REGIONAL DEEP FEATURE AGGREGATION FOR IMAGE RETRIEVAL

Dong-ju Jeong, Sungkwon Choo, Wonkyo Seo, and Nam Ik Cho

Department of Electrical and Computer Engineering, INMC, Seoul National University

ABSTRACT

This paper presents a method to aggregate deep features for an object-based image retrieval system. Several recent works have demonstrated that it is quite important to selectively aggregate features with a weighting scheme and extract features from the limited regions likely to contain specific objects. Hence, the proposed method is to find possible candidate regions in an image, extract region descriptors from each region, and match images in a region-by-region manner. To adhere to using a pre-trained network without retraining or spatial verification, several candidate regions are found in an image and a more sophisticated pooling scheme is used for better performance. Specifically, salient points with active responses are detected in the image and clustered to form the candidate regions. In each region, we aggregate activations of a convolutional layer with the emphasis on more active spatial positions, and generate region descriptors effective for the object-based image retrieval. Our experiments show that the proposed method performs well on several public datasets, especially for the images showing the varied shapes or positions of an object.

Index Terms— Image retrieval, object retrieval, feature aggregation, visual recognition

1. INTRODUCTION

Image features from convolutional or fully-connected layers of deep convolutional neural networks (CNNs) are widely used in visual recognition including image retrieval [1]. Recently, several works have proposed to aggregate the deep features to represent a whole image with global features from a fully-connected layer or local ones from a convolutional/pooling layer, which we focus on in this work.

Since the success of CNNs in visual recognition [2], several works have proposed to exploit the activations within fully-connected layers of a CNN. For example, the "neural codes" algorithm [3] utilizes the fully-connected layers of a fine-tuned CNN, and the MOP-CNN [4] performs multi-scale patch-based pooling also using the fully-connected layers of a pre-trained CNN. Meanwhile, the activations within convolutional/pooling layers are used as dense local descriptors with an aggregation method. The SPoC [1] method weights the local descriptors close to an image center and aggregates them with sum-pooling, and the CroW [5] algorithm emphasizes the descriptors on the positions with strong responses. Regional max-pooled features are also used to represent an image with the strongest response in each channel of a layer [6, 7].

It can be inferred from the weighting methods [1, 5, 6] and the ones using spatial verification [8, 9, 10] that the selective aggregation (or use) of dense local descriptors is quite important. Furthermore, some algorithms using object localization [6] (a kind of spatial verification) or a region proposal network (RPN) [7, 11] have demonstrated that feature extraction from limited areas can improve performance of object-based image retrieval. However, the methods not using spatial verification have the advantage that they can produce the features of database images off-line. Even though applying an RPN or fine-tuning of a CNN can also improve performance [3, 7], it may make an image retrieval system specific for certain categories (e.g., landmarks).

Considering the issues stated above, we focus on devising a system that does not use spatial verification and utilizes only a pre-trained network for general purposes. The proposed method emphasizes salient regional features among the general-purpose features. Specifically, we find spatially important candidate regions, and compare images with the regional descriptors from those regions. In addition, we utilize 2nd-order pooling [12, 13] to aggregate dense local features and produce region descriptors. Our experiments show that the comparison between region descriptors is effective in object-based image retrieval and a simply adapted query expansion (QE) method can be applied to our system.

2. CANDIDATE REGION PROPOSAL

We consider the convolutional layers within a CNN pre-trained for diverse categories of the ImageNet database [14] such as VGG19 [15], where it can be seen as a general-purpose feature extractor, not specific for any certain categories. The activations of a convolutional layer, which has K channels and spatial dimensions $W \times H$, can be seen as dense local descriptors $\mathbf{F} = \{\mathbf{f_p}\} \in \mathbb{R}^{W \times H \times K}$, where $\mathbf{f_p}$ contains the responses at a position $\mathbf{p} = (x, y)$.

