Aggregating Deep Features for Visually Similar Image Retrieval

Summary: In this project, a method of aggregating deep feature for image retrieval is investigated. First, the query images and candidate images are fed into a Deep Convolutional Neural Network and intermediate layer output of the neural network are retained to be aggravated. A series data preprocessing techniques were used to prepare the features for retrieving. During the retrieve, nearest neighbor search was used and a few distance metrics are researched.

1. Definition

There are billions of images on the Internet. But these images are just indexed by titles, not content. How to find the images one need on the Internet fast and accurate? Many methods have been brought out to address this problem. In this project, we investigated the possibilities of using features extracted by Deep Convolutional Neural Network to facilitate image retrieval.

1.1 Project Overview

In this project, we tried to use an image as a query to find a similar image. Deep convolutional Neural Networks were used to extract the raw feature. We then use some method to aggregate the raw for performing image retrievals. It has been realized that the features extracted by Deep Convolutional Neural network can generalize to a variety of tasks and show good performance. So, a method of using the deep features for image retrieval is possible to offer good results.

1.2 Problem Statement

Given images candidates, how to find the similar image to the query image? The similar image here is defined as the images have same building object so that the problem is tangible.

1.3 Metric

Average Precision (AP) is an ideal metric to measure the retrieved image. It not only considers whether relevant images are retrieved, but also take the order of result into account. If more related images are ranked before the irrelevant ones, the system is awarded the higher score. AP is calculated as

$$\mathrm{AP} = \frac{\sum_{k=1}^{n}[P(k)rel(k)]}{number\ of\ relevant\ images\ in\ the\ dataset}$$

n is the number of images in the dataset. $P(k)$ is the precision at the k-th item in the ordered results.

$$\mathrm{P(k)} = \frac{number\ of\ related\ image\ at\ rank\ k}{k}$$

$rel(k)$ is an indicator function, suggesting whether k-th item is relevant or not.

Because entering multiple query will get different Average Precision. We sum over of AP of different queries and take the average. The value is called Mean Average Precision (MAP).

2. Analysis

2.1 Data Exploration

There are a variety of datasets that are available for measuring the performance of image retrieval systems. Some datasets contain fashions of different style. Such datasets are used to train system to make recommendations for customers to find similar commodities. Some datasets compose of images of street views at different time and visual angles. This kind of datasets is aimed at training system to recognize places as man do. Some datasets are collections of landmark images in different views under different weather condition. As the same landmark contains similar visual pattern and often occupies a large portion of an image, so the performance of the system retrieving visually similar image can be measured by whether it can find other images of the same landmark in the dataset. Here, the dataset we use is the Oxford Building Dataset. It is available at http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/.

The Oxford Building Dataset contains 5062 images collected from Flickr by searching for particular Oxford landmarks. Totally 17 key words are used to obtain the dataset. The images in the dataset are named after the key words and IDs. All images has been manually annotated to generate a comprehensive ground truth for 11 different landmarks, each represented by 5 possible queries. This totally gives 55 queries over which an image retrieval system can be evaluated.

Each query consisted of 2 parts. The first part indicates the image containing the target landmark. Then, 4 coordinates were given to describe the position of the target building within the image. The 4 coordinates are the location of up-left corner and bottom-right corner of bounding box for the building. The ground truth file lists the images that should be retrieve by the system. The images of the target buildings are classified into 3 levels. The ‘good’ set contains the images that show a nice, clear view of the building. The ‘OK’ set are the images in which more than 25% percent of the building is clearly visible. If less than 25% of the building is visible, or there are very high levels of occlusion or distortion, the image is put into ‘junk’ set. Only images in ‘good’ set and ‘OK’ set are considered related and requires correctly retrieving.

2.2 Exploratory visualization

Some sample queries are visualized as Figure 1, with a red box indicating the position of the query.

