IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 August 2023, accepted 4 October 2023, date of publication 11 October 2023, date of current version 16 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3323273

RESEARCH ARTICLE

TAKDSR: Teacher Assistant Knowledge Distillation Framework for Graphics Image Super-Resolution

MIN YOON, SEUNGHYUN LEE, (Associate Member, IEEE), AND BYUNG CHEOL SONG, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, Inha University, Incheon 22212, Republic of Korea

Corresponding author: Byung Cheol Song (bcsong@inha.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIT) under Grant 2022R1A2C2010095 and Grant 2022R1A4A1033549; in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korean Government (MSIT), Artificial Intelligence Convergence Innovation Human Resources Development, Inha University, under Grant RS-2022-00155915; and in part by the Development of High-Definition CIS and High-Speed DVS Enabled AI SoC for Object and Motion Recognition under Grant 2022-0-00955.

ABSTRACT This paper presents a framework for effectively applying knowledge distillation (KD) to super-resolution (SR) tasks for computer graphics (CG) images. Specifically, we propose TAKDSR, a KD framework for SR using a teacher assistant (TA) network. Recently, the performance of SR models has improved dramatically thanks to the development of deep learning. SR models have evolved into a form that requires a considerable amount of computation and parameters while adopting a complex neural network structure to improve performance. However, it is difficult to utilize conventional high-performance SR models for real-time up-scaling in CG applications requiring high resolution and high frame rate. To solve this, we employ an approach that applies KD to a lightweight SR model. At this time, if the high-resolution (HR) image is used as input for the teacher to show superior performance to the student, a large performance difference occurs between the two due to the excessive performance of the teacher. As a result, the teacher's knowledge has a significantly hard and complex nature, and when transferred to the student, the effect of KD can be rather weakened. Therefore, we adopt a TA network to facilitate the propagation of knowledge between teacher and student. At the same time, the distribution of compact features (CF), which are the decoder input of the teacher, is discretized so that it is compatible with the input distribution of the student, enabling effective KD. Experimental results demonstrate the proposed TAKDSR significantly improves the performance of a given SR model on CG image datasets.

INDEX TERMS Convolutional neural networks, graphics image, knowledge distillation, single image super-resolution.

I. INTRODUCTION

Single image super-resolution (SISR) is a computer vision task that aims to restore a high-resolution (HR) image from a given low-resolution (LR) image. It is useful because it can be combined with various tasks such as medical imaging and object detection. Meanwhile, SISR is a typical ill-posed problem in which multiple HR images can be restored from the same LR image. Thus, numerous studies have been conducted to solve this problem for a long time.

In early studies, methods based on interpolation or reconstruction were mainstream. But, recent SISR studies are actively utilizing deep learning. Since SRCNN [1], the first deep learning-based SISR using convolutional neural networks (CNNs), was introduced, SR techniques

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tong.

(*e.g.* VDSR [2], EDSR [3], SAN [4], HGSRCNN [5]) that introduced wider and deeper CNN models have been developed. Actually, they achieved groundbreaking quantitative/qualitative performance improvements compared to SRCNN.

On the other hand, industries utilizing computer graphics (CG) images such as virtual reality and metaverse are rapidly developing, and as a result, the range of utilization of CG images is gradually expanding. In addition, with the development of digital display technology, the needs for ultra-high resolution images are increasing. However, in order to produce ultra-high-resolution CG (moving) images, huge amounts of computation, time, and labor are required. The computing power of the hardware is also inevitably considerable. Thus, applying SISR to low-resolution CG images can be considered as a reasonable alternative to solve the problem of insufficient hardware performance and enormous cost. In fact, NVIDIA DLSS [6] and AMD FSR [7] are known as similar approaches adopted by the industry.

It is worth noting that real-time operation is essential in a virtual reality or metaverse environment. This means that SISR for CG images must pursue real-time inference. However, most deep learning-based SISR models have been developed so that they maximize restoration ability rather than speed. As a result, existing SISR models require not only a complex structure but also a significant amount of computation and parameter scale. That is, conventional SISR approaches generally have difficulty satisfying the real-time inference constraint. In order to maintain high performance while utilizing a small amount of computation and parameters, lightweight SR models such as IMDN [8], PAN [9], RFDN [10], and RLFN [11] were proposed. However, the above methods have a critical issue in that performance highly depends on the structural specification.

