Learning Smooth Pooling Regions for Visual Recognition

Mateusz Malinowski
mmalinow@mpi-inf.mpg.de
Mario Fritz
mfritz@mpi-inf.mpg.de

Scalable Learning and Perception
Max Planck Institute for Informatics
Saarbrücken, Germany

From the early HMAX model to Spatial Pyramid Matching [2, 4], spatial pooling has played an important role in visual recognition pipelines. By aggregating local statistics, it equips the recognition architectures with a certain degree of robustness to translation and deformation yet preserving spatial information. Despite of its predominance in current recognition systems, we have seen little progress to fully adapt the pooling strategy to the task at hand, and this critical decision is most prominently based on hand-crafted layouts.

We propose in this paper a flexible parameterization that allows for a richer set of possible pooling regions and show results on classification tasks using two different pipelines [1, 3]. The higher-level pooling representation is learned jointly with the classifier to support the recognition task. In order to deal with the increased flexibility of the model, we investigate different regularizers and efficient learning schemes. In particular, we propose a smoothness regularizer that yields the strongest performance improvements in our experiments.

The simplest form of the spatial pooling is computing histogram over the whole image. This can be expressed as $\Sigma(U) := \sum_{j=1}^{M} u_j$, where $u_j \in \mathbb{R}^K$ is an encoded patch extracted from the image (out of M such codes) and an index j refers to the spatial location that the code originates from[1]. Another popular pooling scheme that has been proven successful [5] is max-pooling: $\mathbb{M}(U) := \max_{j=1}^{M} u_j$. Since the pooling approach looses spatial information of the codes, Lazebnik et al. [2] proposed to first divide the image into subregions, and afterwards to create pooled features by concatenating histograms computed over each subregion. There are two problems with such an approach: first, the division is largely arbitrary and in particular independent of the data; second, discretization artifacts occur as spatially nearby codes can belong to two different regions as the 'hard' division is made.

In our paper we address both problems by using a parameterized version of the pooling operator

$$\Theta_w(U) := \rho_{j=1}^{M}(w_j \circ u_j) \tag{1}$$

where $a \circ b$ is the element-wise multiplication, and $\rho \in \{\max, \sum\}$ is a pooling function. Moreover, we have investigated a few regularization terms on the pooling weights showing that smooth pooling regions are crucial.

Consider a sampling scheme and an encoding method producing M codes each K dimensional. Every coordinate of the code is an input layer for the multilayer perceptron. Then we connect every j-th input unit at the layer k to the l-th pooling unit a_l^k via the relation $w_{lj}^k u_j^k$. Since the receptive field of the pooling unit a_l^k consists of all codes at the layer k, we have $a_l^k := \sum_{j=1}^{M} w_{lj}^k u_j^k$ or $a_l^k := \max_{j=1}^{M} w_{lj}^k u_j^k$. Next, we connect all pooling units with the classifier allowing the information to circulate between the pooling layers and the classifier (Fig. 1). The latter has an access to the class membership and can use this information back to the pooling stage to shape better pooling regions. For the training purpose we have derived the backpropagation rules used for the weights' update.

To make the whole approach more scalable towards bigger dictionaries we introduce two approximations. The first one, called pre-pooling, uses standard pooling scheme to aggregate the codes over small neighbourhood before our joint training of the pooling regions together with the classifier's parameters is applied. The second approximation (batches) divides a K dimensional code into $\frac{K}{D}$ batches, each D dimensional. The latter enables embarrassingly parallel training of the model with sizable dictionaries. Our implementation of the proposed method is publicly available at `http://www.d2.mpi-inf.mpg.de/datasets`.

We have evaluated our method on two classification datasets CIFAR-10 and CIFAR-100 following Coates and Ng [1] pipeline, and UIUC sports events following Li-Jia et al. [3].

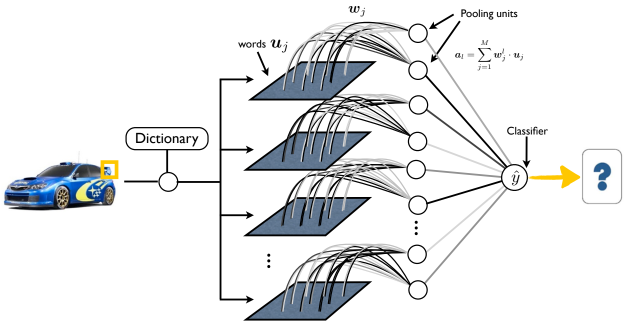
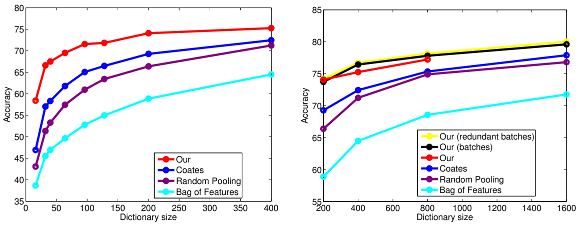


Figure 1: Sketch of our architecture. We encode the patches extracted from the images using popular encoding method. Next, we couple every position of such encoded patches with the classifier via the pooling weights. Our method learns both the pooling weights and classifier's parameters at the same time by using the backpropagation rule.

Figures 2(a) and 2(b) show the classification accuracy of our model against the baseline [1] on CIFAR-10. Our method outperforms the approach of Coates by 10% for dictionary size 16 (our method achieves the accuracy 57.07%, whereas the baseline only 46.93%). For bigger dictionaries (1600) with an accuracy for the batched model of 79.6% we outperform the Coates baseline by 1.7%. On CIFAR-100 our model achieves 56.29% outperforming the baseline by 4.63%. Lastly, we investigate events recognition on the UIUC Sports database based on object bank features [3]. Our learnable pooling strategy achieves accuracy 79.4%, about 3.1% higher then the hand-crafted scheme used in Li-Jia et al. [3]. Our experiments show the importance of the optimized pooling strategy.



(a)

(b)

Figure 2: Figure 2(a) shows accuracy of the classification with respect to the number of dictionary elements on smaller dictionaries. Figure 2(b) shows the accuracy of the classification for bigger dictionaries when our approximation is used (batches, and the redundant batches).

[1] A. Coates and A. Y. Ng. The importance of encoding versus training with sparse coding and vector quantization. In *ICML*, 2011.

[2] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[3] L. Li-Jia, S. Hao, E. P. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *NIPS*, 2010.

[4] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2009.

[5] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[1] That is $j = (x, y)$ where x and y refer to the spatial location of the center of the extracted patch.