The goal of this step is to find salient points in an input image I of size $W_I \times H_I$ and group them into candidate regions where the local descriptors are aggregated at the next

step. To find more active spatial positions, at each position $\mathbf{p}$, we sum the responses over the channels and $l2-$normalize the resultant map as in [5]:

$$\bar{\mathbf{F}} = \{\bar{f}_{\mathbf{p}}\} \in \mathbb{R}^{W \times H},\ \bar{f}_{\mathbf{p}} = \sum_{k=1}^{K} f_{\mathbf{p},k},\ \alpha_{\mathbf{p}} = \frac{\bar{f}_{\mathbf{p}}}{\sqrt{\sum_{\mathbf{p}} \bar{f}_{\mathbf{p}}^2}} \quad (1)$$

where $f_{\mathbf{p},k}$ is the k-th element of $\mathbf{f}_{\mathbf{p}}$. Then we define $\boldsymbol{\alpha} \in \mathbb{R}^{W \times H}$, the elements of which are $\alpha_{\mathbf{p}}$. Using the Hessian-affine detector [16] salient points are detected in I, and sorted in descending order by the value of $\alpha_{\mathbf{p}}$ at their positions, where $\boldsymbol{\alpha}$ is temporarily resized to the size $W_I \times H_I$. Then we find the $N_{\max}$ largest values of $\alpha_{\mathbf{p}}$ and define their positions as a salient point set $\mathbf{P} = [\mathbf{p}_1, ..., \mathbf{p}_{N_{\max}}]$. If the number of salient points is not sufficient, or less than $N_{\min}$ in an image, we select the positions with the largest $N_{\min}$ values within $\boldsymbol{\alpha}$ and then $\mathbf{P}$ is composed of them. Even though a salient point set may be generated for every image in the latter way, the Hessian-affine detector is used to search salient points also at the less active regions in the convolutional layer.

For generating some candidate regions, the global fast k-means clustering [17] is performed to group the salient points in $\mathbf{P}$ into L clusters, $\boldsymbol{\mathcal{C}} = [\mathbf{C}_1, ..., \mathbf{C}_L]$. Then, the power set of $\boldsymbol{\mathcal{C}}$ is produced, and we exclude the empty set from it and add the set of all the positions in I to it. For each element of this set, its convex hull is computed, and we set the interiors of the convex hulls as candidate regions $\boldsymbol{\mathcal{R}} = [\mathbf{R}_1, ..., \mathbf{R}_M]$, where $M = 2^L$ is the number of the candidate regions. We denote $\mathbf{R}_m$ resized by the ratio $[W/W_I, H/H_I]$ as $\mathbf{R}'_m$, and if the size of $\mathbf{R}'_m$ is (close to) zero, $\mathbf{R}_m$ is discarded.

3. DEEP FEATURE AGGREGATION

In addition to the spatial weights $\boldsymbol{\alpha}$, the channel weights $\boldsymbol{\beta} = [\beta_1, ..., \beta_K]^T$ introduced in [5] are adopted to mitigate visual burstiness [18]. The sparser the k-th channel of the layer gets, the larger the value of β_k becomes, because it is defined as:

$$\begin{aligned} Q_k &= \frac{1}{WH} \sum_{\mathbf{p}} \mathbb{1}[f_{\mathbf{p},k} > 0], \\ \beta_k &= \log\left(\frac{K\epsilon + \sum_k Q_k}{\epsilon + Q_k}\right),\ \forall k \end{aligned} \quad (2)$$

where ϵ is a small positive constant. With $\boldsymbol{\beta}$ defined above, we can compute a sum-pooled (1st-order pooled) descriptor $\mathbf{f}^1_{\mathbf{R}_m}$ for each region $\mathbf{R}_m$ as:

$$\mathbf{f}^1_{\mathbf{R}_m} = \boldsymbol{\beta} \circ \sum_{\mathbf{p} \in \mathbf{R}'_m} \mathbf{f}_{\mathbf{p}} \quad (3)$$

where $\circ$ is the Hadamard product, and each $\mathbf{f}^1_{\mathbf{R}_m}$ is $l2$-normalized.