In order to achieve lightweighting independently of the SR model structure, attempts to apply universal lightweighting techniques such as quantization and knowledge distillation (KD) to SR models have recently been reported. For example, PAMS [12], DAQ [13], and FQSR [14] are cases in which quantization is grafted onto SR. However, quantization has a fundamental disadvantage of being dependent on specific hardware. Therefore, we have focused on applying KD (which is less dependent on hardware) to SR models. SRKD [15], PISR [16], JDSR [17], and LSFD [40] are the examples that apply KD to the SR domain. As an initial study case, SRKD used feature distillation to learn the feature distribution of the teacher model so that the student model resembles it. PISR, published after SRKD, used the high-frequency information of the HR image, which is the ground truth (GT), as privileged information to ensure that the teacher model had superior performance than the student model. However, this made the performance of the teacher model increase excessively, resulting in a phenomenon in which the performance gap between the teacher and student models increased, which rather limited the performance improvement.

In fact, [18], where KD is applied to the image classification task, mentioned a problem that may arise when the performance of the teacher model is excessively higher than that of the student model. First, the teacher model has complex knowledge, so even if the student model receives the teacher's knowledge, it does not have sufficient ability to imitate the teacher. In addition, as the knowledge matched by the student model approaches the hard target, the effect of KD is weakened, and as a result, the performance of the student model may not improve. To solve this problem, [18] proposed TAKD in which the teacher assistant (TA) model was embedded. Here, TA model has intermediate performance between the teacher and student models, but has less complex knowledge and soft target-type knowledge than the teacher model. By adding an intermediate stage to the knowledge transfer process from teacher to student, the student model can receive more usable knowledge from the TA. DGKD [19], which appeared after TAKD, adopted a strategy of densely guiding teacher knowledge and the knowledge of all TA models, rather than passing only the knowledge of TA, which is the intermediately preceding step, when transferring knowledge to the student model.

As mentioned above, we observed that the performance gap between the teacher and student models increased as the performance of the PISR teacher model increased excessively, and experimentally analyzed that this tendency rather limited the performance improvement. To solve this problem, we present a new KD-based SR model by introducing the TA model and dense guide strategy which were proposed in TAKD and DGKD, respectively. As far as we know, the proposed framework is the first attempt to improve performance by applying the TA concept to a KD-based SR model. In addition, we experimentally found from the PISR framework that the knowledge generated from the teacher model may not be of full help to the student model because the compact feature (CF), which is the input of the decoder of the teacher model, and the LR image, which is the input of the student model, have different distribution characteristics. To alleviate this problem, discretization is applied so that the CF has a discrete distribution like the LR image, and the input of the TA model is defined using the discretized CF.

Contributions of this paper are summarized as follows:

- In order to compensate for the large performance gap between the teacher and student models, a teacher assistant model specialized for the SR task is introduced. The teacher assistant model is positioned between the teacher and the student to help the teacher's knowledge transfer smoothly to the student.
- A distribution transformation process is added so that the components of the teacher model's compact features have the same discrete distribution as the LR image, which is the input of the student model. In the end, the student model utilizes the knowledge of the teacher model more effectively, resulting in improved SR performance.

The structure of this paper is as follows: Section II introduces the SISR model, lightweight SR model, KD techniques in the SR domain, and existing studies on KD using TA. Section III describes the details of the proposed framework. Section IV provides various experimental results and ablation studies.

II. RELATED WORKS

A. SINGLE IMAGE SUPER-RESOLUTION MODELS

In the meantime, most SISR studies have been conducted on natural images. SRCNN [1], which is an early study of deep learning-based SISR, achieved significant performance improvement compared to conventional SISR approaches by utilizing three convolution layers and ReLU. Since then, methods utilizing residual learning or deeper CNN models with increased number of layers have been developed, such as very deep super-resolution network (VDSR) [2] and enhanced deep super-resolution network (EDSR) [3]. SISR based attention mechanism like second-order attention network (SAN) [4] has also been reported. Most recently, SISR models using a vision transformer structure, such as image restoration using swin transformer (SwinIR) [20] or hybrid attention transformer (HAT) [21], have appeared.

