Learning with Privileged Information for Efficient Image Super-Resolution

Wonkyung Lee*, Junghyup Lee*, Dohyung Kim*, and Bumsub Ham†

Yonsei University

Abstract. Convolutional neural networks (CNNs) have allowed remarkable advances in single image super-resolution (SISR) over the last decade. Most SR methods based on CNNs have focused on achieving performance gains in terms of quality metrics, such as PSNR and SSIM, over classical approaches. They typically require a large amount of memory and computational units. FSRCNN, consisting of few numbers of convolutional layers, has shown promising results, while using an extremely small number of network parameters. We introduce in this paper a novel distillation framework, consisting of teacher and student networks, that allows to boost the performance of FSRCNN drastically. To this end, we propose to use ground-truth high-resolution (HR) images as privileged information. The encoder in the teacher learns the degradation process, subsampling of HR images, using an imitation loss. The student and the decoder in the teacher, having the same network architecture as FSRCNN, try to reconstruct HR images. Intermediate features in the decoder, affordable for the student to learn, are transferred to the student through feature distillation. Experimental results on standard benchmarks demonstrate the effectiveness and the generalization ability of our framework, which significantly boosts the performance of FSRCNN as well as other SR methods. Our code and model are available online: https://cvlab.yonsei.ac.kr/projects/PISR.

Keywords: Privileged information, super-resolution, distillation

1 Introduction

Single image super-resolution (SISR) aims at reconstructing a high-resolution (HR) image from a low-resolution (LR) one, which has proven useful in various tasks including object detection [3], face recognition [17, 63], medical imaging [16], and information forensics [35]. With the great success of deep learning, SRCNN [10] first introduces convolutional neural networks (CNNs) for SISR, outperforming classical approaches by large margins. After that, CNN-based SR methods focus on designing wider [34, 49, 62] or deeper [20, 29, 32, 39, 60, 61] network architectures for the performance gains. They require a high computational cost and a large amount of memory, and thus implementing them directly on a single chip for, *e.g.*, televisions and mobile phones, is extremely hard without neural processing units and off-chip memory.

* equal contribution † corresponding author (bumsub.ham@yonsei.ac.kr)

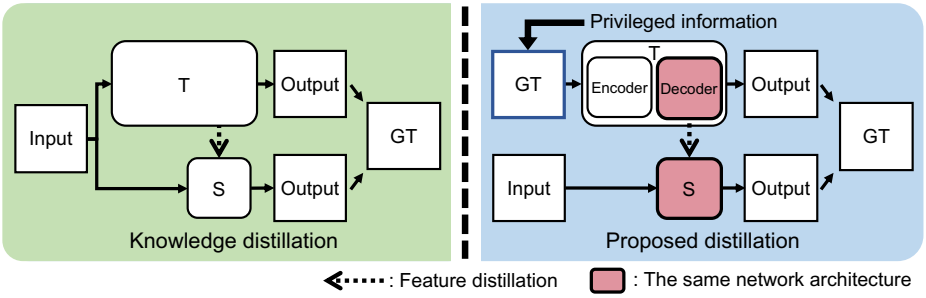

Fig. 1: Compressing networks using knowledge distillation (left) transfers the knowledge from a large teacher model (T) to a small student model (S), with the same input, *e.g.*, LR images in the case of SISR. Differently, the teacher in our framework (right) takes the ground truth (*i.e.*, HR image) as an input, exploiting it as privileged information, and transfers the knowledge via feature distillation. (Best viewed in color.)

Many works introduce cost-effective network architectures [1, 11, 18, 19, 27, 30, 45] to reduce the computational burden and/or required memory, using recursive layers [30, 45] or additional modules specific for SISR [1, 27]. Although they offer a good compromise in terms of PSNR and speed/memory, specially-designed or recursive architectures may be difficult to implement on hardware devices. Network pruning [19] and parameter quantization [18], typically used for network compression, are alternative ways for efficient SR networks, where the pruning removes redundant connections of nodes and the quantization reduces bit-precision of weights or activations. The speedup achieved by the pruning is limited due to irregular memory accesses and poor data localizations [53], and the performance of the network quantization is inherently bound by that of a full-precision model. Knowledge distillation is another way of model compression, where a large model (*i.e.*, a teacher network) transfers a softened version of the output distribution (*i.e.*, logits) [23] or intermediate feature representations [2, 14, 22, 43] to a small one (*i.e.*, a student network), which has shown the effectiveness in particular for the task of image classification. Generalized distillation [37] goes one step further, allowing a teacher to make use of extra (privileged) information at training time, and assisting the training process of a student network with the complementary knowledge [15, 24].

We present in this paper a simple yet effective framework for an efficient SISR method. The basic idea is that ground-truth HR images can be thought of as privileged information (Fig. 1), which has not been explored in both SISR and privileged learning. It is true that the HR image includes the complementary information (*e.g.*, high-frequency components) of LR images, but current SISR methods have used it just to penalize an incorrect reconstruction at the end of CNNs. On the contrary, our approach to using HR images as privileged information allows to extract the complementary features and leverage them explicitly for the SISR task. To implement this idea, we introduce a novel distillation framework where teacher and student networks try to reconstruct HR

image but using different inputs (*i.e.*, ground-truth HR and corresponding LR images for the teacher and the student, respectively), which is clearly different from the conventional knowledge distillation framework (Fig. 1). Specifically, the teacher network has an hourglass architecture consisting of an encoder and a decoder. The encoder extracts compact features from HR images while encouraging them to imitate LR counterparts using an imitation loss. The decoder, which has the same network architecture as the student, reconstructs the HR images again using the compact features. Intermediate features in the decoder are then transferred to the student via feature distillation, such that the student learns the knowledge (*e.g.*, high frequencies or fine details of HR inputs) of the teacher trained with the privileged data (*i.e.*, HR image). Note that our framework is useful in that the student can be initialized with the network parameters of the decoder, which allows to transfer the reconstruction capability of the teacher to the student. We mainly exploit FSRCNN [11] as the student network, since it has a hardware-friendly architecture (*i.e.*, a stack of convolutional layers) and the number of parameters is extremely small compared to other CNN-based SR methods. Experimental results on standard SR benchmarks demonstrate the effectiveness of our approach, which boosts the performance of FSRCNN without any additional modules. To the best of our knowledge, our framework is the first attempt to leverage the privileged information for SISR. The main contributions of our work can be summarized as follows:

- We present a novel distillation framework for SISR that leverages the ground truth (*i.e.*, HR images) as privileged information to transfer the important knowledge of the HR images to a student network.
- We propose to use an imitation loss to train a teacher network, making it possible to distill the knowledge a student is able to learn.
- We demonstrate that our approach boosts the performance of the current SISR methods, significantly, including FSRCNN [11], VDSR [29], IDN [27], and CARN [1]. We show an extensive experimental analysis with ablation studies.

