ODE-inspired Network Design for Single Image Super-Resolution

Xiangyu He[1,2*], Zitao Mo[1,2*], Peisong Wang[1], Yang Liu[4], Mingyuan Yang[4], Jian Cheng[1,2,3] ✉
[1] NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Center for Excellence in Brain Science and Intelligence Technology, CAS, Beijing, China
[4] Alibaba Group, Hangzhou, China
{xiangyu.he, zitao.mo, peisong.wang, jcheng}@nlpr.ia.ac.cn

Abstract

Single image super-resolution, as a high dimensional structured prediction problem, aims to characterize fine-grain information given a low-resolution sample. Recent advances in convolutional neural networks are introduced into super-resolution and push forward progress in this field. Current studies have achieved impressive performance by manually designing deep residual neural networks but overly relies on practical experience. In this paper, we propose to adopt an ordinary differential equation (ODE)-inspired design scheme for single image super-resolution, which have brought us a new understanding of ResNet in classification problems. Not only is it interpretable for super-resolution but it provides a reliable guideline on network designs. By casting the numerical schemes in ODE as blueprints, we derive two types of network structures: LF-block and RK-block, which correspond to the Leapfrog method and Runge-Kutta method in numerical ordinary differential equations. We evaluate our models on benchmark datasets, and the results show that our methods surpass the state-of-the-arts while keeping comparable parameters and operations.

1. Introduction

Super-resolution (SR) enjoys the recent advances in deep learning and has attracted much attention in recent years due to the rapid growth of image and video data. Generally speaking, super-resolution can be applied to many applications including medical image processing [19], satellite and aerial imaging [46], facial image improvement [28], etc. Although obtaining high-resolution images from one or several low-resolution samples can be an ill-posed problem, convolutional neural networks have powered this field, making the resulting images natural and detailed. In this paper, we focus on single image super-resolution (SISR).

In light of the empirical success of convolutional neural networks (CNN) in high-level computer vision tasks such as image classification, Dong et al. [11] proposed a CNN-based SR algorithm. Since then, convolutional neural networks have became the mainstream in the field of super-resolution [29, 20, 22, 12, 31]. Though the performance is improving with tremendous effort, there remain some limitations: 1) Previous researches tend to care less about the computation overhead and introduce deeper convolutional neural networks to enhance performance. The huge amount of calculations makes it intractable to apply the algorithm to real-world applications. 2) Another side effect is that with the depth increases, more training tricks are required. Otherwise, the training procedure becomes numerically unstable [35, 31]. 3) Super-resolution is different from high-level visual tasks such as image classification, which extracts semantic features via a convolutional neural network. In contrast, super-resolution predicts pixel-level fine-grained information. Directly employing state-of-the-art CNNs does not necessarily lead to an optimal solution.

To alleviate these problems, we propose to apply ODE-inspired schemes to Super-Resolution network designs (OISR). First, we revisit the similarity between forward Euler method and residual structure by adopting the view of dynamical system, identifying that we can take advantage of ODEs for SISR network designs. Second, we develop two kinds of building blocks, corresponding to the Leapfrog method and Runge-Kutta method in numerical ODEs. To the best of our knowledge, this is the first attempt to introduce ODE-inspired schemes into single image super-resolution network design directly, providing a helpful viewpoint of single image super-resolution and a relatively reliable guidance on network designs. In this work, both lightweight and deep networks are generated using proposed building blocks. Experimental results on bench-

*These authors contributed equally to this work.

mark datasets demonstrate that our methods outperform the state-of-the-arts, which indicates a better trade-off between performance and computing cost. Lastly, we explore different module G while maintaining a relatively stable amount of computation. It is shown that our deep networks converge rapidly without extra training tricks.

2. Related work

2.1. Single image super-resolution

Single image super-resolution is a classical computer vision task. Learning mapping functions from the low-resolution image to high-resolution one is popular in previous literatures. These algorithms apply traditional machine learning techniques to image super-resolution, including PCA [3], kernel method [4], learning embedding [8], sparse-coding [43], etc. Another kind of methods make use of the image self-similarity without external databases. [14] exploits the patch redundancy to obtain a super-resolution image. Freedman et al.[13] go further and develop a localized searching algorithm. Huang et al. [18] extend this method by using detected perspective geometry to guide the patch search process.

Recent advances in SISR take advantage of the powerful representation capability of convolutional neural networks. Dong et al. [11] introduce SRCNN for SISR, they interpret the hidden layers in CNN as extraction, non-linear mapping, and reconstruction, corresponding to those steps in sparse coding [43]. DRCN [23] takes it a step further by firstly interpolating the LR images to the desired size, which suffers from "details lost" and the huge computational complexity. Kim et al. [20, 22] follow the same design pattern of using bicubic interpolation to upsample the image to the desired size, but they adopt a deep residual convolutional neural network to obtain a better representation. Since then, several deeper CNN-based super-resolution models have been proposed to achieve superior performance, including DRNN [36], LapSRN [24], SRResNet [27], etc.

However, deeper architectures bring about a larger amount of computation with the impressive progress on benchmark datasets. To address this problem, Dong et al. [12] remove the bicubic interpolation in SRCNN and introduced a deconvolution layer at the end of FSRCNN. They also adopt smaller filter sizes and a deeper network structure. In order to further reduce parameters, DRRN [37] introduces the combination of the recursive and residual blocks while compromising the runtime speed. Recently, CARN [1] presents a cascading mechanism upon a residual network, which utilizes the multi-level representation and multiple shortcut connections. To exploit the inter-relation of multi-scale factors, [31] proposes an MSRN model to encourage the feature reuse of different upsampling factors. MSRN [29] is a similar multi-scale method based on ResNet [16], they further propose a hierarchical features fusion architecture to utilize features at different scales.

