Image retrieval method based on two models re-ranking (IRM^{2}R)

Dapeng Zhang[1]
dpzhang25@gmail.com
Gongde Guo[1]
ggd@fjnu.edu.cn
Hui Wang[2]
h.wang@qub.ac.uk

[1] College of Computer and Cyber Security.
Fujian Normal University,
Fujian, CHN

[2] School of Electronics, Electrical Engineering and Computer Science.
Queen's University Belfast Research Portal,
Northern Ireland, UK

Abstract

The fundamental aim of Content-Based Image Retrieval (CBIR) is to find related images from a candidate image database based on a query image. Recent work exploits re-ranking ideas, which firstly retrieve candidate images via similarity retrieval using global image features and then re-rank the candidates by leveraging their local features. However, the results of re-ranking are very dependent on the effectiveness of newly acquired local feature information, and the global features and local features from the same model may focus on the same area, resulting in insignificant changes after re-ranking.

We aim to obtain more effective information to improve the efficiency of re-ranking. We propose a novel re-ranking method, image retrieval method based on two models re-ranking (IRM^{2}R), which allows using of the information in the initial retrieval of the two models to adjust one of the retrieval result sets. The experimental results show that the recall rate ($Recall@1$) of the image retrieval task is respectively improved by 3.8%, 2.3%, 2.2%, and 0.5% On the four datasets of Cub200, Cars196, Sop, and InShop. IRM^{2}R can effectively improve the accuracy of image retrieval.

1 Introduction

It becomes very challenging for users to retrieve large-scale images due to the exponential growth of image data. Therefore, content-based image retrieval (CBIR)[5, 7] becomes particularly important. CBIR aims to retrieve images similar to the input image from a database. Currently, remarkable progress has been made in this field, benefiting many intelligent tasks, including face, clothing, commodity, biological, and medical image retrieval.

For this task, it is effective to re-ranking the initial retrieval results with the newly acquired information. In particular, several studies [6, 21, 22, 26] optimize the embedding space by extracting global features to complete the initial retrieval, and then using local features to re-ranking the initial retrieval results to obtain the final retrieval results. These

methods usually include two primary components: extracting global features and extracting local features. The global features extract the most abstract information in the image to achieve fast retrieval, and the local features perform spatial geometry matching and perform re-ranking of potential candidate images to improve retrieval precision. Re-ranking methods rely on newly acquired local feature information to optimize initial retrieval results. Re-ranking methods rely on newly acquired local feature information to optimize preliminary retrieval results. However, both global features and local features are extracted by the same model, which may focus on the same area, resulting in poor results after re-ranking.

Aiming to address the above-mentioned issues in re-rank based on local features, we propose a novel re-ranking method for image retrieval method based on two models re-ranking (IRM^{2}R), which directly leverages two models for the retrieval results from the information of query images to optimize the initial retrieval results of one of the models. Firstly, we choose two models respectively to extract high-dimensional features of the image, and it is converted into a low-dimensional binary hash code by principal component analysis (PCA) [8] and iterative quantization (ITQ) [1]. Secondly, constructing two preliminary retrieval result sets by first calculating the Hamming distance and then calculating the Euclidean distance. Finally, IRM^{2}R leverages the information of two different retrieval result sets to adjust one of the retrieval result sets to obtain the final retrieval results. In the recall rate (Recall@1), IRM^{2}R surpasses two models by 3.8%, 2.3%, 2.2%, and 0.5% on the four datasets of Cub200, Cars196, Sop, and InShop, by proposing the re-ranking rules module, and significant performance gain can be obtained.

2 Related Work

Image retrieval based on re-ranking. Early works [3, 18, 19] have designed local features, which were used for global retrieval and re-ranking. Owing to the development of deep learning technologies, great progress has been achieved recently in [2, 4, 6, 12, 16, 24, 26], which directly extract image descriptors from CNNs to encode images and measure their similarity. In particular, several studies [6, 12, 21, 22, 26] optimized the embedding space by extracting global features to complete the initial retrieval and then using local features to re-ranking the initial retrieval results to obtain the final retrieval results. Re-ranking-based methods have two assumptions, i) the re-ranking retrieval results should not change much from the initial results; ii) visually similar samples should be close to each other after re-ranking. Therefore, it is challenging to extract correct and valid information for re-ranking faces great challenges. Cao et al. [6] leveraged learning global and local feature vectors using the same model, eliminating certain nearest neighbors using the global feature vector, and then rearranging these candidate images using the local feature vectors. Oriane Simeoni et al. [21] proposed a method of deep spatial matching (DSM) that leverages maximum Stable Extreme Region (MSER) [14] to extracts the regions with a high activation rate of image features as local features, which completes the re-ranking of preliminary results through feature matching. By taking global and local feature vectors for image pairs as the similarity between predicted images, it only needs to use fewer local features. Tan et al. [22] proposed Reranking Transformers (RRT) which is an image retrieval system at the instance level and an operation that exploits geometric verification of local features. It can predict the similarity of image pairs based on their global and local eigenvectors, making one forward pass with fewer local eigenvectors. Yang et al. [26] proposed a single-stage image retrieval method to fuse local information and global information to form the final image descriptor. But in

