CSE534: Final Report

Content-based Image Retrieval System using convolutional neural network and pHash Algorithm

Zheng kai **50247576**

Teammate
Chicheng Ma **50131866**

Date of submission: 05/10/2018

Introduction

Image retrieval system is a field of rapid development. The Content-Based Image Retrieval (CBIR) method utilizes features extracted from the image for retrieval. The commonly used image features are mainly colors, textures, and shapes, including local features and global features. Local features are image descriptors extracted based on an area of the image, such as Scale Invariant Feature Transform (SIFT). Global descriptors are based on descriptors extracted from the entire image, such as GIST. Various coding methods have been proposed, such as the feature bag (BOF), the Fisher vector, and VLAD. In recent years, convolutional neural network is used in extracting image features. By using a CNN, the images can also be given labels. Then using the label as index can make the retrieval process faster.

My teammate and I decide to do a project on Image retrieval system because we think this area is interesting and full of challenge. The most challenge problem is how to get the similar image correct and fast. Another problem is how to define the similarity. So we try to solve these problem on a small demo. This project will implement content-based image retrieval system using convolutional neural network and pHash Algorithm. Image retrieval is one of the most popular research areas of the image processing. Now, there are a lot areas that we can use Image retrieval to solve problems. If we have a picture, and we want to get the similar pictures in a database. We need image retrieval system. I and my teammate decide to implement a simple image retrieval system. I will responsible for implementing the CNN model, expressing image features, build image feature database including image labels and phash code. My teammate will responsible for parameter tuning, evaluation plan and improve the model performance.

Literature Review

[1] Faster R-CNN features for instance search

This paper mainly studies whether the ready-made features and fine-tuned features extracted from target-detected CNN networks can be used in the field of image retrieval. The main contents are as follows:

This paper proposes the global and local convolution features of the image extracted from the pre-trained target detection CNN through one forward propagation. This article uses the location information learned from the RPN to facilitate the provision of approximate location information for the top ranked retrieval images. A simple space rearrangement scheme is designed. This article also analyzes the impact of

fine-tuning the target detection CNN on the search, and concludes that this operation helps to obtain a better image expression.

The key to implement CNN is the “convolution layer”. A convolution layer applies a set of “sliding windows” across the image. Rather than isolate the pixel values, the filters treat them in small groups, that’s the CNN model can learn meaning features and locate them in any part of the image. Like most other “deep learning” networks, CNNs tend to have many layers. In order to obtain a global image representation from the Faster R-CNN, features are extracted from the last convolutional layer. After the convolutional features of the image are obtained, we sum the outputs of each filter to generate an image representation with the same number of dimensions as the convolution filter. Instance retrieval consists of three parts: screening stage, spatial rearrangement, and extended query.

[2] Content-Based Image Retrieval Based on CNN and SVM

The proposed method in this paper constructs a similarity by SVM learning based on a given pre-trained CNN. Suppose that there are two images I1,I2, with their corresponding deep features F1, F2. The similarity measure is defined as: Sim(I1,I2) = ||F1 - F2||, where sim is the similarity degree, and the smaller sim, the more similar between I1 and I2. If use distance metric learning (DML) to learn the similarity measure. The general formula is:

$$sim(I_i, I_j) = (F_i - F_j)^T M (F_i - F_j) \qquad (1)$$

where M is obtained by DML. The goal is to get a matrix M in (1) by learning, and it would be a diagonal matrix, which simply considering the linear relationship between the similarity degree and each feature dimension. VGGnet is used as the pre-trained CNN in this paper. It contains 5 convolutional functions, 3 pooling functions and 1 softmax function. The input image is 224*224 pixels in VGGnet. And and the output is a 1000-dimensional vector, where each element of the vector represents a particular category originally. So in our project, we can give labels to the CNN output.

The first step is extract the deep features of the all input image dataset, stored in a file called F. Then if there is a query image, sign as J, compute J’s deep features by using pre-trained CNN. Then for every image in F, compute the similarity degree. And at the last step, the paper use SVM method because for two images, there are only two choices, similarity and dissimilarity. So at this paper, CNN is used to extract features of images and SVM is used to learn the similarity measures.

