Efficient Indexing of Regional Maximum Activations of Convolutions using Full-Text Search Engines

Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro
ISTI-CNR, via G. Moruzzi 1, 56124, Pisa, Italy
{name.surname}@isti.cnr.it

ABSTRACT

In this paper, we adapt a surrogate text representation technique to develop efficient instance-level image retrieval using Regional Maximum Activations of Convolutions (R-MAC). R-MAC features have recently showed outstanding performance in visual instance retrieval. However, contrary to the activations of hidden layers adopting ReLU (Rectified Linear Unit), these features are dense. This constitutes an obstacle to the direct use of inverted indexes, which rely on sparsity of data. We propose the use of deep permutations, a recent approach for efficient evaluation of permutations, to generate surrogate text representation of R-MAC features, enabling indexing of visual features as text into a standard search-engine. The experiments, conducted on Lucene, show the effectiveness and efficiency of the proposed approach.

KEYWORDS

Similarity Search, Permutation-Based Indexing, Deep Convolutional Neural Network

ACM Reference format:
Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro. 2017. Efficient Indexing of Regional Maximum Activations of Convolutions using Full-Text Search Engines. In *Proceedings of ICMR '17, June 6–9, 2017, Bucharest, Romania,* , 4 pages.
DOI: http://dx.doi.org/10.1145/3078971.3079035

1 INTRODUCTION

Deep learning has rapidly become the state-of-the-art approach for many computer vision tasks such as classification [21], content-based image retrieval [3, 12], cross-media retrieval [10], smart cameras [1]. A practical and convenient way of using Deep Convolutional Neural Networks (DCNNs) to support fast content-based image retrieval is to treat the neuron activations of the hidden layers as global features [12, 22, 27]. Recently, Tolias et al. [31] have gone further, proposing to use Regional Maximum Activation of Convolutions (R-MAC) as a compact and effective image representation for instance-level retrieval. This feature is the result of the spatial aggregation of the activations of a convolution layer of a DCNN, therefore is robust to scale and translation. Gordo et al. [17] extended the R-MAC representation by improving the region pooling mechanism and including it in a differentiable pipeline trained end-to-end for retrieval tasks.

However, these features are typically of high dimensionality, which prevents the use of space-partitioning data structure, such as kd-tree [9]. For instance, in the well-known AlexNet architecture [21] the output of the sixth layer (fc6) has 4,096 dimensions, while the R-MAC extracted by Gordo et al. [17] produces a 2048-dimensional image descriptor.

To overcome this problem, various partitioning methods have been proposed. In particular, the inverted multi-index uses product quantization both to define the coarse level and for coding residual vectors [6, 25]. This approach combined with binary compressed techniques outperforms the state of the art by a large margin [13]. Our approach, as we will see, is implemented on top of an existing text retrieval engine and requires minimal pre-processing.

Mohedano et al. [23] propose a sparse visual descriptor based on a Bag of Local Convolutional Features (BLCF), which allows fast image retrieval by means of an inverted index. This method, however, relies on a priori learning of a codebook (which involves the use k-means) and the VGG16 [30] pre-trained network, which makes it difficult to retrain the whole pipeline on new set of images.

Our approach to tackle the dimensionality curse problem is still based on the application of approximate access methods, but relies on the permutation approach similar to [5, 11, 14, 24]. The key idea is to represent metric objects (i.e., features) as sequences (permutations) of reference objects, chosen from a predefined set of objects. Similarity queries are executed by searching for data objects whose permutation representations are similar to the query permutation representation. Each permutation is generated by sorting the entire set of reference objects according to their distances from the object to be represented. The total number of reference objects to be used for building permutations depends on the size of the dataset to be indexed and can amount to tens of thousands [5]. In these cases, both indexing time and searching time is affected by the cost of generating permutations for objects being inserted or for the queries.

