TWO-STAGE POOLING OF DEEP CONVOLUTIONAL FEATURES FOR IMAGE RETRIEVAL

Tiancheng Zhi, Ling-Yu Duan, Yitong Wang, Tiejun Huang

Institute of Digital Media, School of EE&CS, Peking University, Beijing, 100871, China
e-mail: {tiancheng.zhi, lingyu, wangyitong, tjhuang}@pku.edu.cn

ABSTRACT

Convolutional Neural Network (CNN) based image representations have achieved high performance in image retrieval tasks. However, traditional CNN based global representations either provide high-dimensional features, which incurs large memory consumption and computing cost, or inadequately capture discriminative information in images, which degenerates the functionality of CNN features. To address those issues, we propose a two-stage partial mean pooling (PMP) approach to construct compact and discriminative global feature representations. The proposed PMP is meant to tackle the limits of traditional max pooling and mean (or average) pooling. By injecting the PMP pooling strategy into the CNN based patch-level mid-level feature extraction and representation, we have significantly improved the state-of-the-art retrieval performance over several common benchmark datasets.

Index Terms— image representation, image retrieval, convolutional neural network, feature pooling, compact descriptor

1. INTRODUCTION

Image retrieval has attracted extensive attentions in both academia and industry over the last decade. Feature representation is a core factor influencing both accuracy and efficiency in image retrieval tasks. Traditionally, hand-crafted local features such as SIFT [1] and SURF [2] are aggregated to a global representation by methods such as Bag-of-Words (BoW) [3], Vector of Locally Aggregated Descriptors (VLAD) [4] and Fisher Vector (FV) [5]. In recent years, with the rapid development of deep learning, the features extracted from pre-trained Convolutional Neural Network (CNN) models [6, 7, 8] have achieved higher performance and flexibility than traditional hand-crafted aggregated descriptors in typical image retrieval tasks (e.g., scene retrieval [9], landmark recognition [10], etc).

Generally speaking, the CNN based image representations can be divided into two types. The first type presents whole images to a pre-trained CNN model and get global representations. A simple approach [11] is to extract high level features from fully connected layers such as fc6/fc7 in AlexNet [6] or CaffeNet [7]. However the raw high dimensional CNN features are much less efficient due to time consuming similarity distance computing. Recent work [11] has applied Principal Component Analysis (PCA) [12] to reduce feature dimension further. Although PCA can transform the features to a low-dimensional representation, the transformation matrix is very large , thereby incurring a time-consuming reduction process. Additionally, if convolutional or pooling features are treated as common vectors, the property that each convolutional or pooling feature map is composed by position related responses is ignored, while feature pooling can handle such problem. Therefore, feature pooling based middle layer representations are considered. Babenko et al. [13] proposed sum pooling to reduce the dimension of the last convolutional or pooling layer features and achieved performance improvement. However, image retrieval tasks usually incur a complex scene, involving multiple scales, cluttered background, as well as multiple subjects, which renders it unsuitable or less optimal for the global representations of a whole image to capture necessary semantic information in retrieval tasks.

To deal with the weakness of whole image based global representations, the second type of methods extracts CNN features of image patches from original images and aggregates them into global representations. For example, Gong et al. [14] have proposed a MOP-CNN scheme, which aggregates deep features of sliding windows of different scales using VLAD. However, the VLAD method is limited by dimension curse since the length of global features are C times of patch-level features, where C is the vocabulary size. MOP-CNN uses PCA to reduce feature dimension which still faces the disadvantages of PCA dimension reduction as mentioned above. Some other works [15, 16] prefer simple approaches such as max pooling or mean pooling (also known as "average pooling") to get compact global image representations. Unfortunately, max pooling is easily affected by extreme responses while mean pooling may incur background distractors, thereby degenerating meaningful responses.

To address the drawbacks of previous approaches, we propose an effective and efficient two-stage partial mean pooling (PMP) strategy and embed PMP into an advanced feature extraction framework. The proposed PMP attempts to alleviate the disadvantages of both max pooling and mean pooling. In feature extraction, PMP is applied in two stages: 1. intra-patch pooling to capture the discriminative responses from convolutional feature maps; 2. inter-patch pooling to aggregate patch-level features into compact global representation. Extensive evaluation shows that the proposed PMP is superior to max pooling and mean pooling. Meanwhile, the proposed feature extraction framework significantly improves the state-of-the-art retrieval accuracy on several benchmark datasets, with a fairly short feature dimension.

