Image Restoration by Solving IVP

Seobin Park and **Tae Hyun Kim**
Hanyang University, Seoul, South Korea
{seobinpark, taehyunkim}@hanyang.ac.kr

Abstract

Recent research on image restoration have achieved great success with the aid of deep learning technologies, but, many of them are limited to dealing SR with realistic settings. To alleviate this problem, we introduce a new formulation for image super-resolution to solve arbitrary scale image super-resolution methods. Based on the proposed new SR formulation, we can not only super-resolve images with multiple scales, but also find a new way to analyze the performance of super-resolving process. We demonstrate that the proposed method can generate high-quality images unlike conventional SR methods.

1 Introduction

Super-resolution (SR) task is used to increase the image resolution by estimating its underlying high frequency details. However, SR is a highly ill-posed problem since for any lowresolution input, there are multiple high-resolution solutions, which makes it a challenging problem. Therefore, considerable methods have been studied to solve SR problem. First, interpolation-based methods are simple and efficient, and thus they are widely used in many applications. However, these naive approaches have a clear performance limitation. Deep-learning based methods are being very successful in generating high quality images from lowresolution image, and provide quantitatively promising results. Moreover, several generative SR networks which are composed of a high-resolution image generator and discriminator, can generate visually more plausible results with the help of numerous perceptual losses (e.g. VGG and adversarial losses). However, these previous learning-based SR approaches are limited to fixed scaling factors (e.g., x2, x3, and x4) to allow quick inference. Therefore, there has been several researches to solve SR problem with an arbitrary scaling factor. Recently, there has been increasing attempts to model differential equations with neural network. In particular, Neural ordinary differential equations (Neural ODE), allows to formulate a differential path from low-resolution input to highresolution output with a neural network. However, this Neural ODE based SR approach uses the Neural ODE as an intermediate layer of HR generator without clear reasoning in the process of modeling a differential equation. Therefore, in this work, we tackle this problem and present a new SR approach with Neural ODE, and our contributions can be summarized as follows:

- We propose a new formulation to perform superresolution on arbitrary scale.
- We find a way to analyze the differential path from low-resolution to high-resolution.
- Our method achieves performance close to existing state-of-the-art methods.

2 Related Works

2.1 Image Restoration

Recent works on single image restoration focus on learning mapping functions between degraded image and original image. SRCNN [Dong *et al.*, 2015] proposed to learn the non-linear mapping from LR image to HR image using a CNN model for the first time. VDSR [Kim *et al.*, 2016b] increased the depth of CNN to model more complex LR-HR mappings. Recent studies have applied different kinds of skip connections to ease the optimization process [Lim *et al.*, 2017; Zhang *et al.*, 2018].

Parallel to devising improved feed-forward CNN architectures, many attempts have been done to develop SISR methods that can be better applied to real-world situations. RealSR [Cai *et al.*, 2019] proposed a more realistic degradation model to make more natural training dataset and presented a new method to cope with the given degradation settings. Meta-SR [Hu *et al.*, 2019] proposed Meta-Upscale module to handle arbitrary scale factors for SISR.

2.2 Gradual Image Restoration

gradual approaches have been proven to perform well in low-level vision tasks. In addition to learning a complex non-linear mapping of a low-quality image to a high-quality one, these approaches decompose this process into multiple steps and iteratively refine the output images [Kim *et al.*, 2016a; Tai *et al.*, 2017; Haris *et al.*, 2018; Li *et al.*, 2019]. DBPN [Haris *et al.*, 2018] proposed a dedicated neural network design that provides an iterative error correcting mechanism to address mutual dependencies of LR and HR images. SRFBN [Li *et al.*, 2019] proposed an efficient recurrent neural network that employs the feedback mechanism, which iteratively improves the input of the network in each step. However, these methods do not have a clear underlying formulation or theoretical analysis on the gradual image restoration process, which consequently require a large amount of engineering to develop the neural architectures [Li *et al.*, 2019; Haris *et al.*, 2018] and specialized training strategies [Li *et al.*, 2019; Kim *et al.*, 2016a]. LapSRN [Lai *et al.*, 2017] can produce large SR results (e.g., x8) with intermediate SR results (e.g., x2, x4), but it can only handle the pre-determined discrete scale factors such as x2, x4, and x8.

