Learning Receptive Fields for Pooling from Tensors of Feature Response

Can Xu and Nuno Vasconcelos
Department of Electrical and Computer Engineering
University of California, San Diego
{canxu, nuno}@ucsd.edu

Abstract

A new method for learning pooling receptive fields for recognition is presented. The method exploits the statistics of the 3D tensor of SIFT responses to an image. It is argued that the eigentensors of this tensor contain the information necessary for learning class-specific pooling receptive fields. It is shown that this information can be extracted by a simple PCA analysis of a specific tensor flattening. A novel algorithm is then proposed for fitting box-like receptive fields to the eigenimages extracted from a collection of images. The resulting receptive fields can be combined with any of the recently popular coding strategies for image classification. This combination is experimentally shown to improve classification accuracy for both vector quantization and Fisher vector (FV) encodings. It is then shown that the combination of the FV encoding with the proposed receptive fields has state-of-the-art performance for both object recognition and scene classification. Finally, when compared with previous attempts at learning receptive fields for pooling, the method is simpler and achieves better results.

1. Introduction

Object recognition and scene classification are two important problems in computer vision. A popular approach to these problems is to rely on the spatial pyramid matching (SPM) architecture of Figure 1 [17]. Image descriptors are first extracted from a grid of image locations, mapping the image into a 3D tensor, where two dimensions correspond to image coordinates and the third to features. Each descriptor is then converted to a high-dimensional vector, through a *descriptor encoding* procedure. The entries of the transformed tensor are finally *pooled* along the two spatial dimensions to produce the image representation for recognition. Substantial research has been devoted to different components of this architecture.

Early contributions produced a number of popular descriptors, such as SIFT [20], HOG [5], SURF [1], or LBP [21]. Lately, there has been more emphasis on encoding methods. The classical encoding is based on vector quantization (VQ), assigning each descriptor to the closest codeword in a codebook learned from a generic image set. The image to classify is then represented by an histogram of codeword assignments. Since this is a coarse estimate of the descriptor probability density, many attempts have been made to develop more sophisticated estimates. Early research focused on the clustering algorithm used to produce the codebook, comparing soft to hard codeword assignments [26] discriminant vs. non-discriminative learning [13], etc. More recently, there has been interest in finer density representations, which encode differences between descriptors and codewords, e.g. supervectors (SV) [35] or Fisher vectors (FV) [24, 25]. Alternatively, several authors have proposed sparse coding methods, which represent each descriptor by a sparse linear combination of basis functions from a learned dictionary. Popular sparse representations include the sparse-coded SPM (ScSPM) method of [34], and the locally constrained linear coding (LLC) method of [33].

In contrast to all this work on descriptors and encodings, there has been relatively little research on pooling schemes. SPM partitions the image into predefined subblocks of multiple scales, and relies on a summation operator to combine the descriptor encodings within each block. The resulting encoding is a high-dimensional vector, which can be interpreted as a collection of histograms over the different image blocks. While it is now well established that the SPM pooling operation is critical for the success of the classification, it is unclear that either its arbitrary blocks or pooling operator are optimally tuned to the structure of images. This motivated research on the selection of pooling operators, namely on the benefits of adopting a sum or a maximum operator [29, 11, 3]. Recently, a few works have shown that it can be quite beneficial to learn alternative pooling regions [12, 28]. However, these methods have high computational complexity and are intractable for high dimensional features, such as Fisher vectors.

In this work, we show that similar gains can be obtained with methods of much smaller complexity, by exploiting the

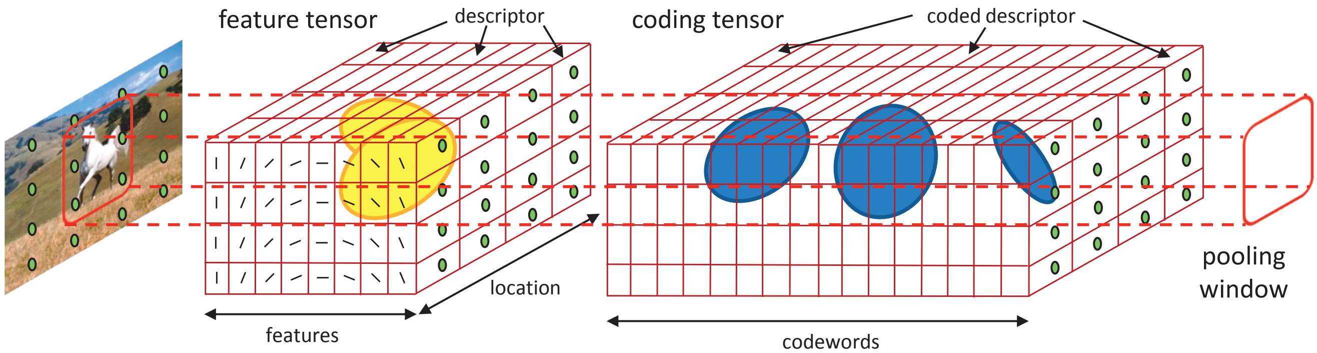


Figure 1: Under SPM, an image is first mapped into a tensor of feature (descriptor) responses and subsequently into a tensor of coding coefficients. Discriminant class information is coded as blobs of energy within these tensors. Discriminant pooling windows can be recovered by detecting the eigenblobs of tensor response to images of the class.

statistics of the 3D feature tensor of Figure 1. The hypothesis is that pooling regions can be learned by identifying blobs of discriminant response, for each image class, in the tensor. We propose to identify these blobs as the locations where responses to images of the class have most energy. This equates pooling regions to eigentensors of the 3D tensor. One appealing property of this hypothesis is that the computation of eigentensors is computationally trivial. It reduces to computing a principal component analysis (PCA) of various 2D flattenings of the tensor [32, 31]. In fact, we show that a particular flattening corresponds to the popular PCA-SIFT representation [14]. For learning to pool this is the most uninteresting flattening, since it does not preserve image topology.

