Learning Separable Filters*

Roberto Rigamonti Amos Sironi Vincent Lepetit Pascal Fua
CVLab, École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
name.surname@epfl.ch

Abstract

Learning filters to produce sparse image representations in terms of overcomplete dictionaries has emerged as a powerful way to create image features for many different purposes. Unfortunately, these filters are usually both numerous and non-separable, making their use computationally expensive.

In this paper, we show that such filters can be computed as linear combinations of a smaller number of separable ones, thus greatly reducing the computational complexity at no cost in terms of performance. This makes filter learning approaches practical even for large images or 3D volumes, and we show that we significantly outperform state-of-the-art methods on the linear structure extraction task, in terms of both accuracy and speed. Moreover, our approach is general and can be used on generic filter banks to reduce the complexity of the convolutions.

1. Introduction

It has been shown that representing images as sparse linear combinations of learned filters [27] yields effective approaches to image denoising and object recognition, which outperform those that rely on hand-crafted features [38]. Among these, convolutional formulations have emerged as particularly appropriate to handle whole images, as opposed to independent patches [18, 23, 30, 39]. Unfortunately, because the filters are both numerous and not separable, they tend to be computationally demanding, which has slowed down their acceptance. Their computational cost is even more damaging when dealing with large 3D image stacks, such as those routinely acquired for biomedical purposes.

In this paper, we show that we can preserve the performance of these convolutional approaches while drastically reducing their cost by learning and using separable filters that approximate the non-separable ones. Fig. 1 demonstrates this behavior in the case of filters designed to classify whether or not a pixel belongs to a blood vessel in retinal scans. Using the learned separable filters is much faster than using either the original non-separable ones or a state-of-the-art implementation of the FFT for all practical filter sizes. We will demonstrate that this is consistently true over a wide range of images.

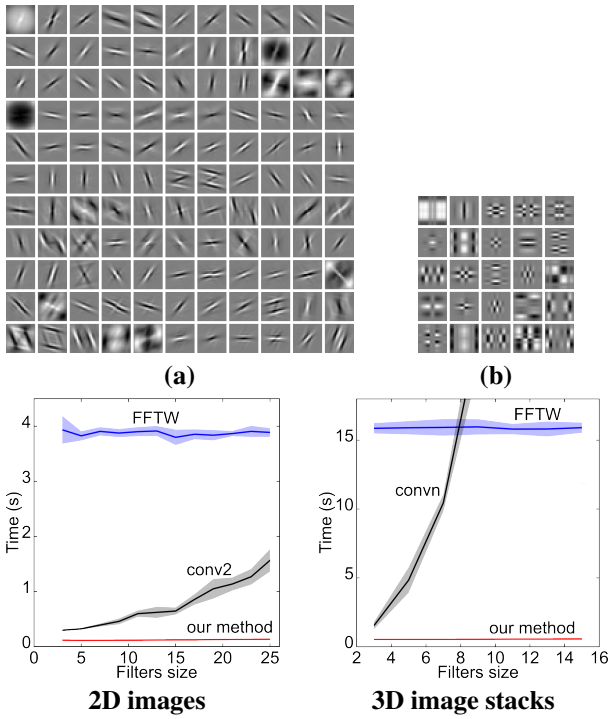


Figure 1. Convolutional filter bank **(a)** learned for the extraction of linear structures in retinal scan images, along with its separable approximation **(b)**. The full-rank filters of **(a)** can be approximated very precisely as linear combinations of the far fewer separable filters of **(b)**. This allows us to use this property to considerably speed up extraction of learned image features compared with convolutions with the original non-separable filters, even when Fast Fourier Transform is used.

As we will see, such a result could be achieved by enforcing the separability constraint as part of a convolu-

*This work was supported in part by the Swiss National Science Foundation and in part by the EU ERC Grant MicroNano.

tional, ℓ_1-based learning framework to directly learn a set of separable filters. However, we have found that an even better result could be obtained by first learning a set of non-separable filters, such as those of Fig. 1(a), and then a second smaller set of separable filters, such as those of Fig. 1(b), whose linear combinations can be used to represent the original filters.

Our contribution is therefore an original approach to approximating non-separable filters as a linear combination of a smaller set of separable ones. It benefits both from the fact that there are fewer filters and that they are separable. Furthermore, for the purpose of finding linear structures, our method is not only faster but also significantly more accurate than one of the best current techniques that relies on hand-designed filters [20].

In the remainder of the paper, we first discuss related work, and then introduce our approach to separable approximation. Finally, we test our method on different applications—pixel and voxel classification as well as image denoising—and show that the speed-up is systematically significant at no loss in performance.

2. Related work

Automatic feature learning has long been an important area in Machine Learning and Computer Vision. Neural networks [21], Restricted Boltzmann Machines [16], Auto-Encoders [5], Linear Discriminant Analysis [6], and many other techniques have been used to learn features in either supervised or unsupervised ways. Recently, creating over-complete dictionaries of features—sparse combinations of which can be used to represent images—has emerged as a powerful tool for object recognition [8, 18, 38] and image denoising [9, 24], among others.