2.3 Neural Ordinary Differential Equation

Recently, many attempts have been done to integrate differential formulations and deep learning methodologies. These attempts have led to neural ordinary differential equations (NODE) [Chen *et al.*, 2018]. NODE is a new family of deep neural network models that parameterizes a differential form using a neural network and produces the output by using an ODE solver. Meanwhile, differential equations have often been involved in image restoration task by modeling non-linear reaction-diffusion and total variation schemes [Chen and Pock, 2016; Rudin *et al.*, 1992]. Such is also the case in deep-learning-based image restoration methods. [He *et al.*, 2019] designed new neural network architectures inspired by the differential equation solving process, such as Leapfrog and Runge–Kutta approaches. In particular, to solve the SR problem, [Scao, 2020] utilized NODE as a part of neural architecture with an integration. However, the interval of the integral is not fixed (open) and they did not present any concrete formulation on the intermediate images involved in the gradual SR process. Consequently, they need to empirically speculate the optimal neural architecture and the integration interval.

3 Proposed Method

In this section, we present a new neural approach for the gradual SR reconstruction. We first formulate the SR problem as a gradual SR process with an ODE. We then elaborate how to perform SR with the proposed formulation and how to train it.

3.1 gradual Super-Resolution Formulation

Existing SR methods utilizing gradual SR process [Lai *et al.*, 2017; Haris *et al.*, 2018; Li *et al.*, 2019] are based on iterative multi-stage approaches and can be viewed as variants of the following:

$$I_n = g_{n-1}(I_{n-1}) \quad (n \leq N), \tag{1}$$

where n denotes the iteration step, I_0 denotes the given initial input LR image, and I_n is the iteratively refined image from its previous state I_{n-1}. These approaches typically produce multiple intermediate HR images during the refinement, and the rendered image at the last N-th iteration [Tai *et al.*, 2017; Li *et al.*, 2019] or a combined version of the multiple intermediate images ($\{I_n\}_{1\leq n\leq N}$) [Kim *et al.*, 2016a; Haris *et al.*, 2018] becomes the final SR result. Although these previous gradual methods show promising SR results, they still have some limitations. First, these methods need plenty of time and effort in determining the network configurations including the number of gradual updates N and hyper-parameter settings, and designing cost functions to train the SR networks g. In addition, well-engineered and dedicated learning strategy, such as curriculum learning [Li *et al.*, 2019] and recursive supervision [Kim *et al.*, 2016a], is required for each method. This complication comes from the lack of clear understanding on their intermediate image states $\{I_n\}$. To alleviate these problems, we formulate the gradual SR process with a differential equation. This allows us to implement and train the SR networks in an established way while outperforming the performance of conventional gradual SR process.

Assume that $(I_{HR})\downarrow_t$ is a downscaled version of a ground-truth clean image I_{HR} using a traditional SR kernel (e.g., bicubic) with a scaling factor $\frac{1}{t}$. We then define $\mathcal{I}(t)$ by upscaling $(I_{HR})\downarrow_t$ using that SR kernel with a scaling factor t so that I_{HR} and $\mathcal{I}(t)$ have the same spatial resolution (see the illustration of "Generating LR image" in (a). Note that $t \geq 1$, and $\mathcal{I}(1)$ denotes the ground-truth clean image I_{HR}. To model a gradual SR process,we first estimate the high-frequency image residual with a neural network. Specifically, when t is a conventional discrete-scaling factor (e.g., x2, x3, and x4), image residual between $\mathcal{I}(t)$ and $\mathcal{I}(t-1)$ can be modeled using a neural network f_{discrete} as:

$$\mathcal{I}(t-1) - \mathcal{I}(t) = f_{\text{discrete}}(\mathcal{I}(t), t). \tag{2}$$