In this paper, we propose an adaptation of surrogate text representation [15] suitable to regional maximum activations of convolutions features, which exploits the so-called deep permutation approach introduced in [4]. An advantage of this approach lies in its low-computational cost since does not require the distance calculation between the reference objects and the objects to be represented.

The rest of the paper is organized as follows. Section 2 provides background for the reader. In Section 3, we introduce our approach to generate permutations for R-MAC features. Section 4 presents some experimental results using real-life datasets. Section5 concludes the paper.

ICMR '17, June 6–9, 2017, Bucharest, Romania

ACM ISBN 978-1-4503-4701-3/17/06…$15.00
DOI: http://dx.doi.org/10.1145/3078971.3079035

2 BACKGROUND

2.1 Permutation-Based Approach

Given a domain $\mathcal{D}$, a *distance function* $d : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}$, and a fixed set of reference objects $P = \{p_1 \ldots p_n\} \subset \mathcal{D}$ that we call *pivots* or *reference objects*, we define a permutation-based representation Π_o (briefly *permutation*) of an object $o \in \mathcal{D}$ as the sequence of pivots identifiers sorted in ascending order by their distance from o [5, 11, 14, 24].

Formally, the permutation-based representation $\Pi_o = (\Pi_o(1), \ldots, \Pi_o(n))$ lists the pivot identifiers in an order such that $\forall j \in \{1, \ldots, n-1\}$, $d(o, p_{\Pi_o(j)}) \leq d(o, p_{\Pi_o(j+1)})$, where $p_{\Pi_o(j)}$ indicates the pivot at position j in the permutation associated with object o. If we denote as $\Pi_o^{-1}(i)$ the position of a pivot p_i, in the permutation of an object $o \in \mathcal{D}$, so that $\Pi_o(\Pi_o^{-1}(i)) = i$, we obtain the equivalent *inverted* representation of permutations $\Pi_o^{-1} = (\Pi_o^{-1}(1), \ldots, \Pi_o^{-1}(n))$. In Π_o the value in each position of the sequence is the identifier of the pivot in that position. In the inverted representation Π_o^{-1}, each position corresponds to a pivot and the value in each position corresponds to the rank of the corresponding pivot. The inverted representation of permutations Π_o^{-1}, is a vector that we refer to as *vector of permutations*, and which allows us to easily define most of the distance functions between permutations.

Permutations are generally compared using Spearman rho, Kendall Tau, or Spearman Footrule distances. In our implementation, we apply a simple algebraic transformation to permutations in order to compute the same ranking scores that we would obtain with the Spearman Rho distance using dot products. Due to the lack of space, we do not present the proof of this result, however, further details can be found in [15].

2.2 Deep Features

Recently, a new class of image descriptor, built upon Deep Convolutional Neural Networks, have been used as effective alternative to descriptors built using local features such as SIFT, SURF, ORB, BRIEF, etc. DCNNs have attracted enormous interest within the Computer Vision community because of the state-of-the-art results [21] achieved in challenging image classification challenges such as ImageNet Large Scale Visual Recognition Challenge (ILSVRC). In computer vision, DCNN have been used to perform several tasks, including image classification, as well as image retrieval [8, 12] and object detection [16], to cite some. In particular, it has been proved that the multiple levels of representation, which are learned by DCNN on specific task (typically supervised) can be used to transfer learning across tasks [12, 27]. The activation of neurons of a specific layers, in particular the last ones, can be used as features for describing the visual content.

2.3 R-MAC Features

Recently, image descriptors built upon activations of convolutional layers have shown brilliant results in image instance retrieval [7, 28, 31]. Tolias et al. [31] proposed the R-MAC feature representation, which encodes and aggregates several regions of the image in a dense and compact global image representation. To compute a R-MAC feature, an input image is fed to a fully convolutional network pre-trained on ImageNet [29]. The output of the last convolutional layer is max-pooled over different spatial regions at different position and scales, obtaining a feature vector for each region. These vectors are then $l2$-normalized, PCA-whitened, $l2$-normalized again, and finally aggregated by summing them together and $l2$-normalizing the final result. The obtained representation is an effective aggregation of local convolutional features at multiple position and scales that can be compared with the cosine similarity function.

