Multiple techniques to enhance performance in CNN-based Image Retrieval systems

Tan Ngoc Pham
University of Information Technology
Vietnam National University
Ho Chi Minh City, Vietnam
19520925@gm.uit.edu.vn

An Vo
University of Information Technology
Vietnam National University
Ho Chi Minh City, Vietnam
19520007@gm.uit.edu.vn

Dzung Tri Bui
University of Information Technology
Vietnam National University
Ho Chi Minh City, Vietnam
19521386@gm.uit.edu.vn

***Abstract*—Activations on Convolutional Neural Networks (CNNs) served as image descriptors have reached its peak in the field of image retrieval due to their outstanding efficiency and compactness of representation. However, there is a massive need of annotated data and high quality annotation is a significance to achieve reasonable results. Throughout this work, we do fine-tune CNNs for image retrieval system on a collection of unordered images automatically. The selection of the train data could be guided using state-of-the-art retrieval and Structure-from-Motion methods to reconstruct 3D models. We additionally apply a novel trainable Generalized-Mean pooling layer generalizing max and average pooling for a boosting in retrieval performance. And we would conduct our experiments with VGG and ResNet architectures on Oxford5k, Paris6k, $\mathcal{R}$Oxford5k and $\mathcal{R}$Paris6k benchmarks. All of our implementations toward this project is in https://github.com/DTA-UIT/ImageRetrieval_System**

***Index Terms*—Image Retrieval, Convolutional Neural Networks, Deep Learning**

I. Introduction

In the scope of image retrieval, Convolutional neural networks (CNNs) is typically used due to its attractive solution to this problem. The strength of CNNs has been widely recognized after the success of Krizhevsky et al. [1] due to the use of enormous annotated datasets, e.g. ImageNet [2]. However, the need for annotated training data is costly and often prone to errors. Fortunately, architectures that are trained for image classification problems have shown their strong adaption abilities. Using activations of CNNs, e.g. Rectified Linear Units [3], which were pretrained for image classification as image descriptors and image features have perform successfully in image search.

Initialization by a pretrained classification network and training for a different task afterwards, or fine-tuning of the network, is an alternative solution to improve the adaptation ability and save more time compared to aforementioned approaches. A method proposed by [4] is to perform fine-tuning guided by geometrically tagged image databases and directly optimize the retrieval results by choosing matched or unmatched pairs to train the neural models.

Throughout this work, we choose the approach as the unsupervised CNNs fine-tuning for image retrieval. Firstly, we harness SfM information and enforce for both hard unmatched and matched examples for CNNs training. Secondly, we let our architectures learn the whitening through the same training data to avoid the short representations that are the limitations from traditional whitening performance. We choose to use a trainable pooling layer which generalizes existing popular pooling schemes for CNNs and thus both enhances the performance and preserving the same descriptor dimensionality as well, lastly.

II. Related works

The application of CNN activations on image retrieval problem has been shown the efficiency with the evidence on high accuracy among a wide variety of previous methods. The networks were first trained for image classification problems on popular datasets, i.e. CIFAR-100, and then were taken a step further by re-training with the target's dataset.

Manual efforts are required to construct specified datasets on the target problem to perform fine-tuning and achieve outstanding performance as in the work of [5] [6]. Another astonishing point in the image retrieval field has appeared recently to be the ground for weakyly supervised fine-tuning network on using geo-tagged datasets with timestamps.

Our work was inspired from the work of [7]. It is an extension of a previous work of [8] in which they propose a novel pooling layer, a novel multi-scale image representation, and a novel query expansion method. Here, we try to reproduce the original work and make some further experiments towards the problem of image retrieving.

