MFC: A MULTI-SCALE FULLY CONVOLUTIONAL APPROACH FOR VISUAL INSTANCE RETRIEVAL

Jiedong Hao[1,3], Wei Wang[1], Jing Dong[1,2] and Tieniu Tan[1]*

[1]National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing 100190
[2]State Key Laboratory of Information Security, Institute of Information Engineering,
Chinese Academy of Sciences, Beijing 100093
[3]University of Chinese Academy of Sciences

jiedong.hao@cripac.ia.ac.cn, {wwang, jdong, tnt}@nlpr.ia.ac.cn

ABSTRACT

Previous work has shown that feature maps of deep convolutional neural networks (CNNs) can be interpreted as feature representation of an image. Image features aggregated from these feature maps have achieved steady progress in terms of performances on visual instance retrieval tasks in recent years. The key to the success of such methods is feature representation. In this paper, we study how to represent an image using discriminative features. We demonstrate first that image size is an important factor which affects the performance of instance retrieval but has not been thoroughly discussed in previous work. Based on experimental evaluations, we propose a multi-scale fully convolutional (**MFC**) approach to encode the image efficiently and effectively. The proposed method is simple to implement, which does not employ sophisticated post-processing techniques such as query expansion, yet shows promising results on four public datasets.

Index Terms— Visual Instance Retrieval, Image Resizing Strategy, Multi-scale Representation, Fully Convolutional Neural Network

1. INTRODUCTION

Image retrieval is an important problem for both academic research and industrial applications. Two slightly different meaning are referred when we see image retrieval in the literature. The first one is category-level image retrieval, where an image in the returned results is deemed to be true positive if it shares the same broad class (e.g., car, cat, dog) with the query image. The second one is instance-level image retrieval, in which an image is considered to match the query if they both contain the same object or scene. In this paper, we focus on instance-level image retrieval. Although it has been studied for many years [1, 2, 3, 4], instance-level image retrieval is still challenging because the scales, orientations, lighting conditions, etc. of the same object instance in different images often have large variations.

Traditionally, the most popular approach is bag of features (BoF) based method which relies on local feature descriptors such as SIFT[5]. In order to boost the retrieval performances, post-processing techniques such as query expansion [6] and spatial verification [2] are also employed. Since the decisive victory [7] over traditional methods in the ImageNet image classification challenge, researchers in the field of visual instance retrieval also shift their interest to CNNs. Their experiments have shown some promising and surprising results [8, 9, 4], which, in the case of global image descriptors without post-processing, are on par with or surpass the performances of traditional methods like BoF and vector of locally aggregated descriptors (VLAD) [10].

Despite all these previous advances on using CNNs for image feature representation, the underlying factors that affect its effectiveness on image retrieval tasks are still not thoroughly explored, e.g., *what is the best strategy for resizing the input image? What are the impacting factors that affect the performance of multi-scale method?* Clarifying these questions will help us advance further towards building a more robust and accurate retrieval system.

In this paper, we aim to answer the questions in the previous paragraph and make three novel contributions. Unlike other papers, we explicitly study the impact of image size on retrieval performances by utilizing a fully convolutional network. We also experiment with different strategies to learn the PCA and whitening matrix for dimension reduction. During experiments, we borrow wisdom from literature and evaluate their usefulness, but find that they are not as effective as some of the simpler design choices. Second, by combining what we have learned during experiments, we propose a new multi-scale image representation, which is compact yet effective. Finally, we evaluate our method on four challeng-

*Corresponding author

This work is supported by NSFC (No. U1536120, U1636201, 61502496), the National Key Research and Development Program of China (No. 2016YFB1001003) and Beijing Natural Science Foundation (No. 4164102).

978-1-5090-6067-2/17/$31.00 ©2017 IEEE

ing datasets, *i.e.*, Oxford5k, Paris6k, Oxford105k and UKB, and achieve very promising results.

2. RELATED WORK

In this section, we briefly review the papers that are directly related to our work.