B. LIGHTWEIGHT SUPER-RESOLUTION MODELS

On the other hand, so-called lightweight SR models, aiming to reduce the number of parameters and FLOPs while maintaining performance, is actively being studied [8], [9], [10], [11], [22], [23], [37], [38], [39]. For instance, FSRCNN [22] utilized deconvolution layers to achieve faster processing speed compared to SRCNN. Cascading residual network-mobile (CARN-M) [23], based on a cascading network architecture with techniques like group convolution, is known for being a lightweight SR model with high accuracy. Information multi-distillation network (IMDN) [8] employs a progressive refinement module to hierarchically extract features step by step. Pixel attention network (PAN) [9] is a model that utilizes an effective pixel attention scheme for restoration. Residual feature distillation network (RFDN) [10] refines the structure of IMDN. In other words, RFDN replaces IMDN's information distillation mechanism (IDM) with feature distillation connections and employs shallow residual blocks, achieving a good balance between lightweight cost compared to IMDN and high performance. Residual local feature network (RLFN) [11] simplifies the feature aggregation process by using three CNN models, thus enhancing the balance between the performance and inference time of the RFDN model.

Note that all the afore-mentioned techniques focused on natural images. As far as we know, RenderSR [24] is the only SISR algorithm targeting CG images. RenderSR is a lightweight SR model designed for upscaling rendered images in mobile game environments, consisting of three convolutional layers and a sharpening filter. RenderSR can be called a so-called bandwidth aware SR network that adjusts the number of channels of the model according to the buffer size of the layer, which is a challenging factor in the bandwidth of mobile SoC. However, as mentioned above, conventional lightweight SR models pursue weight reduction through sophisticated structural design.

C. KNOWLEDGE DISTILLATION IN SR DOMAIN

Knowledge distillation is a concept first proposed by Buciluǎ et al. [26] and Ba et al. [27] and then made famous by Hinton et al. [25]. Thanks to the advantage of being widely applied to various computer vision tasks such as image classification and semantic segmentation, KD has become the most active topic in the field of model compression. The goal of KD is to extract knowledge from a large, pre-trained teacher model and transfer it to a small, untrained student model so that the student model emulates the teacher model. So KD is a technique for condensing deeper and larger model knowledge into a single computationally efficient neural network.

Since Hinton et al.'s paper [25], KD has evolved into a method of transferring knowledge by using logit, which corresponds to predicted probability, as a soft target [18], [28], [29], or by using an intermediate feature of a model to directly transfer [32], [33], or by using correlation [30], [31].

The first study case applying KD to the SR domain is SRKD [15]. Among the afore-mentioned distillation methods, SRKD adopted a distillation strategy that directly transfers features. That is, by focusing on the feature distribution of the model, the student model mimics the features of the teacher model.

After SRKD, PISR [16] and JDSR [17] were developed. PISR uses the high-frequency components of the HR image as privileged information to generate a high-performance teacher model, and then transfers knowledge from the teacher model to the student model by measuring the feature difference between the teacher and student models in an embedding space with Gaussian or Laplacian distribution. However, as mentioned above, in the case of PISR, as the performance of the teacher model is considerably higher than that of the student, the knowledge of the teacher model is excessively complicated, and KD does not operate properly. JDSR improved the internal representation of the SR model through mutual learning and self-distillation using a peer model. Nevertheless, JDSR showed performance improvement similar to PISR. Therefore, we adopt the PISR framework as a benchmark KD-based SR technique.

D. TA-BASED KNOWLEDGE DISTILLATION

In order to compensate for the large performance gap between teacher and student models, a methodology for adding a teacher assistant (TA) model with medium size and medium performance has been developed in the field of image classification. TAKD [18] is the first introduction of the TA model. [18] presented the evidence that TA model helps improve performance and analyzed the effect of TA configuration on

