DCFNet: Deep Neural Network with Decomposed Convolutional Filters

Qiang Qiu [1] **Xiuyuan Cheng** [1] **Robert Calderbank** [1] **Guillermo Sapiro** [1]

Abstract

Filters in a Convolutional Neural Network (CNN) contain model parameters learned from enormous amounts of data. In this paper, we suggest to decompose convolutional filters in CNN as a truncated expansion with pre-fixed bases, namely the Decomposed Convolutional Filters network (DCFNet), where the expansion coefficients remain learned from data. Such a structure not only reduces the number of trainable parameters and computation, but also imposes filter regularity by bases truncation. Through extensive experiments, we consistently observe that DCFNet maintains accuracy for image classification tasks with a significant reduction of model parameters, particularly with Fourier-Bessel (FB) bases, and even with random bases. Theoretically, we analyze the representation stability of DCFNet with respect to input variations, and prove representation stability under generic assumptions on the expansion coefficients. The analysis is consistent with the empirical observations.

1. Introduction

Convolutional Neural Network (CNN) has become one of the most successful computational models in machine learning and artificial intelligence. Remarkable progress has been achieved in the design of successful CNN *network structures*, such as the VGG-Net (Simonyan & Zisserman, 2014), ResNet (He et al., 2016), and DenseNet (Huang et al., 2016). Less attention has been paid to the design of *filter structures* in CNNs. Filters, namely the weights in the convolutional layers, are one of the most important ingredients of a CNN model, as filters contain the actual model parameters learned from enormous amounts of data. Filters in CNNs are typically randomly initialized, and then updated using variants and extensions of gradient descent ("back-propagation").

[1]Duke University, Durham, North Carolina, USA. Work partially supported by NSF, DoD, NIH and AFOSR. Correspondence to: Xiuyuan Cheng <xiuyuan.cheng@duke.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

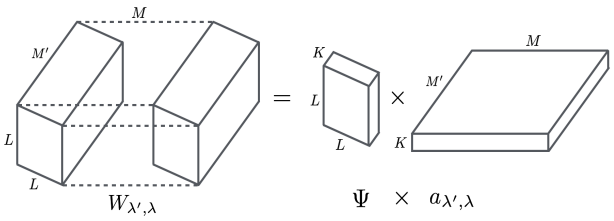


Figure 1. In a DCFNet, an $L \times L \times M' \times M$ convolutional layer is decomposed into the product of K bases of size $L \times L$ (Ψ) and $KM' \times M$ coefficients (a), where Ψ is pre-fixed, and a is learned from data. The basis can carry prior (explainable) structure if available.

As a result, trained CNN filters have no specific structures, which often leads to significant redundancy in the learned model (Denton et al., 2014; Han et al., 2015; Iandola et al., 2016). Filters with improved properties will have a direct impact on the accuracy and efficiency of CNN, and the theoretical analysis of filters is also of central importance to the mathematical understanding of deep networks.

This paper suggests to decompose convolutional filters in CNN into a truncated expansion with pre-fixed bases in the spatial domain, namely the Decomposed Convolutional Filters network (DCFNet), where the expansion coefficients remain learned from data. By representing the filters in terms of functional bases, which can come from prior data or task knowledge, rather than as pixel values, the number of trainable parameters is reduced to the expansion coefficients; and furthermore, regularity conditions can be imposed on the filters via the truncated expansion. For image classification tasks, we empirically observe that DCFNet is able to maintain the accuracy with a significant reduction in the number of parameters. Such observation holds even when random bases are used.

In particular, we adopt in DCFNet the leading Fourier-Bessel (FB) bases (Abramowitz & Stegun, 1964), which correspond to the low-frequency components in the input. We experimentally observe the superior performance of DCFNet with FB bases (DCF-FB) in both image classification and denoising tasks. DCF-FB network reduces the response to the high-frequency components in the input, which are least stable under image variations such as deformation and often do not affect recognition after being

suppressed. Such an intuition is further supported by a mathematical analysis of the CNN representation, where we firstly develop a general result for the CNN representation stability when the input image undergoes a deformation, under proper boundedness conditions of the convolutional filters (Propositions 3.1, 3.3, 3.4). After imposing the DCF structure, we show that as long as the trainable expansion coefficients at each layer of a DCF-FB network satisfy a boundedness condition, the L-th-layer output is stable with respect to input deformation and the difference is bounded by the magnitude of the distortion (Theorems 3.7, 3.8).

Apart from FB bases, the DCFNet structure studied in this paper is compatible with general choices of bases, such as standard Fourier bases, wavelet bases, random bases and PCA bases. We numerically test several options in Section 4. The stability analysis for DCF-FB networks can be extended to other bases choices as well, based upon the general theory developed for CNN representation and using similar techniques.

Our work is related to recent results on the topics of the usage of bases in deep networks, the model reduction of CNN, as well as the stability analysis of the deep representation. We review these connections in Section 1.1. Finally, though the current paper focuses on supervised networks for classification and recognition applications in image data, the introduced DCF layers are a generic concept and can potentially be used in reconstruction and generative models as well. We discuss possible extensions in the last section.

1.1. Related works

Deep network with bases and representation stability. The usage of bases in deep networks has been previously studied, including wavelet bases, PCA bases, learned dictionary atoms, etc. Wavelets are a powerful tool in signal processing (Mallat, 2008) and have been shown to be the optimal basis for data representation under generic settings (Donoho & Johnstone, 1994). As a pioneering mathematical model of CNN, the *scattering transform* (Mallat, 2012; Bruna & Mallat, 2013; Sifre & Mallat, 2013) used pre-fixed weights in the network which are wavelet filters, and showed that the representation produced by a scattering network is stable with respect to certain variations in the input. The extension of the scattering transform has been studied in (Wiatowski & Bölcskei, 2015; 2017) which includes a larger class of bases used in the network. Apart from wavelet, deep network with PCA bases has been studied in (Chan et al., 2015). Making a connection to dictionary learning (Aharon et al., 2006), (Papyan et al., 2016) studied deep networks in form of a cascade of convolutional sparse coding layers with theoretical analysis. Deep networks with random weights have been studied in (Giryes et al., 2016), with proved representation stability. The DCFNet studied in this paper incorporates structured pre-fixed bases combined by *adapted* expansion coefficients learned from data in a supervised way, and demonstrates comparable and even improved classification accuracy on image datasets. While the combination of fixed bases and learned coefficients has been studied in classical signal processing (Freeman et al., 1991; Mahalanobis et al., 1987), dictionary learning (Rubinstein et al., 2010) and computer vision (Henriques et al., 2013; Bertinetto et al., 2016), they were not designed with deep architectures in mind. Meanwhile, the representation stability of DCFNet is inherited thanks to the filter regularity imposed by the truncated bases decomposition.

Network redundancy. Various approaches have been studied to suppress redundancy in the weights of trained CNNs, including model compression and sparse connections. In model compression, network pruning has been studied in (Han et al., 2015) and combined with quantization and Huffman encoding in (Han et al., 2016). (Chen et al., 2015) used hash functions to reduce model size without sacrificing generalization performance. Low-rank compression of filters in CNN has been studied in (Denton et al., 2014; Ioannou et al., 2015). (Iandola et al., 2016; Lin et al., 2014) explored model compression with specific CNN architectures, e.g., replacing regular filters with 1×1 filters. Sparse connections in CNNs have been recently studied in (Ioannou et al., 2016; Anwar et al., 2017; Changpinyo et al., 2017). On the theoretical side, (Bölcskei et al., 2017) showed that a sparsely-connected network can achieve certain asymptotic statistical optimality. The proposed DCFNet relates model redundancy compression to the regularity conditions imposed on the filters. In DCF-FB network, redundancy reduction is achieved by suppressing network response to the high-frequency components in the inputs.

