PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

ESCUELA DE POSGRADO

Separable Dictionary Learning for Convolutional Sparse Coding via Split Updates

TESIS PARA OPTAR AL GRADO ACADÉMICO DE MAGÍSTER EN PROCESAMIENTO DE SEÑALES E IMÁGENES DIGITALES

AUTOR

Jorge Gerardo Quesada Pacora

ASESOR

Paul Rodriguez Valderrama

Febrero, 2019

To my partner, Diana, for her unlimited and undeserved faith in my capabilities.

To my mother and father, for their unconditional love and patience throughout my education.

Abstract

The increasing ubiquity of Convolutional Sparse Representation techniques for several image processing tasks (such as object recognition and classification, as well as image denoising) has recently sparked interest in the use of separable 2D dictionary filter banks (as alternatives to standard non-separable dictionaries) for efficient Convolutional Sparse Coding (CSC) implementations. However, existing methods approximate a set of K non-separable filters via a linear combination of R ($R << K$) separable filters, which puts an upper bound on the latter's quality. Furthermore, this implies the need to learn first the whole set of non-separable filters, and only then compute the separable set, which is not optimal from a computational perspective.

In this context, the purpose of the present work is to propose a method to directly learn a set of K separable dictionary filters from a given image training set by drawing ideas from standard Convolutional Dictionary Learning (CDL) methods. We show that the separable filters obtained by the proposed method match the performance of an equivalent number of non-separable filters. Furthermore, the computational performance of this learning method is shown to be substantially faster than a state-of-the-art non-separable CDL method when either the image training set or the filter set are large. The method and results presented here have been published [1] at the *2018 IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP 2018). Furthermore, a preliminary approach (mentioned at the end of Chapter 2) was also published at ICASSP 2017 [2].

The structure of the document is organized as follows. Chapter 1 introduces the problem of interest and outlines the scope of this work. Chapter 2 provides the reader with a brief summary of the relevant literature in optimization, CDL and previous use of separable filters. Chapter 3 presents the details of the proposed method and some implementation highlights. Chapter 4 reports the attained computational results through several simulations. Chapter 5 summarizes the attained results and draws some final conclusions.

Keywords

Convolutional Sparse Coding, Separable Filters, Dictionary Learning

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Several techniques in the field of image processing, such as feature extraction, object recognition and image denoising (as well as the fields of machine learning and statistics) rely on the fact that a signal of interest admits a sparse representation over some dictionary set. In general, a dictionary set aimed to represent a signal can be either analytically constructed or learned from a collection of training signals [4]. While analytical dictionaries allow for fast implementations with specific mathematical properties, they fail to adapt to any specific class of signals [5]. Learned dictionaries, on the other hand, arise from a set of training signals, which allows for greater adaptability, sparser representations and in turn, better performance for several applications. However, the increased adaptability provided by learned dictionaries comes at the cost of a high computational complexity in the learning process, and optimizing and speeding up this process is still an active area of research.

Unsupervised approaches for dictionary learning can be generally divided into two categories: standard, patch-based representations, and more recently, convolutional formulations. When the interest signal is an image, patch-based sparse representations (which model signals as linear combinations of the learned dictionaries) involve independently computing the representations over a set of overlapping image patches, thus increasing both the computational cost and memory requirements of the problem [6]. Convolutional formulations, on the other side, model the image as a sum over a set of convolutions between dictionary filters and their corresponding feature maps [7] (see Figure 1.1), and thus are intrinsically suited for handling whole images. Due to this fact, recent years have seen an increasing amount of research in the field of Convolutional Sparse Representations, both in Convolutional Sparse Coding (CSC) and Convolutional Dictionary Learning (CDL) [8].

A particular stream of research that has been gaining atention lately in the field of convolutional formulations is the use of separable filters as an alternative to the standard non-separable dictionary filter sets, due to their lower cost at performing the convolution operations. Separable filters have been tested in applications such as Random Forest (RF) classification [3], Convolutional Neural Networks (CNNs) [9], and Convolutional Sparse Coding [2], and found to provide significant improvements in computational performance with respect to non-separable implementations, with little loss in accuracy or reconstruction quality.

In general, most separable filter based methods rely on learning the separable filter set as an approx-

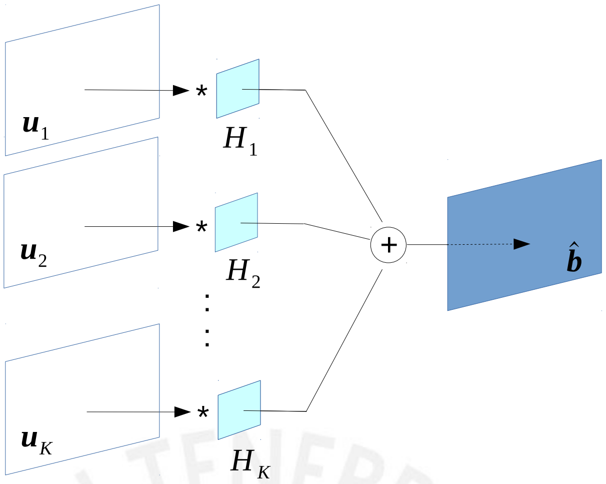


Figure 1.1: Convolutional Sparse Representation model, where $\{H_k\}$ represents the dictionary filters, $\{u_k\}$ represents the corresponding feature maps, and $\hat{b}$ represents the reconstructed image

imation of a previously obtained (usually large) set of non-separable filters, by using the equivalence:

$$H_k \approx \sum_{r=1}^{R} \alpha_{kr} G_r \quad k \in \{1, 2, \dots, K\}, \tag{1.1}$$

which represents each non-separable filter $\{H_k\}$ as a linear combination of a smaller number of separable filters $\{G_r\}$ $(R << K)$ [3]. This approach, however, depends heavily on the quality of the originating non-separable filters to obtain a good separable approximation. Furthermore, it implies a two step procedure: learning first the whole set of standard filters, and only then approximating the separable ones.

In this context, the objective of this thesis is to propose and evaluate an efficient separable filter learning algorithm that can learn the separable set directly from an image training set, without the need of a pre-computed set of non-separable filters. The method is derived through a reformulation of the standard (non-separable) CDL problem, and compared against both standard non-separable dictionaries and separable approximations obtained via (1.1). The filters learned through our method are shown to match the performance of a standard non-separable set (and substantially outperform approximated separable sets) when evaluated through denoising and inpainting tasks. Furthermore, the proposed learning algorithm is shown to be faster than standard non-separable learning approaches for most configurations.

Chapter 2 covers the relevant state of the art for both dictionary learning and separable filters, and is distributed as follows: Section 2.1 presents some preliminary notions regarding the usual optimization strategies used in CDL; Section 2.2 provides a brief review of the existing methods for dictionary learning in the literature, and Section 2.3 further details previous works on separable filter approximations. Section 2.4 details a preliminary approach in which we assess the suitability of separable filters for CSC tasks. Chapter 3 presents the details of our proposed algorithm and relevant information regarding its efficient implementation. Finally, Chapter 4 reports the computational results obtained by comparing our proposed method with existing state-of-the-art aproaches.