The rest of the paper is organized as follows. In Section 2, we introduce the intra-patch and inter-patch pooling strategy, as well as the feature extraction framework. Extensive experiment results and comparison analysis are presented in Section 3. Finally, we conclude this paper in Section 4.

2. PROPOSED APPROACH

In this section, the proposed PMP based feature extraction framework (See Fig.1) is described in detail. Our framework includes three main stages: patch detection, mid-level feature extraction and

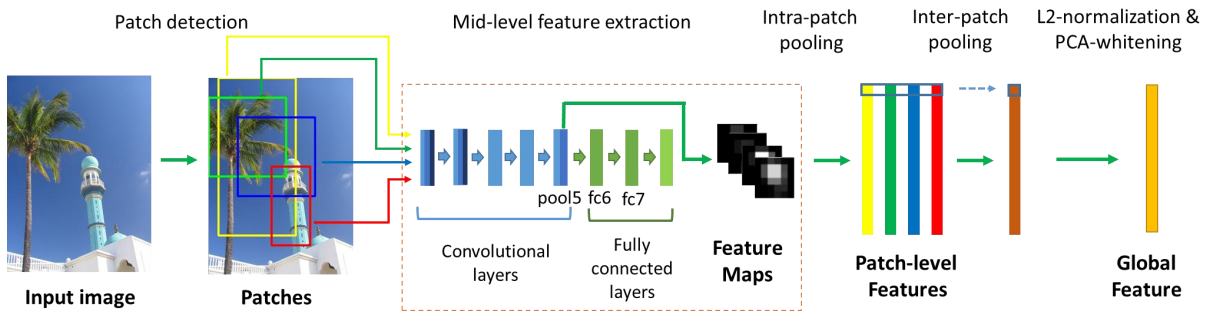

Fig. 1. Overview of the proposed feature extraction framework. The pool5, fc6 and fc7 layers are indicated for CaffeNet. For VGGNet, the network is deeper but remains similar structure.

two-stage partial mean pooling.

2.1. Patch Detection

As aforementioned, presenting whole images to a pre-trained network can hardly deal with the problems of multiple scales, noisy background and abundant subjects. Inspired by recent successful object detection approach using R-CNN [17], we apply object proposal algorithms (e.g., BING [18], Selective Search [19], Edge Boxes [20] and etc.) to detect regions with high objectness followed by region based feature extraction. In this way, the interference effects of background and other distracting objects can be reduced at the patch level. As detected patches are those regions with high objectness and CNN features are good at describing the semantic information of objects [21], the CNN mid-level features will be extracted for each patch and then aggregated to form a global representation.

As analyzed in [22], among most recent object proposal algorithms, BING is with the lowest computation complexity (the whole detection process only takes $\sim$10ms in a single thread over a normal PC with 2.6GHz CPU). Towards high efficient feature extraction, we adopt BING as the patch detection algorithm.

It is worthy to mention that MOP-CNN [14] has proposed to handle scale variance by sliding windows with different scales. However, since the patches are greedily detected at pre-determined scales and sliding steps, there is no guarantee that those patches can capture meaningful objects in appropriate sizes. For example, the parts of two distinct objects may be covered in a single image patch, which yields inferior CNN features. Thus our work does not prefer MOP-CNN.

2.2. Mid-level Feature Extraction

As illustrated in Fig.1, given a pre-trained CNN model, we extract the last convolutional/pooling feature maps for each image patch as mid-level feature. Specifically, the pool5 layer features from CaffeNet [7] and VGGNet (16 layers) [8] are considered in our work. We propose to sort the responses of pool5 layer from CaffeNet or VGGNet in a descending order within each feature map. As detailed below, our experiment study has shown that the sorted features provide better description than raw features.