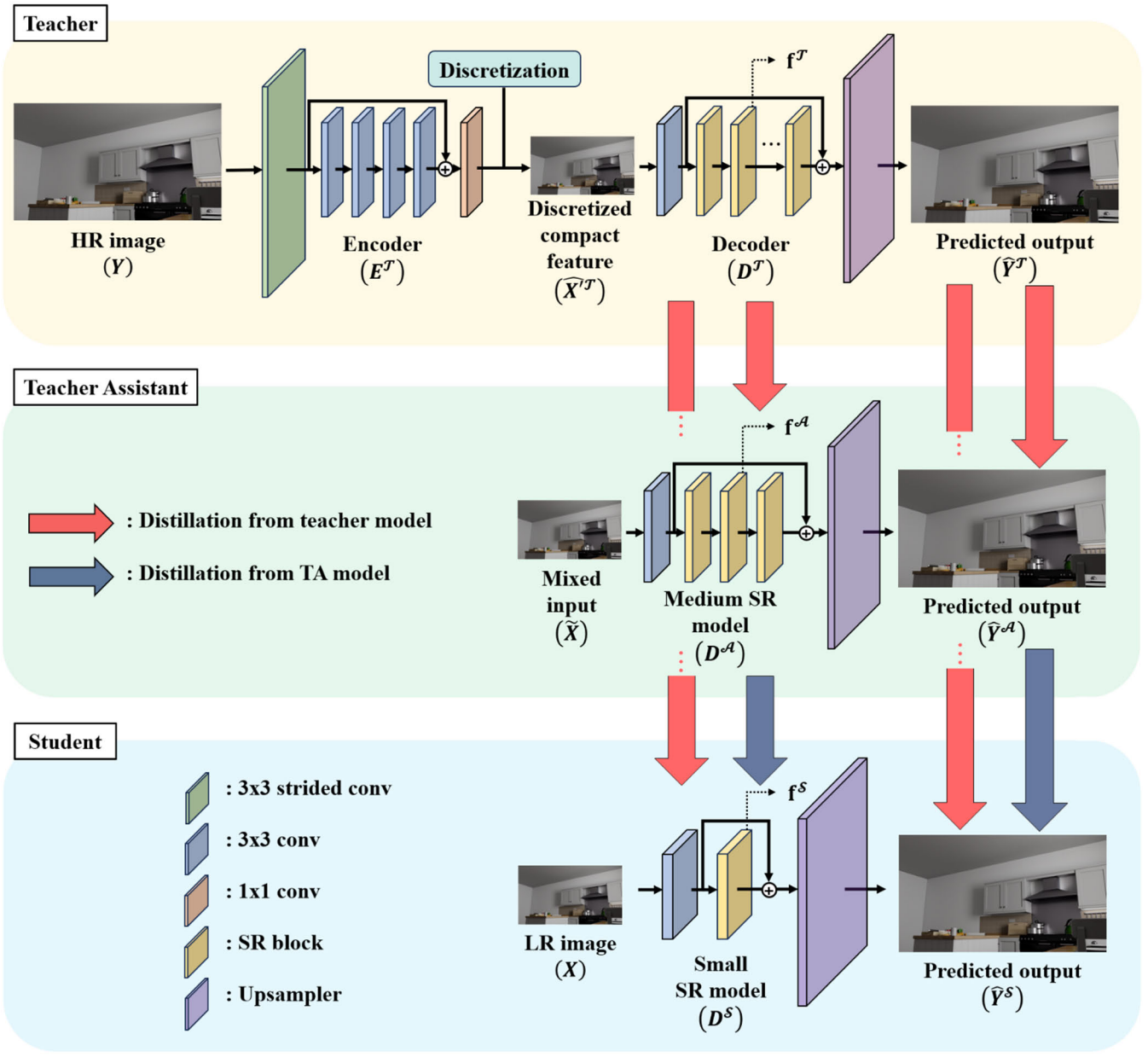

FIGURE 1. Overview of the proposed TAKDSR framework.

performance. Beyond TAKD, which had a single TA model due to memory limitations, DGKD [19] expanded the number of TA models to two or more. In addition, to solve the error avalanche problem of TAKD, they proposed a densely-guided method using dense connection when transferring the knowledge of the teacher and TA models to the student model, resulting in higher performance improvement.

III. PROPOSED METHOD

This paper tackles two critical problems in the process of applying KD to the SR task. First, PISR, which is the representative SOTA algorithm using KD, suffers from an excessively complex problem in order for the knowledge of the teacher model to properly propagate to the student model due to the large performance difference between the teacher model and the student model, as mentioned above. Second, the difference in distribution characteristics between the compact feature of the teacher model and the LR image of the student model also hinders the smooth transfer of knowledge.

As a solution to the first problem, we propose a framework that mitigates the performance gap between the two models by inserting a teacher assistant (TA) model between the teacher model and the student model. A solution to the second problem is the discretization of the CF of the teacher model. This process makes the difference in the characteristics of the

inputs of the two models mitigate, which eventually leads to smooth knowledge transfer.