2 Related work

SISR. Early works on SISR design image priors to constrain the solution space [9,28,55], and leverage external datasets to learn the relationship between HR and LR images [6, 13, 44, 47, 56], since lots of HR images can be reconstructed from a single LR image. CNNs have allowed remarkable advances in SISR. Dong *et al.* pioneer the idea of exploiting CNNs for SISR, and propose SRCNN [10] that learns a mapping function directly from input LR to output HR images. Recent methods using CNNs exploit a much larger number of convolutional layers. Sparse [32,34,39] or dense [20,49,62] skip connections between them prevent a gradient vanishing problem, achieving significant performance gains over classical approaches. More recently, efficient networks for SISR in terms of memory and/or runtime have been introduced. Memory-efficient SR

methods [30, 33, 45, 46] reduce the number of network parameters by reusing them recursively. They further improve the reconstruction performance using residual units [45], memory [46] or feedback [33] modules but at the cost of runtime. Runtime-efficient methods [1, 11, 26, 27] on the other hand are computationally cheap. They use cascaded [1] or multi-branch [26, 27] architectures, or exploit group convolutions [8, 54]. The main drawback of such SR methods is that their hardware implementations are difficult due to the network architectures specially-designed for the SR task. FSRCNN [11] reduces both runtime and memory. It uses typical convolutional operators with a small number of filters and feature channels, except the deconvolution layer at the last part of the network. Although FSRCNN has a hardware-friendly network architecture, it is largely outperformed by current SR methods.

Feature distillation. The purpose of knowledge distillation is to transfer the representation ability of a large model (teacher) to a small one (student) for enhancing the performance of the student model. It has been widely used to compress networks, typically for classification tasks. In this framework, the softmax outputs of a teacher are regarded as soft labels, providing informative clues beyond discrete labels [23]. Recent methods extend this idea to feature distillation, which transfers intermediate feature maps [2, 43], their transformations [22, 58], the differences of features before and after a stack of layers [57], or pairwise relations within feature maps [36]. In particular, the variational information distillation (VID) method [2] transfers the knowledge by maximizing the mutual information between feature maps of teacher and student networks. We exploit VID for feature distillation, but within a different framework. Instead of sharing the same inputs (*i.e.*, LR images) with the student, our teacher network inputs HR images, that contain the complementary information of LR images, to take advantage of privileged information.

Closely related to ours, SRKD [14] applies the feature distillation technique to SISR in order to compress the size of SR network, where a student is trained to have similar feature distributions to those of a teacher. Following the conventional knowledge distillation, the student and teacher networks in SRKD use the same inputs of LR images. This is clearly different from our method in that our teacher takes ground-truth HR images as inputs, allowing to extract more powerful feature representations for image reconstruction.

Generalized distillation. Learning using privileged information [50, 51] is a machine learning paradigm that uses extra information, which requires an additional cost, at training time, but with no accessibility to it at test time. In a broader context, generalized distillation [37] covers both feature distillation and learning using privileged information. The generalized distillation enables transferring the privileged knowledge of a teacher to a student. For example, the works of [15, 24] adopt the generalized distillation approach for object detection and action recognition, where depth images are used as privileged information. In the framework, a teacher is trained to extract useful features from depth im-

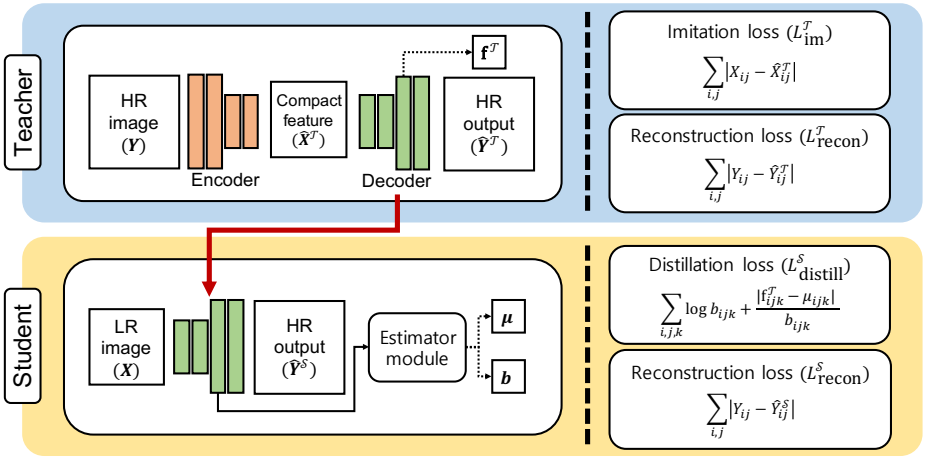

Fig. 2: **Overview of our framework.** A teacher network inputs a HR image $\mathbf{Y}$ and extracts a compact feature representation $\hat{\mathbf{X}}^{\mathcal{T}}$ using an encoder. The decoder in the network then reconstructs a HR output $\hat{\mathbf{Y}}^{\mathcal{T}}$. To train the teacher network, we use imitation $L_{\text{im}}^{\mathcal{T}}$ and reconstruction $L_{\text{recon}}^{\mathcal{T}}$ losses. After training the teacher, a student network is initialized with weights of the decoder in the teacher network (red line), and restores a HR output $\hat{\mathbf{Y}}^{\mathcal{S}}$ from a LR image $\mathbf{X}$. Note that the student network and the decoder share the same network architecture. The estimator module takes intermediate feature maps of the student network, and outputs location and scale maps, $\boldsymbol{\mu}$ and $\mathbf{b}$, respectively. To train the student network, we exploit a reconstruction loss $L_{\text{recon}}^{\mathcal{S}}$ together with a distillation loss $L_{\text{distill}}^{\mathcal{S}}$ using the intermediate representation $\mathbf{f}^{\mathcal{T}}$ of the teacher network and the parameter maps of $\boldsymbol{\mu}$ and $\mathbf{b}$. See text for details. (Best viewed in color.)

ages. They are then transferred to a student which takes RGB images as inputs, allowing the student to learn complementary representations from privileged information. Our method belongs to generalized distillation, since we train a teacher network with ground-truth HR images, which can be viewed as privileged information, and transfer the knowledge to a student network. Different from previous methods, our method does not require an additional cost for privileged information, since the ground truth is readily available at training time.