[3] Medical Image Retrieval using Deep Convolutional Neural Network

This paper propose a framework of deep learning for content-based image retrieval system by using CNN. The deep CNN is trained for classification of medical images. The deep CNN model can identify 24 different classes.

There are two phases to construct the framework. The first phase is classification. This is a supervised learning processing in CNN. Deep learning algorithm learns low-level, mid-level and abstract features directly from the images. So this CNN model can trained for multiclass classification problem. The model contain four conv layers, three max pooling function, three fully connected layers. The model was trained using Stochastic Gradient Descent (SGD) with back propagation because this method will reduce training time. The second phase is features extraction for content-based image retrieval system. This step required establish features database for the whole training data. Then when input a image query, similar images will be given by computing Euclidian distance metric which using features from three fully connected layers.

[4] Advertisement Image Classification Using Convolutional Neural Network

This paper using the image classification model get by CNN and apply for online advertisement judgement. The CNN used is a network with two parameters(n,m), where n is the number of layers and m is number of filters in convolutional layer. When get the suitable parameters, the model will output Yes or No for one input image, which means the image is display clearly or not. So there are four steps of the method. Step 1: Input image. Step 2: Image capture. Step 3: Using CNN to classify images. Step 4: Output the classification conclusion.

At the CNN model part, the paper talk more details about the components. The first part of the model is configurations and initialize component. This component has two inputs: instance and configurations. The instance describes the number of layers , the number of filters on each conv layer, the classification method, size of input images, the kernel of convolution layers. The configuration describes about some parameters. The second part of the model is CNN Learning and Visualization component. This part will extract features of images in many layers.

[5] Fast content-based image retrieval using Convolutional Neural Network and hash function

This paper talk about how to reduce the computational cost of retrieval significantly at the state-of-the-art efficiency level. The paper used deep learning techniques such as convolutional neural networks and a novel end-to-end supervised learning framework that learns probability based semantic-level similarity and feature-level similarity simultaneously.

The field of content-based image retrieval consists of a lot of different indexing structures, schemes, and methods aiding the related retrieval tasks. The goal of the indexing structures is to take an available dataset, and produce a concise and easier to handle index which can be used to search for similar content. The paper using binary hashing because binary hashing is more computational and storage efficiencies. It tries to map high-dimensional image data to compact binary codes in a
Hamming-space while keeping several notion. Using Hamming-distance and binary pattern matching, the efficiency of fast image retrieval can be measured. An efficient end-to-end supervised learning framework is presented for fast image retrieval that learns probability-based semantic level similarity and feature-level similarity concurrently.

