FOCNet: A Fractional Optimal Control Network for Image Denoising

Xixi Jia[1,2], Sanyang Liu[1], Xiangchu Feng[1] and Lei Zhang[*2,3]

[1]School of Mathematics and Statistics, Xidian University, Xi'an, China
[2]Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[3]DAMO Academy, Alibaba Group

hsijiaxidian@gmail.com, liusanyang@126.com, xcfeng@mail.xidian.edu.cn,
cslzhang@comp.polyu.edu.hk

Abstract

Deep convolutional neural networks (DCNN) have been successfully used in many low-level vision problems such as image denoising. Recent studies on the mathematical foundation of DCNN has revealed that the forward propagation of DCNN corresponds to a dynamic system, which can be described by an ordinary differential equation (ODE) and solved by the optimal control method. However, most of these methods employ integer-order differential equation, which has local connectivity in time space and cannot describe the long-term memory of the system. Inspired by the fact that the fractional-order differential equation has long-term memory, in this paper we develop an advanced image denoising network, namely FOCNet, by solving a fractional optimal control (FOC) problem. Specifically, the network structure is designed based on the discretization of a fractional-order differential equation, which enjoys long-term memory in both forward and backward passes. Besides, multi-scale feature interactions are introduced into the FOCNet to strengthen the control of the dynamic system. Extensive experiments demonstrate the leading performance of the proposed FOCNet on image denoising. Code will be made available.

1. Introduction

Image denoising aims to estimate the underlying clean image from its noisy observation. As an indispensable step in many digital imaging and computer vision systems, image denoising has been investigated for decades, while it is still an active research topic. A vast amount of methods

---

have been developed by using different mathematical tools and models, including partial differential equations (PDE) [39, 32], sparse coding [28, 13], low-rank approximation [16, 22], and others [6, 9, 38]. Most of these methods rely on very limited human knowledge or assumptions about the image prior, limiting their capability in recovering complex image structures.

In recent years, deep convolutional neural networks (DCNN) have achieved a great success in many low-level vision problems, including image denoising. In particular, Zhang *et al.* [45] proposed an effective image denoising network called DnCNN by integrating batch normalization into the residual learning framework, which outperforms traditional denoising algorithms by a noticeable margin. By adding symmetric skip connections, Mao *et al.* [29] constructed an improved encoder-decoder network for image denoising. Bae *et al.* [4] suggested to learn CNN on the wavelets sub-bands for image denoising. Based on the wavelet decomposition, Liu *et al.* [26] put forward a multi-level wavelet based denoising DCNN. Tai *et al.* [42] constructed a densely connected denoising network to enable memory of the network. Zhang *et al.* [46] introduced a fast and flexible network (FFDNet) which can process images with nonuniform noise corruption. To exploit the nonlocal property of the image features in DCNN, Plötz *et al.* [35] presented an N^3Net by employing the *k*-nearest neighbor matching in the denoising network.

Although various DCNN methods have been proposed for image denoising, the network design is mostly empirical without clear mathematical bases. Recently, some studies [34, 44, 27, 40, 17, 25] on the mathematical foundation of DCNN have revealed that the forward propagation of DCNN corresponds to a dynamic system, which can be characterized by an ODE and solved by optimal control methods [44, 25]. For example, Pineda [34] studied the

neural network from a viewpoint of dynamic systems, and formulated the forward propagation of neural networks as the discretization of a special ODE.

In [25], Li *et al.* studied the Residual Network (ResNet) [18] via optimal control and showed that the ResNet can be solved by Pontryagin's maximum principle. Lu *et al.* [27] found that the forward propagation of ResNet is the Euler discretization of an ODE. They further concluded that many state-of-the-art network structures can be considered as different discretizations of ODEs, such as FractalNet [24], PolyNet [47] and RevNet [15]. More recently, Ruthotto *et al.* [40] investigated the relation between DCNN and partial differential equation (PDE), and indicated that the forward process of DCNN resembles the diffusion equation. The PDE/diffusion models on one hand provide an alternative perspective to understand DCNN; on the other hand, they help to explain the success of DCNN for image denoising, since PDE/diffusion models have long been effective mathematical tools for developing image denoising algorithms [39, 32, 21].

The differential equations that indwell in the existing DCNNs are integer-order differential equations (IODE), which can only allow short-term feature interactions due to their short-term memory. In practice, the evolution of a system often depends on not only its current state but also its historical states [10, 3]. In optimal control, the long-term memory provides important information for robust control of linear and nonlinear systems [33]. Long-term memory is also beneficial for vision problems such as image denoising [42], since it can preserve better mid/high-frequency information. Therefore, it is not enough to employ only IODE for designing advanced denoising DCNN. Although some DCNNs have been designed to address the long-term memory problem, such as DenseNet [19] and MemNet [42], there still lacks solid theoretical analysis on how the memory is exploited.

It has been found that in a majority of systems such as biological systems, electromagnetic fields and Hamiltonian systems, the long-term memory holds in a certain mode which can be characterized by the power-law [11]. It has also been found that the corresponding memory systems could be described by the fractional-order differential equations (FODE) [10]. The FODE was developed to mitigate the limitations of IODE [5, 31, 36] not only in the memory but also in many other aspects such as the stability of the system. It has been shown that FODE can more accurately describe a lot of dynamic systems than IODE [7].

In this paper, by solving a Fractional Optimal Control (FOC) problem, we naturally design an advanced image denoising network, namely FOCNet. The forward propagation of FOCNet is constructed by an explicit discretization of a FODE with control variables. The advantages of the FODE induced FOCNet over the IODE induced networks

are twofold: 1) The fractional-order FODE can describe the power-law memory mode which has been verified in many practical systems to persist memory; 2) The FOCNet has long-term memory not only in the forward process but also in the backward passes. Instead of characterizing the FODE based FOCNet on only one specific image scale, we further introduce a multi-scale strategy to strengthen the denoising network. Specifically, in the multi-scale model, different scale features propagate forward according to their corresponding FODE, and at the same time multi-scale feature interactions are allowed by a scale transform operator.

To sum up, the contributions of this work are:

- A novel denoising network – FOCNet – is presented by solving a FOC problem using FODE. FOCNet theoretically enjoys the advantages such as long-term memory and better stability.

