MAX-SIFT: FLIPPING INVARIANT DESCRIPTORS FOR WEB LOGO SEARCH

Lingxi Xie[1], *Qi Tian*[2], *and Bo Zhang*[3]

[1,3]State Key Laboratory of Intelligent Technology and Systems (LITS)
[1,3]Tsinghua National Laboratory for Information Science and Technology (TNList)
[1,3]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[2]Department of Computer Science, University of Texas at San Antonio, Texas, USA
[1]198808xc@gmail.com,[2]qitian@cs.utsa.edu, [3]dcszb@mail.tsinghua.edu.cn

ABSTRACT

Logo search is widely required in many real-world applications. As a special case of near-duplicate images, logo pictures have some particular properties, for instance, suffering from *flipping* operations, *e.g.*, geometry-inverted and brightness-inverted operations. Such operations completely change the spatial structure of local descriptors, such as SIFT, so that image search algorithms based on Bag-of-Visual-Words (BoVW) often fail to retrieve the flipped logos.

We propose a novel descriptor named Max-SIFT, which finds the maximal SIFT value sequence for detecting flipping operations. Compared with previous algorithms, our algorithm is extremely easy to implement yet very efficient to carry out. We evaluate the improved descriptor on a large-scale Web logo search dataset, and demonstrate that our method enjoys good performance and low computational costs.

Index Terms— Large-Scale Image Search, Max-SIFT, Flipping Invariant, Experiments.

1. INTRODUCTION

As a special case of near-duplicate image search, logo search implies a wide range of commercial applications. Logo pictures have some particular properties, such as rigidity, which make it easy to detect and describe them with local features. However, there often exist flipping operations, *e.g.*, geometry-inverted or brightness-inverted, which change the spatial structure as well as relative intensity of the small patches on the image. Most often, local descriptors such as SIFT [1] are not stable throughout the flipping operations, which limits the ability of image search systems based on the Bag-of-Visual-Words (BoVW) model [2]. Figure 1 illustrates two examples that we could hardly find efficient SIFT matches between the geometry-inverted and/or brightness-inverted images and the original images.

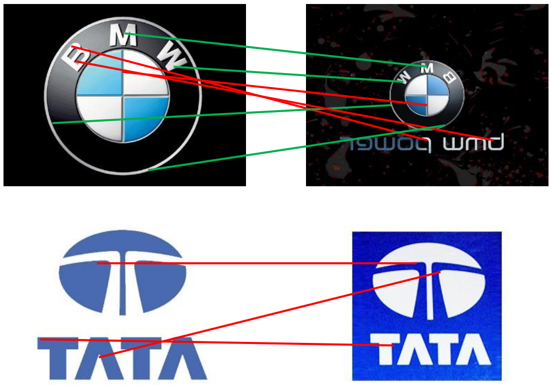

Fig. 1. SIFT descriptors [1] are less robust when the images are geometry-inverted or brightness-inverted. Red and green line segments indicate the incorrect and correct feature matches detected by SIFT descriptors, respectively.

In this paper, we propose Max-SIFT, a specialized descriptor designed for large-scale Web logo search. Our algorithm is inspired by answering the question: how is a SIFT descriptor changed when the image is flipped? We provide an insightful observation to answer the question in Section 3.1. Based this, we divide the 128 indexes of SIFT descriptors into 4 groups according to the correspondence throughout the flipping operations, and find the one with the maximal intensity value sequence for flipping operation detection and reversion. In comparison with the previous algorithms [3] [4] [5], our algorithm is extremely easy to implement yet very efficient to carry out: one needs only few lines of codes to compute the Max-SIFT descriptors, and Max-SIFT requires merely no extra time and memory overheads beyond original SIFT descriptors. We evaluate our algorithm on the CarLogo-51 dataset [6], and verify that the Max-SIFT descriptors enjoy good performance and low computational costs.

The remainder of this paper is organized as follows. Section 2 gives a brief overview of the image search pipeline. The Max-SIFT descriptor is then introduced in Section 3. After the experimental results are shown in Section 4, we draw the conclusion in Section 5.

978-1-4799-5751-4/14/$31.00 ©2014 IEEE

2. THE IMAGE SEARCH PIPELINE

In this section, we provide a brief overview of the image search pipeline based on the Bag-of-Visual-Words (BoVW) model and the inverted index structure.