We explore alternative flattenings, which preserve location information, and use them to identify potential pooling regions. We then introduce an algorithm for learning a set of box-like pooling regions that best approximates the eigentensors over a collection of images. This is a k-means like procedure, iterating between the assignment of eigentensor regions to boxes and the reshaping of boxes to fit regions. It is a generic algorithm that could be applied to many vision problems, e.g. determining an object bounding box from a saliency map or the output of an object detector [19]. Experimental results show that the learned fields significantly outperform SPM and that the gains hold across encodings. We adopt the FV encoding and show that its combination with the proposed learned fields achieves state of the art results on various datasets.

2. Learning receptive fields through tensors

We start by identifying candidate pooling regions from the statistics of the tensor of feature responses.

2.1. From image to tensor

Figure 1 illustrates how the SPM representation can be mapped into a tensor. Images are mapped into 128D SIFT descriptors, which are stacked so as to maintain the image topology. In the resulting 3D tensor, each vector in the direction orthogonal to the image coordinates corresponds to a SIFT descriptor. The resulting structure is denoted the feature tensor. SIFT descriptors are then mapped into high-dimensional codes. For example, into 1,024D vectors whose entries are the assignments of a SIFT descriptor to a codebook of $1,024$ codewords. Again, these vectors are stacked to produce a tensor that respects the image topology. This is the coding tensor. SPM applies a set of pooling operators to this tensor, producing a feature vector that is fed to a support vector machine for classification. Pooling operators are sums over a preset pyramid of multi-scale image windows. The feature vector is an histogram of the codeword assignments in the associated image regions.

2.2. Learning to pool

While SPM relies on a sum operator and arbitrary pooling regions, some alternatives have been investigated in the recent literature. Most of this effort has been on improved pooling operators. For example, Yang *et al.* [34] used max instead of sum pooling. Boureau and Ponce [3] later provided a theoretical analysis, showing that max pooling is well suited for features with a low probability of activation. However, other experiments have shown mixed results for the benefits of the max over the sum operation [11]. Our work does not address the pooling operator, but the learning of regions of support (windows) for pooling operators. These are denoted as the *receptive fields* for pooling.

The learning of receptive fields has been the subject of attention in the last few years. Jia *et al.* [12] started from a large number of candidate rectangular boxes, and proposed a greedy method for learning the optimal receptive fields from this set. Russakovsky *et al.* [28] proposed an object-centric spatial pooling approach, which jointly performs classification and localization through bootstrapping. These methods have demonstrated improved performance on several benchmarks, but have substantial complexity and

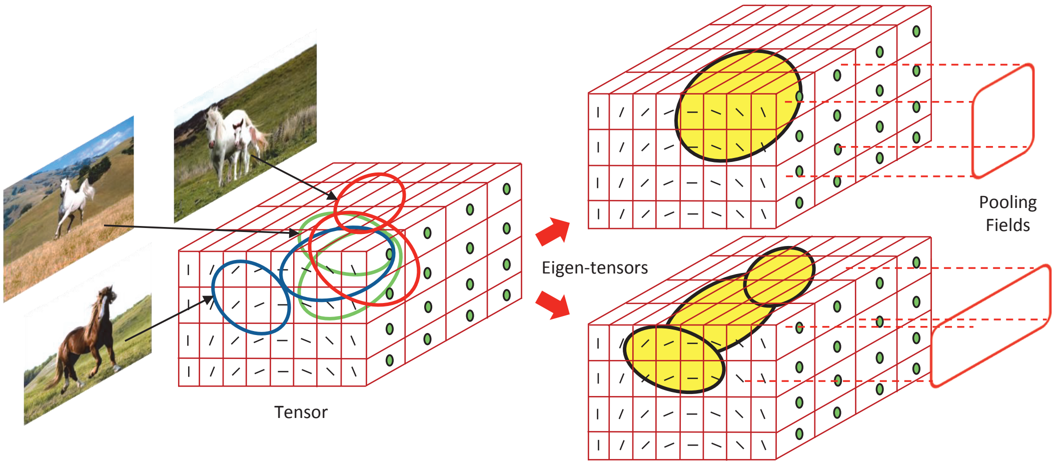


Figure 2: Receptive field learning. An image class defines an energy distribution in tensor space. An eigen-tensor decomposition produces the collection of tensor basis functions that best approximate this energy distribution. Discriminant pooling windows can be obtained from these eigentensors.

become intractable for high dimensional encodings, such as Fisher vectors. For example, the method of [28] is too complex for standard SVM solvers, even when applied to a relatively low-dimensional VQ encoding. A slightly different approach was proposed by [15], which replace SPM with a spatial Fisher vector (SFV). This is a representation of the spatial distribution of SIFT descriptors.

In this work we seek a solution of much lower complexity, based on the intuition of Figure 2. Images are viewed as distributions of energy in the feature tensor. The discriminant visual structures of a given image class give rise to consistent energy blobs. Given a set of images from the class, these blobs can be approximated by a small set of tensor eigenfunctions, denoted eigen-tensors. The image locations containing the bulk of the energy of the responses to the class are then recovered from these eigen-tensors and used as receptive fields for pooling.

