manuscript No.
(will be inserted by the editor)

Adaptive Importance Learning for Improving Lightweight Image Super-resolution Network

Lei Zhang[1,2] · Peng Wang[1] · Chunhua Shen[1*] · Lingqiao Liu[1] · Wei Wei[2] · Yanning Zhang[2] · Anton van den Hengel[1]

Received: date / Accepted: date

Abstract Deep neural networks have achieved remarkable success in single image super-resolution (SISR). The computing and memory requirements of these methods have hindered their application to broad classes of real devices with limited computing power, however. One approach to this problem has been lightweight network architectures that balance the super-resolution performance and the computation burden. In this study, we revisit this problem from an orthogonal view, and propose a novel learning strategy to maximize the pixel-wise fitting capacity of a given lightweight network architecture. Considering that the initial capacity of the lightweight network is very limited, we present an adaptive importance learning scheme for SISR that trains the network with an easy-to-complex paradigm by dynamically updating the importance of image pixels on the basis of the training loss. Specifically, we formulate the network training and the importance learning into a joint optimization problem. With a carefully designed importance penalty function, the importance of individual pixels can be gradually increased through solving a convex optimization problem. The training process thus begins with pixels that are easy to reconstruct, and gradually proceeds to more complex pixels as fitting improves. Furthermore, the proposed learning scheme is able to seamlessly assimilate knowledge from a more powerful teacher network in the form of importance initialization, thus obtaining better initial capacity in the network. Through learning the network parameters, and updating pixel importance, the proposed learning scheme enables smaller, lightweight, networks to achieve better performance than has previously been possible. Extensive experiments on four benchmark datasets demonstrate the potential benefits of the proposed learning strategy in lightweight SISR network enhancement. In some cases, our learned network with only 25% of the parameters and computational complexity can produce comparable or even better results than the corresponding full-parameter network.

Keywords Important learning, single image super-resolution, lightweight network enhancement

1 Introduction

There are a wide variety of applications where the ability to increase the resolution of an image adds to the user experience, from from surveillance and public security Zhang et al (2017a), business and entertainment Liu et al (2017) to remote sensing Wei et al (2017). Single-image super resolution (SISR), the process of increasing the resolution of an image without additional information, has received significant attention (Huang et al, 2015; Kim et al, 2016a; Yang et al, 2014) as a result.

Most early SISR methods focus on exploiting pixel statistics (Efrat et al, 2013; Kim and Kwon, 2010) or the internal patch recurrence (Glasner et al, 2009; Huang et al, 2015) of HR images as priors. These methods typically do not generalise well, because even a small divergence between the properties of the real low-resolution image and the prior embodied in the heuristic causes visible artifacts in the reconstructed HR image. Recently, deep convolution neural network (DCNN) based learning methods (Kim et al, 2016a,b; Ledig et al, 2017; Tai et al, 2017; Wang et al, 2015), have shown remarkable success in SISR, especially on some specific scaling factors (e.g., 2-4). Nevertheless, due to their very deep structures, these methods often exhibit significant memory and computing requirements, which necessitates powerful computational units (e.g., GPUs) thus limiting their application to the many real devices with lim-

* Corresponding author.
[1] School of Computer Science, The University of Adelaide, Australia
[2] School of Computer Science, Northwestern Polytechnical University, Xian, China

ited computing power (and particularly hand-held devices including phones).

To address this problem, some efforts (Dong et al, 2016b; Shi et al, 2016) dedicate to customize specific lightweight network architectures. In this study, we revisit this problem in an orthogonal view and propose to develop an novel learning strategy to maximize the pixel-wise fitting capacity of a given lightweight architecture. To this end, we revisit the traditional training procedure for a SISR network, which seeks the optimal network parameters to minimize the average loss over all pixels in training images. Moreover, pixels of diffident reconstruction difficulty are mixed together to fed into the network for training. However, by doing this, complex pixels that are difficult to reconstruct will mislead the training procedure, which renders the network even failing to handle pixels that are easy to reconstruct, since the initial capacity of the lightweight network is very limited and vulnerable. This is similar to the cognitive process of human which is prone to be confused when starts with a compound of complex and easy tasks and considers them equally. For example, when receiving a compound of easy and hard words one time, a pupil may fail to remember those easy ones that should be well mastered. Alternatively, if he starts with some easy words and gradually attempts to remember more and more hard ones when these easy words have been well mastered, more words will be remembered. Therefore, the basic pattern of human cognitive process is to learn from easy to complex and gradually enhance the capacity of human. Recently, it has been empirically demonstrated that learning as such a paradigm can avoid bad local minima and generalize better (Basu and Christensen, 2013; Khan et al, 2011). Therefore, it is promising to enhance the capacity of the lightweight SISR network with an appropriate easy-to-complex learning paradigm.

Inspired by this, we present an adaptive importance learning scheme for SISR, which assigns importance (i.e., the probability of participating training and zero importance denotes removing the pixel during training) to each image pixel and dynamically updates the importance to control the network training following an easy-to-complex paradigm. To this end, we formulate the network training as well as the pixel-wise importance learning into a bi-convex optimization problem. With introducing a carefully designed importance penalty function, the importance of image pixels can be adaptively updated by solving a convex optimization problem. As a result, the importance is gradually increased according to the network reconstruction error on these pixels. By doing this, the network will start with pixels that are easy to reconstruct for training, and gradually be exposed to more and more complex pixels when its fitting capacity is enhanced. Furthermore, with the proposed importance learning scheme, the network can seamlessly assimilate the knowledge from a more powerful teacher network in the form of pixel importance initialization, which enables the network to generalize better. Through learning the network parameters and updating the pixel importance in an alternative way until convergence, the proposed learning scheme can obviously enhance the network capacity. With extensive experiments on four benchmark datasets and two seminal DCNN architectures for SISR, we demonstrate that the proposed adaptive importance learning scheme is able to enhance the performance of different scales of lightweight networks obviously. Moreover, due to not designing specific lightweight network architecture, it can be conveniently applied to any lightweight SISR networks for enhancement.

In summary, this study mainly contributes in the following four aspects.

- We propose to develop an easy-to-complex learning paradigm to maximize the fitting capacity of a given lightweight network architecture for SISR. To the best of our knowledge, this is the first attempt to do this in SISR.
- We present an adaptive importance learning scheme to train the lightweight SISR network for enhancement.
- We propose to distil knowledge from a more powerful teacher network for better importance initialization.
- We demonstrate the pleasing potential of the proposed learning scheme in extensive experiments.

2 Related work

In this section, we briefly review the following three aspects of works related to this study.

Single image super-resolution. In early stage, SISR are addressed by exploiting the statistical characteristics of HR image as priors. For example, Sun et al. in (Sun et al, 2008) learn a gradient profile prior from extensive natural images and then apply it for SISR. In (Kim and Kwon, 2010), Kim et al. employ a modification of the natural image prior to refine the detailed structure along edges. Different from these methods, Glasner et al. (Glasner et al, 2009) propose to exploit the internal patch recurrence for super-resolution. Huang et al. (Huang et al, 2015) further introduce the geometric variation in searching recurrent patches. Recently, inspired by the success of deep neural networks, especially DCNN, some literatures commence at learning more powerful SISR models with DCNN from extensive LR-HR paris. For example, Dong et al. (Dong et al, 2016a) construct a 3-layer DCNN for SISR which outperforms most of previous non-learning methods. With introducing residual learning, Kim et al. (Kim et al, 2016a) develop a much deeper (e.g., 20 layers) DCNN based SISR model. Tai et al. (Tai et al, 2017) further introduce a recursive block into the global residual structure and gains the state-of-the-art performance. In (Ledig et al, 2017), Ledig et al. present a generative adversarial network to obtain photo-realistic HR images. Although those

deep models achieve satisfactory SISR results, most of them are computational expensive to deploy on real devices. Currently, a few literatures have commenced at handling this problem by developing lightweight network architecture. For example, a compact hourglass-shape DCNN structure and a subpixel convolution structure are designed in (Dong et al, 2016b; Shi et al, 2016), respectively. In this study, we solve this problem in an orthogonal view and propose to maximize the capacity of a given lightweight network with a new learning strategy. In addition, due to not involving network architecture, the proposed scheme can be directly integrated into any lightweight SISR networks for enhancement.

Knowledge distillation. This line of research aims at distilling knowledge from a complicated (or an ensemble of models) teacher model into a compact (or single) alternative without performance drop. Hinton et al. (Hinton et al, 2015) propose to distil knowledge by matching the soften output (e.g., logits) of teacher models. Romero et al. (Romero et al, 2014) further match the intermediate features (e.g., hints) of teacher models. Zhang et al. (Zhang et al, 2017b) integrate the knowledge distillation into a mutual learning framework. Different from matching the output of teacher models, we propose to learn the pixel-wise importance of each example to training loss from a teacher model.

Curriculum and self-paced learning. Similar as this study, these two paradigms learn a model gradually including from easy to complex examples in training phase. In curriculum learning (Bengio et al, 2009), the curriculum (i.e., learning sequence) is often derived by predetermined heuristics. For example, in (Bengio et al, 2009), the curriculum is derived based on the variability in shape to enable shapes with less variability being learned earlier. In (Khan et al, 2011), the common sense of participants are employed to determine the learning sequence of graspability to object. In self-paced learning, the curriculum design is often integrated into the learning objective as a regularization. For example, Jiang et al. (Jiang et al, 2014) jointly optimize the learning objective as well as a binary weight vector which controls the learning pace. In contrast, the proposed adaptive importance learning scheme learns a pixel-wise curriculum based on the reconstruction error of the network and aims at enhancing the capacity of a given lightweight SISR network. Moreover, it enables the network to seamlessly assimilate the knowledge from a more powerful teacher network.