Although the empirical success of CNN-based super-resolution methods is encouraging, most state-of-the-art design their networks empirically. It is clear that this handcrafted process requires a lot of tricks and attempts.

2.2. Bridging network design with ODEs

ResNet [16] and its variants [17, 41, 35] have become popular in a wide range of applications, including low-level visual tasks such as super-resolution. Inspired by its clever insight, many studies on network topology have emerged and promoted the process. Zhang et al. [45] propose Poly-Inception module and enhance the generalization ability. Larsson et al. [26] make use of the self-similarity and develop FractalNet. These networks share a similar idea of feature fusion by establishing multiple connections between different layers, which is proved to be effective.

Many good efforts try to bridge the gap between good performance and a poor understanding of where the effectiveness of residual connections stemming from. These studies mainly start from searching for similar mathematical structures and expect to take advantage of these well-developed theories. Liao and Poggio [30] show that deep ResNet is equivalent to a shallow RNN. The author of [40] first observes the relationship between ResNet and ODE. They grant deep neural networks as a discrete dynamical system, identifying the similarity between ResNet and the discretization of ODEs. Chang et al. [5] do more than explanation. They make use of numerical ODEs to construct reversible neural networks with the stability analysis. Lu et al. [32] focus on the discretization schemes of ODEs. They analyze the similarity between several network structures and numerical ODEs and propose an LM-architecture, which originates from the linear multistep method. Although these studies may not fully identify the true roots of ResNet's success, they shed some light on providing a guidline for network designs.

3. ODE-inspired network design for SISR

Recent advances in single image super-resolution are attributed to the progress of deep-learning, which enables SISR to take a powerful end-to-end framework. Broadly speaking, CNN-based methods map a low-resolution input to a high-resolution image. From a dynamical system perspective, it defines a map that takes input status forward x units of time in the phase space. In CNN semantics, time horizon x corresponds to layers that can be adaptively chosen, while the final status is restricted by labels. However, the problem lies in that, how to design a network that is able to achieve the goal. [40] describes it as a controllability problem and explore the simplified one-dimensional case, giving that there exits such a map generated by an ODE if

the problem is smooth enough. Since SISR is such a low-level visual problem with the constraint of high similarity between inputs and outputs, it tends to approach the conditions intuitively. Therefore, we are encouraged to take the perspective of dynamical system and apply the rich knowledge in ODE to SISR.

3.1. Mapping numerical ODEs to building blocks

In this section, we first revisit the similarity between forward Euler method and ResNet for clearness and self-containment. We consider the dynamical systems which can be described as an ODE, defined as

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x, y). \tag{1}$$

This system gives a map

$$\Phi(y_0, x) = y(x; y_0) \tag{2}$$

with initial status $y_0 \in \mathbb{R}^d$. Suppose $p(y_0)$ is the distribution of input feature y_0 on a domain Ω, if we regard CNN-base SISR as such a dynamical system, then we are supposed to minimize

$$L = \int_{\Omega} \|\Phi(y_0, x) - y\| \, \mathrm{d}p(y_0) \tag{3}$$

where Φ is a map should be learned in SISR, and it is also associated to the solution of Equation (1). When the system is nonlinear, in many cases, there is no simple formula describing the map, we have to turn to numerical methods. As presented in [6], forward Euler method

$$y_{n+1} = y_n + hf(x_n, y_n) \tag{4}$$

provides an approximation, which can be seen as a numerical ODE using the approximation to the integral of y' over an interval of width h: $y_{n+1} - y_n \approx hy'$. Residual block takes a similar form as

$$y_{n+1} = y_n + G(y_n). \tag{5}$$

This formula suggests the relationship [40, 32, 7, 9] and we are able to establish the bridge by defining

$$G(y_n) = hf(x_n, y_n), \tag{6}$$

thus mapping forward Euler to a residual block.

In this paper, we consider the supervised SISR problem where training data is provided to learn such a map Φ from low-resolution images to high-resolution targets. It may take many steps to reach the final status, each step corresponds to a CNN block. Either increasing the number of steps or refining motion of each step helps to achieve the goal, corresponding to increasing block numbers and designing finer blocks.

It should be noted that formula (4) is a first-order method in numerical ODEs. Higher-order methods are supposed to bring about some merits on reaching a more accurate solution. We are enlightened to deploy other numerical methods for a finer block.

LF-Block: LeapFrog method is a second-order linear 2-step method, as a refinement of forward Euler scheme. By doubling the time interval h, we rewrite the approximation of y' in the form of $y' \approx (y_{n+1} - y_{n-1})/2h$, then derive the following equation

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n), \tag{7}$$

which can be directly interpreted into CNN diagram using the definition in formula (6). In order to retain flexibility and obtain a block architecture, every three formulas above are grouped into a block as

$$y_{n-1} = y_{n-3} + 2hf(x_{n-2}, y_{n-2}) \tag{8}$$
$$y_n = y_{n-2} + 2hf(x_{n-1}, y_{n-1}) \tag{9}$$
$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n). \tag{10}$$

Hence, we obtain an interesting structure as shown in Figure 1 (b). Unlike ResNet where G is defined to be a certain combinations of ReLU and convolutions, we do not restrict G to be a fixed module except for its nonlinearity. The details are discussed in section 3.2.

