses.cs.washington.edu/courses/cse576/07sp/notes/CBIR_white.pdf

[3] Gaurav Jaswal (student), Amit Kaul, “Content Based Image Retrieval’’ – A Literature Review National Institute of Technology, Hamirpur-177001, Himachal Pradesh (India)2009

[4] Yong Rui and Thomas S. Huang, “Image Retrieval: Current Techniques” Promising Directions, and Open Issues January 7, 1999

[5] Greg Pass, Ramin Zabih, “Histogram Refinement for Content-Based Image Retrieval” Computer Science Department Cornell University

CBVR

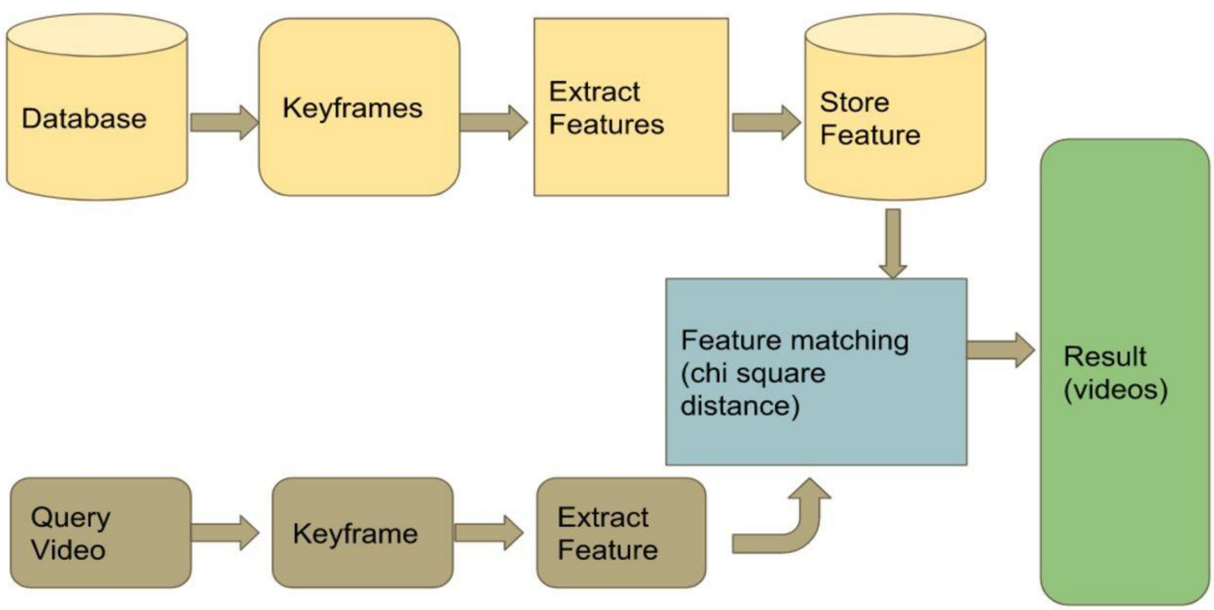
Motivation:

With increase in digital data, it is hard to scan all the data for desired output and hard to give file name uniquely. So, content of the data gives as accurate information about the file and with the help of content analysis we can easily access the desired file within less time, space and with a smaller number of calculations.

Objective:

To develop a video Retrieval system first we need to find a representative frame of video we can call it keyframe which contains all the essential information of the video and stores it. we know that video files are large takes more times to process it, so keyframe are used to minimize the process time. By using the keyframe we can apply content-based image retrieval method to retrieve similar video. So, for the given query video, extract keyframe from it than extract feature of the keyframe match it with previously stored features of the keyframes of the database videos.

Methodology:



Algorithm:

keyframe Extraction Method:

Method 1:

1. Extract all frames of the video.
2. Convert each frame to grayscale image
3. Find average intensity of the frame
4. Find consecutive difference of average intensity of frame
5. Draw a graph of the differences
6. Find the frame with highest and lowest difference in graph as key frame

Limitations:

It gives good results in case of still camera and single object video. For more than one object motion two keyframes are not sufficient. To increase the keyframes we can apply threshold to the difference in average intensity.

Method 2:

a single visual descriptor cannot capture all the pictorial details needed to estimate the changes in the visual content of frames and the visual complexity of a video shot. We must take into account both color properties and structural properties, such as texture. In this algorithm we have selected color histograms, edge direction histogram and Multiresolution wavelet analysis.

Color histograms are frequently used to compare images because they are simple to compute and tend to be robust regarding small changes in camera viewpoint.

Two Sobel filters are applied to obtain the gradient of the horizontal and edges of the luminance frame image.

Multiresolution wavelet analysis provides representations of image data in which both spatial and frequency information are present

Distance between two color histograms

$$d_H(H_t, H_{t+1}) = 1 - \sum[\min(H_t(j), H_{t+1}(j)]$$

Where H_t and H_{t+1} is color histogram of F(t) and F(t+1)

The difference between two edge direction histograms (d_D)

$$d_D(D_t, D_{t+1}) = \sqrt{(\sum(D_t(j) - D_{t+1}(j))_2}$$

The difference between two edge direction wavelet statistics (d_w)

$$d_D(W_t, W_{t+1}) = \sqrt{(\sum(W_t(j) - W_{t+1}(j))^2}$$

The three resulting values (to simplify the notation we have indicated them as d_H, d_W and d_D only) are mapped into the range [0, 1] and then combined to form the final frame difference measure (d_{HWD}) as follows:

$$d_{HWD} = d_H d_W + d_W d_D + d_D d_H$$

For Keyframe Selection dynamically selects the representative frames by analyzing the complexity of the events depicted in the shot in terms of pictorial changes. Draw a graph of the cumulative frame difference by frame difference previously obtained. Sharp slopes indicate significant changes in the visual content due to a moving object, camera motion or the registration of a highly dynamic event. Find high curvature point.
Select a frame as midpoint between two consecutive high curvature points as keyframe for the video.