3 The proposed learning paradigm

In general, with n LR-HR image pairs $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$, we can learn a lightweight network $\mathcal{S}(\cdot, \theta)$ as follows

$$\min_{\theta} \mathbb{E}(\theta) = \frac{1}{n}\sum_{i=1}^{n} l\left(\mathbf{y}_i, \mathcal{S}(\mathbf{x}_i, \theta)\right) \tag{1}$$

where θ denotes the network parameters and l indicates the loss function (e.g., MSE loss or ℓ_1 loss). In the training phase, the optimal θ seeks to minimize the expectation $\mathbb{E}(\theta)$ where all pixel with different reconstruction difficulties are fed together into $\mathcal{S}$ for training. To maximize the pixel-wise fitting capacity of $\mathcal{S}$, we propose to train $\mathcal{S}$ with an adaptive importance learning scheme as

$$\begin{aligned} &\min_{\theta, W} \mathbb{E}(\theta, W) = \frac{1}{n}\sum_{i=1}^{n} \left[l\left(\mathbf{y}_i \odot \mathbf{w}_i, \mathcal{S}(\mathbf{x}_i, \theta) \odot \mathbf{w}_i\right) + h(\mathbf{w}_i)\right], \\ &\text{s.t. } \forall i, 0 \preceq \mathbf{w}_i \preceq 1, \end{aligned} \tag{2}$$

where $\mathbf{w}_i$ indicates the pixel-wise importance vector for each training pair and $W = \{\mathbf{w}_i\}_{i=1}^n$ collects all importance vectors. Since $0 \preceq \mathbf{w}_i \preceq 1$, the pixel-wise importance can be viewed as the probability of each pixel participating the training procedure as Eq. (2), e.g., when the importance is zero, the corresponding pixel will removed from training the network. $\odot$ denotes point-wise multiplication. $h(\mathbf{w}_i)$ represents a penalty function over $\mathbf{w}_i$, which controls the importance learning strategy as well as avoiding trivial solutions of $\mathbf{w}_i$ (e.g., $\mathbf{w}_i = \mathbf{0}$).

In the adaptive importance learning scheme, the network parameter θ and the importance W are jointly optimized. To solve this problem, we can adopt the alternative minimization scheme (Zhang et al, 2018), which reduces this problem into a θ-subproblem and a W-subproblem, and then alternatively optimizes each subproblem until convergence. Different from the traditional learning scheme in Eq. (1) which only trains the network once, the proposed learning scheme will train the network in several rounds. More importantly, with an appropriate $h(\mathbf{w}_i)$, the importance of image pixels can be assigned to any value expected, with which a specific group of pixels can be picked out from all training examples to optimize for the network parameter θ in the next iteration. Through optimizing the network parameter θ and dynamically updating the importance in an alternative way, the proposed learning scheme is able to train the network $\mathcal{S}$ with a specific learning paradigms. In addition, when $h(\mathbf{w}_i)$ is given as the following indicator function,

$$h(\mathbf{w}_i) = \mathcal{I}(\mathbf{w}_i, \mathbf{1}) = \begin{cases} \infty, & \mathbf{w}_i \neq \mathbf{1} \\ 0, & \mathbf{w}_i = \mathbf{1} \end{cases}, \tag{3}$$

the proposed learning scheme will degenerate to the traditional learning scheme in Eq. (1). Therefore, the proposed adaptive importance learning scheme is a general learning framework for SISR.

In this study, we employ the proposed learning scheme in Eq. (2) with a carefully designed $h(\mathbf{w}_i)$ to train a given lightweight SISR network $\mathcal{S}$ with an easy-to-complex paradigm for capacity enhancement. To this end, the importance produced by the designed $h(\mathbf{w}_i)$ are required to conform with

the following requirements. At beginning, the importance of complex pixels that are difficult to reconstruct will be suppressed (i.e., assigned to a small value close to zero) while the importance of pixels that are easy to reconstruct will be highlighted (i.e., assigned to a large value close to one). By doing this, $\mathcal{S}$ is encouraged to focus on learning to reconstruct easy pixels when its initial capacity is limited. Given the learned $\mathcal{S}$, importance W will be gradually increased to expose $\mathcal{S}$ to more complex pixels for the next round of training, and thus the capacity of $\mathcal{S}$ will be enhanced. When the alternative minimization converges, the capacity of $\mathcal{S}$ can be maximized. In the following, we will introduce a carefully designed h to update the importance W as expected.

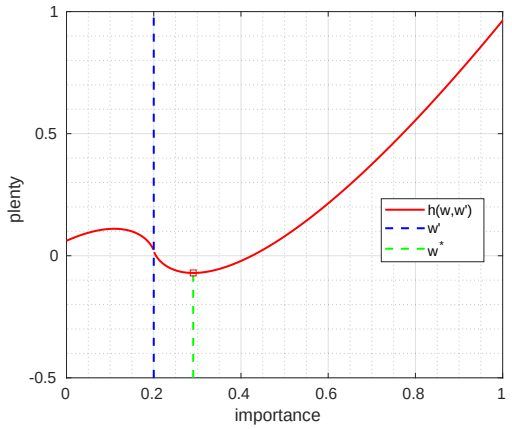


Fig. 1 The designed importance penalty function $h(\mathbf{w}, \mathbf{w}')$ (e.g., $d = 0.1$, $\lambda = 0.1$).

3.1 Adaptive importance learning

According to the discussion above, we find that a basic principle for importance updating is to gradually increase the importance to feed $\mathcal{S}$ with more complex pixels in the next round of training. Moreover, the increment to importance should be determined by a decreasing function over the reconstruction difficulty of image pixels to guarantee the easy-to-complex learning paradigm. However, it is difficult to determine the reconstruction difficulty of pixels given an image. Intuitively, pixels lying on image details or within complex structures often are more difficult to reconstruct than those on flat areas. To quantitatively indicate the reconstruction difficult, we adopt the reconstruction error of the learned network $\mathcal{S}$ on pixels as a rough measure. This is inspired by the observation that most SISR methods can better reconstruct pixels on flat areas than those on image details. In addition, the reconstruction error of network $\mathcal{S}$ on all pixels can be directly indicated by the loss l in Eq. (2). Thus, the key for importance learning is to design an appropriate importance penalty function $h(\mathbf{w}_i)$.

To comply with the importance learning principle mentioned above, we carefully design a penalty function h and reformulate the learning scheme in Eq. (2) as follows

$$\begin{aligned} &\min_{\theta,W} \mathbb{E}(\theta, W) = \frac{1}{n}\sum_{i=1}^{n}\left[l(\mathbf{y}_i \odot \mathbf{w}_i, \mathcal{S}(\mathbf{x}_i, \theta) \odot \mathbf{w}_i) + h(\mathbf{w}_i, \mathbf{w}_i')\right], \\ &\text{s.t. } \forall i, \mathbf{w}_i' \le \mathbf{w}_i \le 1, \end{aligned} \tag{4}$$

where $\mathbf{w}_i'$ denotes the importance vector in previous iteration and $h(\mathbf{w}_i, \mathbf{w}_i')$ is given as

$$h(\mathbf{w}_i, \mathbf{w}_i') = \sum_j \left(w_{ji} - w_{ji}'\right)\left(\ln\frac{w_{ji} - w_{ji}'}{\lambda} - 1\right). \tag{5}$$

In Eq. (5), w_{ji} and w_{ji}' denote the j-th element in $\mathbf{w}_i$ and $\mathbf{w}_i'$, respectively. λ is a predefined positive scalar. In the following, we will discuss the benefits of $h(\mathbf{w}_i, \mathbf{w}_i')$ in details.

Similar as solving Eq. (2), we adopt the alternative minimizing scheme to alternatively optimize θ and W in Eq. (5). Specifically, when the importance vectors W are given, the learning problem for θ can be well addressed by the back-propagation algorithm. When θ is fixed, the learning problem for W can be simplified as

$$\begin{aligned} &\min_{w} dw + (w - w')\left(\ln\frac{w - w'}{\lambda} - 1\right), \\ &\text{s.t. } w' \le w \le 1, \end{aligned} \tag{6}$$

where w denotes the importance of a specific pixel in training samples (e.g., an element from $\mathbf{w}_i$) and w' denotes the corresponding importance value in previous iteration (e.g., the corresponding element from $\mathbf{w}'$). d denotes the reconstruction loss of the learned network $\mathcal{S}$ on the considered pixel. To solve the problem in Eq. (6), we introduce the following result.

Theorem 1 *Considering the constraint $w' \le w \le 1$, function $f(w) = dw + (w - w')\left(\ln\frac{w-w'}{\lambda} - 1\right)$ is a convex function and $f(w^*)$ reaches the minima when*

$$w^* = w' + \lambda \cdot e^{-d}. \tag{7}$$

Proof Given $f(w)$ and the constraint $w' \le w \le 1$, we have $\frac{\partial^2 f(w)}{\partial w^2} = \frac{\partial}{\partial w}\left(d + \ln\frac{w-w'}{\lambda}\right) = \frac{\lambda}{w-w'} > 0$. Thus, with the constraint $w' \le w \le 1$, $f(w)$ is a convex function, and the minima is reached when $\frac{\partial f(w)}{\partial w}|_{w=w^*} = 0$. We have

$$d + \ln\frac{w^* - w'}{\lambda} = 0 \Rightarrow w^* = w' + \lambda \cdot e^{-d}$$

.

To further illustrate this point, a visual example can be found in Figure 1. □

(a) Importance function (b) Image (c) Prediction error (d) Importance map

Fig. 2 Importance initialization from a teacher network (e.g., VDSR (Kim et al, 2016a)). (a) Importance function in Eq. (8) with different parameters. (b) Example image. (c) Prediction error (e.g., ℓ_2 norm) from the teacher network. (d) Importance map for the example image with parameter $\mu_0 = 0.01, \alpha_0 = 100$.

According to Theorem 1, the problem in Eq. (6) has a closed-form solution as Eq. (7). In Eq. (7), the importance w^* is updated by adding an increment to importance value w' in the previous iteration. Since $\lambda \cdot e^{-d} \geq 0$, such a update rule enables to gradually increase the importance in each iteration. Moreover, the increment is determined by an decreasing function over the reconstruction loss of the pre-learned model $\mathcal{S}$ on the corresponding pixel, viz., a small increment is given when the reconstruction loss is large. Both aspects of principle for importance learning mentioned at the beginning of this subsection are satisfied. Therefore, the learning scheme in Eq. (4) with the penalty function $h(\mathbf{w}, \mathbf{w}')$ is able to feed more and more complex pixels into $\mathcal{S}$ for training with an easy-to-complex paradigm through adaptively updating the importance vector as Eq. (7). Furthermore, the proposed learning scheme enables the network to seamlessly assimilate the knowledge from a more powerful teacher network in the form of pixel importance initialization. This will be introduced in details in the following subsection.

