COMP3065 Computer Vision Coursework Report Topic: Image Search

Gaole Dai

20124917

Submitted May 6, 2022

School of Computer Science
University of Nottingham Ningbo China

Contents

1 Introduction

The main focus of this project is the **image search**, where a program is implemented for searching within a set of images based on the query image. Image search, also known as image retrieval search, has been considered an emerging technology and attracted great interest and a wealth of promise in the past few decades [1]. Since the growth of multimedia collections through daily accessible devices, the image retrieval technique has been increasingly applied in many systems as a more user-friendly and efficient information searching method. In addition, some traditional feature extraction algorithms were introduced in this module. Considering both the practical and feasible perspectives, it is worth exploring the image retrieval technique for this project.

The image retrieval frameworks could be divided into two categories: text-based and content-based [2]. The text-based image retrieval could be traced back to the 1970s, when all the images in the database are manually annotated with text descriptions. The image would be then searched based on the text. This approach would cause a large amount of human labor and annotation inaccuracy due to the subjective judgments [3]. The content-based image retrieval (CBIR) approach was introduced in the 1980s to overcome those disadvantages. The CBIR method classifies the images according to their visual contents, such as shapes, textures and color histograms. Figure 1 [1] illustrates the general

[1] https://medium.com/sicara/keras-tutorial-content-based-image-retrieval-convolutional-denoising-autoencoder-dc91450cc511

process of CBIR process. The features of the query image and images in the database are extracted, then a sequence of images is sorted based on the closest features computation results.

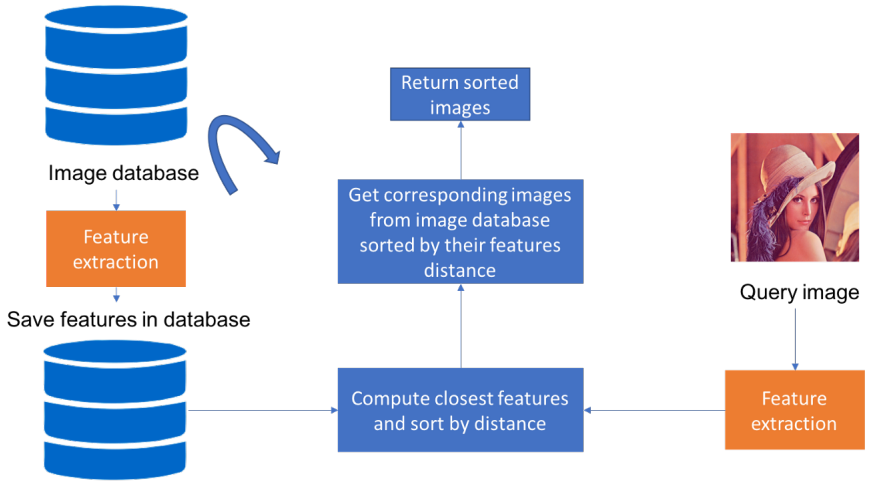

Figure 1: General process of content-based image retrieval method

The critical step in the CBIR search is the implementation approach of *feature extractor*. The conventional methods for feature description includes for example scale-invariant feature transform (SIFT) [4], speeded up robust features (SURF) [5], Histogram of Oriented Gradients (HoG) [6] and GIST [7]. Those feature extraction methods extract aggregation of local and global features and proved to be robust to geometrical translation. In addition, search systems achieve good results with those methods by approaches such as Bag of Words (BoW) [8]. In the recent decade, deep learning algorithms have shown great success in solving many problems [9]. In particular, the convolutional neural networks (CNN) have been applied as the feature extractor to generate the features of images for image retrieval problems.

In this project, we applied **five** different methods based on the existing approaches to tackle the image retrieval problem, which consists of both the text-based and content-based methods, traditional feature descriptor algorithms and CNN models as feature extractor methods. Furthermore, a web-based image search system is proposed to visualize the performance of each method. Users could decide to search for a specific image within a given database, or upload query images to search within the database. The reader may refer to [10] by Karthikeyan et al. for a more comprehensive understanding of image retrieval.

