Analysis of Feature Learning and Feature Pooling for Image Recognition

Y-Lan Boureau[1,2], Francis Bach[1], Yann LeCun[2], and Jean Ponce[3,1]

[1] INRIA - WILLOW Project-Team
Laboratoire d'Informatique de l'Ecole Normale Supérieure (INRIA/ENS/CNRS UMR 8548)
23, avenue d'Italie 75214 Paris CEDEX 13, France
[2] The Courant Institute of Mathematical Sciences - New York University
715 Broadway, 12th Floor, New York, NY 10003
[3] Ecole Normale Supérieure - WILLOW Project-Team
ylan@cs.nyu.edu, francis.bach@inria.fr, yann@cs.nyu.edu, jean.ponce@ens.fr

Modern approaches to category-level object or scene recognition usually follow the classical supervised classification paradigm where, given some global feature extraction process, the feature vectors associated with a number of positive and negative training pictures are used to train a classifier such as a support vector machine. Once trained, this classifier can be used to classify test images using the corresponding feature vectors. A key ingredient in these approaches is the design of the feature extraction process. A three-step pipeline combining 1) non-linear local encoding, 2) local spatial pooling, and 3) concatenation of weighted representations of neighboring regions of interest has emerged as a particularly successful feature extraction module. These operations were used in early models of image recognition (and more recent related models), like the neocognitron [3], convolutional networks [4, 6, 7, 8, 12], the HMAX class of models [13], and Pinto et al.'s object recognition model [11]. Powerful image descriptors like Lowe's *SIFT* descriptor [9], the *GIST* descriptor of Oliva and Torralba [10], the *HOG* operator of Dalal and Triggs [2] can all be viewed as examples of this same three-step paradigm. The original bag of features model [14], the *spatial pyramid* of Lazebnik et al. [5] and its variants [15, 16], extract features by applying sequentially two instances of the three-step module to input images: first to image pixels to obtain local SIFT image descriptors, then again to the resulting set of local descriptors to obtain the global image representation.

Here, we seek to find the best combination of bottom-level descriptors concatenation, encoding and pooling for recognition within the popular spatial pyramid framework. Using denser sampling of bottom-level descriptors, encoding nearby descriptors jointly by a sparse representation over a dictionary trained discriminatively by approximatively minimizing a supervised logistic loss over the whole-image representation, combined with max pooling and a linear classifier, yields the best known recognition rate of 85.6 ± 0.2 on the 15 Scenes dataset, and the best recognition rate of 75.2 ± 0.9 on the Caltech-101 dataset for a system with a single feature type (with 30 training examples).

We also investigate whether the improvements obtained with some specific combinations of steps and classifiers (e.g., sparse encoding with max pooling and linear classification) can be factored into relatively independent contributions from each deviation from the most simple framework. Performance of the bag-of-features framework [17] has been improved dramatically by using more sophisticated coding schemes [15, 16], pooling neighborhoods [5] or operations [16], or category-specific concatenation weights [1], while retaining the same bottom-level module. We compare encoding steps that output either a 1-of-K binary code (hard vector quantization where the only nonzero component is the one corresponding to the closest codeword in some codebook) or a continuous code (soft vector quantization or sparse coding). For the second (pooling) step, the set of all codes are over the cells of a spatial pyramid are summarized by a single feature vector, by taking either the average (average pooling) or the maximum value of each feature (max pooling). We simply concatenate the vectors representing each cell of a three-level spatial pyramid to form the input to the classifier, which can be either a linear SVM, or an intersection kernel SVM. Results of cross-comparisons presented on Table 1 show two consistent patterns: (1) Max pooling almost always improves results over average pooling, and dramatically so when a linear classifier is used, except that results are not significantly different when hard quantization is used with an intersection kernel; (2) irrespective of pooling and classifier, sparse coding performs better than soft quantization, which performs better than hard quantization - the only exception being that soft quantization performs notably worse than hard quantization when combined with max pooling and a linear classifier.

Method	Caltech-101, 30 training examples		15 Scenes, 100 training examples	
	Average Pool	Max Pool	Average Pool	Max Pool
Hard quant., linear	51.42 ± 0.91 [256]	64.26 ± 0.90 [256]	73.90 ± 0.87 [1024]	80.11 ± 0.53 [1024]
Hard quant., intersect	64.17 ± 1.02 [256]	64.26 ± 0.90 [256]	80.84 ± 0.35 [256]	80.11 ± 0.53 [1024]
Soft quant., linear	57.92 ± 1.54 [1024]	62.29 ± 1.38 [512]	75.63 ± 0.54 [1024]	76.83 ± 0.89 [1024]
Soft quant., intersect	66.12 ± 1.19 [512]	70.57 ± 0.97 [1024]	81.19 ± 0.40 [1024]	82.99 ± 0.73 [1024]
Sparse codes, linear	61.32 ± 1.26 [1024]	71.52 ± 1.13 [1024]	76.91 ± 0.57 [1024]	83.12 ± 0.56 [1024]
Sparse codes, intersect	70.27 ± 1.29 [1024]	71.81 ± 0.96 [1024]	83.15 ± 0.35 [1024]	84.13 ± 0.45 [1024]

Table 1: Average recognition rate on Caltech-101 and 15-Scenes benchmarks, for various combinations of coding schemes, pooling, and classifiers. Codebook size (inside brackets) is the one that gives best results between 256, 512 and 1024. Classifier is either a linear or an intersection kernel SVM. Linear and histogram intersection kernels are identical when using hard quantization with max pooling (as taking the minimum or the product is the same for binary vectors), but results have been included for both to preserve the symmetry of the table. (1) Max pooling almost always improves results over average pooling, and dramatically so when a linear classifier is used. (2) Irrespective of pooling and classifier, sparse coding performs better than soft quantization, which performs better than hard quantization - the only exception being that soft quantization performs worse than hard quantization when combined with max pooling and a linear classifier.

Finally, we give a theoretical analysis of pooling schemes, backed by experiments, which shows that max pooling should be preferred over average pooling when features have a low probability of being active (e.g., with large codebooks) and the pool cardinality is large enough. We also demonstrate that the optimal cardinality over which to perform max pooling is not always the full cardinality of all available samples in the pool.

References

[1] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the International Conference on Image and Video Retrieval*, 2007.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, volume 2, pages 886–893, June 2005.

[3] K. Fukushima and S. Miyake. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, 1982.

[4] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *(ICCV'09)*. IEEE, 2009.

[5] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, volume II, pages 2169–2178, 2006.

[6] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In D. Touretzky, editor, *Advances in Neural Information Processing Systems 2 (NIPS*89)*, Denver, CO, 1990. Morgan Kaufman.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[8] H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *International Conference on Machine Learning (to appear)*, 2009.

[9] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. of Comp. Vision*, 60(4):91–110, 2004.

[10] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int. J. of Comp. Vision*, 42(3):145–175, 2001.

[11] N. Pinto, D. Cox, and J. DiCarlo. Why is real-world visual object recognition hard. *PLoS Computational Biology*, 4(1):151–156, 2008.

[12] M. Ranzato, Y. Boureau, and Y. LeCun. Sparse feature learning for deep belief networks. In *Advances in Neural Information Processing Systems (NIPS 2007)*, 2007.

[13] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *CVPR*, 2005.

[14] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, Oct. 2003.

[15] J. van Gemert, J. Geusebroek, C. Veenman, and A. Smeulders. Kernel codebooks for scene categorization. In *ECCV*, 2008.

[16] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009*, 2009.

[17] J. G. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, jun 2007.

Topic: visual processing and pattern recognition
Preference: oral