3 Method

We denote by $\mathbf{X}$ and $\mathbf{Y}$ LR and ground-truth HR images. Given the LR image $\mathbf{X}$, we reconstruct a high-quality HR output $\hat{\mathbf{Y}}^{\mathcal{S}}$ efficiently in terms of both speed and memory. To this end, we present an effective framework consisting of teacher and student networks. The teacher network learns to distill the knowledge from privileged information (*i.e.*, a ground-truth HR image $\mathbf{Y}$). After training the teacher network, we transfer the knowledge distilled from the teacher to the

student to boost the reconstruction performance. We show in Fig. 2 an overview of our framework.

3.1 Teacher

In order to transfer knowledge from a teacher to a student, the teacher should be superior to the student, while extracting informative features. To this end, we treat ground-truth HR images as privileged information, and exploit an *intelligent teacher* [50]. As will be seen in our experiments, the network architecture of the teacher influences the SR performance significantly. As the teacher network inputs ground-truth HR images, it may not be able to extract useful features, and just learn to copy the inputs for the reconstruction of HR images, regardless of its capacity. Moreover, a large difference for the number of network parameters or the performance gap between the teacher and the student discourages the distillation process [7,41]. To reduce the gap while promoting the teacher to capture useful features, we exploit an hourglass architecture for the teacher network. It projects the HR images into a low-dimensional feature space to generate compact features, and reconstructs the original HR images from them, such that the teacher learns to extract better feature representations for an image reconstruction task. Specifically, the teacher network consists of an encoder $G^{\mathcal{T}}$ and a decoder $F^{\mathcal{T}}$. Given a pair of LR and HR images, the encoder $G^{\mathcal{T}}$ transforms the input HR image $\mathbf{Y}$ into the feature representation $\hat{\mathbf{X}}^{\mathcal{T}}$ in a low-dimensional space:

$$\hat{\mathbf{X}}^{\mathcal{T}} = G^{\mathcal{T}}(\mathbf{Y}), \tag{1}$$

where the feature representation of $\hat{\mathbf{X}}^{\mathcal{T}}$ has the same size as the LR image. The decoder $F^{\mathcal{T}}$ reconstructs the HR image $\hat{\mathbf{Y}}^{\mathcal{T}}$ using the compact feature $\hat{\mathbf{X}}^{\mathcal{T}}$:

$$\hat{\mathbf{Y}}^{\mathcal{T}} = F^{\mathcal{T}}(\hat{\mathbf{X}}^{\mathcal{T}}). \tag{2}$$

For the decoder, we use the same architecture as the student network. It allows the teacher to have a similar representational capacity as the student, which has proven useful in [41].

Loss. To train the teacher network, we use reconstruction and imitation losses, denoted by $L^{\mathcal{T}}_{\text{recon}}$ and $L^{\mathcal{T}}_{\text{im}}$, respectively. The reconstruction term computes the mean absolute error (MAE) between the HR image $\mathbf{Y}$ and its reconstruction $\hat{\mathbf{Y}}^{\mathcal{T}}$ defined as:

$$L^{\mathcal{T}}_{\text{recon}} = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}|Y_{ij} - \hat{Y}^{\mathcal{T}}_{ij}|, \tag{3}$$

where H and W are height and width of the HR image, respectively, and we denote by Y_{ij} an intensity value of $\mathbf{Y}$ at position (i, j). It encourages the encoder output (*i.e.*, compact feature $\hat{\mathbf{X}}^{\mathcal{T}}$) to contain useful information for the image reconstruction and forces the decoder to reconstruct the HR image again using the compact feature. The imitation term restricts the representational power of

the encoder, making the output of the encoder close to the LR image. Concretely, we define this term as the MAE between the LR image $\mathbf{X}$ and the encoder output $\hat{\mathbf{X}}^{\mathcal{T}}$:

$$L_{\text{im}}^{\mathcal{T}} = \frac{1}{H'W'} \sum_{i=1}^{H'} \sum_{j=1}^{W'} |X_{ij} - \hat{X}_{ij}^{\mathcal{T}}|, \tag{4}$$

where H' and W' are height and width of the LR image, respectively. This facilitates an initialization of the student network that takes the LR image $\mathbf{X}$ as an input. Note that our framework avoids the trivial solution that the compact feature becomes the LR image since the network parameters in the encoder are updated by both the imitation and reconstruction terms. The overall objective is a sum of reconstruction and imitation terms, balanced by the parameter $\lambda^{\mathcal{T}}$:

$$L_{\text{total}}^{\mathcal{T}} = L_{\text{recon}}^{\mathcal{T}} + \lambda^{\mathcal{T}} L_{\text{im}}^{\mathcal{T}}. \tag{5}$$

3.2 Student

A student network has the same architecture as the decoder $F^{\mathcal{T}}$ in the teacher, but uses a different input. It takes a LR image $\mathbf{X}$ as an input and generates a HR image $\hat{\mathbf{Y}}^{\mathcal{S}}$:

$$\hat{\mathbf{Y}}^{\mathcal{S}} = F^{\mathcal{S}}(\mathbf{X}). \tag{6}$$

We initialize the weights of the student network with those of the decoder in the teacher. This transfers the reconstruction capability of the teacher to the student and provides a good starting point for optimization. Note that several works [15, 24] point out that how to initialize network weights is crucial for the performance of a student. We adopt FSRCNN [11], a hardware-friendly SR architecture, as the student network $F^{\mathcal{S}}$.

