上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

THESIS OF BACHELOR

论文题目 A Hybrid Convolution Network for Image Super-Resolution

学生姓名 高 星

学生学号 5110309216

指导教师 熊红凯教授

专 业 信息工程

学院（系）电子信息与电气工程学院电子工程系

Submitted in total fulfilment of the requirements for the degree of

Bachelor

in EE

A Hybrid Convolution Network for Image Super-Resolution

Xing Gao

Supervisor

Prof. HongKai Xiong

Depart of Electronic Engineer, School of Electronic,Information and Electrical Engineering

Shanghai Jiao Tong University

Shanghai, P.R.China

Jun. 12th, 2015

A Hybrid Convolution Network for Image Super-Resolution

摘　　要

我们设计了一个新颖的卷积神经网络——混合卷积网络，并通过其在图像超分辨应用上取得的当前最高水准的结果验证了其有效性。混合卷积网络由散射卷积网络与卷积神经网络两部分构成并有效地结合了两者的优势。在散射卷积网络部分，复小波函数定义的卷积核可以有效地捕获输入信号的多尺度特性并获得其稀疏表示。而后续的卷积神经网络部分通过在数据集上的训练可以很好地提取针对特定任务的不同数据特征，弥补了散射卷积部分对不同数据针对性不强的缺憾，提升了整个网络的灵活性与自适应性同时有保证了较强的泛化能力。此外，其预定义卷积核与学习得到的卷积核相结合的结构为深度网络的结构设计与训练提供了一种新的机制。更重要的是，混合卷积网络提供了一种将用于稀疏表示的解析字典与从特定数据集中学习得到的字典将结合的一种机制，为字典的设计提供了另一种方案。在图像超分辨的应用中，混合卷积网络取得了目前最好的性能，并由于其端对端的结构在实现时相对于一些基于字典方法更加快速简便。

关键词： 稀疏表示，散射变换，卷积神经网络，图像超分辨率

A Hybrid Convolution Network for Image Super-Resolution

ABSTRACT

We propose a novel convolutional neural network—a hybrid convolution network, and validate its effectiveness by applying it to image super-resolution, where our method achieves state-of-the-art performance. The hybrid convolution network is composed of scattering convolution network and convolutional neural network and incorporates advantages from both sides. At the scattering part, filter kernels predefined by complex wavelet can capture intrinsic multiscale property of the input and obtain its sparse representation. The following convolution part effectively extracts distinct features from specific task, which compensates the drawback of less flexible and adaptive of the scattering part. In addition, the combination of predefined and learnt convolution kernels provides a scheme to design the structure of deep networks, which also makes it easier to train. Moreover, the hybrid convolution network incorporates two kinds of dictionaries, analytic dictionaries and learnt dictionaries, which provides a novel solution to dictionary design. The application to image super-resolution achieves state-of-the-art performance and is easy and fast to implement due to its end-to-end structure.

KEY WORDS: Sparse Representation, Scattering Transform, Convolutional Neural Network, Image Super-Resolution

目　　录

表格索引

插图索引

Chapter 1

Introduction

1.1 Sparse Representation

Meaningful representations that can efficiently capture the useful characteristics and disentangle the underlying explanatory factors from data are essential to signal processing, machine learning, and computer vision. Proper representations may vary with different tasks—for recognition, the representation should be discriminative and kind of invariant; for restoration, the representation should capture detail component effectively; and for compression, the representation should describe a great majority of the signal with only a few coefficients. However, these seemingly different goals mostly converge at simplification which in accordance with Occam's razor.

Since robust statistics advocates sparsity as a key for a wide range of recovery and analysis tasks, this idea has been successfully applied to classical physics, information theory, and signal processing due to its simplicity and effectiveness. Especially, Olshausen and Field in [5] demonstrating that the single assumption of sparsity could account for a fundamental biological visual behavior as well as the emergence of compressed sensing (CS) [6] and the demonstration of the rationale of CS theory from the mathematical perspective [7] promotes the development of sparse representation in computer vision. Sparse and redundant representation seeks to describe signals as linear combinations of a few atoms from a pre-specified dictionary. Therefore, the choice of the dictionary serves an important function, which can be categorized into two general groups: designing analytic dictionaries and learning dictionaries from data. Analytic dictionaries were dominant due to their mathematical simplicity. Especially, the formalization of wavelet theory [8–12], proposed and developed by Grossman, Morlet, Meyer, Daubechies, Mallat and others, which represents signal with a series of translated and dilated mother wavelet, effectively capturing the multi-scale property of natural

signals, captivated majority of researchers and had wide and successful applications, such as deionising [13] and compression [14]. However, the optimal performance of wavelet transform to deal with point singularity can not be extended to curve singularity in multidimensional signals. To address that, other analytic dictionaries that are both localized and oriented such as bandelets [15, 16], curvelets [17, 18], and contourlets [19–21] have been developed. Unfortunately, compared to the complex natural signals, analytic dictionaries tend to be limited expressiveness, leading to the development of learnt dictionaries. The rapid development of machine learning gives promise to learning specific overcomplete dictionaries from data, based on the assumption that the structure of complex natural phenomena can be more accurately extracted directly from the data than by using a mathematical description [22]. Recently, numerous methods of training dictionaries employing $l_0$norm, $l_1$norm or l_p norm with $0 < p < 1$ regularization have been investigated, such as the MOD algorithm [23] and the K-SVD algorithm [24], and accelerate the application of sparse representation in many fields of computer vision. Since the learnt dictionaries are adapted to specific data and thereby more flexible, many of its applications, e.g., super-resolution [25–27], denoising [28], inpainting[29], and compression [24], have achieved state-of-the-art performance. Nevertheless, the ability of learnt dictionaries to capture the intrinsic multi-scale features of natural signals, especially images, is inferior to that of analytic dictionaries. Therefore, how to take advantage of respective superiorities and mediate between the two types of dictionaries becomes an interesting and essential topic and still has not been well studied.

1.2 Convolutional Neural Networks

Convolutional neural networks, as a special class of neural networks, are biologically inspired trainable hierarchical architectures. A typical convolutional neural network consists of one, two or three models: trainable filter banks, non-linearity operators, and pooling operators. Filter banks are usually learnt through back propagation from data to detect different features. Non-linearity operators are used to map a feature into another one. Common non-linearity operators include sigmoid function, rectified linear unit and so on. The effect of pooling operators is to reduce the dimension of

feature. Through weights sharing and local receptive fields, convolutional neural networks reduce the capacity of system and thereby need a relative smaller training set as well as obtain translation-invariant local features. Therefore, convolutional neural networks have shown an explosive prevalent and achieved state-of-the-art performance in a series of pattern recognition tasks [30–32] since back propagation algorithm is applied to train such convolutional neural networks [33].

However, due to its highly non-linearity and all of the parameters learnt from data, it is difficult to understand the relationship between parameters and signal properties. In addition, lack of theoretical basis, it is nearly impossible to know the optimal structure and thereby most of the popular architectures of convolutional neural networks achieving state-of-the-art performance are obtained through numerical experimentations and dependant on significant expertise.

S.Mallat $et\ al.$[3, 34] introduced a special kind of convolution network called scattering convolution network, which consists of a series of wavelet decompositions and modulus pooling operators and produces a kind of representation, both having translation-invariant and incorporating higher order moments. Due to its special structure, scattering convolutional neural networks can avoid over-fitting, reduce the dependance on training data, and optimize its structure according to wavelet theory. However, predefined wavelet coefficients limit their flexibility and adaptivity.

Therefore, how to design a convolutional neural network that has both predefined parameters and learnt filters in order to incorporate both advantages becomes a significant and hard task.

1.3 A Hybrid Convolution Network

In this paper, we propose a hybrid convolution network, which consists of a scattering convolution network and a convolutional neural network. Its novel structure effectively addresses the problem in training networks with deep structure, which has been found to perform better than shallow counterparts, and provides a new scheme to design the structure of neural networks. In addition, it provides a solution to incorporate two kinds of dictionaries and takes advantage of superiorities from both sides from sparse representation perspective. The hybrid convolution network has proven

to both have the ability to capture the intrinsic multiscale property of signals and have adaptivity and flexibility. We also apply this network to practical problem— image super-resolution and achieve state-of-the-art performance.

1.4 Image Super-Resolution

Spatial resolution of an image, defined as the number of pixels per unit, is a measure of the smallest discernible detail in the image and determines the quality of the image, especially detail information. In many applications, images with high-resolution (HR) are desirable and even necessary. For example, a high-resolution medical image is essential to help doctors make a correct diagnosis. The most direct solution to increase spatial resolution is to increase the density of the pixel (reduce the pixel size) through sensor manufacturing techniques. Nonetheless, with the pixel size decreasing, the amount of light available also decreases, which results in the decrease of SNR—shot noise degrading the image quality severely. Therefore, there must be a tradeoff between the pixel size and the quality of image. In other word, we could not reduce the pixel size arbitrarily small. Unfortunately the current image sensor technology has almost reached this limitation. Another method to improve the spatial resolution is to increase the chip size, which leads to an increase in capacitance and will make it difficult to speed up a charge transfer rate, and thereby this approach is not effective. All of these lead to the development of image super resolution (Image SR).

Image super resolution, which aims to restore a high-resolution image from one or serval low-resolution (LR) counterparts, has been a critical issue of computer vision. The technology of the Image SR effectively balances some inherent resolution limitations at the process of image collection and transmission and the growing capability of high-resolution displays, consequently, having a wide application in several fields, such as medical imaging, surveillance imaging, and satellite imaging.

However, image SR task is a severe ill-posed problem, to address which three categories of approaches have been widely investigated: interpolation, reconstruction, and example-based learning. Due to its simplicity and easy to implement, interpolation regimes, such as bilinear and bicubic schemes, have been widely applied. Nevertheless, contrary to assumed analytical smoothness, natural images contain numerous

edges which are strongly discontinuous, resulting in several ringing or jagged artifacts. With regard to reconstruction approaches, HR images are recovered on the basis of reasonable observation model, which relates the original HR image to the observed LR image(s), such as [35, 36]. However, its performance is highly dependent on the reasonability of observation model, and there may not be a general model that can simulate various degraded processes well. Moreover, reconstruction based methods usually get poor performance for large magnification factors [37]. With the development of machine learning, example-based learning approaches, which attempt to learn latent mapping functions between HR and LR pairs from external database, become dominant methods and achieve state-of-the-art results recently. For instance, J.Yang $et\ al.$ [25–27] proposed a series of dictionary based methods to implement image super-resolution on the basis of sparse representation. Chao Dong $et\ al.$ [38] designed a deep convolutional network for image super-resolution and achieved state-of-the-art performance.

Chapter 2

Sparse Representation

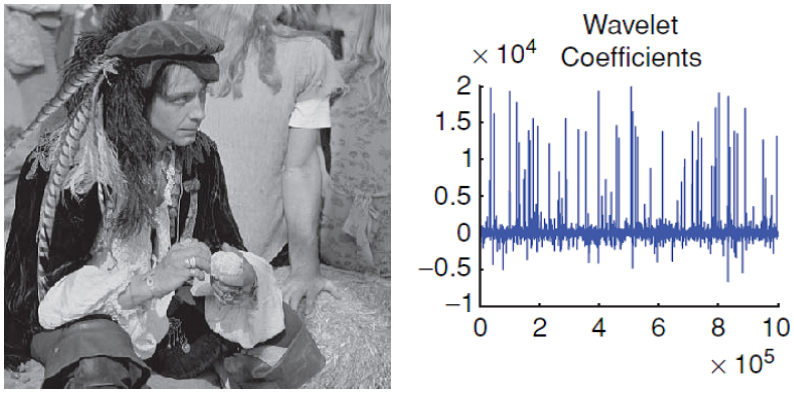
Sparse representation , as the most representative one of linear representation methods, presents an extraordinary power in a wide range of applications, such as machine learning, computer vision, and signal processing, and thereby has received considerable attentions and has been well investigated in recent years. Sparse and redundant representation seeks to describe signals as linear combinations of a few atoms from a pre-specified dictionary, and the choice of the dictionary plays an important role. Therefore, how to choose a proper dictionary becomes a big challenge. At first, a series of analytic dictionaries such as wavelet and curvelet dictionaries were chosen because they are mathematically simple and easy and fast to implement. With the development of machine learning, another category of dictionaries—learnt dictionaries, gradually become dominant and attract much attention because of the capability to be adaptive to specific data and flexibility.

In this chapter, we will do a comprehensive investigation of sparse representation, and review and compare two categories of dictionaries. Firstly, we will discuss the definition of sparsity and the measurement of it. Then we will study some typical analytic dictionaries and followed by the introduction of learnt dictionaries and classical learning schemes. Finally, we will do some comparisons.

2.1 Sparsity

Sparse representation attempts to describe signals as linear combinations of a few atoms from a group of pre-defined basis. Many natural signals indeed can be concisely represented by a proper basis, for example, the image in Fig.2–1 (a) and its wavelet transform coefficients in (b). While most of the image pixels are nonzero, most of wavelet coefficients are small except for a few of large coefficients which capture most

of the information.