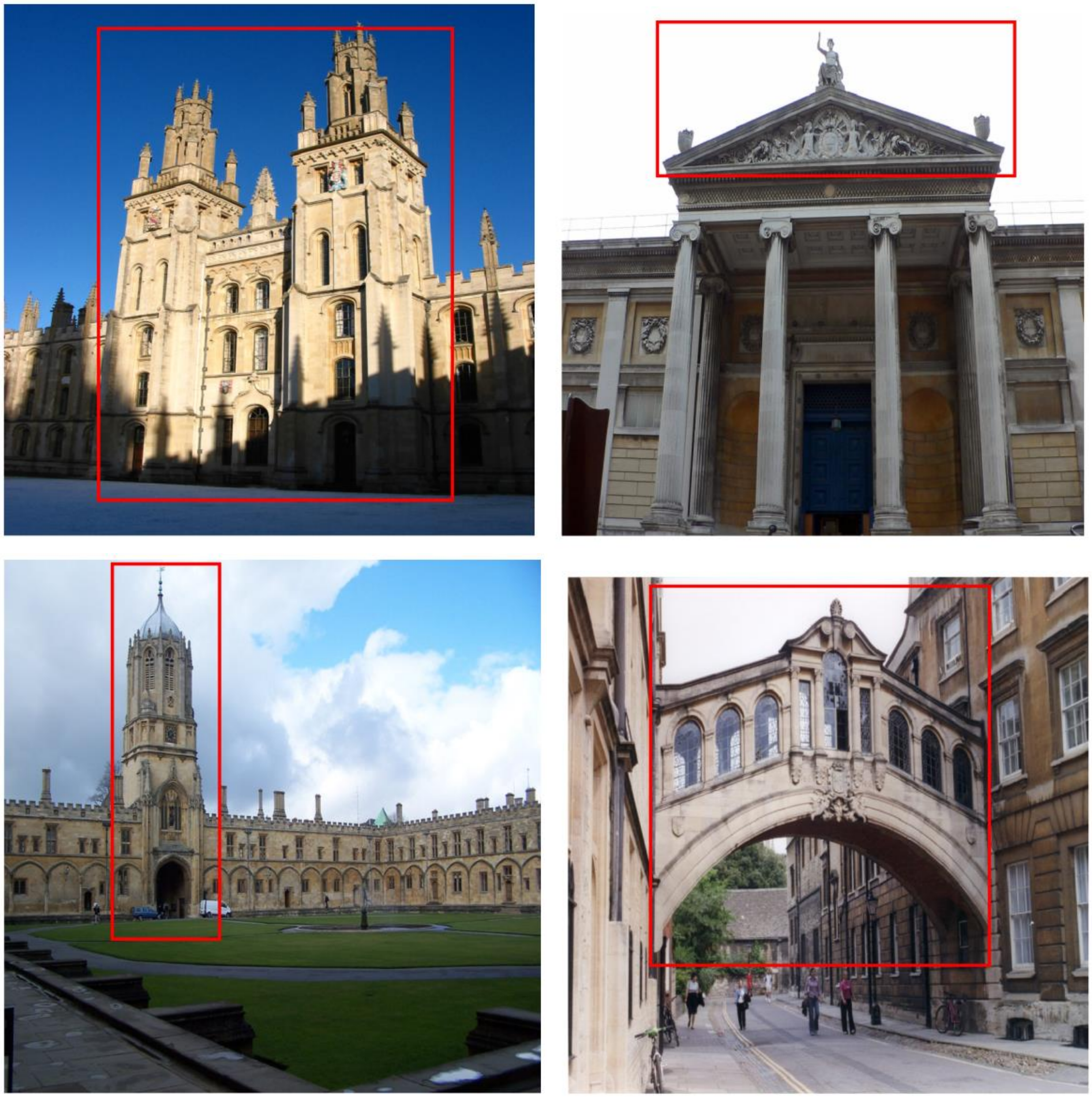

Figure 1: Examples of queries. The red box indicates the query object.