Intuitively, the sorted mid-level features avoid the explicit hard coding of location information and thus handle position variance better than raw features. Sorting can gather higher responses for meaningful objects at different locations, so that the similarity distance computing is much less affected by object location variance. For example, each of 256 feature maps in the pool5 layer in CaffeNet [7] is in the form of a 6×6 matrix, of which each matrix element actually codes information of a distinct position. To capture the structure information of pool5 layer, we could successively concatenate the responses of 36 elements, while the position invariance would not be satisfied. Thus, our mid-level feature extraction sorts the responses to make the feature less sensitive to position variance.

Layer	Dimension	mAP
fc6	4096	75.3
fc7	4096	75.7
pool5	9216	71.6
pool5 (sorted)	9216	**80.4**

Table 1. Retrieval performance of features extracted from different layers of CaffeNet over the Holidays dataset. The sorted pool5 layer features have yielded much better retrieval performance.

We have empirically validated that the mid-level pooling features excluding the explicit injection of position information outperform the high-level fully connected features. Table. 1 compares the retrieval performances of the sorted pool5 features and other features on Holidays [9] over CaffeNet. From Table. 1, the sorted pool5 features work better than the original pool5 features, and even outperform the features of higher layers like fc6 and fc7.

In addition, compared to the memory and time consuming fully connected feature, the use of convolutional or pooling layer can significantly reduce the memory and computation cost because fewer large matrix multiplications are involved in feature extraction.

However, the dimension of sorted mid-level features are still too high for retrieval. Hence, we introduce the two-stage partial mean pooling to construct discriminative and compact features in the following.

2.3. Two-stage Partial Mean Pooling

The two stage partial mean pooling (PMP) is then applied to the patch-level features for generating compact representations.

Intra-patch Pooling. Intra-patch pooling is to capture the discriminative responses on the feature maps and transform the feature to a low-dimensional representation. As a sort of quantization, the proposed PMP based intra-patch pooling can remove the negative effects of position variance in feature representation. We formulate the PMP for intra-patch pooling as follows.

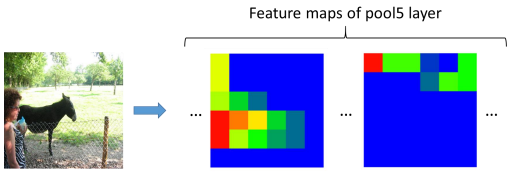

Fig. 2. Response visualization by the feature maps of pool5 layer using CaffeNet. Each feature map is supposed to represent a concept. A moderate number of response values may illustrate a concept meaningful feature map.

Let $\bar{X} = \{X_i | i = 1, 2, ..., M\}$ denote the output M feature maps of a convolutional or pooling layer in a CNN model, and $X_i = \{x_{i,j} | j = 1, 2, ..., h \times w\}$ denote the i_{th} feature map, where w and h denote the height and weight of a feature map. Specifically, $M = 256$ and $w = h = 6$ were set in the pool5 layer of CaffeNet, and $M = 512$ and $w = h = 7$ in the pool5 layer of VGGNet. We first sort $x_{i,1} \sim x_{i,h\times w}$ in descending order as described in Sec. 2.2 and let $x^{'}_{i,1} \sim x^{'}_{i,h\times w}$ denote the sorted responses. Unlike max and mean pooling, PMP calculates the mean value of top K_1 responses to get the pooling feature $Y = \{y_i | i = 1, 2, ..., M\}$ from $\bar{X}$, where y_i is denoted as:

$$y_i = \frac{1}{K_1} \sum_{j=1}^{K_1} x^{'}_{i,j}, \tag{1}$$

where K_1 denotes the number of top ranked responses in each feature map.

Note that PMP may degenerate to the form of traditional max pooling or mean pooling when $K_1 = 1$ or $h \times w$. PMP is meant to seek for a trade-off between max pooling and mean pooling by considering the largest K_1 responses, in which, by appropriately setting K_1 value, better pooling results of feature maps can be expected. As shown in Fig.2, when K_1 is set to an appropriate value, say, $\sim$20% of $h \times w$, the pooling result of PMP shows much better delineation of a meaningful pattern ("donkey", "leaf", etc.). By contrary, max pooling tends to capture noisy strong response values when the majority of responses are mild, while mean pooling would yield a skewed value when the meaningful responses are just from the minority.