However, for most such approaches, run-time feature extraction can be very time-consuming because it involves convolving the image with many non-separable non-sparse filters. It was proposed several years ago to split convolution operations into convergent sums of matrix-valued stages [35]. This principle was exploited in [28] to avoid coarse discretizations of the scale and orientation spaces, yielding steerable separable 2D edge-detection kernels, and has been extended in [14]. These approaches are powerful but restricted to kernels that can be decomposed efficiently by the method. This precludes the arbitrary ones found in a learned dictionary, or the ones handcrafted to suit particular needs. After more than a decade in which the separability property has been either taken for granted or neglected, there is evidence of renewed interest [25, 29]. The scope of those papers is, however, limited in that they are restricted to particular frameworks, while our approach is completely generic. Nonetheless, they prove a growing need for fast feature extraction methods. Two well known attempts to tackle the computational complexity issue by focusing on aspects other than separability are the steerable filters [12] and the gray-code filter kernels [4]. However, as before, their computational advantage comes at the price of restricting the family of representable filters.

Among recent feature-learning works, very few have revisited the run-time efficiency issue. The majority of those advocate exploiting the parallel capabilities of modern hardware [10, 26]. However, programming an FPGA unit as in [10] is cumbersome. Exploiting the Graphics Processing Unit as in [26] is an attractive alternative, but the time required for memory transfers between the CPU and the GPU is often prohibitive in practice.

An interesting recent attempt at reducing computational complexity is the approach of [32], which involves learning a filter bank by composing a few atoms from an handcrafted separable dictionary. Our own approach is in the same spirit but is much more general as we also learn the atoms. As shown in the results section, this results in a smaller number of separable filters that are tuned for the task at hand.

3. Learning 2D Separable Filters

Most dictionary learning algorithms operate on image patches [27, 24, 8], but convolutional approaches [18, 23, 39, 30] 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 [22]. In our work, we consider the convolutional extension of Olshausen and Field's objective function proposed in [30]. Formally, N filters $\{\mathbf{f}^j\}_{1\leq j\leq N}$ are computed as

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

where

- $\mathbf{x}_i$ is an input image;
- $*$ denotes the convolution product operator;
- $\{\mathbf{m}_i^j\}_{j=1\ldots N}$ is the set of feature maps extracted during learning;
- λ_1 is a regularization parameter.

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's gradient and then applying the soft-thresholding operation [1] on the $\mathbf{m}_i^j$s.

In an earlier report [31] we showed that this formulation allows to extract linear structures in a more reliable way than state-of-the-art methods. However, when dealing

[1] Soft-thresholding is the proximal operator for the ℓ_1 penalty term [3]; its expression is $\text{prox}_\lambda(x) = \text{sgn}(x)\max(|x|-\lambda, 0)$. Proximal operators allow to extend gradient descent techniques to some nonsmooth problems.

with large amounts of data, as it is common in the medical domain, the required run-time convolutions are costly because the resulting 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 $\mathcal{O}(\mathbf{p}\cdot\mathbf{q}\cdot\mathbf{s}\cdot\mathbf{t})$ multiplications and additions. By contrast, if the filters were separable, the computational cost would drop to a more manageable $\mathcal{O}(\mathbf{p}\cdot\mathbf{q}\cdot(\mathbf{s}+\mathbf{t}))$.

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

In the remainder of this section, we introduce two different approaches to overcoming this problem. The first relies on a natural extension of the objective function of Eq. (1), and directly forces the learned filters to be separable by lowering their rank. However, this often degrades the results, most probably because of the additional constraints on the filters, therefore we propose a better and even faster solution. Since arbitrary filters of rank R can be expressed as linear combinations of R separable filters [28], we replace the $\mathbf{f}$ filters of Eq. (1) by linear combinations of filters that are forced to be separable by lowering their rank. This solution is more general than the first, and retains the discriminative power of the full-rank filter bank.

3.1. Penalizing High-Rank Filters

A straightforward approach to finding low-rank filters is to add a penalty term to the objective function of Eq. (1) and to solve

$$\underset{\{\mathbf{s}^j\},\{\mathbf{m}_i^j\}}{\operatorname{argmin}} \sum_i \left(\left\| \mathbf{x}_i - \sum_{j=1}^{N} \mathbf{s}^j * \mathbf{m}_i^j \right\|_2^2 + \Gamma_{\mathbf{m},\mathbf{s}}^i \right), \qquad (2)$$

$$\text{with } \Gamma_{\mathbf{m},\mathbf{s}}^i = \lambda_1 \sum_{j=1}^{N} \left\| \mathbf{m}_i^j \right\|_1 + \lambda_* \sum_{j=1}^{N} \left\| \mathbf{s}^j \right\|_*, \qquad (3)$$

where the $\mathbf{s}^j$s are the learned linear filters, $\|\cdot\|_*$ is the nuclear norm, and λ_* is an additional regularization parameter. The nuclear norm of a matrix is the sum of its singular values and is a convex relaxation of the rank [11]. Thus, forcing the nuclear norm to be small amounts to lowering the rank of the filters. Experimentally, for sufficiently high values of λ_*, the $\mathbf{s}^j$ filters become effectively rank 1 and can be written as products of 1D filters.