Meanwhile, the dimension of a local descriptor should be rather small to perform the 2nd-order pooling in a region. To this end, we can produce PCA bases $\mathbf{U}^1$ from the set $\{\mathbf{f}^1_{\mathbf{R}_m}\}$ of a database for dimensionality reduction, so the local descriptors with the reduced dimension of K' are given by:

$$\tilde{\mathbf{f}}_{\mathbf{p}} = \mathbf{U}^{1T}(\boldsymbol{\beta} \circ \mathbf{f}_{\mathbf{p}}) \in \mathbb{R}^{K'}. \quad (4)$$

Given the local descriptors that have the reduced dimension, each region $\mathbf{R}_m$ has its 2nd-order average pooled descriptor [12] with the spatial weights $\boldsymbol{\alpha}$:

$$\begin{aligned} \mathbf{G}^{\text{avg}}_{\mathbf{R}_m} &= \frac{1}{\sum_{\mathbf{p} \in \mathbf{R}'_m} \alpha_{\mathbf{p}}} \sum_{\mathbf{p} \in \mathbf{R}'_m} \alpha_{\mathbf{p}} (\tilde{\mathbf{f}}_{\mathbf{p}} \tilde{\mathbf{f}}_{\mathbf{p}}^T) \\ \mathbf{f}^2_{\mathbf{R}_m} &= \text{vec}(\log(\mathbf{G}^{\text{avg}}_{\mathbf{R}_m})) \in \mathbb{R}^{\frac{K'(K'+1)}{2}} \end{aligned} \quad (5)$$

where $\text{vec}(\cdot)$ vectorizes the upper (or lower) triangular matrix of an input matrix. Then, each $\mathbf{f}^2_{\mathbf{R}_m}$ is $l2$-normalized. At last, the set $\{\mathbf{f}^2_{\mathbf{R}_m}\}$ in a database can also produce a PCA whitening matrix for the 2nd-order pooled descriptors to have their dimension of D:

$$\tilde{\mathbf{f}}^2_{\mathbf{R}_m} = \text{diag}(s_1, ..., s_D)^{-1} \mathbf{U}^{2T} \mathbf{f}^2_{\mathbf{R}_m} \quad (6)$$

where $\mathbf{U}^2$ is the PCA matrix and $\text{diag}(s_1, ..., s_D)$ is the diagonal matrix whose diagonal entries s_d are the associated singular values. $\tilde{\mathbf{f}}^2_{\mathbf{R}_m}$ are $l2$-normalized and used to compute the similarity between an image pair.

4. IMAGE SEARCH

4.1. Similarity measure for an image pair

Through the above procedures, an image gets M D-dimensional region descriptors. When a query image I_q and one of target database images I_t have M_q and M_t region descriptors $\tilde{\mathbf{f}}^{2q}_{\mathbf{R}_m}$ and $\tilde{\mathbf{f}}^{2t}_{\mathbf{R}_m}$ respectively, the similarity between these two images is computed as:

$$\text{ImSim}(I_q, I_t) = \max_{\substack{m \in \{1,...,M_q\} \\ m' \in \{1,...,M_t\}}} \text{VecSim}(\tilde{\mathbf{f}}^{2q}_{\mathbf{R}_m}, \tilde{\mathbf{f}}^{2t}_{\mathbf{R}_{m'}}) \quad (7)$$

where $\text{ImSim}(\cdot,\cdot)$ is the similarity between two images and $\text{VecSim}(\cdot,\cdot)$ is the similarity between two descriptors such as the cosine similarity equivalent to the Euclidean distance due to the $l2$-normalization. By selecting the region best-matched to the query image, we can search relevant images irrespective of the size and positions of the regions.

4.2. Query expansion

To perform re-ranking, the query expansion (QE) technique [19] can be adapted for our method. Because the best-matched regions in images and their descriptors can be found via an initial search, those region descriptors of the top N_{QE} results are averaged and $l2$-normalized to produce a new single descriptor $\tilde{\mathbf{f}}^2_{\text{QE}}$ for a query. Following this procedure, at the second search, (7) can be replaced with:

$$\text{ImSim}(I_q, I_t) = \max_{m' \in \{1,...,M_t\}} \text{VecSim}(\tilde{\mathbf{f}}^{2q}_{\text{QE}}, \tilde{\mathbf{f}}^{2t}_{\mathbf{R}_{m'}}) \quad (8)$$

Table 1. Mean average precision (mAP) of ours and other comparable methods on the experimental datasets. *OF* means the experiments with uncropped (full) input images of *Oxford*.