Multi-scale image representation. Lazebnik et al. [11] propose spatial pyramid matching (SPM) to encode the spatial information of images. They represent an image using a pyramid of several scales. Features from different scales are combined to form the image representation in such a way that coarser levels get less weight while finer levels get more. He et al. [12] introduce an approach called spatial pyramid pooling (SPP), in which output feature maps of the last convolutional layer are divided into a pyramid of several scales. Both the region-level and scale-level features are concatenated to form a fixed-length vector. In our paper, we explore whether it is viable to use deep features in the manner of SPM or SPP. We find that deep features are distinct from traditional local descriptors. None of the above two methods outperforms a simpler feature aggregation method.

Off-the-shelf CNNs for feature representation. Gong et al. [13] introduce multiple orderless pooling to represent an image for the purpose of increasing the invariance of image features. This method is rather complicated and time-consuming. At the same time, Babenko et al. [8] shows that the output of fully-connected layers of AlexNet [7] can be used for retrieval tasks. [14] use the weighted sum of feature maps of last convolutional layer as the image features and achieve improved results. Tolias et al. [4] shows that max-pooling of feature maps is better than sum-pooling [14]. They employ a multi-scale method, called regional maximum activation of convolutions (R-MAC). In this method, the feature maps are divided into 3 scales, each with several overlapping square regions. The number of regions per scale is decided by the aspect ratio of the original image. In our paper, we use a simpler strategy in which the region number in each scale is fixed and the size of each region varies according to the image aspect ratio. All the previous work failed to discuss the impact of image size on performances. [13] and [8] just fix the image size. [14] and [4] also use the fully convolutional networks as we do, but they either use an enlarged version of fixed image size [14] or set the maximum size of the image to a fixed value [4]. Different from these work, We explicitly explore and analyze the impact of different image resizing strategies in our work.

It should be noted that we use off-the-shelf CNNs without further fine-tuning as opposed to the recent work of fine-tuning for instance retrieval [15]. Although end-to-end training boosts retrieval performances, unsupervised method is still an viable solution if the acquirement of domain specific training data is difficult and expensive. Our work can also be readily adapted to the framework of end-to-end training for image retrieval.

3. THE PROPOSED MFC APPROACH

3.1. Background

In this paper, we focus on extracting compact and discriminative image features using the off-the-shelf CNNs in an efficient way. For an image I, we simply subtract the mean value of the RGB channels from the original image and do not employ other sophisticated image pre-processing techniques. Then the image is fed into the convolutional network and goes through a series of convolutions, non-linear activation and pooling operations. The feature activation maps of a certain layer can be interpreted as the raw image features, based on which we build the final image representation. The feature maps form a tensor of size $C \times H \times W$, where C is the number of feature channels, and H and W are the height and width of a feature map. If we represent the set of feature maps as

$$F = \{F_i\}, i = 1, 2, \ldots, C\,, \tag{1}$$

where F_i represents the i^{th} activation feature map. Then the most simple image feature is formulated as:

$$f = [f_1, f_2, \ldots, f_i, \ldots, f_C]^T\,, \tag{2}$$

where f_i is obtained by applying max-pooling on the i^{th} feature map F_i. Throughout this paper, we use feature maps after the ReLU activation so that the elements in each feature map are all non-negative. We also experiment with feature maps prior to ReLU, but find that they lead to inferior results. After the image feature representation is obtained, post-processing techniques such as PCA and whitening can be further applied on these features.

3.2. Image Resizing Strategy

Researchers in various fields including instance retrieval often employ popular pre-trained models such as AlexNet [7] or VGGNet [16] and adapt these networks for their specific needs. In order to meet the network's requirement that input images should have fixed size, previous work on retrieval [13, 14] usually resizes the input images to a fixed size. We postulate that the resizing operation may lead to loss or distortion of important information about object instances in natural images. Ultimately, this operation will hurt the discriminative power of image features extracted from the network, thus degrading the retrieval performances. For the task of instance-level retrieval, we think it is better to keep the images their original sizes and feed them directly to the network whenever possible. In this paper, three kinds of image resizing strategies are explored:

- Both the height and width of images are set to the same fixed value. We denote this as *two-fixed*, *i.e.*, $W =$

(a) scale 1 (b) scale 2 (c) scale 3

Fig. 1: An illustration of 3 scale representation of an image.