Chapter 2

Convolutional Sparse Coding

As we mentioned in the previous Chapter, convolutional sparse coding (CSC) models an entire image a as a sum over a set of convolutions between coefficient maps (of the same size as the target image) $\{\mathbf{u}_k\}$, with their corresponding dictionary filters $\{H_k\}$. A common representation for CSC is through the Convolutional Basis Pursuit Denoising (CBPDN) problem, namely:

$$\underset{\{\mathbf{u}_k\}}{\arg\min} \frac{1}{2} \left\| \sum_{k=1}^{K} H_k * \mathbf{u}_k - \mathbf{b} \right\|_2^2 + \lambda \sum_{k=1}^{K} \|\mathbf{u}_k\|_1 \tag{2.1}$$

where the ℓ_1 norm regulates the sparsity of the feature maps. The corresponding Convolutional Dictionary Learning (CDL) problem for a given image training set $\{\mathbf{b}_s\}$ is

$$\begin{aligned} \min_{\{H_k, \mathbf{u}_{k,s}\}} \frac{1}{2} \sum_{s=1}^{S} \left\| \sum_{k=1}^{K} H_k * \mathbf{u}_{k,s} - \mathbf{b}_s \right\|_2^2 &+ \lambda \sum_{s=1}^{S} \sum_{k=1}^{K} \left\|\mathbf{u}_{k,s}\right\|_1 \\ \text{s.t.} \quad \|H_k\|_2 = 1 \; \forall k \,, & \end{aligned} \tag{2.2}$$

where the ℓ_2 constraint is used to avoid scaling ambiguities.

The main advantage of using the convolutional approach over the patch-based one is that the former provides a *translation invariant* representation, thus eliminating several redundancies occurring in standard dictionaries. However, in most cases the dictionary filters are numerous and non-separable, which implies a significant computational overhead. Some works [3, 10, 9] have addressed this issue by using separable filter approximations (Figure 2.1) to improve runtime efficiency. In this chapter, we give a brief summary of the existing methods for standard (non-separable) Convolutional Dictionary Learning (CDL), as well as review recently proposed methods for approximating separable filters and their applications.

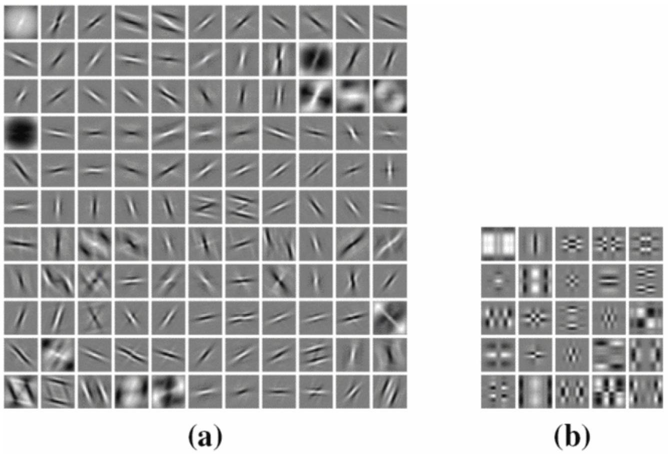


Figure 2.1: Sample dictionary sets. (a) Set of 121 standard (non-separable) filters, (b) Set of 25 separable filters used to approximate (a). Image taken from [3].

2.1 Preliminaries

2.1.1 Alternating Direction Method of Multipliers

The ADMM algorithm [11] is a well-known method, originally derived to blend the benefits of dual decomposition and augmented Lagrangian methods for constrained optimization. The algorithm can be employed to solve an optimization problem of the form

$$\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x}) + g(\mathbf{y}) \qquad \text{s.t. } F\mathbf{x} + G\mathbf{y} - \mathbf{c} = 0. \tag{2.3}$$

where $f(\cdot)$ and $g(\cdot)$ are convex. Several standard problems can be posed as (2.3) by simply splitting the main variable into two parts, thus making ADMM a highly versatile method. The ADMM iterations with scaled dual variable are given by (2.4)-(2.6)

$$\begin{aligned}
\mathbf{x}^{(n+1)} &= \min_{\mathbf{x}} f(\mathbf{x}) + \frac{\rho}{2}\|F\mathbf{x} + G\mathbf{y}^{(n)} - \mathbf{c} + \mathbf{z}^{(n)}\|_2^2 && (2.4)\\
\mathbf{y}^{(n+1)} &= \min_{\mathbf{y}} g(\mathbf{y}) + \frac{\rho}{2}\|F\mathbf{x}^{(n+1)} + G\mathbf{y} - \mathbf{c} + \mathbf{z}^{(n)}\|_2^2 && (2.5)\\
\mathbf{z}^{(n+1)} &= \mathbf{z}^{(n)} + F\mathbf{x}^{(n+1)} + G\mathbf{y}^{(n+1)} - \mathbf{c}\,. && (2.6)
\end{aligned}$$

where the Augmented Lagrangian parameter ρ determines the step size. The ADMM method has been regularly used in the literature as an efficient solution for CSC problems.

2.1.2 Iterative Shrinkage-Thresholding Algorithm and variants

Another well-known approach for CSC problems is the Iterative Shrinkage-Thresholding Algorithm (ISTA) and its ”Fast” variant (FISTA) [12]. These methods are devised for iteratively solving problems of the form

$$\min_{x} f(\mathbf{x}) + \lambda R(\mathbf{x}) \tag{2.7}$$

Algorithm 1 ISTA method.

```
Input: $\lambda$ (parameter), $\mathbf{x}^0$ (initial guess)

  for $n \geq 0$ do
      $\mu_n \in [0, \frac{1}{\|D^T D\|}]$
      $\mathbf{x}^n = \text{shrink}(\mathbf{x}^{n-1} - \mu_n \nabla f(\mathbf{x}^{n-1})), t_n \lambda)$
  end for
```

Algorithm 2 FISTA method.

```
Input: $\lambda$ (parameter), $L$ (Lipschitz constant of $\nabla f(\mathbf{x})$), $\mathbf{x}^0$ (initial guess)

  $\mathbf{y}_1 \leftarrow \mathbf{x}_0$
  $\beta_1 \leftarrow 1$
  for $n \geq 0$ do
      $\mathbf{x}^n \leftarrow \text{shrink}(\mathbf{x}^{n-1} - \frac{1}{L} \nabla f(\mathbf{x})|_{x=y^n}, \frac{\lambda}{L})$
      $\beta_{n+1} \leftarrow \frac{1+\sqrt{1+4\beta_n}}{2}$
      $\mathbf{y}^{n+1} \leftarrow \mathbf{x}^n + \frac{\beta_n - 1}{\beta_{n+1}}(\mathbf{x}^n - \mathbf{x}^{n-1})$
  end for
```

where $f(\cdot)$ is usually a least squares fidelity term such as in the case of the ℓ_2 terms of Eqs. (2.1) and (2.2) (its worth bearing in mind that the convolution operation can be cast as a linear operation).

The corresponding steps for ISTA and FISTA are depicted in algorithm 1 and 2, respectively. As can be observed (and as the name suggests) both methods consist on iteratively updating the interest variable through a gradient step and thresholding (shrinkage when $R(\cdot)$ is the ℓ_1 norm) the result. The main difference between them is that FISTA employs an auxiliary linear combination of previous estimates, which significantly accelerates convergence.