We start from an image $\mathbf{I}$, which is a $W \times H$ matrix $\mathbf{I} = (a_{ij})_{W\times H}$. The local descriptors are extracted on the regions-of-interest (ROI) of images. For ROI detection, popular algorithms include DoG [1], MSER [7], and Hessian Affine [8] operators. For local patch description, we can use SIFT [1], SURF [9] or BRIEF [10] descriptors. Either combination of ROI detection and patch description yields a set of local descriptors $\mathcal{D} = \{(\mathbf{d}_1, \mathbf{l}_1), (\mathbf{d}_2, \mathbf{l}_2), \ldots, (\mathbf{d}_M, \mathbf{l}_M)\}$. where $\mathbf{d}_m$ and $\mathbf{l}_m$ denote the m-th description vector and the corresponding location on the image, respectively.

After descriptors have been extracted, they are often quantized to be compact. Generally, there are two ways of compressing descriptors into visual words. The Vector Quantization (VQ) method requires training a codebook $\mathbf{B}$ with the descriptors sampled beforehand. Approximated K-Means algorithms [11][12] are often adopted. The descriptors are then encoded by the nearest codeword(s) [11][13]. As an alternative choice, the Scalar Quantization (SQ) [14] method encodes descriptors without training a codebook. A D-dimensional SIFT descriptor can be transformed into a bit vector (elements are either 0 or 1) directly by setting a threshold and perform bitwise binarization. Denote the set of local features as $\mathcal{F} = \{(\mathbf{f}_1, \mathbf{l}_1), (\mathbf{f}_2, \mathbf{l}_2), \ldots, (\mathbf{f}_M, \mathbf{l}_M)\}$.

Large-scale image search often requires finding features' nearest or approximate nearest neighbors in a very short time. Therefore the inverted index [2][12][15] is often adopted as an efficient and scalable data structure for storing a large number of images and features. In essence, the inverted index is a compact linked list representation of a sparse matrix, in which rows and columns denote features and images, respectively. Each feature entry is followed by a list of image IDs to record its occurrences. In the online retrieval stage, we need only to check those images sharing common features with the query image. Therefore, the number of enumerated candidate images is greatly reduced with such representation.

Given a query image, local features are also extracted and then used to look up the inverted index. The retrieved image candidates are then ranked according to their frequencies of occurrence. To improve the search quality, post-processing methods are widely adopted. Among these, query expansion [15][16][17] reissues the initial top-ranked results to find valuable features which are not present in the original query; spatial verification [18][12][19][20] filters those false-positives by checking the geometric consistency of matched features, or extracts visual phrases [21][22] to verify matches on the more robust feature groups; diffusion-based algorithms [23][24] propagate affinities or beliefs via the graph structure to capture the high-level connections between images; also there are other methods [25][26][27][6].

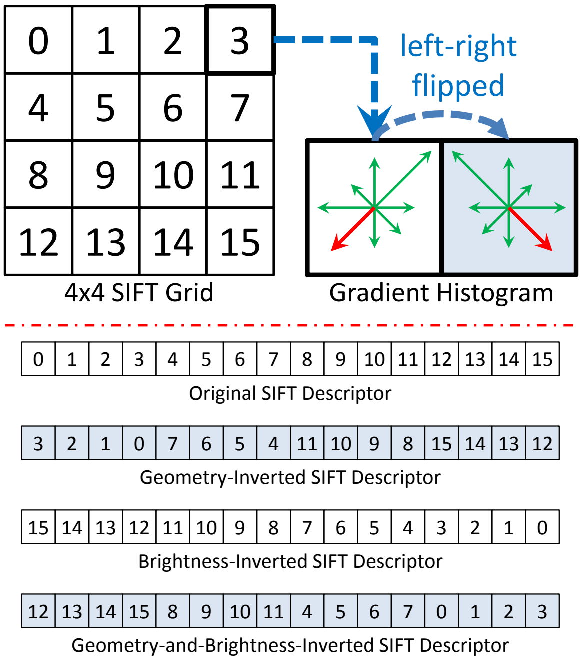

Fig. 2. The impact of flipping operations on local descriptors. The grids with light blue background indicate those bins in which the order of gradient values are flipped.