(a) Image with pixel values in the range [0,255].

(b) Wavelet transform coefficients.

Figure 2–1 Natural image and its wavelet transform coefficients from [1]

For signal transforms, we seek to express a signal $f(t)$ as a linear combination of basis $\Psi = [\psi_1, \psi_1, \cdots, \psi_n]$:

$$f(t) = \sum_{i=1}^{n} x_i \psi_i(t), \tag{2–1}$$

where x_i is the project of $f(t)$ on $\psi_i(t)$. The implication of sparsity is that most of these coefficients x_i are or approximate zeros.

Sparsity can be measured with many metrics. The most intuitive and simplest one is the l_0 norm, which is the number of nonzero coefficients. While it is simple and easy to understand, it is not applied to practical work, because a vector of signal would hardly be expressible by a vector of coefficients containing many strict zeros. Moreover, since the l_0 norm function is non-convex, non-smooth, and discontinuous, it is hard to solve. Therefore, a weaker notion of sparsity is necessary and can be quantified by the l_p norm with $0 < p < 1$, which is defined as:

$$\|x\|_p = (\sum_i |x_i|^p)^{1/p} \tag{2–2}$$

The l_0 norm can be served as a special case of the l_p norm, i.e., the limit form of the l_p

norm as $p \to 0$:

$$\|x\|_0 = \lim_{p\to 0} \|x\|_p^p = \lim_{p\to 0} \sum_i |x_i|^p. \tag{2–3}$$

Fig.2–2 presents the relationship between different norms. As p goes to zero, the l_p norm tends to be the l_0 norm, counting the number of nonzero coefficients. In addition, it is easy to notice that the l_p norm $(0 < p < 1)$ function is also non-convex, non-smooth and non-differentiable at zero point. The l_1 norm function becomes convex but still non-smooth and non-differentiable at zero.

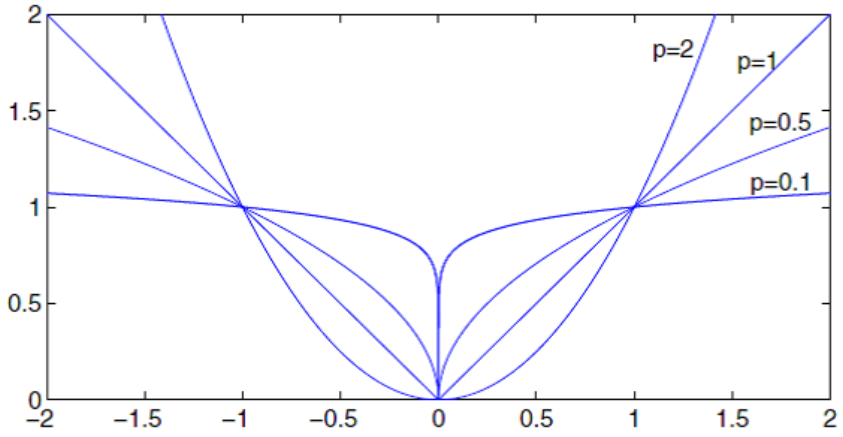


Figure 2–2 The relationship between different norms [2].

2.2 Analytic Dictionaries

Analytic dictionaries are usually designed through Harmonic Analysis, such as windowed fourier transform and wavelet transform, and thereby are generally supported by a series of mathematical analysis and proofs. Most of analytic dictionaries for sparse representation have localized atoms and these atoms gradually have specific orientations in order to process high dimensional signals. We will start the investigation of analytic dictionaries from wavelet transform, followed by curvelet transform and contourlet transform.

2.2.1 Wavelet Transform

To achieve sparsity, it is necessary for atoms to be localized and have concentrated supports. To analyze the signal structures of different scales, it is necessary to employ time-frequency atoms with varying time and frequency supports. Wavelets transform takes advantage of a group of localized, dilated and translated wavelets $\{\psi_{a,b}(u)\}_{j,b}$ to decompose signals. All of these wavelets come from a mother wavelet $\psi(u)$, which has a zero average:

$$\int_{-\infty}^{+\infty} \psi(u)du = 0, \tag{2–4}$$

and usually has a unit norm $\|\psi(u)\|$ as well as centered in the neighborhood of origin (0,0). A dictionary of different scale time-frequency atoms are then obtained by dilating and translating the mother wavelet:

$$\psi_{a,b}(u) = 2^{-2a}\psi(\frac{u-b}{2^a}). \tag{2–5}$$

A signal can be expressed as a linear combination of such wavelets, as follows:

$$f(u) = \frac{1}{C_\varphi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} W_f(a,b)\psi_{a,b}(u)\frac{dadb}{a^2} \tag{2–6}$$

$$W_f(a,b) =< f, \psi_{a,b} >= \int_{-\infty}^{+\infty} f(u)\overline{\psi_{a,b}(u)}du \tag{2–7}$$

An advantage of wavelet transform is its approximate localization property in both time and frequency domain so that signals can be expressed sparsely. In addition, different scale or resolution windows of wavelet transform makes it more powerful. A window having a large scale in time domain has a coarse resolution in frequency domain, which can effectively captures gross features and processes low-frequency signals. In contrast, fast changing signals can be captured by a window with a small scale in time domain, which has a fine resolution in frequency domain. In consequence, wavelets can capture different scale features of signals through varying resolutions. Moreover, it has been proven that for one-dimensional piecewise smooth signals, wavelets provide an optimal representation.

2.2.2 Curvelet Transform

While wavelets perform well for signals with point singularities in one and two dimensions, it is less successful to deal with edge discontinuities in two-dimension. However, many images of different applications exhibit edge discontinuity. To address this problem, David L.Donoho $et\ al.$ introduced curvelet transform in [17, 18]. Like wavelet transform, curvelet transform decomposes signal through a group of atoms, which have different locations, orientations, and scales. The curvelet transform attracted numerous attentions because it does nearly as well as adaptive methods that explicitly track the shape of the discontinuity and use a special adaptive representation dependent on that tracking in representing objects with discontinuities along curves [17]. Just as wavelet transform does well in processing point singularities, curvelet transform is good at processing curvilinear singularities.

2.2.3 Contourlet Transform

Despite curvelet transform does well in representing signals with smooth curves, its discretization turns out to be unsatisfying. Contourlet transform, as a "true" two-dimensional transform capturing the intrinsic geometrical structure, was introduced by Minh N.Do and Martin Vetterli in[19, 21]. In contrast to curvelet, contourlet was initially constructed in discrete domain and then expanded to continuous domain. The contourlet transform has many of the characteristics such as localization, orientation, and parabolic scaling in common with the curvelet transform. The contourlet transform is implemented through a pyramidal band-pass decomposition of the image followed by multidirectional expansion using nonseparable filter banks. It has been shown in [21] that contourlet transform achieves the optimal approximation rate for piecewise smooth functions with discontinuities along twice continuously differentiable curves with parabolic scaling and sufficient directional vanishing moments.

2.2.4 Complex Wavelets

The complex wavelet transform [39], obtained by utilizing of two mother wavelets satisfying a specific relationship between them, is an oriented and near-translation-invariant high-dimensional extension of the wavelet transform. Similar to the original

wavelet transform, the complex wavelet transform is efficient and simple to implement. In addition, because of the added phase information, the complex wavelet transform presents greater directional sensitivity and other favorable properties.

2.3 Learnt Dictionaries

This kind of dictionaries are obtained by learning from data and thereby fit the data better. Recently, numerous dictionary training methods have been investigated and give a promising development of learnt dictionaries. Current dictionary training methods usually take advantage of l_0 and l_1 sparsity measures to ensure sparsity. In this section, we will overview some typical and popular dictionary training methods, such as the MOD and K-SVD algorithm.

2.3.1 The Mod Method

Engan $et\ al.$ [23, 40] came up with a dictionary training algorithm, named method of optimal directions (MOD). The frame of the MOD can be represented mathematically as follow:

$$arg \min_{\mathbf{D},\mathbf{Y}} \|\mathbf{X} - \mathbf{D}\mathbf{Y}\|_2 \quad subject \quad to \quad \forall i, \|\mathbf{y}_i\|_0 \leq \epsilon, \tag{2–8}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ denotes examples, $\mathbf{D}$ represents the dictionary, and $\mathbf{y}_i$ is a column of $\mathbf{Y}$ representing the sparse representation of $\mathbf{x}_i$. Similar to K-means, the MOD alternates sparse-coding and dictionary update. The sparse coding can be achieved through any common method individually . After obtaining the sparse coding, the dictionary can be updated by finding the analytic solution. After a few iterations, it will converge and we can obtain the dictionary $\mathbf{D}$. The advantage of the MOD is that it is simple to update the dictionary. However, since the optimization problem is highly non-convex, we usually obtain a local optimal solution instead of the global optimal one. In addition, the matrix inverse operation at the process of obtaining analytic solution is relative highly complex.

2.3.2 Unions of Orthonormal Basis

Lesage $et\ al.$ [41] introduced a dictionary training method through unite orthonormal basis. The overcomplete dictionary is composed of k orthonormal bases:

$$\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_k],$$

and the sparse representations of $\mathbf{X}$ are obtained through concatenation of pieces respectively referring to different orthonormal basis:

$$\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_k]^T.$$

Training process can also be divided into two stages—sparse coding and dictionary update. Sparse coding is usually achieved through a Block Coordinate Relaxation technique, and the update of dictionary is done sequentially. For each $\mathbf{D}_j$, the update is achieved by computing the singular value decomposition of the residual matrix $\mathbf{E}_j\mathbf{X}_j^T = \mathbf{U}\Lambda\mathbf{V}^T$, where $\mathbf{E}_j = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n] = \mathbf{Y} - \sum_{i\neq j}\mathbf{D}_i\mathbf{X}_i$, and let $\mathbf{D}_j = \mathbf{U}\mathbf{V}^T$. And also $\mathbf{D}_j$ is forced to be orthonormal.

This scheme reduces complexity and thereby can be more efficient by using a structured dictionary instead of a free one. However, compared to other methods, this method tends to have a weak performance due to its relatively restrictive condition and unfavorable to couple different parts of the dictionary.

2.3.3 The K-SVD Algorithm

The K-SVD algorithm, introduced by Aharon $et\ al.$ [24], is an efficient algorithm to train overcomplete dictionary. This algorithm is a direct generalization of the K-means and takes advantage of singular value decomposition to update dictionary so that it was called "K-SVD".

Just as the MOD, the goal of the K-SVD algorithm can be described as follows:

$$arg\min_{\mathbf{D},\mathbf{Y}} \|\mathbf{X} - \mathbf{D}\mathbf{Y}\|_2 \quad subject \quad to \quad \forall i, \|\mathbf{y}_i\|_0 \leq \epsilon. \tag{2–9}$$

The K-SVD algorithm also contains two stages to minimize equation (2–9). First, solv-

ing the best and sparse representations **Y** of **X** with fixed dictionary **D**. Usually the sparse coding **Y** is not the optimal one because it is non-convex, and the pursuit method can be any common algorithm. Once the sparse coding is found, we can update dictionary through singular value decomposition. At each time, only one column of dictionary **D** and its corresponding parameter are updated. After a few iterations, this algorithm will converge. Compared to other methods, the K-SVD algorithm changes the coefficient of sparse coding at the time updating each column of dictionary, which accelerates convergence because the subsequent updates will be dependant on these coefficients. However, the K-SVD algorithm is also faced with non-convexity and thereby may be caught in local optimal solution.

2.3.4 Online Dictionary Learning

J.Mairal $et\ al.$ [42] introduced online dictionary learning scheme, which allows training dictionaries from large-scale data and converges faster. This algorithm is based on the assumption that these examples have an i.i.d distribution. At each time, one example $\mathbf{x}_i$ is chosen, and like other methods , the algorithm alternates between sparse coding step and dictionary update step. Experiments demonstrate that it is significantly faster than batch alternatives such as K-SVD and the MOD.

2.3.5 Parametric Training Methods

The motivation for training a parametric dictionary is to reduce the number of free parameters in order to reduce the number of local minima and accelerate convergence. In addition, smaller number of parameters can reduce the dependance on data and thereby can improve generalization.

Recently, numerous parametric dictionaries are proposed and we will take multiscale dictionaries as a representation. Usually, the dictionary for sparse representation is single scale which can not effectively capture the multiscale property of natural signals. To address this problem, some schemes of designing and training multiscale dictionaries have been proposed, which remains to be a challenging topic. Sallee and Olshausen in [43], introduced a pyramidal wavelet-like signal expansion, generated from the dilations and translations of a set of elementary small trained patches. In [29],

Mairal $et\ al.$ proposed a semi-multiscale extension of the K-SVD. The semi-multiscale structure is obtained through incorporating several fixed-sized learnt dictionaries of different scales. These schemes approximate state-of-the-art performance in different applications.