2.3. Tensor Analysis

A tensor $\mathcal{T} \in \mathbb{R}^{k_1 \times k_2 \times \ldots k_N}$ of order N is an ND array of $k_1 \times k_2 \times \ldots k_N$ entries. In this work, we restrict our attention to tensors of order 3, whose coordinates correspond to image locations (k_1), features (k_2), and scales (k_3). We rely on SIFT features, and different image scales are obtained with a Gaussian pyramid [20]. A k_n-D vector obtained from $\mathcal{T}$ by varying index n while keeping the others fixed is the mode-n vector of $\mathcal{T}$. Mode-n vectors are the column vectors of the matrix $\mathbf{T}_n \in \mathbb{R}^{k_n \times \prod_{i \neq n} k_i}$ that results from flattening the tensor $\mathcal{T}$ with respect to dimension n. Fig. 3 illustrates the three flattenings - $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3$ - of the 3^{rd} order tensor. The N-mode singular value decomposition (SVD), is an extension of the matrix SVD that expresses the tensor as a product of N-orthogonal spaces

$$\mathcal{T} = \mathcal{Z} \times_1 \mathbf{U}_1 \ldots \times_N \mathbf{U}_N \tag{1}$$

where $\mathcal{A} \times_n \mathbf{M}$ is the n-mode product of a tensor $\mathcal{A}$ with a matrix $\mathbf{M}$, defined as

$$\mathcal{B} = \mathcal{A} \times_n \mathbf{M} \quad \Leftrightarrow \quad \mathbf{B}_n = \mathbf{M}\mathbf{A}_n. \tag{2}$$

The mode matrix $\mathbf{U}_n$ of (1) contains the orthonormal vectors spanning the column space of the mode-n flattening matrix $\mathbf{T}_n$ of Figure 3. The tensor SVD is computed by first finding the matrices $\mathbf{U}_i$, through a left-SVD[1] of each matrix $\mathbf{T}_n$, and then computing $\mathcal{Z}$ with

$$\mathcal{Z} = \mathcal{T} \times_1 \mathbf{U}_1^T \ldots \times_N \mathbf{U}_N^T. \tag{3}$$

2.4. Tensor statistics

It is well known that, given a matrix $\mathbf{D}$ whose columns are data vectors $\mathbf{x}_i$, the left-SVD of $\mathbf{D}$ is the matrix $\mathbf{U}$ of principal components (eigenvectors of the sample covariance) of $\mathbf{D}$. The singular values λ_i associated with the columns $\mathbf{u}_i$ of $\mathbf{U}$ measure the variance of the data in $\mathbf{D}$ along the principal components $\mathbf{u}_i$. The tensor SVD is thus equivalent to performing a PCA of the data matrices $\mathbf{T}_n$ associated with the different flattenings of the tensor. When, as is illustrated in Figure 3, $\mathcal{T}$ is flattened along the feature dimension, i.e. for the matrix $\mathbf{T}_1$, each column is a feature vector. If $\mathcal{T}$ is built from image gradients, performing an SVD on this matrix is equivalent to computing the PCA-SIFT descriptor [14], frequently used as a low-dimensional counterpart to SIFT. This is a PCA on a view of the data that ignores feature location.

The location information is, however, available in the tensor. It is captured by the flattening $\mathbf{T}_2$ along the dimension of image locations. The SVD of this matrix produces principal components that reflect the *spatial distribution of energy* in the tensor. In this case, the left singular-vectors $\mathbf{u}_i$ are eigenimages with blob-like structure that reflects the

[1] by left-SVD, we mean computing an SVD and taking the left matrix.

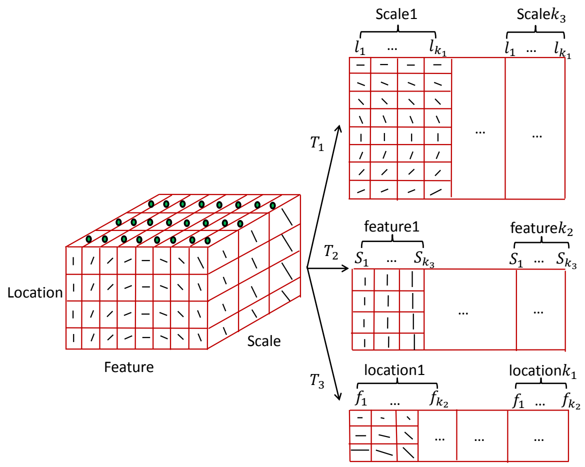


Figure 3: The three possible flattenings, $\mathbf{T}_i, i = \{1, 2, 3\}$ of the tensor of feature responses.

locations of features with correlated activity. We propose to use these eigenimages to generate candidate receptive fields for pooling. Each eigenimage is first binarized, by subtracting its mean, taking the absolute value, and applying a threshold of magnitude equal to the eigenimage's standard deviation. In the remainder of this work, we will refer to the binarized eigenimage as simply the eigenimage $I(\mathbf{x})$. These eigenimages are the optimal pooling regions (under the energy compaction principle of PCA) for the recognition of the class of images used to produce the tensor. We next introduce a procedure to learn the rectangular receptive fields that best approximate them.