3. THE MAX-SIFT DESCRIPTOR

This section presents the Max-SIFT descriptor. Our method is inspired by the observation of how a SIFT descriptor is changed by the geometry/brightness-inverted operations. Based on this, we propose a straightforward solution which cancels up the impact and achieves flipping invariance.

3.1. Flipping Operations and Index Groups

We consider two kinds of flipping operations mentioned above, *i.e.*, the geometry-inverted and brightness-inverted operations. For the same local patch, the impact of flipping operations could be illustrated in Figure 2. One can easily observe that the flipping operations, either geometry-inverted, brightness-inverted or geometry-and-brightness-inverted, only change the order of 16 spatial bins in SIFT and/or the orders of gradient values in each bin of the descriptor. Taking the fourth bin (#3, emphasized in Figure 2) in the original SIFT descriptor as the example. When the image is geometry-inverted, the fourth bin is moved to the first position in SIFT (#0), and the gradients in the bin changes from $(0, 1, 2, 3, 4, 5, 6, 7)$ to $(0, 7, 6, 5, 4, 3, 2, 1)$.

In formal, let us denote the original SIFT descriptor as $\mathbf{d} = (d_0, d_1, \ldots, d_{127})$, where we have $d_{i\times 8+j} = a_{i,j}$ for $i = 0, 1, \ldots, 15$ and $j = 0, 1, \ldots, 7$. With the illustration in Figure 2, we could map each index (0 to 127) of the original descriptor to other three indexes with three flipping operations. Taking d_{27} ($a_{3,3}$, the red arrow in Figure 2) as the example. The same gradient value would appear at d_5 ($a_{0,5}$), d_{99} ($a_{12,3}$) and d_{125} ($a_{15,5}$) when the image (descriptor) is geometry-inverted, brightness-inverted and geometry-and-brightness-inverted, respectively. We denote the mapping as $f^{\mathrm{G}}(27) = 5$, $f^{\mathrm{B}}(27) = 99$ and $f^{\mathrm{GB}}(27) = 125$, respectively, and hence the 27-th, 5-th, 99-th and 125-th indexes form an **index group** of the descriptor. Obviously, we could divide each SIFT descriptor (128 dimensions) into 32 exclusive index groups, and the sequence of smallest indexes in the 32 groups is $\mathcal{L} = \{0, 1, \ldots, 15, 32, 33, \ldots, 47\}$. Please note that the elements in $\mathcal{L}$ is ordered, so that we can mention that $L_0 = 0$, $L_{31} = 47$, etc. We can thereafter define the flipped index sequences $\mathcal{L}^{\mathrm{G}}$, $\mathcal{L}^{\mathrm{B}}$ and $\mathcal{L}^{\mathrm{GB}}$ using the principle like $\mathcal{L}^{\mathrm{G}} = \left\{L_s^{\mathrm{G}} = f^{\mathrm{G}}(L_s), s = 0, 1, \ldots, 31\right\}$. The sequences $\mathcal{L}$, $\mathcal{L}^{\mathrm{G}}$, $\mathcal{L}^{\mathrm{B}}$ and $\mathcal{L}^{\mathrm{GB}}$ are exclusive.

3.2. The Max-SIFT Descriptor

With the definition of index sequences, we would introduce the Max-SIFT descriptor in a straightforward way. Essentially speaking, the Max-SIFT descriptor works by comparing the **index sequences** of the original SIFT descriptor, *i.e.*, $\mathcal{L}$, $\mathcal{L}^{\mathrm{G}}$, $\mathcal{L}^{\mathrm{B}}$ and $\mathcal{L}^{\mathrm{GB}}$. For this, we define the **value sequences**, *i.e.*, $\mathcal{V}$, $\mathcal{V}^{\mathrm{G}}$, $\mathcal{V}^{\mathrm{B}}$ and $\mathcal{V}^{\mathrm{GB}}$. For a given descriptor $\mathbf{d}$, the value sequence $\mathcal{V} = \{v_0, v_1, \ldots, v_{31}\}$ is calculated according to the index sequence $\mathcal{L}$ so that $v_s = d_{L_s}$ for $s = 0, 1, \ldots, 31$. Other value sequences $\mathcal{V}^{\mathrm{G}}$, $\mathcal{V}^{\mathrm{B}}$ and $\mathcal{V}^{\mathrm{GB}}$ are similarly calculated. Then we find the sequence with the maximal alphabetical order, obtaining the **dominant** value sequence, say $\mathcal{V}^{\mathrm{B}}$, implying that the image is inverted by the detected transform, *i.e.*, brightness-inverted operation in this case. The descriptor is then reverted with the same operation. In other words, we guarantee that the sequence $\mathcal{V}$ in the Max-SIFT descriptor being dominant. This is equivalent to performing the detected flipping operation once again on the descriptor.

