A toy system for Content based Image Retrieval

Jiashuo Yu
Fudan University

Abstract

Content based image retrieval (CBIR) is the application of computer vision techniques to image retrieval problem, which is the problem of searching digital images in large databases. Opposed to traditional concepted-based approaches, CBIR analyzes the contents of the image rather than the metadata associated with the image. In this toy system, we use deep learning method, which is based on a convolutional neural network to extract image features with GPU acceleration, and implement image retrieval by comparing features we extracted.

1.Introduction

Content based image retrieval, also known as Query by image content (QBIC), are based on the application of computer vision techniques to the image retrieval problem in large databases. Its goal is to achieve retrieving the most visually similar images to a given image from a database of images [1]. Content based image retrieval task can be separated as two subtasks: feature extraction and query images.

Feature extraction is to extract images features by using specific method. Since the introduction of SIFT feature by Lowe[2], we can easily identify objects and detect local features in images by two steps: key-point localization and feature description. With the explosive research on deep neural network, many researchers used convolutional models to solve image retrieval problems, e.g. Babenko et al. focus on landmark retrieval and fine-tune the pre-trained CNN model on ImageNet with the class corresponding to landmarks and achieved obvious model performance[3][4].

The key problem of query images is to measure the similarity between images. Since features extracted are tensors, we can measure their similarity by computing Euclidean distance in Eq.1 or cosine similarity in Eq.2.

$$\boldsymbol{\rho} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

$$\boldsymbol{\cos(\theta)} = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1{}^2 + y_1{}^2} \times \sqrt{x_2{}^2 + y_2{}^2}} \quad (2)$$

2.Method

In this system, I use Caltech-256 as dataset and pre-trained VGG-16 model on ImageNet to extract features. As for query images, I use Euclidean distance to compute similarity score between the feature of query image and feature of each image in Caltech-256 database.

To evaluate the performance of this toy system, I use mean Average Performance(mAP) as evaluation metric and summarize the time complexity of this toy system.

3.Experiment

Image Preprocessing The image size of dataset Caltech-256 is not uniform, thus I resize them into 256x256 and read images by OpenCV module cv2. Then I transform images into NumPy array and convert array to the shape of (D,W,H). To normalize images, each channel of image is divided by 255, subtracted by their mean value and divided by their standard variance. To simplify the code, I use the mean value and standard variance of ImageNet as a substitution

Feature Extraction To extract the feature, I apply the pre-trained vgg16 pretrained model as my own model while the last layer of vgg16 network is discard, then I put the preprocessed image directly into the pre-trained model to get their feature. Model is transformed to GPU mode in order to accelerate extraction.

Batch Extraction Feature extraction is a time-consuming task, in order to reduce image-query response time and lower time complexity, batch extraction is essential. Therefore, I use a for-loop to store the feature of each image into a h5py file, in this way, each image will be only extracted once and we only need to extract the query image and make feature comparison in practice, which significantly speeds up the entire process.

Image Query As mentioned above, the key problem of image query is similarity computation. Features of database image was extracted and stored into a h5py file before, thus we need to put these features into a list that has the same sequence as the picture list. Then I take the same algorithm to preprocess the query image and extract its feature, after which Euclidean distances between query image feature and each image feature in the database are computed and stored in a dictionary. Finally, the dictionary is sorted and the minimized n pictures are plotted(n is the picture number user want to search)

4. Evaluation

4.1 Mean Average Precision(mAP)

To compute mean Average Precision(mAP), I choose the first image of each category in the Caltech-256 dataset as query image, and compute their average precision respectively, after which I compute their mean value as mAP, which is 0.635.

4.2 Time Efficiency

This content-based image retrieval toy system is consisted of two main algorithms: Feature Extraction and Similarity Computation, thus the time complexity is determined by the slower of these two algorithms above. For the feature extraction, each image is processed only once and their features are all stored in a certain h5py file, which means the time complexity is O(n), where n is the size of whole image dataset. As for the similarity computation, each query image will be extracted first, then their feature will be compared with every single element in the h5py file, which contains features of the whole dataset. Therefore, time complexity of Similarity Computation will also be O(n). In practice, we clearly see that the feature-extraction algorithm costs far more time than the similarity-computation algorithm. In fact, the feature-extraction algorithm on Caltech-256 dataset can cost hours even with GPU acceleration while the image query can be done in few seconds. Therefore, the time efficiency is largely determined by the feature-extraction algorithm, which will be repeated the number of dataset-contained images times.

5. Conclusion

In conclusion, a content-based image retrieval toy system is implemented, which consists of two main algorithms: Feature Extraction and Image Query. In feature extraction, VGG-16 pre-trained network on ImageNet is applied. In image query, feature of each query image is extracted by feature-extraction algorithm, which is sent to a similarity computation algorithm using Euclidean distance for quantification later.

Concretely, a few optimization methods are used: H5py file is used for batch extraction which significantly reduced time complexity, GPU computation is used for acceleration. Generally, this system gets performance with a mAP evaluation of 0.635.

But this toy system remains lots of defects: This toy system has no UI system since I am poorly on web UI framework. Besides, there are too many redundant codes which can be optimized to a relatively simple and more efficient version, which is caused by my poor coding capability. I hope I could continue to improve myself and make my project better in my future learning and coding process.

Reference

[1] Maria Manuela, Cruz-Cunha, Handbook of Research on ICTs and Management Systems for Improving Efficiency in Healthcare and Social Care

[2] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. International journal of computer vision, 2004, 60(2): 91-110.

[3] Zhou W, Li H, Tian Q. Recent advance in content-based image retrieval: A literature survey[J]. arXiv preprint arXiv:1706.06064, 2017.

[4] Babenko A, Slesarev A, Chigorin A, et al. Neural codes for image retrieval[C]//European conference on computer vision. Springer, Cham, 2014: 584-599..