The key features implemented in this project are:

1. Content-based image retrieval with ResNet18 models trained on 189 images as feature descriptor;

2. Content-based image retrieval with ResNet18 models trained on 430 images as feature descriptor;
3. Content-based image retrieval with Autoencoder models as feature descriptor;
4. Content-based image retrieval with SIFT and VLAD algorithm for feature extraction;
5. Text-based image retrieval with keyword search;
6. The web-based application for visualizing the image search results; Image search with query images among database or with user upload images.

2 Design

To better realize the image search function and compare the performance of each algorithm, five different methods are designed to realize the image search. They are text-based image search, two types of supervised learning with CNN-based image search, unsupervised learning with CNN-based image search and SIFT and VLAD algorithms-based image search.

2.1 Text-based Image Retrieval

In text-based image retrieval, the traditional and straightforward method frequently used on World Wide Web—the keyword-based searching—is applied in this project. The images are indexed based on features such as the caption, filename, tags, or web page title for keyword-based search [11]. Common methods include optical character recognition (OCR) techniques and natural language processing (NLP) techniques, which transfer text information into plain text and enhance the performance of text information extraction. Those methods are effective in some perspectives but may lose merit compared to content-based methods. Though the keyword-based searching may produce poor results on large-scale web images because images may not be properly labelled, the database we used is smaller compared to the web image database. Also, the labels of our images have been properly assigned. Therefore, keyword-based searching according to image labels will be a direct and efficient method to implement.

2.2 Context-based Image Retrieval

The context-based image retrieval methods could be categorized into conventional methods and CNN-based methods. In this section, the methods used in this project will be discussed.

2.2.1 Conventional Method

The conventional methods for context-based image retrieval are based on BoW, and could be divided into the representation of local and global features. SIFT is one of the most

used representation algorithms as local descriptors. GIST and HoG are examples of global descriptors. The proposition of the vector of locally a descriptors (VLAD) [12] approach extends the BoW concept and shows better retrieval results compared to BoW approach [13]. Unlike BoW involves counting the number of descriptors for each cluster in a vocabulary and creating a histogram for each descriptor, the residual of each descriptor concerning the corresponding cluster is accumulated and then matched to the closest cluster for VLAD. The difference of each descriptor from the mean in the Voronoi cell is added in the VLAD method, which increases the discriminative property in feature vectors and becomes the principal advantage for VLAD over BoW. In this project, the **SIFT-VLAD** approach will be implemented as the representation of the conventional context-based method.

The general workflow from [14] demonstrates the general process of SIFT-VLAD method, as shown in Figure 2. The SIFT algorithm will be applied for feature extraction, then after possible codebook quantization, aggregation and concatenation, the VLAD will be utilized to assign each descriptor to the nearest cluster. Then the sum of the difference between the centroid of the cluster and the assigned descriptor is calculated.

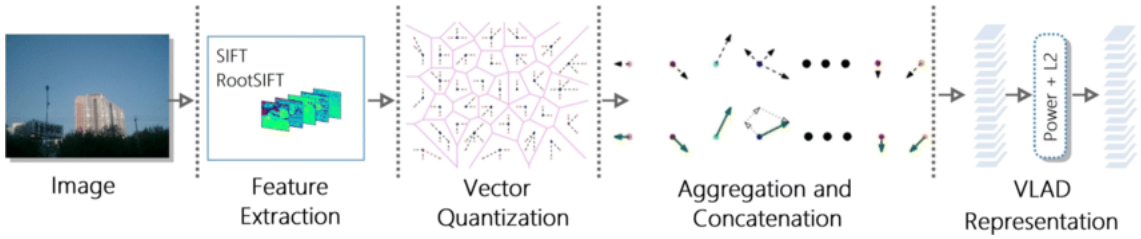

Figure 2: The workflow of the SIFT-VLAD context-based image retrieval method encoding scheme

2.2.2 CNN-based Method

Babenko et al. [15] demonstrated that convolutional neural networks trained to solve the image classification problem could be extracted to generate the 'neural code' of images. Figure 3 [2] shows the function of each CNN layer for image classification task. After the image feature learning, the fully connected layer before the *softmax* activation function could be extracted as the feature descriptor.