2.2 Non-separable (standard) dictionary learning

Since the CDL problem, as posed by (2.2), is non-convex when dealing with both variables ($\{\mathbf{u}_{k,s}\}$ and $\{H_k\}$) simultaneously, but becomes convex when keeping either of them constant, the most widely used minimization approach consists in alternating between the updates for the feature maps $\{\mathbf{u}_{k,s}\}$ (sparse coding) and the filters $\{H_k\}$ (dictionary learning). This section will address the main existing dictionary learning update methods (for a thorough review and comparison of sparse coding and dictionary learning updates and their coupling mechanisms, see [8]), which require solving a constrained convolutional form of the Method of Optimal Directions (MOD [13]), namely:

$$\min_{\{D_k\}} \frac{1}{2} \sum_{s=1}^{S} \left\| \sum_{k=1}^{K} H_k * \mathbf{u}_{k,s} - \mathbf{b}_s \right\|_2^2, \quad s.t. \quad \|H_k\|_2 = 1, \forall k, \tag{2.8}$$

for a given coefficient set $\{\mathbf{u}_{k,s}\}$.

Early methods solved this problem in the spatial domain, via variants of gradient descent [14] and MOD [15], among others [16, 17]. More recent implementations solve the most computationally demanding components of the problem in the frequency domain due to the associated speedup [8].

When performing the convolutions in the frequency domain, the filters must be zero-padded in order to have an adequate spatial support. This requirement can be denoted by a zero-padding projection operator P, and coupled with the normalization constraint into the constraint set:

$$C_{PN} = \{x \in \mathbb{R}^N : (I - PP^T)x = 0, \|x\|_2 = 1\}, \tag{2.9}$$

which allows to write the dictionary update in unconstrained form:

$$\min_{\{H_k\}} \frac{1}{2} \sum_{s=1}^{S} \left\| \sum_{k=1}^{K} H_k * \mathbf{u}_{k,s} - \mathbf{b}_s \right\|_2^2 + \sum_{k=1}^{K} \iota_{C_{\mathrm{PN}}}(H_k), \tag{2.10}$$

where $\iota_{C_{\mathrm{PN}}}(\cdot)$ is the indicator function of the constraint set C_{PN}. Several algorithms have been proposed to solve (2.10), most of which are based on Augmented Lagrangian frameworks, differing primarily on the approach they take to solve the ℓ_2 fidelity term sub-problem. [18] proposed an Alternating Direction Method of Multipliers (ADMM [11]) formulation, which [7] and [19] later improved by efficiently approaching the aforementioned sub-problem using Iterated Sherman Morrison and ADMM consensus solutions, respectively. Furthermore, [20] proposes an ADMM-consensus and a 3D formulation that decouple the problem from the number of training images S, thus improving the computational performance for the learning update.

There are also variants of these methods that perform the dictionary update in an online fashion such as [21] and [22], in order to save either computing time or memory resources during the learning process.

2.3 Separable from non-separable approximation

A straightforward approach to estimate G_r (as defined in Equation (1.1)) from a given set of standard filters $\{H_k\}$ was proposed in [3, 10] by placing a penalty on high-rank filters, namely

$$\min_{\{G_r, \alpha_{rk}\}} \frac{1}{2} \sum_{k=1}^{K} \|H_k - \sum_{r=1}^{R} \alpha_{rk} \cdot G_r\|_F^2 + \lambda \sum_{r=1}^{R} \|G_r\|_*, \tag{2.11}$$

where $\|\cdot\|_*$ is the nuclear norm. [3, 10] highlighted that the choice of λ is a challenging task, and that convergence was slow when estimating high-rank filters. They also proposed a second approach based on tensor decomposition [23] that provides faster performance:

$$\min_{\{\alpha_{rk}, x_r, y_r\}} \frac{1}{2} \sum_{k=1}^{K} \|H_k - \sum_{r=1}^{R} \alpha_{rk} \cdot x_r \circ y_r\|_F^2, \tag{2.12}$$

where x_r and y_r are rank-1 tensors and $\circ$ represents tensor outer product. A reformulation of this problem as a special case of the low-rank basis problem was proposed in [24], but the authors reported that the tensor approach was significantly faster and attained the same accuracy.

An auxiliary variable formulation of (2.11) given by:

$$\min_{\{G_r,\alpha_{rk},F_r\}} \frac{1}{2}\sum_{k=1}^{K} \|H_k - \sum_{r=1}^{R} \alpha_{rm} G_r\|_F^2 + \frac{\lambda}{2}\sum_{r=1}^{R} \|G_r - F_r\|_F^2$$
$$\text{s.t. rank}(F_r) = 1 \quad \forall r. \tag{2.13}$$

was proposed in [25] along with an efficient SVD-based generation of the initial solution. The method was shown to be faster than the tensor decomposition approach for small R (< 40) values while attaining comparable accuracy.

2.4 Preliminary approach: Separable CSC

The convenience of using separable filters was initially assessed through machine learning tasks, such as Random Forest (RF) image classification [3] and Convolutional Neural Network (CNN) acceleration [9]. In both cases the use of separable filters proved to be an efficient alternative to their non-separable counterparts, in the sense that a small set of separable filters provided comparable performance to a larger set of non-separable ones.

Prior to the development of the proposed separable filter learning method, we derived in [2] a FISTA-based CBPDN solver in order to assess the suitability of separable filter sets in CSC tasks against the standard (non-separable) versions. The algorithm exploits the separability property by computing the convolutions in the spatial domain, and enhances the convergence properties of the standard FISTA method by incorporating the optimal step size rule of the Normalized Iterative Hard Thresholding (NIHT) algorithm [26]. The method also incorporated and validated the use of an alternative two-term penalty function combining the standard ℓ_1 norm and the Non-Negative Garrote [27] penalty term.

In general, the proposed FISTA-based method is aimed at learning the feature map representation of a target image with an approximated separable dictionary set by solving:

$$\arg\min_{\{\mathbf{u}_k\}} \frac{1}{2}\left\|\sum_{r=1}^{R} G_r * \left(\sum_{k=1}^{K} \alpha_{kr}\mathbf{u}_k\right) - \mathbf{b}\right\|_2^2 + \lambda\sum_{k=1}^{K} p(\mathbf{u}_k) \tag{2.14}$$

where $p(\mathbf{x}) = \alpha\|\mathbf{x}\|_1 + \beta\phi_{\text{nng}}(\mathbf{x})$ is the aforementioned two-term penalty function, and $\{G_r\}$ and $\{\alpha_k r\}$ are the separable dictionaries and their corresponding linear coefficients, as described in section 2.3. The method is also used to solve

$$\arg\min_{\{\mathbf{v}_k\}} \frac{1}{2}\left\|\sum_{r=1}^{R} G_r * \mathbf{v}_r - \mathbf{b}\right\|_2^2 + \lambda\sum_{r=1}^{K} p(\mathbf{v}_r) \tag{2.15}$$

which implies learning the feature maps directly from the separable filters, without the underlying relation to the non-separable set. This particular setup led to the highest gain in computational performance with a small decrease in reconstruction quality with respect to solving (2.14). This was mainly due to the fact that the separable filter set had been approximated from a non-separable one, and thus the separable filters had not been learned with the purpose of directly approximating images (its worth bearing in mind that at the time of publication of [2], there were no available methods to natively learn separable filters). In general, both sets of experiments found that a small separable set performed competitively with respect to a larger non-separable set, at a fraction of the computational run-time.