III. Method

A. Convolutional neural networks

There are a great number of convolutional neural networks nowadays, e.g. VGG [9], Vision Transformers [10], EfficientNet [11], ResNet [12], DenseNet [13], to be used in Image Retrieval problems and achieve astonishing results despite the fact that the fully connected layers are removed from the original architectures. And this would be the stable

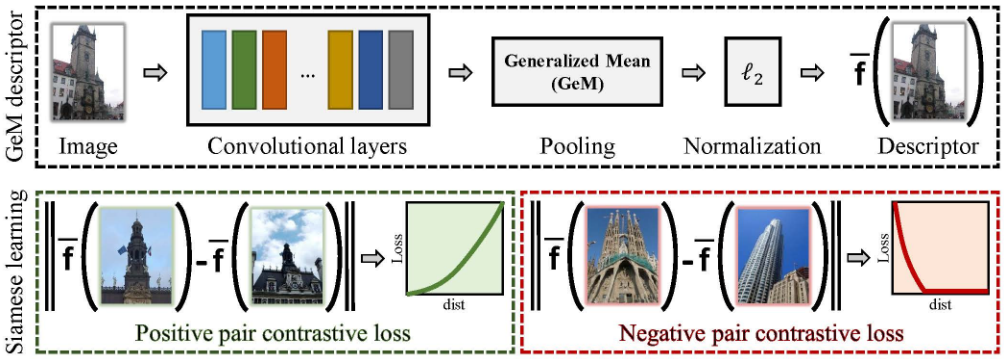

Fig. 1: Our end-to-end approach to the problem [7]

ground on approaching fine-tuning methods.

Let denote χ as the output from the given architecture, such that χ is a tensor, $\chi \in \mathbb{Z}_+^{W\times H\times K}$, and K is the number of feature maps in the last layer. The network output comprises of χ_K activation sets of $W \times H$ 2D activations for feature maps with the assumption that the final layer is a Rectified Linear Unit (ReLU) such that $\chi \in \mathbb{Z}\backslash\mathbb{Z}^-$.

B. Generalized-Mean pooling

We would add a pooling layer with χ as an input and the output for the pooling process is a vector f. The pooling step is exploited using the *Generalized-Mean* (GeM) pooling [14] using Equation 1 and 2 where $k \in \{1, ..., K\}$.

$$f^{(g)} = \left[f_1^{(g)}, ..., f_k^{(g)}, ..., f_K^{(g)}\right] \tag{1}$$

$$f_k^{(g)} = \left(\frac{1}{|\chi_k|}\sum_{x\in\chi_k} x^{p_k}\right)^{\frac{1}{p_k}} \tag{2}$$

C. Image descriptor

The final layer from our architecture consists of an L2-normalization layer. The output vector f from the above process is L2-normalized for the final evaluation with inner product between two images. GeM vector could be interpreted as the L2-normalized vector f and represents the image descriptor.

D. Siamese learning

We train a two-branch network adopted from the siamese architecture. The parameters from one network is shared with the other for each branch. The training input comprises of image pairs (i, j) and labels $Y(i, j) \in \{0, 1\}$ to indicate whether a pair is matched (0) or unmatched (1).

E. Loss function

We utilise the *contrastive loss* [15] as the loss function to show whether the pair is matching or not and expressed as in Equation 3 where $\overline{f}(i)$ is the L2-normalized GeM vector of image i and τ is the threshold to measure the distance of unmatched pairs whether it is large enough to be ignored by the loss. We train our architecture via a large number of training pairs which are automatically created, and we find that the loss function generalizes better and converges at a higher performance than the triplet loss.

$$\mathcal{L}(i,j) = \begin{cases} \frac{1}{2}||\overline{f}(i) - \overline{f}(j)||^2 & \text{if } Y(i,j) = 1 \\ \frac{1}{2}\left(max\{0, \tau - ||\overline{f}(i) - \overline{f}(j)||\}\right)^2 & \text{if } Y(i,j) = 0 \end{cases} \tag{3}$$

F. Whitening and dimensionality reduction

In this section, we would discuss about the post-processing for the fine-tuned GeM vectors. We would harness the labeled data from the 3D models and utilise the *linear discriminant projections* [16]. The projection is separated into two different parts which are whitening and rotation.

The whitening part is computed as the inverse of the square root of the matching pairs covariance matrix C_S and demonstrated as in Equation 4.

$$C_S = \sum_{Y(i,j)=1} \left(\overline{f}(i) - \overline{f}(j)\right)\left(\overline{f}(i) - \overline{f}(j)\right)^\top \tag{4}$$

The rotation part is *Principal Component Analysis* [17] (PCA) of the unmatched-pairs covariance matrix in the whitened space as in Equation 5.

$$C_D = \sum_{Y(i,j)=0} \left(\overline{f}(i) - \overline{f}(j)\right)\left(\overline{f}(i) - \overline{f}(j)\right)^\top \tag{5}$$

Although it is not completely optimized and is performed without using any batches of the training data,, our approach did use every available training pairs efficiently in the whitening optimization by optimizing the GeM descriptor first and then comes the whitening.