$H = s$, (s is a constant, and W, H are the width and height of an image).

- The minimum size of an image is set to a fixed value and the aspect ratio of the original image is kept the same. We denote this as *one-fixed*, *i.e.*, $\min(W, H) = s$.
- The images are kept their original sizes. We denote this as *free*.

3.3. Multi-scale Feature Representation

Unlike local feature descriptors such as SIFT [5], feature vectors extracted from CNNs are global descriptors which encode the general information of image regions. The feature vectors are related more to the semantic class of the image and lack the finer and local information needed to accurately assess the similarity between two images. Inspired by previous work on multi-scale image representation [11, 12], we explore the feasibility of combining this powerful method with the suitable image resizing strategy to obtain discriminative image features. In our approach, an image is represented by a L-scale pyramid, and at each scale, the image is divided evenly into several overlapping or non-overlapping regions, see Fig 1 for an example of 3 scale representation of an image. Then the vector representations of these small regions are computed. After that, regional vectors from the same scale are combined to form the scale-level features, which are then combined and l_2-normalized to form the final image feature. We evaluated 3 different approaches for combining the region-level and scale-level features. The first method is in the same vein to spatial pyramid pooling [12], *i.e.*, regional vectors from all scales are first l_2-normalized and then concatenated to form a high dimensional vector. The dimensionality of the final feature is

$$D = N \times C\,, \tag{3}$$

where N denotes the total number of regions in all scales and C is the number of feature channels of a particular layer. In the second method, region-level features are summed and l_2-normalized to form the scale-level features, which are then concatenated to form the final image feature. The dimensionality of the final feature is given by

$$D = L \times C\,, \tag{4}$$

where L is the number of scales. In the last approach, the regional vectors in each scale are added together and l_2-normalized to form the scale-level features, which are subsequently combined and l_2-normalized to form the image representations. The final image feature dimensionality is equal to C.

Now, we explain more about how to compute the region features efficiently. If we directly feed each small image region into the network to extract its feature, the time cost would be huge, thus unacceptable for instance retrieval tasks. Inspired by Fast R-CNN [17] and R-MAC [4], we assume a linear projection between the original image regions and the regions in the feature maps of a certain layer. Then the regional features are efficiently computed without re-feeding the corresponding image regions. In our experiment, various settings for the multi-scale and scale-level feature combination methods are explored and their retrieval performances are reported and analyzed.

3.4. PCA and whitening

PCA and whitening. Principal Component Analysis (PCA) is a simple but effective method for reducing the dimensionality of feature vectors and decorrelating the feature elements. Previous work [8] has shown evidences that PCA and whitened features can actually boost the performances of image retrieval. In this paper, we show our new findings.

4. EXPERIMENT

4.1. Implementation Details

We use the open source deep learning framework Caffe [18] for our whole experiments. The popular VGG19 network [16] trained on ImageNet is used as our model. The feature maps of the last convolutional layers is used to extract the image features.

Network transformation. The original VGG19 network only accepts an image of fixed size, which is not the optimal choice when extracting image features for retrieval tasks. In order to process images with various sizes and to evaluate different input image resizing strategies, we transformed the original network into its fully convolutional version.

4.2. Datasets and Evaluation Metrics

The **Oxford5k** dataset [2] contains 5,062 images of 11 Oxford landmarks. A total of 55 queries with their ground truth relevant image lists, are provided. For each query, a bounding box annotation is also provided to denote the query region.

Method	full-query	cropped-query
two-fixed	55.5 (864)	38.7 (896)
one-fixed	59.0 (800)	39.3 (736)
free	58.0	52.6

Table 1: Comparison between different image resizing strategies. The numbers in the parentheses denote the sizes in which the maximum mAPs are achieved.