these methods, if the local features cannot learn effective information, this seriously affects the retrieval results after re-ranking.

3 Method

In this section, we introduce our proposed IRM^{2}R. An overview of IRM^{2}R is shown in Fig,1. Firstly, the feature extraction module M composed of two different image retrieval models M_α and model M_β extracts high-dimensional features from the test set images $\mathbf{X} = \{X_i\}_{i=1}^{n}$, and converts them into hash codes through ITQ [8]. Secondly, using Hamming distance and Euclidean distance, the query image gets two sets of results: $\alpha = \{\alpha_i\}_{i=1}^{S_1}$ and $\beta = \{\beta_j\}_{j=1}^{S_2}$. Finally, the re-ranking rules use the information of α and β to adjust α to get the final retrieval results.

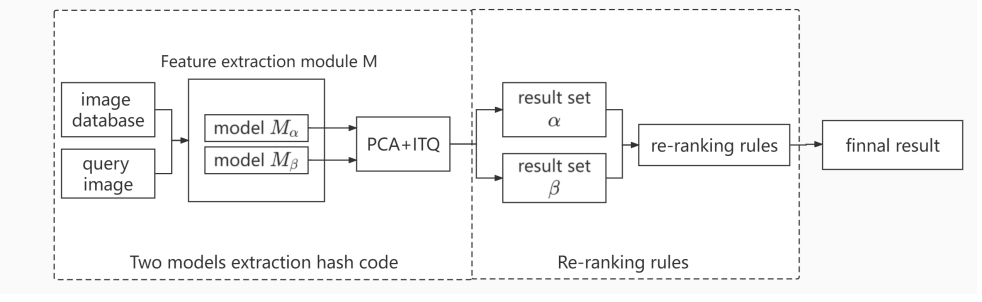


Figure 1: **The network architecture of the proposed IRM^{2}R.** Features from each image are fed into two models to obtain the hash code. Then, the query image first calculates the Hamming distance and then calculates the Euclidean distance to obtain two retrieval result sets. Finally, the final retrieval results is obtained by adjusting one of the retrieval result sets by using the information of the two retrieval result sets through the re-ranking rules.

3.1 Two Models Extraction Hash Code Module

Inspired by the two-model network, we choose two trained models M_α and M_β with different methods to build the feature extraction module M. The feature extraction module M, which takes the test image $\mathbf{X} = \{x_i\}_{i=1}^{n}$ as the input, is used to extract two feature matrix $F_\alpha \in \mathbb{R}^{n\times d}$ and $F_\beta \in \mathbb{R}^{n\times h}$. First, through PCA [1] for F_α and F_β, and then ITQ [8] tries to minimize quantization error their quantization errors, to obtain the optimal binary matrix $B_\alpha = \{b_i\}_{i=1}^{n} \in \{-1,1\}^{c\times n}$ and $B_\beta = \{b_j\}_{j=1}^{n} \in \{-1,1\}^{c\times n}$, where b_i and b_j are the binary code associated with query image x_i, and c is the code length. We define the hash function $: X \to B$ to generate the hash code as follows:

$$B = sign(FW_cQ) = sign(v)(k = 1, \cdots, c) \tag{1}$$

where B is a binary hash matrix of either B_α or B_β, F is the feature matrix F_α or F_β for extracting the image from M, W_c is coefficient matrix of PCA, Q is a orthogonal matrix. Then calculate the quantization loss:

$$\min(L(B,Q)) = \|B - FW_cQ\|_F^2 \tag{2}$$

where the $\|\cdot\|_F^2$ is the L2-norm. The whole process minimizes Equation 2 by continuously updating the orthogonal matrix Q, and finally quantify the feature matrix into a binary hash matrix.