3.2 Importance initialization from the teacher

In Eq. (4), the proposed adaptive importance learning scheme depends on the the importance vectors w' in previous iteration. This brings an intuitive problem in initializing the importance at beginning. According to the discussion at the beginning of Section 3, it is necessary to determine the importance of image pixels according to their reconstruction difficulty and complex pixels are expected to be assigned to smaller importance than that to easy pixels. Since $\mathcal{S}$ is unknown at beginning, it is infeasible to indicate the pixel importance according to the reconstruction error of $\mathcal{S}$ as Section 3.1. To address this problem, we propose to learn important W from a given more powerful teacher network $\mathcal{T}$. Similar as the learned $\mathcal{S}$, $\mathcal{T}$ will produce larger reconstruction error on complex pixels than those easy ones. Then, a decreasing function over the reconstruction error is employed to produce the importance. To well suppressing the complex pixel as well as highlight the easy ones at the beginning, we establish the following importance function

$$g(x) = \frac{z}{1 + e^{(x-\mu_0)\alpha_0}}, \tag{8}$$

where x denotes the reconstruction error (e.g, ℓ_2 norm) of the teacher network $\mathcal{T}$ on a specific pixel and $g(x)$ is the corresponding importance value. μ_0 and α_0 denote the bias and scale parameters in this function. $z = (1 + e^{-\mu_0\alpha_0})$ is a normalization factor which scales the importance into $[0, 1]$. To demonstrate the effectiveness of the importance function in Eq. (8), we plot the profiles of $g(x)$ with different parameters as well as the estimated importance map on an example image in Figure 2. It can be seen that $g(x)$ will produce a small importance when the reconstruction error is large, vice versa. On the example image, we can find that pixels lying on image details (i.e., exhibiting complex structures) are assigned to low importance, while pixels on flat areas are assigned to high importance. This complies with the intuition that pixels on image details are more difficult to reconstruct than those on flat areas.

Given the teacher network $\mathcal{T}$ and the importance function g, we can train the network $\mathcal{S}$ by solving the following problem

$$\begin{aligned} &\min_{\theta} \mathbb{E}(\theta; W) = \frac{1}{n}\sum_{i=1}^{n} l\left(\mathbf{y}_i \odot \mathbf{w}_i, \mathcal{S}(\mathbf{x}_i, \theta) \odot \mathbf{w}_i\right), \\ &\text{s.t. } \forall i, \mathbf{w}_i = g(\mathcal{T}(\mathbf{x}_i)), \end{aligned} \tag{9}$$

where, for a concise formulation, we employ $g(\mathcal{T}(\mathbf{x}_i))$ to denote applying g to the reconstruction error of $\mathcal{T}$ on each pixel in $\mathbf{x}_i$. In this learning scheme, the knowledge from the teacher network is distilled to guide training the network with the easy-complex paradigm.

Relation to focal loss The proposed learning scheme in Eq. (9) is similar to the focal loss based learning scheme (Lin et al, 2017). Both of them dynamically reweight samples

during the training procedure to enhance the capacity of network. However, they totally differ in the following three aspects. 1) With the proposed scheme, the learned model is forced to focus on easy cases, whereas focal loss encourages the network to focus on complex cases. 2) In Eq. (9), the weights to training examples are determined by the prediction error of the given teacher model, while focal loss determines those weights based on the training error of the learned model. 3) Focal loss is proposed for training a more robust classifier or detector, while the proposed scheme aims at learning a more powerful compact SISR model.

3.3 Algorithm

With the alternative minimizing scheme, the overall optimization procedure for the prosed adaptive importance learning scheme in Eq. (2) can be summarized into Algorithm 1. At the beginning, the network $\mathcal{S}$ is trained with the importance vectors W initialized by the given teacher network $\mathcal{T}$ as Eq. (9). Then, the learning scheme in Eq. (4) is carried out in T iterations to gradually enhance the capacity of $\mathcal{S}$.

Algorithm 1: Adaptive importance learning (AIL)

Input: Input HR-LR training pairs $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$, pre-trained teacher model $\mathcal{T}$, importance function g, penalty function h and λ.

1. Importance initialization from teacher:

(1) Learn importance W as Eq. (9);

(2) Update model parameter $\theta^* = \arg\min_\theta \mathbb{E}(\theta; W)$ as Eq. (4);

2. Adaptive importance learning:

For $t \leftarrow 1$ **to** T

(1) Update $W^* = \arg\min_W \mathbb{E}(\theta^*, W)$ as Eq. (7);

(2) Update $\theta^* = \arg\min_\theta \mathbb{E}(\theta, W^*)$ as Eq. (5);

End for

Output: θ-parameterized model $\mathcal{S}$.

It is noticeable that in theory Algorithm 1 can well converge. Specifically, according to Eq. (7), the importance vectors W are gradually increased with the proceeding of iterations. When all elements in W increase to 1, the importance W will be unchanged in the following iterations and Algorithm 1 will converge, since no novel information will be provided by the training examples. More experimental evidence will be provided in Section 5.4.

In addition, different from previous methods (Dong et al, 2016b; Shi et al, 2016) that design new lightweight network architectures to deploy deep SISR methods onto real devices, the proposed adaptive importance learning scheme only focuses on how to enhance the capacity of network with a new training paradigm, and thus it can be directly applied to any given lightweight SISR network architecture. Experimental evidence will provided in Section 5.

4 Customizing lightweight SISR model

Most of state-of-the-art SISR models (Dong et al, 2016a; Kim et al, 2016a; Ledig et al, 2017; Tai et al, 2017) are inspired by the DCNN framework where the basic modules are convolution layer. To obtain a lightweight network, previous literatures (Dong et al, 2016b; Shi et al, 2016) propose to design new architectures (e.g., introducing a hourglass-shape structure or a sub-pixel convolution structure), which, however, cannot be conveniently applied to other DCNNs for SISR, especially when different scales of lightweight networks are required to fit various real devices. In this study, given a teacher network, we customize the lightweight network by directly reducing filters in each convolution layer to reduce the amount of output feature maps by a fixed ratio (e.g., $0 < \rho < 1$). By doing this, we can obtain different scales of lightweight networks with different ρs. Given a fixed ρ, each convolution layer (i.e., except the input and output layer) in the obtained lightweight network reduces $1 - (1 - \rho)^2\%$ parameters as well as computational complexity, compared with that in the teacher network. The parameters and computational complexity of some lightweight networks are provided in Table 3.

It is noticeable that the comparison between different ways of customizing lightweight network architecture is beyond the scope of this study. Our aim of adopting the way of reducing filters is to make it convenient to verify the effectiveness of the proposed learning scheme in enhancing different scales of lightweight networks.

5 Experimental results and analysis

In this section, we conduct extensive experiments to demonstrate the effectiveness of the proposed learning scheme in enhancing a given lightweight SISR network architecture.

5.1 Dataset

Training datasets Current SISR methods often adopt different training datasets. For example, the very large ImageNet dataset is adopted by (Dong et al, 2016a), while literatures (Kim et al, 2016a; Tai et al, 2017) aggregate 91 images from (Yang et al, 2010) and another 200 images from the Berkeley Segmentation Dataset (Martin et al, 2001) together for training. In this study, we adopt the dataset utilized in (Kim et al, 2016a) with 291 images as benchmark to train all networks for fair comparison. In addition, rotation (e.g., with angle 90°, 180°, 270°), flip and downsampling (e.g., with ratio 0.5, 0.7, 1.0) are further employed for data augmentation.

Test datasets Similar as (Huang et al, 2015; Kim et al, 2016a; Tai et al, 2017), we adopt four benchmark datasets

for performance evaluation, namely Set5 (Bevilacqua et al, 2012), Set14 (Zeyde et al, 2010), BSD100 (Timofte et al, 2014) and Urban100 (Huang et al, 2015), which contain 5, 14, 100 and 100 indoor and outdoor natural images, respectively.

5.2 Teacher SISR networks

In this study, we adopt two seminal DCNN architectures for SISR to customize the lightweight network as well as initializing importance for Algorithm 1, including `VDSR` (Kim et al, 2016a) and `DRRN` (Tai et al, 2017). Currently, the network architectures of most state-of-the-art SISR methods (Kim et al, 2016b; Lai et al, 2017; Mao et al, 2016) are inspired by these two models. In VDSR, 20 fully convolution layers with global residual structure are employed to learn a deep mapping from a given LR input to an HR output. This is the first attempt to introduce the global residual structure into SISR, which enables a much deeper model than previous works (Dong et al, 2016a) and improves the SISR performance obviously. According to (Kim et al, 2016a), 64 feature maps are adopted for `VDSR` in this study. Recently, `DRRN` advances replacing the convolution layers in `VDSR` with a recursive block, which further improves the SISR performance as well as reducing the model parameters. As suggested in (Tai et al, 2017), the recursive number and amount of feature maps are set 25 and 128, respectively.

5.3 Training and testing setup

For network training, we follow the standard protocol utilized in (Kim et al, 2016a). Specifically, we implement these two teacher networks mentioned above as well as the corresponding lightweight networks based on the codes released online [1]. With introducing the mean squared error (MSE) loss as l into Eq. (1) and Eq. (2), we train each network in 50 epochs with batch size 128 in the Pytorch framework (Paszke et al, 2017). Learning rate is initially set as 0.1 and then decayed by a factor 10 every 10 epochs. Model parameters are learned by the SGD optimizer with momentum parameter 0.9, weight decay parameter $1e^{-4}$ and gradient clip parameter 0.4. In Algorithm 1, we set the pre-defined parameter $\lambda = 0.15$ and maximum iterations $T = 10$. For the importance function g, the parameter $\alpha_0 = 0.01$ and $\mu_0 = 100$ are fixed in the following experiments.

In testing phase, we employ each learned network to improve the resolution of a given LR image with three different scaling factors $2, 3, 4$. To quantitatively evaluate the performance of each network, we adopt two standard criteria, namely peak signal-to-noise ratio (PSNR) and structured similarity (SSIM) to measure their super-resolution results.