We report results using both the full query images (denoted as full-query) and image regions within the bounding boxes of the query images (denoted as cropped-query).

The **Paris6k** dataset [19] includes 6,412 images of 11 landmark buildings and the general scenes from Paris. Similar to the Oxford5k dataset, 55 queries belonging to 11 groups and the ground truth bounding boxes for each query are provided .

The **Oxford105k** dataset contains the original Oxford5k dataset and additional 100,000 images [2]. The 100,000 images are disjoint with the Oxford5k dataset and are used as distractors to test the retrieval performance when the dataset scales to larger size.

The **UKB** dataset [20] consists of 10,200 photographs of 2,550 objects, with each object having exactly 4 images. The pictures of these objects are all taken indoor with large variations in orientation, scale, lighting and shooting angles. During experiment, each image is used to query the whole dataset.

Evaluation Metrics. The performance of various methods on the first three datasets is evaluated using mean average precision (mAP), while the performance on the UKB dataset is evaluated using $4 \times \text{recall@4}$.

4.3. Results and Discussion

Dataset	$s \leq 500$	$500 < s \leq 800$	$s > 800$
Oxford5k	0.87	96.86	2.27
Paris6k	1.30	95.59	3.11
Flickr100k	1.49	93.91	4.60

Table 2: The proportion (%) of minimum size images on three datasets. s represents the minimum size of an dataset image.

Image resizing. We experiment with the three image resizing strategies described in section 3.2 on the Oxford5k dataset. For the *two-fixed* and *one-fixed* strategy, grid search is used to find the optimal size in terms of retrieval performances. We find that in general increasing the image size will increase the performance of extracted features for both cropped and full query. Table 1 shows the best performance obtained using the three strategies. The *free* input strategy performs best in the cropped-query case. In the full query case, *one-fixed* way is slightly better. But in that setting, the minimum size of dataset images is fixed at 800, which significantly increase the computational cost of feature extraction since the minimum size of most dataset images is below 800. Table 2 shows the statistics of the minimum size of images in three datasets. By employing the *free* way, we can reduce the computational cost of extracting image features and get good retrieval performances in the mean time.

The experimental result suggests that changing the image aspect ratio (*two-fixed*) distorts the image information, thus reducing the performance dramatically. The *one-fixed* way is better than the *two-fixed* method. But information loss still occurs due to the resizing operation. The *free* way is able to capture more natural and undistorted information from the images, which explains its superior performance over the other two methods.

	scale	overlap	weighing	version	full	cropped
(a1)	2	×	×	-	63.5	59.0
(a2)	2	×	✓	-	63.9	61.0
(b1)	3	×	×	-	64.2	60.9
(b2)	3	×	✓	-	62.6	61.0
(b3)	3	s2	×	-	64.8	60.8
(c1)	4	s3	×	v1	65.1	61.4
(c2)	4	s3	✓	v1	64.8	60.7
(c3)	4	s2,s3	×	v1	65.5	60.8
(c4)	4	s2,s3	×	v2	65.9	61.5
(c5)	4	s2,s3	✓	v2	65.4	61.2
(c6)	4	×	×	v3	64.5	61.3
(c7)	4	s3	×	v3	65.8	62.2
(c8)	4	s2,s3	×	v3	**66.3**	**62.6**

Table 3: Comparison between various settings of multi-scale representation. "overlap" denotes whether the regions in each scale have overlapping areas. "s2","s3" mean that overlap occurs in level 2 or 3. "weighing" means if the features from each scale are added using same or different weight. "version" means the different choice of the number of regions in each scale.

The benefit of multi-scale representation. We first conduct experiment to compare the three methods for combining region-level and scale-level features, which are detailed in section 3.3. Experimental results show that the first two methods are inferior to the third one. The performance drop for the first in the case of cropped-query can be as large as 41%. The dimensionality of features obtained by the first two methods is also significantly higher (at least 1.5k) than the third one (512 before employing dimension reduction). Higher dimensionality will lead to longer query times. Considering all these, we choose the third method for combining features.