Model	Oxford5k	Paris6k	Oxford5k (whiten)	Paris6k (whiten)	Oxford5k (elapsed time)	Paris6k (elapsed time)
ResNet101-GeM	81.12	87.80	88.19	92.60	10m9s	12m24s
VGG16-GeM	82.44	82.28	87.23	87.80	9m48s	12m8s

TABLE I: MAP on Oxford5k and Paris6k

Model	$\mathcal{R}$Oxf(E)	$\mathcal{R}$Oxf(M)	$\mathcal{R}$Oxf(H)	$\mathcal{R}$Oxf(E) (whiten)	$\mathcal{R}$Oxf(M) (whiten)	$\mathcal{R}$Oxf(H) (whiten)	$\mathcal{R}$Oxf (elapsed time)
ResNet101-GeM	73.94	55.83	27.68	84.07	65.33	40.12	9m49s
VGG16-GeM	73.88	55.95	26.94	79.17	60.78	32.58	9m40s

TABLE II: MAP on $\mathcal{R}$Oxf

Model	$\mathcal{R}$Par(E)	$\mathcal{R}$Par(M)	$\mathcal{R}$Par(H)	$\mathcal{R}$Par(E) (whiten)	$\mathcal{R}$Par(M) (whiten)	$\mathcal{R}$Par(H) (whiten)	$\mathcal{R}$Par (elapsed time)
ResNet101-GeM	86.52	70.02	44.82	91.62	76.71	55.32	12m17s
VGG16-GeM	80.3	63.01	37.32	86.74	69.29	44.23	12m1s

TABLE III: MAP on $\mathcal{R}$Par

	E	M	H
$\mathcal{R}$Oxf - MAP@k[1, 5, 10]	[92.65, 81.1, 77.15]	[92.86, 80.95, 75.67]	[67.14, 48.71, 40.14]
$\mathcal{R}$Oxf - MAP@k[1, 5, 10] + whiten	[97.06, 90.88, 86.47]	[95.71, 90.0, 86.14]	[82.86, 66.86, 56.43]
$\mathcal{R}$Par - MAP@k[1, 5, 10]	[98.57, 95.43, 93.86]	[100.0, 98.29, 96.71]	[94.29, 87.14, 83.43]
$\mathcal{R}$Par - MAP@k[1, 5, 10] + whiten	[98.57, 96.57, 95.63]	[100.0, 98.86, 98.57]	[97.14, 90.29, 88.29]

TABLE IV: MAP@k of ResNet101-GeM on $\mathcal{R}$Oxf and $\mathcal{R}$Par

	E	M	H
$\mathcal{R}$Oxf - MAP@k[1, 5, 10]	[86.76, 85.29, 80.74]	[88.57, 85.14, 77.43]	[62.86, 47.43, 40.29]
$\mathcal{R}$Oxf - MAP@k[1, 5, 10] + whiten	[94.12, 86.69, 82.95]	[94.29, 87.81, 82.52]	[74.29, 57.52, 51.95]
$\mathcal{R}$Par - MAP@k[1, 5, 10]	[97.14, 94.86, 93.14]	[100.0, 96.86, 95.57]	[95.71, 83.71, 78.71]
$\mathcal{R}$Par - MAP@k[1, 5, 10] + whiten	[98.57, 96.29, 94.48]	[100.0, 99.43, 97.86]	[100.0, 89.14, 84.0]

TABLE V: MAP@k of VGG16-GeM on $\mathcal{R}$Oxf and $\mathcal{R}$Par

G. 3D reconstruction

In this work, we coupled *Bag-of-Words* (Bow) image retrieval and *Structure-from-Motion* (SfM) for automatically selecting our training data via the 3D reconstruction system.