Solving Eq. (2), which has the nuclear norm of the filters as an additional term compared to Eq. (1), requires minimal extra effort. After taking a step in the direction opposite of that of the gradient of the filters, as described in the previous section, we just have to apply the proximal operator of the nuclear norm to the filters. This amounts to performing a Singular Value Decomposition (SVD) $\mathbf{s} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ on each filter $\mathbf{s}$, soft-thresholding the values of the diagonal matrix $\mathbf{D}$ to obtain a new matrix $\widehat{\mathbf{D}}$, and replacing $\mathbf{s}$ 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, the second largest singular value is already almost zero even before clipping.

Choosing appropriate values for the optimization 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

In this second approach, we write the N $\mathbf{f}^j$ filters of Eq. (1) as linear combinations of M separable filters $\{\mathbf{s}_k\}_{1\le k\le M}$. In other words, we seek a set w_k^j of 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 $\mathbf{f}^j$s amounts to convolving it with the separable $\mathbf{s}_k$ filters and then linearly combining the results, without further convolutions. This could be achieved by solving

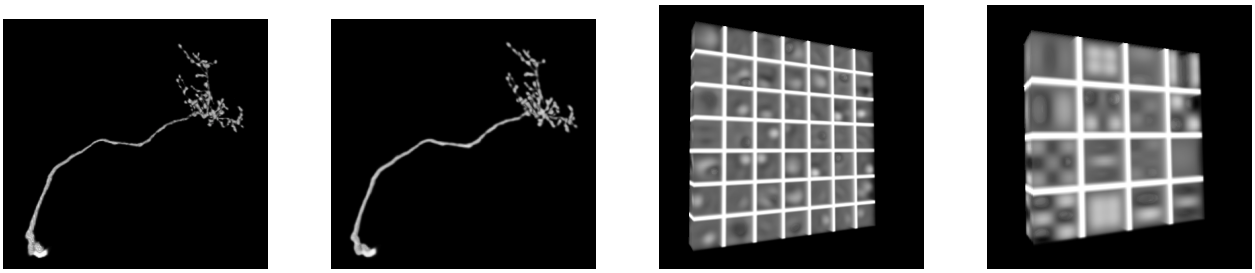
$$\underset{\substack{\{\mathbf{m}_i^j\}\\ \{\mathbf{s}_k\},\{w_k^j\}}}{\operatorname{argmin}} \sum_i \left(\left\| \mathbf{x}_i - \sum_{j=1}^{N} \left(\sum_{k=1}^{M} w_k^j \mathbf{s}_k \right) * \mathbf{m}_i^j \right\|_2^2 + \Gamma_{\mathbf{m},\mathbf{s}}^i \right), \qquad (4)$$

where $\Gamma_{\mathbf{m},\mathbf{s}}^i$ is defined in Eq. (3). This formulation can be seen as a generalization of Eq. (2), which can be retrieved from Eq. (4) by taking $w_k^j = 1$ if $j = k$ and 0 otherwise. Again, we introduce the nuclear norm to force the $\mathbf{s}_k$ filters to be separable. Unfortunately, this objective function 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, making the formulation linear by introducing additional parameters. Parameter tuning is, however, already difficult in the formulation of Eq. (1), and this would therefore only worsen the situation. We tried instead 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 objective function of Eq. (1). We then look for separable filters whose linear combinations approximate the $\mathbf{f}^j$ filters by solving

$$\underset{\{\mathbf{s}_k\},\{w_k^j\}}{\operatorname{argmin}} \sum_j \left\| \mathbf{f}^j - \sum_{k=1}^{M} w_k^j \mathbf{s}_k \right\|_2^2 + \lambda_* \sum_{k=1}^{M} \|\mathbf{s}_k\|_*. \qquad (5)$$

Even though this may seem suboptimal when compared to the global optimization scheme of Eq. (4), it gives superior results in practice because the optimization process is split into two easier tasks and depends on just two parameters, easing their scheduling.



(a) **(b)** **(c)** **(d)**

Figure 2. Examples of non-separable and separable 3D filter banks, learned on the OPF dataset [2]. **(a)** One of the test images. **(b)** Response of the classifier trained on the separable filter bank output **(d)**. **(c)** Non-separable filter bank learned by optimizing Eq. (1). **(d)** The separable filter bank learned by optimizing Eq. (5).

4. Learning Separable 3D Filters

The computational complexity of feature extraction becomes even more daunting when handling 3D volumes such as those of Fig. 2. Fortunately, our approach generalizes naturally to learning 3D separable filters.

