ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Learning Separable Filters[*]

Roberto Rigamonti (roberto.rigamonti@epfl.ch)
http://cvlab.epfl.ch/~rigamont

Vincent Lepetit (vincent.lepetit@epfl.ch)
http://cvlab.epfl.ch/~lepetit

Pascal Fua (pascal.fua@epfl.ch)
http://cvlab.epfl.ch/~fua

School of Computer and Communication Sciences
Swiss Federal Institute of Technology, Lausanne (EPFL)

[*] This work has been supported in part by the Swiss National Science Foundation.

Technical Report

June 25, 2012

Abstract. While learned image features can achieve great accuracy on different Computer Vision problems, their use in real-world situations is still very limited as their extraction is typically time-consuming. We therefore propose a method to learn image features that can be extracted very efficiently using separable filters, by looking for low rank filters. We evaluate our approach on both the image categorization and the pixel classification tasks and show that we obtain similar accuracy as state-of-the-art methods, at a fraction of the computational cost.

1 Introduction

It has been shown that representing images as sparse linear combinations of learned filters [1] yields effective approaches to image denoising and object recognition, which outperform those that rely on hand-crafted features [2, 3]. Among these, convolutional formulations have emerged as particularly appropriate to handle whole images, as opposed to independent patches [4–6]. Unfortunately, because the filters are both numerous and not separable, their implementations tend to be computationally demanding, which has hindered their application to real-world situations.

In this paper, we show that we can match the accuracy of these convolutional approaches using *separable* filters only. Separable filters are desirable because they can be computed very efficiently: Handcrafted filter banks are often made separable [7], and we aim here at the same efficiency with the advantages of learning approaches. To this end we investigate two separate schemes. The first involves directly learning a bank of separable filters by enforcing the separability constraint as part of a convolutional, ℓ_1-based learning framework. The second starts from regular filters and approximates them by linear combinations of a small set of separable ones. In practice, the latter approach usually yields better accuracy by decomposing a very hard optimization problem into two simpler ones that are easier to control.

Both schemes are based on the minimization of the nuclear norm of the filters, which is a convex relaxation of the rank. The second scheme is general, as it can be applied to any filter bank. The first one can easily be adapted to other convolutional frameworks, such as the Deconvolutional Networks of [8], by adding the nuclear norm penalty term and modifying the gradient computation accordingly.

In the remainder of the paper, we first discuss related work. We then introduce our two approaches, and evaluate them on different Computer Vision problems –object recognition, pixel classification, and image denoising– to show they significantly speed up processing at no loss of accuracy.

2 Related work

Automatic feature learning has long been an important area in Machine Learning and Computer Vision. Neural networks [9], Restricted Boltzmann Machines [10], Auto-Encoders [11], Linear Discriminant Analysis [12], and many other techniques have been used to learn features either in supervised or unsupervised ways. Recently, creating overcomplete dictionary of features—sparse combinations of which can be used to represent images—has emerged as a powerful tool for object recognition [2, 13] and image denoising [14, 3], among others.

However, for most such approaches, run-time feature extraction can be very time-consuming because it involves convolving the image with non-separable non-sparse filters. It was proposed many years ago to split convolution operations into convergent sums of matrix-valued stages [15]. This principle was exploited in [16] to avoid a coarse discretization of the scale and orientation spaces, yielding steerable separable 2D edge-detection kernels. This approach is powerful but restricted to kernels that are decomposable in the suggested manner, which does preclude the potentially arbitrary ones that can be found in a learned dictionary.

Most recent feature-learning publications have focused on improving the filter learning schemes and very few have revisited the run-time efficiency issue. Among those, the majority advocates exploiting the parallel capabilities of modern hardware [17–19]. To the best of our knowledge, the only one that considers run-time efficiency from a computational complexity standpoint is the approach of [20], which involves learning a filter bank by composing a few atoms from a handcrafted separable dictionary. Our own approach is in the same spirit but is more general because the basis from which the filters are to be built is not restricted *a priori*. As a result, we can use a small number of separable filters that end up being tuned for the task at hand.

3 Learning Separable Filters

Most dictionary learning algorithms operate on image patches [1, 21, 13] but convolutional approaches [4, 8, 6, 5, 22] have been recently introduced as a more natural way to process arbitrarily-sized images. They generalize the concept of *feature vector* to that of *feature map*, a term borrowed from the Convolutional Neural Network literature [23]. In our work, we consider the convolutional extension of Olshausen and Field's functional proposed in [22]. Formally, N filters $\{\mathbf{f}^j\}_{1 \leq j \leq N}$ are computed as

$$\underset{\{\mathbf{f}^j\},\{\mathbf{m}_i^j\}}{\operatorname{argmin}} \sum_i \left(\left\| \mathbf{x}_i - \sum_{j=1}^N \mathbf{f}^j * \mathbf{m}_i^j \right\|_2^2 + \lambda_1 \sum_{j=1}^N \left\| \mathbf{m}_i^j \right\|_1 \right) , \tag{1}$$

where

- $\mathbf{x}_i$ is an input image;
- $*$ denotes the convolution product operator;
- $\{\mathbf{m}_i^j\}_{j=1...N}$ is the set of extracted feature maps during learning;

– λ_1 is a regularization parameter.