At query time, the query image x_i fed into the two models M_α and M_β to compute two different feature vectors, which are converted into two hash codes B_{α_i} and B_{β_i} by ITQ processing. Then, We obtain two retrieval results by calculating the distance between different values of the hash code of the same number of digits. The Hamming distance function is as follows:

$$d_H(b_x, b_i) = \sum_{j=1}^{c} (b_x^j \oplus b_i^j), \tag{3}$$

where b_x is the hash code converted from the feature vector obtained after the query image x_i is processed by M_α or M_β, b_i indicates that the image in the gallery matches the hash code. Then, we take the top K results with the smallest equation 3 and compute their Euclidean distance from the query image to get two retrieval results: $\alpha = \{\alpha_i\}_{i=1}^{S_1}$ and $\beta = \{\beta_j\}_{j=1}^{S_2}$, representing the retrieval results of model M_α and model M_β for query image x_i, $\alpha_i = \{\rho_\alpha^i, \sigma_\alpha^i, \tau_\alpha^i\}_{i=1}^{S_1}$ and $\beta_j = \{\rho_\beta^j, \sigma_\beta^j, \tau_\beta^j\}_{j=1}^{S_2}$ consist of a triple: the Euclidean distance between the candidate image and the query image ρ, the class of the candidate image σ, and the id of the candidate image τ, the values of i and j indicate the degree of similarity.

At evaluation time, all the images in the test set are converted into hash codes, and each image can retrieve corresponding similar images. To evaluate the retrieval performance, we remove the images with the same id as the query image x_i from the retrieval results to form the final retrieval results of each model for the query image x_i.

3.2 Re-ranking Rules Module

At the re-ranking time, only the information of the query image x_i is unknown, and the information of other images is known. After the above process, α and β have been able to fully represent the retrieval results of M_α and M_β for x_i, so the re-ranking rules module aims to find accurate retrieval information by retrieving high-precision retrieval results in the result sets α and β. We think there are three instances of high-accuracy retrieval results: the class of the first candidate image for both models is the same, the top K of either model has a large number of images of the same class, and both models have the same candidate image. As shown in Figure 2, when instance 1 or instance 2 is satisfied, we obtain a class information l, and when instance 3 is satisfied, we obtain a position information I. The final search result is obtained by adjusting ρ_α^i with these information, and then re-ranking α according to the value of ρ_α^i.

Instance 1: the class of the first candidate image for both models is the same. At present, the recall rate $R@1$ of the classic image retrieval model for these four datasets can reach at least 70%, so it is the same as the class for the first candidate image, indicating that the accuracy of this class $l_\alpha\beta$ is very high. Thus we define f_1 to determine whether this situation is true. f_1 can be formulated as:

$$l_{\alpha\beta} = f_1(\alpha, \beta) = \begin{cases} \sigma_\beta^1, & if \ \ \sigma_\alpha^1 == \sigma_\beta^1 \\ 0, & otherwise \end{cases}, \tag{4}$$

