Practical Tricks to Boost Network Performance for Product Image Retrieval

Haoyu Li*, Junhong Huang*, and Ruijie Mai*

South China University of Technology
{1258119908, 175072347, 983783091} @qq.com

***Abstract*—In this online product image retrieval task, we focus on remarkable tricks applying only CNN-based features instead of merging it with hand-crafted features which show poor robustness of adaptability in various datasets. Considering the image statistic characteristic, we experiment several general tricks and propose some key methods to boost network performance in this task. First, we propose a new hard mining strategy called Batch-Pool Data Mining (BPDM) to larger the margins between two similar images. In the post-processing stage, SWA method and effective Early-late fusion method are introduced to enhance the stability performance of single or merged deep learning models. Our methods finally achieve an MAP@10 of 58.37 on JD Fashion Challenge dataset.**

***Index Terms*—Image retrieval, CNN-based features, SWA, Batch-Pool Data Mining, Early-late fusion**

I. Introduction

In this paper, we aim at the street-to-shop problem, *i.e.*, given a photo of a fashion item taken by users, the goal of this task is to find the same item from online shops. It remains a challenge because of the different photo conditions (*e.g.*, illumination, angle, placement and image quality) between users' images and online shopping photos. The key to dealing with this issue is the extraction of good feature representations. Compared with hand-engineered descriptions like SIFT [1], using deep neural networks to directly learn feature representations from raw data has recently achieved more impressive performance in many tasks. Accordingly, we use deep neural networks based method to learning the descriptions of users' and online shopping images.

We choice 4 popular CNN frameworks (*i.e.*, ResNet [2], SERseNet [3], SEResNext [3] and DenseNet [4]) as feature extractors. In order to improve the quality of features extracted by these CNN backbones, we first propose Batch-Pool Data Mining (BPDM), which not only enlarges negative sample search space but also maintains the search speed. Moreover, we introduce SWA method to stabilize the performance of single model, and propose Early-late Fusion to further improve the performance of multiple models. The final performance of our method achieves an MAP@10 of 58.37 on JD Fashion Challenge dataset with the 1000 query images and 0.15 million images, which are defined as the offline evaluation data in our experiments. We define 1800 pairs from 12000 custom-to-shop image pairs randomly as offline evaluation data@1800 and the other 10200 pairs as training data.

The main contributions of this paper can be summarized as follows:

- We investigate the product image search problem and propose some tricks to boost network performance for large scale product image search. These key methods are coordinated to enhance the robustness and discriminability of feature representations.
- We propose an effective model fusion method called Early-late fusion.
- We propose Batch-Bool Data Mining (BPDM) strategy to search hard-negative samples.
- We introduce SWA [2] method to improve the stability of the performance of single deep learning model, although it is not firstly proposed by us.

We conduct experiments on JD Fashion Challenge dataset. The experimental results demonstrate that our method can achieve remarkable performance on the product image retrieval task.

II. Related Work

CNN-based representation is a remarkable solution for image retrieval and is gradually replacing the hand-crafted local representation. A variety of outstanding methods apply CNN activations on the task of image retrieval. In this work, image retrieval is regarded as a metric learning problem which learns an image embedding and use the Euclidean distance to capture image similarity as Euclidean distance of the same type in the embedded space is smaller than that of the different type. Therefore triplet ranking loss [6] is introduced to get the relative distance between images. Chen et al [7] design a ranking loss which minimizes the cost corresponding to the sum of the gallery ranking disorders. Beyong triplet loss, Chen et al [8] introduces a quadruplet loss by adding another negative sample, which helps to enlarge inter-class variations and reduce intra-class variations. Additionally, [6] employs the contrastive loss aiming at measuring matching and non-matching pairs. Triplet-based loss generalizes better performance than contrastive loss in most cases.

The performance of pre-trained CNN lies in the feature extracting and feature descriptors encoding. The most straightforward method to extract feature representation is to use fully-connected(FC) layer [9] such as 2048-dim vector in Resnet.