RK2-Block: To further explore this design scheme, we now consider the Runge-Kutta family, which is widely used in numerical analysis. Making use of a trapezoidal formula

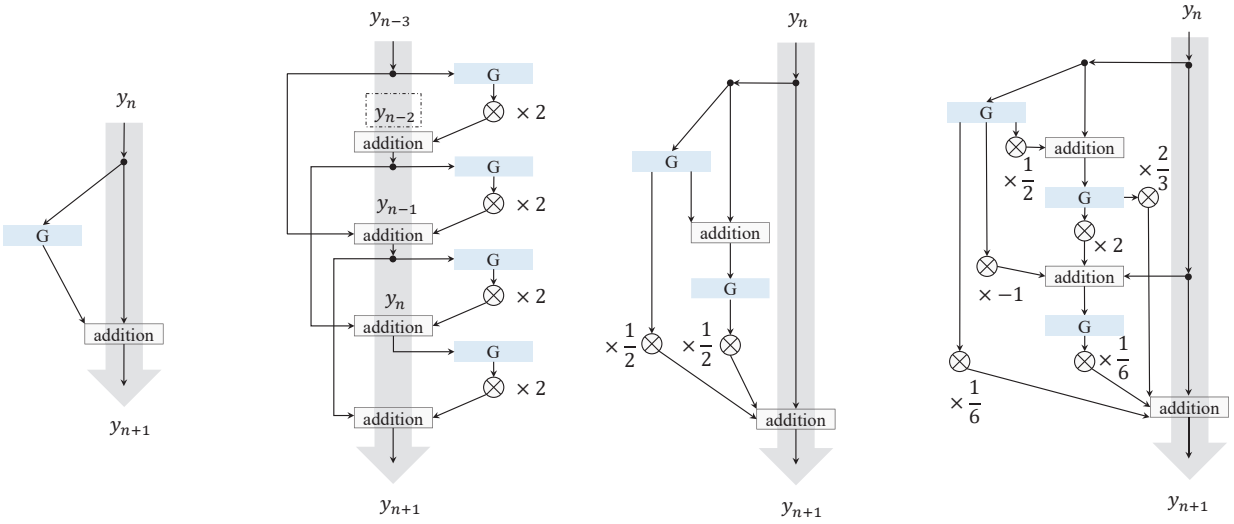
$$y_{n+1} = y_n + \frac{h}{2}(f(x_n, y_n) + f(x_{n+1}, \tilde{y}_{n+1})) \tag{11}$$

and replace $\tilde{y}_{n+1}$ with its first-order approximation (4), we will obtain the following equations

$$y_{n+1} = y_n + \frac{1}{2}(G_1 + G_2) \tag{12}$$
$$G_1 = hf(x_n, y_n) \tag{13}$$
$$G_2 = hf(x_n + h, y_n + G_1). \tag{14}$$

In mathematics, these formulas are refered as Heun's method, which is also a two-stage second-order Runge-Kutta method. In order to map it to a CNN block, we use the aforementioned G. Figure 1(c) further illustrates the interpretation of these formulas. Compared with ResNet, there are multiple branches in RK2-block, which is commonly used in recent popular networks.

RK3-Block: The knowledge of numerical ODEs suggest that higher-order methods (e.g., with order p) obtain a smaller local truncation error (e.g., $O(h^{p+1})$). This fact inspires us to explore higher stages Runge-Kutta methods.



(a) Residual-block (b) LF-block (c) RK2-block (d) RK3-block

Figure 1. (a) **Residual block** is widely used in previous works, e.g., EDSR [31]; (b) **LF-block** is derived from the Leapfrog method, we combine three Leapfrog steps into a CNN block to compensate the missing term y_n in formula (7); (c) **RK2-block** is derived from the second-order Runge-Kutta method, also known as Heun's method; (d) **RK3-block** is derived from the third-order Runge-Kutta method. In all ODE-inspired blocks, we do not constrain the specific form of G (detailed in section 3.2).

Formally, explicit iterative Runge-Kutta methods can be extended to arbitrary n stages by using the following formulas

$$y_{n+1} = y_n + \sum_{i=1}^{n} \gamma_i G_i \tag{15}$$

$$G_1 = hf(x_n, y_n) \tag{16}$$

$$G_i = hf(x_n + \alpha_i h, y_n + \sum_{=1}^{i-1} \beta_{ij} k_j) \tag{17}$$

In particular, 3-stage Lunge-Kutta with third order can be described as

$$y_{n+1} = y_n + \frac{1}{6}(G_1 + 4G_2 + G_3) \tag{18}$$

$$G_1 = hf(x_n, y_n) \tag{19}$$

$$G_2 = hf(x_n + \frac{h}{2}, y_n + \frac{1}{2}G_1) \tag{20}$$

$$G_3 = hf(x_n + h, y_n - G_1 + 2G_2). \tag{21}$$

(Please refer to the Appendix for the derivation of α, β, γ if you are not familiar with numerical ODEs). It is straightforward to map these equations to a CNN block: we just replace G_1, G_2, G_3 with module G as defined above. RK3-block has more branches because 3-stage Runge-Kutta method takes a computation pattern with higher complexity. Generally, higher-order methods tend to generate more complicated blocks.

It should be noted that in (13), (14) and (19)∼(21), G is defined as a function with two variables. Compared with y corresponding to the featuremaps, the semantics of x in CNN is implicit. It can be granted as a snapshot of time, indicating the position of G in a deep neural network. The initial status $y_0 \in \mathbb{R}^d$ is the input featuremaps of the first OISR-blocks. $y(X; y_0) \in \mathbb{R}^d$ refers to the output of the last OISR-blocks with fix time horizon X (i.e., given finite depth). $d = C \times H \times W$ where C, H, and W is channel number, the height of featuremaps and width of featuremaps, respectively. Note that we keep the dimension of input and output featuremaps of G unchanged. This allows us to finish the bridge between ODE and CNNs.