While this formulation achieves state-of-the-art results [5], the required run-time convolutions are costly because the filters are not separable. Quantitatively, if $\mathbf{x}_i \in \mathbb{R}^{\mathbf{p}\times\mathbf{q}}$ and $\mathbf{f}_i^j \in \mathbb{R}^{\mathbf{s}\times\mathbf{t}}$, extracting the feature maps requires $\mathcal{O}\left(\mathbf{p}\cdot\mathbf{q}\cdot\mathbf{s}\cdot\mathbf{t}\right)$ multiplications and additions. By contrast, if the filters were separable, the computational cost would drop to a more manageable $\mathcal{O}\left(\mathbf{p}\cdot\mathbf{q}\cdot(\mathbf{s}+\mathbf{t})\right)$. Incidentally, this discussion also applies to patch-based approaches and their high computational cost when dealing with big images by extracting patches, usually on a regular grid with a small stride, and computing the dot products between these patches and learned vectors.

Our goal therefore is to look for separable filters without compromising the descriptive power of dictionary-learning approaches. An approach to doing this would be to explicitly write the $\mathbf{f}^j$ filters as products of 1D filters and to minimize the criterion of Eq. (1) in terms of their coefficients. Unfortunately, this would result in a quadratic criterion in terms of the unknown and therefore a very difficult optimization problem.

In the remainder of this section, we propose two different approaches to overcoming this problem. The first involves directly forcing the filters to be separable by lowering their rank. The second looks for filters that can all be written as linear combinations of a small number of separable ones. While the first strategy is more straightforward, we will show that the second one yields better results.

3.1 Penalizing High-Rank Filters

Our first approach to finding low-rank filters is to add a penalty term to the criterion of Eq. (1) and to solve

$$\underset{\{\mathbf{f}^j\},\{\mathbf{m}_i^j\}}{\operatorname{argmin}} \sum_i \left(\left\| \mathbf{x}_i - \sum_{j=1}^{N} \mathbf{f}^j * \mathbf{m}_i^j \right\|_2^2 + \lambda_1 \sum_{j=1}^{N} \left\| \mathbf{m}_i^j \right\|_1 + \lambda_* \sum_{j=1}^{N} \left\| \mathbf{f}^j \right\|_* \right), \tag{2}$$

where $\|\cdot\|_*$ is the nuclear norm and λ_* is an additional regularization term. The nuclear norm of a matrix is the sum of its singular values. It is a convex relaxation of the rank [24], thus forcing the nuclear norm to be small amounts to lowering the rank of the filters. Experimentally, for sufficiently high values of λ_*, the filters become effectively rank 1 and they can be written as products of 1D filters.

A standard way to solve Eq. (1) is to alternatively optimize over the $\mathbf{m}_i^j$ representations and the $\mathbf{f}^j$ filters. Stochastic Gradient Descent is used for the latter, while the former is achieved by first taking a step in the direction opposite to the ℓ_2-penalized term gradient and then applying the proximal operator of the ℓ_1 penalty term, which is the soft-thresholding operation, on the $\mathbf{m}_i^j$ [25].

To solve Eq. (2), which has the nuclear norm of the filters as additional term, we apply the proximal operator of the nuclear norm to the filters, in addition to the steps used to solve Eq. (1). This amounts to performing a Singular Value

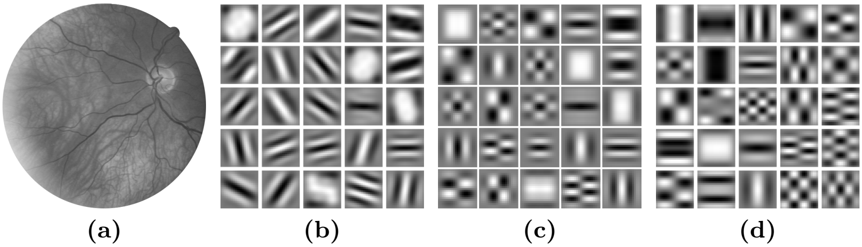

Fig. 1. Examples of non-separable and separable filter banks, learned on the DRIVE dataset [26]. **(a)** One of the training images. **(b)** Non-separable filter bank learned by optimizing Eq. (1). **(c)** Its rank-1 approximation via SVD. **(d)**: The separable filter bank learned by optimizing Eq. (2).

Decomposition (SVD) $\mathbf{f} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ on each filter $\mathbf{f}$, soft-thresholding the values of the diagonal matrix $\mathbf{D}$ to obtain a new matrix $\widehat{\mathbf{D}}$, and replace $\mathbf{f}$ by $\mathbf{U}\widehat{\mathbf{D}}\mathbf{V}^\top$. At convergence, to make sure we obtain separable filters, we apply a similar SVD-based operation but set to 0 all the singular values but the largest one. In practice however, when this strategy is successful, the second largest singular value is already almost zero even before clipping. Fig. 1(d) depicts a filter bank learned in this manner.