Loss. Although the network parameters of the student $F^{\mathcal{S}}$ and the decoder $F^{\mathcal{T}}$ in the teacher are initially set to the same, the features extracted from them are different due to the different inputs. Besides, these parameters are not optimized with input LR images. We further train the student network $F^{\mathcal{S}}$ with a reconstruction loss $L_{\text{recon}}^{\mathcal{S}}$ and a distillation loss $L_{\text{distill}}^{\mathcal{S}}$. The reconstruction term is similarly defined as Eq. (3) using the ground-truth HR image and its reconstruction from the student network, dedicating to the SISR task:

$$L_{\text{recon}}^{\mathcal{S}} = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} |Y_{ij} - \hat{Y}_{ij}^{\mathcal{S}}|. \tag{7}$$

The distillation term focuses on transferring the knowledge of the teacher to the student. Overall, we use the following loss to train the student network:

$$L_{\text{total}}^{\mathcal{S}} = L_{\text{recon}}^{\mathcal{S}} + \lambda^{\mathcal{S}} L_{\text{distill}}^{\mathcal{S}}, \tag{8}$$

where $\lambda^{\mathcal{S}}$ is a distillation parameter. In the following, we describe the distillation loss in detail.

We adopt the distillation loss proposed in the VID method [2], which maximizes mutual information between the teacher and the student. We denote by $\mathbf{f}^{\mathcal{T}}$ and $\mathbf{f}^{\mathcal{S}}$ the intermediate feature maps of the teacher and student networks, respectively, having the same size of $C \times H' \times W'$, where C is the number of channels. We define mutual information $I(\mathbf{f}^{\mathcal{T}}; \mathbf{f}^{\mathcal{S}})$ as follows:

$$I(\mathbf{f}^{\mathcal{T}}; \mathbf{f}^{\mathcal{S}}) = H(\mathbf{f}^{\mathcal{T}}) - H(\mathbf{f}^{\mathcal{T}}|\mathbf{f}^{\mathcal{S}}), \tag{9}$$

where we denote by $H(\mathbf{f}^{\mathcal{T}})$ and $H(\mathbf{f}^{\mathcal{T}}|\mathbf{f}^{\mathcal{S}})$ marginal and conditional entropies, respectively. To maximize the mutual information, we should minimize the conditional entropy $H(\mathbf{f}^{\mathcal{T}}|\mathbf{f}^{\mathcal{S}})$. However, an exact optimization w.r.t the weights of the student is intractable, as it involves an integration over a conditional probability $p(\mathbf{f}^{\mathcal{T}}|\mathbf{f}^{\mathcal{S}})$. The variational information maximization technique [4] instead approximates the conditional distribution $p(\mathbf{f}^{\mathcal{T}}|\mathbf{f}^{\mathcal{S}})$ using a parametric model $q(\mathbf{f}^{\mathcal{T}}|\mathbf{f}^{\mathcal{S}})$, such as the Gaussian or Laplace distributions, making it possible to find a lower bound of the mutual information $I(\mathbf{f}^{\mathcal{T}}; \mathbf{f}^{\mathcal{S}})$. Using this technique, we maximize the lower bound of mutual information $I(\mathbf{f}^{\mathcal{T}}; \mathbf{f}^{\mathcal{S}})$ for feature distillation. As the parametric model $q(\mathbf{f}^{\mathcal{T}}|\mathbf{f}^{\mathcal{S}})$, we use a multivariate Laplace distribution with parameters of location and scale, $\boldsymbol{\mu} \in \mathbb{R}^{C \times H' \times W'}$ and $\mathbf{b} \in \mathbb{R}^{C \times H' \times W'}$, respectively. We define the distillation loss $L^{\mathcal{S}}_{\text{distill}}$ as follows:

$$L^{\mathcal{S}}_{\text{distill}} = \frac{1}{CH'W'} \sum_{i=1}^{C} \sum_{j=1}^{H'} \sum_{k=1}^{W'} \log b_{ijk} + \frac{|f^{\mathcal{T}}_{ijk} - \mu_{ijk}|}{b_{ijk}}, \tag{10}$$

where we denote by μ_{ijk} the element of $\boldsymbol{\mu}$ at the position (i, j, k). This minimizes the distance between the features $\mathbf{f}^{\mathcal{T}}$ of the teacher and the location map $\boldsymbol{\mu}$. The scale map $\mathbf{b}$ controls the extent of distillation. For example, when the student does not benefit from the distillation, the scale parameter b_{ijk} increases in order to reduce the extent of distillation. This is useful for our framework where the teacher and student networks take different inputs, since it adaptively determines the features the student is affordable to learn from the teacher. The term $\log b_{ijk}$ prevents a trivial solution where the scale parameter goes to infinite. We estimate these maps of $\boldsymbol{\mu}$ and $\mathbf{b}$ from the features of the student $\mathbf{f}^{\mathcal{S}}$. Note that other losses designed for feature distillation can also be used in our framework (See the supplementary material).

Estimator module. We use a small network to estimate the parameters of location $\boldsymbol{\mu}$ and scale $\mathbf{b}$ in Eq. (10). It consists of location and scale branches, where each takes the features of the student $\mathbf{f}^{\mathcal{S}}$ and estimates the location and scale maps, separately. Both branches share the same network architecture of two 1×1 convolutional layers and a PReLU [21] between them. For the scale branch, we add the softplus function ($\zeta(x) = \log(1 + e^x)$) [12] at the last layer, forcing the scale parameter to be positive. Note that the estimation module is used only at training time.

4 Experiments

4.1 Experimental details

Implementation details. The encoder in the teacher network consists of 4 blocks of convolutional layers followed by a PReLU [21]. All the layers, except the second one, perform convolutions with stride 1. In the second block, we use the convolution with stride s (*i.e.*, a scale factor) to downsample the size of the HR image to that of the LR image. The kernel sizes of the first two and the last two blocks are 5×5 and 3×3, respectively. The decoder in the teacher and the student network have the same architecture as FSRCNN [11] consisting of five components: Feature extraction, shrinking, mapping, expanding, and deconvolution modules. We add the estimator module for location and scale maps on top of the expanding module in the student network. We use these maps together with the output features of the expanding module in the decoder to compute the distillation loss. We set the hyperparameters for losses using a grid search on the DIV2K dataset [48], and choose the ones ($\lambda^{\mathcal{T}} = 10^{-4}$, $\lambda^{\mathcal{S}} = 10^{-6}$) that give the best performance. We implement our framework using `PyTorch` [42].

Training. To train our network, we use the training split of DIV2K [48] corresponding 800 pairs of LR and HR images, where the LR images are synthesized by bicubic downsampling. We randomly crop HR patches of size 192×192 from the HR images. LR patches are cropped from the corresponding LR images according to the scale factor. For example, LR patches of size 96×96 are used for the scale factor of 2. We use data augmentation techniques, including random rotation and horizontal flipping. The teacher network is trained with random initialization. We train our model with a batch size of 16 about 1000k iterations over the training data. We use the Adam [31] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. As a learning rate, we use 10^{-3} and reduce it until 10^{-5} using a cosine annealing technique [38].