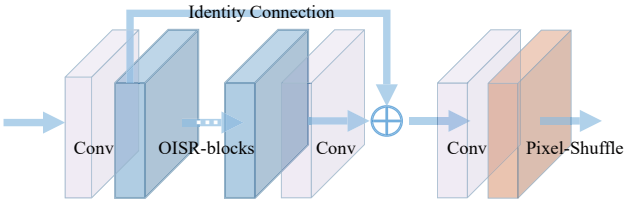


Figure 2. The overall architecture of the proposed ODE-inspired super-resolution network (OISR). For ×2/×3 super-resolution, we use PixelShuffle ×2/×3. In the ×4 model, the upsampler is changed to the cascading of two "Conv+Shuffle×2" modules. We follow the same setting in EDSR [31] since the standard residual-block is widely used. It is clear that OISR can be easily combined with attention mechanisms and dense connections to further improve the performance.

3.2. Overall architecture for OISR

The overall architecture is a Convolution-PixelShuffle framework, depicted in Figure 2. We emphasize that the design scheme is fully inspired by the numerical methods of ODEs, all we need to do is mapping the numerical scheme into a CNN block. Following the setting in [1, 31, 29, 37], we do not use batch normalization layers. Strictly speaking, equation (4) takes the form: $y_{n+1} = y_n + h_n f(x_n, y_n)$. By adaptively choosing h_n (in this case, learned parameter α in ParametricReLU [15]), one can improve the efficiency and stability of the algorithm at the same time [25]. Besides, the small initialization of α corresponds to the fact that the precision of numerical ODEs is related to the step size h_n. If it weren't for finite precision arithmetic, the truncation error goes to zero as h_n goes to zero [34].

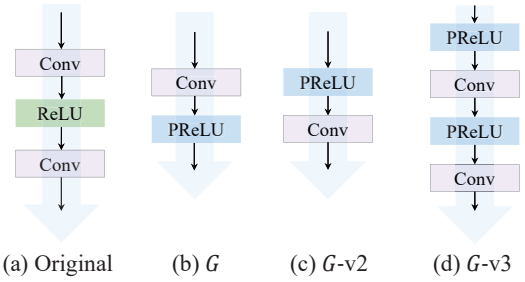


Figure 3. Different structures of module G. (a) is the original G used in EDSR [31]. (b) and (c) consist of a ParametricReLU and a convolutional module. (d) defines an augmented G with the same computing cost as (a).

$\mathbf{G}(\cdot)$: There is a large searching space to search G. Here, we only choose three different forms to illustrate the general effectiveness of ODE-inspired schemes. Each of these designs keeps at least one activation function and one convolutional layer, thus promising the nonlinearity. In fact, different numerical methods (in this case, ODE-inspired blocks) describe different approximation strategies overall, while different G determines local behaviors in each step. Either higher-order method or refined G tends to improve performance.

We build small-scale networks using LF-block and RK2-block for each G. Since different forms of G vary in computation overhead, these networks are developed using different numbers of building blocks but maintaining comparable computation and parameters. This ensures us to have a fair comparison with other lightweight models. Then we develop middle-scale models using LF-block and RK2-block with a similar size and computing cost as the state-of-the-art MSRN [29]. This also enables us to verify that the performance will not degrade as the networks become larger. Finally, we design a deep network using RK3-blocks. Since 3-stage Runge-Kutta is a third-order method, which is supposed to outperform low-order methods in approximation accuracy. For a fair comparison, we keep the overall parameters of RK3 model almost the same as EDSR [31], since residual block has only one G module yet RK3 consists of three G modules.

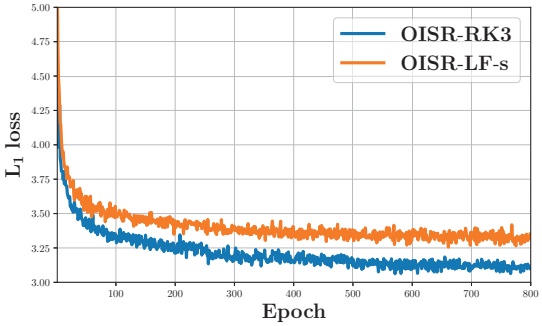


Figure 4. The error curves of training $\times 2$ OISR-LF-s (1.37M parameters) and $\times 2$ OISR-RK3 (42M parameters). Our deep model is easy to train and enjoys similar convergence speed as shallow networks.

Table 1. Ablation studies on $G(\cdot)$. PSNR (db) in terms of $2\times$ SR on DIV2K validation set was reported. "-s" denotes the single scale baseline of EDSR and the small-scale version of OISR.

Method	$G(\cdot)$	PSNR	Params
EDSR-s	Conv+ReLU+Conv	34.61	1.37M
OISR-LF-s	Conv + PReLU	34.64	1.37M
	PReLU + Conv	**34.67**	1.37M
	(PReLU + Conv) x2	34.66	1.37M
OISR-RK2-s	Conv + PReLU	34.62	1.37M
	PReLU + Conv	34.59	1.37M
	(PReLU + Conv) x2	**34.63**	1.37M

4. Experiments

4.1. Datasets

Following the setting in [31, 42, 1, 29], we train our models on the $1^{th} \sim 800^{th}$ training images in DIV2K [39], and evaluate our models on four standard benchmark datasets: Set5 [2], Set14 [44], B100 [33] and Urban100 [18]. The ablation studies on $G(\cdot)$ is determnied on the 100 validation images from DIV2K dataset. We report the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) on the Y channel (i.e., luminance) of transformed YCbCr space and ignore the same amount of pixels as scale from the border [31, 29]. Specifically, for DIV2K validation set, we measure PSNR on full RGB channels and remove the (6+scale) pixels from each border to make a fair comparsion with EDSR. Upscaling factors: $\times 2$, $\times 3$, $\times 4$ are used for training and testing.