2. Decomposed Convolutional Filters

2.1. Notations of CNN

The output at the l-th layer of a convolutional neural network (CNN) can be written as $\{x^{(l)}(u,\lambda)\}_{u\in\mathbb{R}^2,\lambda\in[M_l]}$, where M_l is the number of channels in that layer and $[M] = \{1,\cdots,M\}$ for any integer M. A CNN with L layers can be written as a mapping from $\{x^{(0)}(u,\lambda)\}_{u\in\mathbb{R}^2,\lambda\in[M_0]}$ to $\{x^{(L)}(u,\lambda)\}_{u\in\mathbb{R}^2,\lambda\in[M_L]}$, recursively defined via $x^{(l)}(u,\lambda) = \sigma(x^{(l)}_{\frac{1}{2}}(u,\lambda) + b^{(l)}(\lambda))$, σ being the nonlinear mapping, e.g., ReLU, and

$$x^{(l)}_{\frac{1}{2}}(u,\lambda) = \sum_{\lambda'=1}^{M_{l-1}} \int W^{(l)}_{\lambda',\lambda}(v') x^{(l-1)}(u+v',\lambda') dv'. \quad (1)$$

The filters $W^{(l)}_{\lambda',\lambda}(u)$ and the biases $b^{(l)}$ are the parameters of the CNN. In practice, both $x^{(l)}(u,\lambda)$ and $W^{(l)}_{\lambda',\lambda}(u)$ are discretized on a Cartesian grid, and the continuous convolution

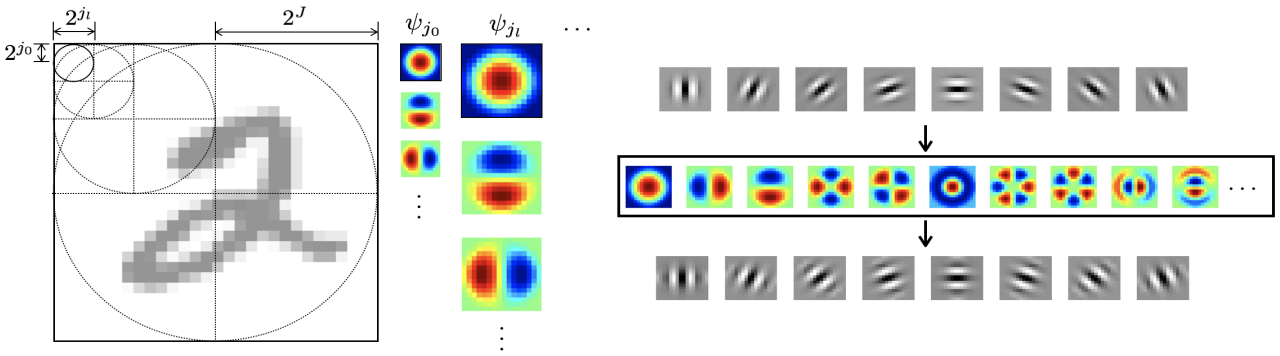


Figure 2. (Left) Multi-scale convolutional filters and Fourier-Bessel bases in various scales, $j_0 \leq \cdots \leq j_l \cdots \leq J$. (Right) $L \times L$ Gabor filters in 8 directions in size of, $L = 11$, and the approximation by K leading FB bases with a reduction rate of $\frac{K}{L^2} = \frac{1}{3}$. The truncation incurs almost no change to the filters. The leading FB bases are shown in the middle panel. Images rescaled for illustration purpose.

in (1) is approximated by its discrete analogue. Throughout the paper we use the continuous spatial variable u for simplicity. Very importantly, the filters $W^{(l)}_{\lambda',\lambda}(u)$ are locally supported, e.g., on 3×3 or 5×5 image patches.

2.2. Decomposition of convolutional filters

CNNs typically represent and store filters as vectors of the size of the local patches, which is equivalent to expanding the filters under the *delta bases*. Delta bases are not optimal for representing smooth functions. For example, regular functions have fast decaying coefficients under Fourier bases, and natural images have sparse representation under wavelet bases. DCF layers represent the convolutional filters as a truncated expansion under basis functions which are *non-adapted* through the training process, while adaption comes via the combination of such bases. Specifically, suppose that the convolutional filters $W_{\lambda',\lambda}(u)$ at certain layer, after a proper rescaling of the spatial variable (detailed in Section 3), are supported on the unit disk D in $\mathbb{R}^2$. Given a bases $\{\psi_k\}_k$ of the space $L^2(D)$, the filters can be represented as

$$W_{\lambda',\lambda}(u) = \sum_{k=1}^{K} (a_{\lambda',\lambda})_k \psi_k(u), \tag{2}$$

where K is the truncation. The decomposition (2) is illustrated in Figure 1, and conceptually, it can be viewed as a two-step scheme of a convolutional layer:

1. (Ψ-step) the input is convolved with each of the basis ψ_k, $k = 1, \cdots, K$, which are *pre-fixed*. The convolution for each input channel is independent from other channels, adding computational efficiency.

2. (a-step) the intermediate output is linearly transformed by an effectively fully-connected weight matrix $(a_{\lambda',\lambda})_k$ mapping from index (λ', k) to λ, which is *adapted* to data.

In (2), ψ_k can be any bases, and we numerically test on different choices in Section 4, including data-adapted bases and random bases. All experiments consistently show that the convolutional layers can be drastically decomposed and compressed with almost no reduction on the classification accuracy, and sometimes even using random bases gives strong performance. In particular, motivated by classical results of harmonic analysis, we use FB bases in DCFNet, with which the regularity of the filters $W_{\lambda',\lambda}$ can be imposed though constraining the magnitude the coefficients $\{(a_{\lambda',\lambda})_k\}_k$ (Proposition 3.6). As an example, Gabor filters approximated using the leading FB bases are plotted in the right of Figure 2. In experiments, DCFNet with FB bases shows superior performance in image classification and denoising tasks compared to original CNN and other bases being tested (Section 4). Theoretically, Section 3 analyzes the representation stability of DCFNet with respect to input variations, which provides a theoretical explanation of the advantage of FB bases.

2.3. Parameter and computation reduction

Suppose that the original convolutional layer is of size $L \times L \times M' \times M$, as shown in Figure 1, where typically $L = 3$, 5 and usually less than 11, M' and M grow from 3 (number of input channels) to a few hundreds in the deep layers in CNN. After switching to the DCFNet as in (2), there are $M' \times M \times K$ tunable parameters $(a_{\lambda',\lambda})_k$. Thus the number of parameters in that layer is a factor $\frac{K}{L^2}$ smaller, which can be significant if K is allowed to be small, particularly when M' and M are large.

The theoretical computational complexity can be calculated directly. Suppose that the input and output activation is $W \times W$ in spatial size, the original convolutional layer needs $M'W^2 \cdot M(1 + 2L^2)$ flops (the number of convolution operations is $M'M$, each take $2L^2W^2$ flops, and the summation over channels take an extra $W^2M'M$). In contract, a DCF layer takes $M'W^2 \cdot 2K(L^2 + M)$ flops, ($M'K$

many convolutions in the Ψ step, and $2KM'MW^2$ flops in the a step). Thus when $M \gg L^2$, the leading computation cost is $\frac{K}{L^2}$ of that of a regular CNN layer.

The reduction rate of $\frac{K}{L^2}$ in both model complexity and theoretical computational flops is confirmed on actual networks used in experiments, c.f. Table 3.

3. Analysis of Representation Stability

The analysis in this section is firstly done for regular CNN and then the conditions on filters are reduced to generic conditions on learnt coefficients in a DCF Net. In the latter, the proof is for the Fourier-Bessel (FB) bases, and can be extended to other bases using similar techniques.

3.1. Stable representation by CNN

We consider the spatial deformation operator denoted by D_τ, where $\tau : \mathbb{R}^2 \to \mathbb{R}^2$ and is C^2, $\rho(u) = u - \tau(u)$, and

$$D_\tau x(u, \lambda) = x(\rho(u), \lambda), \quad \forall u, \lambda.$$

We assume that the distortion is controlled, and specifically,

(A0) $|\nabla\tau|_\infty = \sup_u \|\nabla\tau(u)\| < \frac{1}{5}$, $\|\cdot\|$ being the operator norm.