In addition, sparse dictionaries make a big contribution to parameter dictionaries. Rubinstein, Zibulevsky and Elad [44] proposed a sparse dictionary structure with $\mathbf{D} = \mathbf{BA}$, where **B** is some fixed analytic dictionary with a fast computation, and **A** is a sparse matrix. Thus, the dictionary both has a fast implementation and has adaptivity through the matrix **A**. Morever, the parameterization reduces the dependance on training data and thereby has a good generalization.

2.4 Comparison

Analytic dictionaries are characterized by its fast implementation and multiscale property. Due to the localization property of atoms, signals can be decomposed as a sparse representation through direct calculation. Since analytic dictionaries are usually composed of dilated, translated and even oriented atoms, the intrinsic multiscale property of natural signals can effectively be captured. However, due to its non-adaptiveness, these dictionaries tend to be over-simplistic and less expressive compared to the complexity of natural signals.

Learnt dictionaries overcome the drawback of analytic dictionaries. Based on different dictionary training methods, learnt dictionaries have a wide applications and achieve nearly or even exceed state-of-the-art performance. While these kinds of dictionaries are flexible and can effectively extract intrinsic specific structure of different data, they usually have a poor generalization capability. In addition, since the training process and solving sparse coding are both non-convex, the global optimal solution is usually impossible. Finally, compared to analytic dictionaries, the process of pursuing sparse representation of a signal is more complex and time-consuming.

Therefore, designing a dictionary incorporating two kinds of dictionaries in order to take advantage of both superiorities comes to an interesting and challenging issue, which needs to be well investigated.

Chapter 3

Scattering Convolution Networks

A Scattering network, introduced in [3, 34], is a special convolution neural network whose filters are not learnt but are predefined complex wavelets and produces a kind of representation which is translation-invariant and incorporate higher order moments .

3.1 Scattering Wavelets

A wavelet $\psi(u)$ is a bandpass filter with $\int \psi(u)du = 0$. If $\hat{\psi}(\omega) = 0$ for $\omega < 0$, then it is analytic. Moreover, if a wavelet $\psi(u)$ has p directional vanishing moments along any one-dimensional line of direction $\alpha + \pi/2$ in the plane but does not have directional vanishing moments along the α directional, then it is a directional wavelet.

Let $\psi(u)$ be a single bandpass filter, and two-dimensional directional wavelets can be obtained through dilating it by 2^j for $j \in Z$ and rotating it by $r \in \Theta$ where $\Theta = \{\alpha = 2k\pi/K, 0 \leq k \leq K\}$:

$$\psi_\lambda(u) = 2^{-2j}\psi(2^{-j}r^{-1}u) \quad with \quad \lambda = 2^{-j}r \tag{3–1}$$

Provided $\hat{\psi}(\omega)$ is centered at ω_0, then the support of $\hat{\psi}_\lambda(\omega)$is centered at $\lambda\omega_0$ with a frequency bandwidth proportional to 2^{-j}. In other word, the index λ decides the frequency location and bandwidth of ψ_λ.

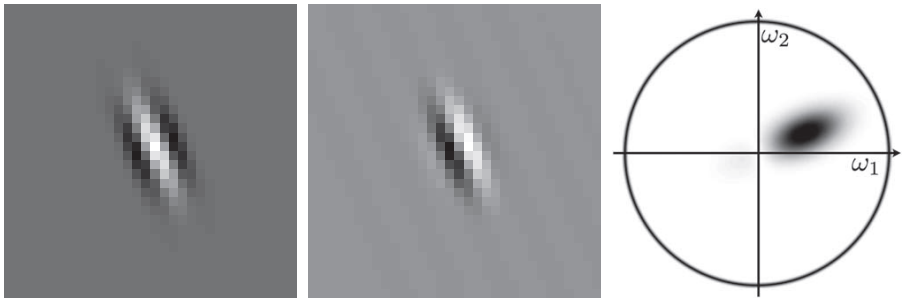
A complex Morlet wavelet, defined as:

$$\psi(u) = \alpha(e^{iu.\xi} - \beta)e^{-\|u\|^2/2\sigma^2}{}^1$$

where β is a parameter so that $\int \psi(u)du = 0$, is a typical example of two-dimensional

[1] u.u' denotes the inner product and $\|u\|$ denotes the norm of u

directional wavelet. Fig. 3–1 shows the Morlet wavelet with $\sigma = 0.85$ and $\xi = 3\pi/4$, used in all experiments.



(a) Real part of $\psi(u)$. (b) Imaginary part of $\psi(u)$. (c) Fourier modulus $\|\hat{\psi_\lambda}(\omega)\|$

Figure 3–1 Complex Morlet Wavelet From [3].

3.2 Scattering Transform

In order to obtain translation invariant and stable to deformation representation, S.Mallat $et\ al.$ introduced scattering transform in [3, 34]. Scattering transform takes advantage of the strong power of wavelet transform to capture intrinsic multi-scale property of natural image and the ability of deep networks to extract invariant features from input signals.

Due to wavelet transform commuting with translations, in order to achieve translate invariant representation, we can use integral to address it. Nevertheless, for any wavelet $\psi(u)$ satisfying $\int \psi(u)du = 0$, it is necessary to additionally introduce a nonlinear operator to obtain invariant property. Moreover, to guarantee the representation is stable to deformation, the nonlinear operator M must be nonexpansive: $\|Mx - My\| \leq \|x - y\|$. Considering all of that, a modulus operator over complex signals $x = x_r + jx_i$:

$$|x| = (|x_r|^2 + |x_i|^2)^{1/2}.$$

is chosen. Therefore, the translation invariant coefficients are then $\mathbf{L}^1(\mathbb{R}^2)$ norms:

$$\|x \star \psi_\lambda\|_1 = \int |x \star \psi_\lambda(u)|du \tag{3–2}$$

The information does not lose after the modulus operator that removes the complex phase of $x \star \psi_\lambda$, while the integration of $x \star \psi_\lambda$ will remove all frequency components resulting information lost. Fortunately, these removed nonzero frequency components can be recovered by a new wavelet decomposition $|x \star \psi_{\lambda 1}| \star \psi_{\lambda 2}$, and the $\mathbf{L}^1(\mathbb{R}^2)$ norms of these wavelet coefficients produce a much larger number of invariants:

$$\| |x \star \psi_{\lambda 1}| \star \psi_{\lambda 2} \|_1 = \int ||x \star \psi_{\lambda 1}(u)| \star \psi_{\lambda 2}| du \tag{3–3}$$

Through further iterating this procedure, more translation invariant coefficients can be obtained.

Let $U[\lambda]x = |x \star \psi_\lambda|$ denote the wavelet transform as well as modulus operator at subband λ. Let any sequence $p = (\lambda_1, \lambda_2, \cdots, \lambda_m)$ represent a path with length m, along which the ordered wavelet decomposition as well as modulus operator is calculated:

$$\begin{aligned} U[p]u &= U[\lambda_m] \cdots U[\lambda_2] U[\lambda_1] u \\ &= |||u \star \psi_{\lambda 1}| \star \psi_{\lambda 2}| \cdots \star \psi_{\lambda m}|. \end{aligned} \tag{3–4}$$

If $p = \emptyset$, then $U[\emptyset]x = x$.

Considering that invariant property to a certain extent is more promising in practical applications, the definition of scattering transform is :

$$\begin{aligned} S[p]u &= U[p]x \star \phi_J(u) \\ &= U[\lambda_m] \cdots U[\lambda_2] U[\lambda_1] u \star \phi_J(u) \\ &= |||x \star \psi_{\lambda 1}| \star \psi_{\lambda 2}| \cdots \star \psi_{\lambda m}| \star \phi_J(u), \end{aligned} \tag{3–5}$$

instead of directly using the $\mathbf{L}^1(\mathbb{R}^2)$ of $U[p]$.

3.3 Frequency Domain Analysis of Scattering Transform

Two-dimensional directional wavelets, which are usually obtained by dilating and rotating a single bandpass filter, divide the frequency domain into sectors. For the same scaling j_m, directional wavelets of different rotations form an annulus, and all of these

annuluses of different scaling j occupy the whole frequency domain.

Scattering transform consists of convolution with directional wavelets $\{\psi_\lambda\}_{\lambda\in\Lambda}$ and modulus operator. According to the convolution property of Fourier Transform, the frequency component of $x\star\psi_\lambda$ is $\hat{x}\cdot\hat{\psi}_\lambda$. Therefore, the frequency spectrum of signal x has been divided by $\hat{\psi}_\lambda$. The effect of modulus operator is to shift each sector into lower frequency and also to change the shape of the sector because modulus operator takes the envelope of signal and removes the oscillations. Due to the frequency varying at different positions in the sector, the degree of shift is different and thereby the shape of each sector will change.

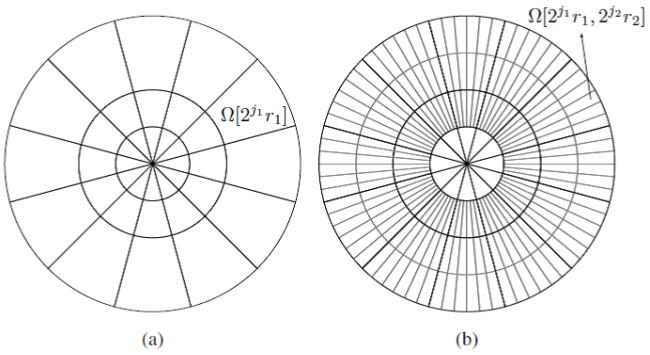


Figure 3–2 Frequency domain division of scattering transform from [4].

To be concrete, at the first layer, the frequency spectrum of signal x is divided into $K \cdot J$ sectors, the position of which are decided by λ. At the second layer, each shifted sector continues to be divided by these bandpass filters $\{\psi_\lambda\}_\lambda$. With the increment of the number of layers, each sector is divided more elaborately. Fig. 3–2 demonstrates this process— the sector $\Omega[\lambda_1]$ denotes the frequency support of $S[\lambda_1]x$ and the sector $\Omega[\lambda_1, \lambda_2]$ denotes the frequency support of $S[\lambda_1, \lambda_2]x$.

3.4 Scattering Convolution Networks

Scattering transform can be implemented through a special convolution network, whose filters are predefined wavelets instead of learnt from data. In contrast to traditional convolution neural network, scattering convolution network produces scattering

coefficients at each layer instead of only the last layer. At each layer, through computing a convolution with lowpass filter ϕ_J and bandpass filters $\{\psi_\lambda\}_\lambda$, we obtain scattering coefficients of this layer and wavelet modulus coefficients of the next layer:

$$\begin{aligned}\widetilde{W}_m U_m x &= \{U_m x \star \phi_J, |U_m x \star \psi_\lambda|\}_{\lambda\in\Lambda} \\ &= (S_m x, U_{m+1} x),\end{aligned} \tag{3–6}$$

where $U_m x$ represents wavelet modulus coefficients of the mth layer, $S_m x$ represents scattering coefficients of the mth layer and $\Lambda = \{\lambda = 2^{-j} r : r \in \Theta, j \leq J\}$. Through repeatedly applying $\widetilde{W}_m$, scattering transform can be accomplished, which is shown in Fig. 3–3.

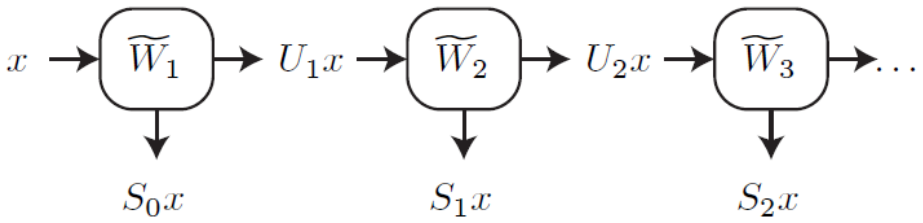


Figure 3–3 Calculate scattering representation from [4].

The architecture of scattering convolution network is shown in Fig. 3–4. At the first layer, applying $\widetilde{W}$ to the input signal x —computing convolution with all of the filters defined by wavelets ψ_λ with $\lambda \in \Lambda$ and another filter defined by scaling function ϕ_J, produces the first layer of feature maps $U_1 x$ as well as scattering coefficients $S_0 x$. We can obtain the second layer of feature maps $U_2 x$ as well as another scattering coefficients $S_1 x$ by applying $\widetilde{W}$ to any feature map of the first layer. Iterating this procedure until the last layer M, we can obtain the scattering representation of signal x .