[2]https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html

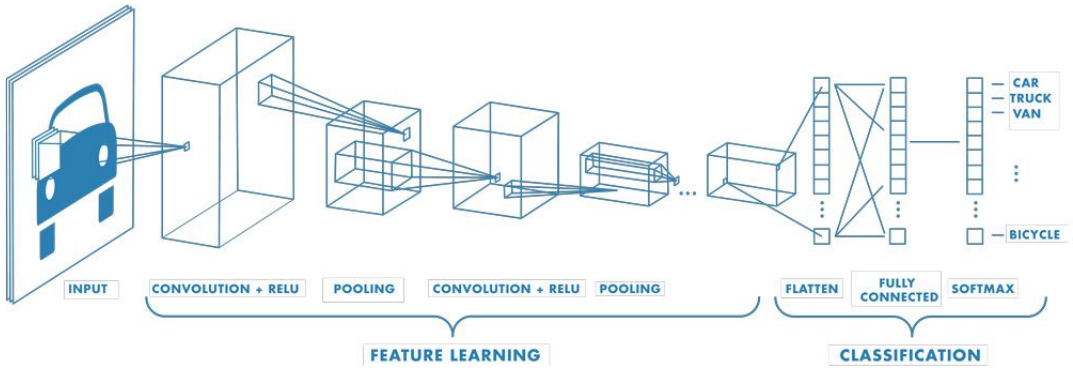

Figure 3: CNN networks with convolutional layers, the function of layers are divided into feature learning and classification

CNN-based image retrieval usually involves an ImageNet pre-trained CNN as the feature extractor, the pre-trained model could be fine-tunning or direct used with the k-nearest neighbor (kNN) approach to calculate the result images. In this project, the pre-trained ResNet18 model will be used as the *supervised learning* method. The overall procedure of CNN based supervised learning algorithm is designed as shown in Figure 4 [3]. The CNN model extracts the features in the images and the cosine similarities are computed. The target images are then sorted according to the cosine similarity.

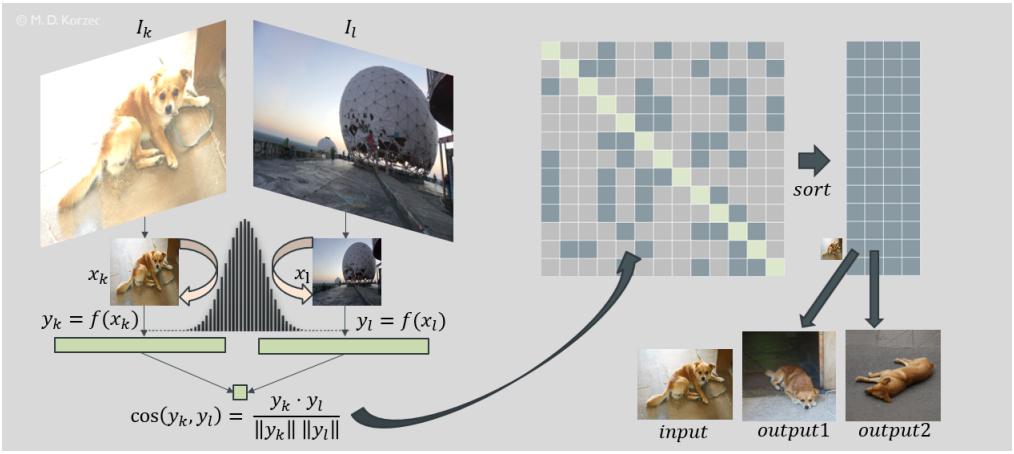

Figure 4: CNN based *supervised learning* image retrieval

Since the supervised learning model requires time-consuming and costly labelling tasks, another way to use the CNN-based method to extract the features is an unsupervised deep learning algorithm. One of the unsupervised learning algorithms utilizes the autoencoder method, which consists of an encoder converting the image to feature embedding representation and a decoder converting the feature embedding to the output image. The encoder and decoder share the same parameters and update by the mean squared error (MSE) between the input image and the output image. The overall autoencoder architecture is shown in Figure 5. The **autoencoder** based unsupervised learning method is designed to be implemented for this project.