- A multi-scale implementation of FODE is elaborated such that fine scale and coarse scale features in FOCNet can be simultaneously utilized to strengthen the denoising system.

Extensive experiments on image denoising are conducted to validate the effectiveness of FOCNet. The results show that FOCNet achieves leading denoising performance in both visual quality and quantitative measures.

2. Related work

In this section we briefly describe some ingredients of DCNN and optimal control relevant to our work. First, we outline the forward framework of DCNN, and its application to image denoising. Then we present the optimal control problem and its connection to DCNN.

2.1. The propagation of DCNN

The DCNN learns a highly nonlinear mapping from a large amount of labeled data by stacking multiple simple nonlinear units. Mathematically, the plain DCNN can be formulated as the following evolution process

$$\boldsymbol{u}_{t+1} = f(\boldsymbol{u}_t, \theta_t) \quad t = 1 \cdots T, \qquad (1)$$

where $\boldsymbol{u}_t \in \mathbb{R}^d$ is the input of the t-th layer of the network, $\boldsymbol{u}_{t+1} \in \mathbb{R}^d$ is the output of the t-th layer and $\theta_t \in \mathbb{R}^m$ is the parameters of the convolution kernel. The nonlinear unit is often modeled as $f(\boldsymbol{u}_t, \theta_t) = \sigma(\theta_t * \boldsymbol{u}_t)$ [1], where $\sigma : \mathbb{R}^d \to \mathbb{R}^d$ denotes the nonlinear activation function. After T layers evolution, a loss function is used to measure the distance between the output and the label. The optimal network parameters are obtained by minimizing the loss function and the regularization function of the parameters as

$$\min_{\{\theta_t\}_{t=1}^T} \sum_{t=1}^T R(\theta_t) + L(\Phi(\boldsymbol{u}_T), \boldsymbol{x}), \qquad (2)$$

---

[1] For simplicity, the bias term is omited.

where $\Phi : \mathbb{R}^d \to \mathbb{R}^n$ transforms the final layer features to the output, $L : \mathbb{R}^n \to \mathbb{R}$ is the loss function, $R : \mathbb{R}^m \to \mathbb{R}$ is the regularization function on the parameter θ_t and $\boldsymbol{x} \in \mathbb{R}^n$ is the label.

The residual network [18] improves the plain network by adding a skip connection as

$$\boldsymbol{u}_{t+1} = \boldsymbol{u}_t + f(\boldsymbol{u}_t, \theta_t) \quad t = 1 \cdots T. \tag{3}$$

Surprisingly, such a minor change has achieved remarkable success in a lot of computer vision and image processing applications [18].

The DCNN can be directly used for image denoising by setting the input as the transformation of the noisy image $\boldsymbol{u}_0 = \Psi(\boldsymbol{y})$, and label $\boldsymbol{x}$ be the corresponding clean image. For denoising, the loss function is often set to the l_2 loss as $L(\boldsymbol{u}_T, \boldsymbol{x}) = \frac{1}{2}\|\Phi(\boldsymbol{u}_T) - \boldsymbol{x}\|_2^2$. Due to the powerful learning ability, DCNN has been attracting considerable attention in image denoising [45, 42, 35] as we have introduced in Section 1.

2.2. The optimal control problem

The continuous counterpart of deep neural network is optimal control which has been well studied for hundreds of years with solid mathematical theories [14]. Denote by $\boldsymbol{u}_0 \in \mathbb{R}^d$ the initial condition of a dynamic system, the control of the system can be described by the following ordinary differential equation (ODE)

$$\begin{cases} \dot{\boldsymbol{u}}(t) = f(\boldsymbol{u}(t), \theta(t)), \\ \boldsymbol{u}(0) = \boldsymbol{u}_0, t \in [0, T], \end{cases} \tag{4}$$

where $\theta(t) : [0, T] \to \Theta \subset \mathbb{R}^d$ is the control parameter function also called a control [14]. The trajectory $\boldsymbol{u}(t)$ is regarded as the corresponding response of the system.

PAYOFFS. The overall task of optimal control is to determine what is the best control $\theta(t)$ for the system in Eq. (4). For this reason, a payoff functional should be specified such that the optimal control maximizes the payoff [14]. The payoff functional can be defined as [25]

$$P[\theta(\cdot)] := \int_0^T H(\boldsymbol{u}(t), \theta(t))dt + G(\Phi(\boldsymbol{u}(T))) \tag{5}$$

where $H(\cdot)$ is the running payoff and $G(\cdot)$ is the terminal payoff.

Eq. (5) plays the same role as the regularization function and the loss in Eq. (2), and the residual network in Eq. (3) is exactly the explicit Euler forward discretization of Eq. (4) [27]. From the differential equation (4), one can see that the nonlinear function $f(\boldsymbol{u}(t), \theta(t))$ is designed to depict the time derivative of the feature trajectory $\boldsymbol{u}(t)$ at time t. The optimal control viewpoint opens a new way to study deep neural networks in continuous functional space by leveraging the rich results from differential equations and variational calculus [41, 25].

3. Proposed method

In Eq. (4), the integer-order differential equation (IODE) is used to depict the dynamic system, and based on which the network is constructed [27]. However, the IODE system has short-term memory, *i.e.*, the evolution of the feature trajectory depends only on the current state without considering the history of its development.
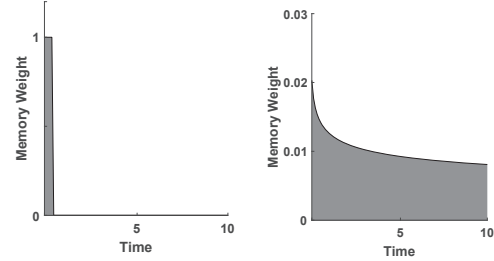

Figure 1. Memory mode. The left subfigure illustrates the short-term memory of IODE, and the right subfigure illustrates the power-law memory (long-term memory) of FODE.