As will be shown in the Results section, the formalism of Eq. (1) yields the best results in the 2D case and we therefore rely on it for the proposed extension. The only difference comes from the fact that minimizing the nuclear norm was achieved by SVD decomposition of the 2D filters, which cannot be done for 3D arrays, also known as tensors. Fortunately there are decomposition methods for tensors [19], some of which have already been used in Computer Vision [37, 15]. The most appropriate one for our purpose is the Canonical Polyadic Decomposition (CPD). It decomposes a R-rank tensor into a sum of R rank-one tensors. As it is not possible to know R *a priori*, this becomes a parameter of the decomposition. Given the new decomposition scheme, the structure of the optimization scheme of Eq. (5) is unchanged. It simply involves a CPD of the filters with R set to a large enough value, followed by a soft-thresholding on their coefficients σ_r^k.

To compute the CPDs we tried a simple alternate least-squares optimization and the CP-OPT algorithm of [1], implemented in the MATLAB tensor toolbox. The second technique gave the best results in term of accuracy and convergence speed. Fig. 2(d) depicts an example of the 3D separable filter we obtained and can be compared to the non-separable ones of Fig. 2(c), which were learned by solving the minimization problem of Eq. (1).

5. Results and Discussion

To demonstrate our approach on both 2D and 3D data, we first compare the performance of our separable filters against that of non-separable ones for the purpose of classifying pixels and voxels in biomedical images as belonging to linear structures or not. We show that our separable filters systematically deliver a very substantial speed-up at no loss in performance. We then demonstrate that they can also approximate very effectively non-separable ones learned for denoising purposes.

For the purpose of these comparisons, we will refer to the non-separable filters obtained by minimizing the objective function of Eq. (1) as *NON-SEP*, and the separable ones learned using the technique of Sections 3.1 and 3.2 as *SEP-DIRECT* and *SEP-COMB*, respectively. We will denote by *SEP-SVD* the separable filters obtained by approximating each *NON-SEP* filter by 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.

As discussed in Section 3.2, linear combinations of the *SEP-COMB* filters can be used to represent the *NON-SEP* ones. However for some applications, such as when the filters' output is fed to a linear classifier, it is not necessary to explicitly compute this linear combination because the classifier can be trained directly on the separable-filters' output instead of that of the non-separable ones. This approach, which we will refer to as *SEP-COMB**, further simplifies the run-time computations because the linear combinations are implicitly learned by the classifier at training-time.

5.1. Detection of Linear Structures

Biomedical image processing is a particularly promising field of application for Computer Vision techniques as it involves large numbers of 2D images and 3D image stacks of ever growing size, while imposing strict requirements on the quality and the efficiency of the processing techniques. Here, we demonstrate the power of our separable filters for the purpose of identifying linear structures, a long-standing Computer Vision problem that still remains wide-open when the image data is noisy.

Over the years, models of increasing complexity and effectiveness have been proposed, and attention has recently turned to Machine Learning techniques. [33, 13] apply a Support Vector Machine classifier to the responses of *ad hoc* filters. [33] considers the Hessian's eigenvalues while [13] relies on steerable filters. In [31], it was shown that convolving images with non-separable filter banks learned by

solving the problem of Eq. (1) and training an SVM on the output of those filters outperforms these other methods. Unfortunately, this requires many such non-separable filters, making it an impractical approach for large images or image stacks, whose usage is becoming standard practice in medical imaging. We show that our approach solves this difficulty.

5.1.1 Pixel Classification

In the 2D case we considered the three biomedical datasets of Fig. 3:

- The DRIVE dataset [34] is a set of 40 retinal scans captured for the diagnosis of various diseases. The dataset is split into 20 training images and 20 test images, with two different ground truth sets traced by two different human experts for the test images.
- The STARE dataset [17] is composed of 20 RGB retinal fundus slides. Half of the images come from healthy patients and are therefore rather clean, while the others present pathologies. Moreover, some images are affected by severe illumination changes, which challenge automated algorithms. It is therefore less clean than the DRIVE dataset.
- The BF2D dataset is composed of 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.

We tested all the methods described above on all these images. To this end, we compute the feature maps extracted by the different convolutional filters, and feed them to a Random Forests classifier [7]. Note that we do not need to compute the linear combination of the filter outputs in the case of *SEP-COMB*, since the Random Forest classifier relies on linear projections. We will therefore opt for *SEP-COMB**.

As discussed above, it has been shown in [31] that the *NON-SEP* approach outperforms other recent approaches that rely on Machine Learning but is slow. Our goal is therefore to achieve the same level of performance but much faster. For completeness, we also compare our results to those obtained using Optimally Oriented Flux (*OOF*) [20], widely acknowledged to be one of the best techniques for finding linear structures using hand-designed filters, and a reimplementation of *NON-SEP* that relies on the Fast Fourier Transform to perform the convolutions. This approach is known to speed-up the convolution for large enough filters, and we will refer to it as *NON-SEP-FFT*.

