International Journal of Interdisciplinary Innovative Research &Development (IJIIRD)
ISSN: 2456-236X
Vol. 05 Special Issue 01 | 2020

Image Retrieval Based on Region Based Similarity Measure Using CBIR Technique

[1]Sandhya Shinde, [2]Rachit Jain, [3]Sumit Yadav, [4]Irfan Minsurwale.

[1]Assistant Professor, Dept. of Electronics & Tele-Communication Engineering, Dr. D.Y. Patil Institute of Engineering, Management and Research, Maharashtra, India.
[2,3,4] Student, Dept. of Electronics & Tele-Communication Engineering, Dr. D.Y. Patil Institute of Engineering, Management and Research, Maharashtra, India.

ABSTRACT

With many potential practical applications, content- based image retrieval (CBIR) has attracted substantial attention during the past few years. A variety of relevance feedback (RF) schemes have been developed as a powerful tool to bridge the semantic gap between low-level visual features and high-level semantic concepts, and thus to improve the performance of CBIR systems. Among various RF approaches, support-vector-machine (SVM)-based RF is one of the most popular techniques in CBIR. Despite the success, directly using SVM as an RF scheme has two main drawbacks. First, it treats the positive and negative feedbacks equally, which is not appropriate since the two groups of training feedbacks have distinct properties. Second, as the size of image database increases search and retrieval become slow and it affects the performance of the system. To explore solutions to overcome these two drawbacks, CBIR system is implemented using RBIR and which make use of the properties of images.

Key Words: Content-based image retrieval (CBIR),Support Vector Machine(SVM),Region Based Image Retrieval(RBIR).

1. INTRODUCTION

During the past few years, content-based image retrieval (CBIR) has gained much attention for its potential applications in multimedia management . Content-based image retrieval, a technique which uses visual contents to search images from large scale image databases according to user interests, has been an active and fast advancing research area since the 1990s. Content-based image retrieval, also known as query by image content (QBIC).It is motivated by the explosive growth of image records and the online accessibility of remotely stored images. An effective search scheme is urgently required to manage the huge image database. Different from the traditional search engine, in CBIR, an image query is described by using one or more example images, and low-level visual features (e.g., color, texture, shape, etc.) are automatically extracted to represent the images in the database. However, the low-level features captured from the images may not accurately characterize the high-level semantic concepts. To reduce the inconsistency problem, the image retrieval is carried out according to the image contents; such strategy is called content-based image retrieval. In Content-Based Approach, Images can be search based on visual features, such as color, texture, and edge information shown in fig 1 .

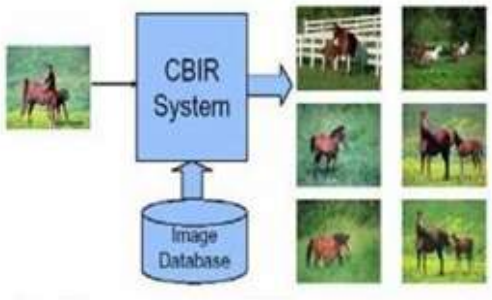

Fig.1. Content based image retrieval

To narrow down the so-called semantic gap, relevance feedback (RF) was introduced as a powerful tool to enhance the performance of CBIR [4]. A self-organizing map was used to construct the RF algorithms. In one-class support vector machine (SVM) estimated the density of positive feedback samples[3]. Derived from one-class SVM, a biased SVM inherited the merits of one-class SVM but incorporated the negative feedback samples. Considering the geometry structure of image low-level visual features, and proposed manifold-learning-based approaches to find the intrinsic structure of images and improve the retrieval performance. With the observation that *"all positive examples are alike; each negative example is negative in its own way*," RF was formulated as a biased subspace learning problem, in which there is an unknown number of classes, but the user is only concerned about the positive class.

Figure 2: Typical set of positive and negative feedbacks samples in RF iteration.

SVM RF approaches ignore the basic difference between the two distinct groups of feedbacks, i.e., all positive feedbacks share a similar concept while each negative feedback usually varies with different. A typical set of feedback samples in RF iteration is shown in fig 1. Traditional SVMRF techniques treat positive and negative feedbacks equally. Directly using SVM as an RF scheme is potentially damaging to the performance of CBIR systems. One problem stems from the fact that different semantic concepts live in different subspaces and each image can live in many different subspaces, and it is the goal of RF schemes to figure out "which one". However, it will be a burden for traditional SVM-based RF schemes to tune the internal parameters to adapt to the changes of the subspace. Such difficulties have severely degraded the effectiveness of traditional SVM RF approaches for CBIR. This problem overcomes by implementing CBIR as both offline and online.

Before we get too in-depth, let's take a little bit of time to define a few important terms. When building an image search engine we will first have to ***index our dataset***. Indexing a dataset is the process of quantifying our dataset by utilizing an ***image descriptor*** to extract ***features*** from each image.

An ***image descriptor*** defines the algorithm that we are utilizing to describe our image.