*The three authors contributed equally to this work and can be considered co-first authors.

FC descriptors yields fair retrieval results along with Euclidean distance and recently be improved with L2 normalization and pooling method [10]. R-mac pooling use a fixed regid grid followed by a max pooling. Generalized-mean pooling introduced a more general pooling beyong max pooling and average pooling. It has proved that the last convolution layer after pooling could get outstanding accuracy rate to the fully-connected layer. After pooling method. descriptor whitening is widely used for jointly down-weighting co-occurrences since Chum [5] and PCA whitening is an effective one. In [11], Babenko et al. impose a Guassian mask on the last convolution layer before using pooling.

III. Methods to boost Network performance

Pipeline of the CNN-based method for image retrieval is shown in Fig. 1. As Shown in the figure, before training the network, image preprocessing (such as dirty data removal, data augmentation and so on) is applied to the data to make data more suitable for CNN training. Then, a CNN is used to extract image features, usually forced by a contrastive loss or triplet loss. After the features are extracted by the CNN, a pooling mechanism is used to construct a global image descriptor. Finally, a descriptor whitening is applied to the image features generated by the CNN. In the following sections, we will give a detail description of how to boost network performance on image retrieval.

A. Data preprocessing

In order to improve the generalization ability of processing different type of image in network, we list several data augmentations used in our experiments and give corresponding interpretations.

1) Image classification: Feature representation in CNN contains global structure information and local patch details. The global structure feature represents characteristics of the entire product including shape, style and positions, while the local feature focus on colors, textures and strong edges, such as buttons and shoes logo. Some images labeled with different categories have very similar textures, which may result in an incorrect search. Therefore, we introduce a classifier to divide the evaluation dataset into three categories and then search in a category.

We use 10200 cropped images labeled with bounding box and categories to train four classifier network (*i.e.*, ResNet-101, ResNet-152, DenseNet, and SeResNet) and 1800 as offline evaluation data.

In order to reduce the retrieval error caused by misclassification, confidence threshold is set to 0.85. If an image's confidence score in any category is larger than 0.85, it is considered as classified correctly and just searched in this corresponding category, otherwise it is searched in the whole dataset.

2) Random Erasing: Occlusion, such as billboard, watermark or sundries, is a critical influencing factor on the generalization ability of the model, which is commonly seen in custom-to-shop image retrieval task. When some part of a product is occluded, the represent of this product changes, although it is expected to be invariant. In order to improve the robustness of the model for occlusion, during training, we randomly choose a rectangle region of an image and erase its pixels with the ImageNet mean pixel value to generate various levels of occlusion. Compared with manually adding occluded regions to the training images, this method can be implemented easily and improves the generalization ability of the model.

3) Rotate&ColorJitter: Shoes images in JD Fashion Challenge dataset are characteristic as specific orientations like 90°, 180° and 270°. For clothes images between customer and shops, color difference and distortion are ubiquitous, while these are seldom seen in bags images. We consider four specific angles of rotation for shoes data with 30% probability and no rotation for other category. Experiments show that this approach improves 3% image searching top-1 accuracy in shoes and seldom decreases the accuracy in bags or clothes. To solve the color difference problem, ColorJitter is used to randomly change the brightness, contrast and saturation of an image, however it makes no effort on the improvement of the top-1 accuracy. We attribute this result to the color correction capability in the backbone network, which is not sensitive to limited color difference.

4) Multi Scale: Due to the existence of large difference in image resolution, the attention regions might have various size after applying resize method. It is considerable that multi-scale works well in dealing with such problem.In our experiment, the input image is resized to 480×480. Then, three re-scale versions with scaling factor of 1, $1/\sqrt{2}$, 1/2 are fed to the network to receive three scale-descriptors. Finally, the CNN-based descriptors are combined into a single descriptor by pooling for all methods.