Gordo et al. [17] built on the work of Tolias et al. [31] and inserted the R-MAC feature extractor in a end-to-end differentiable pipeline in order to learn a representation optimized for visual instance retrieval through back-propagation. The whole pipeline is composed by a fully convolutional neural network, a region proposal network, the R-MAC extractor and PCA-like dimensionality reduction layers, and it is trained using a ranking loss based on image triplets. The obtained pipeline can extract an optimized R-MAC feature representation that outperforms methods based on costly local features and spatial geometry verification.

An additional performance boost is obtained using state-of-the-art deep convolutional architectures, such as very deep residual networks [18], and aggregating descriptors extracted at multiple resolutions. A multi-resolution R-MAC descriptor is obtained feeding the network with images at different resolutions and then aggregating the obtained representations by summing them together and then performing a final $l2$-normalization.

In our work, we used the ResNet-101 trained model provided by [17] as a R-MAC feature extractor, which has been shown to achieve the best performance on standard benchmarmks. We extracted the R-MAC features using fixed regions at two different scales as proposed in [31] instead of using the region proposal network. Defined S as the size in pixel of the minimum side of an image, we extracted both single-resolution descriptors (with $S = 800$) and multi-resolution ones (aggregating descriptors with $S = 550$, 800 and 1050).

3 SURROGATE TEXT REPRESENTATION FOR DEEP FEATURES

As introduced above, the basic idea of permutation-based indexing techniques is to represent feature objects as permutations built using a set of reference object identifiers as permutants.

Using the permutation-based representation, the similarity between two objects is estimated computing the similarity between the two corresponding permutations, rather than using the original distance function. The rationale behind this is that, when permutations are built using this strategy, objects that are very close one to the other, have similar permutation representations as well. In other words, if two objects are very close one to the other, they will sort the set of reference objects in a very similar way.

Notice however that the relevant aspect when building permutations is the capability of generating sequences of identifiers (permutations) in such a way that similar objects have similar permutations as well. Sorting a set of reference objects, according to their distance with the object to be represented is just one, yet effective, approach.

When objects to be indexed are vectors as in our case of deep features, we can use the approach presented in [4], which allows us to generate sequence of identifiers not associated with reference objects. The basic idea is as follows. Permutants are the indexes of elements of the deep feature vectors. Given a deep feature vector, the corresponding permutation is obtained by sorting the indexes of the elements of the vector, in descending order with respect to the values of the corresponding elements. Suppose for instance the feature vector is $fv = [0.1, 0.3, 0.4, 0, 0.2]$[1]. The permutation-based representation of fv is $\Pi_{fv} = (3, 2, 5, 1, 4)$, that is permutant (index) 3 is in position 1, permutant 2 is in position 2, permutant 5 is in position 3, etc. The permutation vectors, introduced in Section 2 is $\Pi^{-1}_{fv} = (4, 2, 1, 5, 3)$, that is permutant (index) 1 is in position 4, permutant 2 is in position 2, permutant 3 is in position 1, etc.

The intuition behind this is that features in the high levels of the neural network carry out some sort of high-level visual information. We can imagine that individual dimensions of the deep feature vectors represent some sort of visual concept, and that the value of each dimension specifies the importance of that visual concept in the image. Similar deep feature vectors sort the visual concepts (the dimensions) in the same way, according to the activation values.

Without entering the technical details of this approach (for that, we refer the reader to [4]), let us just stress the fact that although the vector of permutations are of the same dimension of DCNN vectors, the advantage consists in that they be easily encoded into an inverted index. Moreover, following the intuition that the most relevant information of the permutation is in the very first, we can truncate the vector of permutations to the top-K (i.e., truncated permutations at K). The element of the vectors beyond K can be ignored, this approach allows us to modulate the size of vectors and reduce the size of the index by introducing more sparsity.