3. Receptive field clustering

The receptive fields used for pooling are modeled as soft-edged boxes. Starting from the sigmoid $f_\sigma(x) = (1 + e^{-\frac{x}{\sigma}})^{-1}$, we define a 1D receptive field of width $2a$

$$h_\sigma(x; a) = f_\sigma(x + a) - f_\sigma(x - a). \tag{4}$$

This is similar to a pedestal function of the same width, but has soft edges of smoothness controlled by σ. A receptive field of length $2a$ centered at location u is given by $h_\sigma(x - u; a)$. A 2D receptive field of size $2\mathbf{a} = 2(a_1, a_2)$, centered at image location $\mathbf{u}$, is finally given by $H_\sigma(\mathbf{x} - \mathbf{u}; \mathbf{a})$, with

$$H_\sigma(\mathbf{x}, \mathbf{a}) = h_\sigma(x_1; a_1) \times h_\sigma(x_2; a_2). \tag{5}$$

3.1. Single image

We start by considering the problem of fitting the receptive field above to an eigenimage $I(\mathbf{x})$. This is assumed to be a 2D binary function of amplitude one over some finite set $\mathcal{R}$. The goal is to determine the parameters $(\mathbf{a}^*, \mathbf{u}^*)$ of the receptive field of maximum normalized cross-correlation with $I(\mathbf{x})$

$$\begin{aligned}(\mathbf{a}^*, \mathbf{u}^*) &= \arg\max_{\mathbf{a},\mathbf{u}} \frac{\langle I(\mathbf{x}), H_\sigma(\mathbf{x}-\mathbf{u};\mathbf{a})\rangle}{||I(\mathbf{x})||\,||H_\sigma(\mathbf{x}-\mathbf{u};\mathbf{a})||} && (6)\\ &= \arg\max_{\mathbf{a},\mathbf{u}} \frac{\langle I(\mathbf{x}), H_\sigma(\mathbf{x}-\mathbf{u};\mathbf{a})\rangle}{||H_\sigma(\mathbf{x}-\mathbf{u};\mathbf{a})||} && (7)\end{aligned}$$

For this we note that, up to boundary artifacts,

$$||H_\sigma(\mathbf{x} - \mathbf{u}; \mathbf{a})||^2 = ||H_\sigma(\mathbf{x}; \mathbf{a})||^2 = \gamma(\mathbf{a}) \quad \forall \mathbf{u} \tag{8}$$

with

$$\gamma(\mathbf{a}) \approx 4a_1a_2. \tag{9}$$

It follows that

$$\begin{aligned}(\mathbf{a}^*, \mathbf{u}^*) = \arg\max_{\mathbf{a}}\{&\frac{1}{\sqrt{\gamma(\mathbf{a})}} \times \\ &\arg\max_{\mathbf{u}} \langle I(\mathbf{x}), H_\sigma(\mathbf{x} - \mathbf{u}; \mathbf{a})\rangle\}.\end{aligned} \tag{10}$$

This reflects a trade-off between two requirements: that the receptive field has 1) large dot-product with the eigenimage, and 2) small size. The solution can be computed efficiently by exploiting the symmetry of $H_\sigma(\mathbf{x}; \mathbf{a})$, since

$$\begin{aligned}\mathcal{C}(\mathbf{u}; \mathbf{a}) &= \langle I(\mathbf{x}), H_\sigma(\mathbf{x} - \mathbf{u}; \mathbf{a})\rangle \\ &= \sum_{x_1,x_2} I(x_1, x_2) H_\sigma(x_1 - u_1, x_2 - u_2; \mathbf{a}) \\ &= \sum_{x_1,x_2} I(x_1, x_2) H_\sigma(u_1 - x_1, u_2 - x_2; \mathbf{a}).\end{aligned}$$

It follows from the properties of the convolution that

$$\begin{aligned}\mathcal{F}\{\mathcal{C}(\mathbf{u}; \mathbf{a})\} &= \mathcal{F}(I(\mathbf{u})) \times \mathcal{F}(H_\sigma(\mathbf{u}; \mathbf{a})) && (11)\\ \mathcal{C}(\mathbf{u}; \mathbf{a}) &= \mathcal{F}^{-1}\{\mathcal{F}(I(\mathbf{u})) \times \mathcal{F}(H_\sigma(\mathbf{u}; \mathbf{a}))\} && (12)\end{aligned}$$

where $\mathcal{F}$ is the Fourier transform. This simplifies the computation of (10), which is equivalent to

$$(\mathbf{a}^*, \mathbf{u}^*) = \arg\max_{\mathbf{a}} \left\{ \frac{1}{\sqrt{\gamma(\mathbf{a})}} \arg\max_{\mathbf{u}} \mathcal{C}(\mathbf{u}; \mathbf{a}) \right\}. \tag{13}$$

For each receptive field size $\mathbf{a}$, it suffices to 1) multiply the Fourier transform of $I(\mathbf{x})$ by that of $H_\sigma(\mathbf{x}; \mathbf{a})$, 2) compute the inverse Fourier transform of the product, 3) find its peak location $\mathbf{u}^*$, and 4) divide $\sqrt{\gamma(\mathbf{a})}$. The complexity of this procedure is $O(s(\log k_1k_2)k_1k_2)$, where s is the number of receptive field sizes and k_i the image dimensions. This is significantly smaller than the complexity $O(s(k_1k_2)^2)$ of exhaustive search.

3.2. Multiple images

We next consider the search for a set of receptive fields $\{H_\sigma(\mathbf{x} - \mathbf{u}_i; \mathbf{a}_i)\}_{i=1}^m$ that best approximates a collection of eigenimages $\{I_i\}_{i=1}^n$. For each receptive field, the parameter pair $(\mathbf{a}_i, \mathbf{u}_i)$ defines a pooling region of size $\mathbf{a}_i$ and location $\mathbf{u}_i$. The optimal receptive fields are the solution of

$$\{(\mathbf{a}_i^*, \mathbf{u}_i^*)\}_{i=1}^m = \arg\max_{\mathbf{a}_i, \mathbf{u}_i} \sum_{i,n} \frac{\langle I_n(x), H_\sigma(\mathbf{x} - \mathbf{u}_i; \mathbf{a}_i)\rangle}{||I_n(x)|| || H_\sigma(\mathbf{x} - \mathbf{u}_i; \mathbf{a}_i)||}$$

This is a clustering problem with centroids $H_\sigma(\mathbf{x} - \mathbf{u}_i; \mathbf{a}_i)$. It can be solved by the *receptive field clustering* (RFC) algorithm. This is an algorithm that iterates between two steps, similar to those of k-means, as follows.