Table 2. Quantitative comparisons of our models with the well-designed lightweight methods on benchmark datasets (PSNR(dB) / SSIM). Red indicates the best performance and blue indicates the second best. "MAC" denotes the number of multiply-accumulate operations ($a \leftarrow a + (b \times c)$). The smale-scale network designs are suffixed by "-s". We assume that the generated SR image is 720P (1280×720).

Method	Scale	Params	MAC	Set5		Set14		B100		Urban100	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
FSRCNN [12]	×2	0.01M	6.0G	37.00	0.9558	32.63	0.9088	31.53	0.8920	29.88	0.9020
DRRN [37]	×2	0.30M	6796.9G	37.74	0.9591	33.23	0.9136	32.05	0.8973	31.23	0.9188
MemNet [38]	×2	0.68M	623.9G	37.78	0.9597	33.28	0.9143	32.08	0.8978	31.31	0.9195
SelNet [10]	×2	0.97M	225.7G	37.89	0.9598	33.61	0.9160	32.08	0.8984	–	–
CARN[1]	×2	1.59M	222.8G	37.76	0.9590	33.52	0.9166	32.09	0.8978	31.92	0.9256
OISR-RK2-s	×2	1.37M	316.2G	37.98	0.9604	33.58	0.9172	32.18	0.8996	32.09	0.9281
OISR-LF-s	×2	1.37M	316.2G	38.02	0.9605	33.62	0.9178	32.20	0.9000	32.21	0.9290
MSRN [29]	×2	5.89M	1356.8G	38.08	0.9605	33.74	0.9170	32.23	0.9013	32.22	0.9326
OISR-RK2	×2	4.97M	1145.7G	38.12	0.9609	33.80	0.9193	32.26	0.9006	32.48	0.9317
OISR-LF	×2	4.97M	1145.7G	38.12	0.9609	33.78	0.9196	32.26	0.9007	32.52	0.9320
FSRCNN [12]	×3	0.01M	5.0G	33.16	0.9140	29.43	0.8242	28.53	0.7910	26.43	0.8080
DRRN [37]	×3	0.30M	6796.9G	34.03	0.9244	29.96	0.8349	28.95	0.8004	27.53	0.8378
MemNet [38]	×3	0.68M	623.9G	34.09	0.9248	30.00	0.8385	28.96	0.8001	27.56	0.8376
SelNet [10]	×3	1.16M	120.0G	34.27	0.9257	30.30	0.8399	28.97	0.8025	–	–
CARN[1]	×3	1.59M	118.8G	34.29	0.9255	30.29	0.8407	29.06	0.8034	28.06	0.8493
OISR-RK2-s	×3	1.55M	160.1G	34.43	0.9273	30.33	0.8420	29.10	0.8053	28.20	0.8534
OISR-LF-s	×3	1.55M	160.1G	34.39	0.9272	30.35	0.8426	29.11	0.8058	28.24	0.8544
MSRN [29]	×3	6.08M	621.2G	34.38	0.9262	30.34	0.8395	29.08	0.8041	28.08	0.8554
OISR-RK2	×3	5.64M	578.6G	34.55	0.9282	30.46	0.8443	29.18	0.8075	28.50	0.8597
OISR-LF	×3	5.64M	578.6G	34.56	0.9284	30.46	0.8450	29.20	0.8077	28.56	0.8606
FSRCNN [12]	×4	0.01M	4.6G	30.48	0.8628	27.49	0.7503	26.90	0.7101	24.52	0.7221
DRRN [37]	×4	0.30M	6796.9G	31.68	0.8888	28.21	0.7720	27.38	0.7284	25.44	0.7638
MemNet [38]	×4	0.68M	623.9G	31.74	0.8893	28.26	0.7723	27.40	0.7281	25.50	0.7630
SelNet [10]	×4	1.42M	83.1G	32.00	0.8931	28.49	0.7783	27.44	0.7325	–	–
CARN[1]	×4	1.59M	90.9G	32.13	0.8937	28.60	0.7806	27.58	0.7349	26.07	0.7837
OISR-RK2-s	×4	1.52M	114.2G	32.21	0.8950	28.63	0.7822	27.58	0.7364	26.14	0.7874
OISR-LF-s	×4	1.52M	114.2G	32.14	0.8947	28.63	0.7819	27.60	0.7369	26.17	0.7888
MSRN [29]	×4	6.33M	365.1G	32.07	0.8903	28.60	0.7751	27.52	0.7273	26.04	0.7896
OISR-RK2	×4	5.50M	412.2G	32.32	0.8965	28.72	0.7843	27.66	0.7390	26.37	0.7953
OISR-LF	×4	5.50M	412.2G	32.33	0.8968	28.73	0.7845	27.66	0.7389	26.38	0.7953

4.2. Training details

In the training phase, we use the RGB input patches of size 48×48 from the low-resolution image with the corresponding high-resolution patches. All the images are preprocessed by subtracting the mean RGB value of the DIV2K dataset and then augmented with random horizontal flips and 90° rotations [31]. We set the minibatch size as 16 and use ADAM optimizer to train our model with the settings of $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. The learning rate is initialized as 0.0001 and halved at every 250 epochs. Training is terminated at 800 epochs. The objective of training OISR is the popular ℓ_1 loss function.