Method	Dim.	*Oxford*	*OF*	*Paris*	*Holidays*
Neural Codes [3]	128	—	0.433	—	0.747
	256	—	0.435	—	0.749
	512	—	0.435	—	0.749
MOP-CNN [4]	2,048	—	—	—	0.802
SPoC [1]	256	0.531	0.589	—	0.802
CroW [5]	128	0.592	—	0.746	—
	256	0.654	—	0.779	0.831
	512	0.682	—	0.796	0.849
R-MAC [6]	256	0.561	—	0.729	—
	512	0.669	—	**0.830**	—
Proposed	128M	0.682	0.690	0.786	0.875
	256M	0.725	0.717	0.798	0.882
	512M	**0.741**	**0.730**	0.802	**0.886**
CroW + QE [5]	128	—	—	0.827	—
	256	0.692	—	0.850	—
	512	0.722	—	0.855	—
Proposed + QE	128M	0.750	0.745	0.851	—
	256M	0.791	0.768	0.862	—
	512M	**0.816**	**0.781**	**0.864**	—

5. EXPERIMENTAL RESULTS

In our experiments, *Oxford* (Oxford Buildings) [20], *Paris* [21], and *Holidays* (INRIA Holidays) [22] datasets are used to evaluate the performance of our method and compare it with other ones. *Oxford* consists of 5,062 Flickr images of Oxford landmarks including 55 queries. *Paris* is similar to *Oxford* except that it contains 6,412 Paris landmark images. The standard evaluation protocol of *Oxford* and *Paris* accompanies each query with a bounding box with which the query is cropped. *Holidays* is composed of 1,491 vacation photographs including 500 queries. For each dataset, the performance is reported as mean average precision (mAP) over its queries. Following the experiments in [1, 3] etc., we manually fixed some of *Holidays* images that are wrongly rotated by ± 90 degrees. In extra experiments, the performance with the original *Holidays* dataset was worse by approximately 0.04 mAP.

It is noticeable that the size of an input image has much influence on performance from the fact that SPoC without center-prior [1] and uCroW [5] are almost the same but they are quite different in their reported mAP. Hence, we keep the original sizes of *Oxford* and *Paris* images, and the *Holidays* images are resized so that their longer dimension becomes 1,024. *Caffe* [23] package is used for CNNs and the dense local descriptors are from the last convolutional layer of the VGG19 model [15], where $K = 512$. Several parameters are involved in the proposed method: $N_{\max}$, $N_{\min}$, L for the candidate region proposal step, and K', D for the feature aggregation step. We consistently set $N_{\max} = 4,000, N_{\min} = 300, L = 3, K' = 128$ and D varies according to experiments ($D \in \{128, 256, 512\}$).

Table 1 shows the comparison of our method with those that extract features from pre-trained networks and do not use spatial verification. Because, as for *Oxford* and *Paris*, the standard evaluation protocol provides the bounding boxes of query objects, we use only the whole cropped image $\mathbf{R}_M$ to compute the descriptor for each query. The number of samples for the query expansion, N_{QE}, is set to 10 as in [5], and the QE of CroW sums the descriptors representing their images, while our QE procedure uses a single regional vector for each image, following (8). The QE process is not applied to *Holidays* since the number of relevant images for each query is too small to sum the descriptors of top-ranked images. Even though, in most cases, our image descriptors are less compact than those of the other methods due to the use of regional descriptors, it is shown in Table 1 that our algorithm outperforms the others for *Oxford* and *Holidays*. As for *Paris*, the queries and their relevant images have a tendency to be very similar to each other on the whole, so it seems that the use of regional descriptors may bring the side effect to this dataset.

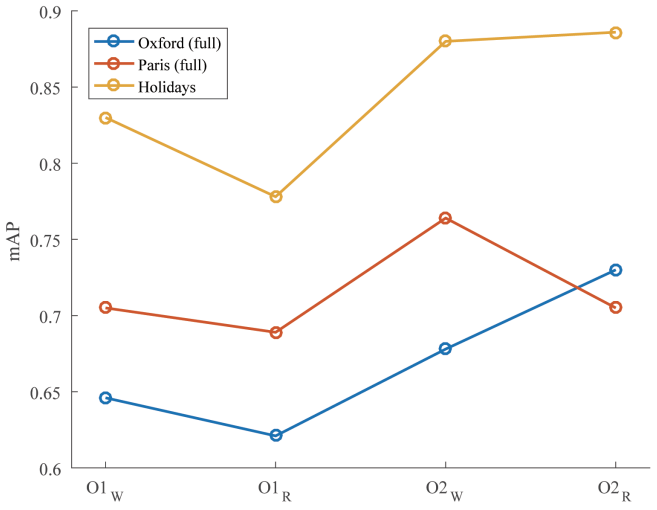

Fig. 1. Comparison of variations of our method. $O1$ and $O2$ indicate the 1st-order and the 2nd-order pooling respectively, while the subscripts "R" and "W" mean the use of descriptors representing all of $[\mathbf{R}_1, ..., \mathbf{R}_M]$ and only $\mathbf{R}_M$ respectively. The word "full" means that query images are not cropped.