One can easily observe that the Max-SIFT descriptor is consistent throughout the flipping operations. It is equivalent to calculate the original descriptor in 4 different ways, *i.e.*, no change, geometry-inverted, brightness-inverted and geometry-and-brightness-inverted, and Max-SIFT is just the one with the largest alphabetical value sequence. For a SIFT descriptor and its variations after any flipping operations, the Max-SIFT should be the same.

3.3. Comparison to Previous Works

Here we would like to discuss the relationship between the Max-SIFT descriptor and other flipping invariant descriptors, *i.e.*, the MI-SIFT [3], FIND [4] and F-SIFT [5] descriptors. The MI-SIFT descriptor [3] considers both the geometry-inverted and brightness-inverted operations on the descriptor, and replaces the values in the bins with symmetric values, making the descriptor invariance throughout the flipping operations. The FIND descriptor [4] defines another set of spatial bins to calculate descriptor values, and determines the order of spatial bins according to the intensity of gradients next to the dominant orientation. The F-SIFT descriptor [5] ensures the flipping invariance of local features by enforcing the flows of all the regions should follow a pre-defined direction indicated by the sign of a Gaussian kernel. The proposed Max-SIFT descriptor is very similar to MI-SIFT [3], and the only difference is that we do not use advanced arithmetic operations to generate the invariance, but simply rearrange the groups of bins to map all the flipped versions of descriptors onto the same one. Our solution is even more efficient as we do not require any floating calculations, and we shall see in the experimental results that this simple idea works extremely well on the logo images.

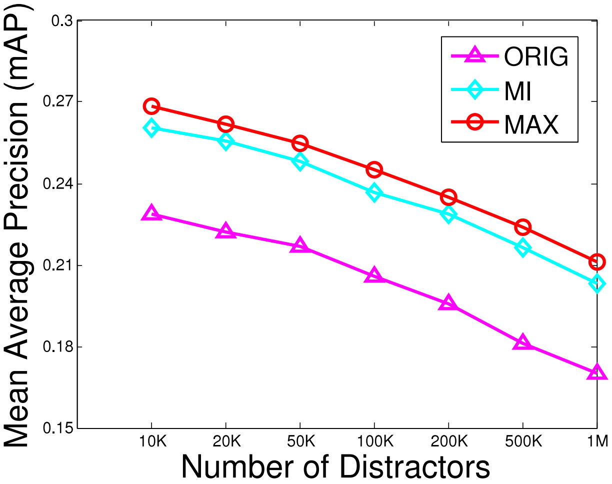

Fig. 3. The mAP values of image search systems using original, MI-SIFT [3] and Max-SIFT descriptors. Max-SIFT descriptors work slightly better than MI-SIFT, and much better than the original SIFT descriptors.

4. EXPERIMENTS

4.1. Dataset and Setting

We use the CarLogo-51 dataset [6] for logo search. This dataset contains 51 famous car logos and 11903 images. Samples images are shown in Figure 1 and 4. We mix the dataset with one million irrelevant images crawled from the Internet. To evaluate the performance with respect to the number of distractors, we mix the basic dataset with smaller sets containing 10K, 20K, 50K, 100K, 200K, and 500K distractors, respectively. We select 204 images (4 images per category) as query images for the large-scale Web logo search task.