The choice of the constant $\frac{1}{5}$ is purely technical. Thus ρ^{-1} exists, at least locally. Our goal is to control $\|x^{(L)}[D_\tau x^{(0)}] - x^{(L)}[x^{(0)}]\|$, namely when the input undergoes a deformation the output at L-the layer is not severely changed. We achieve this in two steps: (1) We show that $\|D_\tau x^{(L)}[x^{(0)}] - x^{(L)}[D_\tau x^{(0)}]\|$ is bounded by the magnitude of deformation up to a constant proportional to the norm of the signal, c.f. Proposition 3.3. (2) We show that $x^{(L)}$ is stable under D_τ when L is large, c.f. Proposition 3.4. To proceed, define the L^2 norm of $x(u, \lambda)$ to be

$$\|x\|^2 = \frac{1}{M} \sum_{\lambda \in [M]} \frac{1}{|\Omega|} \int_{\mathbb{R}^2} |x(u, \lambda)|^2 du, \tag{3}$$

where $|\Omega|^2 = (2 \cdot 2^J)^2$ is the area of the image-support domain, c.f. Figure 2. We assume that

(A1) $\sigma : \mathbb{R} \to \mathbb{R}$ is non-expansive,

which holds for ReLU. We also define the constants

$$B_l := \max\{\sup_\lambda \sum_{\lambda'=1}^{M_{l-1}} \|W^{(l)}_{\lambda',\lambda}\|_1, \sup_{\lambda'} \frac{M_{l-1}}{M_l} \sum_{\lambda=1}^{M_l} \|W^{(l)}_{\lambda',\lambda}\|_1\},$$

$$C_l := \max\{\sup_\lambda \sum_{\lambda'=1}^{M_{l-1}} \||v|\|\nabla W^{(l)}_{\lambda',\lambda}(v)|\|_1, \sup_{\lambda'} \frac{M_{l-1}}{M_l} \sum_{\lambda=1}^{M_l} \||v|\|\nabla W^{(l)}_{\lambda',\lambda}(v)|\|_1\}, \tag{4}$$

where $\||v|\|\nabla W(v)|\|_1$ denotes $\int_{\mathbb{R}^2} |v|\|\nabla W(v)|dv$.

Firstly, the following proposition shows that the layer-wise mapping is non-expansive whenever $B_l \leq 1$, the proof of which is left to Supplementary Material (S.M.).

Proposition 3.1. *In a CNN, under (A1), if $B_l \leq 1$ for all l,*

(a) The mapping of the l-th convolutional layer (including σ), denoted as $x^{(l)}[x^{(l-1)}]$, is non-expansive, i.e., $\|x^{(l)}[x_1] - x^{(l)}[x_2]\| \leq \|x_1 - x_2\|$ for arbitrary x_1 and x_2.

(b) $\|x^{(l)}_c\| \leq \|x^{(l-1)}_c\|$ for all l, where $x^{(l)}_c(u, \lambda) = x^{(l)}(u, \lambda) - x^{(l)}_0(\lambda)$ is the centered version of $x^{(l)}$, $x^{(l)}_0$ being the output at the l-th layer from a zero input at the bottom layer. As a result, $\|x^{(l)}_c\| \leq \|x^{(0)}_c\| = \|x^{(0)}\|$.

To switch the operator D_τ with the L-layer mapping $x^{(L)}[x^{(0)}]$, the idea is to control the residual of the switching at each layer, which is the following lemma proved in S.M..

Lemma 3.2. *In a CNN, under (A0) (A1), B_l, C_l as in (4),*

$$\|D_\tau x^{(l)}[x^{(l-1)}] - x^{(l)}[D_\tau x^{(l-1)}]\| \leq 4(B_l + C_l) \cdot |\nabla\tau|_\infty \|x^{(l-1)}_c\|,$$

where $x^{(l)}_c$ is as in Proposition 3.1.

We thus impose the assumption on the filters to be

(A2) For all l, B_l and C_l as in (4) are less than 1.

The assumption (A2) corresponds to a proper scaling of the convolutional filters so that the mapping in each convolutional layer is non-expansive (Proposition 3.1), and in practice, this can be qualitatively maintained by the standard normalization layers in CNN.

Now we can bound the residual of a L-layer switching to be additive as L increases:

Proposition 3.3. *In a CNN, under (A0), (A1), (A2),*

$$\|D_\tau x^{(L)}[x^{(0)}] - x^{(L)}[D_\tau x^{(0)}]\| \leq 8L|\nabla\tau|_\infty \|x^{(0)}\|. \tag{5}$$

Proof is left to S.M. We remark that it is possible to derive a more technical bound in terms of the constants B_l, C_l without assuming (A2), using the same technique. We present the simplified result here.

In the later analysis of DCF Net, (A2) will be implied by a single condition on the bases expansion coefficients, c.f. (A2').

To be able to control $\|D_\tau x^{(L)} - x^{(L)}\|$, we have the following proposition, proved in S.M.

Proposition 3.4. *In a CNN, under (A1),*

$$\|D_\tau x^{(l)} - x^{(l)}\| \leq 2|\tau|_\infty D_l \|x^{(l-1)}_c\|,$$

where $x_c^{(l)}$ *is as in Proposition 3.1, and* $D_l := \max\{\sup_\lambda \sum_{\lambda'=1}^{M_{l-1}} \|\nabla W_{\lambda',\lambda}^{(l)}\|_1, \sup_{\lambda'} \frac{M_{l-1}}{M_l} \sum_{\lambda=1}^{M_l} \|\nabla W_{\lambda',\lambda}^{(l)}\|_1\}$.

One may notice that $|\tau|_\infty$ is not proportional to $|\nabla\tau|_\infty$ when the deformation happens on a large domain, e.g., a rotation. It turns out that the multi-scale architecture of CNN induces a decrease of the quantity D_l proportional to the inverse of the domain diameter, which compensate the increase of $|\tau|_\infty$ as scale grows, as long as the rescaled filters are properly bounded in integral. Thus a unified deformation theory can be derived for DCFNets, see next section.

3.2. Multi-scale filters and Fourier Bessel (FB) bases

Due to the downsampling ("pooling") in CNN, the support of the l-th layer filters $W_{\lambda',\lambda}^{(l)}$ enlarges as l increases. Suppose that the input is supported on Ω which is a $(2\cdot 2^J)\times(2\cdot 2^J)$ domain, and the CNN has L layers. In accordance with the 2×2 pooling, we assume that $W_{\lambda',\lambda}^{(l)}$ is supported on $D(j_l)$, vanishing on the boundary, where $D(j)$ is a disk of radius 2^j, $j_0 \le \cdots \le j_L \le J$, and $D(j_0)$ is of size of patches at the smallest scale. Let $\{\psi_k\}_k$ be a set of bases supported on the unit disk $D(0)$, and we introduce the rescaled bases

$$\psi_{j,k}(u) = 2^{-2j}\psi_k(2^{-j}u), \quad u \in D(j),$$

where the normalization 2^{-2j} is introduced so that $\|\psi_{j,k}\|_1 = \|\psi_k\|_1$, where $\|f\|_1 := \int_{\mathbb{R}^2} |f(u)|du$. The multiscale filters and bases are illustrated in the left of Figure 2. By (2), we have that

$$W_{\lambda',\lambda}^{(l)}(u) = \sum_k (a_{\lambda',\lambda}^{(l)})_k \psi_{j_l,k}(u), \quad u \in D(j_l). \tag{6}$$

While DCFNet is compatible with general choices of bases, we focus on the FB bases in this section as an example. FB bases ψ_k are indexed by $k=(m,q)$ where m and q are the angular and radial frequencies respectively. They are supported on the unit disk $D = D(0)$, and in polar coordinates,

$$\psi_{m,q}(r,\theta) = c_{m,q}J_m(R_{m,q}r)e^{im\theta},\ r\in[0,1],\ \theta\in[0,2\pi],$$

where J_m is the Bessel function of the first kind, m are integers, $q = 1,2,\cdots$, $R_{m,q}$ is the q-th root of J_m, and $c_{m,q}$ is the normalizing constant s.t. $\langle\psi_{m,q},\psi_{m',q'}\rangle = \int_D \psi_{m,q}(u)\psi^*_{m',q'}(u)du = \pi\delta_{m,m'}\delta_{q,q'}$. Furthermore, FB bases are eigenfunctions of the Dirichlet Laplacian on D, i.e., $-\triangle\psi_k = \mu_k\psi_k$, where $\mu_{m,q} = R_{m,q}^2$. The eigenvalue μ_k grows as k increases (Weyl's law). Thus FB bases can be ordered by k so that μ_k increases, of which the leading few are shown in Table 1 and illustrated in Fig. 2. In principle, the frequency q and m should be truncated according to the Nyquist sampling rate. This truncation turned out to be not often used in our setting, due to the significant bases truncation in DCFNet.

k	1	2,3	4,5	6	7,8	9,10	11,12	13,14
m	0	1	2	0	3	1	4	2
q	1	1	1	2	1	2	1	2
μ_k	5.78	14.68	26.37	30.47	40.71	49.22	57.58	70.85

Table 1. The angular frequency m, radial frequency q and Dirichlet eigenvalue μ_k of the first 14 Fourier-Bessel bases. Two k corresponds to one pair of (m,q) when $m\neq 0$ due to that both real and complex parts of the bases are used as real-valued bases.