3.5 Analysis of Scattering Properties

The most important property of scattering transform is that it is energy conservation:

$$\|x\|^2 = \|Sx\|^2 = \sum_{p\in P^\infty} \|S[p]x\|^2. \tag{3–7}$$

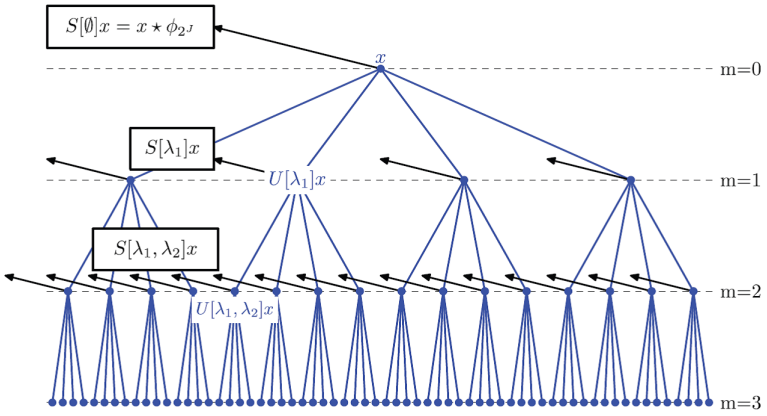


Figure 3–4 The architecture of scattering convolution network from [3].

This property also means that the depth of scattering convolution networks can be limited, which is essential to numerical applications. In addition, the property of energy conservation illustrates that the more sparse the wavelet coefficients, the more energy propagates to deeper layers. Because the energy of scattering coefficients are fixed, the more sparse $x \star \psi_\lambda$, the smaller $\|x \star \psi_\lambda\|_1$ and thereby the more energy deeper layers have.

Moreover, due to energy conservation, there exists special paths where most of the energy is concentrated. Because the modulus operator takes the envelope of signal and removes the oscillations, $|x \star \psi_\lambda|$ is more regular than $x \star \psi_\lambda$, which means that $|x \star \psi_{\lambda 1}| \star \psi_{\lambda 2}$ is nonnegligible only if the frequency support of $\psi_{\lambda 2}$ locates at lower frequency than that of $\psi_{\lambda 1}$, namely $|\psi_{\lambda 2}| < |\psi_{\lambda 1}|$. We call such path $p = (2^{-j_1} r_1, 2^{-j_2} r_2, \cdots, 2^{-j_m} r_m)$ satisfying $0 < j_k \leq j_{k+1} \leq J$ frequency decreasing path. Scattering coefficients along other paths have a negligible energy. Therefore it is sufficient to compute the scattering transform only along frequency decreasing pathes, which produces a fast implementation algorithm and reduces the computational complexity exponentially.

3.6 Comparison with Traditional Convolution Networks

Compared to conventional neural networks, scattering convolution neural network has two distinctions. On the one hand, filters of traditional convolution networks are

usually learnt from data through back propagation algorithm or unsupervised training such as auto-encoders, while filters of scattering convolution networks are predefined wavelets. On the other hand, we obtain scattering coefficients from each layer instead of just the last layer.

Due to its special structure, scattering convolutional neural networks have several advantages. Above all, predefined filter coefficients avoid over-fitting and reduce the dependance on training data, which makes it more generic. In addition, mature wavelet theory provides scattering convolutional neural networks solid theoretical foundation to optimize its structure, while the design of common convolution networks usually relies on numerical experimentations and significant expertise. Moreover, taking advantage of the sparsity and multi-scale property, scattering convolution networks can capture intrinsic multi-scale character of natural image.

However, scattering convolution networks inevitably have some drawbacks. These predefined coefficients limit their ability to obtain distinct features of different data and thereby are less flexible and adaptive.

Chapter 4

A Hybrid Convolution Network

4.1 Focus on Convolutional Neural Networks

Convolutional neural networks (CNN), also known as convolution networks, are biologically-inspired variants of multi-layer perceptions. According to the study on cat's visual cortex [45], visual cortex is composed of complex arrangements of cells, which are sensitive to sub-regions of the visual field called receptive field. The total visual field is covered by tiling the receptive fields. It is convincing that the animal visual cortex is the most effective and powerful visual processing system because of evolution. Hence, it is natural to emulate such system and thereby convolutional neural networks attract numerous attention and have been well studied for a long time.

Convolutional neural networks, as a special branch of neural networks, achieve great success in practical applications. Compared to other neural networks, convolutional neural networks can work with inputs with any size and thereby are more flexible and avoid the effect of preprocessing such as wrapping on input signals. In addition, there are three distinct characteristics: sparse connectivity, parameter sharing and translation-invariant representation that make a contribution to the success of convolutional neural networks.

Each neuron of neural networks usually has a full connection with neurons of previous layer. However, convolutional neural networks employ a local connection with neurons of the previous layer. This is achieved by making the size of receptive field smaller than its input. For instance, the input signal is an image with 256×256 pixels while the size of receptive field is 5×5. Fig.4–1 illustrates this concept concretely. If layer $m-1$ is the input layer, then each neuron of the first hidden layer is only connected with three neurons of the input layer instead of five neurons. In this way, we can reduce the number of free parameters and thereby reduce the dependance on training data and

save memories. In addition, the property of sparse connectivity makes sure that learnt filters have a strong response to local features of input such as edges of an input image.

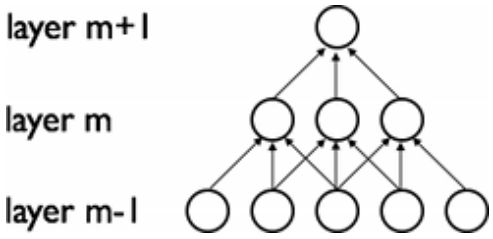


Figure 4–1 Sparse connectivity of CNN.

Parameter sharing means producing each feature map only using the same parameter. To be concrete, if the input signal is an image with 256×256 pixels and there are ten neurons each having a filter with 5×5 pixels in the first layer, then for each feature map of the first hidden layer only 25 parameters are used and 250 parameters in total. In contrast, we need $256 \times 256 \times 10$ parameters in neural networks. Through this method, the number of parameters can be reduced effectively. In addition, parameter sharing leads to the property of translation invariant. Regardless of the position in the visual field, we will obtain the same features.

4.1.1 Composition

A convolutional neural network can be considered as a function F, which maps input x to output y. The function F is composed of a cascade of convolution, non-linear operator, and pooling operator. Let us start from studying the role of one neuron of convolutional neural networks, which is shown in Fig.4–2. Here, $x \in \mathbb{R}^{H \times W \times D \times N}$ represents the input signal, w are coefficients of filters, f describes the function of each neuron, and y denotes the output. Firstly, the neuron calculates the convolution between x and w, and then the result activates function f which may be a sigmoid function and cascaded with max pooling to produce output y. A series of these neurons are arranged deliberately in order to achieve a complex map function.

In order to understand the function of convolutional neural networks, it is necessary to investigate these models— filter banks, non-linear operators, and pooling operators separately.

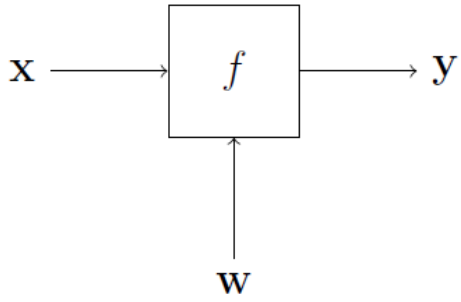


Figure 4–2 A neuron of convolutional neural networks.

4.1.1.1 Filter Banks

Supposing the input is a three dimensional signal $x \in \mathbb{R}^{H\times W\times D}$ and each neuron has a bank of K filters $f \in \mathbb{R}^{H'\times W'\times D'\times K}$, then the convolution can be calculated as follows:

$$y_{i''j''k} = b_k + \sum_{i'=1}^{H'}\sum_{j'=1}^{W'}\sum_{d=1}^{D} f_{i'j'd} \times x_{i''+i',j''+j',d,k}. \tag{4–1}$$

Coefficients of filter banks are usually learnt from data and each filter extracts a particular feature from the input.

4.1.1.2 Non-Linear Operators

The role of non-linear operators is to implement affine transformations and non-linear mapping. The degree and location of these non-linearities are controlled by the choice of parameters. With the appropriate choice of parameters, multi-layer neural networks can approximate any smooth function. Here we will summarize a series of popular non-linearity operators.

Sigmoid: $h(x) = \frac{1}{1+e^{-(b+\mathbf{w}\cdot\mathbf{x})}}$. The sigmoid function looks like a character "s" with range **[** 0,1**]**.

Hyperbolic tangent: $h(x) = tanh(b + \mathbf{w}\cdot\mathbf{x})$.

Rectified linear unit (ReLU): $h(x) = max(0, b + \mathbf{w}\cdot\mathbf{x})$.

Radial basis function (RBF): $h(x) = e^{-\|w-x\|^2/\sigma^2}$. This function is usually used in kernel SVM.

Hard tanh: $h(x) = max(-1, min(1, x))$.

Absolute value rectification: $h(x) = |x|$ It makes sense to seek features that are invariant under a polarity reversal of the input and thereby is used to recognize object.

4.1.1.3 Pooling Operators

The pooling operator summarizes the responses of a neighborhood through analyzing the statistic property of the nearby outputs and discards the exact position of features. For example, the average pooling operation provides an average output within a neighborhood. Max pooling and weighted average pooling are another popular pooling operators. In all these cases, pooling operators play an important role in two aspects. On the one hand, the feature becomes translation invariant to some extent through pooling. On the other hand, it is essential to address inputs of varying sizes. The size of input signal may vary but the input to classifier sometime is fixed. Therefore, through adjusting the stride of pooling we can obtain a fixed length feature, which can be fed into classifier. However, the effectiveness of pooling operator is based on the assumption that exact positions of features are not important and the property of translation invariant is desired. If we want to know the location of the feature, we may not use any kind of pooling operator.

4.1.2 Network Training – Back Propagation Algorithm

4.1.2.1 Loss Function

Convolutional neural networks can be viewed as a general class of parametric nonlinear functions, which map the input **x** to the output **y**. And the degree of approximation to a function f can be measured by a series of loss functions. The most commonly used loss function is the sum-of-squares error function:

$$\|E(f_\theta(\mathbf{x}) - \mathbf{y})\|^2 = \frac{1}{2}\sum_{n=1}^{m} \|f_\theta(\mathbf{x}_n) - \mathbf{y}_n\|^2.$$

We can interpret this loss function from the probability perspective. Assuming that these examples are generally i.i.d from a distribution $P(X, Y)$, minimizing the loss

function is equivalent to maximize the likelihood estimation of parameters θ , which yields $f_\theta(\mathbf{x}) = E[Y|X = \mathbf{x}]$. This interprets the role of convolutional neural networks.

4.1.2.2 Error Backpropagation

The term back-propagation means the scheme for computing gradients in convolutional neural networks. It has been proven that the back-propagation algorithm is the optimal one to calculate the derivatives of such networks. Once having the derivatives, we can compute the adjustments to be made to optimize parameters.

The basic idea of back-propagation comes from the chain rule:

$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f(g(x))}{\partial g(x)} \cdot \frac{\partial g(x)}{\partial x}. \tag{4–2}$$

First, just consider one neuron of a convolutional neural network. In general, each union first computes a weighted sum of its input:

$$a_j = \sum_i w_{ji} z_i, \tag{4–3}$$

where z_i is the input of each neuron and w_{ji} is the weight associated with the input. Then the sum activates the nonlinear function $h(\cdot)$ to give the output of neuron j:

$$z_j = h(a_j).$$

Now calculate the derivative of E with respect to a weight w_{ji}. We should pay attention to that E is dependant on w_{ji} only through the summed input a_j to neuron j. Therefore, we can apply the chain rule (4–2):

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial a_j} \cdot \frac{\partial a_j}{\partial w_{ji}}. \tag{4–4}$$

Let

$$\delta_j \equiv \frac{\partial E}{\partial a_j}, \tag{4–5}$$

where δ_j refers to errors, and according to (4–3), we can write (4–4) as:

$$\frac{\partial E}{\partial w_{ji}} = \delta_j z_i. \tag{4–6}$$

According to (4–6), we know that the required derivative can be obtained just by multiplying the error δ by the input z corresponding to the wight. Next, we will calculate the δ for each neuron. By using the chain rule again, we can obtain :

$$\delta_j \equiv \frac{\partial E}{\partial a_j} = \sum_k \frac{\partial E}{\partial a_k}\frac{\partial a_k}{\partial a_k}, \tag{4–7}$$

where the sum runs over all neurons k to which neuron j sends connections. By substituting (4–5), we obtain the back propagation formula:

$$\delta_j = h^{'}(a_j)\sum_k w_{kj}\delta_k. \tag{4–8}$$

The formula (4–8) illustrates that the error δ for each neuron can be obtained through propagating the $\delta^{'}s$ backward from neurons located higher up in the network.

In summary, the procedure of back-propagation is listed as follows:

1.Forward propagating the input until output neurons;

2.Calculating the error δ for output neurons;

3.Back propagating the error to evaluate the error δ for each hidden neuron;

4.Calculating the derivatives.