Power-law memory. As previously discussed, the long-term memory is the key to the success of a lot of dynamic systems. In general, memory obeys the power-law property, in which the weight of the previous state at time t_i in defining the present stage at time t is proportional to $(t - t_i)^{\beta-1}$ [11], where $1 > \beta > 0$. Recent studies in [12] have shown that the power-law memory system can be described by a fractional-order differential equation (FODE), where the fractional-order derivative $\mathscr{D}^\beta \boldsymbol{u}(t)$ of a function $\boldsymbol{u}(t)$ is defined by [2]

Definition 1 *[31](Grünwald-Letnikov)*

$$\mathscr{D}^\beta \boldsymbol{u}(t) = \lim_{h \to 0} \frac{1}{h^\beta} \sum_{k=0}^{\left[\frac{t}{h}\right]} (-1)^k \binom{\beta}{k} \boldsymbol{u}(t - kh) \tag{6}$$

where β is the order of the derivative ($0 < \beta < 1$), $[\cdot]$ means the interger part and h is the step size.

To show how the memory mode of FODE is different from that of IODE, an illustration is given in Figure 1, and where the vertical-axis is the memory weight of the previous state t_i in defining the present stage at time t, the horizontal-axis is the time that has been past. One can see that when time evolves, the memory on the previous states disappear in IODE, while it lasts in FODE. To take full advantage of the FODE in memory persistent, we propose to construct an image denoising network from the fractional optimal control (FOC) viewpoint.

---

[2] Note that there are several definitions of the fractional-order derivative. Here we adopt the widely used Grünwald-Letnikov's definition.
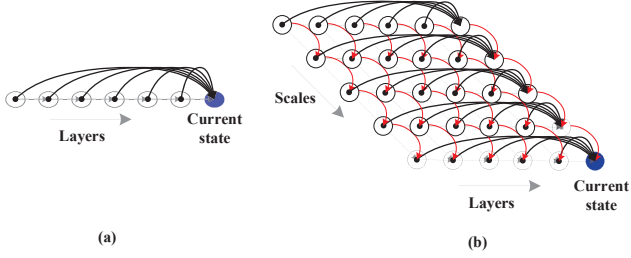
Figure 2. Comparison between the single scale memory system and multi-scale memory system. The blue dot represents the current state, the black arrow lines represent the connections of different layers in one scale, and the red arrow lines represent the cross-scale feature interactions.

3.1. Fractional optimal control view of image denoising

We consider the fractional-order system in [36], and assume that $\boldsymbol{u}$ is not only continuous in time but also continuous in space as a functional $\boldsymbol{u}(t,s)$ to depict the image feature trajectory, where $s \in \Omega$ is the two dimensional spatial position. Thus the control problem can be mathematically described as

$$\min_{\theta(t)} \frac{1}{2} \int_{\Omega} \left(\Phi(\boldsymbol{u}\left(T,s\right)) - \boldsymbol{x}\left(s\right) \right)^2 ds$$
$$s.t. \begin{cases} \mathscr{D}_t^{\beta} \boldsymbol{u}(t,s) = f\left(\boldsymbol{u}(t,s),\theta(t)\right) \\ \boldsymbol{u}(0,s) = \Psi(\boldsymbol{y}(s)), t \in [0,T], \end{cases} \quad (7)$$

where $\boldsymbol{y}(s)$ is the input noisy image, $\boldsymbol{x}(s)$ is the ground truth clean image and $\mathscr{D}_t^{\beta} \boldsymbol{u}(t,s)$ is the β-th order derivative of $\boldsymbol{u}(t,s)$ *w.r.t.* time t. Φ and Ψ are linear transformations, *e.g.,* convolution. The problem (7) aims to find the optimal control $\theta(t)$ such that the objective loss is minimized. The state equation in problem (7) characterizes the whole evolution process of the denoising system given the noisy input $\boldsymbol{u}(0,s)$.

3.2. Multi-scale memory systems

The continuous model (7) is independent of the actual image resolution. In practice, different resolutions of an image represent different scales of features. Enabling the long-term memory on different scale features can naturally strengthen the representation power of the system. Therefore, to make the best use of our memory system, we propose a multi-scale model by applying the FODE in Eq. (7) to multi-scale image features, which can be described as follows

$$\begin{cases} \mathscr{D}_t^{\beta} \boldsymbol{u}(t,s,l_1) = f(\boldsymbol{u}(t,s,l_1), g(\boldsymbol{u}(t,s,l_{1+1})), \theta_1(t)) \\ \mathscr{D}_t^{\beta} \boldsymbol{u}(t,s,l_2) = f(\boldsymbol{u}(t,s,l_2), g(\boldsymbol{u}(t,s,l_{2\pm1})), \theta_2(t)) \\ \quad \cdots \\ \mathscr{D}_t^{\beta} \boldsymbol{u}(t,s,l_k) = f(\boldsymbol{u}(t,s,l_k), g(\boldsymbol{u}(t,s,l_{k-1})), \theta_k(t)) \\ \boldsymbol{u}(0,s,l_1) = \Psi\boldsymbol{y}(s), \boldsymbol{u}(0,s,l_i) = T_{\downarrow}\boldsymbol{u}(1,s,l_{i-1}) \\ 1 \le l_i \le k, 0 \le t \le T, \end{cases}$$
$$(8)$$

where $\mathscr{D}_t^{\beta} \boldsymbol{u}(t,s,l_i)$ represents the fractional order derivatives in Eq. (7) at scale l_i. l_1 is the original scale space and l_k is the down-sampled k-th scale space. In Eq. (8), adjacent scale interactions are allowed by a scale switch function which is defined by $g(x) = wT(x)$, where $w \in \{0,1\}$ is a binary variable and $T(\cdot)$ is the pooling or unpooling operation [3]. The average pooling is adopted in this paper such that high scale features represent mostly the low frequency information (coarse features).

The multi-scale memory system increases the expressivity of current state features by memorizing the previous state features and different scale features, as shown in Figure 2. Consider only the current state (blue dot), Figure 2 (a) is the single scale memory system, in which the current state is explicitly connected to the previous layers. Figure 2 (b) is the multi-scale memory system, in which the current state is not only explicitly connnected to the previous layers (black arrow lines) but also implicitly connect to the features of all the scales (red arrow lines), thus the features of the current state can be more expressive.

3.3. Architecture of FOCNet

The architecture of our FOCNet is based on the discretization of the foregoing continuous FOC model in Eq. (8). In discretization, we set h in Eq. (6) to be 1. Based on the multi-scale dynamic system defined in Eq. (8), we could build our network whose architecture is shown in Figure. 3. The image features evolve in different scales according to FODE in Eq. (8).