The key technical quantities in the stability analysis of CNN are $\|W_{\lambda',\lambda}^{(l)}\|_1$ and $\||v||\nabla W_{\lambda',\lambda}^{(l)}(v)|\|_1$, and with FB bases, these integrals are bounded by a μ_k-weighted L^2-norm of $a_{\lambda',\lambda}^{(l)}$ defined as $\|a\|_{FB} = (\sum_k \mu_k a_k^2)^{1/2}$ for all l. The following lemma and proposition are proved in S.M.

Lemma 3.5. *Suppose that* $\{\psi_k\}$ *are FB bases, the function* $F(u) = \sum_k a_k\psi_k(u)$ *is smooth on the unit disk. Then* $\frac{1}{\sqrt{\pi}}\|\nabla F\|_2 = \|a\|_{FB}$*, where* μ_k *are the eigenvalues of* ψ_k *as eigenfunctions of the negative Dirichlet laplacian on the unit disk. As a result,* $\|\nabla F\|_1 \le \pi\|a\|_{FB}$.

Proposition 3.6. *Using FB bases,* $\||v||\nabla W_{\lambda',\lambda}^{(l)}(v)|\|_1$ *and* $\|W_{\lambda',\lambda}^{(l)}\|_1$ *are bounded by* $\pi\|a_{\lambda',\lambda}^{(l)}\|_{FB}$ *for all* λ',λ *and* l.

Notice that the boundedness of $\|a\|_{FB}$ implies a decay of $|a_k|$ at least as fast as $\mu_k^{-1/2}$. This justifies the truncation of the FB expansion to the leading few bases, which correspond to the low-frequency modes.

Proposition 3.6 implies that B_l and C_l are all bounded by A_l defined as

$$A_l := \pi\max\{\sup_\lambda \sum_{\lambda'=1}^{M_{l-1}} \|a_{\lambda',\lambda}^{(l)}\|_{FB}, \sup_{\lambda'} \frac{M_{l-1}}{M_l}\sum_{\lambda=1}^{M_l}\|a_{\lambda',\lambda}^{(l)}\|_{FB}\}.$$

Then we introduce

(A2') For all l, $A_l \le 1$,

and the result of Proposition 3.3 extends to DCFNet:

Theorem 3.7. *In a DCFNet with FB bases, under (A0),(A1), (A2'), then*

$$\|D_\tau x^{(L)}[x^{(0)}] - x^{(L)}[D_\tau x^{(0)}]\| \le 8L|\nabla\tau|_\infty\|x^{(0)}\|.$$

Combined with Proposition 3.4, we have the following deformation stability bound, proved in S.M.:

Theorem 3.8. *In a DCFNet with FB bases, under (A0),(A1), (A2'),*

$$\|x^{(L)}[x^{(0)}] - x^{(L)}[D_\tau x^{(0)}]\| \le (8L|\nabla\tau|_\infty + 2\cdot 2^{-j_L}|\tau|_\infty)\|x^{(0)}\|. \tag{7}$$

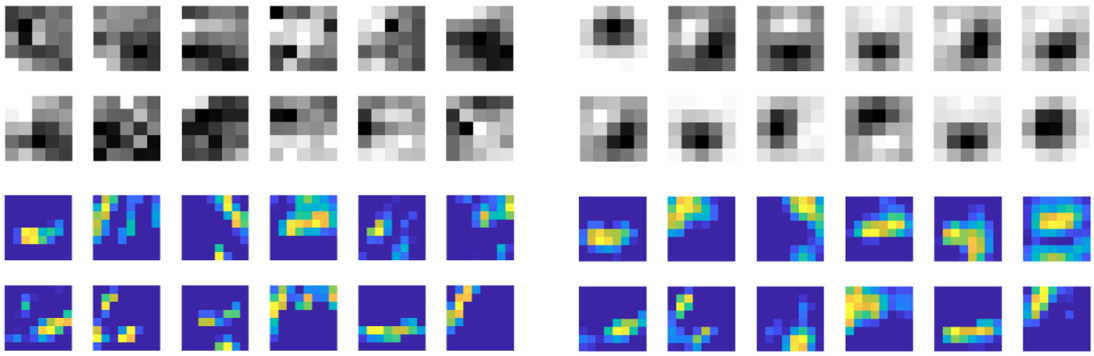


Figure 3. Example convolutional filters (upper) and network outputs (bottom) in the second layer of a Conv-2 net trained on MNIST (left) and the corresponding DCFNet using 3 FB bases (right). The filters in DCFNet are visibly smoother than those in the CNN, so are the network outputs. Classification accuracy of the two networks is comparable, c.f. Table 3.

4. Experiments

In this section, we experimentally demonstrate that convolutional filters in CNN can be decomposed as a truncated expansion with pre-fixed bases, where the expansion coefficients remain learned from data. Though the number of trainable parameters are significantly reduced, the accuracy in tasks such as image classification and face verification is still maintained. Such empirical observations hold for data-independent Fourier-Bessel (FB) and random bases, and data-dependent PCA bases.

4.1. Datasets

We perform an experimental evaluation on DCFNets using the following public datasets:

MNIST. 28×28 grayscale images of digits from 0 to 9, with 60,000 training and 10,000 testing samples.

SVHN. The Street View House Numbers (SVHN) dataset (Netzer et al., 2011) contains 32×32 colored images of digits 0 to 9, with 73,257 training and 26,032 testing samples. The additional training images were not used.

CIFAR10. The dataset (Krizhevsky, 2009) contains 32×32 colored images from 10 object classes, with 50,000 training and 10,000 testing samples.

Conv-2	Conv-3
c5x5x1x16 ReLu mp3x3	c5x5x3x64 ReLu mp3x3
c5x5x16x64 ReLu mp3x3	c5x5x64x128 ReLu mp3x3
fc128 ReLu fc10	c5x5x128x256 ReLu mp3x3
	fc512 ReLu fc10

Table 2. CNN network architectures used in MNIST, SVHN, and CIFAR10 experiments. cLxLxM'xM stands for a convolutional layer of patch size LxL and input (output) channel M' (M). mp$L$xL stands for LxL max-pooling. For the corresponding DCFNets, each LxLxM'xM CNN conv layer is expended over K $L \times L$ bases for trainable coefficients implemented as a $1 \times 1 \times M'K \times M$ conv layer.

VGG-Face. A large-scale face dataset, which contains about 2.6M face images from over 2.6K people (Parkhi et al., 2015). [1]

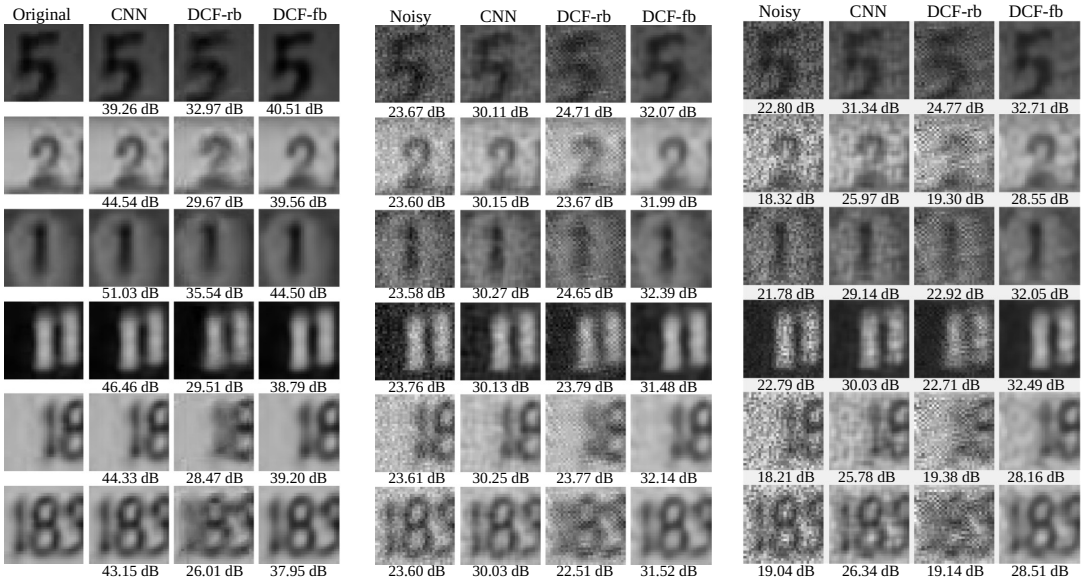
4.2. Object classification

In our object classification experiments, we evaluate the DCFNet with three types of predefined bases: Fourier-Bessel bases (DCF-FB), random bases which are generated by Gaussian vectors (DCF-RB), and PCA bases which are principal components of the convolutional filters in a pre-trained corresponding CNN model (DCF-PCA).