We conduct extensive experiments to decide the best configurations for the multi-scale approach and report our results in Table 3. We first explore the impact of the number of scales

method	D	Oxford5k		Paris6k		Oxford105k		UKB
		full	cropped	full	cropped	full	cropped	
Razavian et al. [21]	256	58.9	-	57.8	-	-	-	3.65
SPoC [14]	256	58.9	53.1	-	-	57.8	50.1	3.65
Neural codes[8]	512	55.7	-	-	-	52.2	-	3.56
R-MAC [4]	512	-	66.8	-	83.0	-	61.6	-
Ours	256	72.2	68.4	82.5	83.4	68.0	62.9	3.75
Ours	512	73.0	70.6	82.0	83.3	68.9	65.3	3.75

Table 4: Comparison with other methods.

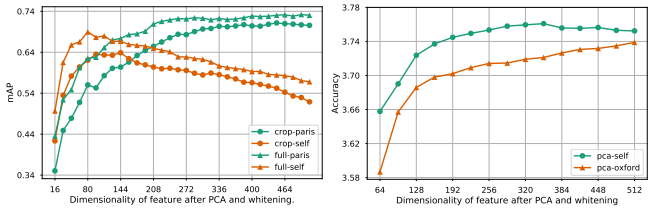
on retrieval. For the 2 and 3 scale case, The region number for each scale are $\{1 \times 1, 2 \times 2\}$ and $\{1 \times 1, 2 \times 2, 3 \times 3\}$. For the 4 scale case, 3 versions are used. They differ in the number of regions in each scale: for "v1", "v2", and "v3", the number of regions are $\{1 \times 1, 2 \times 2, 3 \times 3, 4 \times 4\}$, $\{1 \times 1, 2 \times 2, 3 \times 3, 5 \times 5\}$ and $\{1 \times 1, 2 \times 2, 3 \times 3, 6 \times 6\}$. Table 3 (a1)(b1)(c6) show the performances of using 2, 3 and 4 scales to represent the dataset images, respectively. Clearly, more scale improve the results and in the case of cropped-query, increase the performance by 3.9%.

Originally, scale-level features are summed up using the same weight. So we conduct experiment to study whether weighing the different scale-level features will lead to improved performance. The weighing method for features from different scales is similar to the manner of SPM [11], *i.e.*, features from coarser level are given less weight while features from the finer levels are given more weight. Suppose the features from different scales for an L scale representation are $f^1, f^2, \ldots, f^L$, the final image feature f is expressed as:

$$f = \frac{1}{2^{L-1}} f^1 + \sum_{i=2}^{L} \frac{1}{2^{L-i+1}} f^i \,. \tag{5}$$

For more details about SPM, we refer the readers to [11]. Comparing the results of row (a1) and (a2), it seems that weighing different scales leads to better performance. But after more experiments, we find that the weighing method generally leads to inferior results as the number of scales increase, *e.g.*, compare the results of row pair(b1)(b2) and (c1)(c2). These results suggest that deep features are different from the traditional local feature descriptors. We should exercise with caution when we apply the traditional wisdom used with SIFT to the deep convolutional descriptors, which is also suggested in [14]. Based on the result of this experiment, we do not use scale-level feature weighing in computing the final image features.

We then look into the issue of region overlapping between different scales and try to verify its usefulness. For each scale and its different versions, we set some overlapping areas between the neighbouring regions in either one or two scales of the pyramid. From the row pair (b1)(b3) and (c1)(c3), we observe that overlapping increase the performance for full-query but decrease that for cropped-query. But for 4 scale v3 (note the pair(c7)(c8)), we see a consistent improvement for both the full and cropped queries. So we use region overlapping in scale 2 and 3 in computing the final features.



(a) Oxford5k (b) UKB

Fig. 2: Relationship between the number of principal component preserved and mAP when learning the PCA and whitening matrix on the same or different dataset for the Oxford5k (**Left**) and UKB (**Right**) dataset.