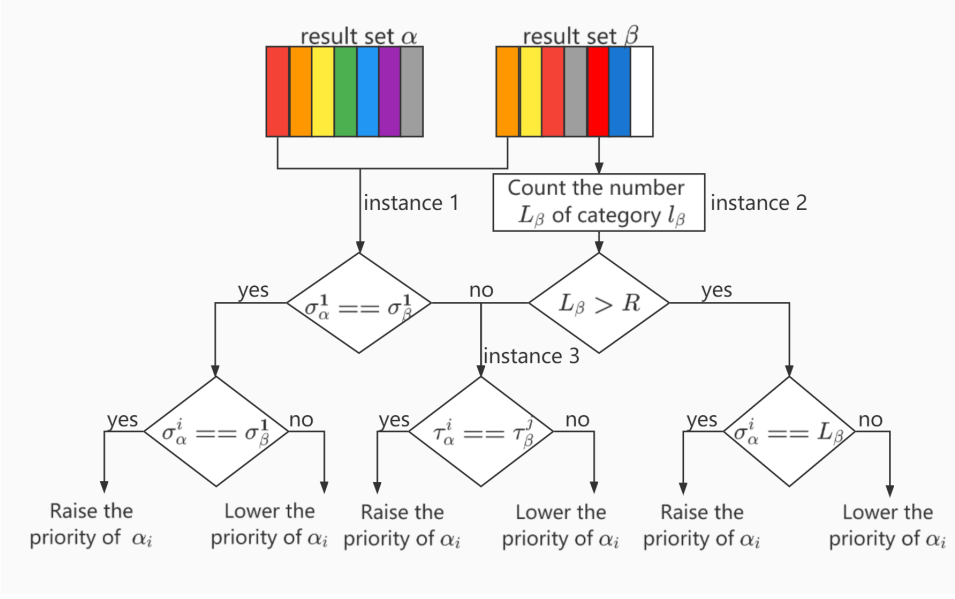


Figure 2: **The re-ranking rules module used in our method.** The module consists of 3 instances.

where σ_α^1, σ_β^1 refer to the class of M_α and M_β for the first candidate image of query image x_i, So when $l_{\alpha\beta}$ is not 0, it means the class of the first candidate image of the two models

If $l_{\alpha\beta}$ is not 0, the instance 1 is established. We use information of $l_{\alpha\beta}$ to adjust retrieval result sets α. Specifically, it increases the priority of the retrieval results in the same category as $l_{\alpha\beta}$ in M_α. This is done by first comparing σ_α^i with $l_{\alpha\beta}$ of candidate images traversing α, and changing ρ_α^i according to the result of the comparison. This process can be defined as:

$$F_1(\alpha, l_{\alpha\beta}) = \begin{cases} \rho_\alpha^i - \lambda_1, & if \ \ \sigma_\alpha^i == l_{\alpha\beta} \\ \rho_\alpha^i + \lambda_1, & otherwise \end{cases} \quad (i = 1, 2, \cdots, S_1), \tag{5}$$

where λ_1 is an adjust the distance parameter, ρ_α^i is the Euclidean distance between the candidate image α_i and the query image x_i.

instance 2: the top K of either model has a large number of images of the same class. Suppose l_β, L_β refer to the class with the most top K candidate images of M_β and its corresponding number. When there are enough l_β, it means that the retrieval results of M_β for x_i are relatively consistent, and the retrieval results have high reliability. on the contrary, it means that the class of the top K retrieval results of M_β are not uniform, and the reliability of the retrieval results is low. Thus we define f_2 to determine whether this situation is true. f_2 can be formulated as:

$$l_\beta = f_2(\alpha, \beta) = \begin{cases} l_\beta, & if \ \ L_\beta \geq R \\ 0, & otherwise \end{cases}, \tag{6}$$

where R is an adjust the distance parameter, $l_\beta \neq 0$ denote instance 2 is true. it means that the number of top K of the same class in the retrieval results of M_β to x_i is greater than that of R.

If $l_\beta \neq 0$, the instance 2 is established. We use information of l_β to adjust retrieval result sets α. Specifically, increase the priority of the retrieval results in M_α and l_β of the same class. This is done by first comparing σ_α^i with l_β of candidate images traversing α, and changing ρ_α^i according to the result of the comparison. This process can be defined as:

$$F_2(\alpha, l_\beta) = \begin{cases} \rho_\alpha^i - \lambda_2, & if \ \ \sigma_\alpha^i == l_\beta \\ \rho_\alpha^i, & otherwise \end{cases} \quad (i = 1, 2, \cdots, S_1), \tag{7}$$

where λ_2 is an adjust the distance parameter.

instance 3: both models have the same candidate image. If a result of Equation 4 or Equation 6 returns 0, it means that the retrieval results of M_α and M_β for x_i are not uniform. We aim to improve the priority of the same candidate images of the same M_α and M_β. The location information indicates how similar x_i is, so the location information must be carefully considered. At each traversal of β, β_j with the same image id as the candidate image α_i is computed the position information I_i. At the same time, in order to reduce the priority of inconsistent candidate images, we record I_i as a negative value for every 8 images that do not have the same candidate image. This process can be defined as:

$$I_i = f_3(\alpha, \beta) = \begin{cases} j, & if \ \ \tau_\alpha^i == \tau_\beta^j \\ -\lfloor \frac{8}{j} \rfloor, & otherwise \end{cases} (i = 1, \cdots, S_1), (j = 1, \cdots, S_2), \tag{8}$$

when $I_i > 0$, it means that the same image id of the $i-th$ candidate image retrieved by α and the $j-th$ candidate image retrieved by β. On the contrary, I_i means that no candidate image with the same image id as α_i is found in $\{\beta_j\}_{j=1}^{I_i \times 8}$.