In each scale, the network connects the long-term features according to the specific definition of the fractional-order derivative. Moreover, different scale features are interacted via a scale transform (pooling/unpooling) operator together with a learned scale transform switch function g in Eq. (8), which determines whether cross-scale feature interactions are allowed or not by taking a binary value $\{0,1\}$. Mathematically, the evolution process of our image denoising network can be expressed as

$$\boldsymbol{u}_{t+1}^{l_l} = \sum_{k=0}^{t} w_k \boldsymbol{u}_k^{l_l} + \sigma \left(\theta_t * \left(\boldsymbol{u}_t^{l_l} + g(\boldsymbol{u}_t^{l_l \pm 1}) \right) \right) \quad (9)$$

where $\boldsymbol{u}_{t+1}^{l_l}$ is the output of the t-th layer in scale l_l. The weight w_k is set according to Eq. (6) as $w_k = (-1)^{t-k+2} \binom{\beta}{t-k+1}$, which can be calculated by $w_t = \beta, w_{k-1} = (1 - \frac{1+\beta}{t-k+2}) w_k, k = 1, \cdots t$. The nonlinear unit $\sigma(\cdot)$ consists of "Convolution + BN + Relu". $\boldsymbol{u}_t^{l_l \pm 1}$ denotes that either upper scale features $\boldsymbol{u}_t^{l_l+1}$ or lower scale features $\boldsymbol{u}_t^{l_l-1}$ are used [4].

---

[3]Pooling is denoted by $T_{\downarrow}$ and unpooling is denoted by $T_{\uparrow}$.

[4]Note that cross-scale feature interactions are not necessary for every layer in each scale, the specific connections are shown in Figure 3.
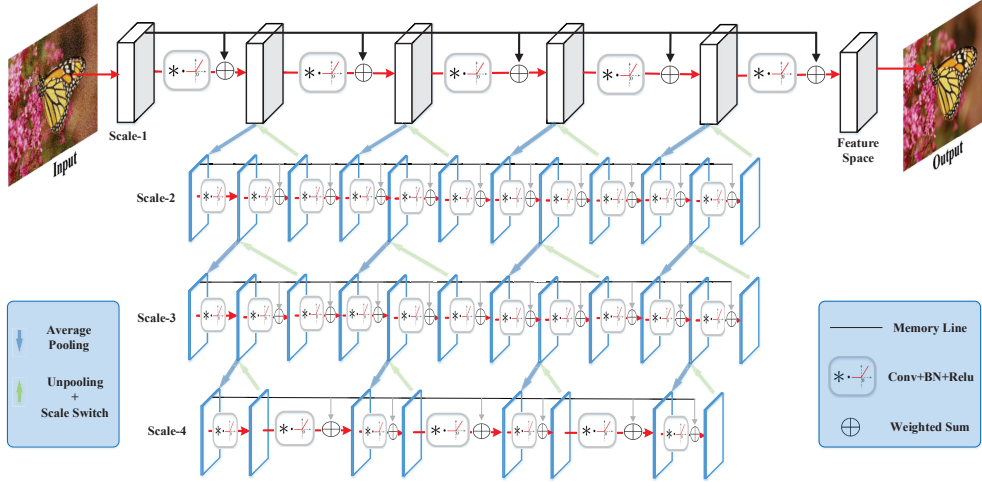
Figure 3. Architecture of the FOCNet. The memory lines are constructed according to the Grünwald-Letnikov's definition of the fractional-order derivative in Eq. (6). The block "Conv+BN+Relu" corresponds to Eq. (9). The scale switch function is defined in Eq. (8).

Let $\{(\boldsymbol{y}^i, \boldsymbol{x}^i)\}_{i=1}^N$ be a training set, where $\boldsymbol{y}^i$ is the input noisy image and $\boldsymbol{x}^i$ is the ground truth image label (clean image). Denote by $\boldsymbol{u}_T^i(\Theta)$ the final output of our network and Θ represents the parameters of the network, then according to Eq. (7), the loss functional in our FOCNet is

$$\mathscr{L}(\boldsymbol{u}_T^i(\Theta)) = \frac{1}{2N} \sum_{i=1}^N \|\Phi(\boldsymbol{u}_T^i(\Theta)) - \boldsymbol{x}^i\|_F^2 \qquad (10)$$

The optimal parameter set Θ^* is obtained by minimizing the loss function $\mathscr{L}(\boldsymbol{u}_T^i(\Theta))$ using the ADAM algorithm [23].

3.4. Property of the long-term memory

By exploiting the optimal conditions of the control problem (7), we show that our FOCNet has the long-term memory not only in the forward but also in the backward propagation. To begin with, we define the Hamiltonian H : $\mathbb{R}^d \times \mathbb{R}^d \times \Theta \to \mathbb{R}$ of the control problem

$$H(\boldsymbol{u}(t), p(t), \theta(t)) := p(t) \cdot f(\boldsymbol{u}(t), \theta(t)). \qquad (11)$$

According to the Pontryagin's maximum principle, the optimal conditions of the minimization problem (7) can be characterized by the following lemma.

Lemma 1 (*Pontryagin's maximum principle*) *[2] Let $(\hat{\boldsymbol{u}}(t), \hat{\theta}(t))$ be the optimal control process for (7). Then there exists an absolutely continuous co-state process $\hat{p}(t)$: $[0, T] \to \mathbb{R}^d$ such that the Hamilton's equation*

$$\mathscr{D}_t^\beta \hat{\boldsymbol{u}}(t) = \nabla_p H(\hat{\boldsymbol{u}}(t), \hat{p}(t), \hat{\theta}(t)), \qquad (12)$$

$$\mathscr{D}_t^\beta \hat{p}(t) = -\nabla_{\boldsymbol{u}} H(\hat{\boldsymbol{u}}(t), \hat{p}(t), \hat{\theta}(t)), \qquad (13)$$

with initial conditions

$$\hat{\boldsymbol{u}}(0) = \Phi(\boldsymbol{y}), \quad \hat{p}(T) = -\nabla \mathscr{L}(\hat{\boldsymbol{u}}(T))$$

are satisfied. Moreover, for each $t \in [0, T]$, we have the Hamiltonian maximization condition

$$H(\hat{\boldsymbol{u}}(t), \hat{p}(t), \hat{\theta}(t)) \geq H(\hat{\boldsymbol{u}}(t), \hat{p}(t), \theta(t)). \qquad (14)$$

for all $\theta(t)$.