Three CNN network architectures are used for classification, Conv-2 and Conv-3 shown in Table 2, and VGG-16 (Simonyan & Zisserman, 2014). To generate the corresponding DCFNet structure from CNN, each CNN conv layer is expended over a set of pre-defined bases, and the obtained trainable expansion coefficients are implemented as a 1×1 conv layer. For example, a $5 \times 5 \times M' \times M$ conv layer is expended over K 5×5 bases for trainable coefficients in a $1 \times 1 \times M'K \times M$ convolutional layer. K denotes the number of basis used, and we evaluate multiple K for different levels of parameter reduction. In order to be compatible with existing deep learning frameworks, pre-fixed bases are currently implemented as regular convolutional layers with zero learning rate. The additional memory cost incurred in such convenient implementation can be eliminated with a more careful implementation, as bases are pre-fixed and the addition across channels can be computed on the fly.

The classification accuracy using DCFNets on various datasets are shown in Table 3. We observe that, by using only 3 Fourier-Bessel (FB) bases, we already obtain comparable accuracy as the original full CNN models on all

[1] The software is publicly available at https://github.com/xycheng/DCFNet.



(a) Original (b) Gaussian noise (c) Speckle noise

Figure 4. Examples (randomly selected) of image denoising on the SVHN dataset with PSNR values shown. The average PSNR over the entire test set including 26,032 samples: with Gaussian noise, 30.01 for CNN, 31.24 for DCF-fb; with Speckle noise, 28.15 for CNN, 29.84 for DCF-fb.

datasets, while using 12% parameters for 5×5 filters. When more FB bases are used, DCFNets outperform corresponding CNN models, still with significantly less parameters. As FB bases correspond to the low-frequency components in the inputs, DCF-FB network responds less to the high-frequency nuance details, which are often irrelevant for classification tasks. The superiority of DCF-FB network is further shown with less training data. For SVHN with 500 training samples, the testing accuracy (on a 50,000 testing set) of regular CNN and DCF-FB are 63.88% and 66.79% respectively. With 1000 training samples, the test accuracy are 73.53% v.s. 75.45%. Surprisingly, we observe that DCF with random bases also report acceptable performance.

Both the FB and random bases are data independent. For comparison purposes, we also evaluate DCFNets with data dependent PCA bases, which are principal components of corresponding convolutional filters in pre-trained CNN models. When the CNN model is pre-trained with all training data, PCA bases (pca-f) shows comparable performance as FB bases. However, the quality of the PCA bases (pca-s) degenerates, when only a randomly selected subset of the training set is used for the pre-training.

4.3. Image denoising

To gain intuitions behind the superior classification performance of DCFNet, we conduct a set of "toy" image denoising experiments on the SVHN image dataset. We take the first three 5×5 convolution blocks from the Conv-3 CNN network in Table 2, which is used in our SVHN object classification experiments. We remove all pooling layers, and append at the end an FC-256 followed with a Euclidean loss layer. We then decompose each 5×5 conv layer in this CNN network over 3 random bases and 3 FB bases respectively, to produce DCF-RB and DCF-FB networks.

We use SVHN training images with their gray-scale version as labels to train all three networks to simply reconstruct an input image (in gray-scale). Figure 4 shows how three trained networks behave while reconstructing examples from the SVHN testing images. Without noise added to input images, Figure 4a, all three networks report decent reconstruction, while DCF-RB shows inferior to both CNN and DCF-FB. PSNR values indicate CNN often produces more precise reconstructions; however, those missing high-frequency components in DCF-FB reconstructions are mostly nuance details. With noise added as in figures 4b and 4c, DCF-FB produces significantly superior reconstruction over both CNN and DCF-RB, with about one tenth of the parameter number of CNN.

The above empirical observations clearly indicate that Fourier-Bessel bases, which correspond to the low-frequency components in the inputs, enable DCF to ignore the high-frequency nuance details, which are often less stable under input variations, and mostly irrelevant for tasks such as classification. Such empirical observation provides good intuitions behind the superior classification performance of DCF, and is also consistent with the theoretical analysis on representation stability in Section 3.

MNIST conv-2, 5x5						
	fb	rb	pca-s	pca-f	# param.	# MFlops
CNN	99.40				2.61×10^4	3.37
K=14	99.47	99.35	99.38	99.41	1.46×10^4	2.40
K=8	99.48	99.26	99.28	99.45	8.40×10^3	1.37
K=5	99.39	99.28	99.28	99.43	5.28×10^3	0.86
K=3	99.40	98.69	99.19	99.35	3.20×10^3	0.51
SVHN conv-3, 5x5						
	fb	rb	pca-s	pca-f	# param.	# MFlops
CNN	94.22				1.03×10^6	201.64
K=14	94.63	93.75	94.52	94.42	5.74×10^5	121.91
K=8	94.39	92.05	93.85	94.30	3.30×10^5	69.67
K=5	93.93	91.28	92.34	94.03	2.06×10^5	43.55
K=3	92.84	88.47	91.88	93.10	1.24×10^5	26.13
Cifar10 conv-3, 5x5						
	fb	rb	pca-s	pca-f	# param.	# MFlops
CNN	85.66					
K=14	85.88	84.76	85.27	85.34	(same as above)	
K=8	85.30	81.27	84.70	85.09		
K=5	84.35	77.96	83.12	83.94		
K=3	83.12	74.05	80.94	82.91		
Cifar10 vgg-16, 3x3						
	fb	rb	pca-s	pca-f	# param.	# MFlops
CNN	87.02				1.47×10^7	547.20
K=5	87.79	84.16	87.98	87.60	8.18×10^6	311.68
K=3	88.21	78.46	87.45	87.54	4.91×10^6	187.02

Table 3. Classification accuracy using DCFNets on various image benchmarks with different number of bases K. "fb" and "rb" stand for Fourier-Bessel bases and random bases respectively. "pca-s" and "pca-f" stand for PCA bases computed from a network pre-trained on a small subset of training images (1,000 random samples) and the full training set respectively. "# param." is number of parameters in all convolutional layers, and MFlops is the number of flops in all convolutional layers (including ReLU).

4.4. Face verification

We present a further evaluation of DCFNet on face verification tasks using "very deep" network architectures, which comprise a long sequence of convolutional layers. In order to train such complex networks, we adopt a very large scale *VGG-face* (Parkhi et al., 2015) dataset, which contains about 2.6M face images from over 2.6K people.

As shown in Table 4, we adopt the VGG-Very-Deep-16 CNN architecture as detailed in (Parkhi et al., 2015) by modifying layer 32 and 35 to change output features from 4,096 dimension to 512. Such CNN network comprises 16 weight layers, and all except the last Fully-Connected (FC) layer utilize 3×3 or 5×5 filters.

The input to both CNN and DCFNet are face images of size 224×224 (with the average face image subtracted). As shown in Table 5, with FB bases, even only using $\frac{1}{3}$ parameters at weight layers ($K = 3$ for 3×3, $K = 8$ for 5×5), the DCFNet shows similar verification accuracy as the CNN structure on the challenging LFW benchmark. Note that our CNN model outperforms the *VGG-face* model in (Parkhi et al., 2015), and such improvement is mostly due to the smaller output dimension we adopted, as both models share similar architecture and are trained on the same face dataset.