Chapter 3

Separable Dictionary Learning

3.1 Proposed method derivation

The separable filter learning problem is given by replacing the dictionary term $\{H_k\}$ in the CDL problem presented in Chapter 2, with the vertical and horizontal filter components $\{v_r\}$ and $\{h_r\}$, which yields:

$$\begin{aligned} \min_{\{h_r, v_r, \mathbf{u}_{r,s}\}} \frac{1}{2} \sum_{s=1}^{S} \left\| \sum_{r=1}^{R} v_r * h_r * \mathbf{u}_{r,s} - \mathbf{b}_s \right\|_2^2 + \lambda \sum_{s=1}^{S} \sum_{r=1}^{R} \|\mathbf{u}_{r,s}\|_1 \\ \text{s.t.} \quad \|\mathbf{h}_r\|_2 = \|\mathbf{v}_r\|_2 = 1 \; \forall r \,, \end{aligned} \tag{3.1}$$

Writing the dictionary update for (3.1) and coupling the norm constraint with the zero-padding restriction described in Chapter 2.2 gives the unconstrained problem

$$\min_{\{h_r, v_r\}} \frac{1}{2} \sum_{s=1}^{S} \left\| \sum_{r=1}^{R} v_r * h_r * \mathbf{u}_{r,s} - \mathbf{b}_s \right\|_2^2 + \sum_{r=1}^{R} \iota_{C_{\mathrm{PhN}}}(h_r) + \iota_{C_{\mathrm{PvN}}}(v_r), \tag{3.2}$$

where $\iota_{C_{\mathrm{PN}}}(\cdot)$ is the indicator function of the constraint set defined in (2.9), and the zero-padding operator P is applied either along the horizontal or vertical dimension, depending on the filter set.

We approach the solution of (3.2) by alternating between updating the horizontal filters h_r and the vertical ones v_r. Considering only the solution for the vertical filters v_r (assuming fixed horizontal filters), and reformulating the problem in ADMM-compatible form in a fashion reminiscent of [7] leads to the following expression:

$$\begin{aligned} \min_{\{v_r, g_r\}} \frac{1}{2} \sum_{s=1}^{S} \left\| \sum_{r=1}^{R} v_r * \mathbf{u}'_{r,s} - \mathbf{b}_s \right\|_2^2 + \sum_{r=1}^{R} \iota_{C_{\mathrm{PvN}}}(g_r) \\ s.t. \quad v_r - g_r = 0, \forall r, \end{aligned} \tag{3.3}$$

where $\mathbf{u}'_{r,s}$ is the result of convolving the horizontal filters h_r with the feature maps $\mathbf{u}_{r,s}$. The associated subproblems would be:

$$v_r^{(i+1)} = \arg\min_{v_r} \frac{1}{2}\sum_{s=1}^{S}\left\|\sum_{r=1}^{R} v_r * \mathbf{u'}_{r,s} - \mathbf{b}_s\right\|_2^2 + \frac{\rho}{2}\sum_{r=1}^{R}\left\|v_r - g_r^{(i)} + f_r^{(i)}\right\|_2^2 \tag{3.4}$$

$$g_r^{(i+1)} = \arg\min_{g_r} \sum_{r=1}^{R} \iota_{C_{\mathrm{PvN}}}(g_r) + \frac{\rho}{2}\sum_{r=1}^{R}\left\|v_r^{(i+1)} - g_r + f_r^{(i)}\right\|_2^2 \tag{3.5}$$

$$f_r^{(i+1)} = f_r^{(i)} + v_r^{(i+1)} - g_r^{(i+1)} \tag{3.6}$$

It can be observed that (3.5) has the following form:

$$\arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 + \iota_{C_{\mathrm{PvN}}}(\mathbf{x}) = \mathrm{prox}_{\iota_{C_{\mathrm{PvN}}}}(\mathbf{y}). \tag{3.7}$$

where $\mathrm{prox}_{\iota_{C_{\mathrm{PvN}}}}(\cdot)$ is the proximal operator [28] of the indicator function $\iota_{C_{\mathrm{PvN}}}(\cdot)$. Thus the minimizer for (3.5) is given by:

$$\mathrm{prox}_{\iota_{C_{\mathrm{PvN}}}}(\mathbf{y}) = \frac{PP^T\mathbf{y}}{\|PP^T\mathbf{y}\|_2}, \tag{3.8}$$

For notational simplicity we rewrite (3.4) as

$$v_r^{(i+1)} = \arg\min_{v_r} \frac{1}{2}\sum_{s=1}^{S}\left\|\sum_{r=1}^{R} v_r * \mathbf{u'}_{r,s} - \mathbf{b}_s\right\|_2^2 + \frac{\rho}{2}\sum_{r=1}^{R}\|v_r - z_r\|_2^2, \tag{3.9}$$

where $z_r = g_r^{(i)} - f_r^{(i)}$.

When performing standard CDL [7], the non-separable equivalent of (3.9) is solved by switching to the DFT domain and solving the associated linear system. In the separable case, however, it's worth noting that since the filters $\{v_r\}$ are 1-D, whereas the coefficient maps $\{\mathbf{u'}_{r,s}\}$ are 2-D, moving directly onto the frequency domain would require the DFT solution ($\hat{v}_r$) to be a 2-D matrix composed of replicating columns, which would make the convolution operation in (8) have the form

$$\begin{bmatrix} | & | & & | \\ \hat{\mathbf{v}}_r & \hat{\mathbf{v}}_r & \cdots & \hat{\mathbf{v}}_r \\ | & | & & | \end{bmatrix} .^{*} \begin{bmatrix} & \hat{\mathbf{U}}'_{r,s} & \end{bmatrix}$$

This would mean including an additional constraint and further increasing the complexity of the problem. Instead we choose to rewrite the fidelity ℓ_2-norm term as a sum over columns:

$$\left\|\sum_{r=1}^{R} v_r * \mathbf{u'}_{r,s} - \mathbf{b}_s\right\|_2^2 = \sum_{i=1}^{I}\left\|\sum_{r=1}^{R} v_r * \mathbf{u'}_{r,s}[i] - \mathbf{b}_s[i]\right\|_2^2, \tag{3.10}$$

where $\mathbf{u'}_{r,s}[i]$ and $\mathbf{b}_s[i]$ are the i-th columns of the corresponding feature map and training image respectively. Replacing the equality in (3.9):

$$\arg\min_{v_r} \frac{1}{2}\sum_{s=1}^{S}\sum_{i=1}^{I}\left\|\sum_{r=1}^{R} v_r * \mathbf{u'}_{r,s}[i] - \mathbf{b}_s[i]\right\|_2^2 + \frac{\rho}{2}\sum_{r=1}^{R}\|v_r - z_r\|_2^2 \tag{3.11}$$

Switching to the DFT domain, and defining $\hat{\mathbf{U}}'_{r,s}[i] = \mathrm{diag}(\hat{\mathbf{u}}'_{r,s}[i])$ gives:

$$\arg\min_{v_r} \frac{1}{2} \sum_{s=1}^{S} \sum_{i=1}^{I} \left\| \sum_{r=1}^{R} \hat{\mathbf{U}}'_{r,s}[i] \hat{v}_r - \hat{\mathbf{b}}_s[i] \right\|_2^2 + \frac{\rho}{2} \sum_{r=1}^{R} \| \hat{v}_r - \hat{z}_r \|_2^2 \tag{3.12}$$

Defining:

$$\hat{\mathbf{U}}'_s[i] = (\hat{\mathbf{U}}'_{0,s}[i] \quad \hat{\mathbf{U}}'_{1,s}[i] \quad \ldots) \quad \hat{v} = \begin{bmatrix} \hat{v}_1 \\ \hat{v}_2 \\ \vdots \\ \hat{v}_R \end{bmatrix} \quad \hat{z} = \begin{bmatrix} \hat{z}_1 \\ \hat{z}_2 \\ \vdots \\ \hat{z}_R \end{bmatrix} \tag{3.13}$$

the problem can be expressed as

$$\arg\min_{v_r} \frac{1}{2} \sum_{s=1}^{S} \sum_{i=1}^{I} \| \hat{\mathbf{U}}'_s[i] \hat{v} - \hat{\mathbf{b}}_s[i] \|_2^2 + \frac{\rho}{2} \| \hat{v} - \hat{z} \|_2^2 \tag{3.14}$$