In order to index the permutation vectors with a text-retrieval engine as Lucene, we use the surrogate text representation introduced in [15], which simply consists in assigning a codeword to each item of the permutation vector Π^{-1} and repeating the codewords a number of times equal to the complement of the rank of each item within the permutation. For instance, let τ_i be the codeword corresponding to the i-th component of the permutation vector, for the vector $\Pi^{-1}_{fv} = (4, 2, 1, 5, 3)$, we generate the following surrogate text: "$\tau_1\ \tau_1\ \tau_2\ \tau_2\ \tau_2\ \tau_2\ \tau_3\ \tau_3\ \tau_3\ \tau_3\ \tau_3\ \tau_4\ \tau_5\ \tau_5\ \tau_5$".

4 EXPERIMENTAL EVALUATION

The assessment of the proposed algorithm in a multimedia information retrieval task was performed using the R-MAC features extracted as explained above from *INRIA Holidays* [19] and *Oxford Buildings* [26] datasets.

INRIA Holidays [19] is a collection of 1,491 images, which mainly contains personal holidays photos. The images are of high resolution and represent a large variety of scene type (natural, man-made, water, fire effects,etc). The authors selected 500 queries and manually identified a list of qualified results for each of them. As in [20], we merged the Holidays dataset with the distraction dataset MIRFlickr including 1M images [2].

Oxford Buildings [26] is composed by 5062 images of 11 Oxford landmarks downloaded from Flickr. A manually labelled groundtruth is available for 5 queries for each landmark, for a total of 55 queries. As for *INRIA Holidays*, we merged the dataset with the distraction dataset Flickr100k including 100k images [3].

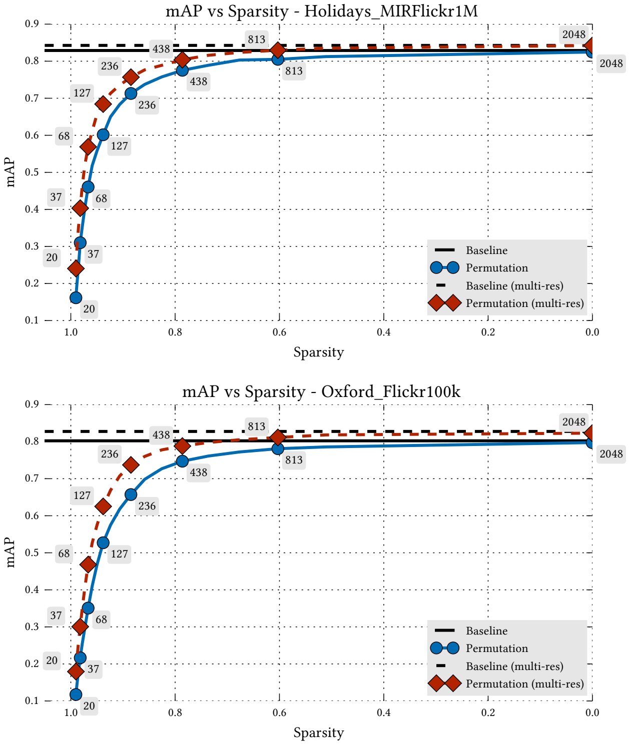

Figure 1: mAP vs sparsity for increasing values of K (reported near each point). Note that, the sparsity of the dataset is given by $\frac{K}{2048}$. The horizontal lines represent the levels of *mAPs* for the baselines, i.e., using the original R-MAC vectors.