[3] https://towardsdatascience.com/recommending-similar-images-using-pytorch-da019282770c

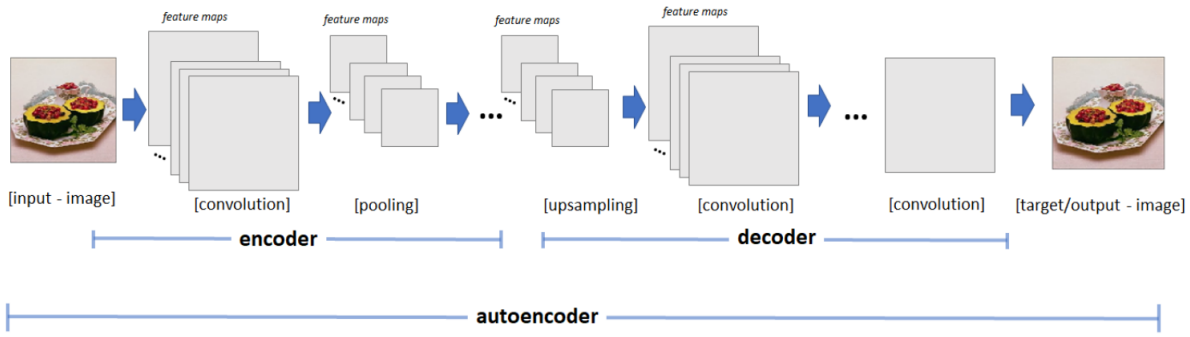

Figure 5: Autoencoder for *unsupervised learning* image retrieval

2.3 Dataset

The standard benchmark datasets for image retrieval include Holidays, Oxford Building, Paris, Deep Fashion and Tiny Images. Considering the time limitation, the subset of the Oxford Building [16] dataset will be used in this project. The Oxford Building dataset consists of 5062 images of 11 different Oxford landmarks in total, and there are 5 possible query images for each landmark. The subset used in this project consists of 189 'good' images ('good' images include those with a nice and clear picture of the object/building). The 189 images also include a total of 11 landmarks.

2.4 Performance Measurement

Mean Average Precision (mAP) is usually used for image retrieval model performance measurement. However, considering the small number of the image used in this project, the performance will be measured with landmark labels. If the query image belongs to 'All Souls Oxford', the output image with the same label will be regarded as a good match. The proportion of the good match over the total match will be calculated as the performance measurement.

3 Implementation

The main programming language for this project is Python, where the PyTorch framework is used for model training, the OpenCV framework for image processing and built-in computer vision algorithm applications. The HTML, CSS, JQuery and Flask are used in the implementation of the web-based application.

3.1 Keyword-based Image Retrieval

The text-based image retrieval is implemented according to the image labels. The regular expression determines the output images based on the searching keywords. Images in the database with labels containing the query keywords are filtered as the target images.

3.2 SIFT and VLAD Algorithm

The implementation of SIFT and VLAD algorithm for image retrieval includes the following steps:

Step 1: Apply the SIFT algorithm on every image in the database to extract features. A for loop is used for processing the images in the database, every loop the OpenCV built-in SIFT algorithm is executed to compute the feature descriptor.

Step 2: The 12 centroid of the cluster objects are obtained with the sklearn.cluster `KMeans.fit()` function.

Step 3: The descriptors are then assigned to the nearest cluster by the `KMeans.predict()` function. The sum of residuals between the descriptor and centroid is calculated. After that, the L2-norm is applied for quantization. The clusters and descriptors are stored with Pickle.

Step 4: For prediction, the SIFT descriptor will be calculated for the query image, then according to the descriptor, the query object will be assigned to the nearest clusters, and the distance between the query VLAD and database VLAD will be calculated, the top 5 nearest images will be sorted and returned as the output images.

3.3 Deep Supervised Learning Method

The subset of Oxford Building images is manually labeled with the landmark and stored in a .csv file. Then the CNN based feature descriptor for image retrieval is obtained with the following steps:

Step 1: Data preparation. Images are first resized to dimension 224×224 with `transforms.Resize()` function, to fit the input dimension of ResNet18. The mean and standard deviation are computed, with `transforms.Normalize([0.4814, 0.4865, 0.4683], [0.2736, 0.2776, 0.3152])`, the input image pixel values are normalized to [0, 1]. Then the `OxfordBuilding` dataset and dataloader are generated, 10% of the images are test data. The batch size for train data is 32, and for test data is 16.