Finally, to overcome the column indexing we define:

$$\hat{\mathbf{U}}'_s = \begin{bmatrix} \hat{\mathbf{U}}'_s[1] \\ \hat{\mathbf{U}}'_s[2] \\ \vdots \\ \hat{\mathbf{U}}'_s[I] \end{bmatrix}. \tag{3.15}$$

Substituting $\hat{\mathbf{U}}'_s$ and recovering the full vectorized DFT training images $\hat{\mathbf{b}}_s$ leads to the problem being expressed as:

$$\arg\min_{v_r} \frac{1}{2} \sum_{s=1}^{S} \| \hat{\mathbf{U}}'_s \hat{v} - \hat{\mathbf{b}}_s \|_2^2 + \frac{\rho}{2} \| \hat{v} - \hat{z} \|_2^2, \tag{3.16}$$

with solution:

$$(\sum_s \hat{\mathbf{U}}_s'^H \hat{\mathbf{U}}'_s + \rho I) \hat{v} = \sum_s \hat{\mathbf{U}}_s'^H \hat{\mathbf{b}}_s + \rho \hat{z}. \tag{3.17}$$

Due to the commutativity property of the convolution operation, the update for the horizontal filters h_r can be easily derived by fixing the vertical filters, defining $\mathbf{u'}_{r,s} = v_r * \mathbf{u}_{r,s}$, and following an analogous chain of derivations as the one described in this section.

3.2 Implementation remarks

The most widely used method to deal with the non-separable equivalent of (3.17) is the Iterative Sherman Morrison (ISM) procedure from [7], which is designed for solving multiple diagonal block linear systems of the form $(\sum_k A_k^H A_k + B)x = c$, namely

$$\begin{pmatrix} A_{0,0}^H A_{0,0} + A_{1,0}^H A_{1,0} + \ldots + B_0 & A_{0,0}^H A_{0,1} + A_{1,0}^H A_{1,1} + \ldots & \ldots \\ A_{0,1}^H A_{0,0} + A_{1,1}^H A_{1,0} + \ldots & A_{0,1}^H A_{0,1} + A_{1,1}^H A_{1,1} + \ldots + B_1 & \ldots \\ A_{0,2}^H A_{0,0} + A_{1,2}^H A_{1,0} + \ldots & A_{0,2}^H A_{0,1} + A_{1,2}^H A_{1,1} + \ldots & \ldots \\ \vdots & \vdots & \ddots \end{pmatrix}$$
$$\times \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \end{pmatrix}. \tag{3.18}$$

where $A_k = (A_{k,0} \quad A_{k,1} \quad \ldots \quad A_{k,M-1})$, and both $A_{i,j}$ and B_i are diagonal matrices. While (3.17) is compatible with the ISM procedure, the additional column indexing we introduce here drastically increases the number of terms in each block summation, which entails a significant computational overhead for this method (about an order of magnitude slower than the non-separable scenario), rendering ISM impractical for this task.

We thus solve the linear system given by (3.17) by applying Conjugate Gradient (CG) method [29], which attains competitive or even superior computational performance with respect to the non-separable scenario. Furthermore, in order to minimize the number of inner CG iterations, we use the solution for each previous update as the initial value, as suggested in [8].

The full dictionary learning algorithm is implemented by combining the proposed update method for $\{v_r\}$ and $\{h_r\}$ with the ADMM-based sparse coding update proposed in [7]. Based on standard non-separable implementations, and the results provided by [20], we interleave a single iteration of each update per outer loop, and transfer the auxiliary variables of each ADMM framework across the other update steps, which has been shown to provide the most stable convergence ratio among the other possible choices.

Chapter 4

Results

In this section we assess the performance of the proposed separable dictionary learning method in terms of reconstruction performance for denoising and inpainting CSC tasks, along with convergence and computational runtime for the learning process.

4.1 Experimental framework

For the denoising tests, we used a set of 5 well-known images (see Figure A.1 in the Appendix) corrupted with AWGN ($\sigma = 0.2$), to perform CBPDN using the following sets of filters of different sizes:

- **Nat-sep**: 36 Natively learned separable filters (our proposed method)
- **Apr-sep**: 36 Separable filters approximated from 36 non-separable ones via [25]
- **Non-sep**: 36 Standard non-separable filters learned via [7]

Since the CBPDN problem (as defined in (2.1)) has a tunable parameter λ, in order to ensure fair evaluation we solve for a grid of λ values in the range $[10^{-1}, 1]$, and compare only the optimal performance (in terms of the SSIM metric) for each of the evaluated filter sets. An example of the entire simulation results for a single image is given in Figure 4.1.

For the separable dictionaries (nat-sep and apr-sep), we use the ℓ_1 version of the FISTA-based CBPDN solver proposed in [2] that exploits filter separability by computing the convolutions in the spatial domain. For the non-separable dictionaries (non-sep), we use the ADMM-based solver from [7], which is considered to be state-of-the-art for this problem.

Our inpainting comparisons use a similar setup to the denoising ones. We use the same batch of 5 images as before and randomly discard half the pixels in each one, and evaluate the suitability of each filter set to reconstruct the original images (to this end we use the inpainting CBPDN solver from the SPORCO library [30]). In this case we also perform the evaluation across a grid of λ values in the range $[10^{-3}, 10^{-1}]$, and report the optimal value attained by each filter set.

For the computational performance simulations, we evaluate the learning time for different training set sizes (S) and filter set sizes ($R = K$) against a state-of-the-art ADMMM-based non-separable CDL

	Dict. Size	barbara	mandrill	parrots	boats	goldhill	Time
nat-sep	8x8	0.6175	0.5188	0.7188	0.6438	0.6709	40,77
	12x12	0.6370	0.5248	0.7219	0.6532	0.6730	60,57
	16x16	0.6285	0.5223	0.7197	0.6554	0.6728	70,11
non-sep	8x8	0.6189	0.5218	0.7207	0.6449	0.6732	70,3
	12x12	0.6330	0.5300	0.7225	0.6507	0.6763	104,7
	16x16	0.6283	0.5257	0.7218	0.6536	0.6741	112,6
apr-sep	8x8	0.6147	0.5015	0.7118	0.6335	0.6659	40,82
	12x12	0.6122	0.5186	0.7132	0.6396	0.6640	60,64
	16x16	0.6207	0.5157	0.7151	0.6502	0.6686	70,52

Table 4.1: Denoising performance (SSIM) for different filter sizes. For further detail, see Appendix A.

method [7]. The runtime performance of both methods is assessed for a fixed value of 200 iterations, employing the Conjugate Gradient (CG) method to solve the main linear system in both cases. These simulations were performed on an Intel Xeon E5-2640 CPU (2,50 GHz , 128Gb RAM, 2x NVidia Tesla K40m GPU). Our matlab code [31] can be used to reproduce our experimental results.