Calculated by Equation 8, I_i can represent the position information of each α_i in β. Then adjust the priority of the corresponding α_i according to I_i. This process can be defined as:

$$F_3(\alpha, I_i) = \begin{cases} \rho_\alpha^i - \lambda_3 (1 - \frac{I_i}{S_2}), & if \ \ I_i > 0 \\ \rho_\alpha^i - \lambda_3 \times I_i, & otherwise \end{cases} \quad (i = 1, 2, \cdots, S_1), \tag{9}$$

where λ_3, λ_4 are an adjust the distance parameter.

Finally, retrieval results for x_i can be obtained by re-ranking α against ρ_α^i.

4 Experimental Results

4.1 Implementation Details

Model selection and parameter setting. Our criteria for selecting models focus on the high accuracy of the model itself and the difference between the models, ensuring the accuracy of the model as much as possible, and at the same time, it is hoped that the concerns of the two models are as different as possible. We choose CGD [10] and ProxyNAC++ [23] to build the feature extraction module M. Both models are pre-trained on Resnet50 [9] using the common ImageNet [20] and are ranked top for retrieval accuracy on the four datasets. **CGD** represents an image by combining multiple pooling techniques to generate multiple global descriptors. **ProxyNAC++** solves the small gradient problem of proxy by proposing a fast-moving proxy component based on Proxy-NCA [15]. The role of the proxy is to handle the relationship between the data in a batch. Therefore, the two models have different concerns, which may

lead to different understandings of image recognition. In all of our experiments, The image features are extracted by ProxyNAC++ to generate a 2048-dimensional vector. CGD selects three global descriptions to represent image features to generate a 1536-dimensional vector. In addition, we keep the original CGD and ProxyNAC++ parameter settings and only use the training data set to optimize hyperparameters.

Datasets and Evaluation metric. Four image datasets are used to evaluate our approach: the Caltech-UCSD Birds (Cub200) [25], the Stanford Cars (Cars196) [11], the Stanford Online Products (Sop) [17], and the In Shop Clothing Retrieval (InShop) [13]. Table 1 shows an overview of each dataset's makeup in terms of the number of images and classes. **Caltech-UCSD Birds** is developed for images image classification and image retrieval. It contains a total of 11,788 bird images and 200 classes. The first 100 classes are used to train the model and the last 100 classes are used to evaluate the model. There are approximately 60 images in each class. **Stanford Cars** contains a total of 16185 different types of cars and 196 classes. The first 98 classes are used to train the model and the last 98 classes are used to evaluate the model. There are approximately 80 images in each class. **Stanford Online Products** is developed for images image classification and image retrieval. It has 22,634 classes with 120,053 product images. The first 11,318 classes (59,551 images) are split for training and the other 11,316 (60,502 images) classes are used for testing. There are approximately 10 images in each class. **In-shop Clothes Retrieval** evaluates the performance of in-shop Clothes Retrieval. It contains a total of 7,982 clothing items and 52,712 in-shop clothes images, and the number of instances per class is very low for InShop datasets. There are approximately 7-8 images in each class. It is different from other datasets in that this dataset divides the data into three parts: training images, query images, and gallery images. Using the same evaluation protocols detailed in [cgd, ProxyNCA++]. To evaluate our model, we evaluate retrieval performance based on $Recall@K$, $R@k = \frac{1}{n}\sum_{i=1}^{n}(score_i)$ if the query image has at least one of the same class as the first k images returned, then $score_i = 1$, otherwise $score_i = 0$.

Table 1: **The composition of all four image retrieval datasets.** The number of classes for the Sop and InShop datasets is large when compared to CUB200 and Cars196 datasetr. However, the number of instances per class is very low for the Sop and InShop datasets. resulting in few correct answers in the gallery, which will affect the re-ranking retrieval results.

dataset name	images	classes	train	test	gallery correct answers
Cub200	11788	200	5864	5924	50-60
Cars196	16185	196	8054	8131	70-80
Sop	120053	22634	59551	60502	5-6
InShop	52712	11967	20052	126123	2-3

4.2 Result

4.2.1 Comparison with CGD and ProxyNCA++

Tables 2 and 3 show a comparison between the results of CGD and ProxyNCA++ on Cub200, Cars196, Sop, and InShop. Compared with CGD and ProxyNCA++, Our IRM^{2}R improved by 3.8%, 2.3%, 2.2%, and 0.5% on $R@1$, and our proposed re-ranking rules module exhibits

superior performance without the training model. Because of the nature of the re-ranking rules module, The performance improvement of the proposed model is not obvious in In-Shop. We think that the re-rank rule module utilizes a lot of wrong information because the query image has fewer numbers of the same class as the image in the gallery.