Figure 1 shows the framework of knowledge distillation-based SR using TA (TAKDSR). The TAKDSR framework consists of teacher, TA and student models. The teacher model has an hourglass structure composed of an encoder and a decoder. Note that the CF of the teacher model is generated by the discretization process. In addition, the TA model and the student model have a structure in which only a decoder exists without an encoder and are designed to have an appropriately reduced number of blocks compared to the decoder of the teacher model. This allows the TA model to have the effect of performance control and induces a fast inference speed of the student model.

A. DETAILS OF THE PROPOSED TAKDSR

In the PISR framework, the teacher model receives the HR image ($\boldsymbol{Y}$) corresponding to GT as input so that the teacher model has superior performance than the student. Then, the CF $\left(\widehat{\boldsymbol{X}}^{\mathcal{T}}\right)$ is generated by an encoder $\left(E^{\mathcal{T}}\right)$. And the decoder ($D^{\mathcal{T}}$) receives the CF and outputs a super-resolved image $\left(\widehat{\boldsymbol{Y}}^{\mathcal{T}}\right)$. TAKDSR has two major differences from PISR. First, the TA model is added between the teacher and student models, and the second is to transform the distribution characteristics of CF. Figure 1 describes the proposed TAKDSR in detail.

TAKDSR's teacher model consists of an encoder $\left(E^{\mathcal{T}}\right)$ and a decoder ($D^{\mathcal{T}}$). $E^{\mathcal{T}}$ is a CNN model composed of strided convolutional layers with a stride of s and skip connections. Here, s means the scale factor of SR. As a $D^{\mathcal{T}}$, various backbone SR models can be employed. In this paper, PAN [9], RFDN [10], and RLFN [11] known as lightweight SR models are adopted as decoders. For example, RLFN was used as a decoder in Figure 1. Note that the so-called discretization is applied to CF $\left(\widehat{\boldsymbol{X}}^{\mathcal{T}}\right)$, i.e., the output of $E^{\mathcal{T}}$. Let the discretized CF be $\widehat{\boldsymbol{X}'}^{\mathcal{T}}$. The purpose of discretization is to ensure that the LR image and CF of the student model have similar characteristics. It is discussed in detail in Section III-B, and experimental evidence is given in Section IV.

The specific process of discretization is as follows: First, the range of the CF components is limited by applying the clipping operation to $\widehat{\boldsymbol{X}}^{\mathcal{T}}$. That is, by clipping to [0, 255], the CF components have the same range as the LR image. Then, as shown in Eq. (1), they are transformed into integers through rounding.

$$\widehat{\boldsymbol{X}'}^{\mathcal{T}} = \lfloor \langle \widehat{\boldsymbol{X}'}^{\mathcal{T}}, 0, 255 \rangle \rceil \tag{1}$$

where

$$\widehat{\boldsymbol{X}}^{\mathcal{T}} = E^{\mathcal{T}}(\boldsymbol{Y}) \tag{2}$$

$\langle\rangle$ means clipping operation, and $\lfloor \rceil$ means round operation. As a result, the final output $\widehat{\boldsymbol{Y}}^{\mathcal{T}}$ of the teacher model is generated as follows.

$$\widehat{\boldsymbol{Y}}^{\mathcal{T}} = D^{\mathcal{T}}\left(\widehat{\boldsymbol{X}'}^{\mathcal{T}}\right) \tag{3}$$

The TA model of TAKDSR is located between the teacher model and the student model as shown in Figure 1. It plays a role in ensuring that the teacher's knowledge is properly propagated to the student. Note that the TA model has mixed inputs $\left(\tilde{\boldsymbol{X}}\right)$. The mixed input was adjusted so that the performance of the TA model is close to the middle of the teacher and student models. This follows the analysis of [18], where TAKD was proposed. That is, [18] experimentally showed that the performance of the student model is best when a TA model with performance corresponding to the middle of the teacher and student models is given. We also obtained similar experimental results, which are given in the ablation study in Section IV. Specifically, the mixed input is a weighted sum of $\widehat{\boldsymbol{X}'}^{\mathcal{T}}$ generated in the learning process of the teacher model and the input $\boldsymbol{X}$ of the student model as in Eq. (4).

$$\tilde{\boldsymbol{X}} = \lambda_1 \cdot \widehat{\boldsymbol{X}'}^{\mathcal{T}} + \lambda_2 \cdot \boldsymbol{X} \tag{4}$$

where λ_1 and λ_2 are hyperparameters. Since they experimentally showed the best performance at 0.3 and 0.7, respectively, the mixed input set as such was used. The TA decoder mode has a medium size. That is, the number of blocks was reduced by half compared to the decoder of the teacher model. Here, the decoder model size was determined for the same reason as the mixed input design. Meanwhile, the number of TA networks in TAKDSR can be one or more, but in this paper, the number of TAs is fixed to one as shown in Figure 1. The related ablation study is given in Section IV.