Inter-patch Pooling. In inter-patch pooling stage, the features of different patches are aggregated to form a global representation. Rather than max or mean pooling, PMP is applied to aggregate the features of different patches as well. Fig.3 illustrates different pooling effects of max, mean and PMP methods. Note that we do not apply the typical aggregation approaches such as VLAD and FV as they incur high-dimensional representations. Actually, recent aggregation work has preferred simple but effective pooling [15, 16]. Let N denote the number of patches and $\bar{Y}$ denote the features of all patches after intra-patch pooling:

$$\bar{Y} = \begin{pmatrix} y_{1,1} & \cdots & y_{1,M} \\ \vdots & \ddots & \vdots \\ y_{N,1} & \cdots & y_{N,M} \end{pmatrix} \tag{2}$$

where a row denotes the feature of each patch Y and $y_{i,j}$ refers to the y_j in the i-th pooling feature Y. For each column, $y_{1,j} \sim y_{N,j}$ are sorted to generate the list of $y^{'}_{1,j} \sim y^{'}_{N,j}$ in descending order and PMP is applied to calculate the inter-patch pooling result as $Z = \{z_i | i = 1, 2, ..., M\}$, and z_i is denoted as:

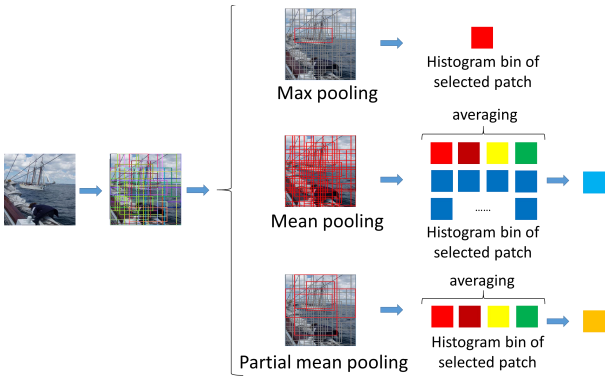

Fig. 3. A toy example of inter-patch pooling to compare max, mean and partial mean pooling (note the top 50 patches are shown). Each block indicates a response value. The block colors (from blue to red) indicate different response strength (from weak to strong). The proposed PMP provides more discriminative pooling results for "ship" object. Compared with mean pooling, PMP discards noisy background patches. Compared with max pooling, PMP can better handle scale and position variance by averaging the features of meaningful patches.

$$z_i = \frac{1}{K_2} \sum_{j=1}^{K_2} y^{'}_{j,i}, \tag{3}$$

where K_2 denotes the number of interest patches in each column.

To further improve the discriminative power, as shown in Fig.1, additional L2-normalization and PCA-whitening is subsequently applied to Z. As Z is with very low dimension, the overhead of PCA transformation can be ignored.

3. EXPERIMENTS

Datasets. We evaluate the proposed approach on three benchmark datasets: INRIA Holidays (Holidays) [9], University of Kentucky Benchmark (UKBench) [24] and Oxford building dataset (Oxford5K) [10]. Holidays dataset contains 1491 images, which can be divided into 500 query images and 991 reference images. The retrieval performance is measured by mean average precision (mAP). UKBench dataset consists of 2550 groups, each of which contains 4 pictures of the same object captured from different viewpoints. Each image is treated as query once and mean precision at 4 is used for evaluation. Oxford5K dataset contains 5063 building images. In total 55 query images of 11 different buildings are applied and the performance is evaluated by mAP. An external subset of 10,000 images from the ImagetNet dataset [25] is used to train PCA parameters. To measure the feature similarity, cosine distance is applied in the subsequent experiments.

Impact of Parameters. Referring to Equ.1, 2 and 3, we study the impact of three parameters: parameter K_1 in intra-patch pooling, parameter K_2 in inter-patch pooling and the number N of selected patches. Towards comprehensive performance evaluation, we extend the parameters study to two typical networks: CaffeNet and VGGNet.

