ECE 515 - Computer Vision and Image Analysis II

Content-based Image Retrieval System (CBIR)

Group 12

Student Names

Aayushi Agarwal	Divya Swaroopa Rani Ogirala	Pavan Kumar Srikanth Naik
(6699402393)	(658601058)	(669940624)

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 (QBIC) and content-based visual information retrieval (CBVIR), is the application of computer vision techniques to the image retrieval problem, that is, the problem of searching for digital images in large databases.

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.

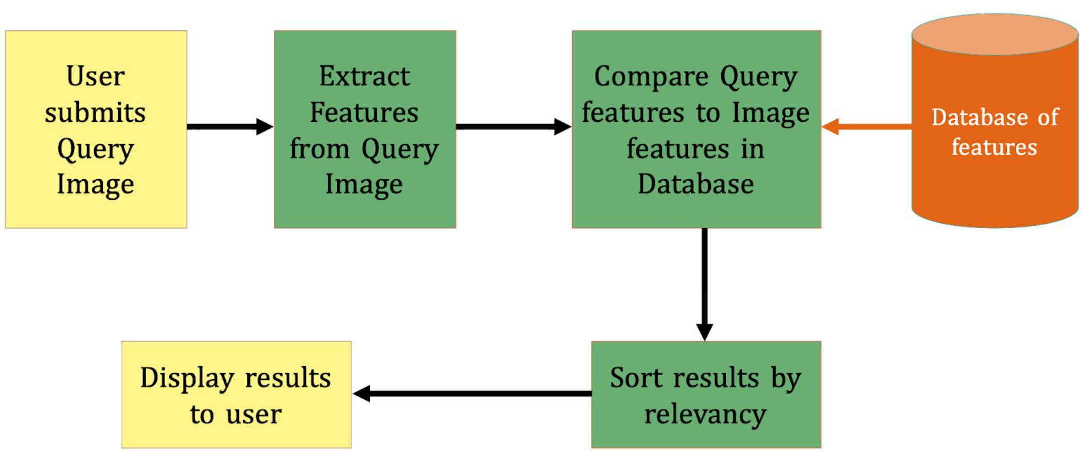
Defining the similarity metric: Now that we have a bunch of feature vectors. How do we compare them? Popular choices include the Euclidean distance, Cosine distance, and chi-squared distance, but the actual choice is highly dependent on (1) your dataset and (2) the types of features you extracted.

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

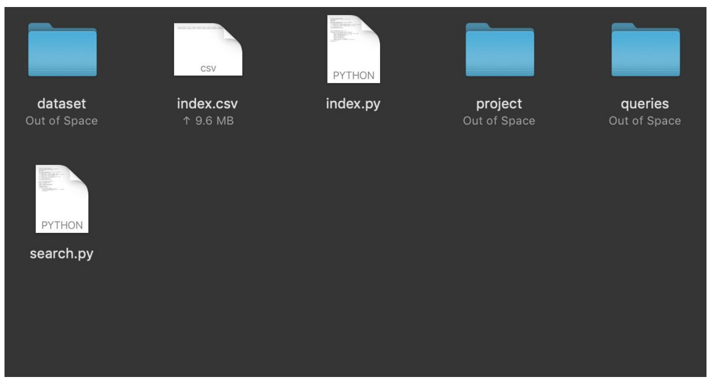
Feature Extraction:

Performing search on CBIR:

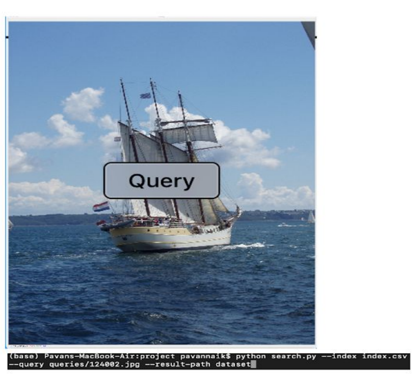


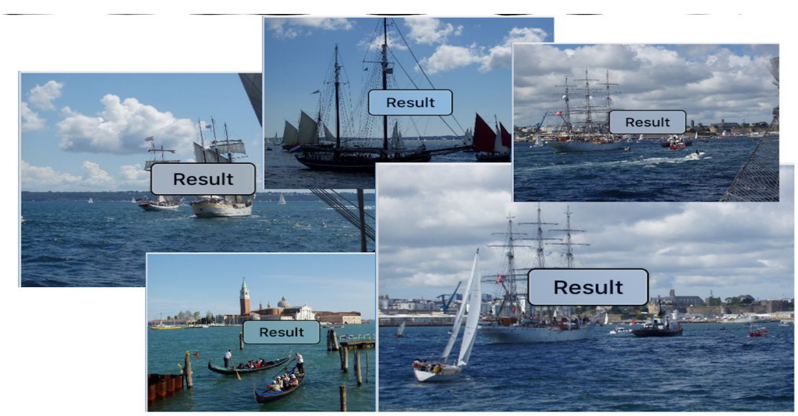
Execution steps:

- 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.

Contribution

- Aayushi Agarwal – Image Descriptor, Indexing, Presentation
- Divya Swaroopa Rani Ogirala - Similarity metrics, Report, Presentation
- Pavan Kumar Srikanth Naik – Feature Extractor, Search, Presentation

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