Evaluation. We evaluate our framework on standard benchmarks including Set5 [5], Set14 [59], B100 [40], and Urban100 [25]. Following the experimental protocol in [34], we use the peak signal to noise ratio (PSNR) and the structural similarity index (SSIM) [52] on the luminance channel as evaluation metrics.

4.2 Ablation studies

We present an ablation analysis on each component of our framework. We report quantitative results in terms of the average PSNR on Set5 [5] with the scale factor of 2. The results on a large dataset (*i.e.*, B100 [40]) can be seen in the supplementary material. We show in Table 1 the average PSNR for student networks trained with variants of our framework. The results of the baseline in the first row are obtained using FSRCNN [11]. From the second row, we can clearly see that feature distillation boosts the PSNR performance. For the teacher network

Table 1: Average PSNR of student and teacher networks, trained with variants of our framework, on the Set5 [5] dataset. We use FSRCNN [11], reproduced by ourselves using the DIV2K [48] dataset without distillation, as the baseline in the first row. We denote by VID_G and VID_L VID losses [2] with the Gaussian and Laplace distributions, respectively. The performance gains of each component over the baseline are shown in the parentheses. The number in bold indicates the best performance and underscored one is the second best.

Hourglass architecture	Weight transfer	$L_{\text{im}}^{\mathcal{T}}$	$L_{\text{distill}}^{\mathcal{S}}$	Student PSNR	Teacher PSNR
-	-	-	-	37.15 (baseline)	-
✗	-	-	MAE	37.19 (+0.04)	57.60
✓	✗	✗	MAE	37.22 (+0.07)	37.70
✓	✓	✗	MAE	37.23 (+0.08)	37.70
✓	✓	✓	MAE	37.27 (+0.12)	37.65
✓	✓	✓	VID_G [2]	<u>37.31</u> (+0.16)	37.65
✓	✓	✓	VID_L [2]	**37.33** (+0.18)	37.65

in the second row, we use the same network architecture as FSRCNN except for the deconvolution layers. In contrast to FSRCNN, the teacher inputs HR images, and thus we replace the deconvolution layer with a convolutional layer, preserving the size of the inputs. We can see from the third row that a teacher network with an hourglass architecture improves the student performance. The hourglass architecture limits the performance of the teacher and degrades the performance (*e.g.*, a 19.9dB decrease compared to that of the teacher in the second row), reducing the performance gap between the teacher and the student. This allows the feature distillation to be more effective, thus the student of the third row performs better (37.22dB) than that of the second row (37.19dB), which can also be found in recent works [7, 41]. The fourth row shows that the student network benefits from initializing the network weights with those of the decoder in the teacher, since this provides a good starting point for learning, and transfers the reconstruction capability of the teacher. From the fifth row, we observe that an imitation loss further improves the PSNR performance, making it easier for the student to learn features from the teacher. The next two rows show that the VID loss [2], especially with the Laplace distribution (VID_L), provides better results than the MAE, and combining all components gives the best performance. The distillation loss based on the MAE forces the feature maps of the student and teacher networks to be the same. This strong constraint on the feature maps is, however, problematic in our case, since we use different inputs for the student and teacher networks. The VID method allows the student to learn important features adaptively. We also compare the performance of our framework and a typical distillation approach with different losses in the supplementary material.

4.3 Analysis on compact features

In Fig. 3, we show an analysis on compact features in spatial and frequency domains. Compared to the LR image, the compact features $\hat{\mathbf{X}}^{\mathcal{T}}$ show high-

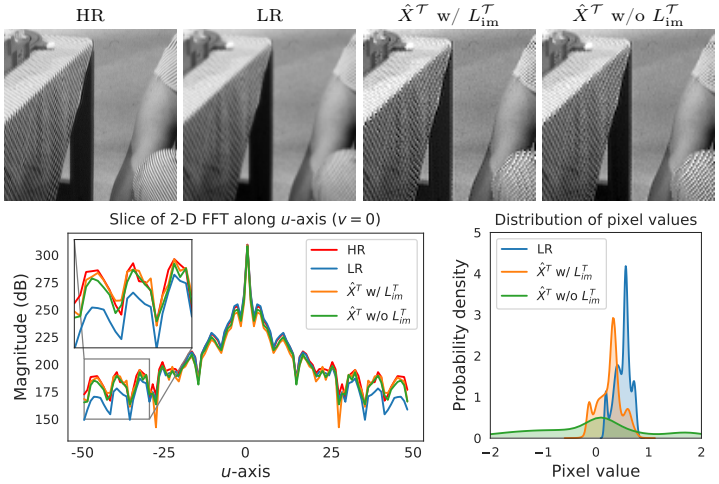

Fig. 3: Analysis on compact features in spatial (top) and frequency (bottom left) domains and the distribution of pixel values (bottom right). To visualize the compact features in the frequency domain, we apply the 2D Fast Fourier Transform (FFT) to the image, obtaining its magnitude spectrum. It is then sliced along the u-axis. (Best viewed in color.)

frequency details regardless of whether the imitation loss $L_{\text{im}}^{\mathcal{T}}$ is used or not. This can also be observed in the frequency domain – The compact features contain more high-frequency components than the LR image, and the magnitude spectrums of them are more similar to that of the HR image especially for high-frequency components. By taking these features as inputs, the decoder in the teacher shows the better performance than the student (Table 1) despite the fact that they have the same architecture. This demonstrates that the compact features extracted from the ground truth contain useful information for reconstructing the HR image, encouraging the student to reconstruct more accurate results via feature distillation. In the bottom right of the Fig. 3, we can see that the pixel distributions of the LR image and the compact feature are largely different without the imitation loss, discouraging the weight transfer to the student. The imitation loss $L_{\text{im}}^{\mathcal{T}}$ alleviates this problem by encouraging the distributions of the LR image and the compact feature to be similar.