Unfortunately, this trick does not work for improving the performance of our CNN-based descriptors. The main reason is that the given bounding box could provide a sufficiently high IoU score, and reducing the image scale will greatly damage the local feature representation with lower resolution. Additionally, upsampling may introduce useless noisy information. In our experiments, the model trained with 480×480 image size achieves the best performance.

B. CNN Network Backbone

We use 4 highly influential CNN backbones, namely ResNet101, SEResNet101, SEResNext101 and DenseNet. These backbones have different network architectures and thus have different results. In our experiment, SEResNet101, SEResNext and DenseNet achieve similar performance but all of them are quiet better than ResNet101 (2% higher). We also conduct some experiments about the depth of networks but we can not get a unified conclusion. For ResNet, Resnet101 (the deeper one) is better than ResNet50 (the shallower one) but for Densenet, Densenet161 (the shallower one) is better than Densenet169 (the deeper one). Thus the depth of networks has to be obtained through practical experiments. We also do some research on some techniques of network design, such as feature pyramid networks (FPN) [15] or FPN with attention

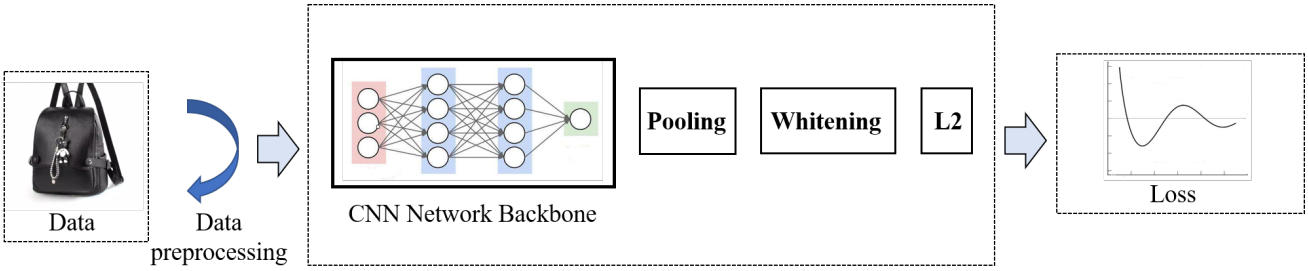

Fig. 1. Pipeline of CNN-based method for image retrieval.

mechanism [16] to generate multi scale features. These multi scale features can generate better results in offline evaluation dataset.

C. Pooling Methods

In order to generate a global image descriptor from CNN features maps, a common practice is to perform a pooling mechanism to the feature maps. Here we consider two kinds of pooling: R-mac [13] and GEM [19], [20].

1) R-mac Pooling: R-mac is first introduced by Tolias et al. [13] and modified Albert Gordo *et al.* [17]. R-mac is equivalent to the Region of Interest (ROI) pooling [18] using a fixed rigid grid followed by a max pooling.

ROI pooling can extract local and multi-scale features since the features are from different regions of different scales in the images. These local features are then max-pooled and l2-normalized independently to produce a single feature vector per region. Finally, a sum-aggregation is applied to generate the final image vector.

2) GeM Pooling: Generalized-mean (GeM) pooling is first proposed by G. Papandreou *et al.* [19] and is introduced to image retrieval by [20]. GEM is given by:

$$f^{(g)} = \left[f_1^{(g)} \quad \cdots \quad f_k^{(g)} \quad \cdots \quad f_K^{(g)} \right] = \left(\frac{1}{|X_k|} \sum_{x \in X_k} x^{p_k} \right)^{\frac{1}{p_k}} \tag{1}$$

max pooling and average pooling are special cases of GeM pooling, *i.e.*, max pooling when $p_k \to 1$ and average pooling for $p_k = 1$. The pooling parameter p_k can be manually set or learned since this operation is differentiable and can be part of the back-propagation.