Note that TAKDSR's student model is a small size model with the number of blocks limited to one. This enables very fast inference speed. Unlike the student model of PISR, the reason why the size of the student model can be drastically reduced is due to the structure using the TA model. We experimentally confirmed that reliable performance was achieved even when the number of blocks of the student model was reduced to one.

B. CONSIDERATION ON DISCRETIZED CF

When learning the teacher model in the PISR framework, CF ($\widehat{\boldsymbol{X}}^{\mathcal{T}}$) is used as the input of the decoder corresponding to the backbone SR model. CF means the result generated by passing the HR image ($\boldsymbol{Y}$) through the encoder ($E^{\mathcal{T}}$), and has the same size as the LR image ($\boldsymbol{X}$) but contains the high-frequency information of the HR image. Here, the components of CF are continuous values and have a corresponding distribution. On the other hand, the LR image, which is the input of the student model, has an inherently discrete distribution. In summary, the teacher and student models of PISR generate knowledge from inputs of continuous and discrete distributions, respectively. This mismatch can be a

factor in performance degradation when the knowledge generated from the teacher model is transferred to the student model.

To solve this problem, we adopted a strategy to discretize the CF of the teacher model as in Figure 1. That is, a CF having a discrete distribution identical to the LR image can be created. As a result, the student model can utilize the knowledge of the teacher model more effectively. This leads to improved performance of the student model. The performance with and without discretization is evaluated in the ablation study of Section IV.

C. LOSS FUNCTION DESIGN

The teacher model is learned by a reconstruction loss $\left(\mathcal{L}_{recon}^{\mathcal{T}}\right)$ and an imitation loss $\left(\mathcal{L}_{im}^{\mathcal{T}}\right)$, the same as the PISR framework. $\mathcal{L}_{recon}^{\mathcal{T}}$ is the pixel-wise L1 loss between the GT image and the SR image.

$$\mathcal{L}_{recon}^{\mathcal{T}} = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\left|Y_{ij} - \widehat{Y}_{ij}^{\mathcal{T}}\right| \tag{5}$$

Here, $\boldsymbol{H}$ and $\boldsymbol{W}$ mean the height and width of the HR image. Similarly, $\mathcal{L}_{im}^{\boldsymbol{T}}$ is defined as the pixel-wise L1 loss between LR image and discretized CF.

$$\mathcal{L}_{im}^{\mathcal{T}} = \frac{1}{H'W'}\sum_{i=1}^{H'}\sum_{j=1}^{W'}\left|X_{ij} - \widehat{X'}_{ij}^{\mathcal{T}}\right| \tag{6}$$

where $\boldsymbol{H'}$ and $\boldsymbol{W'}$ mean the height and width of the LR image. The total loss for learning the teacher model is defined as follows based on $\mathcal{L}_{recon}^{\mathcal{T}}$ and $\mathcal{L}_{im}^{\mathcal{T}}$.

$$\mathcal{L}_{total}^{\mathcal{T}} = L_{recon}^{\mathcal{T}} + \lambda^{\mathcal{T}} L_{im}^{\mathcal{T}} \tag{7}$$

where $\lambda^{\mathcal{T}}$ is a hyperparameter.

TA and student models are trained based on reconstruction loss ($\mathcal{L}_{recon}^{\mathcal{A}}$, $\mathcal{L}_{recon}^{\mathcal{S}}$) and distillation loss $\mathcal{L}_{distill}^{\mathcal{A}}$, $\mathcal{L}_{distill}^{\mathcal{S}}$). Unlike the teacher model, the reconstruction loss here is defined by considering not only the pixel-wise L1 loss between the GT image and the SR image, but also the pixel-wise L1 loss between the outputs of the higher-level model and the output of the current-level model. That is, $\mathcal{L}_{recon}^{\mathcal{A}}$ and $\mathcal{L}_{recon}^{\mathcal{S}}$ are each represented as follows.