Choosing appropriate values for the optimization's parameters, the gradient step size, λ_1, and λ_*, is challenging because they express contrasting needs. We have found it effective to start with a low value of λ_*, solve the system, and then progressively increase it until the filter ranks are close to one.

3.2 Linear Combinations of Separable Filters

The approach described above assumes that an image can be reconstructed by convolving each feature map with a separable filter. This is a strong assumption, and we also considered an approach that relies on a weaker one, while still requiring convolutions with separable filters only at run-time. As we will discuss, this approach is also more general as it can be applied to an existing filter bank, not necessarily one that is directly learned by the method.

In this second approach, we write the N $\mathbf{f}^j$ filters of Eq. (1) as linear combinations of M separable filters $\{\mathbf{s}_k\}_{1 \leq k \leq M}$. In other words, we seed a set of w_k^j of linear weights such that $\forall j, \mathbf{f}^j = \sum_{k=1}^{M} w_k^j \mathbf{s}_k$ and convolving the image with all the $\mathbf{f}^j$ amounts to convolving it with the separable $\mathbf{s}_k$ and then linearly combining the results, without further convolutions.

This could be achieved by solving

$$\underset{\substack{\{\mathbf{m}_i^j\} \\ \{\mathbf{s}_k\},\{w_k^j\}}}{\operatorname{argmin}} \sum_i \left(\left\| \mathbf{x}_i - \sum_{j=1}^{N} \left(\sum_{k=1}^{M} w_k^j \mathbf{s}_k \right) * \mathbf{m}_i^j \right\|_2^2 + \Gamma \left(\{\mathbf{m}_i^j\}, \{\mathbf{s}_k\} \right) \right) , \qquad (3)$$

with

$$\Gamma\left(\{\mathbf{m}_i^j\},\{\mathbf{s}_k\}\right) = \lambda_1 \sum_{j=1}^{N} \left\|\mathbf{m}_i^j\right\|_1 + \lambda_* \sum_{k=1}^{M} \|\mathbf{s}_k\|_* \;. \tag{4}$$

Again we introduce the nuclear norm to force the $\mathbf{s}_k$ filters to be separable. Unfortunately, this functional is difficult to optimize as the first term contains products of three unknowns.

A standard way to handle this difficulty is to introduce auxiliary unknowns, such as the non-separable filter coefficients. We could therefore solve

$$\underset{\substack{\{\mathbf{f}^j\},\{\mathbf{m}_i^j\}\\ \{\mathbf{s}_k\},\{w_k^j\}}}{\operatorname{argmin}} \sum_i \left(\left\| \mathbf{x}_i - \sum_{j=1}^{N} \mathbf{f}^j * \mathbf{m}_i^j \right\|_2^2 + \lambda_s \sum_{j=1}^{N} \left\| \mathbf{f}^j - \sum_{k=1}^{M} w_k^j \mathbf{s}_k \right\|_2^2 + \Gamma\left(\{\mathbf{m}_i^j\},\{\mathbf{s}_k\}\right) \right) , \tag{5}$$

where the second term forces the non-separable filters and the linear combinations of separable filters to be close to each other and λ_s controls the quality of the approximation. This makes the formulation linear, but at the cost of introducing an additional parameter that has proved very hard to tune.

Instead, we tried a simpler approach, which has yielded better results by decoupling the computation of the non-separable filters from that of the separable ones. We first learn a set of non-separable filters $\{\mathbf{f}^j\}$ by optimizing the original functional of Eq. (1). We then look for separable filters such that linear combinations of them approximate the $\mathbf{f}^j$ filters by solving

$$\underset{\{\mathbf{s}_k\},\{w_k^j\}}{\operatorname{argmin}} \sum_j \left\| \mathbf{f}^j - \sum_{k=1}^{M} w_k^j \mathbf{s}_k \right\|_2^2 + \lambda_* \sum_{k=1}^{M} \|\mathbf{s}_k\|_* \;. \tag{6}$$

Even though this may seem suboptimal when compared to the global optimization scheme of Eq. (3), it gives superior results in practice: The optimization process is split into two clear tasks and depends on two parameters instead of four, which eliminates the need for a careful scheduling of λ_s.

At run-time, we just have to linearly combine the feature maps extracted by the separable filters to have an approximation of the feature maps which would have been obtained by the full-rank filter bank. In the next section, we will show that a surprisingly small number of separable filters is required to approximate all the original non-separable ones.

4 Results and Discussion

In this section, we first evaluate our approach on two very different problems, object category recognition and pixel classification for segmentation of medical images. We then use the case of image denoising to demonstrate that our technique can also be used to approximate existing filter banks.

From now on, we will refer to the non-separable filter bank obtained by optimizing Eq. (1) as *NON-SEP*, to the separable filter bank learned using the technique of Section 3.1 as *SEP-DIRECT*, and to the one learned using the technique of Section 3.2 as *SEP-COMB*. Moreover, we will denote by *SEP-SVD* the separable filter bank obtained by approximating each filter from a full-rank filter bank with the outer product of its first left singular vector with its first right singular vector, which is the simplest way to approximate a non-separable filter by a separable one.