We compare these two pooling methods in the given data. We find that Gem with a learnable pooling parameter p_k results in approximately 2% higher than R-mac on offline evaluation data @1800.

D. Loss Functions

1) Contrastive Loss: As implemented in [6], the contrastive loss function is defined as:

$$loss(f(a_i), f(b_i)) = (1-y)\max\{0, m - D(f(a_i), f(b_i))\}^2, \tag{2}$$

where $f(.)$ is an image embedding function mapping an image to a feature vector, y is label indicating that a_i contains the same item with b_i if $y = 1$ and otherwise if $y = 0$, and $D(.,.)$ is the distance between two feature vectors, such as Euclidean distance, cosine distance, hamming distance, *etc*. The margin m enforces distance between images of different items large, which has an effect of learning to rank.

2) Triplet loss: Triplet loss is first employed by [6], which achieves good results in face recognition and clustering. Triplet loss enforces distance between images of different items large and distance between images of same item small. A triplet loss function has the following form:

$$\begin{aligned} loss(f(a_i), f(b_i), f(n_i)) = \max\{(0, m + D(f(a_i), f(p_i)))^2 \\ - (f(a_i), f(n_i))^2\}, \end{aligned} \tag{3}$$

where a_i, p_i, n_i denotes anchor example, positive example, negative example respectively. a_i shares the same label with p_i, while n_i has a different label. m is a margin between positive and negative pairs.

3) Hard Data Mining Strategy: Mining hard-negative samples aims to enlarge the margins between two similar classes and improve the model's discriminability. To obtain difficult negative samples, which are predicted as matched pairs with high probabilities by the current network, we introduce Batch-Pool Data Mining (BPDM). Each negative sample n_i is assigned with a score s_k in each training batch-pool data. This batch-pool data consists of training batch data and pool samples randomly selected from the dataset. A sample with the smallest s_k and different ID is considered as a difficult negative sample of the corresponding query image.

During our experiments, we find that Triplet Loss is better than Contrastive Loss by a large margin, indicating that Triplet Loss is more suitable for this task. We also do some research on hard data mining strategy, finding that our proposed Batch-Pool Data Mining (BPDM) is slightly better than the original Batch Data Mining.

E. Whitening Method

Descriptor whitening is known to be essential for image retrieval. Here we include three whitening, namely PCA whitening, end-to-end whitening [17], and projections whitening [20]. PCA whitening is a common methods for image descriptor whitening. end-to-end whitening is proposed by [17] and proved to be worked in image retrieval. Its main idea is that the PCA projection can be seen as a combination

of a shifting (for the mean centering) and a fully connected (FC) layer (for the projection with the eigenvectors), with weights that can be learned. Thus, transferring PCA into a layer in the network and can be learned with the network. projections whitening is decomposed into two parts, whitening and rotation and it can be optimized with the training data after training the network.

In our experiment, we find that PCA whitening and end-to-end whitening almost bring no improvement but projections whitening can increase the result by 5% on offline evaluation dataset @1800 and increase by 4% on online test dataset.

F. Stochastic Weight Averaging

After training a model, it is difficult for us to elect the final model in several final epochs. Because what we know is the evaluation result on offline evaluation dataset, but we can not infer the test result on online test dataset. One Common way to solve this problem is to select out some models in the final epochs and averaging or voting their results. However, this is a kind of model fusion and requires testing in several models and is computationally inefficient.

In order to improve the generalization ability in one single model. We introduce Stochastic Weight Averaging (SWA) [2] to image retrieval. Different from [2], We modify SWA and the modified SWA algorithm in our paper proceeds the following steps:

- Training the network with learning rate decayed by 0.5 every 10 epochs.
- Getting model weights from 21_{th}, 22_{th} and 23_{th} epochs.
- Averaging the model weights of these three models to generate the final model weights, which is given by:

$$w_{SWA} = \frac{w_{21th} + w_{22th} + w_{23th}}{3} \tag{4}$$

- Computing the running mean and standard deviation of the activations for each layer of the network found by SWA after the training is finished.
- Generating one single model for testing, whose model weights are updated by three model.