Notably, $\mathcal{I}(t-1)$ includes more high-frequency details than $\mathcal{I}(t)$ without loss of generality. In our method, we model the slightest image difference to formulate a continuously gradual SR process. Therefore, we take the scale factor t to continuous domain, and reformulate (2) as an ODE with a neural network f as:

where θ denotes the trainable parameter of the network f. Using this formulation, we can predict the high-frequency image detail required to slightly enhance $\mathcal{I}(t)$ with the network f. (Note that we can obtain $\mathcal{I}(t)$ with any rational number t by adding padding to the border of image before resizing and then center cropping the image.) As existing SR neural networks have been proven to be successful at predicting the high-frequency residual image [Kim *et al.*, 2016b], we can use conventional SR architectures as our network f without major changes.

3.2 Single Image Super-Resolution with Neural Ordinary Differential Equation

In this section, we explain how to super-resolve a given LR image with a continuous scaling factor using our ODE-based SR formulation.

First, we obtain $\mathcal{I}(t_0)$ by upscaling the given LR input image ("Test time LR image" in (a) using the bicubic SR kernel to a desired output resolution with a scaling factor t_0 . Next, we solve the ODE initial value problem with the initial condition $\mathcal{I}(t_0)$ by integrating the neural network f from t_0 to 1 to acquire the high-quality image $\mathcal{I}(1)$ as follows:

As the neural network f is modeled to predict desired high-frequency details, our formulation gradually adds the predicted fine details from the input LR image $\mathcal{I}(t_0)$ through the integration shown as the solid orange line in (a). Thus, our SR approach becomes a gradual SR process which adds the high-frequency details gradually. To compute the integration with f in the proposed formulation, we use conventional ODE solvers to numerically calculate the output image $\mathcal{I}(1)$. Specifically, we approximate the high-quality image $\mathcal{I}(1)$ given a fully trained neural network f, network parameter θ, initial condition $\mathcal{I}(t_0)$, and integral interval $[t_0, 1]$ using an ODE solver ($ODESolve()$) as:

$$\mathcal{I}(1) \approx HR \tag{3}$$

Thus, our method does not need to consider the stop condition (i.e., the number of feedback iterations) of the gradual SR process unlike conventional approaches [Tai *et al.*, 2017; Li *et al.*, 2019]. Notably, we can use conventional ODE solvers to render the desired outputs for the inference, but the solutions should be differentiable to train the network f through the backpropagation scheme. We compare the SR performance with different ODE solvers (e.g., Runge-Kutta and Euler methods) in our experiments.

Our formulation is made upon a continuous context, allows a continuous scale factor t_0 where $t_0 \geq 1$. This makes our method able to handle the arbitrary-scale SR problem. But unlike conventional multi-scale SR methods [Kim *et al.*, 2016b; Lim *et al.*, 2017; Hu *et al.*, 2019] that successfully learn multi-scale SR tasks by sharing common features across various scales, we explicitly learn the relationship between images with different scales in image domain itself rather than the feature space.

3.3 Training

To train the deep neural network f, and learn the parameter θ in (3), we minimize the loss summed over scale factors t using the L1 loss function as:

$$\mathcal{L}(\theta) = \sum_t \|I_{HR} - ODESolve([t, 1])\|_1. \tag{4}$$

By minimizing the proposed loss function, our network parameter θ is trained to estimate the image detail to be added into the network input.

Notably, during the training phase, we need to employ an ODE solver which allows end-to-end training using backpropagation with other components such as the neural network f. Unlike other gradual SR methods [Kim *et al.*, 2016a; Li *et al.*, 2019], we do not require any other learning strategies like curriculum learning during the training phase.

4 Experimental Results

In this section, we carry out extensive experiments to demonstrate the superiority of the proposed method, and add various quantitative and qualitative comparison results. We also provide detailed analysis of our experimental results. We will release our source code upon acceptance.

