Content-Based Image Retrieval System

CS60092 Term Project Report

ABSTRACT

To develop an efficient and effective system that can retrieve relevant images from a large database based on their visual content, such as colour, texture, shape and spatial layout. The main challenge is to bridge the semantic gap between low-level features extracted from images and high-level concepts that users are searching for. The ultimate aim of CBIR is to provide an intuitive and efficient way for users to access and retrieve the information they need from large image databases.

1. MOTIVATION

CBIR has numerous applications in areas such as image and video search, medical imaging (can be sued to analyze medical images and doctors to make more accurate diagnoses), surveillance (to identify and track individauls or objects of interest) etc. Moreover, with the exponencial growth of digital images, it has become increasingly difficult to search and retrieve imges manually. CBIR systems can help to automate the process and it faster and more efficient.

Overall, the motivation for the project is to improve the efficiency, accuracy and usability of image retrieval systems (baseline explained later).

2. LITERATURE REVIEW

CBIR has been a popular research area for several years and numerous studies have been conducted to improve the efficiency of CBIR systems. IBM developed the first commercial version of CBIR system naming QBIC (Query By Image Content)[2] in 1995. Bag-of-Visual-Words (BoVW)[3] model integrates visual words with local intensity order pattern (LIOP) feature and local binary pattern variance (LBPV) feature to reduce the semantic gap issue and enhance CBIR performance. Scale Invariant Feature Transform (SIFT)[4] model works on the basis of visual words fusion of and local intensity order pattern descriptors. These include few models proposed before the introduction to Deep Learning Neural Networks.

After the introduction and evolution of DCNN's, the performance of CBIR has got a boost, because by the help of deep models we can finally extract higher-level features along with the low-level features from the image to reduce the semantic gap. Back-propagation Feedforward Neural Network (BFNN)[5] can be used for classification in CBIR after exploiting some features of images e.g. geometric, colour and texture. As a model architecture goes into depth, it starts to learn high-level features from the low-level features. Our baseline model[1], CBIR Using Features Derived by Deep Learning, these higher-level features for the feature representation of images are used. So, the last softmax activation layer used for calculating probabilities of each class is removed and the preceding fully connected layer is selected to be feature vector representation for CBIR.

3. FEATURES ADDED AND TECHNIQUES

3.1 Similarity Measure

Our model uses cosine similarity between the query image vector and database image vector as a measure of similarity, replacing the siple Euclidean Distance metric in the baseline model. Result is the rise in avg. precision by around 17.5%, run on CorelDB database. See fig. 1. Here, cosine similarity includes more complex operations which implies a bit higher time complexity. Hence, there is a tradeoff between precision and retrieval time.

Category	**Euclidean Distance**	**Cosine Similarity**
African People	79.35	93
Beach	96.6	100
Building	93.55	96
Bus	100	100
Dinosaurs	100	100
Elephant	100	100
Flower	97.25	98
Horse	99.9	100
Mountain	98.95	100
Food	95.55	98.25

Fig 1. Average Precision (%) using Eyclidean Distance vs Cosine similarity metric

3.2 Rotational Invariacy

Often if we try to retrieve similar images given a same query image but with different orientation angles at every time, the retrieval results will change significantly. Our algorithm allows invariant retrieval for query images rotated at different angles.

For any given query image, we consider three more rotational variants (say 90°, 180°, 270°). The similarity measure with the query image for each image in the database shall be the maximum score among the four similarity measure values obtained with all four rotational variants. This way, we retrieve the images that match the highest with various orientations of query image. This approach has a drawback. There is a possibility of an image matching with an unintended variant and can be shown relevant. But this is a very rare situation which we have never encoutered during our experiments.

3.3 Image Clustering

Our model is quite fast, however, keeping in mind the real time speed requirements for retrieval from very large databases, the concept of pre-clustering is introduced. With this, clustering of the database images is done allowing search for images only within a specified cluster.

The technique used is explained step by step below.

i. First, we calculate the last dense layer feature extraction and also calculate the probabilities of assigning to each of the 1000 pre-specified classes for each of the images in the database.

ii. Now according to the probabilities, we assign the each of images to the top 5 classes. We also save the last dense layer feature values for each image in the database in memory.

iii. At the time of retrieval, we calculate both the last dense layer feature values and class probabilities for the query image as well and assigned the query image to top 5 classes based on the probabilities.

iv. We accumulate all the images which has any of these classes in their top five classes and calculate similarity measure with the last dense layer feature dimension and retrieve the similar images with only in the accumulated images.

v. This gives us a much fast retrieval with respect to the previous method as now we are searching relevant in only a small subset of the whole database (30607 images) instead of searching in the whole 30607 images. For searching in the top 5 classes the average number of images to be searched for any image retrieval becomes only 1568.

4. EXPERIMENTS

1. Model run with CorelDB database with cosine similarity as similarity metric.
   Results are shown in fig. 1.
2. Model run including and excluding rotational invariancy on CoralDB database over rotated images of various categories.
   Results include:
   Dinosaurs: no effect
   Buses: no effect
   Elephant: without is 73% avg. precision;
   With is 92% avg. precision.
3. Model run including rotational invariancy on CoralDB database over erect images of various categories.
   Result:
   Dinosaurs: no effect
   Buses: no effect
   Others: slight decrement in avg. precision.
4. Model run including rotational invariancy on Caltech database over erect images of various categories.
5. Model run including Image Clustering on CoralDB and Caltech database.
   Result: couldn't differetiate run time (given size of the databases but shows significant results in real-time retrieval) and slight decrement in avg. precision.

5. FUTURE SCOPE

The performance can be greatly improved for a specific dataset by introducing the user feedback which is called as Relevance Feedback. Relevance Feedback is basically the feedback from users after each retrieval regarding which results are relevant to the query images and which are not. Using this feedback the CBIR system will start learning and will improve the result gradually. Implementing Reinforcement Learning techniques is a bsic step for this improvement.

A query image often contains more than one object or a dominating background taht might interfere with retrieval process. To avoid this, the model should learn to identify the foreground object and prioritising them incase of presence of more than one significant object.