After obtaining these derivatives, we can employ optimization algorithms such as gradient decent algorithm to adjust parameters in order to minimize the loss function and thereby train the convolutional neural networks to implement specific task.

4.2 A Hybrid Convolution Network

4.2.1 Motivation

Convolutional neural networks, as a special case of neural networks, have achieved a big success due to its distinct structure, which brings in sparse connectivity, parameter

sharing and translation invariant property. Its special design of structure is based on the prior assumption that local receptive field is more efficient to capture visual features, which is inspired by the mammalian visual cortex. Therefore, involving proper prior assumptions can improve the performance of networks. Recently, with researchers' finding that artificial neural networks with deep structure will perform better than shallow counterpart, networks with a deep structure in order to extract features from data more efficiently becomes dominant. However, how to train such deep convolutional neural networks comes to a big challenge. Although back propagation algorithm can still work for such deep networks, we usually get a local optimal solution which may highly differ from the global optimal one due to its increasing non-convexity with increment of the depth of networks. To address this problem, some unsupervised or semi-supervised methods have been employed, which trains each layer individually and then refines parameters. These schemes improve the performance of deep networks to some extent but the method to train a deep network is still a challenging issue to investigate.

To address this problem, we design a hybrid convolution network which is composed of predefined and learnt parameters. In this way, the number of free parameters reduces sharply and therefore we can train it effectively just through common methods such as back propagation algorithm.

4.2.2 The Architecture of The Hybrid Convolution Network

A hybrid convolution network is constructed by cascading a scattering convolution network discussed in Chapter 3 with a common convolutional neural network. The architecture of the hybrid convolution network is shown in Fig. 4–3.

A hybrid convolution network can be divided into two parts. The first part, called scattering part, is a refined scattering convolution network, which is composed of predefined complex wavelet filter banks and the modulus operator. Compared to scattering convolution networks discussed in Chapter 3, we abandon the step of calculating scattering coefficients and take advantage of modulus of wavelet coefficients directly based on the consideration of its application to image super resolution, which cares more about high frequency components. However, if the hybrid convolution network is applied to recognition or classification, scattering coefficients are necessary. The

second part of the hybrid convolution network, called convolution part, is a convolutional neural network, and its filter banks are learnt from data by employing back propagation algorithm.

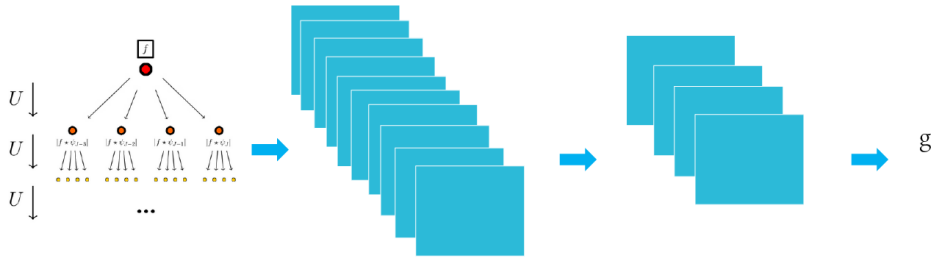


Figure 4–3 The architecture of the hybrid convolution network.

According to the energy conversation property discussed in section 3.5, the depth of scattering part is only three. All of the filter coefficients are defined by complex Morlet wavelet with J scales and L directions. The coefficients are calculated by fast algorithm along frequency decreasing pathes. The role of scattering part is to take advantages of scattering transform to extract features from data. According to the frequency domain analysis of scattering transform, we know that each feature map describes a specific part information of the frequency domain of the input. Due to the multiscale property of scattering transform, the scattering part can effectively capture the intrinsic multiscale property of signals.

The convolution part is composed of two layers and takes the modulus of wavelet coefficients of each layer as input. The first layer of the convolution part has $n_0 \times n_1$ filters and the size of each filter has $m_1 \times m_1$ parameters, where n_0 is the number of input feature maps, and n_1 is the number of neurons of the first layer, and m_1 is the spatial size of each filter . We use $W_1(n_0 \times m_1 \times m_1 \times n_1)$ to denote filter parameters of the first layer and B_1 to donate biases. Each neuron of the first layer employs the ReLU as the non-linearity operator. Therefore, the operation of the first layer is:

$$f_1(Y) = max(0, W_1 * X + B_1). \tag{4–9}$$

The second layer of the convolution part takes the output of the first layer as input and

produces the final output. There are only one neuron of the second layer with the size of each receptive field being $m_2 \times m_2$. Therefore, the size of the convolution kernel of the second layer is $n_1 \times m_2 \times m_2$ denoted by W_2 and the bias is just a real number represented by B_2. The operation of the second layer can be expressed as follows:

$$f_2(Y) = W_2 * f_1(Y) + B_2. \tag{4–10}$$

The role of the convolution part varies with different applications. For inverse problems such as image super-resolution and image denoising, the convolution part can be viewed as a reconstruction from the scattering representation. For recognition and classification problem, the role of convolution part is to further extract features from scattering representations. In all cases, the convolution part complements the limited expressiveness and increase flexibility by taking advantage of back propagation algorithm to be adaptive to specific task.

The hybrid convolution network provides a novel scheme to design the structure of convolutional neural networks. On the one hand, we can involve prior knowledge into the network and thereby achieve desirable performance by using predefined filter banks. On the other hand, the learnt filters can be adaptive to data and thereby can capture the complex structure hidden in data. In addition, the hybrid convolution also provides a novel training scheme. By predefining some of filter coefficients, we can reduce the number of free parameters and thereby can decrease the complexity of the network.

Form both theoretical and application perspectives, the hybrid convolution network is effective and powerful. The application to image super-resolution is discussed in the following chapter, where we will present that our scheme achieves state-of-the-art performance.

4.2.3 Analysis From Sparse Representation Perspective

In Chapter 2, we have investigated two kinds of dictionaries used for sparse representation. From the sparse representation perspective, we will show that the hybrid convolution network also provides a scheme to incorporate two kinds of dictionaries and takes advantage of superiorities from both sides.

On the one hand, the scattering part can be viewed as an analytic dictionary. Scattering transform, as a special case of wavelet transform, implements sparsity by employing localized and oriented complex wavelet. Compared to real wavelets, complex wavelets contain more information by its phase component. The modulus operator achieves a non-linear mapping which leads to a frequency spectrum shift in frequency domain. By a cascade of these complex wavelet decompositions and modulus operator, the frequency spectrum of the input signal has been divided into sectors. Therefore, filters of different layers and different positions of the same layer can process features of different frequency components —different scales. Because all of these parameters are predefined instead of learnt from data, it is convincing to take the scattering part as an analytic dictionary.

On the other hand, the convolution part can be considered as a learnt dictionary. Similar to other learnt dictionaries, all of the parameters of the convolution part are learnt from data, which results in that the convolution part can be adaptive to specific task and data. Through learning, the convolution part compensates the limited ability of analytic dictionaries to capture the complex structure of big data.

In addition, the hybrid convolution network inherits the advantage of analytic dictionaries, i.e., easy to implementation, through its end-to-end structure, which avoids seeking sparse representation of signals explicitly.

In summary, the hybrid convolution network, which both has the ability to capture the intrinsic multiscale property of signals and has adaptivity and flexibility, incorporates two kinds of dictionaries effectively. With advantages of both sides, the hybrid convolution network provides a promising scheme to previous applications based on sparse representation. We will just investigate its application to image super-resolution and its state-of-the-art performance will be presented in Chapter 5.

Chapter 5

Application to Image Super-Resolution

In this chapter, we will apply the hybrid convolution network to image super-resolution and compare its performance with that of prior art. First, we will introduce some previous methods which are based on sparse representation. Then we will introduce two toolboxes used in our experiment and followed by presenting the implementation detail of our experiment. Finally, the performance will be presented and compared.

5.1 Prior Art

In this section, we will survey some of achieving state-of-the-art performance methods, based on sparse representation, and introduce their implementations briefly.

5.1.1 Joint Dictionary Training for Image SR

Based on the assumption that image patches can be well-represented as a sparse linear combination of elements from an appropriate chosen over-complete dictionary and that under mild condition the sparse representation can be correctly recovered from the downsampling signals, J.Yang $et\ al.$ [27] trained a joint dictionary to implement image SR. By enforcing the sparse representation of a low-resolution image patch and that of the corresponding high-resolution image patch are the same, they jointly trained two dictionaries for the low-resolution and high-resolution image patches.

To be concrete, let x_i, y_i denote high-resolution and corresponding low-resolution image patch respectively, D_x and D_y represent high-resolution and low-resolution dictionaries, and α_i describe the sparse representation of x_i, y_i over each dictionary. The problem is to train the dictionaries from datasets. However, training such different dictionaries simultaneously is so difficult that they group two dictionaries into a joint

dictionary, as follows:

$$\bar{x_i} = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \bar{D} = \begin{bmatrix} D_x \\ D_y \end{bmatrix}. \tag{5–1}$$

The training stage can be divided into two stages. First, they seek to find the sparse representation α_i of the $\bar{x_i}$ with the joint dictionary $\bar{D}$ fixed. And then with the fixed sparse representation α_i, they update the joint dictionary $\bar{D}$. By iterating such procedure, it will converge. Once the joint dictionary has been trained, a high-resolution image can be obtained by finding the sparse representation of input low-resolution image and then reconstructing it by the high-resolution dictionary D_y.

5.1.2 The K-SVD Method for Image SR

Similar to the joint dictionary method, this approach is also based on the sparse representation. Roman Zeyde $et\ al.$ [46] took a similar framework as the joint dictionary method and made some modifications. Above all, they also assumed that a low-resolution image patch and its high-resolution counterpart share a common sparse representation. In stead of directly using features of low-resolution patches to train low-resolution dictionary D_x, they applied the Principal Component Analysis (PCA) algorithm on these features to reduce dimension because the low-resolution patches could not occupy the full space as the high-resolution ones. Through such a dimensionality reduction, computations can be saved. Then they first learnt the low-resolution dictionary D_x instead of the joint dictionary by employing K-SVD dictionary training procedure discussed in section 2.3.3. After obtaining the low-resolution dictionary, they solved the high-resolution dictionary D_y analytically by finding the inverse of matrix.

Once having two dictionaries, image super-resolution can be achieved as follows:

1. Scale the input image by the upscaling factor by bicubic interpolation;

2. Extract features from low-resolution images;

3. Extract patches from these features and reduce their dimensionality by multiplying them with the projection operator.

4. Find the sparse representation.

5. Reconstruct the high-resolution image by the high-solution dictionary.

5.1.3 A Convolutional Network for Image SR

Chao Dong $et\ al.$ [38] designed a deep convolutional neural network for image super-resolution. This network is composed of three layers. The first layer is used to extract patches from the low-resolution image and represent each patch as a high-dimensional vector, which comprises a series of feature maps. The function of the second layer is to map each high-dimensional vector onto another high-resolution vector. And each mapped vector is conceptually the representation of the corresponding high-resolution patch. The third layer, aggregating the above high-resolution patchwise representations to generate the final high-resolution image, is employed to reconstruction.

Neurons of the first and second layer employ the ReLU non-linear operator and there is no pooling operator in this network. The training is implemented by employing gradient descent algorithm with the MSE loss function. Once the network has been trained, a high-resolution image can obtained just by feeding a low-resolution image to the network.

5.2 Introduction to ScatNet

ScatNet is a MATLAB scientific toolbox, developed by Laurent Sifre $et\ al.$, which implements the scattering transform discussed in chapter 3 and provides a tool to take advantage of scattering transform to further investigation and application. In order to demonstrate implementation details of our experiment to be discussed in section 5.4, we excerpt and list some descriptions of the functions used in our experiment from the instruction of ScatNet [47] in the following subsections.

5.2.1 The Implementation of The Scattering Transform

Assuming that the cell array of linear operators Wop is already built, the computation of the scattering transform of some input signal x is a fast operation, supported by the function *scat*. Interestingly, the array x may either be of dimensions: $N \times 1$, $N_1 \times N_2$, $N_1 \times 1 \times N_2$. In all cases, the network of scattering coefficients is obtained

through the single command :

$$S = scat(x, Wop).$$

Each layer S_m of scattering coefficients proceeds from an averaging of modulus coefficients U_{m-1} computed at the previous layer. These modulus coefficients can be separately obtained with scat, as an optional second output argument :

$$[S, U] = scat(x, Wop).$$

Each operator $Wop\{1 + m\}$ performs two actions, leading to separate outputs : 1. an energy averaging according to the largest scale, by means of a low-pass filter ϕ, and 2. an energy scattering along all scales, by means of band-pass filters ψ_j .After initializing U_0 to x, *scat* executes the loop to calculate scattering coefficients S and wavelet coefficients U of each layer.

The layer format of each $S\{1 + m\}$ and $U\{1 + m\}$ consists of two fields, namely signal and meta. The former is a cell array of real valued signals, while the latter is a structure containing meta-information.