4.1 Implementation details

Network configuration. We use VDSR [Kim *et al.*, 2016b] and RDN [Zhang *et al.*, 2018] as backbone CNN architectures for our network f with slight modifications.

For each CNN architecture, we change the first layer to feed the scale factor t as an additional input. To be specific, we extend the input channel from 3 to 4, and the pixel locations of the newly concatenated channel (4-th channel) are filled with a scalar value t as shown in (b). In addition, for RDN, we remove the last upsampling layer so that input and output resolutions are the same in our work. Note that, no extra parameters are added except for the first layers of the networks.

To train and infer the proposed SR process, we use the Python torchdiffeq library [Chen *et al.*, 2018] to employ Runge–Kutta (RK4) method as our ODE solver in (4), which requires only 6 additional lines of code with PyTorch.

For simplicity, our approaches with VDSR and RDN backbones are called vdsr and RDN in the remaining parts of the experiments, respectively.

Dataset and evaluation. We use the DIV2K [Agustsson and Timofte, 2017] dataset to train our vdsr and RDN. During the training phase, we augment the dataset using random cropping, rotating, and flipping.

During the test phase, we evaluate the SR results in terms of PSNR and SSIM metrics on the standard benchmark datasets (Set14 [Zeyde *et al.*, 2010], B100 (BSD100) [Martin *et al.*, 2001], and Urban100 [Huang *et al.*, 2015]). To be consistent with previous works, quantitative results are evaluated on the Y (luminance) channel in the YCbCr color space.

Training setting. We train the network by minimizing the L1 loss in (4) with the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) [Kingma and Ba, 2015]. The initial learning rate is set as 10^{-4}, which is then decreased by half every 100k gradient update steps, and trained for 600k iterations in total. The mini-batch size of vdsr is 16 (200x200 patches), but our vdsr takes 8 patches as a mini-batch (130x130 patches) owing to the memory limit of our graphic units. Similar to the training settings in Meta-SR [Hu *et al.*, 2019], we train the network f by randomly changing the scale factor t in (4) from 1 to 4 with a stride of 0.1 (i.e., $t \in \{1.1, 1.2, 1.3, ..., 4\}$).

4.2 Comparison with gradual SR Methods

First, we compare our RDN with several state-of-the-art gradual SR methods: DRCN [Kim *et al.*, 2016a], LapSRN [Lai *et al.*, 2017], DRRN [Tai *et al.*, 2017], D-DPBN [Haris *et al.*, 2018], and SRFBN [Li *et al.*, 2019]. As in [Lim *et al.*, 2017], self-ensemble method is used to further improve RDN (denoted as RDN+). Note that, our RDN and RDN+ can handle multiple scale factors t including non-integer scale factors (e.g., x1.5) using the same network parameter. In contrast, other approaches are required to be trained for certain discrete integer scale factors (x2, x3, and x4) separately, resulting in a distinct parameter set for each scale factor. Nevertheless, quantitative restoration results show that our RDN, RDN+ consistently outperforms conventional gradual SR methods for the discrete integer scaling factors (x2, x3, and x4) in terms of PSNR.

We investigate intermediate images produced during the gradual SR process with the scale factors x2 and x4. Final results by DRRN are obtained after 25 iterations, and the final results by SRFBN are obtained with 4 iterations as in