2.3 Algorithm and Technique

Retrieve visually similar image is not a new question. Much effort has been made since the problem was brought out. People already invented a branch of machine learning methods and handcraft methods to better help people find relevant image. But the result was not so ideal. And the algorithm is easy to fail if the datasets are different. Since Deep Learning was introduced to Computer Vision, many hard problems were solved. In this project, we tried to use the features extracted by deep convolutional neural networks to facilitate image retrieval.

2.3.1 General Deep Convolutional Neural Network Architecture

It has been found that convolutional neural networks (CNN) are very suitable to deal with images. A typical deep convolutional neural network consists of several typical layers. The most important layer is the convolutional layer. Each convolutional layer has tens or a few hundred filters. The size of individual filter is very small compared to the whole image, usually is 3 X 3 pixels or 5 X 5 pixels. If the image is an RGB image, the individual filter also connects to the 3 channels. Translating the filter across the image generates a map of response corresponding to different locations of the image. Response maps of all the filter composes of a volume. The response volume can be passed to another layer for further processing. Last, it should be mentioned that the filters are trained on images instead of being handcrafted.

Usually, the size of the image is very large, so is the response or activation of the convolutional layers. To reduce the dimension of the activation, pooling layer was introduced. There are several ways to do pooling. The most common way is called max-pooling. Dividing the activation of the former layer into small patches and choose the largest activation value as new activation and pass it to following layer. The patches can be overlapped or non-overlapped. Another pooling layer is the sum-pooling. All the activation value within the patch is added together and then pass to the next layer.

The overall architecture of deep convolutional neural networks typically starts with convolution layers interlaced with pooling layers for dimensional reduction. Several consecutive full connected layers with ReLU activation ($y = x \text{ if } x > 0, y = 0 \text{ if } x < 0$) are at last to do classification or regression.

In this project, we conduct our experiments on VGG16 network pre-trained on ImageNet []. The structure of VGG network can be found in Table 1.

Layer	Features Referred in the Project
Input (RGB image)	
Convolutional Layer 3x3-64	
Convolutional Layer 3x3-64	
Maxpool 2x2	Conv1
Convolutional Layer 3x3-128	
Convolutional Layer 3x3-128	
Maxpool 2x2	Conv2
Convolutional Layer 3x3-256	
Convolutional Layer 3x3-256	
Convolutional Layer 3x3-256	
Maxpool 2x2	Conv3
Convolutional Layer 3x3-512	
Convolutional Layer 3x3-512	
Convolutional Layer 3x3-512	
Maxpool 2x2	Conv4
Convolutional Layer 3x3-512	
Convolutional Layer 3x3-512	
Convolutional Layer 3x3-512	
Maxpool 2x2	Conv5
Fully Connected Layer-4096	FC1
Fully Connected Layer-4096	FC2
Fully Connected Layer-4096	
Soft-max Loss	

Table 1: Archtecture of VGG16. The second column indicate name of the output of left layer in this project.

2.3.2 Aggregating features extracted by deep convolutional neural networks

It has been found that the features generated by convolutional neural networks can generalize to a series of tasks. In this project, we exploit the ability of deep convolutional features to search for visually similar images.

As shown in Figure [], we indicate the output of each pooling layer as 'Conv5', 'Conv4', 'Conv3' …, the output of each fully connected layer as 'FC1' and 'FC2'. People have tried to aggregate the features extracted from different layers of the convolutional neural networks and found the Conv4, Conv5, FC1 layer usually generate better results. These features are corresponding to large image fields (each time the layer number increase 1, the neurons in that layer connect to more pixels on the image through the intermediate layers) compared the low-level features. Besides, the Conv4, Conv5, FC1 layer neurons are not tuned too particular to fit special task. Sometimes, people also use the features before the pooling layers, e. g. direct output of the convolution layer.

Some studies have recently suggested that the Conv5 features works the best under certain aggregation scheme. Though it is not wide accepted yet, in this project we use features extracted from Conv5 to perform image retrieval task.