1. Given a set of receptive fields $(\mathbf{a}_i, \mathbf{u}_i), i = 1, \dots, m$ cluster eigenimages by finding, for each I_n,

   $$i_n^* = \arg\max_i \frac{\langle I_n(\mathbf{x}), H_\sigma(\mathbf{x} - \mathbf{u}_i; \mathbf{a}_i)\rangle}{||H_\sigma(\mathbf{x} - \mathbf{u}_i; \mathbf{a}_i)||}$$

   This has complexity $O(mnk_1k_2)$. Define clusters $C_i = \{I_n | i_n^* = i\}$.

2. Given eigenimage clusters C_i, determine $(\mathbf{a}_i, \mathbf{u}_i), i = 1, \dots, m$ with

   $$\begin{aligned} (\mathbf{a}_i^*, \mathbf{u}_i^*) &= \qquad\qquad\qquad\qquad (14) \\ &= \arg\max_{\mathbf{a},\mathbf{u}} \left\{ \sum_{n \in C_i} \frac{\langle I_n(\mathbf{x}), H_\sigma(\mathbf{x} - \mathbf{u}; \mathbf{a})\rangle}{||I_n(\mathbf{x})|| ||H_\sigma(\mathbf{x} - \mathbf{u}; \mathbf{a})||} \right\} \\ &= \arg\max_{\mathbf{a},\mathbf{u}} \left\{ \frac{\left\langle \sum_{n \in C_i} \frac{I_n(\mathbf{x})}{||I_n(\mathbf{x})||}, H_\sigma(\mathbf{x} - \mathbf{u}; \mathbf{a}) \right\rangle}{||H_\sigma(\mathbf{x} - \mathbf{u}; \mathbf{a})||} \right\} \\ &= \arg\max_{\mathbf{a},\mathbf{u}} \left\{ \frac{\langle S_i, H_\sigma(\mathbf{x} - \mathbf{u}; \mathbf{a})\rangle}{||H_\sigma(\mathbf{x} - \mathbf{u}; \mathbf{a})||} \right\} \end{aligned}$$

   where

   $$S_i = \frac{1}{|C_i|} \sum_{n \in C_i} \frac{I_n(\mathbf{x})}{||I_n(\mathbf{x})||} \qquad (15)$$

   is the average normalized eigenimage in the i^{th} cluster. This can be solved with the algorithm of the previous section, using S_i as I.

Similarly to k-means, RFC only guarantees a locally optimal solution, which depends on its initialization. In this work, we adopt the spatial pyramid as the set of initial pooling regions, using 3 levels of spatial partitioning frequently used in object recognition tasks, 1×1, 3×1 and 2×2 image sub-blocks. This results in 8 spatial bins. During clustering, we restrict the scale search to sizes $\mathbf{a}$ such that $\frac{1}{16} \leq \frac{a_1 a_2}{I_h I_w} \leq \frac{1}{2}$, where I_h, I_w is the image size. This avoids regions that are either too large or too small.

4. Experimental Evaluation

Several experiments were conducted to evaluate the performance of learned receptive fields.

4.1. Feature tensor

All experiments were based on feature tensors that follow standard practices in the recognition literature. Image patches were sampled over 8 scales, separated by a factor of 1.2, with step-size equal to half the patch size [15]. In most experiments, the images were converted to grayscale and a 128D SIFT descriptor computed per image patch. On PASCAL VOC 2007 we also report performance for color features, obtained as in [25]. In this case, each image patch was divided into 4×4 sub-regions, for which the mean and standard deviation of the RGB channels was computed, producing a 96D feature vector.

4.2. Coding tensor

We started with a series of experiments to evaluate the gains of learning pooling fields, using the Caltech-256 dataset [10]. This contains $30,607$ images from 256 object categories plus a background category, where each category contains at least 80 images. As is standard in the literature, we randomly selected n images from each category for training and the rest for testing. Results are reported for various values of n. Class-specific tensors were built from normalized images of 160×192 pixels. Two popular coding schemes were considered: vector quantization (VQ) and Fisher vector (FV). For VQ, codebooks of size $1,024$ were learned by k-means clustering of SIFT descriptors extracted from a random sample of $500,000$ image patches. For FV, feature dimensions were first reduced from 128 to 64, using PCA-SIFT [14]. A Gaussian mixture model (GMM) of 256 components was then learned with the EM algorithm. Vectors of Fisher scores were finally computed and subjected to L_2 and power normalization, as in [25]. In all experiments, pooled features were fed to a multi-class SVM. This used an intersection kernel for VQ and was a linear SVM for FV.