Methods \ Scale	x1.1	x1.2	x1.3	x1.4	x1.5	x1.6	x1.7	x1.8	x1.9	x2.0	x2.1	x2.2	x2.3	x2.4	x2.5
bicubic	36.56	35.01	33.84	32.93	32.14	31.49	30.90	30.38	29.97	29.55	29.18	28.87	28.57	28.31	28.13
VDSR	-	-	-	-	-	-	-	-	-	31.90	-	-	-	-	-
VDSR+t	39.51	38.44	37.15	36.04	34.98	34.15	33.39	32.78	32.22	31.70	31.27	30.86	30.53	30.2	29.91
vdsr (ours)	**41.46**	**39.36**	**37.75**	**36.51**	**35.38**	**34.49**	**33.70**	**33.07**	**32.50**	**31.95**	**31.52**	**31.09**	**30.76**	**30.42**	**30.12**
RDN	-	-	-	-	-	-	-	-	-	32.34	-	-	-	-	-
RDN+t	42.83	39.92	38.18	36.87	35.71	34.80	33.99	33.34	32.77	32.22	31.76	31.33	30.99	30.64	30.34
Meta-RDN	42.82	40.04	38.28	36.95	35.86	34.90	34.13	33.45	32.86	32.35	31.82	31.41	31.06	30.62	30.45
RDN (ours)	43.22	40.06	38.35	37.02	35.86	34.95	34.14	33.47	32.89	32.34	31.89	31.46	31.12	30.76	30.46
RDN+ (ours)	**43.33**	**40.13**	**38.40**	**37.07**	**35.90**	**34.99**	**34.17**	**33.50**	**32.93**	**32.38**	**31.93**	**31.50**	**31.16**	**30.80**	**30.50**

Methods \ Scale	x2.6	x2.7	x2.8	x2.9	x3.0	x3.1	x3.2	x3.3	x3.4	x3.5	x3.6	x3.7	x3.8	x3.9	x4.0
bicubic	27.89	27.66	27.51	27.31	27.19	26.98	26.89	26.59	26.60	26.42	26.35	26.15	26.07	26.01	25.96
VDSR	-	-	-	-	28.83	-	-	-	-	-	-	-	-	-	27.29
VDSR+t	29.64	29.39	29.15	28.93	28.74	28.55	28.38	28.22	28.05	27.89	27.76	27.58	27.47	27.34	27.20
vdsr (ours)	**29.85**	**29.61**	**29.36**	**29.14**	**28.94**	**28.75**	**28.58**	**28.41**	**28.25**	**28.08**	**27.96**	**27.79**	**27.66**	**27.54**	**27.40**
RDN	-	-	-	-	29.26	-	-	-	-	-	-	-	-	-	27.72
RDN+t	30.06	29.80	29.55	29.33	29.12	28.92	28.76	28.59	28.43	28.26	28.13	27.95	27.84	27.71	27.58
Meta-RDN	30.13	29.82	29.67	29.40	**29.30**	28.87	28.79	28.68	28.54	28.32	28.27	28.04	27.92	27.82	27.75
RDN (ours)	30.18	29.93	29.67	29.45	29.25	29.05	28.88	28.71	28.54	28.37	28.24	28.07	27.96	27.81	27.72
RDN+ (ours)	**30.22**	**29.97**	**29.71**	**29.49**	29.28	**29.05**	**28.92**	**28.74**	**28.58**	**28.41**	**28.28**	**28.12**	**28.00**	**27.87**	**27.75**

Table 1: Average PSNR values on the B100 dataset evaluated with different scale factors. The best performance is shown in **bold number**.

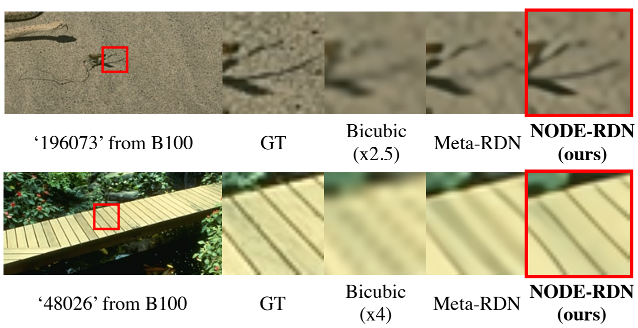

Figure 1: Visual comparison of RDN (ours) with Meta-RDN on scale x2.5 and x4.

their original settings. We provide 4 intermediate HR images during the updates for visual comparisons. For our RDN, intermediate image states are represented as $\hat{\mathcal{I}}(t_i)$ where $1 \leq t_i \leq t_0$ and $\hat{\mathcal{I}}(t_i) = ODESolve(\mathcal{I}(t_0), f, \theta, [t_0, t_i])$. We observe that DRRN and SRFBN fail to gradually refine patches with high-frequency details, while our RDN can gradually improve the intermediate images and render promising results at the final states.