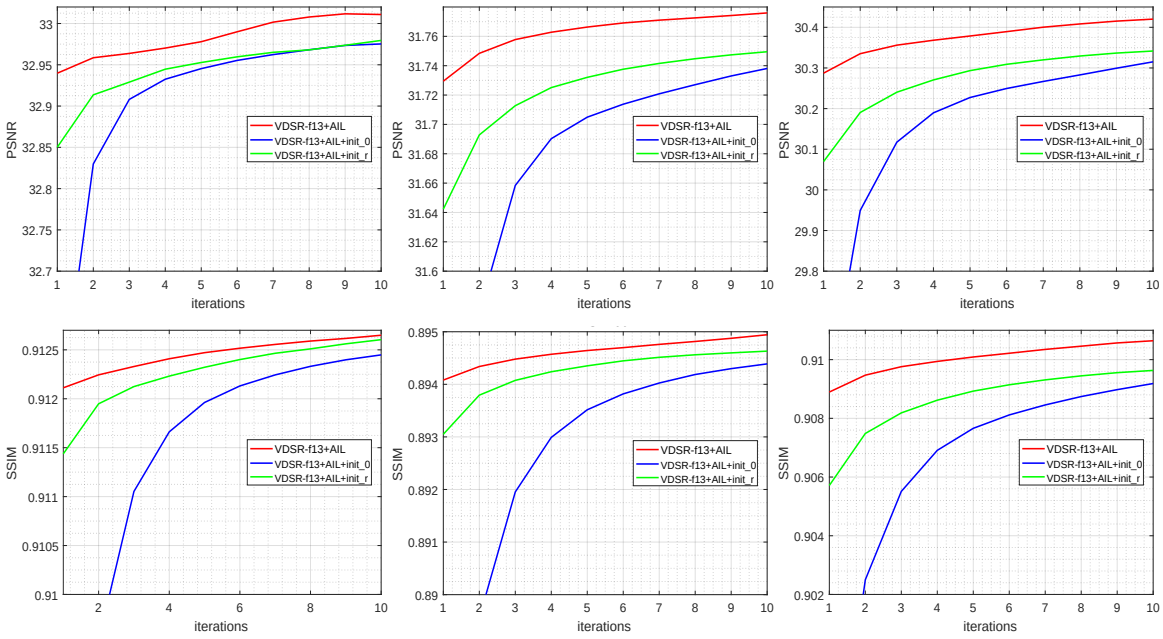
[1] VDSR: https://github.com/twtygqyy/pytorch-vdsr
DRRN: https://github.com/jt827859032/DRRN-pytorch

5.4 Ablation study

In this part, we mainly focus on demonstrating the effect of the proposed adaptive importance learning scheme and the importance initialization scheme, the difference between the proposed learning scheme and the knowledge distillation and the convergence of Algorithm 1. To this end, we adopt `VDSR` as the teacher network $\mathcal{T}$ and obtain the corresponding lightweight network by reducing the amount of feature maps in each convolution layer with a fixed ratio $\rho = 0.8$ as Section 4. Concretely, the amount of feature maps in each convolution layer of the lightweight network is reduced from 64 to 13, viz., the parameters and the computational complexity is only 4% of that in `VDSR`, shown as Table 3.

5.4.1 Effect of adaptive importance learning

We propose the adaptive importance learning scheme to train a given lightweight network with an easy-to-complex principle as well as gradually enhance the network generalization capacity. To demonstrate this point, we train the given lightweight network above with Algorithm 1 and evaluate it on three test datasets (e.g., Set14, BSD100 and Urban100). For simplicity, we term the obtained network `VDSR-f13+AIL` where `-f13` denotes the amount of feature maps in each convolution layer of the given lightweight network. The performance (e.g., PSNR and SSIM) curves of `VDSR-f13+AIL` within $T = 10$ iterations are depicted in Figure 3. It can be seen that on each dataset both the PSNR and SSIM measures of `VDSR-f13+AIL` are gradually increased with the proceeding of iterations. To further clarify this point, we implement two variants of `VDSR-f13+AIL` by training the same lightweight network with Algorithm 1 but initializing the importance W as zeros and random values, respectively. For simplicity, we term these two variants `VDSR-f13+AIL+init_0` and `VDSR-f13+AIL+init_r`. The corresponding performance curves for these two variants are also provided in Figure 3. We can find that the adaptive importance learning scheme always gradually enhance the super-resolution performance with the proceeding of iterations, which is robust to the initialization of importance. This is because that in Eq. (7) the importance is gradually increased based on its previous value, which enables to expose the network with more and more complex pixels. In addition, the final numerical results of these three methods on four test datasets are reported in Table 1. To illustrate their superiority, we also implement a baseline method, `VDSR-f13`, which is obtained by training the given lightweight network with the traditional learning scheme in Eq. (1). It can be seen that



(a) Set14 (b) BSD100 (c) Urban100

Fig. 3 The performance (e.g., PSNR and SSIM) curves of `VDSR-f13+AIL`, `VDSR-f13+AIL+init_0` and `VDSR-f13+AIL+init_r` within 10 iterations on three test datasets.

Table 1 Average PSNR/SSIM of `VDSR-f13`, `VDSR-f13+ILT` and `VDSR-f13+AIL` on four test datasets. The best results are in bold. ↑PSNR/SSIM and ↓PSNR/SSIM denote the performance increase and decrease over `VDSR-f13`, respectively.

Dataset	scale	VDSR-f13	VDSR-f13+AIL+init_0		VDSR-f13+AIL+init_r		VDSR-f13+AIL	
Set5	×2	37.18/0.9580	37.39/0.9589	↑0.21/↑0.0009	37.41/0.9590	↑0.23/↑0.0010	**37.43/0.9591**	↑0.25/↑0.0011
	×3	33.07/0.9155	33.26/0.9184	↑0.19/↑0.0029	33.27/0.9183	↑0.21/↑0.0028	**33.36/0.9196**	↑0.29/↑0.0041
	×4	30.74/0.8724	30.87/0.8764	↑0.13/↑0.0040	30.90/0.8769	↑0.16/↑0.0045	**30.99/0.8788**	↑0.25/↑0.0065
Set14	×2	32.84/0.9115	32.98/0.9124	↑0.13/↑0.0009	32.98/0.9126	↑0.14/↑0.0011	**33.01/0.9126**	↑0.17/↑0.0011
	×3	29.53/0.8269	29.65/0.8295	↑0.11/↑0.0025	29.68/0.8300	↑0.15/↑0.0030	**29.73/0.8309**	↑0.20/↑0.0040
	×4	27.75/0.7600	27.84/0.7633	↑0.10/↑0.0033	27.87/0.7636	↑0.12/↑0.0036	**27.93/0.7654**	↑0.18/↑0.0054
BSD100	×2	31.63/0.8930	31.74/0.8944	↑0.11/↑0.0014	31.75/0.8946	↑0.12/↑0.0017	**31.78/0.8949**	↑0.15/↑0.0020
	×3	28.54/0.7906	28.62/0.7932	↑0.08/↑0.0026	28.65/0.7935	↑0.11/↑0.0030	**28.69/0.7946**	↑0.14/↑0.0041
	×4	27.03/0.7169	27.08/0.7193	↑0.05/↑0.0024	27.11/0.7199	↑0.08/↑0.0030	**27.14/0.7212**	↑0.11/↑0.0043
Urban100	×2	30.06/0.9056	30.32/0.9092	↑0.25/↑0.0036	30.34/0.9096	↑0.28/↑0.0041	**30.42/0.9106**	↑0.36/↑0.0051
	×3	26.42/0.8081	26.60/0.8146	↑0.18/↑0.0065	26.66/0.8158	↑0.24/↑0.0077	**26.76/0.8189**	↑0.34/↑0.0108
	×4	24.64/0.7312	24.74/0.7369	↑0.10/↑0.0057	24.79/0.7379	↑0.15/↑0.0067	**24.87/0.7416**	↑0.22/↑0.0105

`VDSR-f13+AIL` and the other two variants obviously outperforms `VDSR-f13` in all cases. This demonstrates that the proposed easy-to-complex learning strategy can better exploit the super-resolution capacity of the given lightweight network than the traditional learning scheme in Eq. (1).

In summary, we can conclude that the proposed adaptive importance learning scheme is able to gradually enhance the capacity of the given lightweight network and ultimately obviously improve the super-resolution performance, which, furthermore, is robust to the importance initialization.

5.4.2 Effect of importance initialization from teacher

In the proposed adaptive importance learning scheme as Algorithm 1, we initialize the importance W by distilling knowledge from a given teacher network as Eq. (9). It is noticeable that this is not the unique way for importance initialization. As mentioned in Section 5.4.1, the importance can be simply initialized as zeros or random values. To illustrate the effectiveness of the proposed importance initialization scheme, we compare `VSDR-f13+AIL` with its two variants, namely `VDSR-f13+AIL+init_0` and `VDSR-f13+AIL+init_r`. Their performance curves and the numerical comparison re-

Table 2 Average PSNR/SSIM of VDSR-f13, VDSR-f13+Distil and VDSR-f13+AIL on four test datasets. The best results are in bold. ↑PSNR/SSIM and ↓PSNR/SSIM denote the performance increase and decrease over VDSR-f13, respectively.

Dataset	scale	VDSR-f13	VDSR-f13+Distil		VDSR-f13+AIL	
Set5	×2	37.18/0.9580	37.21/0.9581	↑0.03/↑0.0001	**37.43/0.9591**	↑0.25/↑0.0011
	×3	33.07/0.9155	33.09/0.9158	↑0.02/↑0.0003	**33.36/0.9196**	↑0.29/↑0.0041
	×4	30.74/0.8724	30.68/0.8705	↓0.06/↓0.0019	**30.99/0.8788**	↑0.25/↑0.0065
Set14	×2	32.84/0.9115	32.88/0.9115	↑0.04/↑0.0000	**33.01/0.9126**	↑0.17/↑0.0011
	×3	29.53/0.8269	29.55/0.8272	↑0.01/↑0.0003	**29.73/0.8309**	↑0.20/↑0.0040
	×4	27.75/0.7600	27.71/0.7587	↓0.03/↓0.0013	**27.93/0.7654**	↑0.18/↑0.0054
BSD100	×2	31.63/0.8930	31.65/0.8932	↑0.02/↑0.0002	**31.78/0.8949**	↑0.15/↑0.0020
	×3	28.54/0.7906	28.56/0.7911	↑0.02/↑0.0005	**28.69/0.7946**	↑0.14/↑0.0041
	×4	27.03/0.7169	27.01/0.7160	↓0.02/↓0.0009	**27.14/0.7212**	↑0.11/↑0.0043
Urban100	×2	30.06/0.9056	30.07/0.9058	↑0.01/↑0.0003	**30.42/0.9106**	↑0.36/↑0.0051
	×3	26.42/0.8081	26.47/0.8097	↑0.04/↑0.0015	**26.76/0.8189**	↑0.34/↑0.0108
	×4	24.64/0.7312	24.59/0.7290	↓0.05/↓0.0022	**24.87/0.7416**	↑0.22/↑0.0105

sults can be found in Figure 3 and Table 1. As shown in Figure 3, importance initialization from a teacher network in VDSR-f13+AIL leads to much better initial capacity of network in the first iteration than that from both the zero and the random importance initialization in other two variants. For example, on Urban100 dataset, the superiority of VDSR-f13+AIL over other two variants is up to 0.2db. Moreover, with the proceeding of iterations, VDSR-f13+AIL obviously outperforms the other two variants in all cases and VDSR-f13+AIL+init_r often surpasses VDSR-f13+AIL+init_0. Similar results also occur on the numerical results of these three methods, shown as Table 1. The reason for their performance difference comes from the following two aspects. On one hand, when the importance W is initialized as zeros, no examples will be chosen to train the network in the *Importance initialization from teacher* step of Algorithm 1, and the network with randomly initialized weights will be directly fed into the *Adaptive importance learning* step to update the importance based on its reconstruction error. Thus, the resulted importance will render the learning scheme deviating from starting with easy pixels and the following training procedures are prone to be trapped into a bad local minima. On the other hand, when W is randomly initialized, the learning scheme is also prone to deviate from the principle of starting with easy pixels. In contrast to the case with zero-initialized W, randomly initialized W enables to train the network in the *Importance learning from teacher* step of Algorithm 1 with some selected pixels, which leads to better initial network capacity as well as the final results, shown as the results of VDSR-f13+AIL+init_r and VDSR-f13+AIL+init_0 in Figure 3 and Table 1. In this study, the proposed importance initialization from teacher enables the network to start with easy pixels, thus producing the best performance. Therefore, we can conclude that importance initialization from teacher can benefit providing better initial capacity of network as well as the ultimate super-resolution performance.