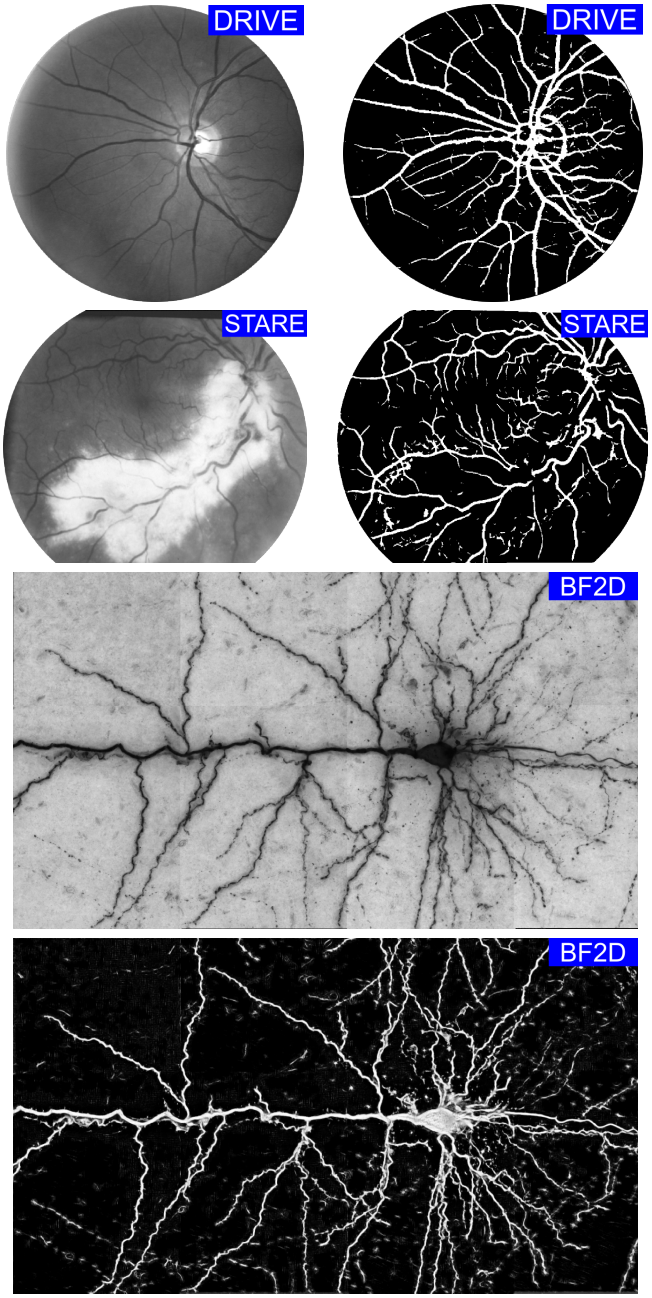


Figure 3. Representative images from the 2D medical datasets considered, together with the corresponding pixel classification results obtained with our *SEP-COMB** method.

An analysis of the computational costs of the different approaches is presented in Fig. 1(bottom). In particular, the graph for the 2D case reports the time in second needed to convolve a 512×512 2D image with a bank of 121 filters, as a function of the filter size, by using the MATLAB's `conv2` function, the FFTW library, and our method [2]. Timings increase quadratically for conv2, and

[2] All the algorithms were optimized to provide a fair comparison.

only linearly for our methods. They are constant for FFTW but much higher even for relatively large filters and even if the image size is optimal for FFTW as it is a power of 2. When a $128 \times 128 \times 64$ 3D volume is considered, the advantage of our method is even clearer, as the cubic complexity makes the computations in the non-separable case impractical even when very few filters are considered. Indeed, the reduction in computational time is largely due to the separability of the filters, and only partially to the reduction of the numbers of the filters involved. Additional results showing this are included in the supplemental material [3].

We first learned a filter bank with 121 learned filters of size 21×21 on the DRIVE dataset and one on the BF2D dataset, as these parameters provided us with the best results. To assess how well our approach generalized, we also used the filter bank learned for the DRIVE dataset for the STARE dataset. 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.

As there is no universally accepted metric to evaluate pixel classification performance, we used several to compare our method to others. In the supplementary material, we report results in terms of the F-measure [36], the Area Under the Curve (AUC) computed on ROC curves, PR and ROC curves, Variation of Information (VI), and Rand Index (RI). We plot these accuracy results against the time it takes to obtain them.

Fig. 4 summarizes these results in the case of the F-measure. More specifically, we treat *OOF* as our baseline and, for each one of the other methods and for every image, we compute the ratio of the F-measure it produces to that of *OOF*. If this ratio is greater than one, the other method performs better than *OOF*. We then plot the average of these ratios over all the images belonging to the same dataset against the time it took to perform the convolutions required to perform the classification.

SEP-COMB* performs consistently best, closely matching the performance of *NON-SEP* but with a significant speed-up. *SEP-DIRECT* is just as fast but entails a loss of accuracy. Somewhat surprisingly, *SEP-SVD* falls between *SEP-DIRECT* and *SEP-COMB* in terms of accuracy but is much slower than both. Finally, *NON-SEP-FFT* yields exactly the same results as *NON-SEP*, but it is much slower than plain 2D convolutions. The costs of the Fourier Transform are indeed amortised only for extremely large image and filter sizes.