According to the analysis in [25] and the mechanism of backward propagation (BP), one can easily verify that the discretization of Eq. (12) and Eq. (13) correspond to the forward propagation (FP) and BP of the FOCNet, respectively. Thus, the forward and backward of FOCNet correspond to a specific FODE which has the long-term memory.

3.5. Discussions

We construct a FOCNet by solving a multi-scale fractional-order optimal control problem to address the long-term memory in denoising DCNN. Some denoising DCNN have also been constructed to deal with the long-term memory and multi-scale interactions but from totally different angles.

Relation to MemNets. The MemNet [42] was proposed to address the long-term memory by concatenating the output of previous layers to generate large size features, then the large size features are contracted to a small one by learning a contraction filter with a huge amount of parameters. In addition, the mechanism for characterizing the long-term memory using concatenation is not theoretically solid. In contrast, our FOCNet can characterize well the power-law memory principle. Moreover, it does not need to learn the huge amount of contraction filters to combine the image features.

Relation to Unet. The Unet consists of a contraction and an expansion subnets. In the contraction stage, the image features are successively convolved and down-sampled to

generate multi-scale image features. In the expansion stage, the image features are convolved and up-sampled successively to generate the final output. In our proposed method, image features propagate in multiple scales, and the Unet can be considered as a special case of FOCNet in which the forward process evolves only one step for each scale. Besides, a scale transform switch is designed in our FOCNet so that the across-scale feature interactions can be adaptive, which is not available in Unet. In brief, our proposed network structure is more flexible and general than the Unet architecture [37].

4. Experiments

4.1. Experimental setting

Dataset generation. Before training the FOCNet model, we need to prepare a training dataset with image pairs $\{\boldsymbol{y}_i, \boldsymbol{x}_i\}_{i=1}^N$. Here $\boldsymbol{y}_i$ is generated by adding AWGN with specific noise levels to the latent clean image $\boldsymbol{x}_i$, *i.e.*, $\boldsymbol{y}_i = \boldsymbol{x}_i + \boldsymbol{n}$. Following [26], we consider three noise levels, *i.e.*, $\sigma = 15, 25, 50$. We collect clean images from two datasets, including 200 images from Berkeley Segmentation Dataset [30] and 500 images from DIV2K [1] to generate the training data. We randomly crop $N = 64 \times 2000$ image patches of size 80×80 from the collected images for training.

Network training. The residual mapping strategy presented in [45] is employed in our FOCNet. To learn the optimal parameters, the ADAM optimizer [23] is used to minimize the loss function. The default setting of the hyper-parameters of ADAM is adopted. The network parameters are initialized by random values as in DnCNN [23] and the mini-batch size is 64. We set the scale number as 4, the first scale has 4 convoution layers, the second and the third scale have 11 convolution layers and the fourth scale has 7 convolution layers. For all the 4 scales, we set the feature channel as 128 and the size of the convolution filters as $3 \times 3 \times 128 \times 128$.

We use the MatConvNet package [43] with cuDNN 8.0 to train FOCNet. All the experiments are conducted in the Matlab (R2017a) environment running on a PC with Inter(R) Xeon(R) E5-2620 CPU 2.10GHz and an Nvidia TITAN Xp GPU. We learn for each noise level a FOCNet model. The learning algorithm converges very fast within 40 epoch, thus we train 35 epoch for our FOCNet, the learning rate is decayed exponentially from 10^{-3} to 10^{-4} in the 35 epochs. It takes about two days to train a FOCNet.

4.2. Ablation study

The setting of β. The setting of parameter β in Eq. (6) is important to our FOCNet. When β is an integer number, Eq. (6) reduces to an integer-order derivative. To figure out how the parameter β influences the denoising results, we conduct experiments with different β values as:

Table 1. Comparison of the denoising results with different β. The PSNR values are the average results on Set12 with noise level 50.

β	0.1	0.2	0.5	0.7	1.0	2.0
PSNR	27.37	**27.42**	27.30	27.25	27.12	27.15

Table 2. Comparison of the denoising results (dB) with different scales. The noise level is set as $\sigma = 50$.

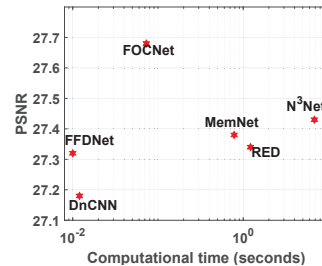| Dataset	one scale	two scales	three scales	four scales	five scales
Set12	27.42	27.49	27.58	27.68	27.69
BSD68	26.32	26.38	26.45	26.48	26.50
Urban100	26.80	27.02	27.21	27.40	27.40

Figure 4. Comparison of computational time and PSNR values.

$\beta = 0.1, 0.2, 0.5, 0.7, 1.0$ and 2.0 for single scale FOCNet. We test FOCNet on the dataset Set12 for noise level $\sigma = 50$. The results (average PSNR) are shown in Table 1. One can see that the FOCNet achieves the highest PSNR when $\beta = 0.2$ and gets inferior PSNR value for $\beta = 1, 2$, which indicates that the long-term memory is useful to image denoising.

The setting of scale. We would also like to show the effectiveness of the multi-scale strategy in FOCNet for image denoising. We test FOCNet from 1 to 5 scales, and the average PSNR values on three datasets are shown in Table 2. We see that with the increase of the number of scales, the PSNR values increase as well. However, when the scale number is up to 4, further increase of the scale only gives negligible improvement. To balance the efficiency and effectiveness, we set the scale number to be 4.

4.3. Denoising results

We evaluate the denoising performance of FOCNet on three widely used test sets, *i.e.*, Set12 [45], BSD68 [30] and Urban100 [20]. The experimental results of the FOCNet are compared with the following state-of-the-art and representative denoising methods: BM3D [9], WNNM [16], TNRD [8], DnCNN [45], FFDNet [46], RED [29], MemNet [42] and N^{3}Net [35]. Among the compared methods, except for BM3D, WNNM and TNRD, all the remaining methods are based on DCNN. Similar to FOCNet, the MemNet exploits the long-term memory as well.