5.4.3 Comparison with knowledge distillation

The proposed adaptive importance learning scheme initializes the importance from a given teacher network, which is similar to the prevailing knowledge distillation scheme (Hinton et al, 2015). Both of them distil specific knowledge from a given teacher model to train the student model for better generalization capacity. The difference is that the knowledge distillation scheme forces the student network to mimic the soften output of the given teacher network, whereas the proposed scheme distils the importance from the teacher network to guide the lightweight network focusing on handling easy pixels at beginning. To further clarify their difference, we implement a variant of VDSR-f13+AIL by training the same lightweight network with the knowledge distillation scheme (Hinton et al, 2015) as

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \left[l\left(\mathbf{y}_i, \mathcal{S}(\mathbf{x}_i, \theta)\right) + \beta \cdot l\left(\mathcal{T}(\mathbf{x}_i), \mathcal{S}(\mathbf{x}_i, \theta)\right) \right] \tag{10}$$

where β is set as 0.1 for the best performance. The numerical results of this variant (i.e., termed VDSR-f13+Distil), VDSR-f13+AIL and VDSR-f13 on four test datasets are provided in Table 2. It can be found that VDSR-f13+Distil only gives comparable results to that of the baseline VDSR-f13 and is far inferior to VDSR-f13+AIL. To further clarify this point, we depict some visual results of these three networks in Figure 4. We can find that compared with VDSR-f13 and VDSR-f13+Distil, VDSR-f13+AIL recovers more image details and the produced results are even close to that of VDSR with full parameters. The reason is intuitive. In (Hinton et al, 2015), knowledge distillation scheme is utilized in the classification problem where the soften output of

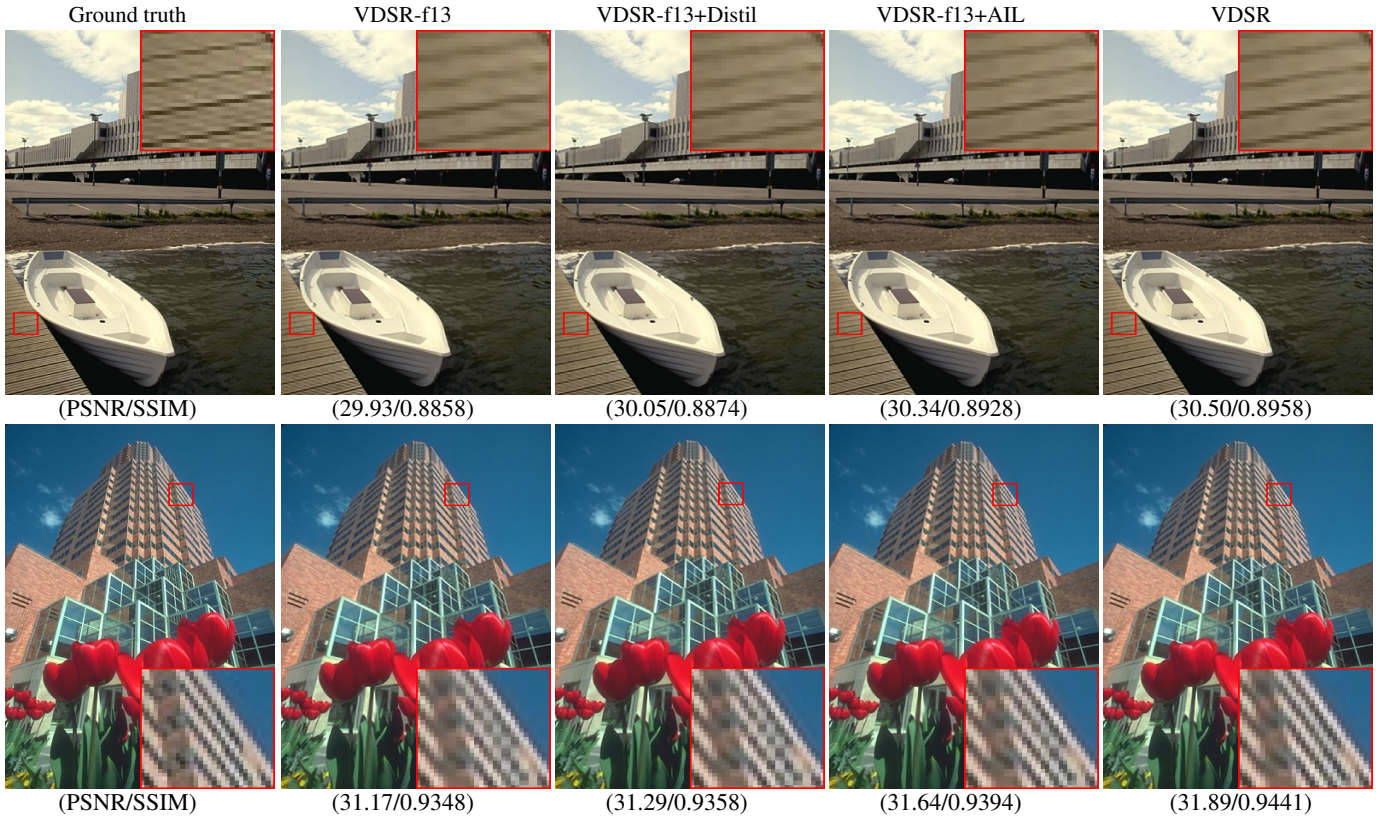


Fig. 4 Visual super-resolution results of `VDSR-f13`, `VDSR-f13+ILT`, `VDSR-f13+Distil` and `VDSR`. First row: the super-resolution results for image '78004' from BSD100 dataset when scaling factor is 2. Second row: the super-resolution results for image '86000' from BSD100 dataset when scaling factor is 2.

the teacher network can provide more valuable information than the discrete labels in ground truth. However, SISR is a regression problem where the ground truth is inherently continuous. Thus, the output of teacher network fails to provide more valuable information than the ground truth. In contrast, the proposed scheme enables to train the lightweight network with an easy-to-complex paradigm, which can enhance the generalization capacity of network.

Table 3 Parameters, computation complexity (e.g., FLOPS) and average running time (e.g., seconds) of different scales of lightweight networks for `VDSR`. The running time is evaluated over Set5 dataset with scaling factor 3 on a single CPU.

Method	Parameters	Complexity	Time
VDSR	$665K$	$2491M$	5.26
VDSR-f32/+AIL	$166K$	$642M$	2.13
VDSR-f22/+AIL	$79K$	$311M$	1.46
VDSR-f16/+AIL	$42K$	$170M$	0.96
VDSR-f13/+AIL	$28K$	$115M$	0.77
DRRN	$297K$	$483337M$	65.98
DRRN-f25/+AIL	$12K$	$18463M$	4.21

5.4.4 Convergence

In Algorithm 1, the network training and the importance learning are conducted in an alternative way. Thus, it is necessary to analysis the convergence of Algorithm 1. In addition to the theoretical illustration in Section 3.3, we further depict the PSNR and SSIM curves of `VSDR-f13+AIL` within $T = 10$ iterations on three test datasets in Figure 3. It can be found that `VSDR-f13+AIL` gradually improves the performance and ultimately converges with the proceeding of iterations.

5.5 Enhancing different scales of lightweight networks

In this part, we employ the proposed learning scheme to enhance the capacity of different scales of lightweight networks for the given `VDSR` teacher network. Specifically, we implement three different scales of lightweight networks with 16 (i.e., $\rho = 0.75$), 22 (i.e., $\rho = 0.66$) and 32 (i.e., $\rho = 0.5$) features maps in each convolution layer. Similar as experiments above, we separately train each lightweight net-

Table 4 Average PSNR/SSIM of `VDSR-f16`, `VDSR-f16+AIL` and `VDSR` on four test datasets. ↑PSNR/SSIM and ↓PSNR/SSIM denote the performance increase and decrease over VDSR-f16, respectively.

Dataset	scale	VDSR-f16	VDSR-f16+AIL		VDSR
Set5	×2	37.23/0.9581	37.51/0.9594	↑0.28/↑0.0013	37.59/0.9596
	×3	33.21/0.9174	33.45/0.9205	↑0.25/↑0.0031	33.69/0.9227
	×4	30.77/0.8731	31.05/0.8805	↑0.29/↑0.0073	31.34/0.8846
Set14	×2	32.85/0.9115	33.05/0.9131	↑0.19/↑0.0017	33.08/0.9135
	×3	29.63/0.8288	29.79/0.8318	↑0.15/↑0.0030	29.90/0.8339
	×4	27.78/0.7608	27.96/0.7665	↑0.18/↑0.0058	28.10/0.7699
BSD100	×2	31.65/0.8931	31.82/0.8955	↑0.17/↑0.0024	31.87/0.8961
	×3	28.61/0.7926	28.72/0.7956	↑0.10/↑0.0029	28.81/0.7979
	×4	27.05/0.7175	27.17/0.7222	↑0.11/↑0.0047	27.26/0.7253
Urban100	×2	30.06/0.9056	30.53/0.9120	↑0.47/↑0.0064	30.65/0.9135
	×3	26.58/0.8132	26.83/0.8210	↑0.25/↑0.0077	27.08/0.8273
	×4	24.68/0.7327	24.93/0.7439	↑0.24/↑0.0112	25.15/0.7520

Table 5 Average PSNR/SSIM of `VDSR-f22`, `VDSR-f22+AIL` and `VDSR` on four test datasets. The results of `VDSR-f32+AIL` comparable to or over `VDSR` are in bold. ↑PSNR/SSIM and ↓PSNR/SSIM denote the performance increase and decrease over `VDSR-f22`, respectively.