[3]The supplemental material, the source codes, and the parameters can be found in the project's web page at `http://cvlab.epfl.ch/research`

All the filter-based methods are more accurate than *OOF*, although the latter does not need a classification step. However, the accuracy of *OOF* is significantly lower than that of filtering-based approaches.

5.1.2 Voxel Classification

We also evaluated our method on classifying voxels as belonging or not to linear structures in 3D volumes of Olfactory Projection Fibers (OPF) from the DIADEM challenge [2], which were captured by a confocal microscope. We learned the 3D filter bank made of 49 $13 \times 13 \times 13$ pixel filters depicted by Fig. 2(c) and the 16 separable filters of Fig. 2(d) using the approach of Sec. 4. As in the 2D case, we then trained classifiers to use these filters, but we used ℓ_1-regularized logistic regressors instead of Random Forests since they have proved faster without significant performance loss. For training we used a set of $200,000$ samples, randomly selected from 4 train images. Since these classifiers do not require us to compute the linear combination of the separable filter outputs, we chose again the *SEP-COMB** approach for our experiments.

As in Section 5.1.1, we use *NON-SEP* as our baseline. We compare *SEP-COMB** against *NON-SEP-FFT*, a Fourier-based implementation of *NON-SEP*, a 3D version of *OOF*, and *SEP-CPD*, which approximates each filter by its rank-one CPD decomposition and is therefore a 3D equivalent of *SEP-SVD*.

The results are essentially the same as in the 2D-case. *SEP-COMB** is 30 times faster than *NON-SEP-FFT* for virtually the same accuracy. It is 4 times faster than *SEP-CPD*, but the latter is also less accurate. As before, *OOF* is even worse in terms of accuracy. Again, we refer the interested reader to the supplementary material for a more detailed set of individual results.

5.2. Denoising

To evaluate how well our *SEP-COMB* approach is at representing a set of generic filters in a very different context, we used it to approximate the 256 denoising filters computed by the K-SVD algorithm [9], some of which are depicted by Fig. 6(b). We experimented with different sizes of the approximating separable filter bank, and reported the results in Tab. 1. As can be seen, the 36 separable filters shown in Fig. 6(a) are already enough to obtain a very accurate approximation, giving a perfect reconstruction of the original filters up to a nearly imperceptible smoothing of the filters with many high-frequency components.

We also compared our results with the *SEP-SVD* approach, and we observed that our method performs similarly or better than it, although the latter requires several times more filters. Table 1 reports the denoising scores, measured using the Peak Signal-to-Noise Ratio (PSNR). [32] also considered the approximation of filter banks learned with the K-SVD algorithm by using sparse linear combinations

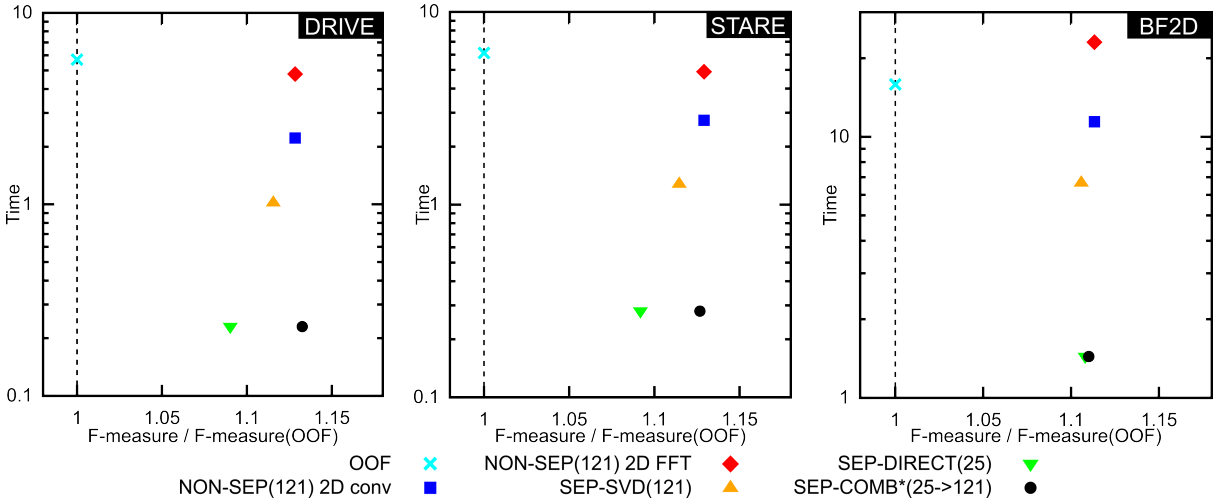


Figure 4. Pixel classification results for the three considered 2D biomedical datasets. The graphs compare the F-measure [36] obtained by the different approaches, normalizing the result by the F-measure obtained with the Optimally Oriented Flux filter [20]. Our filtering approach outperforms the OOF results in all the datasets, and the separable filters do so at a fraction of the computational costs of the non-separable filters, while retaining their accuracy. Times are given in seconds and represent the time it takes to convolve the input image with the considered filter banks. More results are given in the supplementary material.