Fig. 1 shows the comparison of several variations of our algorithm: the 1st-order pooling and the use of only a single

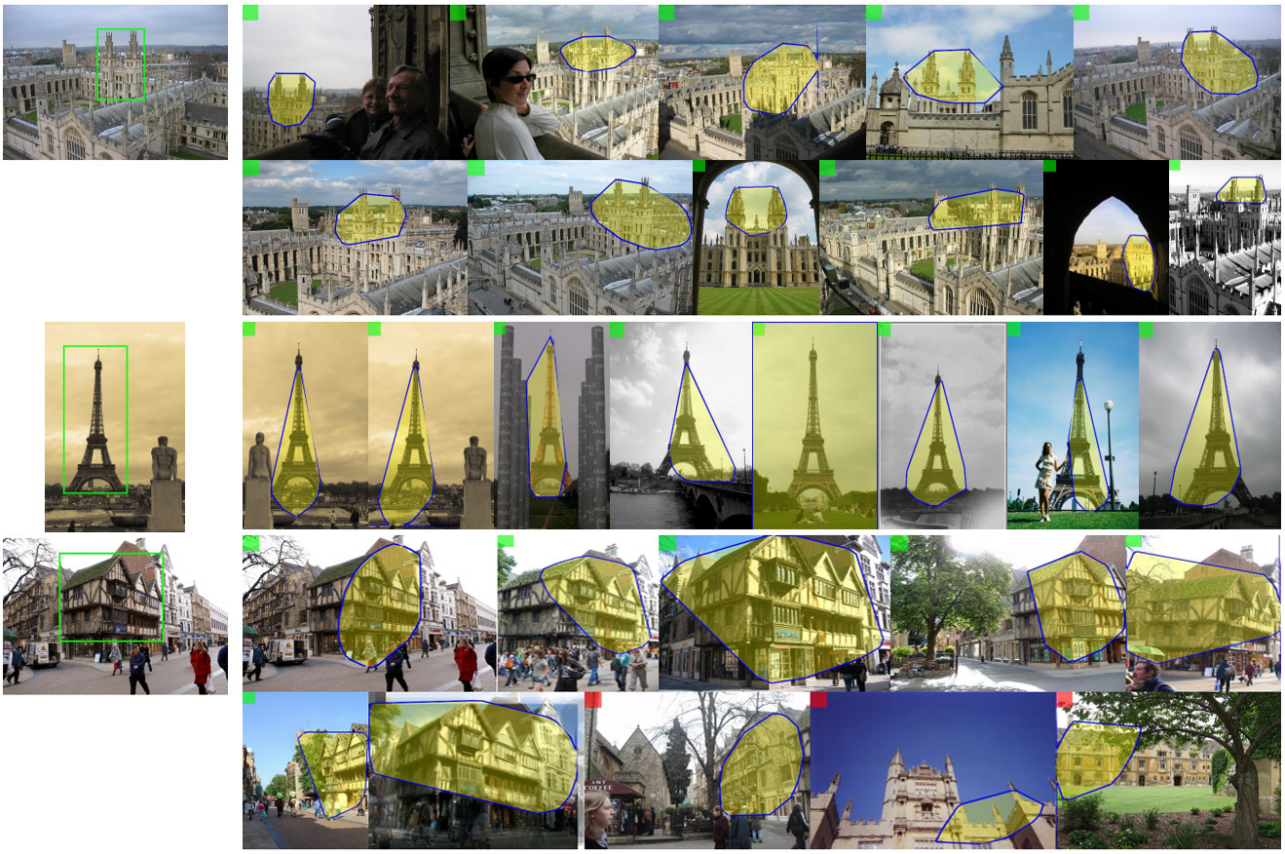

Fig. 2. Examples of our retrieval results for several queries with (green) bounding boxes. The leftmost images are the queries and the other ones are the top-ranked results for each query. At the top-left corners of images, green, red, and gray marks indicate relevant, irrelevant, and ignored ("junk"-labeled) images, respectively. In each image, a yellow-shaded region indicates the one best-matched to its query out of candidate regions, and it is smaller than its receptive field.