Table 3 exhibits the average PSNR results of the competing methods on the three datasets and Table 4 lists the PSNR values of the competing methods for each image in Set12. The best results are highlighted in red. It can be seen from Table 3 and Table 4 that the PSNR results of FOCNet have remarkable improvements over the competing methods not

(a) Original Image　　　(b) Noisy Image　　　(c) WNNM [16] (25.44)　　　(d) TNRD [8] (25.42)　　　(e) DnCNN [45] (25.70)

(f) FFDNet [46] (25.77)　　　(g) RED [29] (25.80)　　　(h) MemNet [42] (25.76)　　　(i) N³Net [35] (26.00)　　　(j) FOCNet (26.19)
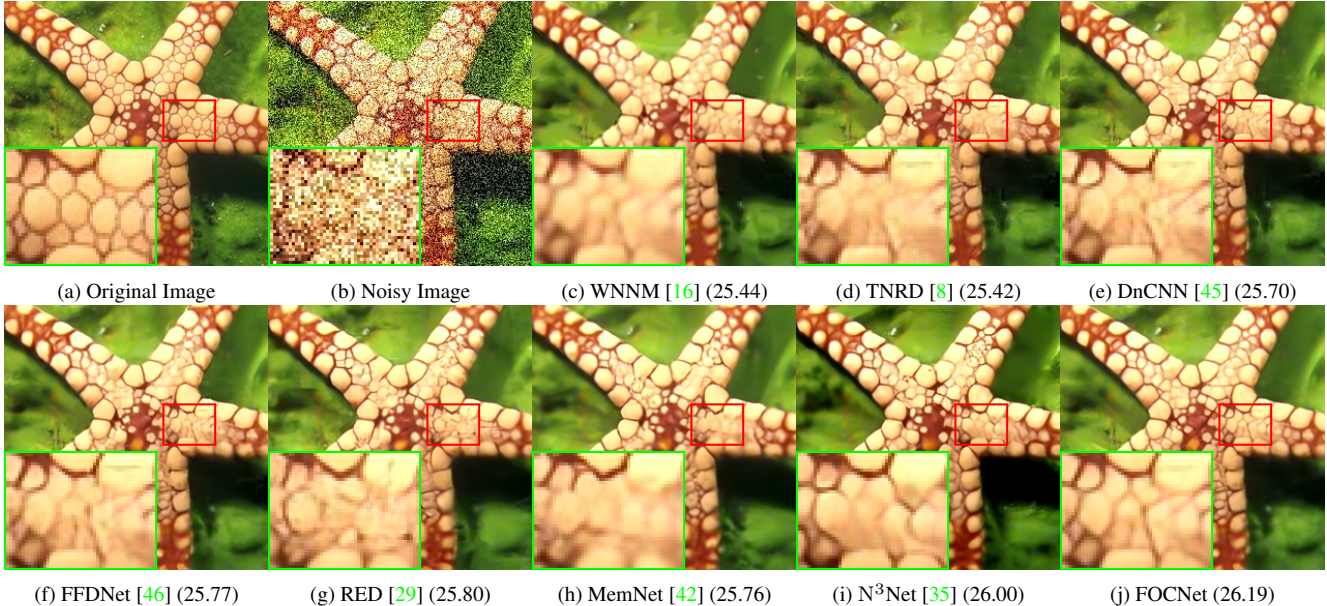
Figure 5. Visual quality comparison. The referenced region (red rectangle) is zoom in at the bottom left green rectangle for better comparison. The PSNR value of each recovered result is given in the parentheses.

Table 3. Average PSNR values for noise level 15, 25 and 50 on Set12, BSD68 and Urban100. The best results are highlighted in red. The symbol "-" denotes that the results are not provided by the corresponding authors.

Dataset	σ	BM3D [9]	WNNM [16]	TNRD [8]	DnCNN [45]	FFDNet [46]	RED [29]	MemNet [42]	N³Net [35]	FOCNet
Set12	15	32.37	32.70	32.50	32.86	32.75	-	-	-	33.07
	25	29.97	30.28	30.05	30.44	30.43	-	-	30.50	30.73
	50	26.72	27.05	26.82	27.18	27.32	27.34	27.38	27.43	27.68
BSD68	15	31.08	31.37	31.42	31.73	31.63	-	-	-	31.83
	25	28.57	28.83	28.92	29.23	29.19	-	-	29.30	29.38
	50	25.60	25.87	25.97	26.23	26.29	26.35	26.35	26.39	26.50
Urban100	15	32.34	32.97	31.98	32.67	32.42	-	-	-	33.15
	25	29.70	30.39	29.29	29.97	29.92	-	-	30.19	30.64
	50	25.94	26.83	25.71	26.28	26.52	26.48	26.64	26.82	27.40

only on average but also for each test image. Specifically, FOCNet improves the traditional methods such as BM3D and WNNM by $0.5 \sim 0.9$ dB on Set12 with noise level $\sigma = 50$, and compared with DnCNN, the improvement is still up to $0.3 \sim 0.5$dB. On the Urban100 dataset, the differences between FOCNet and the compared methods become more distinct. FOCNet also outperforms the memory persistent network MemNet by a large margin in all the test sets. We also compare FOCNet with the nonlocal based denoising network N³Net, and the results verified the superiority of FOCNet over N³Net, despite that there are no time consuming nonlocal operations in FOCNet.

Visual quality comparisons are given in Fig. 5 and Fig. 6, in which the clean images are corrupted by Gaussian noise with noise level $\sigma = 50$. In Fig. 5, we can see that FOCNet is able to recovery the fine details of the corrupted image, such as the region in the red rectangle, which is zoomed-in at the bottom left of the image marked by green rectangle. In contrast, the competing methods over-smooth the details of the image. In Fig. 6, the recovered result provided by FOCNet is more faithful to the clean image

than all the competing methods. As shown in the zoomed-in part (the windows of the building and the wall), FOCNet can accurately estimate the clean image with clear structures, while the results of competing methods either over-smooth or blur much the structures. All these facts indicate that FOCNet is superior to the existing methods not only in quantitative results but also in perceptual quality.

In addition to the denoising performance, we also compare the run time together with the corresponding PSNR value of the competing methods in Figure 4. Only the CNN based methods are considered in this comparison. The computational time of FOCNet is much less than RED [29], MemNet [42] and N³Net [35]. In comparison to DnCNN [45] and FFDNet [46], FOCNet is a little slower but it delivers much more satisfactory results in terms of PSNR. Overall, FOCNet is not only efficient but also effective.