4.3. Gains of RFC

The first set of experiments compared the performance of SPM, the receptive fields learned with RFC, and the combination of the two (SPM+RFC). Both SPM and RFC used 8 pooling regions. The resulting feature encoding vectors were concatenated to produce the SPM+RFC features. This is an encoding of 16 regions. Figure 4(a) shows a set of the learned pooling fields, some of which are superimposed on Caltech images in Figure 4(b). The images shown in each row are from a common class. Note how the red windows are tunned for bear bodies, while the green windows specialize in glasses, and the blue windows in billiards tables. Each classification experiment was repeated for 5 trials. The average classification accuracies (when 30 images are used for training) are reported in Table 1. For both encodings, RFC performed better than SPM, achieving a gain of 1.5% for the VQ and 2.3% for the FV encoding. The SPM+RFC combination introduced an additional gain of 0.9%

Table 1: Comparison of SPM and RFC on Caltech-256. C is the number of spatial bins.

Encoding	Pooling	C	Acc.(%)
VQ [10]	SPM	8	34.1±0.2
VQ	RFC	8	35.6±0.5
VQ	SPM+RFC	16	36.7±0.2
FV [25]	SPM	8	40.8±0.1
FV	RFC	8	43.1 ±0.3
FV	SPM+RFC	16	43.7 ±0.3

Table 2: Classification accuracy of various state-of-the-art classifiers on Caltech-256. When available, the standard deviation is shown in ( ).

training images	15	30	45	60
VQ+SPM [10]	-	34.1(0.2)	-	-
ScSPM [34]	27.7	34.0	37.5	40.1
FV+SPM [25]	34.7(0.2)	40.8(0.1)	45.0(0.2)	47.9(0.4)
LLC+SPM [33]	34.4	41.2	45.3	47.7
CRBM [30]	35.1	42.1	45.7	47.9
FV+LRF	$\mathbf{35.6(0.2)}$	$\mathbf{43.7(0.3)}$	$\mathbf{48.3(0.1)}$	$\mathbf{51.4(0.2)}$

for VQ and 0.6% for FV. These results show that there are benefits to learning pooling regions. Given the simplicity of RFC, the gains in recognition accuracy (overall gain between 2.4% and 2.9%) can be considered quite significant. Finally, while both RFC and SPM capture information relevant for classification, the gains of RFC are larger for the more powerful FV encoding.

4.4. Comparison on various datasets

Since the combination of FV encoding, SPM+RFC pooling, and linear SVM achieved the best performance in the experiments above, we adopted this classifier in all remaining experiments. For simplicity, we refer to it as FV+LRF (FV with learned receptive fields). Several experiments were performed to compare its performance to state-of-the-art results on several popular benchmarks: Caltech-256 [10] and PASCAL-VOC2007 for object recognition, and MIT-Scenes [27] and 15-scenes [17] for scene classification.

Caltech-256 [10]: Table 2 presents the average classification accuracy (with standard deviation) over 5 classification trials on Caltech-256. Results are presented for training sets ranging from 15 to 60 images per class. Also presented are equivalent results for the VQ+SPM classifier of [10], the sparse coded SPM (ScSPM) of [34], the locality constrained linear coding (LLC) of [33], the FV+SPM of [25] and the convolutional Restricted Boltzmann machine (CRBM) of [30]. All these methods rely on either SPM or a similar strategy for pooling, focusing on alternative encodings. Note that these encodings can require the solution of sophisticated optimization problems during learning and have increased complexity during classification. Yet, the performance of the different methods is very similar to (or worse than) that of FV+SPM. On the other hand, the proposed FV+LRF has substantially better performance (around 3% for most training set sizes) and a marginal increase in complexity. These results suggest that, while much attention has been devoted to encodings, more emphasis should be given to the learning of pooling operators.

PASCAL-VOC 2007 [7]: In this dataset of 20 categories, 5,011 training, and 4,952 test images, classification accuracy is usually evaluated by average precision (AP). Class-specific tensors were built from normalized images of 160 × 200 pixels. Table 3 reports the AP of the 20 classes. The methods above the line use only SIFT features, those below use both SIFT and color features. For color, we reduced the feature dimension from 96D to 64D and learned a GMM and SVM separately from the SIFT features. The final score was the average of the two scores (SIFT and color). The grayscale methods include VQ+SPM [28], object-centered pooling (OCP) [28], combinations of both VQ and FV with the spatial Fisher vector (SFV) of [15], the winner of the VOC [7], and the proposed FV+LRF. The latter achieves the highest mAP of all grayscale methods. This is probably the most telling experiment that we report, since all competing methods somehow attempt to overcome the limitations of SPM. SFV is a Fisher vector computed from a GMM fitted to the spatial coordinates of SIFT vectors, OCP a method for learning receptive fields for pooling, and the VOC winner a method that combines object recognition and localization using a fairly complex classifier. The color methods include the late-fusion method of [18], FV+SPM with color [25], and the application of the FV+LRF to both SIFT and color features. The latter again achieves superior performance.

MIT-Scenes [27]: This dataset contains 15,620 images from 67 indoor scene categories. Following [27], we used 80 images per class for training and 20 images for testing. Class-specific tensors were built from normalized images of 160 × 194 pixels. Classification accuracy is reported on the training/testing split provided on the author's web page. Table 4 compares FV+LRF to VQ+SPM [22], the deformable parts models (DPM) of [9], the reconfigurable models (R-BoW) of [23], the combination of SPM, DPM, and color GIST of [22] (denoted MF for multi-features), the SPM on the semantic manifold (SPM-SM) classifier of [16], and the hierarchical matching pursuit (HMP) of [2]. On this dataset, FV+LRF achieves substantial higher accuracy than all competing methods. To the best of our knowledge, it has the best results published on this dataset.