Instead of exhaustively enumerating the complete parameter space, we study the impact of each parameter separately. We first analyze the influence of K_1 over Holidays dataset. For simplicity,

Method	Dimension	Holidays	UKBench	Oxford5K
SIFT + TE + DA [23]	1024	72.0	3.51	56.0
Neural Codes [11]	4096	79.3	-	-
MOP-CNN + PCA [14]	512	78.3	-	-
CNN + SPoC [13]	256	80.2	3.65	58.9
CNN + DPS + Max Pooling [15]	4096	81.0	3.67	56.0
Intra-patch Pooling (CaffeNet) + VLAD ($C = 64$)	16384	82.3	3.43	43.1
Intra-patch Pooling (CaffeNet) + FV ($C = 64$)	16384	84.3	3.44	44.5
Two-stage Pooling (CaffeNet, without PCA-whitening)	256	84.5	3.70	48.2
Two-stage Pooling (CaffeNet, with PCA-whitening)	256	85.1	3.76	56.8
Two-stage Pooling (VGGNet, without PCA-whitening)	512	86.0	3.80	57.3
Two-stage Pooling (VGGNet, with PCA-whitening)	512	**86.6**	**3.80**	**64.0**

Table 2. Comparisons with state-of-the-art global image representations. C denotes the vocabulary size.

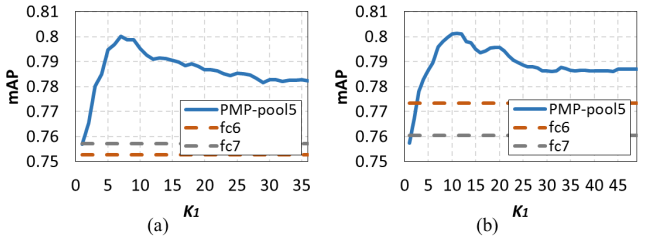

Fig. 4. Performance influence of PMP parameter K_1 on the Holidays dataset. (a) the results via CaffeNet, (b) the results via VGGNet.

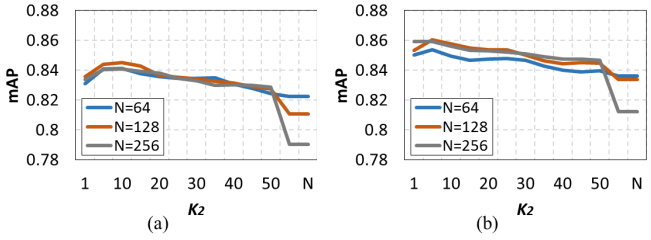

Fig. 5. Performance influence of PMP parameters N and K_2 on the Holidays dataset. (a) the results via CaffeNet, (b) the results via VGGNet.

we consider the case of the whole image as a single patch to explore the $K1$ impact of intra-patch pooling, in which $N = K_2 = 1$. Note that PMP degrades to max pooling when $K_1 = 1$ and degrades to mean pooling when $K_1 = w \times h$. As shown in Fig.4, the best retrieval performance is achieved on Holidays when $K_1 = 7$ on CaffeNet and $K_1 = 11$ on VGGNet, which is much better than the results of max pooling and mean pooling.

To further study the influence of N and K_2, we fix $K_1 = 7$ on CaffeNet and $K_1 = 11$ on VGGNet. We empirically set $N = 64, 128, 256$ and $K_2 = 1, 5 \sim 50$ (step size = 5), N. As shown in Fig.5, PMP demonstrates its superiority. Thus we empirically set $N = 128$, $K_1 = 7$, $K_2 = 10$ for CaffeNet and $N = 128$, $K_1 = 11$, $K_2 = 5$ for VGGNet in the following performance comparison experiments over more benchmark datasets.

Performance Comparison. We setup several baselines for extensive performance comparison, including: (1) SIFT+TE+DA [23]: the state-of-the-art SIFT based compact aggregated descriptors; (2) Neural Codes [11]: extracting raw fully-connected features from a pre-trained CNN model over the whole image; (3) MOP-CNN + PCA [14]: aggregating fully-connected features of CNN from multi-scale patches with VLAD, and performing PCA to reduce feature dimension; (4) CNN + SPoC [13]: sum pooling and whitening of the image level global convolutional features; (5) CNN + DPS + Max Pooling [15]: selecting patches with a statistical model and aggregating fully-connected features of CNN from selected patches with max pooling; (6) Intra-patch Pooling + VLAD: performing the proposed PMP for intra-patch pooling and aggregating pooling features with VLAD approach [4]; (7) Intra-patch Pooling + FV: performing proposed PMP for intra-patch pooling and aggregating pooling features with FV [5]; (8) Two-stage Pooling: injecting the proposed PMP into the two-stage pooling feature extraction pipeline. Except the Baseline (5) is performed via VGGNet, other CNN baselines are carried out via CaffeNet or similar structure.