Layer	CNN	DCFNet
1	conv $3 \times 3 \times 3 \times 64$	3 3×3 basis conv $1 \times 1 \times 9 \times 64$
2	ReLu	
3	conv $3 \times 3 \times 64 \times 64$	3 3×3 basis conv $1 \times 1 \times 192 \times 64$
4-5	ReLu, maxPool 2×2	
6	conv $3 \times 3 \times 64 \times 128$	3 3×3 basis conv $1 \times 1 \times 192 \times 128$
7	ReLu	
8	conv $3 \times 3 \times 128 \times 128$	3 3×3 basis conv $1 \times 1 \times 384 \times 128$
9-10	ReLu, maxPool 2×2	
(1-31 CNN layers are identical to *vgg-face* model in (Parkhi et al., 2015).)		
32	conv $5 \times 5 \times 512 \times 512$	8 5×5 basis conv $1 \times 1 \times 4096 \times 512$
33-34	ReLu, dropout	
35	conv $3 \times 3 \times 512 \times 512$	3 3×3 basis conv $1 \times 1 \times 1536 \times 512$
36-39	ReLu, dropout, FC, softmax	

Table 4. Network architecture for face experiments. For the corresponding DCFNet, each LxLxM'xM CNN conv layer is expended over K $L \times L$ bases for trainable coefficients implemented as a $1 \times 1 \times M'K \times M$ conv layer ($K = 3$ for 3×3, $K = 8$ for 5×5).

	Accuracy	# param.	# GFlops
VGG-face	97.27 %	-	-
CNN	97.65 %	21.26×10^6	30.05
DCFNet	97.32 %	7.01×10^6	10.09

Table 5. Face verification accuracy on the LFW benchmark.

5. Conclusion and Discussion

The paper studies CNNs where the convolutional filters are represented as a truncated expansion under pre-fixed bases and the expansion coefficients are learned from labeled data. Experimentally, we observe that on various object recognition datasets the classification accuracy are maintained with a significant reduction of the number of parameters, and the performance of Fourier-Bessel (FB) bases is constantly superior. The truncated FB expansion in DCFNet can be viewed as a regularization of the filters. In other words, DCF-FB is less susceptible to the high-frequency components in the input, which are least stable under expected input variations and often do not affect recognition when suppressed. This interpretation is supported by image denoising experiments, where DCF-FB performs preferably over the original CNN and other basis options on noisy inputs. The stability of DCFNet representation is also proved theoretically, showing that the perturbation of the deep features with respect to input variations can be bounded under generic conditions on the decomposed filters.

To extend the work, firstly, DCF layers can be incorporated in networks for unsupervised learning, for which the denoising experiment serves as a first step. The stability analysis can be extended by testing the resilience to adversarial noise. Finally, more structures may be imposed across the channels, concurrently with the structures of the filters in space.

References

Abramowitz, Milton and Stegun, Irene A. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Courier Corporation, 1964.

Aharon, Michal, Elad, Michael, and Bruckstein, Alfred. rmk-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

Anwar, Sajid, Hwang, Kyuyeon, and Sung, Wonyong. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3):32, 2017.

Bertinetto, Luca, Valmadre, Jack, Golodetz, Stuart, Miksik, Ondrej, and Torr, Philip HS. Staple: Complementary learners for real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1401–1409, 2016.

Bölcskei, Helmut, Grohs, Philipp, Kutyniok, Gitta, and Petersen, Philipp. Optimal approximation with sparsely connected deep neural networks. *arXiv preprint arXiv:1705.01714*, 2017.

Bruna, Joan and Mallat, Stéphane. Invariant scattering convolution networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1872–1886, 2013.

Chan, Tsung-Han, Jia, Kui, Gao, Shenghua, Lu, Jiwen, Zeng, Zinan, and Ma, Yi. Pcanet: A simple deep learning baseline for image classification? *IEEE Transactions on Image Processing*, 24(12):5017–5032, 2015.

Changpinyo, Soravit, Sandler, Mark, and Zhmoginov, Andrey. The power of sparsity in convolutional neural networks. *arXiv preprint arXiv:1702.06257*, 2017.

Chen, Wenlin, Wilson, James, Tyree, Stephen, Weinberger, Kilian, and Chen, Yixin. Compressing neural networks with the hashing trick. In *International Conference on Machine Learning*, pp. 2285–2294, 2015.

Denton, Emily L., Zaremba, Wojciech, Bruna, Joan, LeCun, Yann, and Fergus, Rob. Exploiting linear structure within convolutional networks for efficient evaluation. In *NIPS*, pp. 1269–1277, 2014.

Donoho, David L and Johnstone, Jain M. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3):425–455, 1994.

Freeman, William T, Adelson, Edward H, et al. The design and use of filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991.

Giryes, Raja, Sapiro, Guillermo, and Bronstein, Alexander M. Deep neural networks with random gaussian weights: a universal classification strategy? *IEEE Trans. Signal Processing*, 64(13):3444–3457, 2016.

Han, Song, Pool, Jeff, Tran, John, and Dally, William. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, pp. 1135–1143, 2015.

Han, Song, Mao, Huizi, and Dally, William J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *International Conference on Learning Representations (ICLR)*, 2016.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Henriques, Joao F, Carreira, Joao, Caseiro, Rui, and Batista, Jorge. Beyond hard negative mining: Efficient detector learning via block-circulant decomposition. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 2760–2767. IEEE, 2013.

Huang, Gao, Liu, Zhuang, Weinberger, Kilian Q, and van der Maaten, Laurens. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

Iandola, Forrest N., Han, Song, Moskewicz, Matthew W., Ashraf, Khalid, Dally, William J., and Keutzer, Kurt. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. *arXiv:1602.07360*, 2016.

Ioannou, Yani, Robertson, Duncan, Shotton, Jamie, Cipolla, Roberto, and Criminisi, Antonio. Training cnns with low-rank filters for efficient image classification. *arXiv preprint arXiv:1511.06744*, 2015.

Ioannou, Yani, Robertson, Duncan, Cipolla, Roberto, and Criminisi, Antonio. Deep roots: Improving cnn efficiency with hierarchical filter groups. *arXiv preprint arXiv:1605.06489*, 2016.

Krizhevsky, Alex. Learning multiple layers of features from tiny images. Technical report, 2009.

Lin, Min, Chen, Qiang, and Yan, Shuicheng. Network In Network. *ICLR*, 2014.

Mahalanobis, Abhijit, Kumar, BVK Vijaya, and Casasent, David. Minimum average correlation energy filters. *Applied Optics*, 26(17):3633–3640, 1987.

Mallat, Stephane. *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.

Mallat, Stéphane. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

Netzer, Yuval, Wang, Tao, Coates, Adam, Bissacco, Alessandro, Wu, Bo, and Ng, Andrew Y. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

Papyan, Vardan, Romano, Yaniv, and Elad, Michael. Convolutional neural networks analyzed via convolutional sparse coding. *stat*, 1050:27, 2016.

Parkhi, O. M., Vedaldi, A., and Zisserman, A. Deep face recognition. In *British Machine Vision Conference*, 2015.

Rubinstein, Ron, Zibulevsky, Michael, and Elad, Michael. Double sparsity: Learning sparse dictionaries for sparse signal approximation. *IEEE Transactions on signal processing*, 58(3):1553–1564, 2010.

Sifre, Laurent and Mallat, Stéphane. Rotation, scaling and deformation invariant scattering for texture discrimination. In *CVPR*, pp. 1233–1240, 2013.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Wiatowski, Thomas and Bölcskei, Helmut. Deep convolutional neural networks based on semi-discrete frames. In *Information Theory (ISIT), 2015 IEEE International Symposium on*, pp. 1212–1216. IEEE, 2015.

Wiatowski, Thomas and Bölcskei, Helmut. A mathematical theory of deep convolutional neural networks for feature extraction. *IEEE Transactions on Information Theory*, 2017.

A. Proofs

In the proofs, some technical details are omitted for brevity and readability. The full proofs are left to the long version of the work.