4.2 Experiments

In Table 4.1 we illustrate the results of the denoising comparisons (in terms of SSIM metric) between the 3 evaluated filter sets (each with 36 filters) for filter sizes 8×8, 12×12 and 16×16, and also report the average runtime for each method across the grid of λ values. It can be observed that the natively separable filters consistently outperform the approximated (separable) ones, and show equivalent performance to the standard non-separable filters. The runtime results also show that solving the CSC problem with separable filters is almost two times faster than doing it with non separable ones, which is consistent with the results reported in [2]. As an example, we illustrate in Figure 4.1 the entire set of denoising simulations across the λ grid for a single image. We also show in Figure 4.2 the reconstructed images for the optimal λ value across this grid.

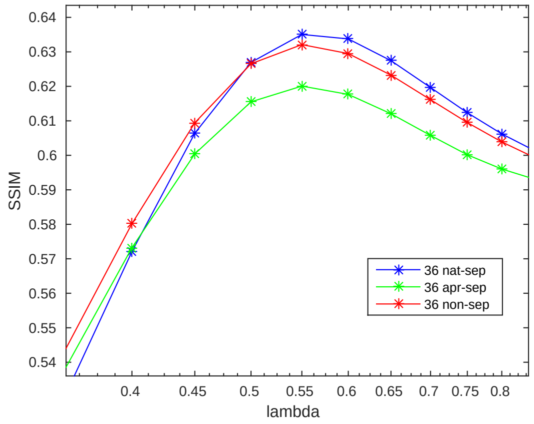
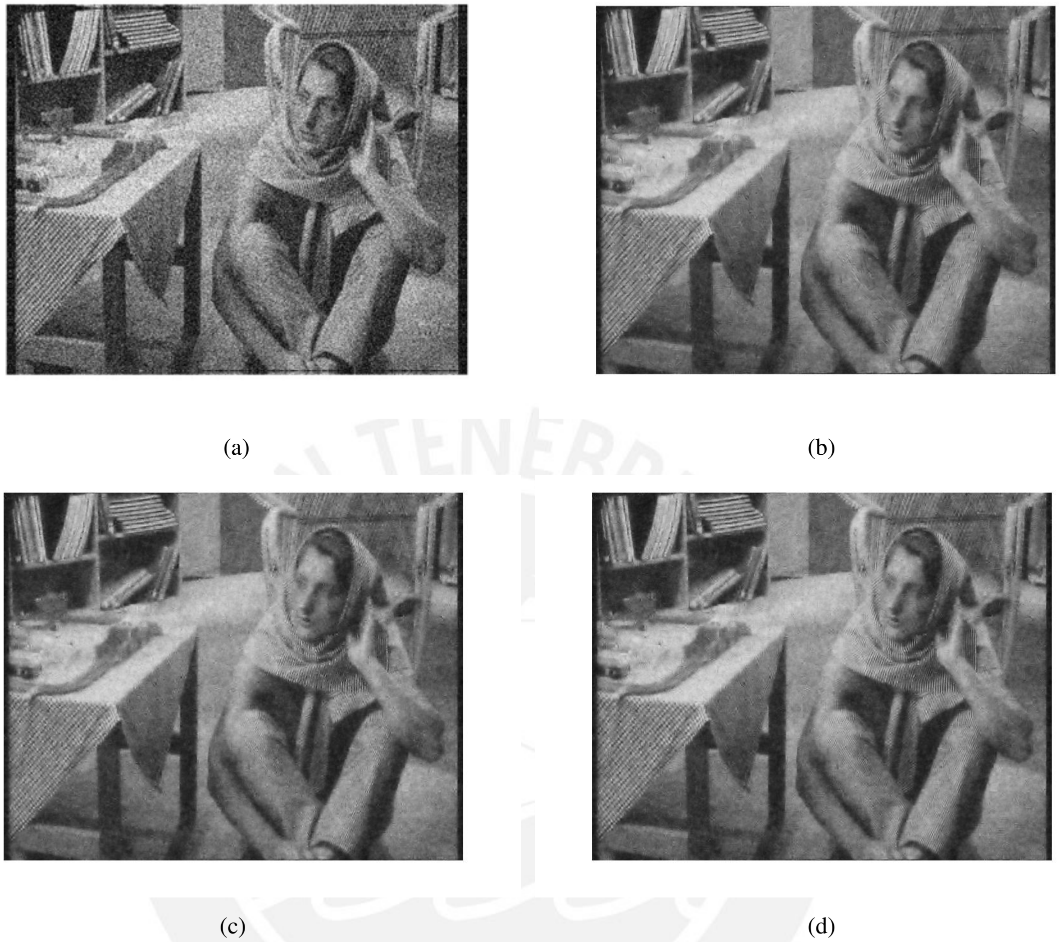


Figure 4.1: Denoising results on λ grid for 'barbara' image, where *apr-sep, nat-sep* and *non-sep* are the labels defined in Section 4.1


(a)

(b)

(c)

(d)

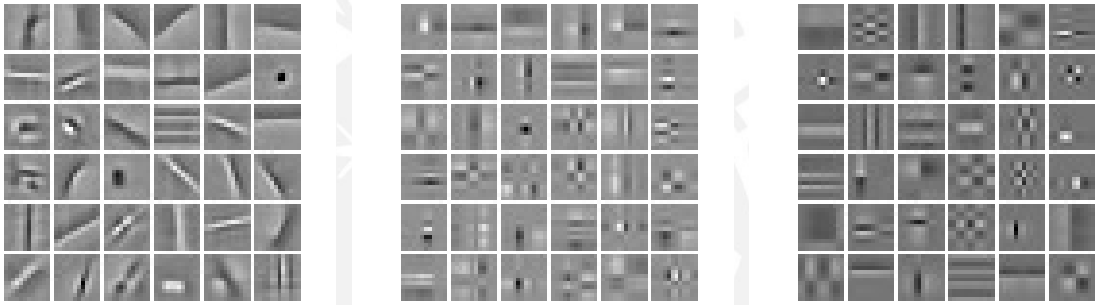
Figure 4.2: Example of denoising perfomance for each filter category with barbara image. (a): Original noisy image, (b): Reconstruction with non-sep filters with SSIM = 0.6330, (c) Reconstruction with apr-sep filters with SSIM = 0.6122, (d) Reconstruction with nat-sep filters with SSIM = 0.6370. Results taken at optimal value $\lambda = 0.55$

We report in Table 4.2 the results obtained in the inpainting simulations in terms of SSIM metric for each evaluated filter set. As in the previous case, it can be observed that our natively learned separable filters show no significant difference in performance with respect to the non-separable ones, and substantially outperform the approximated separable ones. Since the same solver from the SPORCO [30] library was used for all the considered filter categories, the runtime performance was approximately equivalent in the three scenarios and does not provide any relevant insight.

For illustrative purposes, we depict in Figure 4.3 an example set of each filter type for the size of 12×12. Interestingly, it can be observed that while the non-separable filter set is mostly composed of horizontal, vertical and diagonal edges, the separable ones consist of horizontal and vertical edges, as well as ’checkerboard’ paterns. It’s worth noting that for the case of the natively learned separable filters, this difference does not imply a decrease in reconstruction quality, as can be observed in Table 4.1 and Table 4.2.