descriptor representing a whole image. The 1st-order pooled descriptors are given by:

$$\mathbf{f}^1_{\mathbf{R}_m} = \boldsymbol{\beta} \circ \sum_{\mathbf{p} \in \mathbf{R}'_m} \alpha_{\mathbf{p}} \mathbf{f}_{\mathbf{p}} \tag{9}$$

that is a slightly-tweaked version of (3), and only using a single descriptor of a whole image means that only the biggest region $\mathbf{R}_M$ is used for aggregation. Considering the values of $O1_W$, $O2_W$, and $O2_R$, it is notable that both the 2nd-order pooling and the use of regional descriptors are complementary to our method. However, the regional aggregation is not effective for the 1st-order pooled descriptors in all the cases, and confirmed to be also not helpful for *Paris* for the reason mentioned above.

Lastly, our retrieval examples are shown in Fig. 2. We confirm that the regions best-matched to each query have the shapes similar to each other, which helps to enhance the retrieval results. However, it is shown in the last two images of Fig. 2 that some small irrelevant regions may be matched to a query due to partly similar shapes or limited information.

6. CONCLUSIONS

We have proposed an object-based image retrieval method, which finds candidate regions to extract region descriptors and matches images in a region-by-region manner. For candidate region proposal, salient points with active responses of a CNN are grouped into several clusters, the power set of which forms the candidate regions. Second-order pooled descriptors are extracted from the regions, and we compute the similarity between two images with such region descriptors. Experimental results show that the proposed method performs well on the image retrieval datasets with diverse classes, but indicate some limitations in that the use of regional descriptors might bring a side effect in the case of images very similar to each other on the whole or including so partly similar regions.

Acknowledgements

This research was supported in part by Projects for Research and Development of Police science and Technology under Center for Research and Development of Police science and Technology and Korean National Police Agency (PA-C000001), and in part by the Brain Korea 21 Plus Project in 2016.

7. REFERENCES

[1] A. B. Yandex and V. Lempitsky, "Aggregating deep convolutional features for image retrieval," in *IEEE International Conference on Computer Vision*, 2015, pp. 1269–1277.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012.

[3] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *European Conference on Computer Vision*, 2014, pp. 584–599.

[4] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *European Conference on Computer Vision*, 2014, pp. 392–407.

[5] Y. Kalantidis, C. Mellina, and S. Osindero, "Cross-dimensional weighting for aggregated deep convolutional features," *ArXiv preprint arxiv:1512.04065*, Dec 2015.

[6] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," in *International Conference on Learning Representations*, 2014.

[7] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," *ArXiv preprint arxiv:1604.01325*, Jul 2016.

[8] X. Li, M. Larson, and A. Hanjalic, "Pairwise geometric matching for large-scale object retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5153–5161.

[9] G. Tolias, Y. Avrithis, and H. Jégou, "Image search with selective match kernels: Aggregation across single and multiple images," *International Journal of Computer Vision*, vol. 116, no. 3, pp. 247–261, Feb 2016.

[10] G. Tolias and H. Jégou, "Visual query expansion with or without geometry: Refining local descriptors by feature aggregation," *Pattern Recognition*, vol. 47, no. 10, pp. 3466–3476, Oct 2014.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN," in *Advances in Neural Information Processing Systems*, 2015.

[12] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, "Semantic segmentation with second-order pooling," in *European Conference on Computer Vision*, 2012, pp. 430–443.

[13] C. Ionescu, J. Carreira, and C. Sminchisescu, "Iterated second-order label sensitive pooling for 3d human pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1661–1668.

[14] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[16] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, Oct 2004.

[17] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, Feb 2003.

[18] H. Jégou, M. Douze, and C. Schmid, "On the burstiness of visual elements," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1169–1176.

[19] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[20] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[21] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[22] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European Conference on Computer Vision*, 2008, pp. 304–317.

[23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.