5. Conclusion

In this paper, we used the fractional optimal control (FOC) theory to model the deep convolution neural net-

(a) Original Image   (b) Noisy Image   (c) WNNM [16] (25.40)   (d) TNRD [8] (25.06)   (e) DnCNN [45] (25.56)

(f) FFDNet [46] (25.75)   (g) RED [29] (25.78)   (h) MemNet [42] (25.82)   (i) N³Net [35] (25.91)   (j) FOCNet (26.14)
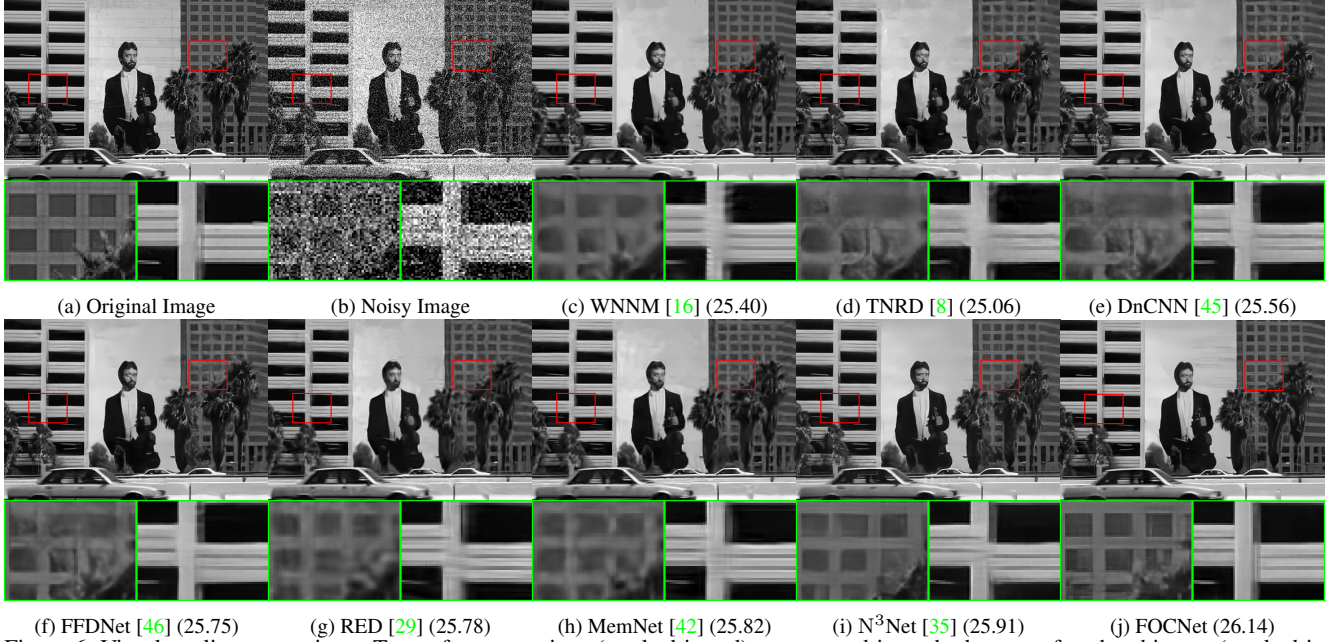
Figure 6. Visual quality comparison. Two reference regions (marked in red) are zoomed in at the bottom of each subimage (marked in green). The PSNR value of each recovered result is given in the parentheses.

Table 4. The PSNR results of different methods on Set12 dataset with noise level 15, 25 and 50. The best results are highlighted in red.

Images	C.man	House	Peppers	Starfish	Monarch	Airplane	Parrot	Lena	Barbara	Boat	Man	Couple
Noise Level	$\sigma = 15$											
BM3D [9]	31.91	34.93	32.69	31.14	31.85	31.07	31.37	34.26	33.10	32.13	31.92	32.10
WNNM [16]	32.17	35.13	32.99	31.82	32.71	31.39	31.62	34.27	33.60	32.27	32.11	32.17
TNRD [8]	32.19	34.53	33.04	31.75	32.56	31.46	31.63	34.24	32.13	32.14	32.23	32.11
DnCNN [45]	32.61	34.97	33.30	32.20	33.09	31.70	31.83	34.62	32.64	32.42	32.46	32.47
FFDNet [46]	32.42	35.01	33.10	32.02	32.77	31.58	31.77	34.63	32.50	32.35	32.40	32.45
FOCNet	32.71	35.44	33.41	32.40	33.29	31.82	31.98	34.85	33.09	32.62	32.56	32.64
Noise Level	$\sigma = 25$											
BM3D [9]	29.45	32.85	30.16	28.56	29.25	28.42	28.93	32.07	30.71	29.90	29.61	29.71
WNNM [16]	29.64	33.22	30.42	29.03	29.84	28.69	29.15	32.24	31.24	30.03	29.76	29.82
TNRD [8]	29.72	32.53	30.57	29.02	29.85	28.88	29.18	32.00	29.41	29.91	29.87	29.71
DnCNN [45]	30.18	33.06	30.87	29.41	30.28	29.13	29.43	32.44	30.00	30.21	30.10	30.12
FFDNet [46]	30.06	33.27	30.79	29.33	30.14	29.05	29.43	32.59	29.98	30.23	30.10	30.18
N³Net [35]	30.08	33.25	30.90	29.55	30.45	29.02	29.45	32.59	30.22	30.26	30.12	30.12
FOCNet	30.35	33.63	31.00	29.75	30.49	29.26	29.58	32.83	30.74	30.46	30.22	30.40
Noise Level	$\sigma = 50$											
BM3D [9]	26.13	29.69	26.68	25.04	25.82	25.10	25.90	29.05	27.22	26.78	26.81	26.46
WNNM [16]	26.45	30.33	26.95	25.44	26.32	25.42	26.14	29.25	27.79	26.97	26.94	26.64
TNRD [8]	26.62	29.48	27.10	25.42	26.31	25.59	26.16	28.93	25.70	26.94	26.98	26.50
DnCNN [45]	27.03	30.00	27.32	25.70	26.78	25.87	26.48	29.39	26.22	27.20	27.24	26.90
FFDNet [46]	27.03	30.43	27.43	25.77	26.88	25.90	26.58	29.68	26.48	27.32	27.30	27.07
RED [29]	27.02	30.46	27.22	25.80	26.99	25.94	26.45	29.58	26.65	27.32	27.20	27.08
MemNet [42]	27.23	30.70	27.51	25.76	27.19	25.96	26.49	29.63	26.67	27.29	27.24	27.14
N³Net [35]	27.14	30.50	27.58	26.00	27.03	25.75	26.50	29.67	27.01	27.32	27.33	27.04
FOCNet	27.36	30.91	27.57	26.19	27.10	26.06	26.75	29.98	27.60	27.53	27.42	27.39