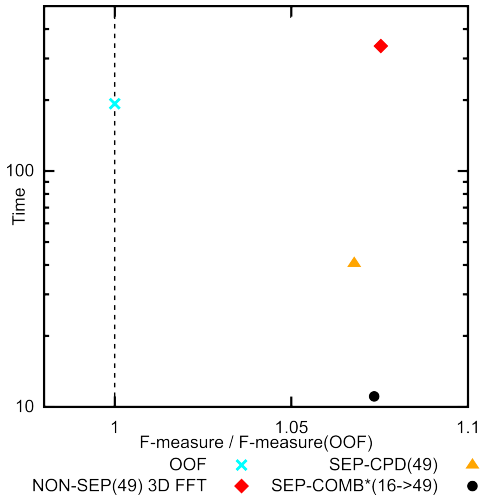


Figure 5. Pixel classification results on the OPF image stack. The F-measure is normalized by the F-measure obtained with the Optimally Oriented Flux filter [20]. More results are given in the supplementary material.

of 1D DCT basis. However, we need significantly fewer separable filters, only 36 compared to the 100 for [32].

Interestingly, the basis of separable filters we learn seem general. We proved that by taking the filters that were learned to approximate a filter bank of a specific image, and we used them to reconstruct the filter banks of the other images. In other words, we kept the same $\mathbf{s}_k$ filters learned for the Barbara image, and only optimized on the w_k^j weights in Eq. (5). The results are summarized in Tab. 1.

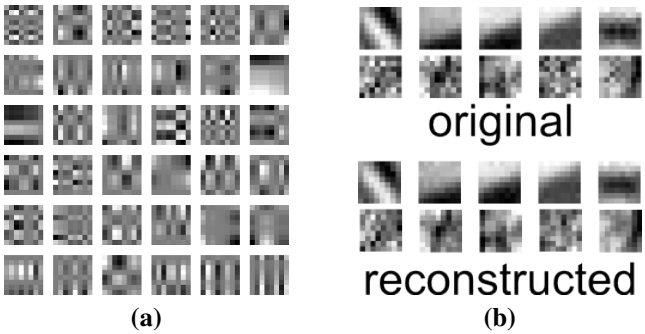


Figure 6. Approximating an existing filter bank. **(a)** The 36 separable filters learned by *SEP-COMB* to approximate a bank of 256 filters learned by K-SVD algorithm of [9]. **(b)** Comparison between some of the original filters learned by K-SVD (top row) and their approximations reconstructed by our algorithm (bottom row). While filters with a regular structure are very well approximated, noisy filters are slightly smoothed by the approximation. Their role in the denoising process is, however, marginal, and therefore this engenders no performance penalty.

6. Conclusion

We have proposed a learning-based filtering scheme applied to the extraction of linear structures, along with two learning-based strategies for obtaining separable filter banks. The first one directly learns separable filters by modifying the regular objective function. The second one learns a basis of separable filters to approximate an existing filter bank, and not only gets the same performance of the original, but also considerably reduces the number of filters, and thus convolutions, required. Although we have presented our results in a convolutional framework, the same conclusions apply to patch-based approaches.

Table 1. Results for the image denoising task. We give here the image Peak Signal-to-Noise Ratio (PSNR) in decibels for different methods. The images were first artificially corrupted by additive Gaussian white noise with standard deviation 20, and denoised with the K-SVD method [9], using the bank of 256 filters computed by the original method and its approximations we obtained with our *SEP-COMB* methods. We obtain similar results with much fewer filters. *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. This filter bank seems general as it does not degrade the results. For all of the experiments no tuning of the parameters for either the approximation or of the denoising algorithms was performed. More results are given in the supplementary material.

	Barbara	**Boat**	**Lena**	**Peppers**
Noisy image	22.12	22.09	22.09	22.13
K-SVD	30.88	30.36	32.42	32.25
SEP-SVD(256)	30.23	30.20	32.08	32.06
SEP-COMB(25)	30.21	30.27	32.40	31.99
SEP-COMB(36)	30.77	30.36	32.42	32.08
SEP-COMB(49)	30.87	30.36	32.42	32.17
SEP-COMB(64)	30.88	30.36	32.42	32.25
SEP-COMB-Barbara(36)	-	30.26	32.43	31.97
SEP-COMB-Barbara(64)	-	30.36	32.43	32.23

Our techniques also bring to learning approaches one of the most coveted properties of handcrafted filters, namely separability, and therefore reduce the computational burden traditionally associated with them. Moreover, 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 using few separable filters with our approach.

References

[1] E. Acar, D. M. Dunlavy, and T. G. Kolda. A Scalable Optimization Approach for Fitting Canonical Tensor Decompositions. *Journal of Chemometrics*, 2011.