Features Extracted:

1. Color Feature:
    - convert the image to the HSV color space
    - grab the dimensions and compute the center of the image

- divide the image into four rectangles/segments (top-left, top-right, bottom-right, bottom-left)
- construct an elliptical mask representing the center of the image
- loop over the segments
  - construct a mask for each corner of the image, subtracting the elliptical center from it
  - extract a color histogram from the image, then update the feature vector.

- extract a color histogram from the elliptical region and update the feature vector
- return the feature vector

2. ORB (Oriented FAST and Rotated BRIEF):

ORB if Fast key point detector and BRIEF descriptor. It takes the descriptor of one feature in first set and is matched with all other features in second set using some distance calculation. And the closest one is returned.

Similarity Measure:

To find similarity between two images we have to find feature vector and measured the similarity between the features

1. Chi-squared

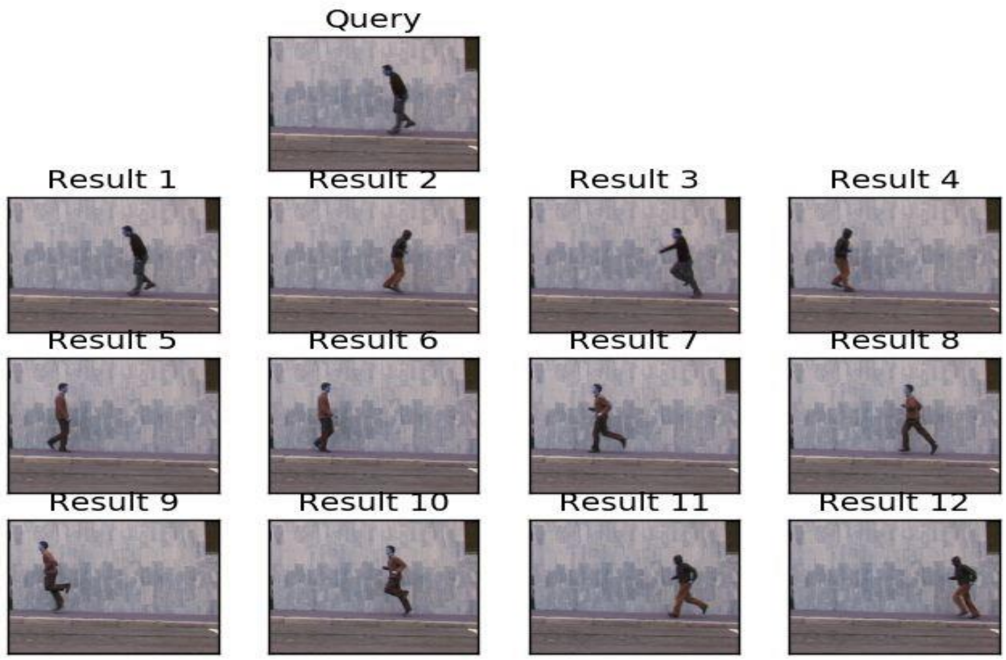
d(x,y) = sum((xi-yi)^2 / (xi+yi)) / 2

Images that have a chi-squared similarity of 0 will be deemed to be ***identical*** to each other. As the chi-squared similarity value increases, the images are considered to be ***less similar*** to each other.

2. cv2.BFMatcher(): Brute-Force matcher, it takes the descriptor of one feature in first set and is matched with all other features in second set using some distance calculation. And the closest one is returned.

Results:

Keyframe similar to the query video.



The result is improved using new keyframe method as compare to tradition keyframe extraction method. By using this method, we can overcome the limitation of the previous method. Clearly new approach is better but can be improved by some motion detection or Object movement. ORB feature is more efficient than simple color feature. With enough number of inputs data, we can apply machine learning algorithm to get more accurate result.

Contribution

name	code	Role
Mostafa Mohamed Badawi Farhat	**16X0127**	Feature Extractor, Similarity metrics & CBVR
Ahmed Hamed Ahmed Abdallah	**16W0002**	Feature Extractor, Similarity metrics & CBVR
Ahmed Mohamed Fahmy Aly	**15X0015**	Image Descriptor, Indexing, Search, Report
Salah Soliman El-Sayed Soliman	**15T0510**	Image Descriptor, Indexing, Search, Presentation

References:

1. G. Ciocca and R. Schettini, "An innovative algorithm for keyframe extraction in video summarization," Journal of Real-Time Image Pro- cessing (Springer), vol. 1, no. 1, pp. 69–88, 2006.

2. C. Sujatha and U. Mudenagudi, "A Study on Keyframe Extraction Methods for Video Summary," *2011 International Conference on Computational Intelligence and Communication Networks*, Gwalior, 2011, pp. 73-77

3.Y. Yang, B. C. Lovell and F. Dadgostar, "Content-Based Video Retrieval (CBVR) System for CCTV Surveillance Videos," 2009 Digital Image Computing: Techniques and Applications, Melbourne, VIC, 2009, pp. 183-187.

4. *W. Wolf, "Key frame selection by motion analysis," 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings, Atlanta, GA, 1996, pp. 1228-1231 vol. 2. doi: 10.1109/ICASSP.1996.543588*

5. *Idris, F., Panchanathan, S.: Storage and retrieval of compressed images using wavelet vector quantization. J. Vis.Lang. Comput. 8, 289–301 (1997)*