4.1 Object Category Recognition

A successful trend in object category recognition has seen the use of modular architectures, where the choice of each component and its parameters are tuned to improve the final performance [27–29, 13]. To test our approach in this context, we have adopted the shallow modular architecture of [22] for its simplicity, as we want the changes we make in the feature extraction step to be directly reflected on the final recognition rate, with as little interference as possible from the other steps in the pipeline. We consider here the CIFAR-10 dataset [30, 31]. It is composed of 32×32 pixels images of ten different objects, yet it exhibits a large variability in pose, appearance, scale, and background composition, making it a challenging test framework.

We operate on grayscale images and set up a simple classification pipeline following the guidelines of [22]: For a given input image, we first extract feature maps by convolving the image with the filter bank we want to evaluate, apply a simple non-linear operation to the feature maps and a pooling operation. We then feed the results to a multiclass SVM classifier with RBF kernels. As the performances of the SVM classifier may depend on several parameters—the penalty factor, ϵ, and the kernel parameters—we also gauged the Nearest Neighbor performance to provide another measure of the discriminative capabilities of the feature maps.

We first learned two filter banks composed by 11×11 non-separable filters, one with cardinality 25, and the other with cardinality 49, by solving the optimization problem in Eq. (1). We approximated these two sets of filters by simple SVD (*SEP-SVD*) and by applying our second strategy and solving Eq. (6) (*SEP-COMB*). When the original filter bank is composed by 49 filters, an approximating separable filter bank with cardinality 16 can already give a good approximation. However, when the original filter bank has just 25 filters, their structure is harder to approximate, and we have to resort to 25 separable filters to get good results. For this reasons, we experimented with 16 and 25 separable filters in the former case, 16, 25, and 30 in the latter.

Despite a methodical search of the parameter space, we have not been able to make *SEP-DIRECT* produce satisfying results on this dataset: The filters learned with this approach tend to stay full-rank, with only the last two or three singular values reaching small values. This probably means that the separable filters that can be reached by minimizing Eq. (2) only through over a very small range for the parameters, or they are too restrictive to learn the local structure of the images in the CIFAR dataset. Interestingly, it will not be the case for our second test-case presented in the next section, where we consider the extraction of linear structures in medical images.

Table 1. Category recognition results on the CIFAR-10 dataset. The superior performance of the *SEP-COMB* approach can be ascribed to the regularization effect implicit in the approximation process.

Filter bank	**Classification rate**		**Filter bank**	**Classification rate**	
	SVM	**NN**		**SVM**	**NN**
NON-SEP(25)	74.40%	46.96%	*NON-SEP*(49)	**74.62%**	**46.14%**
SEP-SVD(25)	71.29%	44.76%	*SEP-SVD*(49)	69.26%	42.10%
SEP-COMB(16)	74.37%	46.07%	*SEP-COMB*(16)	71.28%	43.05%
SEP-COMB(25)	74.84%	47.30%	*SEP-COMB*(25)	74.02%	45.75%
SEP-COMB(30)	**76.00%**	**47.36%**			

The performances of these different filter banks are reported in Table 1. Nearest-Neighbor always perform worse than SVMs, but the performance trends are similar in both cases. *SEP-COMB* is clearly the best approach. It even outperforms the original filter bank in the 25-filter case, probably because of a regularization effect implicit in the approximation.

Lastly, since the sought for correlations in the data captured by the learned filters are complex, their separable approximation given by *SEP-SVD* scored poorly, despite the presence of a greater number of filters.

4.2 Pixel classification

The medical domain is a particularly promising application field for automated Computer Vision techniques, as it produces huge amounts of images with ever increasing dimensionality, while imposing strict requirements on the quality and the efficiency of the processing techniques. We chose to demonstrate our idea on the segmentation of tubular structures, a long-standing Computer Vision problem, because of its practical importance.

Over the course of the years, models of increasing complexity and effectiveness have been proposed, and the attention has recently turned to Machine Learning techniques. [32, 33] apply a Support Vector Machine classifier to the responses of *ad hoc* filters. [32] considers the Hessian's eigenvalues, while the Rotational Features of [33] use steerable filters. A dictionary learning method was used in [5] to learn a set of linear filters on images of linear structures instead of hard-coding them. It was further shown that convolving images with this filter bank gives responses that, when fed to an SVM, outperform state-of-the-art methods. Unfortunately, it requires a large number of non-separable filters, making it an impractical approach for large images. We demonstrate that our approach resolves this issue.

Please note that, although our filter learning algorithm is unsupervised, we still need ground truth segmentations delineated by human experts to train the classifiers. The availability of these segmentations played therefore a key role in the choice of the datasets. We have considered three different 2D medical

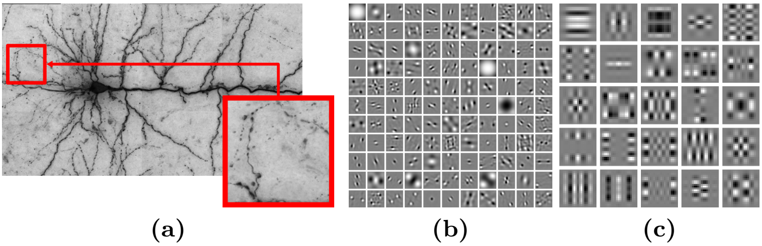

Fig. 2. Learning filter banks for the extraction of linear structures. **(a)** Test image from the BF2D dataset. In the red box, a detail showing the visual appearance of the sought structures. **(b)** Filter bank, with 121 filters of size 21 × 21, learned by optimizing Eq. (1). **(c)** Filter bank composed by 25 separable filters learned by optimizing Eq. (6).