Table 2: Recall@k (%) on Cub200 and Cars196

datasets	Cub200				Cars196			
R@k	1	2	4	8	1	2	4	8
ProxyNCA++	72.8	82.6	89.2	93.5	88.8	93.7	96.6	98.2
CGD	76.8	84.8	90.6	94.3	92.5	96.1	97.8	98.6
IRM^{2}M	**80.6**	**85.9**	**90.9**	**94.5**	**94.8**	**96.9**	**98.1**	**98.8**

Table 3: Recall@k (%) on Sop and InShop

datasets	Sop				InShop			
R@k	1	10	100	1000	1	10	20	30
ProxyNCA++	81.2	92.2	**96.8**	**98.9**	90.4	**98.1**	**98.8**	**99.2**
CGD	79.3	90.6	95.8	98.6	83.6	95.7	97.1	98.1
IRM^{2}M	**83.4**	**92.8**	**96.8**	**98.7**	**90.9**	**98.0**	**98.8**	**99.1**

Table 4: **Ablation study for re-ranking rules**, Comparison of three instances in re-ranking rules, trained separately, with different instances. We report R@k results of images retrieval performance from the Caltech-UCSD Birds datasets (Cub200).

instance 1	instance 2	instance 3	R@1	R@2	R@4	R@8
0	0	0	76.8	84.8	90.6	94.3
1	1	1	**80.6**	85.9	90.9	94.8
1	1	0	77.4	84.7	89.6	93.3
1	0	**1**	79.0	**86.3**	**91.6**	**95.0**
0	1	1	79.5	86.2	91.5	94.8

Table 5: **Count the number and accuracy of two retrieval result sets that satisfy instance 1 and instance 2**. When the two retrieval result sets satisfy instance 1 or instance 2, we get a class to compare with the class of the query image, and count the correct rate and quantity.

instance	images	re-ranking images	correct images	accuracy
1	5924	4053	3762	92.8
2	5924	4305	3964	92.1

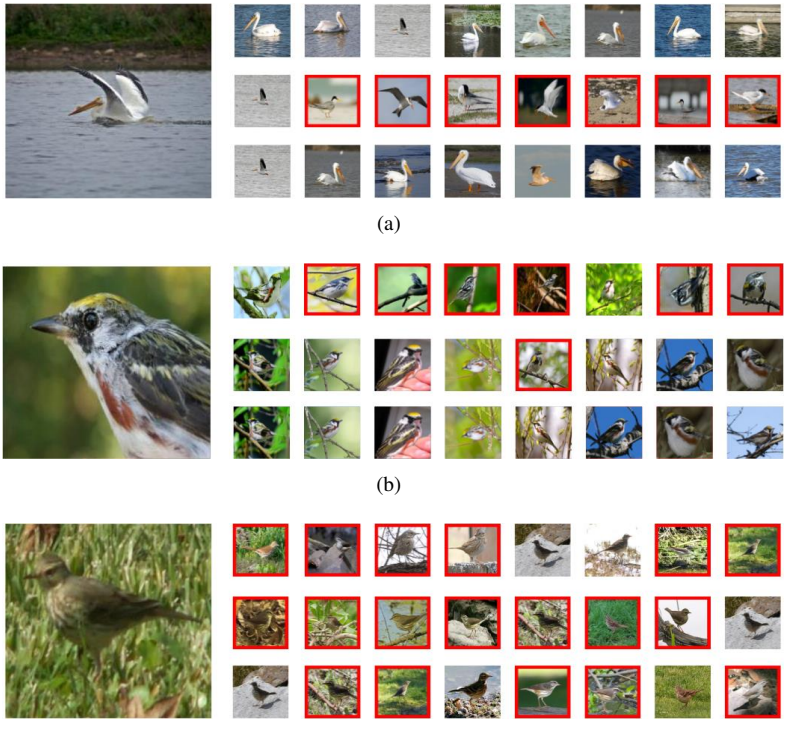
4.2.2 Ablation Studies

We show the impact of the re-ranking rules module on the retrieval performance of the Cub200 dataset in Table 4. We cancel either instances in the re-ranking rules and opti-

mize the parameters on the training dataset, and then evaluate the retrieval performance. re-ranking improves retrieval performance, especially when all three instances are applied.