Dataset	scale	VDSR-f22	VDSR-f22+ILT		VDSR
Set5	×2	37.34/0.9586	**37.59/0.9597**	↑0.25/↑0.0011	37.59/0.9596
	×3	33.32/0.9190	33.54/0.9211	↑0.22/↑0.0022	33.69/0.9227
	×4	30.87/0.8759	31.14/0.8818	↑0.27/↑0.0059	31.34/0.8846
Set14	×2	32.94/0.9123	**33.11/0.9136**	↑0.17/↑0.0013	33.08/0.9135
	×3	29.71/0.8304	29.82/0.8324	↑0.11/↑0.0020	29.90/0.8339
	×4	27.86/0.7632	28.00/0.7676	↑0.14/↑0.0044	28.10/0.7699
BSD100	×2	31.73/0.8942	**31.87/0.8961**	↑0.14/↑0.0019	31.87/0.8961
	×3	28.67/0.7942	28.75/0.7964	↑0.08/↑0.0022	28.81/0.7979
	×4	27.10/0.7194	27.19/0.7231	↑0.09/↑0.0037	27.26/0.7253
Urban100	×2	30.28/0.9086	**30.64/0.9133**	↑0.36/↑0.0048	30.65/0.9135
	×3	26.72/0.8170	26.92/0.8233	↑0.19/↑0.0063	27.08/0.8273
	×4	24.78/0.7368	24.99/0.7464	↑0.22/↑0.0096	25.15/0.7520

Table 6 Average PSNR/SSIM of `VDSR-f32`, `VDSR-f32+AIL` and `VDSR` on four test datasets. The results of `VDSR-f32+AIL` comparable to or over `VDSR` are in bold. ↑PSNR/SSIM and ↓PSNR/SSIM denote the performance increase and decrease over `VDSR-f32`, respectively.

Dataset	scale	VDSR-f32	VDSR-f32+AIL		VDSR
Set5	×2	37.49/0.9593	**37.68/0.9601**	↑0.19/↑0.0008	37.59/0.9596
	×3	33.35/0.9191	**33.68/0.9227**	↑0.32/↑0.0036	33.69/0.9227
	×4	31.01/0.8783	31.26/0.8840	↑0.25/↑0.0057	31.34/0.8846
Set14	×2	33.03/0.9130	**33.19/0.9144**	↑0.16/↑0.0014	33.08/0.9135
	×3	29.71/0.8306	**29.89/0.8339**	↑0.18/↑0.0032	29.90/0.8339
	×4	27.94/0.7654	28.08/0.7695	↑0.14/↑0.0041	28.10/0.7699
BSD100	×2	31.81/0.8953	31.93/0.8970	↑0.12/↑0.0017	31.87/0.8961
	×3	28.67/0.7943	28.80/0.7979	↑0.13/↑0.0035	28.81/0.7979
	×4	27.15/0.7212	27.24/0.7248	↑0.09/↑0.0037	27.26/0.7253
Urban100	×2	30.51/0.9116	**30.84/0.9155**	↑0.32/↑0.0040	30.65/0.9135
	×3	26.74/0.8177	27.05/0.8270	↑0.31/↑0.0093	27.08/0.8273
	×4	24.88/0.7412	25.10/0.7506	↑0.21/↑0.0094	25.15/0.7520

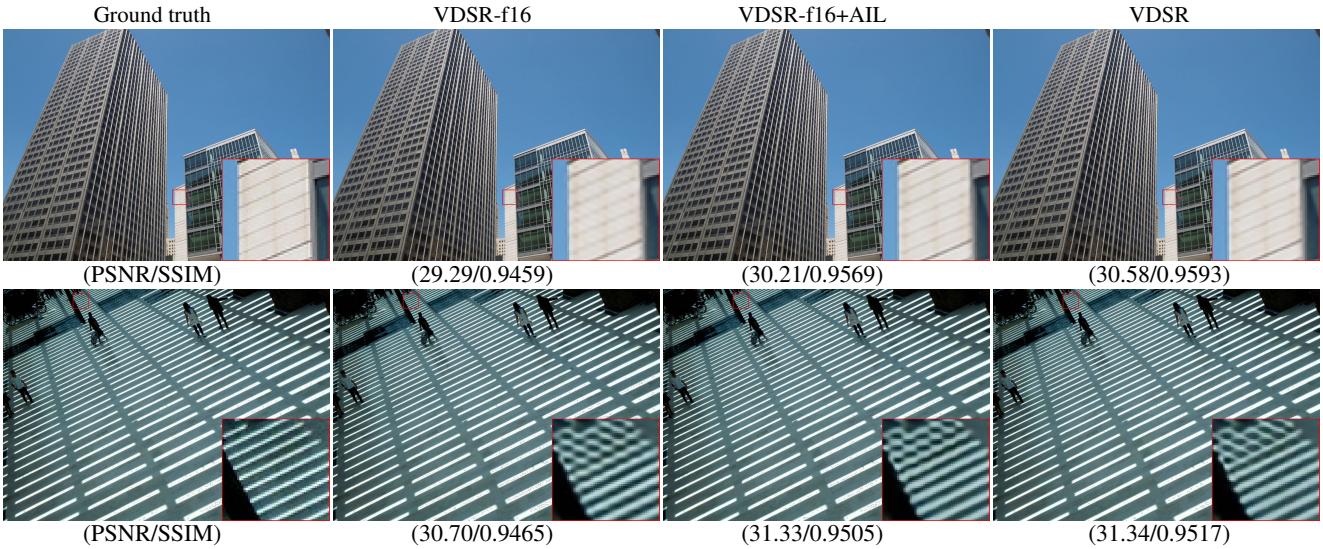


Fig. 5 Visual super-resolution results of `VDSR-f16`, `VDSR-f16+AIL` and `VDSR`. First row: the super-resolution results for image '96' from Urban100 dataset when scaling factor is 2. Second row: the super-resolution results for image '93' from Urban100 dataset when scaling factor is 3.

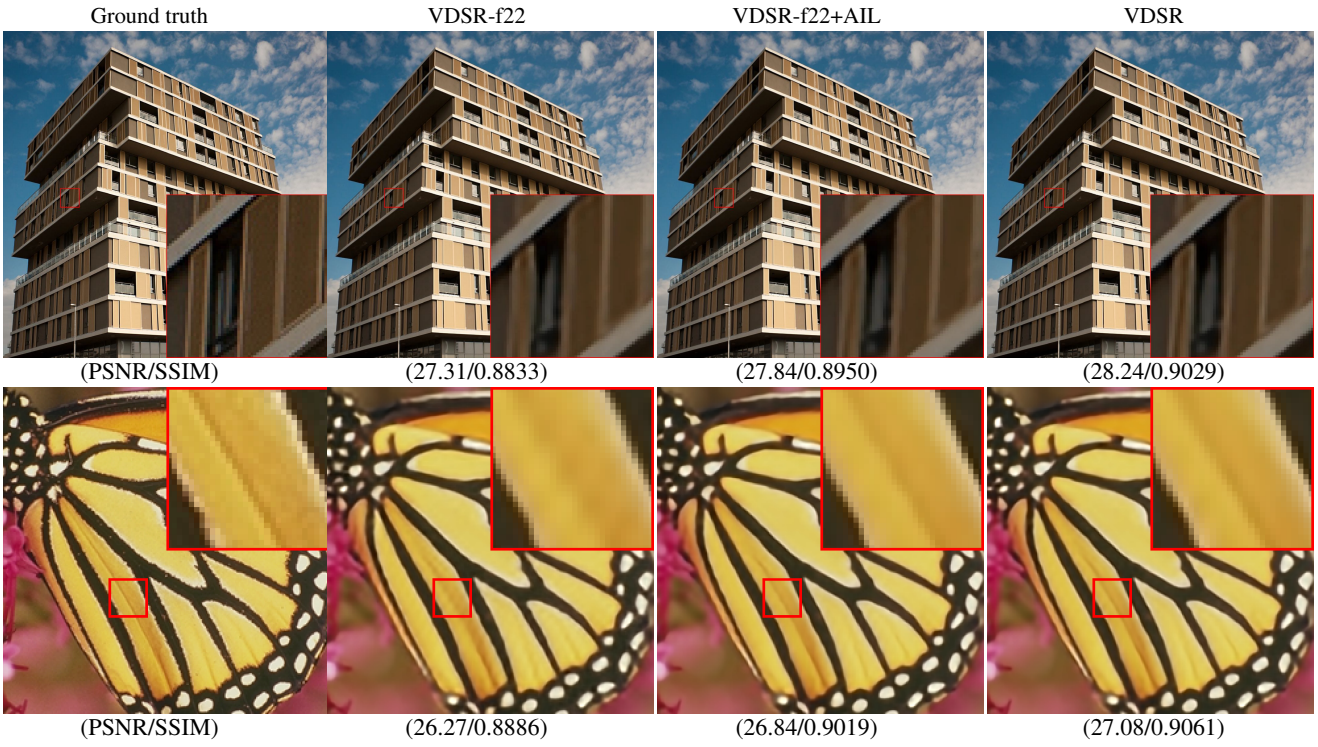


Fig. 6 Visual super-resolution results of `VDSR-f22`, `VDSR-f22+AIL` and `VDSR`. First row: the super-resolution results for image '87' from Urban100 dataset when scaling factor is 3. Second row: the super-resolution results for image 'butterfly' from Set5 dataset when scaling factor is 4.