We implement the Scalar Quantization algorithm [14] with original, MI-SIFT and Max-SIFT descriptors. We tem-

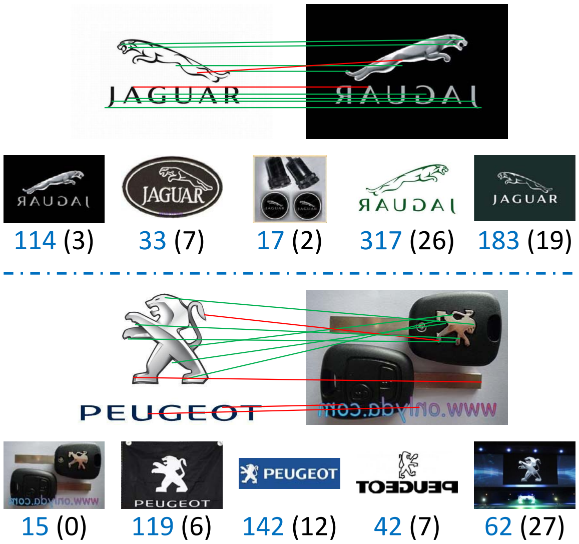

Fig. 4. Sample image matching and search results of two queries in the CarLogo-51 dataset [6]. We show 10 most significant feature matches on the sample images. The blue and black numbers below the candidate images indicate the number of matches by Max-SIFT and SIFT descriptors.

porarily have not implemented FIND and F-SIFT descriptors. The mean average precision (mAP) values for the queries are recorded, and averaged as the final score of the system.

4.2. Search Performance

The mAP values of all three systems are reported in Figure 3. It is observed that our descriptor achieves competitive performance in retrieving flipped Web logos. Moreover, the Max-SIFT descriptors outperform the original SIFT descriptors significantly, and even work better than the MI-SIFT descriptors [3] which requires more computational resources.

Some search samples are shown in Figure 4. One can see that on geometry-inverted and brightness-inverted images or those with significant contrast changes, the numbers of correct feature matches are greatly increased by adopting the Max-SIFT descriptors. Therefore, the proposed method is very efficient at retrieving the flipped logos which SIFT fails to find.

4.3. Time and Memory Complexity

Compared with the previous works, our solution is the easiest to implement. Our algorithm need only to compare several integer values in the fixed position of each SIFT descriptor and, if needed, perform the swapping operation to re-arrange the numeric values of the descriptor. The time cost by all the methods to calculate the flipping invariant descriptors are listed in Table 1. It is straightforward to see that, our algorithm is very efficient to carry out, requiring almost no extra computational overheads. The memory complexity of Max-SIFT is exactly the same as the original SIFT descriptors.

Table 1. Comparison of computational time costs (per image) of several flipping invariant descriptors (in milliseconds). We only compare the percentage of extra time costs.

Algorithm	Original	Modified	Extra Time (%)
MI-SIFT [3]	1840	1930	+4.89%
FIND [4] (L)	1000	383	−61.70%
FIND [4] (H)	1000	1749	+74.90%
F-SIFT [5]	651	779	+19.66%
Max-SIFT	247	250	**+1.21%**

4.4. Discussions

Experiments have revealed the effectiveness of the proposed Max-SIFT descriptors on Web logo matching and search. Moreover, Max-SIFT descriptors enjoy the lower time and memory complexity to the very related work, MI-SIFT [3]. Here, we would like to emphasize that Max-SIFT is especially designed for near-duplicate image search. When we are dealing with a large number of real-world images suffering deformable objects and/or non-rigid shapes, the Max-SIFT descriptors are not guaranteed to work very well.

5. CONCLUSIONS

In this paper, we propose Max-SIFT, a flipping invariant descriptor designed for Web logo search. Based on the clear intuition of how flipping operations will impact on the SIFT descriptors, we adopt an extremely simple idea to find the maximal group in the SIFT descriptors for detecting flipping operations. Experiments reveal that the Max-SIFT descriptors enjoy both high efficiency and excellent search accuracy. In the future, we will investigate the use of Max-SIFT descriptors in object recognition and detection tasks.

6. ACKNOWLEDGEMENT

This work was supported by the National Basic Research Program (973 Program) of China (Grant Nos. 2013CB329403 and 2012CB316301), the National Natural Science Foundation of China (Grant Nos. 61332007, 61273023, 91120011 and 61128007), the Beijing Natural Science Foundation (Grant No. 4132046), and the Tsinghua University Initiative Scientific Research Program (Grant No. 20121088071). This work was also supported in part to Dr. Tian by ARO grant W911NF-12-1-0057, Faculty Research Awards by NEC Laboratories of America, and 2012 UTSA START-R Research Award, respectively.