Proof of Proposition 3.1. To prove (a), omitting (l) in $W^{(l)}$, and let $M = M_l$, $M' = M_{l-1}$, $B_{\lambda',\lambda} = \|W_{\lambda',\lambda}\|_1$. By definition of B_l, we have that

$$\begin{aligned} \sum_{\lambda' \in [M']} B_{\lambda',\lambda} &\leq B_l, \quad \forall \lambda \\ \sum_{\lambda \in [M]} B_{\lambda',\lambda} &\leq B_l \frac{M}{M'}, \quad \forall \lambda'. \end{aligned} \tag{A1}$$

We essentially use Schur's test, being more careful with the summation over λ'. We derive by Cauchy-Schwarz which is equivalent to Schur's test:

$$\begin{aligned} &\|x^{(l)}[x_1] - x^{(l)}[x_2]\|^2 \cdot |\Omega| M \\ =& \sum_{\lambda \in [M]} \int \left| \sigma\Big(\sum_{\lambda' \in [M']} \int x_1(u+v', \lambda') W_{\lambda',\lambda}(v') dv' + b(\lambda) \Big) - \sigma\Big(\sum_{\lambda' \in [M']} \int x_2(u+v', \lambda') W_{\lambda',\lambda}(v') dv' + b(\lambda) \Big) \right|^2 du \\ \leq& \sum_{\lambda \in [M]} \int \left| \sum_{\lambda' \in [M']} \int x_1(u+v', \lambda') W_{\lambda',\lambda}(v') dv' - \sum_{\lambda' \in [M']} \int x_2(u+v', \lambda') W_{\lambda',\lambda}(v') dv' \right|^2 du \\ =& \sum_{\lambda \in [M]} \int \left| \sum_{\lambda' \in [M']} \int (x_1 - x_2)(\tilde{v}, \lambda') W_{\lambda',\lambda}(\tilde{v} - u) d\tilde{v} \right|^2 du \\ \leq& \sum_{\lambda \in [M]} \int \left(\sum_{\lambda'_1 \in [M']} \int |(x_1 - x_2)(v_1, \lambda'_1)|^2 \left| W_{\lambda'_1,\lambda}(v_1 - u) \right| dv_1 \right) \cdot \left(\sum_{\lambda'_2 \in [M']} \|W_{\lambda'_2,\lambda}\|_1 \right) du \\ \leq& B_l \cdot \sum_{\lambda'_1 \in [M']} \int |(x_1 - x_2)(v_1, \lambda'_1)|^2 \left(\sum_{\lambda \in [M]} \|W_{\lambda'_1,\lambda}\|_1 \right) dv_1 \\ \leq& B_l \cdot B_l \frac{M}{M'} \cdot \|x_1 - x_2\|^2 |\Omega| M' = B_l^2 M \|x_1 - x_2\|^2 |\Omega|, \end{aligned}$$

which means that

$$\|x^{(l)}[x_1] - x^{(l)}[x_2]\| \leq B_l \|x_1 - x_2\|.$$

Thus $B_l \leq 1$ implies (a).

To prove (b), we firstly verify that $x_0^{(l)}(\lambda)$ indeed is a constant over space for all λ and l. When $l = 0$, $x_0^{(0)}$ is all zero, so the claim is true. Suppose that the claim holds for $l - 1$, then

$$x_0^{(l)}(u, \lambda) = \sigma\left(\sum_{\lambda'} \int x_0^{(l-1)}(\lambda') W_{\lambda',\lambda}^{(l)}(v') dv' + b^{(l)}(\lambda) \right)$$

which again does not depend on u. So we can write $x_0^{(l)}$ as $x_0^{(l)}(\lambda)$. Now by (a),

$$\|x_c^{(l)}\| = \|x^{(l)}[x^{(l-1)}] - x^{(l)}[x_0^{(l-1)}]\| \leq \|x^{(l-1)} - x_0^{(l-1)}\| = \|x_c^{(l-1)}\|,$$

which proves (b). □

Proof of Lemma 3.2. To illustrate the idea, we first prove the lemma in the one-dimensional case, i.e. $u \in \mathbb{R}$ instead of $\mathbb{R}^2$. We then extend to the 2D case. In the 1D case, the constant c_1 can be improved to be 2, and we only need $|\tau'|_\infty < \frac{1}{2}$. In the 2D case, we need $c_1 = 4$ as in the final claim.

To simply notation, we denote the mapping $x^{(l)}[x^{(l-1)}]$ as $y[x]$, $x_c^{(l-1)}$ by x_c, $M_{l-1} = M'$, $M_l = M$, and $W^{(l)}$ by W. Let $C_{\lambda',\lambda} = \int |v||\frac{d}{dv}W_{\lambda',\lambda}(v)|dv$, and $B_{\lambda',\lambda} = \int |W_{\lambda',\lambda}(v)|dv$, then (A1) holds, and the same relation holds for $C_{\lambda',\lambda}$ and C_l.

By definition,

$$D_\tau y[x](u,\lambda) = \sigma\left(\sum_{\lambda' \in [M']} \int x(\rho(u)+v',\lambda')W_{\lambda',\lambda}(v')dv' + b(\lambda)\right),$$
$$y[D_\tau x](u,\lambda) = \sigma\left(\sum_{\lambda' \in [M']} \int x(\rho(u+v'),\lambda')W_{\lambda',\lambda}(v')dv' + b(\lambda)\right).$$

Relaxing by removing σ as in the proof of Proposition 3.1, one can derive that

$$\|D_\tau y[x] - y[D_\tau x]\|^2 \cdot |\Omega|M \le \|E_1 + E_2\|^2,$$

where

$$E_1(u,\lambda) = \sum_{\lambda' \in [M']} \int x_c(v,\lambda')(W_{\lambda',\lambda}(v-\rho(u)) - W_{\lambda',\lambda}(\rho^{-1}(v)-u))dv,$$
$$E_2(u,\lambda) = \sum_{\lambda' \in [M']} \int x_c(v,\lambda')W_{\lambda',\lambda}(\rho^{-1}(v)-u)(|(\rho^{-1})'(v)|-1)dv.$$

Notice that x is replaced by x_c due to the fact that x and x_c differ by a constant field over space for each channel λ'. We bound $\|E_1\|$ and $\|E_2\|$ respectively.

For E_1, we introduce $k^{(1)}_{\lambda',\lambda}(v,u) = W_{\lambda',\lambda}(v-\rho(u)) - W_{\lambda',\lambda}(\rho^{-1}(v)-u)$, and re-write it as

$$E_1(u,\lambda) = \sum_{\lambda' \in [M']} \int x_c(v,\lambda')k^{(1)}_{\lambda',\lambda}(v,u)dv.$$

Applying Schur's test as in the proof of Proposition 3.1, one can show that

$$\|E_1\| \le 2|\tau'|_\infty C_l \sqrt{M|\Omega|}\|x_c\|$$

as long as for all λ', λ,

$$\sup_u \int \left|k^{(1)}_{\lambda',\lambda}(v,u)\right| dv, \sup_v \int \left|k^{(1)}_{\lambda',\lambda}(v,u)\right| du \le 2C_{\lambda',\lambda}|\tau'|_\infty. \tag{A2}$$

(A2) can be verified by 1D change of variable, and details omitted.

For E_2, we introduce $k^{(2)}_{\lambda',\lambda}(v,u) = W_{\lambda',\lambda}(\rho^{-1}(v)-u)(|(\rho^{-1})'(v)|-1)$, and then we have that

$$\int |k^{(2)}_{\lambda',\lambda}(v,u)|du \le |(\rho^{-1})'(v)-1| \cdot \int |W_{\lambda',\lambda}(u)|du \le 2|\tau'|_\infty B_{\lambda',\lambda}, \quad \forall v,$$

where we use $1-(\rho^{-1})'(t) = \frac{-\tau'(\rho^{-1}(t))}{1-\tau'(\rho^{-1}(t))}$ and $|\tau'| < \frac{1}{2}$ to obtain the factor 2. Meanwhile,

$$\int |k^{(2)}_{\lambda',\lambda}(v,u)|dv = \int |W_{\lambda',\lambda}(\tilde{v}-u)||1-|\rho'(\tilde{v})||d\tilde{v} \le |\tau'|_\infty B_{\lambda',\lambda}, \quad \forall u.$$

This gives that

$$\|E_2\| \le 2|\tau'|_\infty B_l \sqrt{M|\Omega|}\|x_c\|.$$

Putting together we have that

$$\sqrt{M|\Omega|}\|D_\tau y[x] - y[D_\tau x]\| \le \|E_1 + E_2\| \le \|E_1\| + \|E_2\| \le 2|\tau'|_\infty (C_l + B_l)\sqrt{M|\Omega|}\|x_c\|$$

which proves the claim in the 1D case.