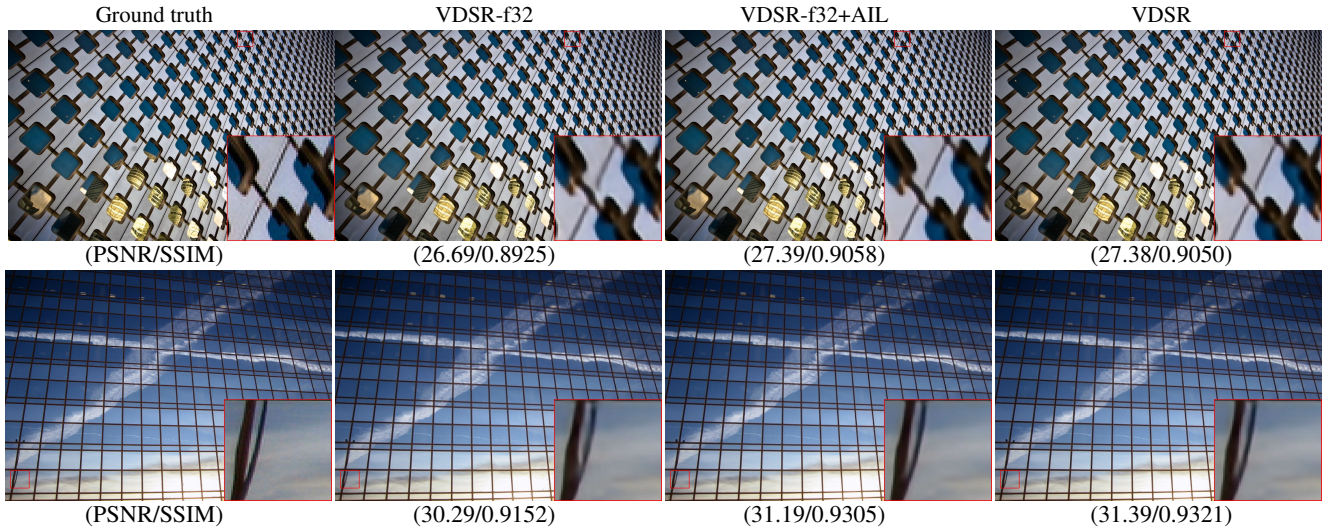


Fig. 7 Visual super-resolution results of VDSR−f32, VDSR−f32+AIL and VDSR. First row: the super-resolution results for image 'ppt3' from Set14 dataset when scaling factor is 3. Second row: the super-resolution results for image '55' from Urban100 dataset when scaling factor is 4.

work with the traditional learning scheme in Eq. (1) and the proposed one in Algorithm 1. The resulted networks are termed with the same naming way as Section 5.4. For example, two trained lightweight networks with 16 feature maps are termed VDSR−f16 and VDSR−f16+AIL, respectively. VDSR−f16 denotes the baseline method.

Before discussing the performance of each network, we first analysis their amount of parameters as well as the computational complexity. Providing that the testing image is of size 256×256, the parameters and theoretical computational complexity of these lightweight networks as well as the teacher network VDSR are given in Table 3. For example, the amount of parameters as well as the computational complexity of VDSR−f32 and VDSR−f32+AIL are only 25% of that for VDSR.

Under the same experimental settings, the quantitative results of all networks on four test datasets are provided in Table 4, Table 5 and Table 6. It can be found that the proposed adaptive importance learning scheme enhances the performance of lightweight networks obviously. For example, in Table 4, when the scaling factor is 2 on the Set5 dataset, the superiority of VDSR−f16+AIL over VDSR−f16 in PSNR and SSIM is up to 0.28db and 0.0013, respectively. Moreover, the superiority of VDSR−f32+AIL is more obvious on the more challenging dataset. For example, when the scaling factor is 2 on the Urban100 dataset, the superiority of VDSR−f16+AIL over VDSR−f16 in PSNR and SSIM is even up to 0.47db and 0.0064, respectively. In addition, we find that the proposed learning scheme performs the best on scaling factor 2 among three scaling factors. For example, as shown in Table 5 and Table 6, VDSR−f22+AIL produces comparable results on four test datasets to that of VDSR, and VDSR−f32+AIL even outperforms VDSR, especially on the Urban100 dataset on which the superiority is up to 0.19db in PSNR. The reason is intuitive. Compared with other two scaling factors, the SISR task on scaling factor 2 is relatively easier and contains many pixels that cannot be well reconstructed by the baseline network (e.g., VDSR−f16) but may be well reconstructed when the capacity of the lightweight network is maximized. Thus, with the easy-to-complex learning paradigm, the proposed scheme is able to improve the performance more obviously. In contrast, the SISR task on other two scaling factors contains extensive complex pixels beyond the maximum capacity of the network, which cannot be well reconstructed even with the easy-to-hard learning paradigm. According to these results, we can conclude that the proposed adaptive importance learning scheme is able to enhance the performance of different scales of lightweight networks in SISR. More evidence in visual results can be found in Figure 5, 6 and 7.

5.6 Enhancing lightweight network with other architectures

Due to not involving modifying the architecture of network, the proposed learning scheme can be directly applied to any lightweight DCNN based SISR methods. To demonstrate this point, we further evaluate the proposed learning scheme on another seminal network for SISR, DRRN (Tai et al, 2017). Specifically, we implement a lightweight network with 25 feature maps (i.e., $\rho = 0.8$) in each convolution layer. The corresponding parameters as well computational complexity can be found in Table 3. Then, we train this lightweight network with the traditional learning scheme in Eq. (1) and the proposed adaptive importance learning scheme as Algorithm 1. In the proposed learning scheme, the pre-trained

Table 7 Average PSNR/SSIM of DRRN−f25, DRRN−f25+AIL and DRRN on four test datasets. ↑PSNR/SSIM and ↓PSNR/SSIM denote the performance increase and decrease over DRRN−f25, respectively.

Dataset	scale	DRRN-f25	DRRN-f25+AIL		DRRN
Set5	×2	37.02/0.9575	37.52/0.9593	↑0.50/↑0.0018	37.69/0.9602
	×3	33.26/0.9181	33.64/0.9214	↑0.37/↑0.0033	34.02/0.9257
	×4	30.92/0.8740	31.28/0.8832	↑0.36/↑0.0092	31.69/0.8899
Set14	×2	32.80/0.9110	33.11/0.9135	↑0.31/↑0.0025	33.31/0.9152
	×3	29.64/0.8290	29.83/0.8325	↑0.19/↑0.0035	30.05/0.8369
	×4	27.82/0.7606	28.05/0.7682	↑0.23/↑0.0076	28.35/0.7752
BSD100	×2	31.58/0.8926	31.84/0.8955	↑0.26/↑0.0030	32.04/0.8984
	×3	28.60/0.7929	28.76/0.7960	↑0.16/↑0.0030	28.96/0.8015
	×4	27.03/0.7167	27.21/0.7234	↑0.18/↑0.0067	27.42/0.7299
Urban100	×2	29.95/0.9048	30.58/0.9123	↑0.63/↑0.0075	31.22/0.9195
	×3	26.62/0.8140	26.96/0.8233	↑0.34/↑0.0093	27.57/0.8390
	×4	24.72/0.7331	25.04/0.7474	↑0.32/↑0.0143	25.57/0.7668

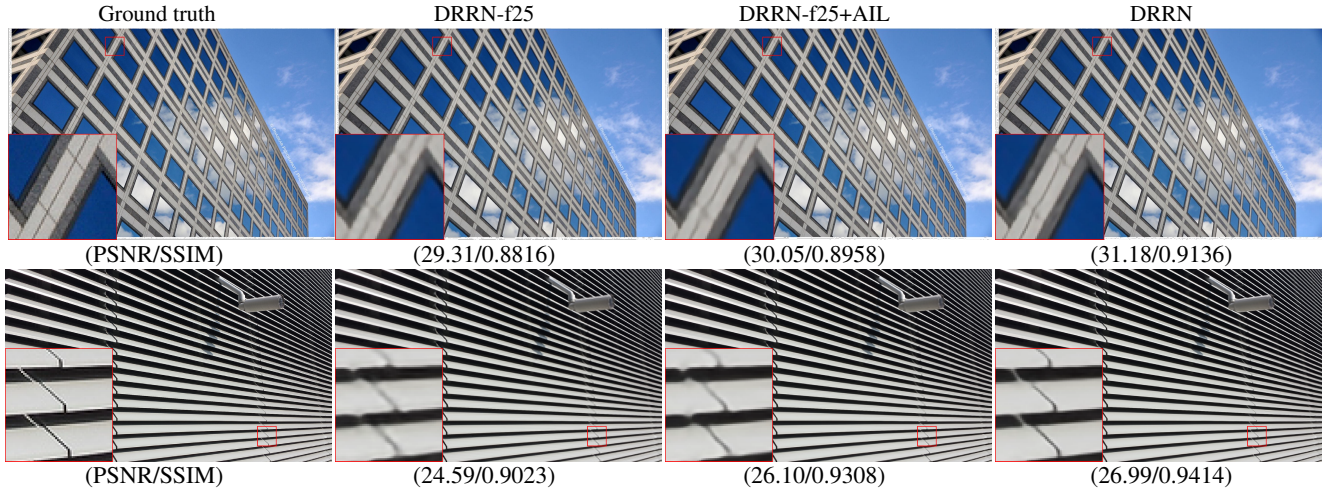


Fig. 8 Visual super-resolution results of DRRN−f25, DRRN−f25+AIL and DRRN. First row: the super-resolution results for image '35' from Urban100 dataset when scaling factor is 3. Second row: the super-resolution results for image '40' from Urban100 dataset when scaling factor is 4.

DRRN is utilized to initialized the importance. The obtained two networks are termed DRRN−f25 and DRRN−f25+AIL, respectively. Similar as that in Section 5.5, the quantitative and visual results of these two networks are provided in Table 7 and Figure 8. We can find that the propose learning scheme can obviously improve the performance of the corresponding lightweight network. For example, when the scaling factor is 2 on the Urban100 dataset, DRRN−f25+AIL outperforms DRRN−f25 in PSNR and SSIm by 0.51db and 0.0059, respectively. In Figure 8, DRRN−f25+ILT and DRRN−f25+AIL produces more sharp and clear results than that of DRRN−f25.