PCA and whitening. We perform PCA and whitening experiment on Oxford5k and UKB dataset. For Oxford5k, the PCA and whitening matrix is learned either from the Oxford5k (pca-self) or the Paris6k (pca-paris) dataset, while for UKB, it is learned from the UKB (pca-oxford) or the Oxford5k (pca-oxford) dataset. We find that for Oxford5k dataset, learning the PCA and whitening matrix on Paris6k is beneficial. But for UKB, since it is different from Oxford5k and Paris6k in style, learning the PCA and whitening matrix on Oxford5k turns out to be harmful for the performances. Figure 2 shows this phenomenon clearly. We also observe a similar trend when leaning PCA and whitening matrix on self or Oxford5k for the Paris6k dataset.

4.4. Comparison with Other Methods

Based on previous experimental results and our analysis of different impacting factors on the retrieval performances, we propose **MFC** — a new multi-scale fully convolutional feature representation approach. For an image in the dataset, **MFC** proceeds by first feeding it into the network without resizing (the *free* way). A 4-scale image representation is built

on top of the feature maps of the last convolutional layers. In the multi-scale representation step, max-pooling of feature maps and region overlapping are used and both the region-level as well the scale-level features are summed and then l_2-normalized. After that, we apply PCA and whitening on the features from the first step. The datasets where we learn the PCA and whitening matrices are as follows: for Oxford5k and Oxford105k, it is Paris6k, while for Paris6k and UKB, it is Oxford5k and UKB respectively. The final PCA and whitened image features are used for reporting our method's performances.

We compare the performance of our method with several state-of-the-art methods which use small footprint representations and do not employ complicated post-processing techniques such as geometric re-ranking [2] and query expansion [6]. The results of using different feature dimensionality are shown in Table 4. In all the datasets tested, our method shows promising results compared to other methods with comparable cost.

5. CONCLUSION

In this paper, we introduce **MFC** — a multi-scale fully convolutional approach for instance-level image retrieval. Our method is built by a systematic evaluation of some of factors that affect the discriminative power of image features. We find that image resizing strategies affect the retrieval performances greatly. The effectiveness of our method is verified on four public datasets used for instance retrieval. Experimental results shows that our method is promising. We consider adapting **MFC** to the framework of end-to-end training for instance retrieval in our future work.

6. REFERENCES

[1] Josef Sivic and Andrew Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*, 2003.

[2] 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*, 2007.

[3] Wei Zhang and Chong-Wah Ngo, "Topological spatial verification for instance search," *IEEE Trans. Multimedia*, vol. 17, pp. 1236–1247, 2015.

[4] Giorgos Tolias, Ronan Sicre, and Hervé Jégou, "Particular object retrieval with integral max-pooling of cnn activations," *CoRR*, vol. abs/1511.05879, 2015.

[5] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[6] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *2007 IEEE 11th International Conference on Computer Vision*, 2007.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[8] Artem Babenko, Anton Slesarev, Alexander Chigorin, and Victor S. Lempitsky, "Neural codes for image retrieval," *CoRR*, vol. abs/1404.1777, 2014.

[9] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014.

[10] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[11] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *European Conference on Computer Vision*, 2014.

[13] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *ECCV*, 2014.

[14] Artem Babenko and Victor S. Lempitsky, "Aggregating local deep features for image retrieval," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[15] Albert Gordo, Jon Almazán, Jérôme Revaud, and Diane Larlus, "Deep image retrieval: Learning global representations for image search," in *ECCV*, 2016.

[16] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[17] Ross Girshick, "Fast r-cnn," in *International Conference on Computer Vision (ICCV)*, 2015.

[18] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B. Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM Multimedia*, 2014.

[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*, 2008.

[20] David Nistér and Henrik Stewénius, "Scalable recognition with a vocabulary tree," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006.

[21] Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson, "Visual instance retrieval with deep convolutional networks," *CoRR*, vol. abs/1412.6574, 2014.