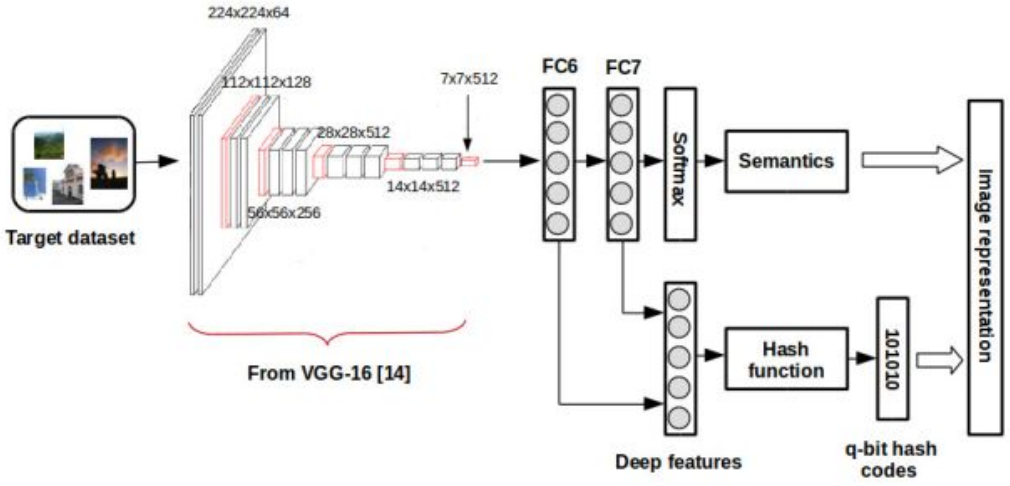


figure from this paper

As shown in the above picture, the algorithm consists of two main steps. In the first step, obtain the image representations with the help of a CNN which was supervised pre-trained on the ImageNet. The CNN model contains five convolutional layers, two fully-connected layers, and a softmax classifier. This model incorporates a huge amount of semantic information, since it was trained on ImageNet 2012 classification dataset which consists of more than 1 million images and it is able to classify into 1.000 categories. The output of the last fully-connected layer is splitted into two ways. One part eventuates in an n-ways softmax classifier where n stands for the number of categories of the target dataset. The other part makes up a hash-like function which composes the features obtained by CNN to hash codes.

[6] A Duplicate Image Detection Scheme using Hash Functions for Database Retrieval

This paper presents a new duplicate image detection scheme that employs hash functions to cope with the problem resulting from the large image database

containing the fingerprints extracted from the images. Moreover, it utilizes the extracted features of the host image to generate a fingerprint. When the authentication of suspect images is needed, the scheme only compares the matches from the same image to decrease the comparison time. The scheme uses the feature vector from the fingerprints to build a hash pool structures, and utilizes the phase angle calculated from the feature vector to query the probably duplicate images in the image database.

The scheme contains three phases: the preprocessing phase, the index construction phase, and the fingerprint quer phase. The preprocessing phase generates feature vectors that can represent the feature of the input image. The index construction phase builds the fingerprint image feature vector database by using hash fnctions. The fingerprint query phase compares the suspect images with the database images when authentication verification is requested.

Our Proposed Approach

In our project, there are two different kind of images. Here we use two cartoon images, pikuchu and squirtle. The first step is using convolutional neural network to classify the images. So at first, we will construct a CNN. After we training the model, the model should give us a label for every image. And when we input a new image, the model can return the label of this image. As we know, there are still many different features between two pikachu images, such as the color of the image, the action of the pikachu in the image. So the first step is only give us a big scope of the similar images. Here we want to find more similar images in our system. So the next step we use pHash algorithm. The pHash work process is as follows:

(1) Reduce the size: pHash starts with a small picture, but the picture is 32*32 is the best. The purpose of this is to simplify the calculation of the DCT instead of reducing the frequency.

(2) Simplify colors: Convert pictures into grayscale images to further simplify calculations.

(3) Calculation of DCT: The DCT transform of the picture is calculated to obtain a 32*32 DCT coefficient matrix.

(4) Reducing the DCT: Although the result of the DCT is a 32*32 matrix, we only need to keep the 16*16 matrix in the upper left corner.

(5) Calculate the average: As the mean hash, calculate the mean of the DCT.

(6) Calculate the hash value: This is the most important step. According to the 16*16DCT matrix, set the 256-bit hash value of 0 or 1 to be greater than or equal to the DCT mean value set to "1", less than the DCT mean value is set to "0". Together, they form a 256-bit integer, which is the fingerprint of this image.

We evaluate our solution on a separate validation dataset using classiciation error rate:

$$E = \frac{N_{\text{wrong}}}{N_{\text{V}}}$$

where N_{wrong} is the number of misclassification and N_{V} is the size of the validation dataset.

The overall system design

Our content-based image retrieval system consists three parts, the CNN model part, the phash algorithm part and the check similarity part.