4.3. Results on benchmark datasets

We first do the ablation study on the implementation of G. As listed in Table 1, "PReLU+Conv", namely G-v2 is suitable for LF-blocks and RK2-blocks should be equipped with G-v3. As mentioned in previous works [1, 31, 29], deep models are difficult to train. To fully examine the effectiveness of the ODE-inspired scheme, we conduct the seemingly irrational behavior that applying the worst-performing G-v2 in OISR-RK2 to our deep model OISR-RK3. Then, we compare our results with other state-of-the-arts on two commonly-used metrics PSNR and SSIM. As presented in Table 2, our small-scale models outperform other methods on different upscaling factors and datasets, except a slightly behind on Urban100 with upscaling factor $\times 2$. In addition, we compare our middle-scale models with MSRN. Our networks surpass MSRN with only two exceptions on B100 and Urban100 SSIM when the upscaling factor is 2. These results illustrate that our methods better overcome the dilemma of performance enhance-

Table 3. Quantitative comparisons of our models with hand-crafted deep residual SISR networks on benchmark datasets (PSNR(dB) / SSIM). Red indicates the best performance and blue indicates the second best. We assume that the generated SR image is 720P (1280×720).

Method	Scale	Params	MAC	Set5		Set14		B100		Urban100	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
LapSRN [24]	×2	0.81M	29.9G	37.52	0.9581	33.08	0.9109	31.80	0.8949	30.41	0.9112
VDSR [21]	×2	0.67M	612.6G	37.53	0.9587	33.03	0.9127	31.90	0.8960	30.76	0.9140
DRCN [23]	×2	1.77M	17974G	37.63	0.9588	33.04	0.9118	31.85	0.8942	30.75	0.9133
MDSR [31]	×2	6.92M	1592.2G	38.11	0.9602	33.85	0.9198	32.29	0.9007	32.84	0.9347
RDN [42]	×2	22.12M	5096.2G	38.24	0.9614	34.01	0.9212	32.34	0.9017	32.89	0.9353
EDSR [31]	×2	40.73M	9384.7G	38.11	0.9601	33.92	0.9195	32.32	0.9013	32.93	0.9351
OISR-RK3	×2	41.91M	9656.5G	38.21	0.9612	33.94	0.9206	32.36	0.9019	33.03	0.9365
VDSR [21]	×3	0.67M	612.6G	33.66	0.9213	29.77	0.8314	28.82	0.7976	27.14	0.8279
DRCN [23]	×3	1.77M	17974G	33.82	0.9226	29.76	0.8311	28.80	0.7963	27.15	0.8276
MDSR [31]	×3	7.51M	768.1G	34.66	0.9280	30.44	0.8452	29.25	0.8091	28.79	0.8655
RDN [42]	×3	22.31M	2281.2G	34.71	0.9296	30.57	0.8468	29.26	0.8093	28.80	0.8653
EDSR [31]	×3	43.68M	4469.5G	34.65	0.9282	30.52	0.8462	29.25	0.8093	28.80	0.8653
OISR-RK3	×3	44.86M	4590.1G	34.72	0.9297	30.57	0.8470	29.29	0.8103	28.95	0.8680
LapSRN [24]	×4	0.81M	149.4G	31.54	0.8855	28.19	0.7722	27.32	0.7280	25.21	0.7553
VDSR [21]	×4	0.67M	612.6G	31.35	0.8838	28.01	0.7674	27.29	0.7251	25.18	0.7524
DRCN [23]	×4	1.77M	17974G	31.53	0.8854	28.02	0.7670	27.23	0.7233	25.14	0.7510
MDSR [31]	×4	7.88M	480.4G	32.50	0.8973	28.72	0.7857	27.72	0.7418	26.67	0.8041
RDN [42]	×4	22.27M	1309.2G	32.47	0.8990	28.81	0.7871	27.72	0.7419	26.61	0.8028
EDSR [31]	×4	43.10M	2894.5G	32.46	0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033
OISR-RK3	×4	44.27M	2962.5G	32.53	0.8992	28.86	0.7878	27.75	0.7428	26.79	0.8068

ment and computation overhead.

For current state-of-the-art deep residual methods in Table 3, OISR-RK3 achieves the best performances in most cases. Though we apply the worst-performing G in OISR-RK2 to OISR-RK3, it still achieves noticeable results. Moreover, Figure 4 further presents a comparison of the convergence rate between OISR-LF-s and OISR-RK3, which indicates that deeper OISR-RK3 is not difficult to train compared with much smaller OISR-LF-s. These results empirically verify the effectiveness of ODE-inspired design scheme, and the behaviors of OISR tend to vary from different numerical ODEs with different order of truncation error.

4.4. Discussion on ODE-inspired design schemes

In this work, we have developed an ODE-Inspired scheme for SISR. Novel architectures are designed by introducing numerical ODEs into CNNs. Table 1,3 have illustrated that, in the case of comparable computation and parameters, OISR-LF and OISR-RK outperform EDSR [31]. These results suggest the superiority of our methods. Here we expand the discussion in ODE viewpoint.

As mentioned in section 3.1, residual block, LF-block, and RK2-block can be regarded as mappings of numerical ODEs. EDSR develops deep architecture using residual block, related to the first-order forward Euler method. We propose to build our networks using RK-block and LF-block, corresponding to the higher order methods in numerical ODEs. Experimental results suggest that higher order methods tend to enhance the performances. A similar fact holds for deep networks as presented in Table 3, third-order method RK3 performs better than first-order EDSR. If we take the dynamical system viewpoint, higher-order methods tend to make a better approximation of the map locally, which enables them to approach the final target with smaller global (truncation) error, sustained in all the steps.

Table 1 presents the performances of proposed models with different G. Our small-scale model outperforms EDSR for each G with the same parameters and computing cost. These results illustrate that ODE-inspired scheme is generally effective for SISR. Note that for OISR-RK2 and OISR-LF, the best-performing G is different. OISR-LF tends to take a simple G-v2 while OISR-RK2 prefers an augmented G-v3. By adopting the view of the dynamical system, since we keep a comparable computation, networks with G-v2 allows more steps to give rise to the target (smaller h_n), while networks with G-v3 provides a better local approximation. In order to control the amount of computation, augmented G and more building blocks become the two sides of a coin. Leaves out the computation overhead, one could enhance the performance by designing finer G or using more building blocks, as two strategies in deepening our networks.