We generated different sets of permutations from the original features with different values of K (i.e., we consider truncated at K of the permutations), and for each K, we measured the *mAP* obtained and the query execution times. Results on both datasets are reported in Figure 1. The experiments show the *mAP* of our Lucene implementation as function of the sparsity of the database introduced by the truncation at K. The greater is K (indicated near the point in the graphs) the lower is the sparsity, and hence the greater is the *mAP*. The levels of *mAPs* for the baselines, i.e., using the original R-MAC vectors, are also reported in the figure. These levels can be considered as the upper-bounds for our approach since we are dealing with an approximate approach. However, as it is possible to see, for a sparsity level of about 80%, we reach satisfactory levels of effectiveness.

In order to see the impact of the sparsity of the database, in Figure 2 we report the average query time versus the parameter K. Clearly, by increasing K the query time increases. However, for larger database sizes, a strategy of query reduction similar to the one presented in [2] can be used.

[1] In reality, the number of dimensions is 2,048 or more.

[2] http://press.liacs.nl/mirflickr/

[3] http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/

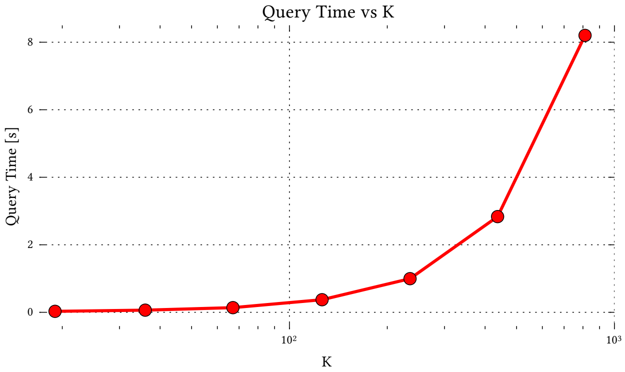

Figure 2: Average query times in seconds on INRIA Holidays dataset + MIRFlickr1M distractor set for different values of K on Lucene.

5 CONCLUSION

In this paper, we present an approach for indexing R-MAC features as permutations built upon the full-text retrieval engine Lucene that exploits surrogate text representation. The advantage is that comparing to the classical approach based on permutation, this technique does not need to compute distances between pivots and data objects but uses the same activation values of the neural network as a source for associating Deep Feature vectors with permutations.

In this preliminary study, we obtained encouraging results on two important benchmarks of image retrieval. In future, we plan test our approach on the more challenging Yahoo Flickr Creative Commons 100 Million (YFCC100M) dataset available at http://bit.ly/yfcc100md.

REFERENCES

[1] Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, Carlo Meghini, and Claudio Vairo. 2017. Deep learning for decentralized parking lot occupancy detection. *Expert Systems with Applications* 72 (2017), 327–334.

[2] Giuseppe Amato, Franca Debole, Fabrizio Falchi, Claudio Gennaro, and Fausto Rabitti. 2016. Large Scale Indexing and Searching Deep Convolutional Neural Network Features. In *Proceeding of the 18th International Conference on Big Data Analytics and Knowledge Discovery (DaWaK 2016)*. Springer. to appear.

[3] Giuseppe Amato, Fabrizio Falchi, Claudio Gennaro, and Fausto Rabitti. 2016. YFCC100M HybridNet fc6 Deep Features for Content-Based Image Retrieval. In *Proceedings of the 2016 ACM Workshop on Multimedia COMMONS*. ACM, 11–18.

[4] Giuseppe Amato, Fabrizio Falchi, Claudio Gennaro, and Lucia Vadicamo. 2016. Deep Permutations: Deep Convolutional Neural Networks and Permutation-Based Indexing. In *International Conference on Similarity Search and Applications*. Springer, 93–106.

[5] Giuseppe Amato, Claudio Gennaro, and Pasquale Savino. 2014. MI-File: using inverted files for scalable approximate similarity search. *Multimedia Tools and Applications* 71, 3 (2014), 1333–1362. DOI: http://dx.doi.org/10.1007/s11042-012-1271-1

[6] Artem Babenko and Victor Lempitsky. 2012. The inverted multi-index. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 3069–3076.

[7] Artem Babenko and Victor Lempitsky. 2015. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE international conference on computer vision*. 1269–1277.