Table 2 lists the comparison results on Holidays, UKBench and Oxford5K. Our approach significantly outperforms state-of-the-art SIFT based compact aggregated descriptors (SIFT+TE+DA [23]). Like other state-of-the-art CNN schemes, PCA-whitening has improved the performance of our proposed approach. The superior performance of our proposed two stage PMP strategy has demonstrated its strength in eliminating the negative effects of scale, translation, cluttered background, etc. Baselines (6) and (7) replace our inter-patch pooling approach with the typical aggregation methods VLAD and FV. However, our results are superior to (6) and (7) in both accuracy and compactness, which indicates the advantage of the inter-patch PMP in aggregation. Note that the VGGNet [8] features benefit from a deeper network structure and its derived strong generalization and thereby outperform the CaffeNet [7] features. Overall, compared with other baselines, the proposed approach yields better retrieval performance at a very short feature dimension.

4. CONCLUSION

We have proposed a two-stage partial mean pooling strategy towards an advanced CNN feature extraction framework. The proposed compact and discriminative image representation outperforms state-of-the-art methods. How to incorporate low-level invariant features into this feature extraction framework (i.e., the effective combination of CNN and SIFT features) will be included in our future work.

5. ACKNOWLEDGEMENT

This work was supported by the National Hightech R&D Program of China (863 Program): 2015AA016302, and Chinese Natural Science Foundation: 61271311, 61390515, 61421062.

6. REFERENCES

[1] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in *ECCV*, pp. 404–417. 2006.

[3] Josef Sivic and Andrew Zisserman, "Video google: A text retrieval approach to object matching in videos," in *CVPR*, 2003, pp. 1470–1477.

[4] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez, "Aggregating local descriptors into a compact image representation," in *CVPR*, 2010, pp. 3304–3311.

[5] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier, "Large-scale image retrieval with compressed fisher vectors," in *CVPR*, 2010, pp. 3384–3391.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[7] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*, 2014, pp. 675–678.

[8] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[9] Herve Jegou, Matthijs Douze, and Cordelia Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV*, pp. 304–317. 2008.

[10] James Philbin, Ondřej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*, 2007, pp. 1–8.

[11] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky, "Neural codes for image retrieval," in *ECCV*, pp. 584–599. 2014.

[12] Svante Wold, Kim Esbensen, and Paul Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1, pp. 37–52, 1987.

[13] Artem Babenko and Victor Lempitsky, "Aggregating deep convolutional features for image retrieval," in *ICCV*, 2015.

[14] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *ECCV*, pp. 392–407. 2014.

[15] Wei-Lin Ku, Hung-Chun Chou, and Wen-Hsiao Peng, "Discriminatively-learned global image representation using cnn as a local feature extractor for image retrieval," in *2015 Visual Communications and Image Processing (VCIP)*. IEEE, 2015, pp. 1–4.

[16] Konda Reddy Mopuri and R Venkatesh Babu, "Object level deep feature pooling for compact image representation," in *CVPR Workshop*, 2015.

[17] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.

[18] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *CVPR*, 2014, pp. 3286–3293.

[19] Koen EA Van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders, "Segmentation as selective search for object recognition," in *ICCV*, 2011, pp. 1879–1886.

[20] C Lawrence Zitnick and Piotr Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV*, pp. 391–405. 2014.

[21] Ali S Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *CVPR Workshop*, 2014.

[22] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?," *PAMI*, 2015.

[23] Hervé Jégou and Andrew Zisserman, "Triangulation embedding and democratic aggregation for image search," in *CVPR*, 2014, pp. 3310–3317.

[24] David Nister and Henrik Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR*, 2006, vol. 2, pp. 2161–2168.

[25] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.