For example:

- The mean and standard deviation of each Red, Green, and Blue channel, respectively,
- The statistical moments of the image to characterize shape.
- The gradient magnitude and orientation to describe both shape and texture.

The important takeaway here is that the ***image descriptor governs how*** the image is quantified. ***Features***, on the other hand, are the output of an ***image descriptor***. When you put an image into an image descriptor, you will get ***features*** out the other end.

In the most basic terms, ***features*** (or ***feature vectors***) are just a list of numbers used to abstractly represent and quantify images. Take a look at the example figure below:

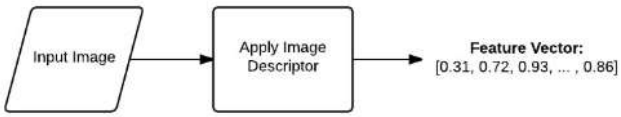

Figure 3: The pipeline of an image descriptor. An input image is presented to the descriptor, the image descriptor is applied, and a feature vector (i.e a list of numbers) is returned, used to quantify the contents of the image.

2. METHODOLOGY

Defining your image descriptor: At this phase you need to decide what aspect of the image you 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 your 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 your similarity metric: Cool, now you have a bunch of feature vectors. But how are you going to compare them? Popular choices include the Euclidean distance, Cosine distance, and chi-squared distance, but the actual choice is highly dependent on your dataset and 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 your 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, you simply return the most relevant results according to your similarity function.

Again, these are the most basic 4 steps of any CBIR system. As they become more complex and utilize different feature representations, the numbers of steps grow and you'll add a substantial number of sub-steps to each step mentioned above. But for the time being, let's keep things simple and utilize just these 4 steps.

Let's take a look at a few graphics to make these high-level steps a little more concrete. The figure below details Steps 1 and 2:

Figure 4: A flowchart representing the process of extracting features from each image in the dataset.
We start by taking our dataset of images, extracting features from each image, and then storing these features in a database.

We can then move on to performing a search (Steps 3 and 4):

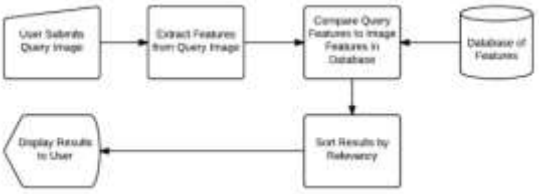

Figure 5: Performing a search on a CBIR system. A user submits a query, the query image is described, the query features are compared to existing features in the database, results are sorted by relevancy and then presented to the user.

First, a user must submit a query image to our image search engine. We then take the query image and extract features from it. These "query features" are then compared to the features of the images we already indexed in our dataset. Finally, the results are then sorted by relevancy and presented to the user.

3. THE GOAL

Our goal here is to build a personal image search engine. Given our dataset of vacation 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. For instance, if I submit a photo of sail boats gliding across a river, our image search engine should be able to find and retrieve our vacation photos of when we toured the marina and docks. Take a look at the example below where I have submitted an photo of the boats on the water and have found relevant in image over vacation photo collection.

Figure 6

In order to build this system, we'll be using a simple, yet effective image descriptor: **the color histogram**. By utilizing a color histogram as our image descriptor, we'll be we'll be relying on the *color*

distribution of the image. Because of this, we have to make an important assumption regarding our image search engine:

Assumption: Images that have similar color distributions will be considered relevant to each other. Even if images have dramatically different contents, they will still be considered "similar" provided that their color distributions are similar as well.

This is a really important assumption, but is normally a fair and reasonable assumption to make when using color histograms as image descriptors.

4. CONCLUSIONS

In this paper we explored how to build an image retrieval engine to make our vacation photos search-able.
We utilized a color histogram to characterize the color distribution of our 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, a popular choice when comparing discrete probability distributions.
From there, we implemented the necessary logic to accept a query image and then return relevant results.
The proposed system tries to fulfil the following objectives:
1] To narrow down the searching space in the similarity computation step and to enhance the retrieval speed, we add classification procedure into our method.
2] Considering the tough work of the representative samples selection, we adopt SSL to decrease the workload of choosing train set.
3] Due to the inevitable misclassification, a classification error recovery scheme is presented here to reduce the influence to the retrieval behavior.

5. REFERENCES

[1] Licheng Jiao, "SAR Images Retrieval Based on Region-Based Similarity Measure for Earth Observation" IEEE journal of selected topics in applied earth observations and remote sensing, VOL. 8, NO. 8, AUGUST 2015.
[2] Daniela Espinoza-Molina, "Earth-Observation Image Retrieval Based on Content, Semantics, and Metadata" Manuscript received September 28, 2012; revised January 11, 2013 and March 20, 2013; accepted April 15, 2013.
[3] Xu Tang, "SAR Image Content Retrieval Based on Fuzzy Similarity and Relevance Feedback" Manuscript received May 10, 2016; revised October 19, 2016 and January 22, 2017; accepted February 1, 2017
[4] Xu Tang," Fusion Similarity-Based Re-ranking for SAR Image Retrieval" Manuscript received July 4, 2016; revised November 25, 2016; accepted December 3, 2016.