There are varieties of ways to aggregate the Conv5 features. Conv5 features consist neuron activation across spatial and multiple filter channels. One can do sum-pooling and max-pooling to the activation over spatial domain or channels, or calculate particular weights for activation in certain special location and channels. Even though calculate pooling weights per channel and per location can improve the final performance by 1-5%, just doing sum-pooling over all spatial locations is a lot simpler and provide acceptable performance. In this project, we use sum-pooling to aggregate the Conv5 features.

2.4 benchmark

Previously, Neural Code was brought out that tried to image retrieval by utilizing features extract from Convolutional Neural Network. They used a network trained on ImageNet whose structure is like AlexNet. They did not aggregate the features but used PCA on the raw features directly. Without retraining the network on a landmark dataset of around 200,000 images, their best MAP is 0.435. After retraining the network, the MAP increased to 0.557. In this project, we aim to explore a method that can achieve similar precision but do not need retraining.

3. Methodology

3.1 Data Pre-processing

In this step, we follow the standard procedure of processing images before feed them into VGG16 network, e. g. using means of each channel to calibrate the input image. One thing worth mentioning is that usually, to conduct the classification task, one need to resize the image to fit the allowed input size of the neural networks. As we have removed all the fully connected layer on the top of the neural network, we can just use the convolutional neural network to extract features from the full-sized images.

3.2 Implementation

The step to implement image retrieval is listed below:

1. Extract Raw Features

Conv5 features are extract using Keras. Keras is a high level library built on Tensorflow and Theano to speed up deep learning prototype. All the image in the dataset was first feed into the VGG16 network to extract Conv5 features. Because the sizes of the input image are different the feature sizes for different image also vary. Assuming the size of the input image is (W, H). Then the output feature size is

$$W_{Conv5} = \frac{W}{2^4}, H_{Conv5} = \frac{H}{2^4}, \mathrm{C} = 512$$

The input size is divided by 2^4 because the network contains 4 pooling layer and each layer down sample the input by 2. Before feeding the activation into the convolutional layer, the input is padded so that the output size will not change after passing convolutional layer.

2. Aggregating Raw Features

The raw features extracted from individual image are aggregated by summing over all spatial positions. Using i, j to indicate the spatial position of the feature, the aggregated feature f_c in channel c is

$$f_c = \sum_{i=1}^{H_{Conv5}} \sum_{j=1}^{W_{Conv5}} \chi_{i,j,c}$$

where $\chi_{\mathrm{i,j,c}}$ is value of raw feature at position (i, j) in channel c. Finally, an image can be represented by a vector in the form

$$\mathcal{F} = \{f_1,\ f_2,\ f_3, \ldots f_c\} \qquad 1 \leq \mathrm{c} \leq 512$$

3. Normalizations, PCA and Whitening

The above image feature vectors are normalized to unit length. Then PCA and whitening are conducted on the features. The PCA and whitening parameter can be learnt from the same datasets or relevant dataset.

4. Final normalization

Last, the features are normalized again to ensure they are unit vectors.

5. Extract query Image feature and perform image retrieval

The same pre-processing steps are performed on query images and nearest neighbour search is used to find the visually similar image. In the search, Euclidean distance is calculated to decide the best-match images.

3.3 Refinement

Some ways are explored to improve the performance of the method.

1. Train PCA and whitening Parameters on a different dataset

At first, the PCA and whitening parameters are trained on the same dataset. It was found that training the PCA and whitening parameters on different dataset can improve the performance significantly. The alternative dataset used here is the Pairs Building dataset, which contains 6412 images of Paris landmarks. Paris dataset consisting of more image may help improve the performance of the PCA and whitening. Besides, I manually inspected the quality of the Oxford dataset and found around 1/3 of the pictures in each only contain indoor scenery, parties, nature scenery, or ornament. So, the quality of the 2 datasets do not vary much.