5.2.1.1 Translation-Invariant Representations of Images

The cell array of linear operators Wop and filter banks are produced by calling the function wavelet_factory_2d . It bears the following prototype:

$$[Wop, filters] = wavelet\_factory\_2d(size(x), filt\_opt, scat\_opt).$$

The first output argument is a cell array of function handles, which is the parameter of the prototype of *scat*. The second output is not mandatory for the processing of x but it is useful for the visualisation of the filter bank themselves and helpful to understand the role of each filter. In addition, ScatNet provides another wavelet factory for images, called wavelet_factory_2d_pyramid, which is implemented by taking advantage of a fast algorithm similar to the Mallat algorithm. While pursuing the same goal as wavelet_factory_2d, it happens to be computationally faster, and does not require to be

given the size of the signal x as a first input. It bears the following prototype:

$$[Wop, filters] = wavelet\_factory\_2d\_pyramid(size(x), filt\_opt, scat\_opt).$$

In spite of these advantages, wavelet_factory_2d_pyramid is not as customizable as its counterpart, especially in terms of anti-aliasing and numerical approximations. As a matter of fact, these two functions mainly differ in the format of filters : while wavelet_factory_2d stores its filters in the Fourier domain, wavelet_factory_2d_pyramid uses their spatial form.

5.2.1.2 Roto-Translation Invariant Representations of Images

Finally, the function wavelet_factory_3d enables a scattering representation of images which is invariant to both translations and rotations. With wavelet_factory_3d, at the first order, the energy is scattered along the spatial dimensions of the image, similarly to wavelet_factory_2d. At higher orders, however, wavelet_factory_3d computes a one-dimensional wavelet transform along the orientations of the previous coefficients. Therefore, for a scattering transform of order 2, each coefficient bank bears three indices vertical scale, horizontal scale, and angle hence the "3d" denomination. It has the following prototype:

$$[Wop, filters, filters\_rot] = wavelet\_factory\_3d(size(x), \cdots \\ filt\_opt, filt\_rotopt, scat\_opt).$$

Likewise the translation-invariant case, ScatNet provides an alternative built-in factory for wavelet_3d, which relies on the cascade algorithm.

$$[Wop, filters, filters\_rot] = wavelet\_factory\_3d\_pyramid(filt\_opt, \cdots \\ filt\_rotopt, scat\_opt).$$

Contrary to what their name might suggest, neither wavelet_3d nor wavelet_3d_pyramid is adapted to an input array x of size $(N1 \times N2 \times N3)$. Up to now, there is no built-in factory for the processing of three-dimensional signals, such as video clips or voxel-based measurements. Nevertheless, it is not difficult to derive a custom pipeline that

matches the geometric dimension of the data, in a similar fashion to the available factories.

5.2.1.3 Arguments

The scat_opt, as a argument of wavelet factory, offers a high degree of flexibility for the user. Here, we will review the available fields in scat opt according to the chosen factory. For instance, one may activate the single floating-point precision by the following instruction :

$$scat\_opt.precision =' single'.$$

If *scat_opt* is an empty data structure or is omitted, all these fields are set to their default values. If the user provides an additional field to scat opt or spell one of the available fields mistakenly, ScatNet will throw an error. The fields of scat_opt are listed as follows:

scat_opt.M: a positive integer, encodes the maximal order of the scattering transform, i.e. the depth of the associated scattering network.When set to 1, the scattering transform is merely the modulus of a wavelet transform. By default, it is equal to 2, whatever be the chosen built-in factory. In most cases, increasing scat opt.M marginally improves experiment results at the cost of a great computation.

scat_opt.oversampling: a positive integer, encodes the $log2$ of the desired oversampling factor. By default, scat opt.oversampling is equal to 1, which means that the scattering coefficients are computed right at the critical sample rate, and that no oversampling is carried out at all. Although oversampling S_x may be useful for post-processing or visualisation, note that the required memory increases exponentially with the integer scat opt.oversampling.

scat_opt.x resolution: a integer, encodes the $log2$ of the resolution of the input signal x, with respect to the finest resolution of the filter bank. By default, scat opt.x resolution is equal to 0. In addition, increasing scat_opt.x resolution is equivalent to an interpolation of x, whereas decreasing it below 0 is equivalent to a downsampling operation. Likewise the parameter scat_opt.oversampling (see above), the required memory increases exponentially with the integer scat_opt.oversampling. In summary,

relatively to the scattering transform, the integer scat opt.x resolution behaves as a prior interpolation, while scat_opt.oversampling behaves as a posterior interpolation.

scat_opt.precision: a string, encodes the numerical precision of the scattering coefficients. Its default value is 'double', which means that real numbers occupy 8 bytes (64 bits) of computer memory, in accordance with the default format in MATLAB. In order to alleviate computational load, the floating-point precision may be manually downgraded to 4 bytes by setting scat_opt.precision to 'single'. However, note that this choice may damage the discriminative performance of the resulting machine learning algorithms.

5.2.2 Filter Banks

Filter banks correspond to predefined sets of filters. The nature of the used filters is highly important since it determines the result of the scattering transform. Filters bank can be created using the wavelet factory set of functions.

5.2.2.1 Morlet Filters Bank

The Gabor wavelets consist of Gaussian envelopes modulated by complex exponentials to cover the entire frequency spectrum. Morlet wavelets are derived from these by subtracting the envelope multiplied by a constant such that the integral of the filter equals zero. A 2D Morlet filter bank consists of a Gaussian window

$$\phi_i(u) = 2^{-2j/Q}\phi(2^{-j/Q(u,v)})$$

and dilated and rotated Morlet filters

$$\psi_{j,\theta} = 2^{-2j/Q}\psi(r_\theta 2^{-j/Q}(u,v)).$$

And The mother Morlet filter is defined by

$$\psi(u,v) = e^{\frac{u^2+s^2v^2}{2\sigma^2}}(K - s^{iu\xi}),$$

where σ is the spread of the gaussian envelope, s the eccentricity of the elliptical gaussian envelope, and ξ the frequency of the oscillatory exponential. The full filter bank

can be obtained by calling the function morlet_filter_bank_2d.

5.2.2.2 Shanon Filters Bank

Shanon wavelets are separable wavelets with a a finite support in frequency. Thus they suffer from a bad localization in time since their Fourier transform is not regular. Here they are designed to show filter banks that respects the Littlewood paley condition. Shanon mother wavelet of the Shanon filters bank is defined by

$$\widehat{\psi}(u,v) = e^{i2\pi(u+v)}.$$

In our implementation, these wavelets are set so that the Littlewood paley condition is satisfied.

5.3 Introduction to MatConvNet

MatConvNet, developed by Andrea Vedaldib and Karel Lenc $et\ al.$ [48], is a MATLAB toolbox of convolutional neural networks. The design of toolbox emphasizes simplicity and flexibility. MatConvNet implements most of the components of convolutional neural networks, and makes each component as a building block, such as convolution, pooling, and ReLU. Therefore, it is easy and convenient to use, and can be applied to design various convolution neural networks for different tasks. In addition, the training of convolution neural networks is dependant on a great number of data and thereby needs a lot of calculations. Consequently, the support of efficient computation on GPU is necessary. Fortunately, MatConvNet can work well both on CPU and GPU. In the following subsections, we will introduce the use of common building blocks of convolutional neural networks, which are used in our experiment.

5.3.1 Computational Blocks

5.3.1.1 Convolution

The convolutional operator is implemented by the function vl_nnconv, which computes the convolution of the input signal x with a bank of k multi-dimensional filters f and biases b. The convolution operator can produce a feature map with the

same size of image by padding. Assume that the input x has width W and that the filter f has width $W^{'} \leq W$. Then there are

$$W^{''} = W - W^{'} + 1$$

possible translations of the filters in the horizontal direction such that the filter is entirely contained in the input x. Hence, by default the filtered signal y has width $W^{"}$. However, if we pad the signal with parameters $[P_t; P_b; P_l; P_r]$, the effect of which is to virtually pad with zeros the signal x in the top, bottom, left, and right spatial directions respectively, we can adjust the size of the output to be the same as that of input. In addition, some of layers are fully connected in some applications. MatConvNet addresses this case by setting $W^{''} = 1$ instead of providing another function.

For additional flexibility, vl_nnconv allows to group input feature channels and apply to them different filter groups. To to do so, specify as input a bank of K filters such that $D^{'}$ divides the number of input dimensions D. These are treated as $g = D/D^{'}$ filter groups.

5.3.1.2 Pooling

The pooling operator is implemented by function vl_nnpool, which contains max pooling and sum pooling. The max pooling calculates the maximum response within a neighborhood for each channel. And the sum pooling produces the sum of response within a neighborhood. Function vl_nnpool also supports padding the input. For max pooling, it is equivalent to extending the input data with $-\infty$, while it is to pad the signal with zeros for sum pooling.

5.3.1.3 ReLU

The Rectified Linear Unit (ReLU), a non-linear operator, is implemented by function vl_nnrelu, which computes :

$$y_i = max\{0, x_i\}.$$

5.4 Experiment Design

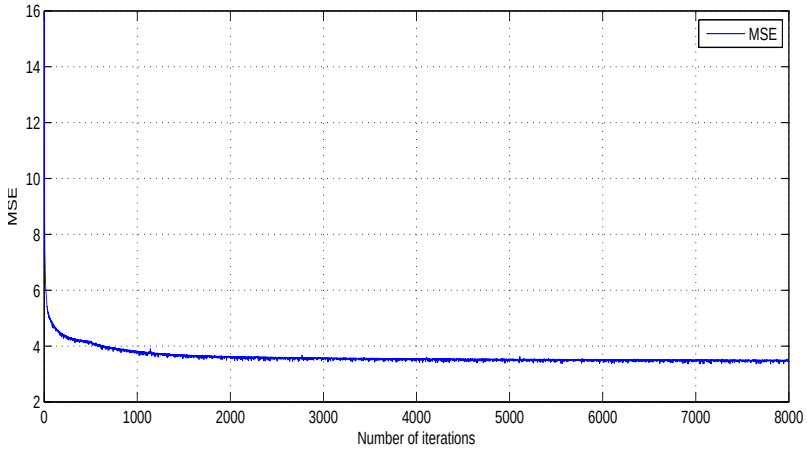
In this section, we apply the hybrid convolution network discussed in chapter 4 to image super resolution and analyse the performance with state-of-the-art methods.

Configuration. Parameters of the hybrid convolution network discussed in chapter 4 are set as follows. For the scattering part, scattering wavelet is chosen as the complex Morlet wavelet and the number of scale J and orientation L is set to be 3 and 4 respectively. Through the fast algorithm of scattering transform, there are 125 feature maps fed into the convolution part namely $n_0 = 125$. And the number of neuron of the first layer n_1 is 50. The size of filters of the first layer m_1 is 9 and the size of the second layer m_2 is 5. Employing this configuration, we train three hybrid convolution networks corresponding to upscaling factor $\in \{2, 3, 4\}$ respectively.

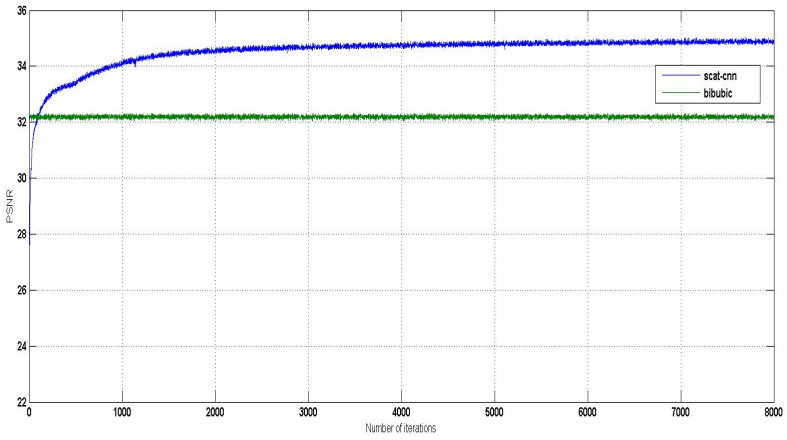
Datasets. For a fair comparison with traditional learning-based schemes, we employs the same training set, test sets as in [25–27, 38]. Concretely, the training set is composed of 91 natural images and the testing set contains Set5 (5 images) and Set14 (images), which are totaly different from the 91 training images. Similar to previous work, the Set5 is used to evaluate the performance of upscaling factors 2,3, and 4, and the Set14 is used for upscaling factor 3.

Training procedure. The training process is implemented on the MATLAB platform with ScatNet and MatConvNet toolboxes, which have been introduced in previous sections. The low-resolution images are obtained by first blurring the original image with Gaussian kernel, then down sampling it by the upscaling factor, and upscale it by the same upscaling factor through bicubic interpolation. For fast convergence, we subtract the mean of each image from the image and then scale it to have a standard deviation 1. Similar to [25–27, 38], we only care about illuminance channel for color images because humans are more sensitive to illuminance changes. To avoid border effect, there is no padding for all of the convolutional layers. For fair comparison, the results of other methods are also cropped likewise.