4.4 Results

Quantitative comparison. We compare in Table 2 the performance of our student model with the state of the art, particularly for efficient SISR methods [1, 10, 11, 26, 27, 30, 33, 45, 46]. For a quantitative comparison, we report the average PSNR and SSIM [52] for upsampling factors of 2, 3, and 4, on standard benchmarks [5, 25, 40, 59]. We also report the number of model parameters and operations (MultiAdds), required to reconstruct a HR image of size 1280×720,

Table 2: Quantitative comparison with the state of the art on SISR. We report the average PSNR/SSIM for different scale factors (2×, 3×, and 4×) on Set5 [5], Set14 [59], B100 [40], and Urban100 [25]. *: models reproduced by ourselves using the DIV2K [48] dataset without distillation; Ours: student networks of our framework.

Scale	Methods	Param.	MultiAdds	Runtime	Set5 [5] PSNR/SSIM	Set14 [59] PSNR/SSIM	B100 [40] PSNR/SSIM	Urban100 [25] PSNR/SSIM
2	FSRCNN [11]	13K	6.0G	0.83ms	37.05/0.9560	32.66/0.9090	31.53/0.8920	29.88/0.9020
	FSRCNN*	13K	6.0G	0.83ms	37.15/0.9568	32.71/0.9095	31.58/0.8913	30.05/0.9041
	FSRCNN (Ours)	13K	6.0G	0.83ms	**37.33**/**0.9576**	**32.79**/**0.9105**	**31.65**/**0.8926**	**30.24**/**0.9071**
	Bicubic Int.	-	-	-	33.66/0.9299	30.24/0.8688	29.56/0.8431	26.88/0.8403
	DRCN [30]	1,774K	17,974.3G	239.93ms	37.63/0.9588	33.04/0.9118	31.85/0.8942	30.75/0.9133
	DRRN [45]	297K	6,796.9G	105.76ms	37.74/0.9591	33.23/0.9136	32.05/0.8973	31.23/0.9188
	MemNet [46]	677K	2,662.4G	21.06ms	37.78/0.9597	33.28/0.9142	32.08/0.8978	31.31/0.9195
	CARN [1]	1,592K	222.8G	8.43ms	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256
	IDN [27]	591K	136.5G	7.01ms	37.83/0.9600	33.30/0.9148	32.08/0.8985	31.27/0.9196
	SRFBN [33]	3,631K	1,126.7G	108.52ms	38.11/0.9609	33.82/0.9196	32.29/0.9010	32.62/0.9328
	IMDN [26]	694K	159.6G	6.97ms	38.00/0.9605	33.63/0.9177	32.19/0.8996	32.17/0.9283
3	FSRCNN [11]	13K	5.0G	0.72ms	33.18/0.9140	29.37/0.8240	28.53/0.7910	26.43/0.8080
	FSRCNN*	13K	5.0G	0.72ms	33.15/0.9157	29.45/0.8250	28.52/0.7895	26.49/0.8089
	FSRCNN (Ours)	13K	5.0G	0.72ms	**33.31**/**0.9179**	**29.57**/**0.8276**	**28.61**/**0.7919**	**26.67**/**0.8153**
	Bicubic Int.	-	-	-	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349
	DRCN [30]	1,774K	17,974.3G	239.19ms	33.82/0.9226	29.76/0.8311	28.80/0.7963	27.15/0.8276
	DRRN [45]	297K	6,796.9G	98.58ms	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378
	MemNet [46]	677K	2,662.4G	11.33ms	34.09/0.9248	30.00/0.8350	28.96/0.8001	27.56/0.8376
	CARN [1]	1,592K	118.8G	3.86ms	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493
	IDN [27]	591K	60.6G	3.62ms	34.11/0.9253	29.99/0.8354	28.95/0.8013	27.42/0.8359
	SRFBN [33]	3,631K	500.8G	76.74ms	34.70/0.9292	30.51/0.8461	29.24/0.8084	28.73/0.8641
	IMDN [26]	703K	71.7G	5.36ms	34.36/0.9270	30.32/0.8417	29.09/0.8046	28.17/0.8519
4	FSRCNN [11]	13K	4.6G	0.67ms	30.72/0.8660	27.61/0.7550	26.98/0.7150	24.62/0.7280
	FSRCNN*	13K	4.6G	0.67ms	30.89/0.8748	27.72/0.7599	27.05/0.7176	24.76/0.7358
	FSRCNN (Ours)	13K	4.6G	0.67ms	**30.95**/**0.8759**	**27.77**/**0.7615**	**27.08**/**0.7188**	**24.82**/**0.7393**
	Bicubic Int.	-	-	-	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577
	DRCN [30]	1,774K	17,974.3G	243.62ms	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510
	DRRN [45]	297K	6,796.9G	57.09ms	31.68/0.8888	28.21/0.7721	27.38/0.7284	25.44/0.7638
	MemNet [46]	677K	2,662.4G	8.55ms	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630
	CARN [1]	1,592K	90.9G	3.16ms	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837
	IDN [27]	591K	34.1G	3.08ms	31.82/0.8903	28.25/0.7730	27.41/0.7297	25.41/0.7632
	SRFBN [33]	3,631K	281.7G	48.39ms	32.47/0.8983	28.81/0.7868	27.72/0.7409	26.60/0.8015
	IMDN [26]	715K	41.1G	4.38ms	32.21/0.8948	28.58/0.7811	27.56/0.7353	26.04/0.7838

and present the average runtime of each method measured on the Set5 [5] using the same machine with a NVIDIA Titan RTX GPU. From this table, we can observe two things: (1) Our student model trained with the proposed framework outperforms FSRCNN [11] by a large margin, consistently for all scale factors, even both have the same network architecture. It demonstrates the effectiveness of our approach to exploiting ground-truth HR images as privileged information; (2) The model trained with our framework offers a good compromise in terms of PSNR/SSIM and the number of parameters/operations/runtimes. For example, DRCN [30] requires 1,774K parameters, 17,974.3G operations and average runtime of 233.93ms to achieve the average PSNR of 30.75dB on Urban100 [25] for a factor of 2. On the contrary, our framework further boosts FSRCNN without modifying the network architecture, achieving the average PSNR of 30.24dB with 13K parameters/6.0G operations only, while taking 0.83ms for inference.

In Table 3, we show the performances of student networks, adopting the architectures of other SR methods, trained with our framework using the DIV2K dataset [48]. We reproduce their models (denoted by *) using the same training

Table 3: Quantitative results of student networks using other SR methods. We report the average PSNR for different scale factors (2×, 3×, and 4×) on Set5 [5] and B100 [40]. *: models reproduced by ourselves using the DIV2K [48] dataset; Ours: student networks of our framework.