$$\mathcal{L}_{recon}^{\mathcal{A}} = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\left(\left|Y_{ij} - \widehat{Y}_{ij}^{\mathcal{A}}\right| + \left|\widehat{Y}_{ij}^{\mathcal{T}} - \widehat{Y}_{ij}^{\mathcal{A}}\right|\right) \tag{8}$$

$$L_{recon}^{\mathcal{S}} = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\left(\left|Y_{ij} - \widehat{Y}_{ij}^{\mathcal{S}}\right| + \left|\widehat{Y}_{ij}^{\mathcal{T}} - \widehat{Y}_{ij}^{\mathcal{S}}\right| + \left|\widehat{Y}_{ij}^{\mathcal{A}} - \widehat{Y}_{ij}^{\mathcal{S}}\right|\right) \tag{9}$$

Distillation loss is defined as the pixel-wise L1 loss between the intermediate feature maps generated by the current-level model and the higher-level models.

$$\mathcal{L}_{distill}^{\mathcal{A}} = \frac{1}{CH'W'}\sum_{i=1}^{C}\sum_{j=1}^{H'}\sum_{k=1}^{W'}\left|f_{ijk}^{\mathcal{T}} - f_{ijk}^{\mathcal{A}}\right| \tag{10}$$

$$\mathcal{L}_{distill}^{\mathcal{S}} = \frac{1}{CH'W'}\sum_{i=1}^{C}\sum_{j=1}^{H'}\sum_{k=1}^{W'} \times \left(\left|\mathrm{f}_{ijk}^{\mathrm{T}} - \mathrm{f}_{ijk}^{\mathrm{S}}\right| + \left|\mathrm{f}_{ijk}^{\mathcal{A}} - \mathrm{f}_{ijk}^{\mathrm{S}}\right|\right) \tag{11}$$

where $\boldsymbol{C}$ means the number of channels of the intermediate feature map. Finally, the total loss for learning the TA model and the student model is defined as follows.

$$\mathcal{L}_{total}^{\mathcal{A}} = \mathcal{L}_{recon}^{\mathcal{A}} + \lambda^{\mathcal{A}}\mathcal{L}_{distill}^{\mathcal{A}} \tag{12}$$

$$\mathcal{L}_{total}^{\mathcal{S}} = \mathcal{L}_{recon}^{\mathcal{S}} + \lambda^{\mathcal{S}}\mathcal{L}_{distill}^{\mathcal{S}} \tag{13}$$

where $\lambda^{\mathcal{A}}$ and $\lambda^{\mathcal{S}}$ are hyperparameters.

IV. EXPERIMENTS

A. DATASETS

Unfortunately, there is no publicly available dataset for SISR of the CG images we target. So, we built a dataset by defining high-resolution CG images that are used for purposes different from ours as HR (GT) images. At this time, by defining the image obtained by down-sampling the HR image with a bicubic filter as an LR image, a CG image dataset for SISR in the form of an LR-HR pair was obtained.

Specifically, the rendered images of the Spring [34] dataset were used as a dataset for training and validation. The dataset consists of a total of 47 scenes and 6000 images. Of these, 37 scenes and 5000 images were adopted as the train dataset, and the remaining 10 scenes and 1000 images were used as the validation dataset. BMFR [35] and Tungsten [36] datasets were adopted as datasets for the test. The BMFR dataset consists of 6 scenes and 360 images, respectively, and the Tungsten dataset consists of 8 scenes and 750 images. In this paper, quantitative and qualitative results for each scene in the dataset are evaluated. As metrics for evaluation, peak-to-noise ratio (PSNR) and structural similarity index measure (SSIM), which are most widely used in SR tasks, were adopted.

B. IMPLEMENTATION DETAILS

First, TAKDSR's teacher model is learned based on Eq. (7). If the learning of the teacher model is completed, the learning of the TA model begins. Prior to learning the TA model, the parameters of the TA model are initialized with the previously learned parameters of the teacher model. This is to obtain a fast optimization effect during learning. TA model is learned based on Eq. (12). At this time, the output of the teacher model and the intermediate feature information of the decoder are transferred to the TA model as knowledge. If the training of the TA model is completed, the student model is finally trained. Similar to the TA model, the student model is initialized with the pre-learned parameters of the teacher model. The student model is trained based on Eq. (13). At this time, the knowledge generated from the teacher model and the TA model is transferred to the student model.