4.3 Comparison with Multi-scale SR Methods

Our approach can handle a continuous scale factor for the SR task, thus we compare ours with existing multi-scale SR methods that can handle continuous scale factors: VDSR [Kim *et al.*, 2016b] and Meta-SR [Hu *et al.*, 2019]. Notably, Meta-SR implemented using RDN (i.e., Meta-RDN) is the current state-of-the-art SR approach.

In Table 1, we show quantitative results compared to existing SR methods (VDSR, RDN, and Meta-RDN). Note that, VDSR+t and RDN+t are modified versions of VDSR and RDN to take the scale factor t as an additional input of the networks and have the same input and output resolutions as in our network f. We also compare our method with these new baselines (VDSR+t and RDN+t) for fair comparisons.

We evaluate the SR performance on the B100 benchmark dataset by increasing the scaling factor from 1.1 to 4.

Interestingly, we observe that vdsr outperforms VDSR and VDSR+t at every scale by a large margin although VDSR and VDSR+t have similar network architecture to our vdsr. Similarly, RDN shows better performance than Meta-RDN and RDN+t. We also provide qualitative comparison results with Meta-SR in Figure 1, and we see that our RDN recovers much clearer edges than Meta-RDN.

4.4 Detailed Analysis

Interpolation and extrapolation. We experiment our method on various scale factors that are not shown during the training phase. In Figure 2, we plot PSNR values from vdsr and RDN by changing the scale factor on the B100 dataset. We see that our method learns an interpolation ability and can successfully deal with unseen scales between 1 and 4 (e.g., 1.15, 1.25, ... 3.95). Moreover, ours also learns an extrapolation ability and handles unseen scale factors larger than 4 (e.g., 4.1, 4.2, ... , 4.5). To sum up, our proposed SR process has a power of generalization (i.e., interpolation and extrapolation abilities), even the network is trained with only a limited number of scale factors.

SR performance with different ODE solvers. We experiment our method with different ODE solvers (e.g., Euler and RK4 methods). Note that Euler method is computationally cheaper than RK4, but RK4 provides more accurate approximation results generally. Similarly, in Table 2, we see that vdsr trained with RK4 shows slightly better SR performance than vdsr trained with Euler method on the B100 and Set5 datasets. This result suggests that, we can employ conventional ODE solvers to solve our own SR problem, but the quality of the predicted HR images are relying on the performance of the employed ODE solver.

Visual output of the network f. In Figure 3, to see the intermediate results by the network f during the gradual SR

	Euler Method			Runge-Kutta Method		
	x2	x3	x4	x2	x3	x4
B100	31.92	28.89	27.33	**31.96**	**28.94**	**27.38**
Set5	37.57	33.92	31.50	**37.58**	**34.03**	**31.68**

Table 2: Benchmark results of vdsr trained with Euler and Runge-Kutta methods on different scale factors. **Bold number** indicates better SR performance.

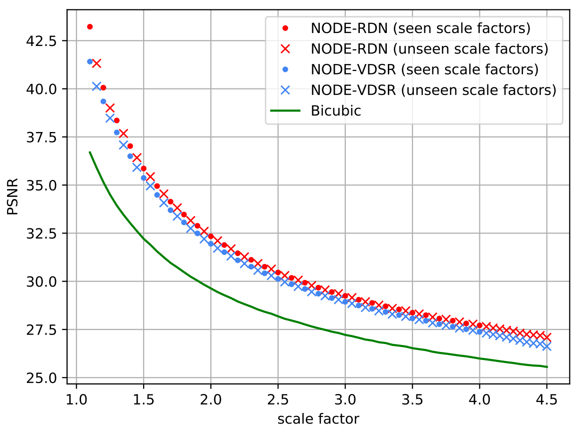

Figure 2: PSNR evaluations of bicubic upscaling, vdsr, and RDN on the B100 dataset by changing the scale factor from 1.1 to 4.5 with stride 0.05. Dotted marks correspond to seen scale factors during the training process (e.g., 1.1, 1.2, ..., 4.0) and cross marks correspond to unseen scale factors (e.g., 1.15, 1.25, ..., 4.5) during the training.

procedure at the test stage, we compute absolute value of $f(\hat{\mathcal{I}}(t), t, \theta)$ where t decreases from 4 to 1, and the initial condition is $\mathcal{I}(t_0 = 4)$.