Step 2: Fine-tuning. The structure and parameters for ResNet18 are extracted, and the final fully connected layer is changed to 11, which is the number of target classes. Next, the model is trained with the OxfordBuilding dataset. The optimizer is the Adam and the learning rate is 0.001, total training epoch is 30.

Step 3: Extract the feature descriptor. The images in the database are predicted with the trained model, and the fully connected layer after the average pooling layer is extracted as the feature descriptor for the image.

Step 4: Obtain Similar Images. The cosine similarity among all the feature descriptors is calculated, and the top 5 similar images for every image in the dataset are obtained and stored.

Step 5: Prediction. With the stored similar images calculated in **Step 4**, the similar image list for the query image in the database could be extracted and plotted.

3.4 Autoencoder based Unsupervised Learning Method

The overall procedure for the autoencoder feature extraction is similar to the supervised learning algorithm mentioned in Section 3.3, except for **Step 2** the model training part. For the encoder, it has 4 convolutional layers with (3, 3) stride and (1, 1) padding, and 4 max pooling layers, the activation is $ReLU$. The decoder has the reverse structure, with 4 transposed convolutional layers.

3.5 Web-based Application

The web-based application is an image search engine, which is implemented for visualizing the image search results. **Three** main functions are realized, which include the image search within the database with four different methods, keyword-based image search and user's upload image search among the database. The Flask framework is used to implement the back-end. The demonstration video could be found in `https://drive.google.com/file/d/1_DivjrptO42tvEio7E7tHXZqN87uXQy5/view?usp=sharing`.

Figure 6 shows the index page of the web application, where users could choose among the 189 database images as a query image to conduct an image search.

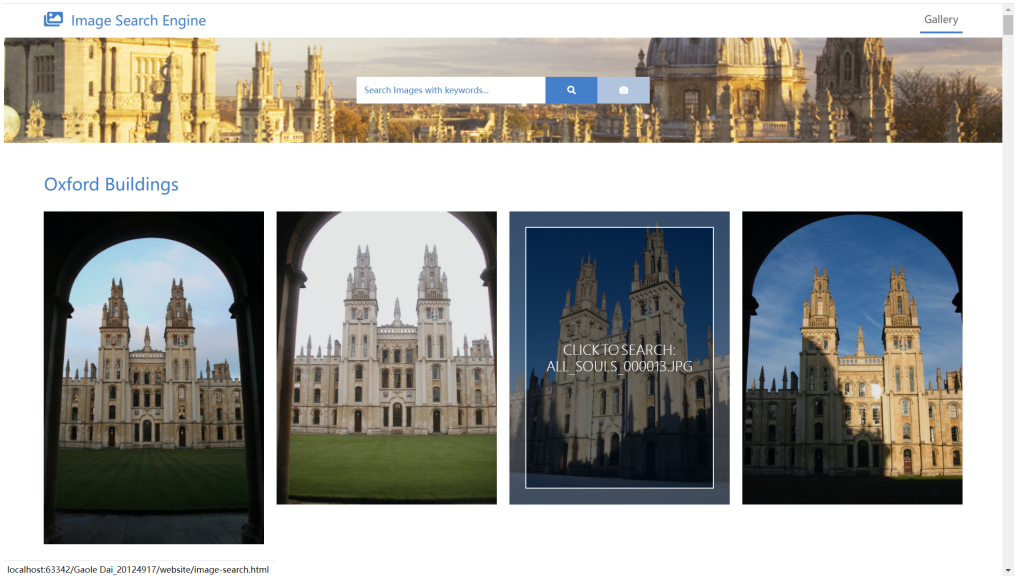

Figure 6: Index page of the web application

Once the user clicks one of the images, it will redirect to the search page (shown in Figure 7), where users could choose one of the methods for the search. The first choice is the ResNet18 with 430 training images as a descriptor, the second choice is the ResNet18 with 189 training images, the third choice is the autoencoder model, the last choice is the

SIFT and VLAD algorithm. Figure 8 illustrates the search results with the given query image.