On the contrary to the methods that acquires training-data for image search, we ignore the need for human-annotated data or any other assumptions for the training dataset. We could accomplish this by using the geometrical camera positions from automatically reconstructed 3D models by applying BoW-based image retrieval to collect images of the objects and exploiting the SOTA retrieval-SfM pipeline.

The SOTA retrieval-SfM pipeline takes an unordered image collection as the input and it tries to construct as many 3D models as possible. And *local features based fast image clustering* [18] is also applied to make the process more efficient. The SfM filters out almost every mismatched images and provides image-to-model matches as well as camera positions for all matched images in the cluster. The total process mentioned in this section is fully automatic.

IV. Experiments

A. Metrics

On experimental process, we use *Mean Average Precision* [19] (MAP) as the primary metric for our work.

Average precision (AP) is the average of the precision values at the points at which each relevant document is retrieved. And the MAP is the mean over the average precision value for a set of queries. MAP is the most commonly used measure since it gives us non-interpolated average precision, which captures both precision and recall and is sensitive to the rank of each relevant document. However, MAP requires many relevance judgements in text collection.

B. Implementation details

Train datasets: Our training samples are derived from the dataset of [20] which consists of 7.4 million images on the theme of popular landmarks, cities and countries around the world. The extensive retrieval Structure from Motion reconstruction of the whole dataset after the clustering procedure and removing overlapping 3D models gives us 713 3D models containing more than 163 000 unique images from the initial dataset. And the dataset also, on purpose, consists of all images from Oxford5k datasets.

Training pairs: The size of the 3D models varies from 25 to 11 000 images. 551 models for training and 162 for validation (around 133 000 images for training and 30 000 images for validation) are randomly selected. The number of training queries is from 10% per 3D model to less or equal than 30%. Each epoch has around 6000 and 1700 images

respectively.

Test datasets and evaluations: The approach is evaluated on Oxford5k [21] and Paris6k [22] datasets, as well as $\mathcal{R}$Oxford5k and $\mathcal{R}$Paris6k [23]. The performance is measured using MAP as mentioned in section IV.A. We additionally use We also apply the standard evaluation protocol to crop the query images and the cropped zone is treated as input to the architecture.

Configuration: We used pre-trained ResNet101-GeM [7] and VGG16-GeM [7] to perform the fine-tuning. We conduct our experiments using NVIDIA @ RTX 3060 GPU, 16GB RAM with 11th Gen Intel® Core™ i7-11700K @ 3.60GHz×16 CPU and PyTorch framework.

C. Results

Net	Method	F-tuned	Dim	Oxford5k	Paris6k
Compact representation using deep networks					
VGG	MAC	no	512	56.4	72.3
	SPoC	no	512	68.1	78.2
	CroW	no	512	70.8	79.7
	R-MAC	no	512	66.9	83.0
	BoW-CNN	no	na	73.9	82.0
	NetVLAD	no	4096	66.6	77.4
	NetVLAD	yes	512	67.6	74.9
	NetVLAD	yes	4096	71.6	79.7
	Fisher Vector	yes	512	81.5	82.4
	R-MAC	yes	512	83.1	87.1
	* Our	yes	512	87.23	87.80
Res	R-MAC	no	2048	69.4	85.2
	R-MAC	yes	2048	86.1	94.5
	* Our	yes	2048	88.19	92.6

TABLE VI: Comparison on our results among other SOTA approaches.

Comparision with the state-of-the-art: We compare our results with the SOTA performance on compact image representations. The results for the fine-tuned GeM based networks are given in Table VI. The methods that we used for this work has outperformed the SOTA on all datasets once the VGG architecture and initialization are applied. However, the result by the work of [24] is better than us on Paris dataset with ResNet model, but it has a worse score in Oxford dataset.