work. With the aid of fractional-order derivative, a FOCNet was elaborated to mitigate the short-term memory of the existing integer-order derivative based network. The FOCNet obeys the power-law memory mode which has verified to be effective in a lot of dynamic systems. According to the optimal condition given by Pontryagin's maximum principle, we showed that the FOCNet has long-term memory not only in the forward process but also in the backward process. In addition, a multi-scale strategy was adopted to strengthen the network by promoting cross-scale feature interactions. Experimental results on image denoising verified that FOCNet achieves the leading PSNR results while having a reasonable runing speed.

References

[1] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, volume 3, page 2, 2017. 6

[2] H. M. Ali, F. L. Pereira, and S. M. Gama. A new approach to the pontryagin maximum principle for nonlinear fractional optimal control problems. *Mathematical Methods in the Applied Sciences*, 39(13):3640–3649, 2016. 5

[3] J. R. Anderson. *Learning and memory: An integrated approach*. John Wiley & Sons Inc, 2000. 2

[4] W. Bae, J. J. Yoo, and J. C. Ye. Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification. In *CVPR Workshops*, pages 1141–1149, 2017. 1

[5] J. Bai and X.-C. Feng. Fractional-order anisotropic diffusion for image denoising. *IEEE transactions on image processing*, 16(10):2492–2502, 2007. 2

[6] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005. 1

[7] R. Caponetto. *Fractional order systems: modeling and control applications*, volume 72. World Scientific, 2010. 2

[8] Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1256–1272, 2017. 6, 7, 8

[9] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. 1, 6, 7, 8

[10] M. Edelman. Fractional dynamical systems. *arXiv preprint arXiv:1401.0048*, 2013. 2

[11] M. Edelman. Fractional maps as maps with power-law memory. In *Nonlinear dynamics and complexity*, pages 79–120. Springer, 2014. 2, 3

[12] M. Edelman. Universality in systems with power-law memory and fractional dynamics. In *Chaotic, Fractional, and Complex Dynamics: New Insights and Perspectives*, pages 147–171. Springer, 2018. 3

[13] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006. 1

[14] L. C. Evans. An introduction to mathematical optimal control theory. *Lecture Notes, University of California, Department of Mathematics, Berkeley*, 2005. 3

[15] A. N. Gomez, M. Ren, R. Urtasun, and R. B. Grosse. The reversible residual network: Backpropagation without storing activations. In *Advances in Neural Information Processing Systems*, pages 2214–2224, 2017. 2

[16] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2862–2869, 2014. 1, 6, 7, 8

[17] E. Haber and L. Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1), 2018. 1

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 3

[19] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269. IEEE, 2017. 2

[20] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015. 6

[21] A. Jain and J. Jain. Partial differential equations and finite difference methods in image processing–part ii: Image restoration. *IEEE Transactions on Automatic Control*, 23(5):817–834, 1978. 2

[22] X. Jia, X. Feng, and W. Wang. Adaptive regularizer learning for low rank approximation with application to image denoising. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3096–3100. IEEE, 2016. 1

[23] D. P. Kingma and J. L. Ba. Adam: Amethod for stochastic optimization. 5, 6

[24] G. Larsson, M. Maire, and G. Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*, 2016. 2

[25] Q. Li, L. Chen, C. Tai, and E. Weinan. Maximum principle based algorithms for deep learning. *The Journal of Machine Learning Research*, 18(1):5998–6026, 2017. 1, 2, 3, 5

[26] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo. Multi-level wavelet-cnn for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 773–782, 2018. 1, 6

[27] Y. Lu, A. Zhong, Q. Li, and B. Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. *arXiv preprint arXiv:1710.10121*, 2017. 1, 2, 3

[28] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2272–2279. IEEE, 2009. 1

[29] X. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in neural information processing systems*, pages 2802–2810, 2016. 1, 6, 7, 8

[30] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 416–423. IEEE, 2001. 6

[31] C. A. Monje, Y. Chen, B. M. Vinagre, D. Xue, and V. Feliu-Batlle. *Fractional-order systems and controls: fundamentals and applications*. Springer Science & Business Media, 2010. 2, 3

[32] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):629–639, 1990. 1, 2

[33] I. Petráš. *Fractional-order nonlinear systems: modeling, analysis and simulation.* Springer Science & Business Media, 2011. 2

[34] F. J. Pineda. Dynamics and architecture for neural computation. *Journal of Complexity*, 4(3):216–245, 1988. 1

[35] T. Plötz and S. Roth. Neural nearest neighbors networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 1, 3, 6, 7, 8

[36] I. Podlubny. Fractional-order systems and pi/sup/spl lambda//d/sup/spl mu//-controllers. *IEEE Transactions on automatic control*, 44(1):208–214, 1999. 2, 4

[37] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 6

[38] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 2009. 1

[39] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992. 1, 2

[40] L. Ruthotto and E. Haber. Deep neural networks motivated by partial differential equations. *arXiv preprint arXiv:1804.04272*, 2018. 1, 2

[41] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen. *Variational methods in imaging*. Springer, 2009. 3

[42] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4539–4547, 2017. 1, 2, 3, 5, 6, 7, 8

[43] A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692. ACM, 2015. 6

[44] E. Weinan. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1):1–11, 2017. 1

[45] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 1, 3, 6, 7, 8

[46] K. Zhang, W. Zuo, and L. Zhang. Ffdnet: Toward a fast and flexible solution for cnn based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. 1, 6, 7, 8

[47] X. Zhang, Z. Li, C. C. Loy, and D. Lin. Polynet: A pursuit of structural diversity in very deep networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 3900–3908. IEEE, 2017. 2