Table 2. Analytic measures of the quality of the pixel classification task over the different datasets. The VI and the RI values are computed on the classification thresholded at the value found using the F-measure. VI assumes values in $[\,0, \infty)$, the lower the better, and RI assumes values in $[0, 1]$, the higher the better. The values are averages over 5 random trials and over the whole dataset images. The number after the ± sign is the standard deviation. For the learning-based approaches, a training set of 50,000 positive and 50,000 negative samples and a Random Forests classifier have been used, except for the SVM case where the number of samples was limited to 2,500 as in [5].

Method	**AUC**	**F-measure**	**VI**	**RI**
DRIVE				
Ground truth		0.788	0.380	0.930
NON-SEP(121)	**0.959** ±0.010	0.782 ±0.028	0.554 ±0.084	0.890 ±0.023
NON-SEP(121)-SVM	0.944 ±0.014	0.764 ±0.033	0.590 ±0.039	0.800 ±0.017
NON-SEP(49)	0.956 ±0.011	0.773 ±0.024	0.564 ±0.065	0.887 ±0.017
SEP-SVD(121)	0.955 ±0.012	0.773 ±0.030	0.563 ±0.092	0.887 ±0.026
SEP-SVD(49)	0.954 ±0.011	0.763 ±0.024	0.583 ±0.667	0.883 ±0.017
SEP-DIRECT(49)	0.952 ±0.011	0.762 ±0.024	0.577 ±0.066	0.883 ±0.017
SEP-COMB(25)	**0.959** ±0.011	**0.785** ±0.029	**0.541** ±0.069	**0.894** ±0.017
BF2D				
NON-SEP(121)	**0.983** ±0.000	**0.754** ±0.001	**0.300** ±0.001	**0.945** ±0.000
SEP-SVD(121)	0.982 ±0.000	0.749 ±0.005	0.306 ±0.004	0.943 ±0.001
SEP-DIRECT(25)	0.980 ±0.000	0.750 ±0.001	0.306 ±0.002	0.944 ±0.001
SEP-COMB(25)	0.981 ±0.000	0.752 ±0.002	0.301 ±0.002	0.944 ±0.001
STARE				
Ground truth		0.740	0.424	0.909
NON-SEP(121)	**0.968** ±0.014	**0.769** ±0.049	**0.537** ±0.166	**0.885** ±0.060
SEP-SVD(121)	0.965 ±0.015	0.760 ±0.056	0.548 ±0.159	0.882 ±0.055
SEP-DIRECT(25)	0.963 ±0.013	0.743 ±0.047	0.580 ±0.165	0.873 ±0.062
SEP-COMB(25)	0.966 ±0.015	0.767 ±0.052	0.539 ±0.165	**0.885** ±0.057