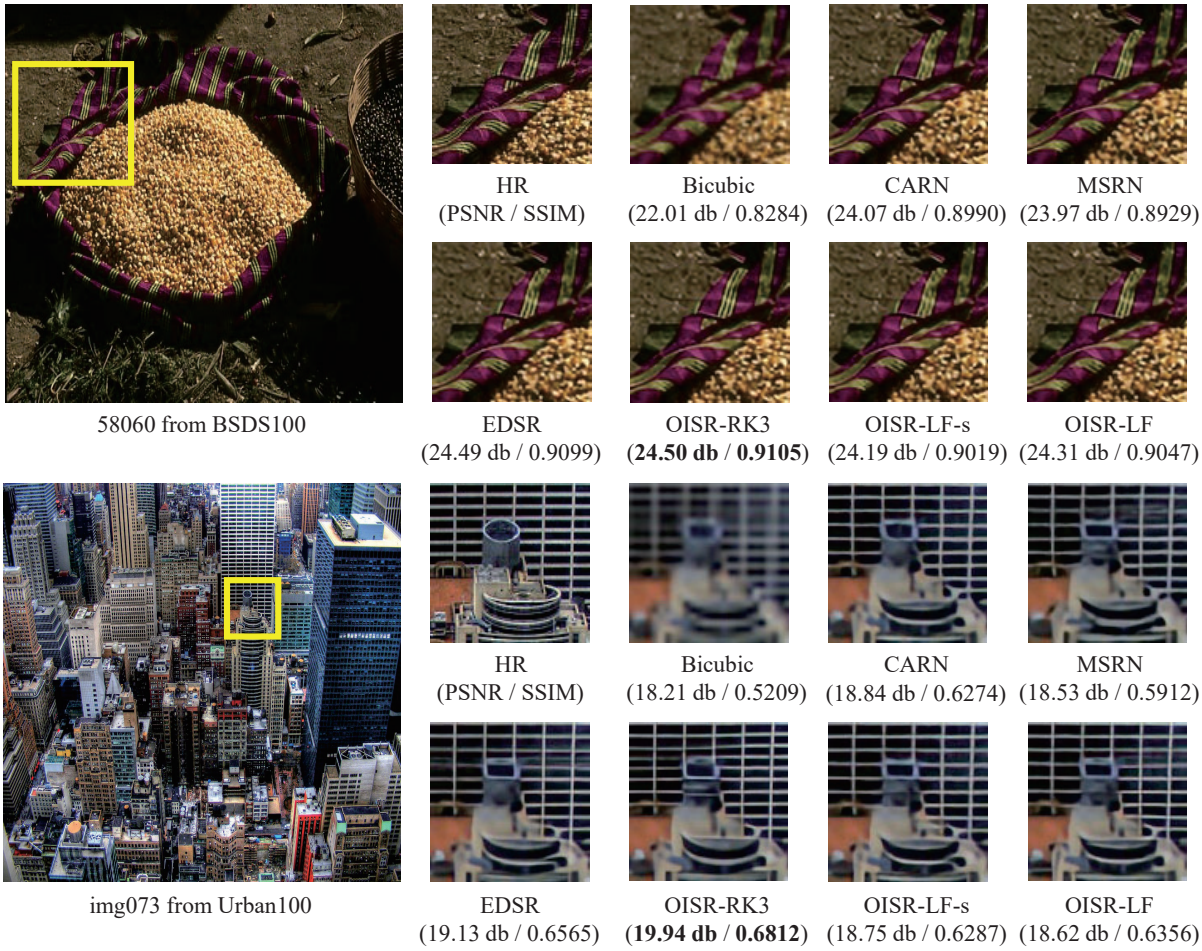


Figure 5. Qualitative comparsions of our methods with other state-of-the-arts on $\times 2$ super resolution (top) and $\times 4$ super resolution (bottom). OISRs can reconstruct more detailed images with less blurring.

In general, these results not only illustrate the effectiveness of ODE-inspired design scheme but suggest the reasonability of ODE perspective.

5. Conclusions

In this paper, we propose to apply ODE-inspired scheme to design CNN for SISR. By introducing the concept of dynamical system one can establish the bridge between CNN and numerical ODEs. This connection enables us to design LF-block, RK2-block, and RK3-block inspired by Leapfrog method and Runge-Kutta method in numerical analysis. Experimental results show that our methods exceed other state-of-the-art methods while keeping comparable computation, which relieves the dilemma of enhancing performance and reducing computation overhead. To fully verify the effectiveness of our method, we design several building blocks with different G. Experimental results demonstrate that ODE-inspired scheme works well in most cases. This finding provides us with a relatively reliable guideline to design networks for SISR.

6. Acknowledgement

This work was supported in part by National Natural Science Foundation of China (No.61876182, 61872364), the Strategic Priority Research Program of Chinese Academy of Science(No.XDB32050200).

References

[1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*, pages 256–272, 2018.

[2] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding.

In *British Machine Vision Conference, BMVC 2012, Surrey, UK, September 3-7, 2012*, pages 1–10, 2012.

[3] David P. Capel and Andrew Zisserman. Super-resolution from multiple views using learnt image models. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM, 8-14 December 2001, Kauai, HI, USA*, pages 627–634, 2001.

[4] Ayan Chakrabarti, A. N. Rajagopalan, and Rama Chellappa. Super-resolution of face images using kernel pca-based prior. *IEEE Trans. Multimedia*, 9(4):888–892, 2007.