The extension to the 2D case uses standard elementary techniques. The assumption $|\nabla\tau|_\infty < \frac{1}{5}$ is used to derive that $||J\rho| - 1|$, $||J\rho^{-1}| - 1| \le 4|\nabla\tau|_\infty$, and $|J\rho|$, $|J\rho^{-1}| \le 2$. In all the formula, $|(\rho^{-1})'(v)|$ is replaced by the Jacobian determinant $|J\rho^{-1}(v)|$. The integration in 1D is replaced by that along a segment in the 2D space. Details omitted. □

Proof of Prop. 3.3. Under these conditions, Proposition 3.1 applies. Let $c_1 = 4$. Introduce the notation

$$y_l = x^{(L)} \circ \cdots \circ D_\tau x^{(l)} \circ \cdots \circ x^{(0)}, \quad l = 0, \cdots, L$$

where $y_0 = x^{(L)}[D_\tau x^{(0)}]$, and $y_L = D_\tau x^{(L)}[x^{(0)}]$. The l.h.s equals $\|y_0 - y_L\|$, and we will bound it by $\|y_L - y_0\| \le \sum_{l=1}^{L} \|y_l - y_{l-1}\|$. For each $l = 1, \cdots, L$,

$$\begin{aligned}
\|y_l - y_{l-1}\| =& \|x^{(L)} \circ \cdots \circ D_\tau x^{(l)} \circ x^{(l-1)} \\
&- x^{(L)} \circ \cdots \circ x^{(l)} \circ D_\tau x^{(l-1)}\| \\
\le& \|D_\tau x^{(l)} \circ x^{(l-1)} - x^{(l)} \circ D_\tau x^{(l-1)}\| \\
\le& c_1 (C_l + B_l) |\nabla\tau|_\infty \|x_c^{(l-1)}\| \\
\le& 2c_1 |\nabla\tau|_\infty \|x_c^{(l-1)}\| \\
\le& 2c_1 |\nabla\tau|_\infty \|x^{(0)}\|,
\end{aligned}$$

where the first inequality is by the nonexpansiveness of the $(l+1)$ to L-th layer, the second by Lemma 3.2, the third by (A2), and the last by Proposition 3.1 (b). Thus, $\sum_{l=1}^{L} \|y_l - y_{l-1}\| \le 2c_1 L |\nabla\tau|_\infty \|x^{(0)}\|$. □

Proof of Proposition 3.4. The technique is similar to that in the proof of Lemma 3.2. Let the constant on the r.h.s be denoted by c_2. In the 1D case, the constant c_2 can be improved to be 1. In the 2D case, $c_2 = 2$ as in the final claim. Details omitted. □

Proof of Lemma 3.5. The first claim is a classical result, and has a direct proof as $\int_{D(0)} |\nabla F|^2 = -\int_{D(0)} F\Delta F = \langle \sum_k a_k \psi_k, \sum_k a_k \mu_k \psi_k \rangle = \pi \sum_k a_k^2 \mu_k$ by the orthogonality of ψ_k, as stated above in the text. By Cauchy-Schwarz, $\|\nabla F\|_1 \le \sqrt{\pi}\|\nabla F\|_2$. Putting together gives the second claim. □

Proof of Proposition 3.6. Omitting λ', λ, l, and let $j_l = j$, we write $W(u) = \sum_k a_k \psi_{j,k}(u)$. Rescaled to $D(0)$, we consider $w(u) = \sum_k a_k \psi_k(u)$, and one can verify that $\||v|\nabla W(v)|\|_1 = \||v|\nabla w(v)|\|_1$, and $\|W\|_1 = \|w\|_1$. Meanwhile, $\int_{D(0)} |v||\nabla w(v)| dv \le \int_{D(0)} |\nabla w(v)| dv$ by that $|v| \le 1$, and $\|w\|_1 \le \|\nabla w\|_1$ by Poincaré inequality, using the fact that w vanishes on the boundary of $D(0)$. Thus $\||v|\nabla w\|_1, \|w\|_1 \le \|\nabla w\|_1$. The claim of the proposition follows by applying Lemma 3.5 to w. □

Proof of Theorem 3.8. Let $c_1 = 4$, $c_2 = 2$. The l.h.s. is bounded by $\|x^{(L)} - D_\tau x^{(L)}\| + \|D_\tau x^{(L)}[x^{(0)}] - x^{(L)}[D_\tau x^{(0)}]\|$. The second term is less than $2c_1 L|\nabla\tau|_\infty \|x^{(0)}\|$ by Theorem 3.7. To bound the first term, we apply Proposition 3.4, and notice that for all λ', λ, $\|\nabla W^{(L)}_{\lambda',\lambda}\|_1 \le 2^{-j_L}\pi \|a^{(L)}_{\lambda,\lambda}\|_{FB}$ (consider $W^{(L)}_{\lambda',\lambda}(u) = W(u) = \sum_k a_k \psi_{J,k}(u) = 2^{-2J} \sum_k a_k \psi_k(2^{-J}u)$, $J = j_L$, let $w(u) = \sum_k a_k \psi_k(u)$, then $W(u) = 2^{-2J} w(2^{-J}u)$, and $\|\nabla W\|_1 = 2^{-J}\|\nabla w\|_1$, where $\|\nabla w\|_1 \le \sqrt{\pi}\|a\|_{FB}$ by Lemma 3.5), and thus $D_L \le 2^{-j_L} A_L$. By (A2'), this gives that $\|D_\tau x^{(L)} - x^{(L)}\| \le c_2 2^{-j_L} |\tau|_\infty \|x_c^{(L-1)}\|$, and $\|x_c^{(L-1)}\| \le \|x^{(0)}\|$ by Proposition 3.1 (b). □

B. Experimental Details

The training of a Conv-2 DCF-FB network (Table 2) on MNIST dataset:

The network is trained using standard Stochastic Gradient Descent (SGD) with momentum 0.9 and batch size 100 for 100 epochs. L^2 regularization ("weightdecay") of 10^{-4} is used on the trainable parameters a's. The learning rate decreases from 10^{-2} to 10^{-4} over the 100 epochs. Batch normalization is used after each convolutional layer. The typical evolution of training and testing losses and errors over epochs are shown in Figure B.1.

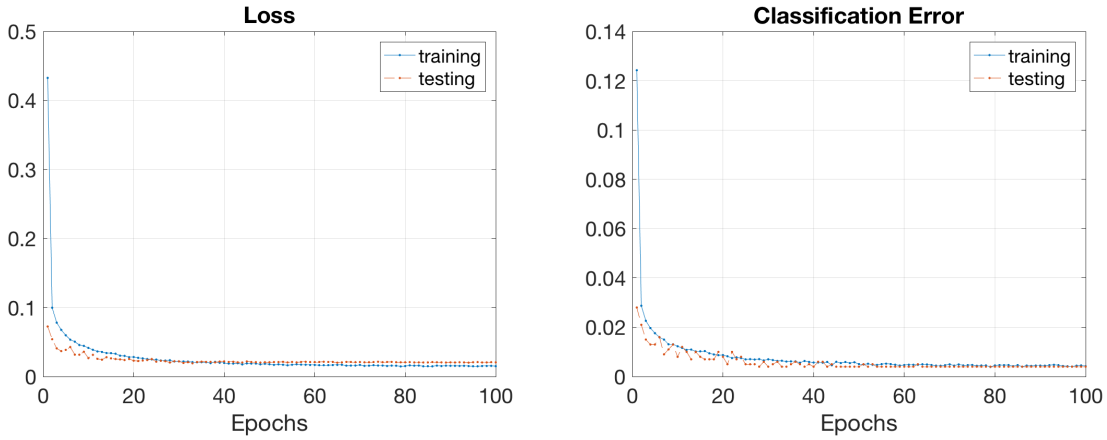


Figure B.1. The evolution of training and validation losses (left) and errors (right) over the epochs of a Conv-2 DCF-FB network trained on 50K MNIST using SGD.