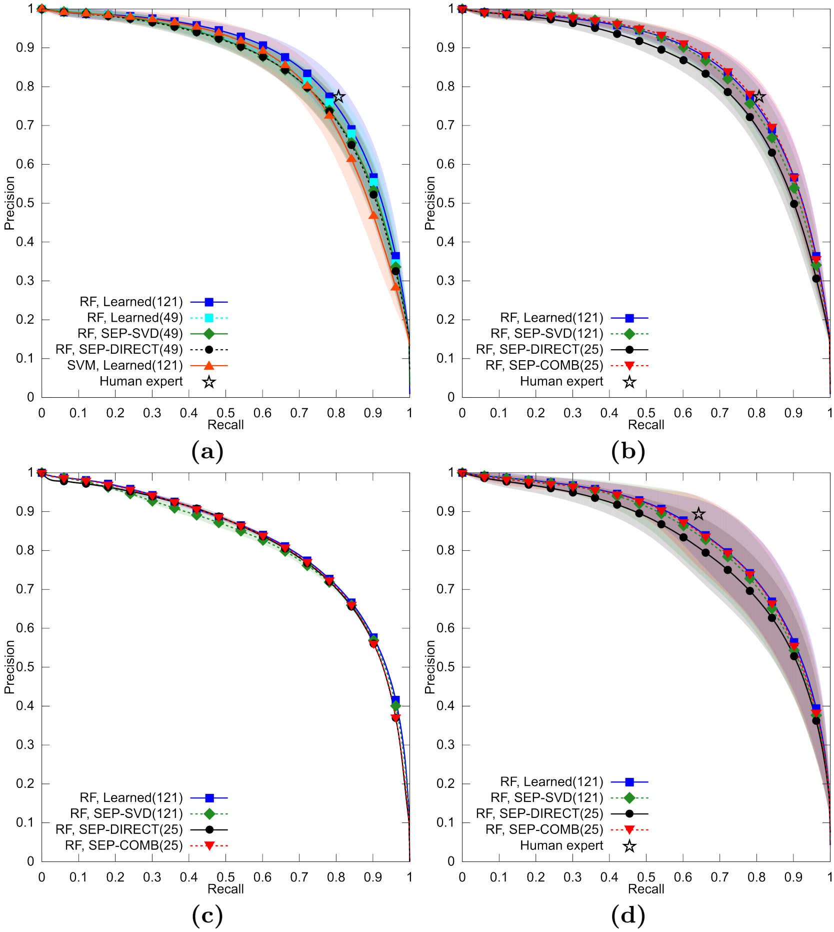

Fig. 3. Precision/Recall curves computed for the pixel classification task over the different datasets. Where available, the ground truth provided by the second human expert was used to define a target score for the algorithms. The shades represent 1 standard deviation around the mean value. *This figure is best viewed in colors.* **(a)** DRIVE dataset, 49 separable filters: The performance of *SEP-SVD*(49) is identical to that of *SEP-DIRECT*(49), and both approaches perform slightly worse than the full-rank filter bank of corresponding size. Random Forests are not only faster than SVMs, but also score better. **(b)** DRIVE dataset, 25 separable filters: Reconstructing the feature maps of the full-rank filter bank (*SEP-COMB*(25)) performs slightly better than the original filter bank, probably because the approximation induces regularization. *SEP-DIRECT*(25) gave the worst results. **(c)** and **(d)**: Results for the BF2D and the STARE datasets respectively.

datasets, two of which consist of retinal scans so that we can evaluate the generalization capabilities of the learned filter banks:

Table 3. Computational time measured for the convolutions for different image sizes and different filter banks. Time is expressed in seconds, and is measured on a single-core, unoptimized Matlab implementation. Using 25 separable filters to approximate 121 non-separable filters yields a speed up by a factor of 25.

Dataset	**121 filters**		**25 filters**
	full rank	separable	separable
565 × 584 (DRIVE)	9.80	1.91	0.38
896 × 1792 (BF2D)	47.52	10.00	1.86
700 × 605 (STARE)	12.68	2.58	0.46

- The STARE dataset [34] is composed of 20 RGB retinal fundus slides. Half of the images come from healthy patients and are therefore rather clean, while the other half presents pathologies which partly occlude the underlying vasculature and alter its appearance. Moreover, some images are affected by severe illumination changes which challenge automated algorithms.
- The DRIVE dataset [26] is a set of 40 retinal scans captured for the diagnosis of various diseases. It is cleaner than the STARE dataset in that the pathologies affect the image quality less. The dataset is splitted in 20 training images and 20 test images, with two different ground truth sets traced by two different human experts for the test images.
- The BF2D dataset is made by minimum intensity projections of bright-field micrographs of neurons. The images have a very high resolution but exhibit a low signal-to-noise ratio, because of irregularities in the staining process. Furthermore, parts of the dendrites often appear as point-like structures that can be easily mistaken for the structured and unstructured noise affecting the images, as it can be seen in Fig. 2(a). As a consequence, the quality of the ground truth segmentations is poor. Also, only two images have been segmented by a human expert. For this reason we have selected the image with the best ground truth as test image, and used the other image for training.

We have adopted the same framework as in [5], but we have replaced SVM classifiers by Random Forests [35]. Experimentally, we have noticed that this not only brought a considerable speed improvement, but also led to better performance, as can be seen in the comparison presented in Fig. 3(a).

To be consistent with [5], we started by learning one filter bank with 121 filters of size 21 × 21 on the DRIVE dataset and one on the BF2D dataset. The filter bank learned for the latter dataset is depicted by Fig. 2(b). For the STARE dataset, we have used the filter bank learned on the DRIVE dataset to gauge the impact of the approximation on images with similar but not identical statistics. The classification in this latter case was performed on each image in turn, leaving the rest of the dataset as training set. We have then learned other filter banks of reduced cardinality, both full-rank and separable, to assess the impact of the filter bank size on the final classification performance.

For the pixel classification case we have performed experiments with both the strategy of Eq. (2) (*SEP-DIRECT*) and that of Eq. (6) (*SEP-COMB*). The resulting performances are summarized in Fig. 3, while Fig. 2(c) depicts the approximating filter bank learned for *SEP-COMB* on the BF2D dataset.

It is difficult to define a representative metric for evaluating image segmentation. To avoid relying on a single one, which could be misleading, we considered multiple ones instead: Area Under Curve (AUC) computed on ROC curves, the F-measure, the Variation of Information (VI) [36], and the Rand Index (RI) [37].

The different methods are compared in Table 2. *SEP-COMB* scored best again, closely matching the performance of the full-rank filter banks. In this particular case, the results of the SVD-based separable filter approximation (*SEP-SVD*) are remarkably good, and virtually identical to that of *SEP-DIRECT*. This result comes as no surprise, since the structures to which the filters are sensitive are linear, and therefore well matched even by the crude approximation of *SEP-SVD*. However, this technique requires the same number of filters as the original filter bank, while *SEP-DIRECT* and *SEP-COMB* can resort to just a fraction of that quantity. The superior performance of *SEP-COMB* over that of *SEP-DIRECT* can be ascribed to the fact that the optimization algorithm in Eq. (1) has already steered the optimization process in the correct direction.