7. REFERENCES

[1] D.G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal on Computer Vision*, 2004.

[2] J. Sivic and A. Zisserman, “Video Google: A Text Retrieval Approach to Object Matching in Videos,” *International Conference on Computer Vision*, 2003.

[3] R. Ma, J. Chen, and Z. Su, “MI-SIFT: Mirror and Inversion Invariant Generalization for SIFT Descriptor,” *ACM International Conference on Image and Video Retrieval*, 2010.

[4] X. Guo and X. Cao, “FIND: A Neat Flip Invariant Descriptor,” *International Conference on Pattern Recognition*, 2010.

[5] W.L. Zhao and C.W. Ngo, “Flip-Invariant SIFT for Copy and Object Detection,” *IEEE Transactions on Image Processing*, 2013.

[6] L. Xie, Q. Tian, W. Zhou, and B. Zhang, “Fast and Accurate Large-Scale Web Image Search via Graph Propagation and Search Process Tradeoff,” *Computer Vision and Image Understanding*, 2014.

[7] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust Wide-Baseline Stereo from Maximally Stable Extremal Regions,” *Image and Vision Computing*, 2004.

[8] K. Mikolajczyk and C. Schmid, “Scale & Affine Invariant Interest Point Detectors,” *International Journal on Computer Vision*, 2004.

[9] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded Up Robust Features,” *European Conference on Computer Vision*, 2006.

[10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary Robust Independent Elementary Features,” *European Conference on Computer Vision*, 2010.

[11] D. Nister and H. Stewenius, “Scalable Recognition with a Vocabulary Tree,” *Computer Vision and Pattern Recognition*, 2006.

[12] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object Retrieval with Large Vocabularies and Fast Spatial Matching,” *Computer Vision and Pattern Recognition*, 2007.

[13] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases,” *Computer Vision and Pattern Recognition*, 2008.

[14] W. Zhou, Y. Lu, H. Li, and Q. Tian, “Scalar Quantization for Large Scale Image Search,” *ACM Multimedia*, 2012.

[15] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, “Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval,” *International Conference on Computer Vision*, 2007.

[16] Y.H. Kuo, K.T. Chen, C.H. Chiang, and W.H. Hsu, “Query Expansion for Hash-Based Image Object Retrieval,” *ACM Multimedia*, 2009.

[17] O. Chum, A. Mikulik, M. Perdoch, and J. Matas, “Total Recall II: Query Expansion Revisited,” *Computer Vision and Pattern Recognition*, 2011.

[18] O. Chum, M. Perdoch, and J. Matas, “Geometric Min-Hashing: Finding a (Thick) Needle in a Haystack,” *Computer Vision and Pattern Recognition*, 2009.

[19] H. Jegou, M. Douze, and C. Schmid, “Hamming Embedding and Weak Geometric Consistency for Large S-cale Image Search,” *European Conference on Computer Vision*, 2008.

[20] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian, “Spatial Coding for Large Scale Partial-Duplicate Web Image Search,” *ACM Multimedia*, 2010.

[21] S. Zhang, Q. Huang, G. Hua, S. Jiang, W. Gao, and Q. Tian, “Building Contextual Visual Vocabulary for Large-Scale Image Applications,” *ACM Multimedia*, 2010.

[22] L. Xie, Q. Tian, M. Wang, and B. Zhang, “Spatial Pooling of Heterogeneous Features for Image Classification,” *IEEE Transactions on Image Processing*, 2014.

[23] R. Fergus, P. Perona, and A. Zisserman, “A Visual Category Filter for Google Images,” in *European Conference on Computer Vision*, 2004.

[24] Y. Jing and S. Baluja, “VisualRank: Applying PageRank to Large-Scale Image Search,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[25] R. Arandjelovic and A. Zisserman, “Three Things Everyone Should Know to Improve Object Retrieval,” *Computer Vision and Pattern Recognition*, 2012.

[26] D. Qin and L. Van Gool, “Query Adaptive Similarity for Large Scale Object Retrieval,” *Computer Vision and Pattern Recognition*, 2013.

[27] L. Zheng, S. Wang, Z. Liu, and Q. Tian, “Lp-norm IDF for Large Scale Image Search,” *Computer Vision and Pattern Recognition*, 2013.