SWA can be regarded as a implicit model fusion and requires no more computation. It is applied in weight space and only require one single model in the final test, but it can result in better generalization. During our experiments, we find that model generated by SWA results in 2% better than the original model, indicating that SWA is extremely useful for image retrieval.

G. Early-late Fusion

Basically, there exists two main streams for multiple feature fusion: early and late fusion. In early fusion, descriptors are combined at feature level. Then, the combined features are processed together to improve the robustness of the representations of products. On the other hand, late fusion refers to fusion at score or decision levels. In this paper, we propose a novel feature fusion method, which combines early and late fusion, called Early-late Fusion. In JD Fashion Challenge Competition, we fuse four different deep learning based models using early-late fusion. We split these four models into four groups, where each group contains three different models. The features extracted by models in one group are concatenated and then used to compute the distance between a query image q and a gallery image g. We thus get four distance scores (*i.e.*, d_1, d_2, d_3, d_4) from four groups. The final distance is defined as follow:

$$d_{final} = \alpha d_1 + \beta d_2 + \gamma d_3 + \delta d_4, \tag{5}$$

where α, β, γ and δ are trade-off parameters. For simplicity, we set them equally.

In JD Fashion Challenge Competition, our proposed Early-late Fusion is better than the original Early Fusion and Late Fusion. The performance of our proposed Early-late Fusion is 1% higher than that of the original Early Fusion and 2% higher than that of the original Late Fusion.

REFERENCES

[1] "Sift: Predicting amino acid changes that affect protein function," Nucleic acids research, vol. 31, no. 13, pp. 3812–3814, 2003.

[2] Izmailov P, Podoprikhin D, Garipov T, et al. Averaging Weights Leads to Wider Optima and Better Generalization[J]. 2018.

[3] Hu J, Shen L, Sun G. Squeeze-and-Excitation Networks[J]. 2017.

[4] Huang G, Liu Z, Maaten L V D, et al. Densely Connected Convolutional Networks[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2017:2261-2269.

[5] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in ECCV, 2014. 1, 3

[6] "Facenet: A unified embedding for face recognition and clustering," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.

[7] S.-Z. Chen, C.-C. Guo, and J.-H. Lai. Deep ranking for person re-identification via joint representation learning. TIP, 25(5): 2353–2367, 2016.

[8] Chen W, Chen X, Zhang J, et al. Beyond Triplet Loss: A Deep Quadruplet Network for Person Re-identification[J]. 2017:1320-1329.

[9] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in ECCV, 2014.

[10] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in ECCV, 2010

[11] A. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," in ICCV, 2015.

[12] "Learning a similarity metric discriminatively, with application to face verification," in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1. IEEE, 2005, pp. 539–546.

[13] Tolias G, Sicre R, Jégou H. Particular object retrieval with integral max-pooling of CNN activations[J]. Computer Science, 2015.

[14] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. 2015:770-778.

[15] Lin T Y, Dollar P, Girshick R, et al. Feature Pyramid Networks for Object Detection[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2017:936-944.

[16] Yu C, Wang J, Peng C, et al. Learning a Discriminative Feature Network for Semantic Segmentation[J]. 2018.

[17] Gordo A, Almazán J, Revaud J, et al. End-to-End Learning of Deep Visual Representations for Image Retrieval[J]. International Journal of Computer Vision, 2016:1-18.

[18] Girshick R. Fast R-CNN[J]. Computer Science, 2015.

[19] Papandreou G, Kokkinos I, Savalle P A. Modeling local and global deformations in Deep Learning: Epitomic convolution, Multiple Instance Learning, and sliding window detection[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2015:390-399.

[20] Radenović F, Tolias G, Chum O. Fine-tuning CNN Image Retrieval with No Human Annotation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, PP(99):1-1.