In previous experiments, we customize all lightweight networks by reducing the amount of filters in each convolution layer from a given teacher network. As mentioned in Section 4, there are some other choices Dong et al (2016b); Shi et al (2016) that focus on investigating new architecture. To further demonstrate the effectiveness of the proposed learning scheme on those network with specialized lightweight architectures, we employ it to train the FSRCNN (Dong et al, 2016b) which exhibits a hourglass-shape structure. Similar as previous experiments, given the lightweight network, we train it separately with the traditional learning scheme as Eq. (1) and the proposed adaptive importance learning in Algorithm 1. The learned networks are termed FSRCNN and FSRCNN+AIL, respectively. For training FSRCNN+AIL, we adopt the pre-trained VDSR as the teacher network for importance initialization. The numerical results of these two networks on four test datasets are reported in Table 8. Since we adopt a larger training dataset, the performance of the FSRCNN is slightly higher that in Dong et al (2016b). In Table 8, we can find that FSRCNN+AIL surpasses FSRCNN

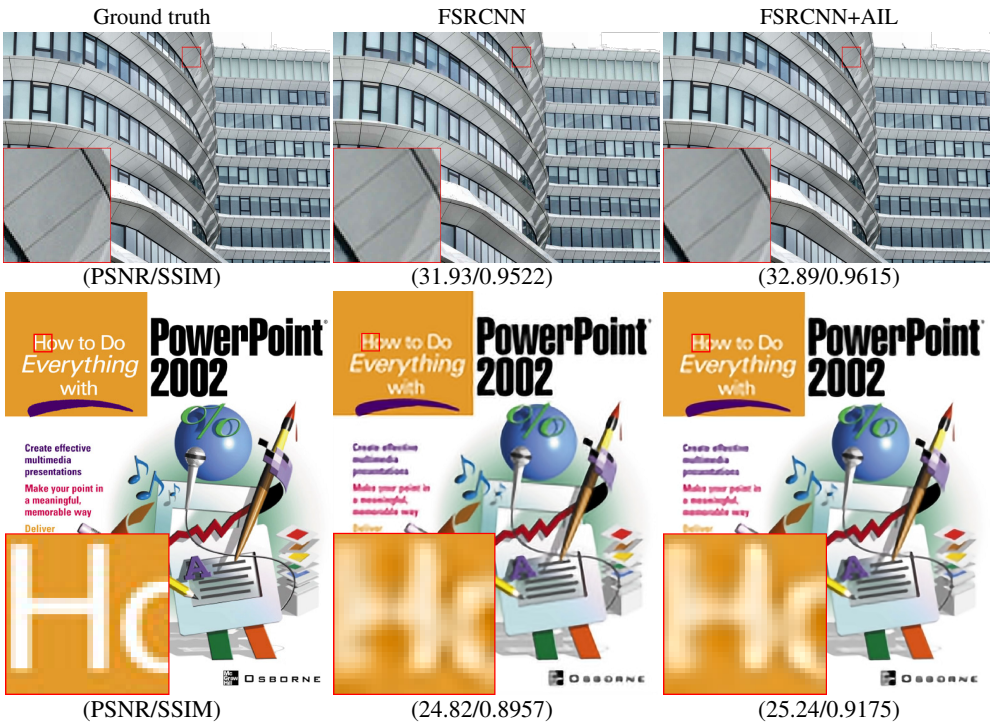


Fig. 9 Visual super-resolution results of FSRCNN and FSRCNN+AIL. First row: the super-resolution results for image '52' from Urban100 dataset when scaling factor is 2. Second row: the super-resolution results for image 'ppt3' from Set14 dataset when scaling factor is 4.

Table 8 Average PSNR/SSIM of FSRCNN and FSRCNN+AIL on four test datasets. ↑PSNR/SSIM and ↓PSNR/SSIM denote the performance increase and decrease over FSRCNN, respectively.

Dataset	scale	FSRCNN	FSRCNN+AIL	
Set5	×2	37.01/0.9570	**37.41/0.9587**	↑0.40/↑0.0017
	×3	33.01/0.9143	**33.33/0.9188**	↑0.32/↑0.0045
	×4	30.66/0.8699	**30.98/0.8776**	↑0.31/↑0.0077
Set14	×2	32.74/0.9103	**33.00/0.9123**	↑0.26/↑0.0020
	×3	29.56/0.8262	**29.73/0.8300**	↑0.17/↑0.0038
	×4	27.71/0.7583	**27.92/0.7643**	↑0.20/↑0.0061
BSD100	×2	31.55/0.8921	**31.72/0.8941**	↑0.18/↑0.0020
	×3	28.55/0.7904	**28.67/0.7938**	↑0.12/↑0.0034
	×4	27.02/0.7162	**27.13/0.7204**	↑0.11/↑0.0042
Urban100	×2	29.77/0.9010	**30.24/0.9076**	↑0.47/↑0.0066
	×3	26.43/0.8071	**26.70/0.8158**	↑0.27/↑0.0088
	×4	24.61/0.7279	**24.83/0.7393**	↑0.23/↑0.0115

clearly in all cases. For example, when the scaling factor is 2 on both Set5 and Urban100 datasets, FSRCNN+AIL improves the PSRN of FSRCNN at least by 0.4db. More visual evidence can be found in Figure 9.

Therefore, we can conclude that the proposed adaptive importance learning scheme is a general SISR learning scheme and can be applied to any given lightweight network architectures for performance enhancement.

6 Conclusion

In this study, we present an easy-to-complex learning strategy, termed adaptive importance learning scheme, to enhance the fitting capacity of a given lightweight SISR network architecture. The propose learning scheme integrates network training and pixel-wise importance learning into a joint optimization framework, which can be well addressed in an alternative way. Through dynamically updating the importance of image pixels, the network starts with learning to reconstruct easy pixel at the beginning, and then are exposed to more and more complex pixels for training. By doing this, the fitting capacity can be gradually enhanced and ultimately maximized when the learning scheme converges. In addition, the learning scheme enables seamlessly assimilating the knowledge from a more powerful teacher network to initialize the importance of image pixels, which leads to better initial capacity of the network as well as the ultimate super-resolution performance. Extensive experimental results on four benchmark datasets demonstrate that the proposed learning strategy is able to enhance the super-resolution performance of a given lightweight network with different architectures or scales.

It is noteworthy that the proposed adaptive importance learning is general learning paradigm for enhancing the lightweight regression networks. In the future, we will further ex-

ploit its potential benefits in other regression problems, e.g., image denoising, image deblurring and image inpainting etc.

References

Basu S, Christensen J (2013) Teaching classification boundaries to humans. In: AAAI

Bengio Y, Louradour J, Collobert R, Weston J (2009) Curriculum learning. In: Proceedings of the 26th annual international conference on machine learning, ACM, pp 41–48

Bevilacqua M, Roumy A, Guillemot C, Alberi-Morel ML (2012) Low-complexity single-image super-resolution based on nonnegative neighbor embedding

Dong C, Loy CC, He K, Tang X (2016a) Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence 38(2):295–307

Dong C, Loy CC, Tang X (2016b) Accelerating the super-resolution convolutional neural network. In: European Conference on Computer Vision, Springer, pp 391–407

Efrat N, Glasner D, Apartsin A, Nadler B, Levin A (2013) Accurate blur models vs. image priors in single image super-resolution. In: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, pp 2832–2839

Glasner D, Bagon S, Irani M (2009) Super-resolution from a single image. In: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, pp 349–356

Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arXiv preprint arXiv:150302531

Huang JB, Singh A, Ahuja N (2015) Single image super-resolution from transformed self-exemplars. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 5197–5206

Jiang L, Meng D, Mitamura T, Hauptmann AG (2014) Easy samples first: Self-paced reranking for zero-example multimedia search. In: Proceedings of the 22nd ACM international conference on Multimedia, ACM, pp 547–556

Khan F, Mutlu B, Zhu X (2011) How do humans teach: On curriculum learning and teaching dimension. In: Advances in Neural Information Processing Systems, pp 1449–1457

Kim J, Kwon Lee J, Mu Lee K (2016a) Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1646–1654

Kim J, Kwon Lee J, Mu Lee K (2016b) Deeply-recursive convolutional network for image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1637–1645

Kim KI, Kwon Y (2010) Single-image super-resolution using sparse regression and natural image prior. IEEE transactions on pattern analysis and machine intelligence 32(6):1127–1133

Lai WS, Huang JB, Ahuja N, Yang MH (2017) Deep laplacian pyramid networks for fast and accurate super-resolution. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp 624–632

Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, et al (2017) Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4681–4690

Lin TY, Goyal P, Girshick R, He K, Dollar P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2980–2988

Liu L, Wang P, Shen C, Wang L, Van Den Hengel A, Wang C, Shen HT (2017) Compositional model based fisher vector coding for image classification. IEEE transactions on pattern analysis and machine intelligence 39(12):2335–2348

Mao XJ, Shen C, Yang YB (2016) Image restoration using convolutional auto-encoders with symmetric skip connections. arxiv preprint. arXiv preprint arXiv:160608921 2

Martin D, Fowlkes C, Tal D, Malik J (2001) 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, vol 2, pp 416–423

Paszke A, Gross S, Chintala S, Chanan G (2017) Pytorch

Romero A, Ballas N, Kahou SE, Chassang A, Gatta C, Bengio Y (2014) Fitnets: Hints for thin deep nets. arXiv preprint arXiv:14126550

Shi W, Caballero J, Huszár F, Totz J, Aitken AP, Bishop R, Rueckert D, Wang Z (2016) Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1874–1883

Sun J, Xu Z, Shum HY (2008) Image super-resolution using gradient profile prior. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, pp 1–8

Tai Y, Yang J, Liu X (2017) Image super-resolution via deep recursive residual network. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol 1

Timofte R, De Smet V, Van Gool L (2014) A+: Adjusted anchored neighborhood regression for fast super-resolution. In: Asian Conference on Computer Vision, Springer, pp 111–126

Wang Z, Liu D, Yang J, Han W, Huang T (2015) Deep networks for image super-resolution with sparse prior. In: Proceedings of the IEEE International Conference on Computer Vision, pp 370–378

Wei W, Zhang L, Tian C, Plaza A, Zhang Y (2017) Structured sparse coding-based hyperspectral imagery denoising with intracluster filtering. IEEE Transactions on Geoscience and Remote Sensing 55(12):6860–6876

Yang CY, Ma C, Yang MH (2014) Single-image super-resolution: A benchmark. In: European Conference on Computer Vision, Springer, pp 372–386

Yang J, Wright J, Huang TS, Ma Y (2010) Image super-resolution via sparse representation. IEEE transactions on image processing 19(11):2861–2873

Zeyde R, Elad M, Protter M (2010) On single image scale-up using sparse-representations. In: International conference on curves and surfaces, Springer, pp 711–730

Zhang L, Wei W, Shi Q, Shen C, Hengel Avd, Zhang Y (2017a) Beyond low rank: A data-adaptive tensor completion method. arXiv preprint arXiv:170801008

Zhang L, Wei W, Zhang Y, Shen C, van den Hengel A, Shi Q (2018) Cluster sparsity field: An internal hyperspectral imagery prior for reconstruction. International Journal of Computer Vision pp 1–25

Zhang Y, Xiang T, Hospedales TM, Lu H (2017b) Deep mutual learning. arXiv preprint arXiv:170600384