15 Scenes [8]: The 15-Scenes dataset contains 15 scene categories, with 200 to 400 images per class, for a total $4,485$ images of average size around 300 × 250. As suggested in [17], 100 images per class were randomly selected for training. Class-specific tensors were built from normalized

Table 3: Image classification results (mAP) on PASCAL VOC 2007 dataset

	aero	bicyc	bird	boat	bottle	bus	car	cat	chair	cow	
VQ + SPM [28]	72.5	56.3	49.5	63.5	22.4	60.1	76.4	57.5	51.9	42.2	
VQ + OCP [28]	74.2	63.1	45.1	65.9	29.5	64.7	79.2	61.4	51.0	45.0	
FV + SPM [25]	75.7	64.8	52.8	70.6	30.0	64.1	77.5	55.5	55.6	41.8	
Winner [7]	77.5	63.6	56.1	71.9	**33.1**	60.6	**78.0**	58.8	53.5	42.6	
FV + LRF	77.4	63.8	51.7	70.5	28.5	66.5	77.9	59.3	53.5	44.9	
Late-fusion [18]	**78.0**	**64.9**	**58.0**	**73.1**	32.2	64.0	76.4	**62.4**	**57.3**	44.6	
FV + LRF + color	**78.0**	63.6	57.8	70.5	29.3	**67.7**	77.6	60.2	54.3	**46.8**	
	table	dog	horse	moto	person	plant	sheep	sofa	train	tv	avg
VQ + SFV [15]	-	-	-	-	-	-	-	-	-	-	52.9
FV + SFV [15]	-	-	-	-	-	-	-	-	-	-	56.6
VQ + SPM [28]	48.8	38.1	75.1	62.8	82.9	20.5	38.1	46.0	71.7	50.5	54.3
VQ + OCP [28]	54.8	45.4	76.3	67.1	84.4	21.8	44.3	48.8	70.7	51.7	57.2
FV + SPM [25]	56.3	41.7	76.3	64.4	82.7	28.3	39.7	56.6	79.7	51.5	58.3
Winner [7]	54.9	45.8	77.5	64.0	85.9	36.3	44.7	50.9	79.2	53.2	59.4
FV + LRF	55.2	45.6	78.0	68.0	83.6	32.4	49.9	56.4	78.4	52.9	59.7
Late-fusion [18]	56.7	**51.0**	**80.4**	65.9	**87.5**	**46.5**	46.3	49.7	**82.9**	55.3	61.7
FV + SPM + color [25]	-	-	-	-	-	-	-	-	-	-	60.3
FV + LRF + color	**62.6**	49.1	**80.4**	**69.4**	86.3	40.3	51.2	**57.3**	79.6	**55.7**	**61.9**

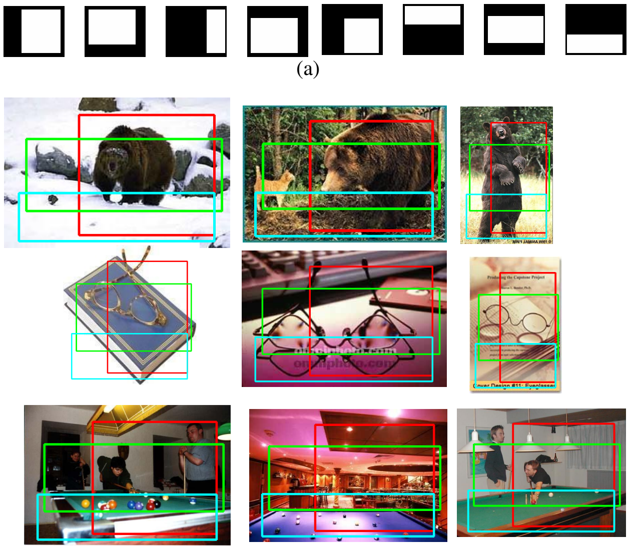
images of 300×200 pixels. Table 5 presents averaged classification accuracies (± standard deviation) over 10 trials. The performance of the proposed classifier is compared to VQ+SPM [17], ScSPM [34], SPM-SM [16], the macrofeatures (Macro) of [4], and the model adaptation (MA+SPM) method of [6]. The proposed FV+LRF has competitive or superior performance to all other methods. Only MA+SPM has slightly superior performance. This is likely due to the fact that it uses model adaptation to obtain better probability estimates [6]. Model adaptation is substantially more complex than the proposed receptive field learning method and could also be used to obtain better Fisher scores. We have not attempted to do so yet.

4.5. Comparison to RF learning methods

The literature on the learning of pooling operators is fairly small. Jia *et al*. [12] proposed a method akin to feature selection, which selects optimal receptive fields within a large pool of pre-defined windows. This involves a computationally intensive optimization, which is simplified with recourse to a greedy approximation. Nevertheless, the computation is still substantial, orders of magnitude larger than that of the proposed RFC. Perhaps due to this, results are only available for datasets of much smaller scale and variability than those used above, e.g. CIFAR-10, MNIST, or Caltech-101. We have not considered these datasets in our evaluation. On Caltech-101, this method achieved results in between those of ScSPM [34] and Macrofeatures [3], which FV+LFR outperformed in the datasets above. Russakovsky *et al*. [28] proposed an object-centric RF learning algorithm. This is an iterative process, where each iteration involves bootstrapping thousands of image regions and learning many high-dimensional SVM classifiers. Its complexity is again large, and this method was only tested on PASCAL VOC 2007. As can be seen from Table 3, its results are inferior to the much simpler FV+LRF classifier now proposed. Finally, Krapac *et al.* [15] proposed the SFV. While this is not strictly a method to learn pooling fields, it aims for a similar goal: to characterize the spatial statistics of SIFT descriptors. Again, Table 3 shows that this spatial encoding has much weaker performance than the proposed learning of receptive fields.

