Content Based Image Retrieval (CBIR) - Histogram based descriptors
PIV - Prog1

Gil Jiménez Canellas
gil.jimenez@estudiantat.upc.edu

Gerard Agustina Ribera
gerard.agustina@estudiantat.upc.edu

Abstract

In this report, we present a comprehensive study focused on the utilization of a Content-Based Image Retrieval (CBIR) algorithm, which characterizes images through histograms. Content-Based Image Retrieval [3] involves the application of computer vision techniques to address the challenge of retrieving digital images from extensive databases. This study specifically investigates the algorithm's efficacy when utilizing histogram-based descriptors. Our evaluation encompasses the overall performance of the image search system, analyzing quantitative metrics including the F-measure and computational cost. Through this analysis, we aim to provide insights into the algorithm's effectiveness and efficiency in practical image retrieval scenarios.

I. INTRODUCTION

The primary aim of this report is to present the outcomes of a practical assignment dedicated to the development of a Content-Based Image Retrieval (CBIR) system. This system was devised to function within a database housing 2,000 images, each representing 500 distinct objects. The objective is to identify and retrieve the four images within the database that correspond to the same object depicted in the input image provided to the system.

The practical assignment entailed the development of a CBIR system utilizing MATLAB, leveraging image histograms for analysis. Subsequently, the system underwent rigorous testing using a series of query images. The retrieved images were then compared against the anticipated output to assess the system's performance and efficacy.

II. OVERALL SYSTEM DESCRIPTION

The system processes an input file named "input.txt," which contains the name or names of the images to be analyzed. For each image listed in the input file, the system executes the following steps:

Loads the matrix H and extracts the histogram h from the current image. Calculates the distance vector d between h and all histograms in H. Then, identifies the 10 smallest values in the d vector along with their respective indices, which are retained for later use in writing the 10 image names to the output file named "output.txt".

The algorithm conducts a similarity search for each image listed in the input file, comparing its histogram with all histograms in the dataset and selecting the 10 images with the smallest distances. Subsequently, the chosen images are recorded in the output file.

Algorithm 1: CBIR System

```
Input: input.txt
Output: output.txt
foreach img in input.txt do
    h = Histogram(img);
    for i ← 1 to size(H) do
        d_i = distances(h, H[i]);
    [distances, indexes] =sort(distances);
    for i ← 1 to 10 do
        score = score(distances(i));
        nameImg = searchDataSet(indexes(i));
        write(output.txt, nameImg);
return output.txt
```

Distances

We utilized three distinct metrics to compute the dissimilarity between histograms: the Mean Squared Error (MSE), the Bhattacharyya distance, and the Chi-Squared distance.

The MSE is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(h_i - g_i)^2 \tag{1}$$

where h_i and g_i represent the i-th bins of the two histograms being compared, and n is the total number of bins.

The Manhattan Distance (MD) is defined as:

$$\text{MD} = \frac{1}{n}\sum_{i=1}^{n}|h_i - g_i| \quad (2)$$

which is similar to the MSE, but taking the absolute value.

The Chi-Squared distance [2] is defined as:

$$\chi^2(h,g) = \sum_{i=1}^{n}\frac{(h_i - g_i)^2}{h_i + g_i + \epsilon} \quad (3)$$

where h_i and g_i represent the i-th bins of the two histograms, and ϵ is a small term introduced to prevent division by zero.

Finally, the Bhattacharyya distance [1] is given by:

$$B(h,g) = -\ln\sum_{i=1}^{n}\sqrt{h_i g_i} \quad (4)$$

where h_i and g_i are the probabilities of the i-th bin in each histogram.

III. RESULTS

We will begin by assessing the system's performance using the input file provided by the lab professor. To accomplish this, we'll generate a Recall-Precision graph and compute the F-measure. This evaluation will facilitate a comparison of the system's performance across the four distinct measures.

The performance curves for the Chi-squared and Bhattacharyya distances are similar, while the Mean Squared Error and Manhattan Distance curves shows significantly worse performance. This discrepancy arises because both the Chi-squared and Bhattacharyya distances measure the similarity between two probability distributions. This metric is particularly advantageous in image retrieval applications, where images are often represented as histograms.

On the other hand, the MSE and MD measures the mean square difference between two continuous variables and is less suitable for comparing histograms. Consequently, this two approaches exhibits poorer performance compared to the other two distances.

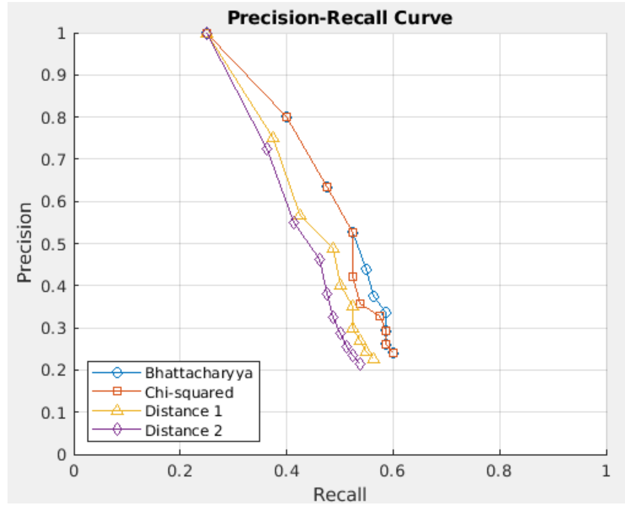

Figure 1: Precision-Recall Curve

Metric	F-score	Runtime
MD	0.4742	0.54283 seconds
MSE	0.4833	0.58936 seconds
Chi-squared $\chi^2(h,g)$	0.5429	0.57899 seconds
Bhattacharyya $B(h,g)$	0.5429	0.52934 seconds

Table 1: Comparison of Metrics

Based on the outcomes, it's evident that the χ^2 and Bhattacharyya distances exhibit similar performance, boasting an F-score of 0.5429, marginally surpassing the F-score of MSE and MD at 0.4833 and 0.4742. Nevertheless, it's essential to acknowledge that there is no significant difference in execution time between the different distance measurements. Hence, the F-score should be the chosen metric for evaluating performance.

IV. CONCLUSIONS

It's important to highlight that the findings presented herein are derived from a specific experiment, focusing on the images within the input.txt dataset. These results may not be universally applicable, warranting cautious interpretation. Further experimentation with alternative input.txt datasets is advisable to validate our conclusions.

The optimal system configuration, excluding runtime considerations, indicates a preference for utilizing the Bhattacharyya distance. However, it's essential to acknowledge potential variability in performance when applied to different input.txt datasets. Relying solely on one dataset may lead to overfitting the system to specific data, compromising its generalizability.

In conclusion, the identified optimal system configuration is contingent upon the specifics of our experiment and may not readily generalize to other contexts. Further experimentation with alternative input files is warranted to validate our conclusions thoroughly.

Improvements

Exploring avenues for improvement is imperative. One potential enhancement could involve integrating color histograms to mitigate the information loss inherent in grayscale image processing. Additionally, investigating alternative algorithms beyond histograms, which lack spatial or geometric information, could yield valuable insights in addressing the problem at hand.

References

[1] Authorless. Bhattacharyya distance. `https://en.wikipedia.org/wiki/Bhattacharyya_distance`.

[2] Authorless. Chi-squared distance. `https://en.wikipedia.org/wiki/Chi-square`.

[3] Authorless. Content based image retrieval. `https://en.wikipedia.org/wiki/Content-based_image_retrieval`.