Methods	2x Set5/B100	3x Set5/B100	4x Set5/B100
FSRCNN-L*	37.59/31.90	33.76/28.81	31.47/27.29
FSRCNN-L (Ours)	**37.65**/**31.92**	**33.85**/**28.83**	**31.52**/**27.30**
VDSR [29]	37.53/31.90	33.67/28.82	31.35/27.29
VDSR*	37.64/31.96	33.80/28.83	31.37/27.25
VDSR (Ours)	**37.77**/**32.00**	**33.85**/**28.86**	**31.51**/**27.29**
IDN [27]	37.83/32.08	34.11/28.95	31.82/27.41
IDN*	37.88/32.12	34.22/29.02	**32.03**/27.49
IDN (Ours)	**37.93**/**32.14**	**34.31**/**29.03**	32.01/**27.51**
CARN [1]	37.76/32.09	34.29/29.06	32.13/27.58
CARN*	37.75/32.02	34.08/28.94	31.77/27.44
CARN (Ours)	**37.82**/**32.08**	**34.10**/**28.95**	**31.83**/**27.45**

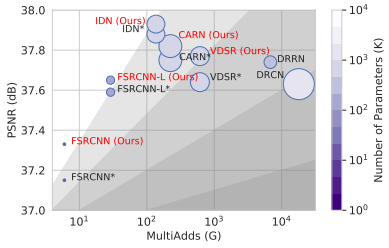

Fig. 4: Trade-off between the number of operations and the average PSNR on Set5 [5] (2×). The size of the circle and background color indicate the number of parameters and the efficiency of the model (white: high, black: low), respectively. (Best viewed in color.)

setting but without distillation. The FSRCNN-L has the same components as FSRCNN [11] but with much more parameters (126K vs. 13K), where the numbers of filters in feature extraction and shrinking components are both 56, and the mapping module consists of 4 blocks of convolutional layers. Note that the multi-scale learning strategy in the CARN [1] is not used for training the network, and thus the performance is slightly lower than the original one. We can see that all the SISR methods benefit from our framework except for IDN [27] for the scale factor of 4 on Set5. In particular, the performances of the variant of FSRCNN [10] and VDSR [29] are significantly boosted through our framework. Additionally, our framework further improves the performances of the cost-effective SR methods [1, 27], which are specially-designed to reduce the number of parameters and operations while improving the reconstruction performance. Considering the performance gains of recent SR methods, the results are significant, demonstrating the effectiveness and generalization ability of our framework. For example IDN [27] and SRFBN [33] outperform the second-best methods by 0.05dB and 0.02dB, respectively, in terms of PSNR on Set5 [5] for a factor of 2. We visualize in Fig. 4 the performance comparison of student networks using various SR methods and the state of the art in terms of the number of operations and parameters. It confirms once more the efficiency of our framework.

Qualitative results. We show in Fig. 5 reconstruction examples on the Urban100 [25] and Set14 [59] datasets using the student networks. We can clearly see that the student models provide better qualitative results than their baselines. In particular, our models remove artifacts (*e.g.*, the borders around the sculpture in the first row) and reconstruct small-scale structures (*e.g.*, windows in the second row and the iron railings in the last row) and textures (*e.g.*, the patterns of the tablecloth in the third row). More qualitative results can be seen in the supplementary material.

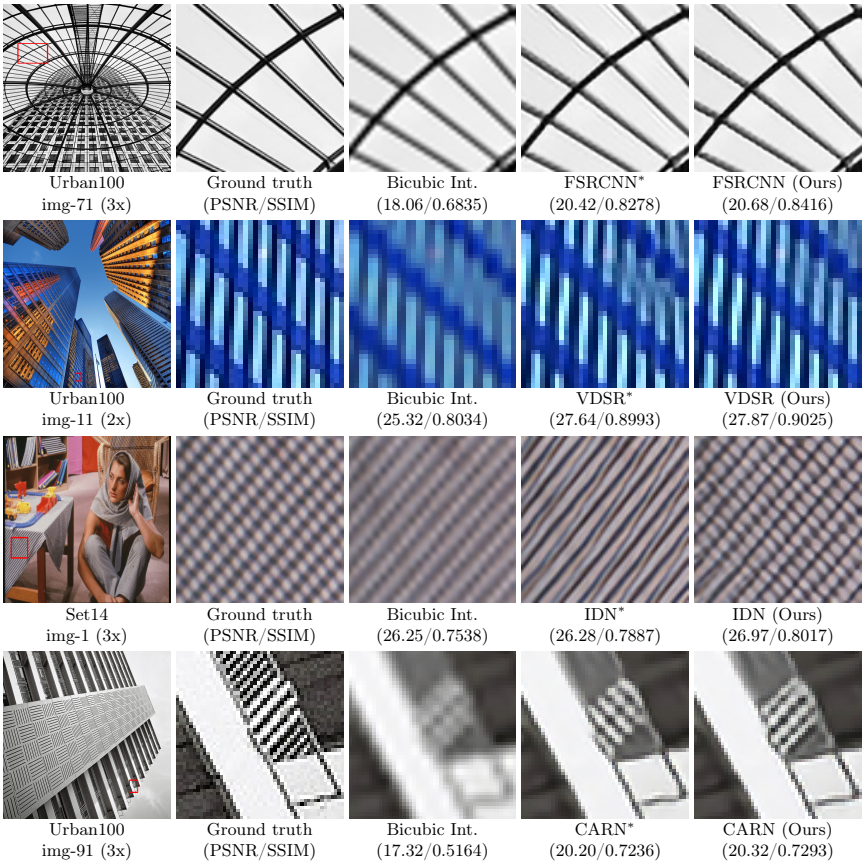

Fig. 5: Visual comparison of reconstructed HR images (2× and 3×) on Urban100 [25] and Set14 [59]. We report the average PSNR/SSIM in the parentheses. (Best viewed in color.)

5 Conclusion

We have presented a novel distillation framework for SISR leveraging ground-truth HR images as privileged information. The detailed analysis on each component of our framework clearly demonstrates the effectiveness of our approach. We have shown that the proposed framework substantially improves the performance of FSRCNN as well as other methods. In future work, we will explore distillation losses specific to our model to further boost the performance.

Acknowledgement.

This research was supported by the Samsung Research Funding & Incubation Center for Future Technology (SRFC-IT1802-06).