[8] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. 2014. Neural codes for image retrieval. In *Computer Vision–ECCV 2014*. Springer, 584–599.

[9] Jon Louis Bentley. 1975. Multidimensional binary search trees used for associative searching. *Commun. ACM* 18, 9 (1975), 509–517.

[10] Fabio Carrara, Andrea Esuli, Tiziano Fagni, Fabrizio Falchi, and Alejandro Moreo Fernández. 2016. Picture it in your mind: Generating high level visual representations from textual descriptions. *arXiv preprint arXiv:1606.07287* (2016).

[11] Edgar Chávez, Karina Figueroa, and Gonzalo Navarro. 2008. Effective Proximity Retrieval by Ordering Permutations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30, 9 (2008), 1647–1658.

[12] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2013. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531* (2013).

[13] Matthijs Douze, Hervé Jégou, and Florent Perronnin. 2016. *Polysemous Codes*. Springer International Publishing, Cham, 785–801.

[14] Andrea Esuli. 2012. Use of permutation prefixes for efficient and scalable approximate similarity search. *Information Processing & Management* 48, 5 (2012), 889–902.

[15] Claudio Gennaro, Giuseppe Amato, Paolo Bolettieri, and Pasquale Savino. 2010. An Approach to Content-Based Image Retrieval Based on the Lucene Search Engine Library. In *Research and Advanced Technology for Digital Libraries*, Mounia Lalmas, Joemon Jose, Andreas Rauber, Fabrizio Sebastiani, and Ingo Frommholz (Eds.). Lecture Notes in Computer Science, Vol. 6273. Springer Berlin Heidelberg, 55–66. http://dx.doi.org/10.1007/978-3-642-15464-5_8

[16] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 580–587.

[17] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. 2016. End-to-end learning of deep visual representations for image retrieval. *arXiv preprint arXiv:1610.07940* (2016).

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385* (2015).

[19] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. 2008. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In *Computer Vision – ECCV 2008*, David Forsyth, Philip Torr, and Andrew Zisserman (Eds.). Lecture Notes in Computer Science, Vol. 5302. Springer Berlin Heidelberg, 304–317. http://dx.doi.org/10.1007/978-3-540-88682-2_24

[20] H. Jégou, M. Douze, and C. Schmid. 2009. Packing bag-of-features. In *Computer Vision, 2009 IEEE 12th International Conference on*. 2357 –2364. DOI: http://dx.doi.org/10.1109/ICCV.2009.5459419

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[23] Eva Mohedano, Kevin McGuinness, Noel E. O’Connor, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. 2016. Bags of Local Convolutional Features for Scalable Instance Search. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval (ICMR ’16)*. ACM, New York, NY, USA, 327–331.

[24] David Novak, Martin Kyselak, and Pavel Zezula. 2010. On locality-sensitive indexing in generic metric spaces. In *Proceedings of the Third International Conference on Similarity Search and Applications (SISAP ’10)*. ACM, 59–66.

[25] Loïc Paulevé, Hervé Jégou, and Laurent Amsaleg. 2010. Locality sensitive hashing: A comparison of hash function types and querying mechanisms. *Pattern Recognition Letters* 31, 11 (2010), 1348–1358.

[26] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. 2007. Object Retrieval with Large Vocabularies and Fast Spatial Matching. In *Computer Vision and Pattern Recognition, 2007. CVPR 2007. IEEE Conference on*. 1–8.

[27] Ali S Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*. IEEE, 512–519.

[28] Ali Sharif Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. 2014. Visual instance retrieval with deep convolutional networks. *arXiv preprint arXiv:1412.6574* (2014).

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115, 3 (2015), 211–252. DOI: http://dx.doi.org/10.1007/s11263-015-0816-y

[30] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[31] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. 2015. Particular object retrieval with integral max-pooling of CNN activations. *arXiv preprint arXiv:1511.05879* (2015).