Let X_i denote the ground truth high-resolution image, Y_i describe the corresponding low-resolution image, and $F(Y_i; \theta)$ represent the high-resolution image produced by the hybrid convolution network. We choose mean squared error (MSE) as the loss



(a) The MSE for upscaling factor 2 during the training process.



(b) The PSNR for upscaling factor 2 during the training process.

Figure 5–1 Training results for upscaling factor 2.

function:

$$L(\theta) = \frac{1}{m}\sum_{n=1}^{m} \|F(Y_i;\theta) - X_i\|^2. \tag{5–2}$$

We use back propagation algorithm to calculate the derivatives and employ stochastic gradient descent with mini-batch algorithm to optimize parameters. The filter weights are initialized by randomly generating from a Gaussian distribution with zero mean and standard deviation 0.02 and the biases are all set to be zeros. Fig. 5.1(a), Fig. 5.1(b) shows the change of the MSE, PSNR for upscaling factor 2 during the training

process respectively. The performance of our model enhances quickly at first and then gradually tends to converge. According to Fig. 5.1(b), it is easy to find the performance of our model overpass that of bicubic just after a few iterations.

5.5 Experiment Results

We evaluate the results from both visually and qualitatively in Peak Signal to Noise Ratio (PSNR) and compare with previous methods which achieve state-of-the-art performance. The implementations of these methods are all from the publicly available codes provided by the authors.

Table 5–1 presents the results on Set5 test images. From the table, we can easily find that the proposed method achieves the highest average PSNR for all scales. On the larger test dataset Set14, the proposed method also overperforms all of the other methods, which is shown in Table 5–2. Although for a few images of the testing set our model does not achieve the highest PSNR, our visual results of these images are still appealing and satisfying, which can be observed from Fig. 5–2, Fig. 5–4. In addition, with the increment of number of iterations, our model will achieve better performance. Moreover, more training data may also enhances the performance effectively due to the prior knowledge of machine learning.

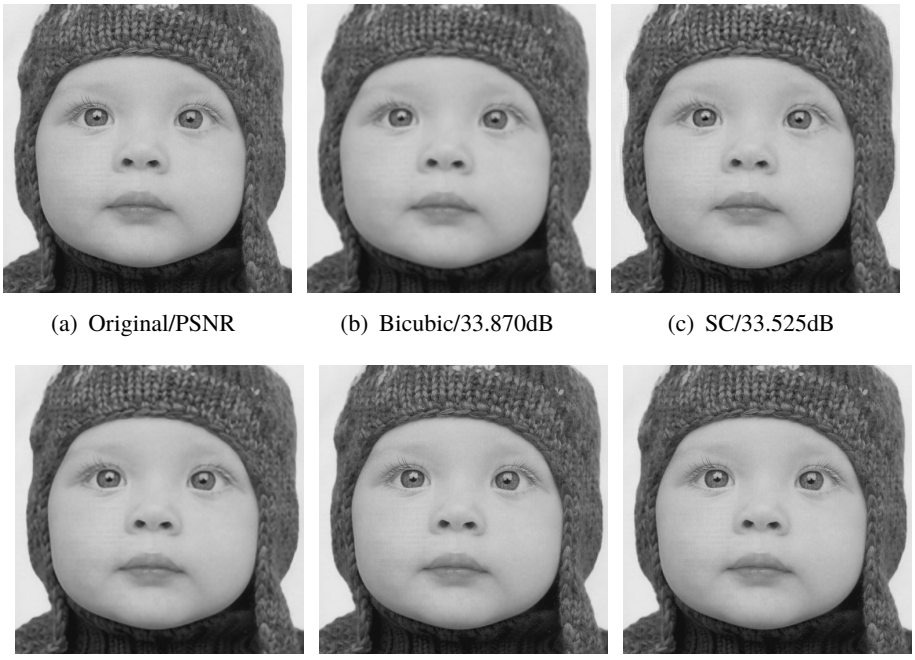
Fig. 5–2 $\sim$ Fig.5–7 presents the visual results of different methods on the test set Set5 and Set14 for upscaling factor 3. It is easy to find that the proposed method also produces better visual results with sharper edges and less artifacts.

Table 5–1 The result of PSNR (dB) on the Set5 dataset.

Set5	Scale	Bicubic	SC [27]	K-SVD [46]	CNN[38]	Proposed
baby	2	37.005	-	-	38.238	**38.320**
bird	2	36.849	-	-	40.663	**41.030**
butterfly	2	27.446	-	-	**32.310**	32.284
head	2	34.804	-	-	35.589	**35.648**
woman	2	32.230	-	-	34.977	**35.298**
average	2	33.667	-	-	36.355	**36.516**
baby	3	33.870	33.525	**35.050**	34.968	34.924
bird	3	32.648	33.170	34.662	35.052	**35.477**
butterfly	3	24.064	24.892	26.036	**27.677**	27.369
head	3	32.824	33.025	33.529	33.499	**33.728**
woman	3	28.654	29.197	30.434	31.007	**31.261**
average	3	30.412	30.762	31.942	32.441	**32.552**
baby	4	31.752	-	-	**32.955**	32.837
bird	4	30.200	-	-	32.012	**32.280**
butterfly	4	22.129	-	-	**25.151**	24.710
head	4	31.548	-	-	32.149	**32.365**
woman	4	26.524	-	-	28.249	**28.737**
average	4	28.431	-	-	30.103	**30.186**

Table 5–2 The result of PSNR (dB) on the Set14 dataset.

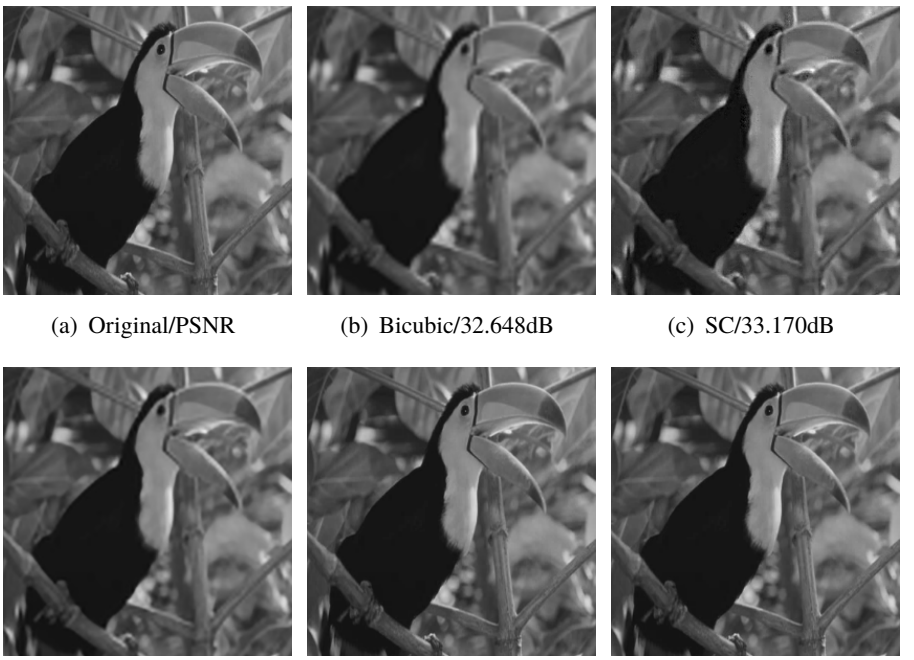
Set14	Scale	Bicubic	SC [27]	K-SVD [46]	CNN [38]	Proposed
baboon	3	23.210	23.330	23.527	23.605	**23.606**
barbara	3	26.191	26.072	**26.726**	26.597	26.557
bridge	3	24.427	24.548	25.039	25.104	**25.196**
coastguard	3	26.715	26.886	27.175	27.196	**27.200**
comic	3	23.045	23.450	23.896	24.307	**24.371**
face	3	32.759	33.006	33.487	33.525	**33.712**
flowers	3	27.151	27.602	28.359	28.894	**28.973**
foreman	3	31.667	32.583	33.771	34.339	**34.742**
lenna	3	31.618	32.017	32.966	33.344	**33.418**
man	3	26.978	27.366	27.880	28.150	**28.262**
monarch	3	29.348	29.930	31.047	**32.316**	32.165
pepper	3	32.435	32.885	34.128	34.447	**34.657**
ppt3	3	23.619	24.309	25.126	25.933	**26.069**
zebra	3	26.564	27.005	28.460	28.818	**28.920**
average	3	27.552	27.928	28.685	29.041	**29.132**



(a) Original/PSNR (b) Bicubic/33.870dB (c) SC/33.525dB

(d) K-SVD/**35.050dB** (e) CNN/34.968dB (f) Proposed/34.924dB

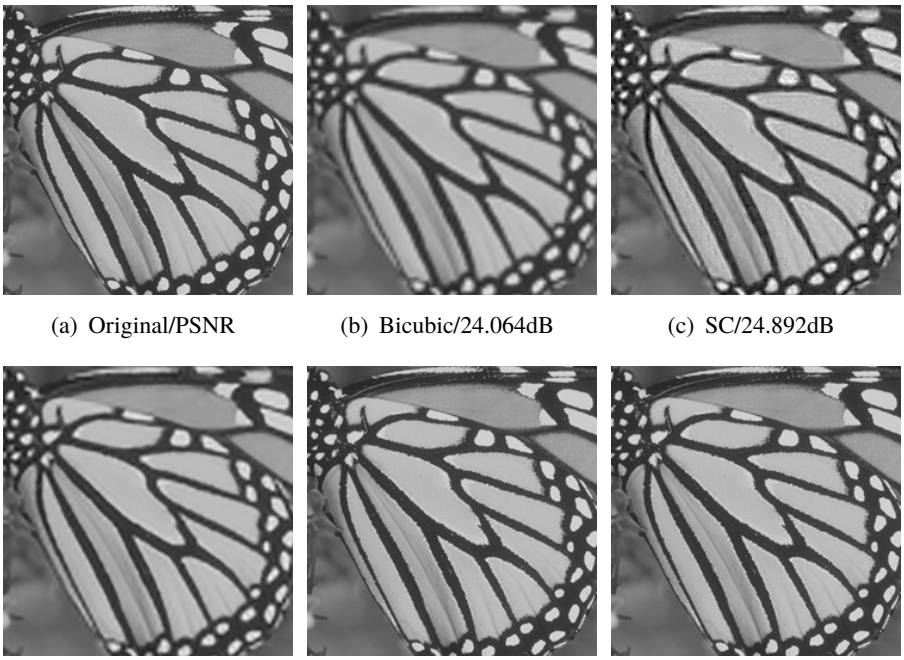
Figure 5–2 “Baby” image from Set5 for upscaling factor 3.



(a) Original/PSNR (b) Bicubic/32.648dB (c) SC/33.170dB

(d) K-SVD/34.662 dB (e) CNN/35.052dB (f) Proposed/**35.477dB**

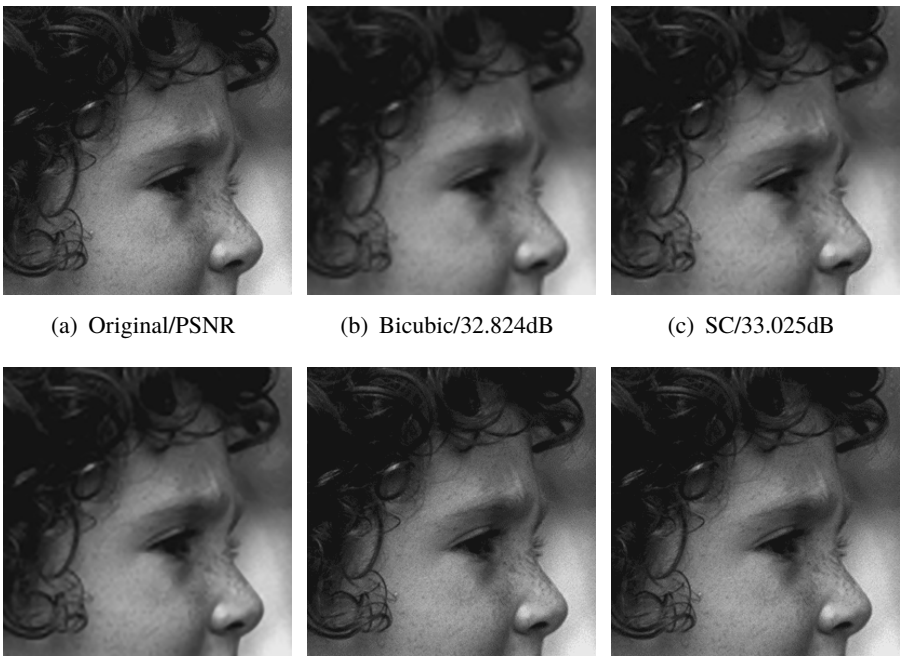
Figure 5–3 “Bird” image from Set5 for upscaling factor 3.