For the CNN part, dataset is split into a training set, validation set, and the test set. The training set contains 160 pictures(50% pikachu, 50% squirtle) , the validation set contains 32 pictures(50% pikachu, 50% squirtle), the test set contains 40 pictures(50% pikachu, 50% squirtle). Each picture will be treated into 64*64*3 image array, and apply the filters to train the CNN model. The validation set is used to pick hyper parameter for each model, and the test set is used to evaluate the performance of each model. Convolutional neural network (CNN design): The model was consulted from tensorflow and tflearn. The major part is referred from Tensorflow and tflearn official website. We used 9 layers deep neural network to ensure the loss function converges.

For the pHash part, the goal is to get more similar images in our database. Calculate the pHash values of two images separately Calculate the Hamming Distance of the two pictures by the pHash value. Determine the similarity of the two pictures by the size of the Hamming distance.

Software implementation

We use python based on tensorflow library to build the CNN model. For the pHash part, we use java. We choose Mysql as our image feature database.

```
conv_1 = conv_2d(network, 32, 3, activation='relu', name='conv_1')
network = max_pool_2d(conv_1, 2)
conv_2 = conv_2d(network, 64, 3, activation='relu', name='conv_2')
conv_3 = conv_2d(conv_2, 64, 3, activation='relu', name='conv_3')
network = max_pool_2d(conv_3, 2)
network = fully_connected(network, 512, activation='relu')
network = dropout(network, 0.5)
network = fully_connected(network, 2, activation='softmax')
acc = Accuracy(name="Accuracy")
network = regression(network, optimizer='adam',loss='categorical_crossentropy',learning_rate=0.0005, metric=acc)
```

Figure 1: Core code for build the CNN

id	address	label	hashcode
1	D:/image/1.jpg	1	1110011111111111000010011101110101000110011101001000000111111
2	D:/image/2.jpg	1	1110101111111100000100101101011010000000110101111100110101100
3	D:/image/3.jpg	1	1110001111101111100101101011010010001100010110000000100101101
4	D:/image/4.jpg	1	1011001101110001000000100100111110001001110110010000100111011
5	D:/image/5.jpg	1	1011110011100001110001001000000100110111000010010110011001111
6	D:/image/6.jpg	1	1010110000011000001111111011100110011011111001111000110001111

Figure 2: MySql database design

Resource used for implementation

1, http://tflearn.org/

2, tflearn example:

https://github.com/tflearn/tflearn/blob/master/examples/images/convnet_cifar10.py

3, CNN tutor：

http://www.subsubroutine.com/sub-subroutine/2016/9/30/cats-and-dogs-and-convolutional-neural-networks