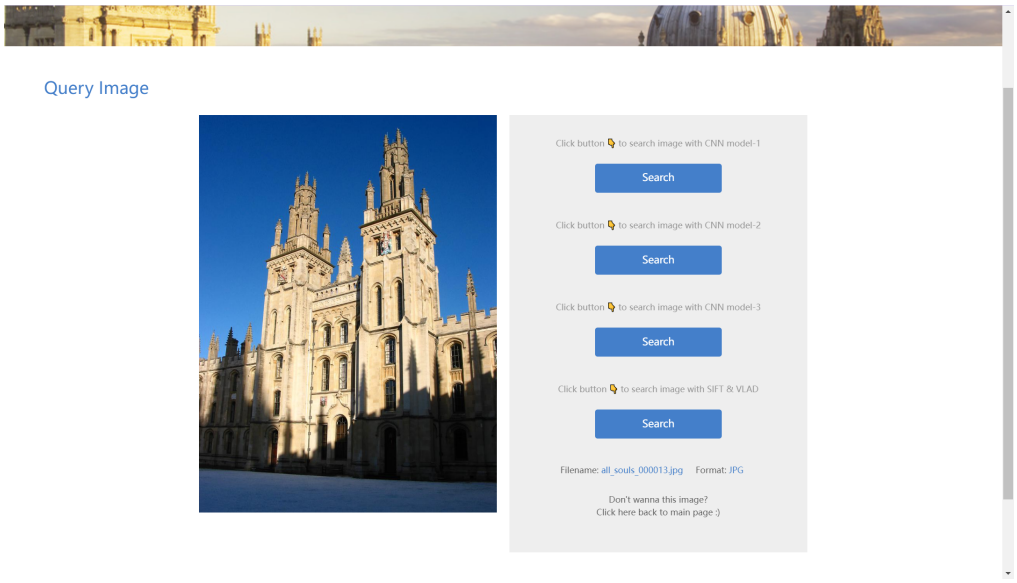

Figure 7: Image search page of the web application

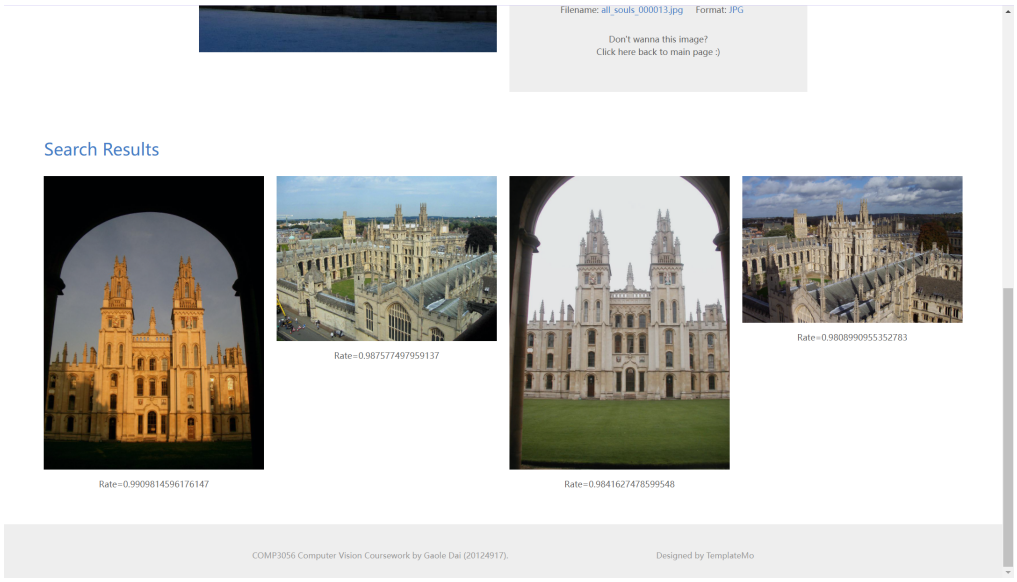

Figure 8: Image search result page of the web application

Users could search for a specific image with keyword or filter the database images. As shown in Figure 9, images with keyword 'magdalen' are filtered.