D. Deployment

We built a demo to demonstrate our work towards the problem. The total time for processing both cropping the uploaded image into the new one and processing the query is 18 seconds on average. The demo video is https://drive.google.com/file/d/1HeqgfqGmo6l2jHVyeT0AOWXnc

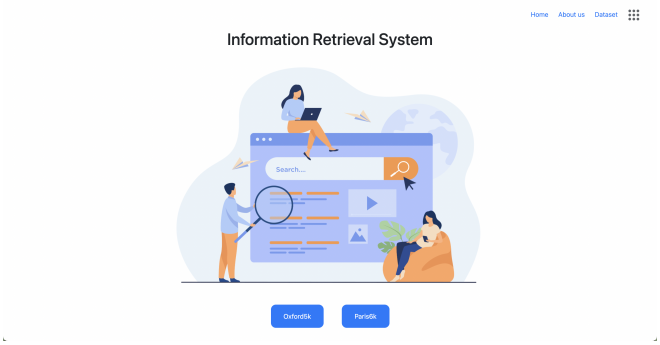

Fig. 2: Home page of our demo

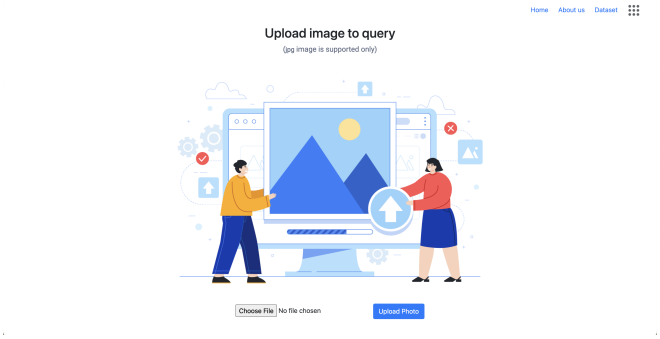

Fig. 3: Upload image to query section

Fig. 4: Confirm image and crop image page

Fig. 5: 20 most relevant images to the original query would be returned

V. Conclusions

In this work, we did apply CNNs architectures' fine-tuning for image retrieval problem. The training data are selected from an automated 3D reconstruction system applied on an

unordered photo collection of which consists buildings and popular landmarks as well as for other rigid 3D objects. The method that we used need no requirement on standard benchmark but still achieve among the best local features based systems while being faster and need less memory. In addition, the pooling layer generalizing previously adopted mechanisms is considered to enhance the retrieval accuracy and be effective for constructing a joint multi-scale representation.

ACKNOWLEDGEMENTS

The authors would like to thank the support from Dr. Thanh Duc Ngo on teaching us CS336 - Multimedia Information Retrieval as well as supervising us on this project.

REFERENCES

[1] S. I. H. Krizhevsky, Alex and G. E, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[3] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[4] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *European conference on computer vision*. Springer, 2014, pp. 584–599.

[5] T. Weyand and B. Leibe, "Discovering details and scene structure with hierarchical iconoid shift," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3479–3486.

[6] G. T. C. Radenovic, tFilip and Ondrej, "Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples," in *European conference on computer vision*. Springer, 2016, pp. 3–20.

[7] G. T. Filip Radenovic and O. Chum, "Fine-tuning cnn image retrieval with no human annotation," 2018.

[8] ——, "Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples," 2016.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[11] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[13] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2018.

[14] O. Morere, J. Lin, A. Veillard, L.-Y. Duan, V. Chandrasekhar, and T. Poggio, "Nested invariance pooling and rbm hashing for image instance retrieval," in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, 2017, pp. 260–268.

[15] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 539–546.

[16] K. Mikolajczyk and J. Matas, "Improving descriptors for fast tree matching by optimal linear projection," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[17] K. P. F.R.S., "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[18] O. Chum *et al.*, "Large-scale discovery of spatially related images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 2, pp. 371–377, 2009.

[19] L. LIU and M. T. OZSU, Eds., *Mean Average Precision*. Boston, MA: Springer US, 2009, pp. 1703–1703. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_3032

[20] J. L. Schonberger, F. Radenovic, O. Chum, and J.-M. Frahm, "From single image query to detailed 3d reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5126–5134.

[21] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *2007 IEEE conference on computer vision and pattern recognition*. IEEE, 2007, pp. 1–8.

[22] ——, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *2008 IEEE conference on computer vision and pattern recognition*. IEEE, 2008, pp. 1–8.

[23] G. T. Y. A. Filip Radenovic, Ahmet Iscen and O. Chum, "Revisiting oxford and paris: Large-scale image retrieval benchmarking," 2018.

[24] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *European conference on computer vision*. Springer, 2016, pp. 241–257.