	Dict. Size	barbara	mandrill	parrots	boats	goldhill
nat-sep	8x8	0.9432	0.8538	0.9719	0.9726	0.9311
	12x12	0.9501	0.8495	0.9725	0.975	0.932
	16x16	0.9445	0.8556	0.973	0.9746	0.9314
non-sep	8x8	0.9448	0.8607	0.974	0.9738	0.934
	12x12	0.9485	0.8596	0.9742	0.9742	0.9337
	16x16	0.9436	0.8645	0.9745	0.9741	0.9349
apr-sep	8x8	0.9354	0.8508	0.9684	0.9661	0.9273
	12x12	0.9338	0.8432	0.9685	0.9734	0.9278
	16x16	0.9277	0.8459	0.9706	0.9738	0.9288

Table 4.2: Inpainting performance (SSIM) for different filter sizes. For further detail, see Appendix A.



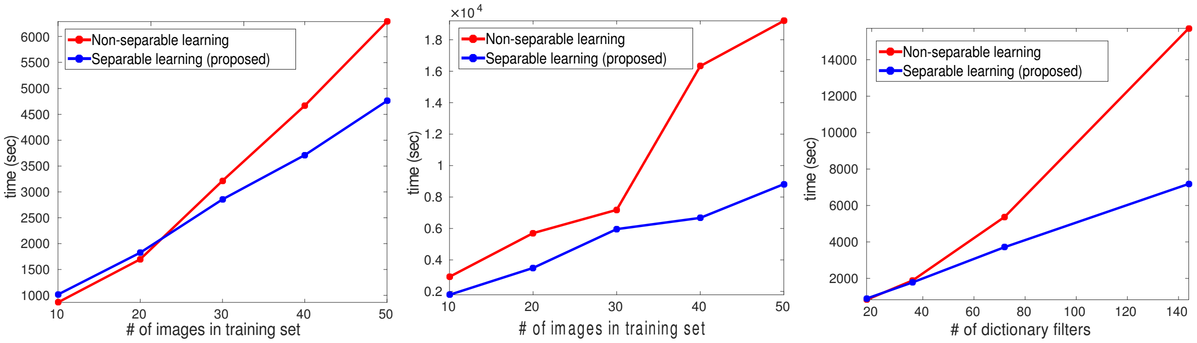
(a) Set of 36 non-separable filters of size 12×12

(b) Set of 36 approximated separable filters of size 12×12

(c) Set of 36 natively learned separable filters of size 12×12

Figure 4.3: Example of a set of 36 filters of each evaluated category

We report in Figure 4.4 (a) and (b) the computational performance comparisons in the learning process for two different CG tolerance values, in terms of runtime (seconds) vs image training set size. We consider a fixed number of 36 separable and non-separable filters for this simulation, and measure the runtime for both training methods for a fixed number of iterations (200). As can be observed in



(a) Tol=10^{-3}

(b) Tol=10^{-5}

(c) Time vs filter set size

Figure 4.4: Computational performance (separable vs. non-separable) results for CDL simulations, with termination at 200 iterations.

Figure 4.4 (a), when the CG tolerance is 10^{-3} the proposed separable method is slightly slower than its non-separable counterpart for small values of the number of images in the training set, and outperforms it when such value increases. When the tolerance value is 10^{-5}, the proposed method significantly outperforms the other one as S increases. Figure 2 (c) depicts a similar runtime comparison where the training set size is fixed ($S = 20$) and the dictionary size (number of filters) is varied (the tolerance value used is 10^{-3}). In this case it is also clear that the proposed method is substantially faster than the non-separable method as the number of filters increases.

An example of the functional value behaviour for a training set size of $S = 20$ and a target set of 36 filters of size 12×12 is shown in Figure 4.5 for 200 iterations. It can be seen from the graph that the proposed separable method converges to a slightly higher functional value than the non-separable method. This difference could be explained by the fact that the solution space of the separable filter learning problem is more constrained than that of its non-separable counterpart, however, this does not seem to have an impact on the performance quality of the learned separable filters, as can be seen on Tables 4.1 and 4.2.

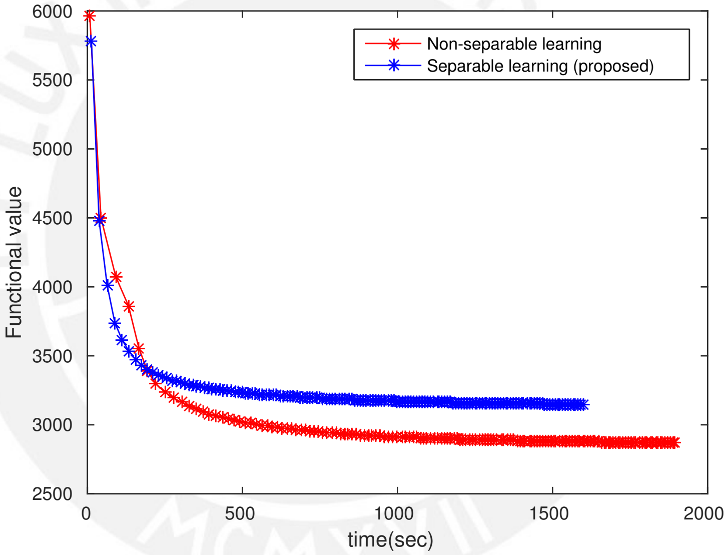


Figure 4.5: Functional value bahaviour comparison for CDL task

Chapter 5

Conclusions

A novel algorithm for natively learning separable filters without the need of a pre-learned set of non-separable ones has been presented. We formulated the proposed approach as an extension of the standard Convolutional Dictionary Learning (CDL) problem, and used the knowledge of existing dictionary learning techniques present in the literature as a base in the design of the method. The resulting approach has been published at the *2018 IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP 2018) [1].

Our computational simulations show that the separable filters learned through our method, when evaluated through denoising and inpainting Convolutional Sparse Coding (CSC) tasks, consistently outperform approximated separable filters, and attain the same reconstruction quality as when using standard non-separable filters. Moreover, the proposed separable learning method is more than 2 times faster than its non-separable counterpart when either the training set or the number of filters to estimate is large. These advantages constitute our learned separable filters into a competitive alternative to standard non-separable ones, which could in turn translate into significant speedups in several applications.

Recomendations

- Since computational performance is of paramount importance for most CDL applications, a potential path for further optimization of our approach would be to migrate our implementation to more flexible languages such as C, C++, or Fortran.
- Another approach to further optimize runtime performance would be to derive a CUDA C version of the proposed method, with explicit focus on the two computational bottlenecks: the ISM approach in the CSC stage and the CG method in the CDL stage.
- Although this thesis work has been focused on deriving the separable filter learning method from the most widely used non-separable approach, namely the ADMM framework, it would be interesting to explore the benefits of learning the separable filters by taking as a basis other common frameworks in the sparse coding literature, such as APG algorithms.

References

[1] J. Quesada, P. Rodríguez, and B. Wohlberg, "Separable dictionary learning for convolutional sparse coding via split updates," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4094–4098, April 2018.

[2] G. Silva, J. Quesada, P. Rodríguez, and B. Wohlberg, "Fast convolutional sparse coding with separable filters," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 6035–6039.

[3] R. Rigamonti, A. Sironi, V. Lepetit, and P. Fua, "Learning separable filters," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 2754–2761.

[4] S. Hawe, M. Seibert, and M. Kleinsteuber, "Separable dictionary learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 438–445.

[5] B. Ophir, "Multi scale dictionary learning for sparse representation of images," *Technion - Israel Institute of Technology, available from https://goo.gl/SGsrHr*, 2016.

[6] B. Wohlberg, "Convolutional sparse representations with gradient penalties," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Apr. 2018, pp. 6528–6532.

[7] B. Wohlberg, "Efficient algorithms for convolutional sparse representations," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 301–315, Jan. 2016.