Table 3 reports the time required by the convolutions in each different approach and for each dataset. Our proposed technique shows a 25× speedup compared to the original filter bank of [5].

4.3 Approximating Existing Filter Banks

The *SEP-COMB* approach presents the advantage of being applicable to situations where an optimal filter bank is already available but a separable approximation is sought for efficiency purposes. To verify that the reconstruction capabilities of our learned basis are sufficient for practical applications, we chose to investigate an image denoising task, where a poor approximation would translate into a bad quality of the denoised image.

To this end, we have considered the image denoising algorithm of [14]. It relies on a filter bank learned with the K-SVD algorithm, which we approximated by a linear combinations of separable filters with *SEP-COMB*. We started from several filter banks learned using the source code provided for [14] by the authors and evaluated the effectiveness of our approximations by plugging them back in the framework of [14]. No specific tuning of neither the approximation algorithm nor the denoising architecture was involved.

As can be seen in Table 4.3, our separable filters approximate the original ones well enough for the end results to be indistinguishable. An example of denoised image, along with the corresponding K-SVD and approximating filter banks, is depicted by Fig. 4. Interestingly, the learned basis of separable filters seems general. We can reuse such a basis learned for a given K-SVD filter bank to closely approximate another K-SVD filter bank corresponding to another image. In other words, we can keep the same $\mathbf{s}_k$ filters, and only optimize on the w_k^j weights in Eq. (6). [20] also considered the approximation of filter banks learned with

Table 4. Results for the image denoising task. The values in the upper part of the table are expressed in decibels, and represent the image Peak Signal-to-Noise Ratio (PSNR). The images were artificially corrupted by additive white Gaussian noise with standard deviation 20. The results have been obtained by replacing the filter bank learned by the K-SVD algorithm with the approximated one in the code provided by [14]. *SEP-COMB-Barbara* denotes the strategy where, instead of grounding the reconstruction on the approximating filter bank corresponding to the image to denoise, the approximating filter bank from the Barbara image is used. In the lower part of the table, the average reconstruction error for the different filter banks, measured as $\mathbb{E}[\|\mathbf{f}^j - \sum_k w_k^j \mathbf{s}_k\|/(\|\mathbf{f}^j\|\|\sum_k w_k^j \mathbf{s}_k\|)]$, is reported. In all of the experiments no tuning of the parameters of neither the approximation nor of the denoising algorithms was performed.

	Barbara	**Boat**	**House**	**Lena**	**Peppers**
Noisy image	22.12	22.09	22.06	22.09	22.13
K-SVD	30.88	30.36	33.34	32.42	32.25
SEP-COMB(25)	30.21	30.27	33.13	32.40	31.99
SEP-COMB(36)	30.77	**30.36**	33.24	32.42	32.08
SEP-COMB(49)	30.87	**30.36**	33.32	32.42	32.17
SEP-COMB(64)	**30.88**	**30.36**	**33.34**	32.42	**32.25**
SEP-COMB-Barbara(36)	-	30.26	32.41	**32.43**	31.97
SEP-COMB-Barbara(64)	-	**30.36**	33.28	**32.43**	32.23
Average filter bank reconstruction error					
SEP-COMB(36)	0.188	0.157	0.262	0.234	0.291
SEP-COMB(64)	0.060	0.054	0.055	0.051	0.050

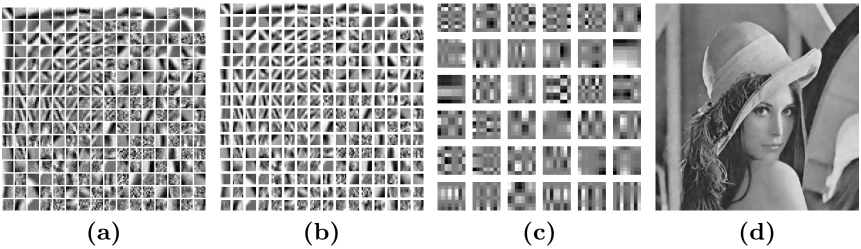

Fig. 4. (a) Filter bank learned by the K-SVD algorithm of [14] on the Lena image corrupted with additive white Gaussian noise of standard deviation 20. **(b)** Approximated filter bank reconstructed from the projection on the separable learned basis depicted in (c). **(c)** The 36 separable filters learned by *SEP-COMB* to approximate this filter bank. **(d)** Lena image denoised using our approximation within the algorithm of [14]. The PSNR is 32.42dB.

the K-SVD algorithm by using sparse linear combinations of separable filters computed from a 1D DCT basis. However, we need significantly less separable filters, only 36 compared to the 100 for [20].

5 Conclusion

In this paper we have proposed two learning-based strategies for obtaining separable filter banks. The first one alters the optimization process for convolutional learning schemes, and allows them to directly learn separable filters. The second one learns a separable basis to approximate an existing filter bank, and not only gets the same performance of its full-rank counterpart, but can also considerably reduce the number of required filters.