[2] G. Ascoli, K. Svoboda, and Y. Liu. Digital Reconstruction of Axonal and Dendritic Morphology DIADEM Challenge, 2010.

[3] F. Bach, R. Jenatton, J. Mairal, and G. Obozienski. Optimization with Sparsity-Inducing Penalties. Technical report, INRIA, 2011.

[4] G. Ben-Artzi, H. Hel-Or, and Y. Hel-Or. The Gray-Code Filter Kernels. *PAMI*, 2007.

[5] Y. Bengio. *Learning Deep Architectures for AI*. Now Publishers, 2009.

[6] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[7] L. Breiman. Random Forests. *Machine Learning*, 2001.

[8] A. Coates and A. Ng. The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization. In *ICML*, 2011.

[9] M. Elad and M. Aharon. Image Denoising via Sparse and Redundant Representations Over Learned Dictionaries. *TIP*, 2006.

[10] C. Farabet, B. Martini, P. Akselrod, S. Talay, Y. LeCun, and E. Culurciello. Hardware Accelerated Convolutional Neural Networks for Synthetic Vision Systems. In *International Symposium on Circuits and Systems*, 2010.

[11] M. Fazel, H. Hindi, and S. Boyd. A Rank Minimization Heuristic with Application to Minimum Order System Approximation. In *ACC*, 2001.

[12] W. Freeman and E. Adelson. The Design and Use of Steerable Filters. *PAMI*, 1991.

[13] G. Gonzalez, F. Fleuret, and P. Fua. Learning Rotational Features for Filament Detection. In *CVPR*, 2009.

[14] C. Gotsman. ConstantTime Filtering by Singular Value Decomposition. *Computer Graphics Forum*, 1994.

[15] T. Hazan, S. Polak, and A. Shashua. Sparse Image Coding Using a 3D Non-Negative Tensor Factorization. In *ICCV*, 2005.

[16] G. Hinton. Learning to Represent Visual Input. *Philosophical Transactions of the Royal Society*, 2010.

[17] A. Hoover, V. Kouznetsova, and M. Goldbaum. Location Blood Vessels in Retinal Images by Piecewise Threshold Probing of a Matched Filter Response. *TMI*, 2000.

[18] K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. LeCun. Learning Convolutional Feature Hierarchies for Visual Recognition. In *NIPS*, 2010.

[19] T. G. Kolda and B. W. Bader. Tensor Decompositions and Applications. *SIAM Review*, 2009.

[20] M. Law and A. Chung. Three Dimensional Curvilinear Structure Detection Using Optimally Oriented Flux. In *ECCV*, 2008.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *PIEEE*, 1998.

[22] Y. LeCun, L. Bottou, G. Orr, and K.-R. Müller. *Efficient Backprop*. Springer, 1998.

[23] H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. In *ICML*, 2009.

[24] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-Local Sparse Models for Image Restoration. In *ICCV*, 2009.

[25] F. Mamalet and C. Garcia. Simplifying Convnets for Fast Learning. In *ICANN*, 2012.

[26] V. Mnih and G. Hinton. Learning to Detect Roads in High-Resolution Aerial Images. In *ECCV*, 2010.

[27] B. Olshausen and D. Field. Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? *Vision Research*, 1997.

[28] P. Perona. Deformable Kernels for Early Vision. *PAMI*, 1995.

[29] H. Pirsiavash and D. Ramanan. Steerable Part Models. In *CVPR*, 2012.

[30] R. Rigamonti, M. Brown, and V. Lepetit. Are Sparse Representations Really Relevant for Image Classification? In *CVPR*, 2011.

[31] R. Rigamonti, E. Türetken, G. González, P. Fua, and V. Lepetit. Filter Learning for Linear Structure Segmentation. Technical report, EPFL, 2011.

[32] R. Rubinstein, M. Zibulevsky, and M. Elad. Double Sparsity: Learning Sparse Dictionaries for Sparse Signal Approximation. *SP*, 2010.

[33] A. Santamaría-Pang, C. Colbert, P. Saggau, and I. Kakadiaris. Automatic Centerline Extraction of Irregular Tubular Structures Using Probability Volumes from Multiphoton Imaging. In *MICCAI*, 2007.

[34] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken. Ridge Based Vessel Segmentation in Color Images of the Retina. *TMI*, 2004.

[35] S. Treitel and J. Shanks. The Design of Multistage Separable Planar Filters. *IEEE Transactions on Geoscience Electronics*, 1971.

[36] C. van Rijsbergen. Foundation of Evaluation. *Journal of Documentation*, 1974.

[37] M. A. O. Vasilescu and D. Terzopoulos. Multilinear Analysis of Image Ensembles: Tensorfaces. In *ECCV*, 2002.

[38] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan. Sparse Representation for Computer Vision and Pattern Recognition. *Proc. IEEE*, 2010.

[39] M. Zeiler, D. Krishnan, G. Taylor, and R. Fergus. Deconvolutional Networks. In *CVPR*, 2010.