(a) Original/PSNR (b) Bicubic/24.064dB (c) SC/24.892dB

(d) K-SVD/26.036dB (e) CNN/**27.677dB** (f) Proposed/27.369dB

Figure 5–4 “Butterfly” image from Set5 for upscaling factor 3.



(a) Original/PSNR (b) Bicubic/32.824dB (c) SC/33.025dB

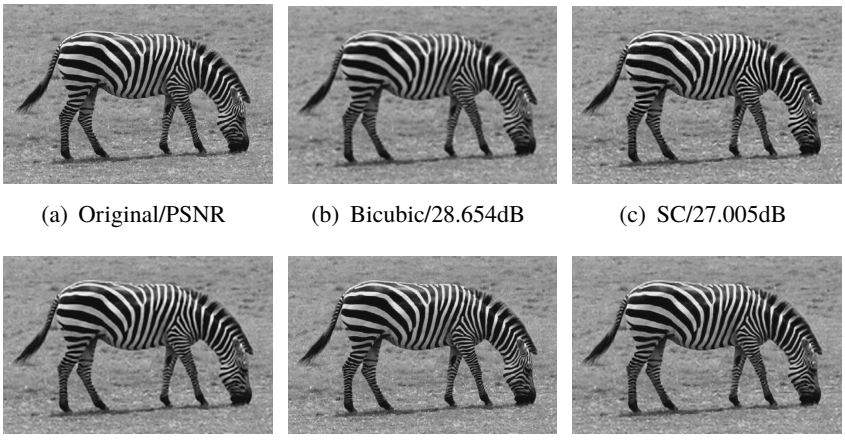
(d) K-SVD/33.529dB (e) CNN/33.499dB (f) Proposed/**33.728dB**

Figure 5–5 “Head” image from Set5 for upscaling factor 3.

(a) Original/PSNR (b) Bicubic/28.654dB (c) SC/29.197dB

(d) K-SVD/30.434dB (e) CNN/31.007dB (f) Proposed/**31.261dB**

Figure 5–6 “Woman” image from Set5 for upscaling factor 3.



(a) Original/PSNR (b) Bicubic/28.654dB (c) SC/27.005dB

(d) K-SVD/28.460dB (e) CNN/28.818dB (f) Proposed/**28.920dB**

Figure 5–7 “Zebra” image from Set14 for upscaling factor 3.

Chapter 6

Conclusion

In this paper, we propose a hybrid convolution network which is composed of scattering convolution network and convolutional neural network, and investigate its application to image super-resolution, which achieves state-of-the-art results. From convolutional neural networks perspective, we provide a scheme to design the structure of networks, which can effectively involve prior knowledge to networks and bring in desirable performance. In addition, since some of filter coefficients are predefined, the number of free parameters is reduced and thereby reduce the dependance on data, which provides a solution to train deep networks. With regard to sparse representation, the hybrid convolution can been viewed as an effective combination of the analytic dictionary and the learnt dictionary. Through the scattering part, the hybrid convolution network can capture the intrinsic multiscale property of the input signal and obtain its sparse representation. The following convolution part can be adaptive to specific data and thereby compensates the limited ability to capture the distinct structure of specific data. Therefore, its novel structure provides a scheme to incorporate two kinds of dictionaries and take advantage of superiorities from both sides. The hybrid convolution network is easy and fast to implement once having been trained because of its end-to-end structure. For image super resolution, our method provides another approach to achieve state-of-the-art performance and also enhances current performance.

Scattering convolution networks are constructed on the foundation of complex directional wavelets. However, contourlet transform is generally thought as the "true" two-dimensional transform capturing the intrinsic geometrical structure. Therefore, optimizing the scattering convolution network according to contourlet transform is promising. In addition, with the development of deep learning, there are a series of novel building blocks such as auto-encoder and restricted boltzmann machine (RBM), which can take place of convolution neural networks. Therefore, superior structure

of networks remains to be design. Moreover, due to time limit, we only apply the hybrid convolution network to one task—image super-resolution. Its applications to other fields in computer vision, such as classification, image denoising and object recognition, are still interesting and appealing.

Bibliography

[1] CANDÈ E J, WAKIN M B. An introduction to compressive sampling[J]. Signal Processing Magazine, IEEE, 2008, 25(2):21–30.

[2] BRUCKSTEIN A M, DONOHO D L, ELAD M. From sparse solutions of systems of equations to sparse modeling of signals and images[J]. SIAM review, 2009, 51(1):34–81.

[3] BRUNA J, MALLAT S. Invariant scattering convolution networks[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2013, 35(8):1872–1886.

[4] SIFRE L, MALLAT S. Rotation, scaling and deformation invariant scattering for texture discrimination[C]//Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. .[S.l.]: [s.n.] , 2013:1233–1240.

[5] OLSHAUSEN B A, FIELD D J. Sparse coding with an overcomplete basis set: A strategy employed by V1?[J]. Vision research, 1997, 37(23):3311–3325.

[6] DONOHO D L. Compressed sensing[J]. Information Theory, IEEE Transactions on, 2006, 52(4):1289–1306.

[7] CANDÈS E J, ROMBERG J, TAO T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information[J]. Information Theory, IEEE Transactions on, 2006, 52(2):489–509.

[8] MALLAT S. A wavelet tour of signal processing: the sparse way[M].[S.l.]: Academic press, 2008.

[9] DAUBECHIES I, et al. Ten lectures on wavelets[M], Vol. 61.[S.l.]: SIAM, 1992.

[10] MEYER Y. Wavelets and operators[M], Vol. 1.[S.l.]: Cambridge university press, 1995.

[11] Grossmann A, Morlet J. Decomposition of Hardy functions into square integrable wavelets of constant shape[J]. SIAM journal on mathematical analysis, 1984, 15(4):723–736.

[12] Mallat S G. A theory for multiresolution signal decomposition: the wavelet representation[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1989, 11(7):674–693.

[13] Donoho D L. De-noising by soft-thresholding[J]. Information Theory, IEEE Transactions on, 1995, 41(3):613–627.

[14] Lewis A S, Knowles G. Image compression using the 2-D wavelet transform[J]. Image Processing, IEEE Transactions on, 1992, 1(2):244–250.

[15] Le Pennec E, Mallat S. Sparse geometric image representations with bandelets[J]. Image Processing, IEEE Transactions on, 2005, 14(4):423–438.

[16] Peyré G, Mallat S. Surface compression with geometric bandelets[J]. ACM Transactions on Graphics (TOG), 2005, 24(3):601–608.

[17] Candes E J, Donoho D L, et al. Curvelets: A surprisingly effective nonadaptive representation for objects with edges[M].[S.l.]: DTIC Document, 1999.

[18] Candes E, Demanet L, Donoho D, et al. Fast discrete curvelet transforms[J]. Multiscale Modeling & Simulation, 2006, 5(3):861–899.

[19] Do M N, Vetterli M. Contourlets: a new directional multiresolution image representation[C]//Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on. .[S.l.]: [s.n.] , 2002 , 1:497–501.

[20] Lu Y, Do M N. A new contourlet transform with sharp frequency localization[C]//Image Processing, 2006 IEEE International Conference on. .[S.l.]: [s.n.] , 2006:1629–1632.

[21] Do M N, Vetterli M. The contourlet transform: an efficient directional multiresolution image representation[J]. Image Processing, IEEE Transactions on, 2005, 14(12):2091–2106.

[22] Rubinstein R, Bruckstein A M, Elad M. Dictionaries for sparse representation modeling[J]. Proceedings of the IEEE, 2010, 98(6):1045–1057.

[23] Engan K, Aase S O, Hakon Husoy J. Method of optimal directions for frame design[C]//Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on. .[S.l.]: [s.n.] , 1999 , 5:2443–2446.

[24] Aharon M, Elad M, Bruckstein A. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation[J]. Signal Processing, IEEE Transactions on, 2006, 54(11):4311–4322.

[25] Yang J, Wang Z, Lin Z, et al. Coupled dictionary training for image super-resolution[J]. Image Processing, IEEE Transactions on, 2012, 21(8):3467–3478.

[26] Yang J, Wright J, Huang T, et al. Image super-resolution as sparse representation of raw image patches[C]//Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. .[S.l.]: [s.n.] , 2008:1–8.

[27] Yang J, Wright J, Huang T S, et al. Image super-resolution via sparse representation[J]. Image Processing, IEEE Transactions on, 2010, 19(11):2861–2873.

[28] Elad M, Aharon M. Image denoising via sparse and redundant representations over learnt dictionaries[J]. Image Processing, IEEE Transactions on, 2006, 15(12):3736–3745.

[29] Mairal J, Sapiro G, Elad M. Learning multiscale sparse representations for image and video restoration[R].[S.l.]: DTIC Document, 2007.

[30] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. .[S.l.]: [s.n.] , 2012:1097–1105.

[31] Lawrence S, Giles C L, Tsoi A C, et al. Face recognition: A convolutional neural-network approach[J]. Neural Networks, IEEE Transactions on, 1997, 8(1):98–113.

[32] JAIN V, SEUNG S. Natural image denoising with convolutional networks[C]//Advances in Neural Information Processing Systems. .[S.l.]: [s.n.] , 2009:769–776.

[33] LE CUN B B, DENKER J S, HENDERSON D, et al. Handwritten digit recognition with a back-propagation network[C]//Advances in neural information processing systems. .[S.l.]: [s.n.] , 1990.

[34] MALLAT S. Group invariant scattering[J]. Communications on Pure and Applied Mathematics, 2012, 65(10):1331–1398.

[35] TIPPING M E, BISHOP C M. Bayesian image super resolution[J]. 2006. US Patent 7,106,914.

[36] HARDIE R, BARNARD K, ARMSTRONG E. Joint MAP registration and high-resolution image estimation using a sequence of undersampled images[J]. Image Processing, IEEE Transactions on, 1997, 6(12):1621–1633.

[37] BAKER S, KANADE T. Limits on super-resolution and how to break them[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2002, 24(9):1167–1183.

[38] DONG C, LOY C C, HE K, et al. Learning a deep convolutional network for image super-resolution[M]//Computer Vision–ECCV 2014.[S.l.]: Springer, 2014:184–199.

[39] KINGSBURY N. Complex wavelets for shift invariant analysis and filtering of signals[J]. Applied and computational harmonic analysis, 2001, 10(3):234–253.

[40] ENGAN K, AASE S O, HUSØY J H. Frame based signal compression using method of optimal directions (MOD)[C]//Circuits and Systems, 1999. ISCAS'99. Proceedings of the 1999 IEEE International Symposium on. .[S.l.]: [s.n.] , 1999 , 4:1–4.

[41] LESAGE S, GRIBONVAL R, BIMBOT F, et al. Learning unions of orthonormal bases with thresholded singular value decomposition[C]//Acoustics, Speech, and Signal

Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on. .[S.l.]: [s.n.] , 2005 , 5:v–293.

[42] Mairal J, Bach F, Ponce J, et al. Online learning for matrix factorization and sparse coding[J]. The Journal of Machine Learning Research, 2010, 11:19–60.

[43] Sallee P, Olshausen B A. Learning sparse multiscale image representations[C]//Advances in neural information processing systems. .[S.l.]: [s.n.] , 2002:1327–1334.

[44] Rubinstein R, Zibulevsky M, Elad M. Double sparsity: Learning sparse dictionaries for sparse signal approximation[J]. Signal Processing, IEEE Transactions on, 2010, 58(3):1553–1564.

[45] Hubel D H, Wiesel T N. Receptive fields and functional architecture of monkey striate cortex[J]. The Journal of physiology, 1968, 195(1):215–243.

[46] Zeyde R, Elad M, Protter M. On single image scale-up using sparse-representations[M]//Curves and Surfaces.[S.l.]: Springer, 2012:711–730.

[47] Andén J, Sifre L, Mallat S, et al. Scatnet[J]. Computer Software. Available: http://www. di. ens. fr/data/software/scatnet/.[Accessed: December 10, 2013], 2014.

[48] Vedaldi A, Lenc K. MatConvNet-convolutional neural networks for MATLAB[J]. arXiv preprint arXiv:1412.4564, 2014.

致　　谢

首先感谢熊红凯教授的选题。通过完成本次毕业论文，我对机器学习与计算机视觉有了初步的了解，并在利用梯度下降算法训练神经网络时真切的感受到了机器学习与人工智能的魅力与巨大潜力。在实验过程中，曾遇到各种纷杂不同的问题，感谢熊老师、实验室师兄师姐与同学的指导、帮助，与他们的交流使得我的实验最终得以顺利完成。当然，这其中有许多问题是必需要自己面对解决的，我也感谢自己的坚持与执著。最后，感谢上海交通大学，在这里我不仅仅学到了前沿的专业知识，还有很多，谢谢！