Interestingly, on the sharp patch corresponding to the eye (red box), the absolute values are higher when t is small. While, on the homogeneous patch corresponding to the cheek (green box), the absolute values are higher when t is large. Recall that $\mathcal{I}(t)$ becomes close to the ground-truth image when t gets small, and the image difference $\frac{d\mathcal{I}(t)}{dt}(\approx f(\hat{\mathcal{I}}(t), t, \theta))$ includes more high-frequency components. Therefore, the absolute values of the network at the eye region which includes high-frequency detail becomes large when t is small, while the absolute values at the homogeneous region which does not require high-frequency detail becomes small when t is small.

5 Conclusion

In this work, we proposed a novel differential equation for the SR task to gradually enhance a given input LR image, and allow continuous-valued scale factor. Image difference between images over different scale factors is physically modeled with a neural network, and formulated as a NODE. To restore a high-quality image, we solve the ODE initial value problem with the initial condition given as an input LR image. The main difference with existing gradual SR methods is that our formulation is based on the physical modeling of

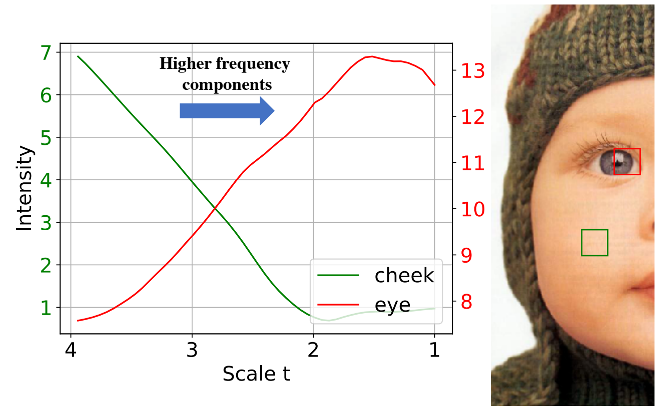

Figure 3: Intensity of intermediate image derivatives by changing scale factor t at two different locations.

the intermediate images, and adds fine high-frequency details gradually. The analysis on the intermediate states during the SR process gives us more insight on the gradual SR reconstruction. Detailed experimental results show that our method achieves superior performance compared to state-of-the-art SR approaches.

References

[Agustsson and Timofte, 2017] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. 2017.

[Cai *et al.*, 2019] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[Chen and Pock, 2016] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE transactions on pattern analysis and machine intelligence*, 2016.

[Chen *et al.*, 2018] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, 2018.

[Dong *et al.*, 2015] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2), 2015.

[Haris *et al.*, 2018] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[He *et al.*, 2019] Xiangyu He, Zitao Mo, Peisong Wang, Yang Liu, Mingyuan Yang, and Jian Cheng. Ode-inspired network design for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[Hu *et al.*, 2019] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[Huang *et al.*, 2015] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[Kim *et al.*, 2016a] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[Kim *et al.*, 2016b] Jiwon Kim, Jung Kwon Lee, and Kyoun Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[Lai *et al.*, 2017] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[Li *et al.*, 2019] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[Lim *et al.*, 2017] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017.

[Martin *et al.*, 2001] David Martin, Charless 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 *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2001.

[Rudin *et al.*, 1992] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 1992.

[Scao, 2020] Teven Le Scao. Neural differential equations for single image super-resolution. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.

[Tai *et al.*, 2017] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[Zeyde *et al.*, 2010] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*. Springer, 2010.

[Zhang *et al.*, 2018] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.