4, pHash Algorithm: http://www.phash.org/

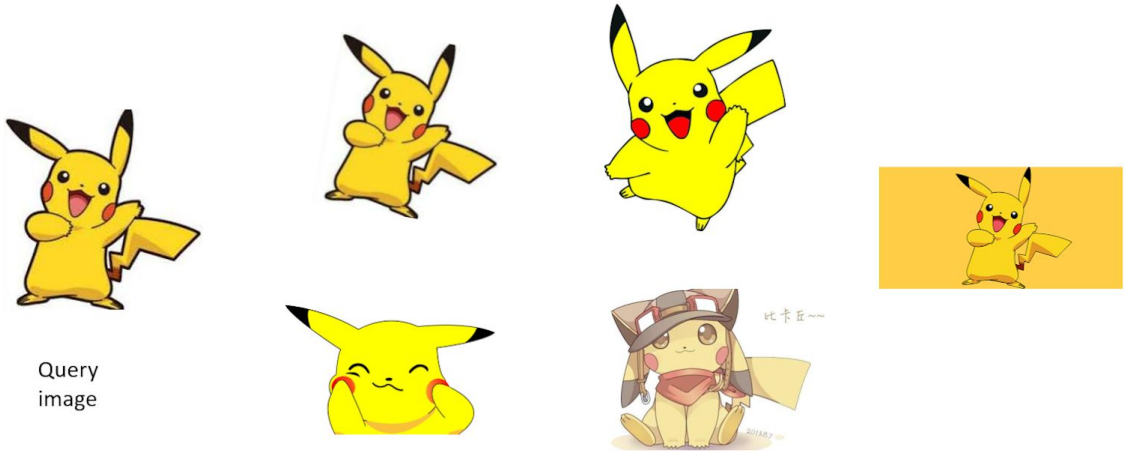
Presentation of the project outcome

After we trained our CNN model and build our image database. The accuracy of our CNN model is 97.05%

```
Training Step: 141  | total loss: 0.19421 | time: 0.837s
| Adam | epoch: 010 | loss: 0.19421 - Accuracy: 0.9636 -- iter: 060/144
Training Step: 142  | total loss: 0.17722 | time: 0.969s
| Adam | epoch: 010 | loss: 0.17722 - Accuracy: 0.9673 -- iter: 070/144
Training Step: 143  | total loss: 0.16606 | time: 1.052s
| Adam | epoch: 010 | loss: 0.16606 - Accuracy: 0.9705 -- iter: 080/144
Training Step: 144  | total loss: 0.40387 | time: 1.141s
| Adam | epoch: 010 | loss: 0.40387 - Accuracy: 0.9235 -- iter: 090/144
Training Step: 145  | total loss: 0.37345 | time: 1.262s
| Adam | epoch: 010 | loss: 0.37345 - Accuracy: 0.9311 -- iter: 100/144
Training Step: 146  | total loss: 0.33635 | time: 1.402s
| Adam | epoch: 010 | loss: 0.33635 - Accuracy: 0.9380 -- iter: 110/144
```

Figure 3: CNN model accuarcy

Then we use a pikachu image for query, the result is as follow:



Then we use a squirtle image for query, the result is as follow:

These five retrieved images' pHash code is similar with the query image pHash code, which the different bits less than 120.

Discussion of the outcome

For the above results, I think the last two is not the better match if we consider the action of pikuchu. We can make sure that after the CNN model, the label is right. So we only need to compare the query image pHash code with the images with the same label in our database. For these steps, we are correct. But for the more similar part, maybe pHash code is not the best algorithm to compare similarity between two different images. Another reason is there is a reducing size process in our step, if we don't reduce the size of picture, then we can get full info of the whole image. This will improve the performance. But this need huge work. Because the original image is generally very high. A 200*200 image has a full 40,000 pixels. Each pixel holds an RGB value, total 40,000 RGB, which is a huge amount of information. A lot of details need to be deal with. So we can't deal such more information.

Summary and Discussion

For the project

For this project, we implement a sample demo for Content-based image retrieval system using convolutional neural network and pHash Algorithm. Compare to other paper, we only have two classifications, so it's easy for us to get the correct label. We can get a good CNN model without huge train data. For the pHash part, we implement the DCT which can make our algorithm more strong because it recognize the distortion of the picture. As long as the degree of deformation does not exceed 25%, they can match the original image. From the project, I learn how to build a small CNN model and the pHash algorithm.

Learn in the course

After a semester of this class, I think this is a good course. Excellent professor and TAs. From this course, I understand the issues that multimedia systems face and the industry's research direction and solutions. I have learned some core concepts and mechanisms about multimedia systems. Not everything is given in the textbook. Attending the lectures and paying attention is important.

Reference List

[1] *Salvador, Amaia, et al. "Faster R-CNN features for instance search." Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on. IEEE, 2016.*

[2] *Content-Based Image Retrieval Based on CNN and SVM -- 2016 2nd IEEE International Conference on Computer and Communications (ICCC)*

[3] *Medical Image Retrieval using Deep Convolutional Neural Network -- Neurocomputing 2017*

[4] *Advertisement Image Classification Using Convolutional Neural Network* -- 2017 9th International Conference on Knowledge and Systems Engineering(KSE)

[5] *Fast content-based image retrieval using Convolutional Neural Network and hash function* -- 2016 IEEE International Conference on Systems, Man, and Cybernetics • SMC 2016 | October 9-12, 2016 • Budapest, Hungary

[6] Hsieh, Shang-Lin, Chuan-Ren Chen, and Chun-Che Chen. "A duplicate image detection scheme using hash functions for database retrieval." *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*. IEEE, 2010.