4.2.3 Verify the accuracy of the re-ranking rules in three instances

We count the number of correct retrievals in the Cub200 retrieval results that satisfy the first two instances (The class of the first candidate image for both models is the same and the top K of any model has a large number of images of the same class) to verify the effectiveness of IRM^{2}R. As shown in table 5. Satisfy instance 1 or instance 2 to obtain the class for re-ranking with an accuracy of more than 92.0%, and the number also reaches a considerable amount. After the adjustment of the re-ranking rules, when the candidate image of the retrieval results is the same as this class, its priority will be increased, which is in line with the user's query habit that the correct answer comes first. As shown in Figures 3(a) and 3(b), we visualize the re-retrieval results, even though the initial retrieval result sets of the two models are quite different, the retrieval results after the re-ranking rules are still good. The application of instance 3 is shown in Figure 3(c), the initial retrieval results of the two models are quite different and do not satisfy instances 1 and 2, but still achieve good results after re-ranking.



(a)

(b)

(c)

Figure 3: **Demonstrates the top 8 retrieval results of CGD, ProxyNAC++, and IRM2R in Cub200.** Results of CGD, ProxyNCA, and our IRM^{2}R are shown from top to bottom. No boxes and red boxes denote positive and negative images, respectively.

5 Conclusion

In this paper, we propose a novel re-rank method based on two models (IRM^{2}R) to fuse the retrieval results of two models effectively. The IRM^{2}R introduces re-ranking rules to extract the correct part of the two retrieval result sets as much as possible to effectively improve retrieval performance. On the four datasets of Cub200, Cars196, Sop, and InShop, extensive experiments show our IRM^{2}R surpasses the best results CGD and ProxyNAC++ by 3.8%, 2.3%, 2.2%, and 0.5%.

6 Acknowledgement

This work was supported by National Natural Science Foundation of China (Grants No. 61976053 and No. 62171131)

7 References

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[3] Yannis Avrithis and Giorgos Tolias. Hough pyramid matching: Speeded-up geometry re-ranking for large scale image retrieval. *International journal of computer vision*, 107(1):1–19, 2014.

[4] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 1269–1277, 2015.

[5] Soumya Jyoti Banerjee, Mohammad Azharuddin, Debanjan Sen, Smruti Savale, Himadri Datta, Anjan Kr Dasgupta, and Soumen Roy. Using complex networks towards information retrieval and diagnostics in multidimensional imaging. *Scientific reports*, 5(1):1–12, 2015.

[6] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In *European Conference on Computer Vision*, pages 726–743. Springer, 2020.

[7] Zhixiang Chen, Xin Yuan, Jiwen Lu, Qi Tian, and Jie Zhou. Deep hashing via discrepancy minimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6838–6847, 2018.

[8] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image

retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 35(12): 2916–2929, 2012.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] HeeJae Jun, Byungsoo Ko, Youngjoon Kim, Insik Kim, and Jongtack Kim. Combination of multiple global descriptors for image retrieval. *arXiv preprint arXiv:1903.10663*, 2019.

[11] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[12] Seongwon Lee, Hongje Seong, Suhyeon Lee, and Euntai Kim. Correlation verification for image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5374–5384, 2022.

[13] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1096–1104, 2016.

[14] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10): 761–767, 2004.

[15] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 360–368, 2017.

[16] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465, 2017.

[17] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4004–4012, 2016.

[18] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.

[19] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115 (3):211–252, 2015.

[21] Oriane Siméoni, Yannis Avrithis, and Ondrej Chum. Local features and visual words emerge in activations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11651–11660, 2019.

[22] Fuwen Tan, Jiangbo Yuan, and Vicente Ordonez. Instance-level image retrieval using reranking transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12105–12115, 2021.

[23] Eu Wern Teh, Terrance DeVries, and Graham W Taylor. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. In *European Conference on Computer Vision*, pages 448–464. Springer, 2020.

[24] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5109–5118, 2019.

[25] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[26] Min Yang, Dongliang He, Miao Fan, Baorong Shi, Xuetong Xue, Fu Li, Errui Ding, and Jizhou Huang. Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11772–11781, 2021.