References

1. Ahn, N., Kang, B., Sohn, K.A.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: ECCV (2018)
2. Ahn, S., Hu, S.X., Damianou, A., Lawrence, N.D., Dai, Z.: Variational information distillation for knowledge transfer. In: CVPR (2019)
3. Bai, Y., Zhang, Y., Ding, M., Ghanem, B.: SOD-MTGAN: Small object detection via multi-task generative adversarial network. In: ECCV (2018)
4. Barber, D., Agakov, F.V.: The IM algorithm: A variational approach to information maximization. In: NIPS (2003)
5. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: BMVC (2012)
6. Chang, H., Yeung, D.Y., Xiong, Y.: Super-resolution through neighbor embedding. In: CVPR (2004)
7. Cho, J.H., Hariharan, B.: On the efficacy of knowledge distillation. In: ICCV (2019)
8. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: CVPR (2017)
9. Dai, S., Han, M., Xu, W., Wu, Y., Gong, Y., Katsaggelos, A.K.: SoftCuts: A soft edge smoothness prior for color image super-resolution. IEEE TIP **18**(5) (2009)
10. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE TPAMI **38**(2) (2015)
11. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: ECCV (2016)
12. Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., Garcia, R.: Incorporating second-order functional knowledge for better option pricing. In: NIPS (2001)
13. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based super-resolution. IEEE CG&A **22**(2) (2002)
14. Gao, Q., Zhao, Y., Li, G., Tong, T.: Image super-resolution using knowledge distillation. In: ACCV (2018)
15. Garcia, N.C., Morerio, P., Murino, V.: Modality distillation with multiple stream networks for action recognition. In: ECCV (2018)
16. Greenspan, H.: Super-resolution in medical imaging. The Computer Journal **52**(1) (2008)
17. Gunturk, B.K., Batur, A.U., Altunbasak, Y., Hayes, M.H., Mersereau, R.M.: Eigenface-domain super-resolution for face recognition. IEEE TIP **12**(5) (2003)
18. Han, S., Mao, H., Dally, W.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In: ICLR (2016)
19. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. In: NIPS (2015)
20. Haris, M., Shakhnarovich, G., Ukita, N.: Deep back-projection networks for super-resolution. In: CVPR (2018)
21. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: ICCV (2015)
22. Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., Choi, J.Y.: A comprehensive overhaul of feature distillation. In: ICCV (2019)
23. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: NIPS Workshop (2014)
24. Hoffman, J., Gupta, S., Darrell, T.: Learning with side information through modality hallucination. In: CVPR (2016)

25. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR (2015)
26. Hui, Z., Gao, X., Yang, Y., Wang, X.: Lightweight image super-resolution with information multi-distillation network. In: ACMMM (2019)
27. Hui, Z., Wang, X., Gao, X.: Fast and accurate single image super-resolution via information distillation network. In: CVPR (2018)
28. Jian Sun, Zongben Xu, Heung-Yeung Shum: Image super-resolution using gradient profile prior. In: CVPR (2008)
29. Kim, J., Kwon Lee, J., Mu Lee, K.: Accurate image super-resolution using very deep convolutional networks. In: CVPR (2016)
30. Kim, J., Kwon Lee, J., Mu Lee, K.: Deeply-recursive convolutional network for image super-resolution. In: CVPR (2016)
31. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
32. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR (2017)
33. Li, Z., Yang, J., Liu, Z., Yang, X., Jeon, G., Wu, W.: Feedback network for image super-resolution. In: CVPR (2019)
34. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: CVPR Workshop (2017)
35. Lin, W.S., Tjoa, S.K., Zhao, H.V., Liu, K.R.: Digital image source coder forensics via intrinsic fingerprints. IEEE TIFS **4**(3) (2009)
36. Liu, Y., Chen, K., Liu, C., Qin, Z., Luo, Z., Wang, J.: Structured knowledge distillation for semantic segmentation. In: CVPR (2019)
37. Lopez-Paz, D., Bottou, L., Schölkopf, B., Vapnik, V.: Unifying distillation and privileged information. In: ICLR (2016)
38. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. In: ICLR (2017)
39. Mao, X., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: NIPS (2016)
40. Martin, D., Fowlkes, C., Tal, D., Malik, J., et al.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV (2001)
41. Mirzadeh, S.I., Farajtabar, M., Li, A., Ghasemzadeh, H.: Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. In: AAAI (2020)
42. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch (2017)
43. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: FitNets: Hints for thin deep nets. In: ICLR (2015)
44. Schulter, S., Leistner, C., Bischof, H.: Fast and accurate image upscaling with super-resolution forests. In: CVPR (2015)
45. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR (2017)
46. Tai, Y., Yang, J., Liu, X., Xu, C.: MemNet: A persistent memory network for image restoration. In: ICCV (2017)
47. Timofte, R., De, V., Gool, L.V.: Anchored neighborhood regression for fast example-based super-resolution. In: ICCV (2013)

48. Timofte, R., Agustsson, E., Van Gool, L., Yang, M.H., Zhang, L.: NTIRE 2017 challenge on single image super-resolution: Methods and results. In: CVPR Workshop (2017)
49. Tong, T., Li, G., Liu, X., Gao, Q.: Image super-resolution using dense skip connections. In: ICCV (2017)
50. Vapnik, V., Izmailov, R.: Learning using privileged information: Similarity control and knowledge transfer. JMLR **16** (2015)
51. Vapnik, V., Vashist, A.: A new learning paradigm: Learning using privileged information. Neural Networks **22**(5-6) (2009)
52. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., et al.: Image quality assessment: from error visibility to structural similarity. IEEE TIP **13**(4) (2004)
53. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: NIPS (2016)
54. Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: CVPR (2017)
55. Yan, Q., Xu, Y., Yang, X., Nguyen, T.Q.: Single image super-resolution based on gradient profile sharpness. IEEE TIP **24**(10) (2015)
56. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution as sparse representation of raw image patches. In: CVPR (2008)
57. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: CVPR (2017)
58. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: ICLR (2017)
59. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Curves and Surfaces (2010)
60. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE TIP **26** (2017)
61. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep CNN denoiser prior for image restoration. In: CVPR (2017)
62. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: CVPR (2018)
63. Zou, W.W., Yuen, P.C.: Very low resolution face recognition problem. IEEE TIP **21**(1) (2011)