2. Use full-sized image as query instead of cropped images

In standard testing settings, the images are first cropped and then fed into Convolutional Neural Networks. We found that if by retaining the full image, instead of just using image patch specified in queries, the system precision will increase substantially. The possible reason is that the same landmark usually occurs in similar background. Some queries are about a specific part of building. And this part only shows in that building. So, providing the information on other parts of the same buildings can help improve system precision.

3. Use cosine distance rather than Euclidean distance to rank the result

There is no solid theoretical reason why we should only use Euclidean distance to rank the results. So, another distance measurement was also experimented on the test. While Euclidean distance measures the distance between 2 points in multidimensional space. The cosine distance measures the angle between the 2 vectors. It was found that adapting cosine distance would hardly affect the system precision.

4. Perform Query Expansion on retrieved images

Query Expansion (QE) is a powerful tool for increase performance of the information retrieval system. When Query Expansion is used, the feature of top-ranked images will be averaged together to conduct a second query and generated ultimate results. We found Query Expansion could efficiently boost the system performance.

4. Results and Justification

4.1 Result

A detailed result of the system can be found in Table 2. The top-10 result for each query is visualized in appendix. The best performance the system can achieve is MAP 0.63.

PCA and Whitening Features	Crop	Distance Metric	MAP	MAP when QE is used
Oxford Building Dataset	Yes	Euclidean	0.46	0.61
Oxford Building Dataset	No	Euclidean	0.61	0.67
Oxford Building Dataset	Yes	Cosine	0.44	0.61
Oxford Building Dataset	No	Cosine	0.57	0.66

Paris Building Dataset	Yes	Euclidean	0.579	0.63
Paris Building Dataset	No	Euclidean	0.65	0.68
Paris Building Dataset	Yes	Cosine	0.58	0.63
Paris Building Dataset	No	Cosine	0.65	0.67

Table 2: System MAP under different settings. 'MAP' means normal setting without Query Expansion (QE).

4.2 Justification

The researched method here provide significantly better results. And it do not need to be retrained. Aggregating the raw feature extracted from Convolutional Neural Network rather than use the directly can improve the performance. The possible reason is that each activation in each channel in the Conv5 features corresponding to certain visual pattern in different location. By summing the activation over all spatial position we got descriptions of strength of different visual pattern in the image. The new feature are global descriptors independent of the position of the pattern. Because the building object can appear in any location in the image, preserve the spatial information may mislead the system.
In terms of classification ability, VGG16 is superior to AlexNet-like structure, which could contribute to the better performance of the method in the project.

5. Conclusion and Reflection

5.1 Conclusion

Features extracted from Deep Convolutional Neural Networks are good representations of the image in terms of visual similarity. Using these features can help us retrieve similar image in an easy manner and achieve good performance.

5.2 Free-Form Visualization

The Top-10 relevant images was displayed in the appendix. We found that the result was quite good in terms of top-10 results. By inspecting the results we found that even though some results were not correct they were quiet reasonable. They were very similar to the query images in fact.

5.3 Reflection

First, the images were fed into a pre-trained neural network for extracting raw features. Then the features were aggregated by summing over all spatial location. Vector normalization, PCA and whitening were used to pre-processing the features. Last, the nearest neighbour search was implemented to obtain the similar images.

The most astonished finding is that we can improve the final performance by training PCA and whitening parameter on relevant dataset. It suggests, apart from the algorithm, training data also affect the final system precision much.

5.4 Improvement

Oxford Building Dataset is a very small dataset compared to industrial level problems. The search is only conducted over few thousands of images which cannot represent all possible searches. The scheme can be investigated on larger and harder datasets. More distractors can be added to the dataset to test the robustness of the system.

Appendix: Visualization of Results

In the below figures, the first image is the query image. The red box indicates the part cropped to perform query. The orange frame suggest wrong result given by the system. Top-10 result returned by the system was visualized.