[8] C. Garcia-Cardona and B. Wohlberg, "Convolutional Dictionary Learning," *ArXiv e-prints*, Sept. 2017.

[9] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions," in *Proceedings of the British Machine Vision Conference*. 2014, BMVA Press.

[10] A. Sironi, B. Tekin, R. Rigamonti, V. Lepetit, and P. Fua, "Learning separable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 94–106, Jan 2015.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[12] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[13] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, 1999, vol. 5, pp. 2443–2446 vol.5.

[14] M. Mørup and M. N. Schmidt, "Transformation invariant sparse coding," in *2011 IEEE International Workshop on Machine Learning for Signal Processing*, Sept 2011, pp. 1–6.

[15] M. D. Zeiler, G. W. Taylor, and R. Fergus, “Adaptive deconvolutional networks for mid and high level feature learning,” in *2011 International Conference on Computer Vision*, Nov 2011, pp. 2018–2025.

[16] Q. Barthelemy, A. Larue, A. Mayoue, D. Mercier, and J. I. Mars, “Shift amp; 2d rotation invariant sparse coding for multivariate signals,” *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 1597–1611, April 2012.

[17] R. Chalasani, J. C. Principe, and N. Ramakrishnan, “A fast proximal method for convolutional sparse coding,” in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Aug 2013, pp. 1–5.

[18] H. Bristow, A. Eriksson, and S. Lucey, “Fast convolutional sparse coding,” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 391–398.

[19] M. Ŝorel and F. Ŝroubek, “Fast convolutional sparse coding using matrix inversion lemma,” *Digital Signal Processing*, vol. 55, pp. 44 – 51, 2016.

[20] C. Garcia-Cardona and B. Wohlberg, “Subproblem coupling in convolutional dictionary learning,” in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Sept. 2017.

[21] K. Degraux, U. S. Kamilov, P. T. Boufounos, and D. Liu, “Online Convolutional Dictionary Learning for Multimodal Imaging,” *ArXiv e-prints*, June 2017.

[22] J. Liu, C. Garcia-Cardona, B. Wohlberg, and W. Yin, “Online Convolutional Dictionary Learning,” *ArXiv e-prints*, June 2017.

[23] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[24] Y. Nakatsukasa, T. Soma, and A. Uschmajew, “Finding a low-rank basis in a matrix subspace,” *Mathematical Programming*, vol. 162, no. 1, pp. 325–361, Mar 2017.

[25] P. Rodríguez, “Alternating optimization low-rank expansion algorithm to estimate a linear combination of separable filters to approximate 2d filter banks,” in *2016 50th Asilomar Conference on Signals, Systems and Computers*, Nov 2016, pp. 954–958.

[26] T. Blumensath and M. E. Davies, “Normalized iterative hard thresholding: Guaranteed stability and performance,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 298–309, 2010.

[27] L. Breiman, “Better subset regression using the nonnegative garrote,” vol. 37, pp. 373–384, 11 1995.

[28] N. Parikh and S. Boyd, *Proximal Algorithms*, Foundations and Trends in Optimization. Now Publishers, 2013.

[29] Gene H. Golub and Charles F. Van Loan, *Matrix Computations (3rd Ed.)*, Johns Hopkins University Press, 1996.

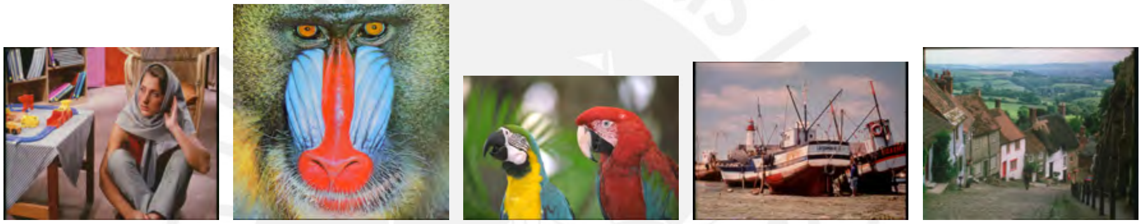
[30] B. Wohlberg, “Sparse optimization research code (SPORCO),” *Matlab library available from goo.gl/BjVgH5*, 2017.

[31] J. Quesada and P. Rodriguez, "Separable filter learning," Available at https://sites.google.com/pucp.pe/jquesada.

Appendices

Appendix A

Additional Figures and Tables



(a) barbara (b) mandrill (c) parrots (d) boats (e) goldhill

Figure A.1: Set of test images used for the denoising and inpainting simulations.

	Dict. Size	barbara	mandrill	parrots	boats	goldhill
nat-sep	8x8	22.87	20.86	27.36	23.76	24.60
	12x12	23.13	20.94	27.59	24.05	24.77
	16x16	23.09	20.92	27.50	24.07	24.83
non-sep	8x8	22.91	20.99	27.54	23.84	24.75
	12x12	23.13	21.08	27.70	24.01	24.99
	16x16	23.07	21.09	27.75	24.23	25.06
apr-sep	8x8	22.81	20.77	27.24	23.60	24.49
	12x12	22.93	20.87	27.40	23.84	24.73
	16x16	22.95	20.89	27.43	24.00	24.80

Table A.1: Denoising performance (PSNR) for different filter sizes

	Dict. Size	barbara	mandrill	parrots	boats	goldhill
nat-sep	8x8	9.37	6.12	12.60	11.50	14.77
	12x12	9.63	6.16	12.74	11.79	14.94
	16x16	9.59	6.18	12.74	11.81	15.00
non-sep	8x8	9.41	6.24	12.78	11.58	14.92
	12x12	9.60	6.34	12.96	11.75	15.16
	16x16	9.55	6.35	12.99	11.77	15.22
apr-sep	8x8	9.30	6.03	12.48	11.34	14.66
	12x12	9.43	6.13	12.68	11.65	14.90
	16x16	9.45	6.15	12.67	11.68	14.97

Table A.2: Denoising performance (SNR) for different filter sizes

	Dict. Size	barbara	mandrill	parrots	boats	goldhill
nat-sep	8x8	30.61	25.06	37.44	36.57	34.53
	12x12	31.57	24.99	38.02	37.89	34.74
	16x16	31.11	25.15	37.98	37.38	34.93
non-sep	8x8	31.04	25.29	38.82	36.80	35.05
	12x12	31.51	25.28	39.38	37.92	35.35
	16x16	31.01	25.37	39.41	37.45	35.44
apr-sep	8x8	30.14	24.91	36.55	34.88	33.94
	12x12	29.92	24.66	37.66	37.15	34.50
	16x16	30.17	24.80	37.88	37.03	34.69

Table A.3: Inpainting performance (PSNR) for different filter sizes

	Dict. Size	barbara	mandrill	parrots	boats	goldhill
nat-sep	8x8	17.10	10.31	22.68	24.31	24.70
	12x12	18.27	10.24	23.26	25.53	24.90
	16x16	17.50	10.40	23.23	25.12	25.09
non-sep	8x8	17.54	10.55	24.06	24.54	25.22
	12x12	18.21	10.54	24.62	25.65	25.51
	16x16	17.51	10.63	24.66	25.29	25.61
apr-sep	8x8	16.64	10.17	21.79	22.62	24.10
	12x12	16.42	9.92	22.90	24.90	24.66
	16x16	16.66	10.05	23.12	24.77	24.85

Table A.4: Inpainting performance (SNR) for different filter sizes