[5] Bo Chang, Lili Meng, Eldad Haber, Lars Ruthotto, David Begert, and Elliot Holtham. Reversible architectures for arbitrarily deep residual neural networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 2811–2818, 2018.

[6] Bo Chang, Lili Meng, Eldad Haber, Frederick Tung, and David Begert. Multi-level residual networks from dynamical systems view. *CoRR*, abs/1710.10348, 2017.

[7] Bo Chang, Lili Meng, Eldad Haber, Frederick Tung, and David Begert. Multi-level residual networks from dynamical systems view. *CoRR*, abs/1710.10348, 2017.

[8] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), with CD-ROM, 27 June - 2 July 2004, Washington, DC, USA*, pages 275–282, 2004.

[9] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 6572–6583, 2018.

[10] Jae-Seok Choi and Munchurl Kim. A deep convolutional neural network with selection units for super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, Honolulu, HI, USA, July 21-26, 2017*, pages 1150–1156, 2017.

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV*, pages 184–199, 2014.

[12] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, pages 391–407, 2016.

[13] Gilad Freedman and Raanan Fattal. Image and video upscaling from local self-examples. *ACM Trans. Graph.*, 30(2):12:1–12:11, 2011.

[14] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, pages 349–356, 2009.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1026–1034, 2015.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pages 630–645, 2016.

[18] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 5197–5206, 2015.

[19] John A. Kennedy, Ora Israel, Alex Frenkel, Rachel Bar-Shalom, and Haim Azhari. Super-resolution in PET imaging. *IEEE Trans. Med. Imaging*, 25(2):137–147, 2006.

[20] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1646–1654, 2016.

[21] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1646–1654, 2016.

[22] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1637–1645, 2016.

[23] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1637–1645, 2016.

[24] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5835–5843, 2017.

[25] John Denholm Lambert. *Numerical methods for ordinary differential systems: the initial value problem*. John Wiley & Sons, Inc., 1991.

[26] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *CoRR*, abs/1605.07648, 2016.

[27] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken,

Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 105–114, 2017.

[28] Bo Li, Hong Chang, Shiguang Shan, and Xilin Chen. Low-resolution face recognition via coupled locality preserving mappings. *IEEE Signal Process. Lett.*, 17(1):20–23, 2010.

[29] Juncheng Li, Faming Fang, Kangfu Mei, and Guixu Zhang. Multi-scale residual network for image super-resolution. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII*, pages 527–542, 2018.

[30] Qianli Liao and Tomaso A. Poggio. Bridging the gaps between residual learning, recurrent neural networks and visual cortex. *CoRR*, abs/1604.03640, 2016.

[31] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, Honolulu, HI, USA, July 21-26, 2017*, pages 1132–1140, 2017.

[32] Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 3282–3291, 2018.

[33] David R. Martin, Charless C. Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, pages 416–425, 2001.

[34] Philip Robinson. Differential equations: a dynamical systems approach, part 1. *The Mathematical Gazette*, 76(477):430–430, 1992.

[35] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 4278–4284, 2017.

[36] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2790–2798, 2017.

[37] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2790–2798, 2017.

[38] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 4549–4557, 2017.

[39] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee, Xintao Wang, Yapeng Tian, Ke Yu, Yulun Zhang, Shixiang Wu, Chao Dong, Liang Lin, Yu Qiao, Chen Change Loy, Woong Bae, Jae Jun Yoo, Yoseob Han, Jong Chul Ye, Jae-Seok Choi, Munchurl Kim, Yuchen Fan, Jiahui Yu, Wei Han, Ding Liu, Haichao Yu, Zhangyang Wang, Honghui Shi, Xinchao Wang, Thomas S. Huang, Yunjin Chen, Kai Zhang, Wangmeng Zuo, Zhimin Tang, Linkai Luo, Shaohui Li, Min Fu, Lei Cao, Wen Heng, Giang Bui, Truc Le, Ye Duan, Dacheng Tao, Ruxin Wang, Xu Lin, Jianxin Pang, Jinchang Xu, Yu Zhao, Xiangyu Xu, Jin-shan Pan, Deqing Sun, Yujin Zhang, Xibin Song, Yuchao Dai, Xueying Qin, Xuan-Phung Huynh, Tiantong Guo, Hojjat Seyed Mousavi, Tiep Huu Vu, Vishal Monga, Cristóvão Cruz, Karen O. Egiazarian, Vladimir Katkovnik, Rakesh Mehta, Arnav Kumar Jain, Abhinav Agarwalla, Ch V. Sai Praveen, Ruofan Zhou, Hongdiao Wen, Che Zhu, Zhiqiang Xia, Zhengtao Wang, and Qi Guo. NTIRE 2017 challenge on single image super-resolution: Methods and results. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, Honolulu, HI, USA, July 21-26, 2017*, pages 1110–1121, 2017.

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

[41] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5987–5995, 2017.

[42] Jiu Xu, Yeongnam Chae, Björn Stenger, and Ankur Datta. Dense bynet: Residual dense network for image super resolution. In *2018 IEEE International Conference on Image Processing, ICIP 2018, Athens, Greece, October 7-10, 2018*, pages 71–75, 2018.

[43] Jianchao Yang, John Wright, Thomas S. Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE Trans. Image Processing*, 19(11):2861–2873, 2010.

[44] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces - 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers*, pages 711–730, 2010.

[45] Xingcheng Zhang, Zhizhong Li, Chen Change Loy, and Dahua Lin. Polynet: A pursuit of structural diversity in very deep networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3900–3908, 2017.

[46] Yun Zhang. Problems in the fusion of commercial high-resolution satellites images as well as landsat 7 images and initial solutions. In *Proceedings of the ISPRS, CIG, and SDH Joint International Symposium on Geospatial Theory, Processing and Applications*, pages 9–12, 2002.