The proposed techniques import in the domain of learning approaches one of the most coveted properties of handcrafted filters, separability, and therefore reduce the computational burden traditionally associated with them. This also means that designers of handcrafted filter banks do not have to restrict themselves to separable filters anymore: They can freely choose filters for the application at hand, and approximate them for efficiency using few separable filters with our approach.

References

1. Olshausen, B., Field, D.: Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? Vision Res. (1997)
2. Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T., Yan, S.: Sparse Representation for Computer Vision and Pattern Recognition. IEEE (2010)
3. Mairal, J., Bach, F., Ponce, J.: Task-Driven Dictionary Learning. Technical report, INRIA (2010)
4. Lee, H., Grosse, R., Ranganath, R., Ng, A.: Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. In: ICML. (2009)
5. Rigamonti, R., Türetken, E., González, G., Fua, P., Lepetit, V.: Filter Learning for Linear Structure Segmentation. Technical report, EPFL (2011)
6. Zeiler, M., Taylor, G., Fergus, R.: Adaptive Deconvolutional Networks for Mid and High Level Feature Learning. In: ICCV. (2011)
7. Leung, T., Malik, J.: Representing and Recognizing the Visual Appearance of Materials Using Three-Dimensional Textons. IJCV (2001)
8. Zeiler, M., Krishnan, D., Taylor, G., Fergus, R.: Deconvolutional Networks. In: CVPR. (2010)
9. LeCun, Y., Bottou, L., Orr, G., Müller, K.R. In: Efficient Backprop. Springer (1998)
10. Hinton, G.: Learning to Represent Visual Input. Phil. Trans. R. Soc. B (2010)
11. Bengio, Y. In: Learning Deep Architectures for AI. Now Publishers (2009)
12. Bishop, C.: Pattern Recognition and Machine Learning. Springer (2006)
13. Coates, A., Ng, A.: The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization. In: ICML. (2011)
14. Elad, M., Aharon, M.: Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries. TIP (2006)
15. Treitel, S., Shanks, J.: The Design of Multistage Separable Planar Filters. IEEE Trans. Geosci. Electron. (1971)
16. Perona, P.: Deformable Kernels for Early Vision. PAMI (1995)

17. Boser, B., Sackinger, E., Bromley, J., LeCun, Y., Jackel, L.: Hardware Requirements for Neural Network Pattern Classifiers. IEEE Micro (1992)
18. Farabet, C., Martini, B., Akselrod, P., Talay, S., LeCun, Y., Culurciello, E.: Hardware Accelerated Convolutional Neural Networks for Synthetic Vision Systems. In: ISCAS. (2010)
19. Mnih, V., Hinton, G.: Learning to Detect Roads in High-Resolution Aerial Images. In: ECCV. (2010)
20. Rubinstein, R., Zibulevsky, M., Elad, M.: Double Sparsity: Learning Sparse Dictionaries for Sparse Signal Approximation. IEEE Trans. Signal Process. (2010)
21. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Non-Local Sparse Models for Image Restoration. In: ICCV. (2009)
22. Rigamonti, R., Brown, M., Lepetit, V.: Are Sparse Representations Really Relevant for Image Classification? In: CVPR. (2011)
23. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. IEEE (1998)
24. Fazel, M., Hindi, H., Boyd, S.: A Rank Minimization Heuristic with Application to Minimum Order System Approximation. In: American Contr. Conf. (2001)
25. Bach, F., Jenatton, R., Mairal, J., Obozinski, G. In: Convex Optimization with Sparsity-Inducing Norms. MIT Press (2011)
26. Staal, J., Abràmoff, M., Niemeijer, M., Viergever, M., van Ginneken, B.: Ridge-Based Vessel Segmentation in Color Images of the Retina. TMI (2004)
27. Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the Best Multi-Stage Architecture for Object Recognition? In: ICCV. (2009)
28. Boureau, Y.L., Bach, F., LeCun, Y., Ponce, J.: Learning Mid-Level Features for Recognition. In: CVPR. (2010)
29. Brown, M., Hua, G., Winder, S.: Discriminative Learning of Local Image Descriptors. PAMI (2010)
30. Torralba, A., Fergus, R., Freeman, W.: 80 Million Tiny Images: A Large Dataset for Non-Parametric Object and Scene Recognition. PAMI (2008)
31. Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images. Master's thesis (2009)
32. Santamaría-Pang, A., Colbert, C., Saggau, P., Kakadiaris, I.: Automatic Centerline Extraction of Irregular Tubular Structures Using Probability Volumes from Multiphoton Imaging. In: MICCAI. (2007)
33. González, G., Fleuret, F., Fua, P.: Learning Rotational Features for Filament Detection. In: CVPR. (2009)
34. Hoover, A., Kouznetsova, V., Goldbaum, M.: Location Blood Vessels in Retinal Images by Piecewise Threshold Probing of a Matched Filter Response. TMI (2000)
35. Breiman, L.: Random Forests. Machine Learning (2001)
36. Meilă, M.: Comparing Clusterings - an Information Based Distance. J. Multivariate Anal. (2007)
37. Unnikrishnan, R., Pantofaru, C., Hebert, M.: Toward Objective Evaluation of Image Segmentation Algorithms. PAMI (2007)