5. Conclusion

In this work, we proposed a simple, yet effective, method for learning receptive fields for pooling. The method exploits the statistics of the 3D tensor of SIFT responses to derive class-specific receptive fields. This requires a simple PCA of a flattening of the tensor and is akin, but complementary, to PCA-SIFT. The resulting eigenimages are informative of the spatial locations of discriminative visual structures of different image classes. We then proposed a k-means like algorithm for fitting box-like receptive fields to the eigenimages extracted from a collection of images. The resulting receptive fields were shown to consistently improve the performance of both VQ and FV encodings. Experiments on a collection of object recognition and scene classification datasets show that their combination with the FV encoding has state-of-the-art performance for both tasks. When compared to previous attempts to learn receptive fields for pooling, the method is simpler and has better results. At a more general level, this work shows that 1) more emphasis should be placed on the pooling operation, and 2) there are benefits to analyzing the complete 3D tensor of feature responses, rather than the 2D slice that corresponds to the popular bag of words image representation. We plan to explore these issues in the future.



(a)

(b)

Figure 4: Learned pooling fields on Caltech 256. (a) a set of the receptive fields learned by RFC. (b) Three pooling windows superimposed on images from three categories (one category per row). The red, green, and blue windows capture regions of discriminant information for the recognition of bears, glasses, and billiards.

Table 4: Classification accuracy on MIT-Scenes

Method	Acc.(%)
DPM [9, 22]	30.4
VQ+SPM [22]	34.4
RBoW [23]	37.9
MF [22]	43.1
SPM-SM [16]	44.0
HMP [2]	47.6
FV+LRF	**60.3**

Table 5: Classification Accuracy on 15-Scenes

Algorithms	Accuracy
VQ+SPM [17]	81.4±0.5
ScSPM [34]	80.4±0.45
SPM-SM [16]	82.3
Macro [4]	84.3±0.5
MA+SPM [6]	**85.4**
FV+LRF	85.0 ±0.6

Acknowledgements. This work was partially funded by NSF awards IIS-1208522 and CCF-0830535.

References

[1] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *ECCV*, 2006. 1

[2] L. Bo, X. Ren, and D. Fox. Unsupervised feature learning for rgb-d based object recognition. In *ISER*, 2012. 6, 8

[3] Y. Boureau and J. Ponce. A theoretical analysis of feature pooling in visual recognition. In *ICML*, 2010. 1, 2, 7

[4] Y.-L. Boureau, Y. L. Francis Bach, and J. Ponce. Learning mid-level features for recognition. In *CVPR*, 2010. 6, 8

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1

[6] M. Dixit, N. Rasiwasia, and N. Vasconcelos. Adapted gaussian models for image classifications. In *CVPR*, 2011. 6, 8

[7] M. Everingham, L. Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2007 (voc2007) results, 2007. 6, 7

[8] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005. 6

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence*, 32:1627 – 1645, 2010. 6, 8

[10] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. california institute of technology, 2007. Supplied as additional material `tr.pdf`. 5, 6

[11] S. Han and N. Vasconcelos. Biologically plausible saliency mechanisms improve feedforward object recognition. *Vision Research*, 50(22):2295–2307, 2010. 1, 2

[12] Y. Jia, C. Huang, and T. Darrell. Beyond spatial pyramids: Receptive field learning for pooled image features. In *CVPR*, 2012. 1, 2, 7

[13] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *CVPR*, 2011. 1

[14] Y. Ke and R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In *CVPR*, 2004. 2, 3, 5

[15] J. Krapac, J. Verbeek, and F. Jurie. Modeling spatial layout with fisher vectors for image categorization. In *ICCV*, 2011. 3, 5, 6, 7

[16] R. Kwitt, N. Vasconcelos, and N. Rasiwasia. Scene recognition on the semantic manifold. In *ECCV*, 2012. 6, 8

[17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1, 6, 8

[18] D. Liu, K.-T. Lai, G. Ye, M.-S. Chen, and S.-F. Chang. Sample-specific late fusion for visual category recognition. In *CVPR*, 2013. 6, 7

[19] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H. Shum. Learning to detect a salient object. In *CVPR*, 2007. 2

[20] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1, 3

[21] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29:51–59, 1996. 1

[22] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011. 6, 8

[23] S. Parizi, J. Oberlin, and P. Felzenszwalb. Reconfigurable models for scene recognition. In *CVPR*, 2012. 6, 8

[24] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007. 1

[25] F. Perronnin, J. Sanchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *CVPR*, 2010. 1, 5, 6, 7

[26] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 1

[27] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009. 6

[28] O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei. Object-centric spatial pooling for image classification. In *ECCV*, 2012. 1, 2, 3, 6, 7

[29] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *Vision Research*, 29(3):411–426, 2007. 1

[30] K. Sohn, D. Yon, J.-H. Lee, and A. Hero. Efficient learning of sparse, distributed, convolutional feature representations for object recognition. In *ICCV*, 2011. 6

[31] M. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *ECCV*, 2002. 2

[32] M. Vasilescu and D. Terzopoulos. Multilinear subspace analysis of image ensembles. In *CVPR*, volume 2, pages 93–99, 2003. 2

[33] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality constrained linear coding for image classification. In *CVPR*, 2010. 1, 6

[34] J. Yang, K. Yu, and Y. Gong. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009. 1, 2, 6, 7, 8

[35] X. Zhou, K. Yu, T. Zhang, and T. S. Huang. Image classification using super-vector coding of local image descriptors. In *ECCV*, 2010. 1