For training of each model, the number of epochs was set to 100, and the Adam optimizer was used. The learning rate was initially set at 10^{-3} and then reduced to 10^{-4} through cosine annealing.

Also, in the learning process, the HR image was randomly cropped with 192 × 192 patches, and then the LR image

TABLE 1. PSNR results for tungsten dataset.

Scale	Models	FLOPs [G]	Class room	Living room	Living room2	Living room3	Kitchen	Dining room	Stair case	Bed room	Avg.
2x	PAN_s	20.81	35.73	36.96	41.05	44.18	37.91	40.26	36.54	38.36	38.87
	PAN_s-PISR′	20.81	35.85	37.03	41.14	44.25	38.00	40.33	36.59	38.45	38.96
	PAN_s-TAKDSR	20.81	**36.07**	**37.25**	**41.28**	**44.54**	**38.21**	**40.49**	**36.61**	**38.59**	**39.13**
	RFDN_s	16.40	35.22	36.45	40.57	43.15	37.32	39.50	36.27	37.86	38.29
	RFDN_s-PISR′	16.40	35.35	36.57	40.67	43.23	37.45	39.54	36.37	37.91	38.39
	RFDN_s-TAKDSR	16.40	**35.68**	**36.91**	**40.96**	**44.06**	**37.79**	**39.99**	**36.47**	**38.28**	**38.77**
	RLFN_s	12.15	35.03	36.28	40.39	42.75	37.07	39.33	36.17	37.70	38.09
	RLFN_s-PISR′	12.15	35.14	36.33	40.46	42.94	37.18	39.41	36.31	37.80	38.20
	RLFN_s-TAKDSR	12.15	**35.39**	**36.58**	**40.70**	**43.39**	**37.46**	**39.75**	**36.37**	**37.97**	**38.45**
	PAN [9]	70.86	36.37	37.62	41.52	45.10	38.50	40.64	36.64	38.84	39.40
	RFDN [10]	91.55	36.16	37.37	41.34	44.81	38.28	40.50	36.58	38.64	39.21
	RLFN [11]	99.00	36.18	37.39	41.42	44.87	38.37	40.57	36.57	38.66	39.25
	HGSRCNN [5]	529.66	36.50	37.73	41.60	45.20	38.59	40.76	36.80	38.93	39.51

TABLE 2. SSIM results for tungsten dataset.

Scale	Models	FLOPs [G]	Class room	Living room	Living room2	Living room3	Kitchen	Dining room	Stair case	Bed room	Avg.
2x	PAN_s	20.81	0.672	0.636	0.616	0.648	0.600	0.723	0.721	0.689	0.663
	PAN_s-PISR′	20.81	0.674	0.636	0.619	0.648	0.601	0.726	0.721	0.689	0.664
	PAN_s-TAKDSR	20.81	**0.676**	**0.638**	**0.620**	**0.649**	**0.603**	**0.730**	**0.721**	**0.692**	**0.666**
	RFDN_s	16.40	0.665	0.628	0.605	0.624	0.589	0.716	0.724	0.679	0.654
	RFDN_s-PISR′	16.40	0.671	0.633	0.614	0.637	0.595	0.721	0.728	0.684	0.660
	RFDN_s-TAKDSR	16.40	**0.674**	**0.635**	**0.616**	**0.639**	**0.596**	**0.721**	**0.728**	**0.686**	**0.662**
	RLFN_s	12.15	0.665	0.631	0.604	0.619	0.587	0.714	0.721	0.675	0.652
	RLFN_s-PISR′	12.15	0.667	0.640	0.611	0.636	0.591	0.723	0.728	0.683	0.660
	RLFN_s-TAKDSR	12.15	**0.671**	**0.642**	**0.619**	**0.644**	**0.600**	**0.725**	**0.730**	**0.689**	**0.665**
	PAN [9]	70.86	0.684	0.647	0.628	0.654	0.612	0.728	0.726	0.703	0.673
	RFDN [10]	91.55	0.677	0.628	0.632	0.643	0.616	0.723	0.720	0.688	0.666
	RLFN [11]	99.00	0.680	0.643	0.620	0.648	0.606	0.733	0.733	0.693	0.670
	HGSRCNN [5]	529.66	0.692	0.651	0.643	0.665	0.674	0.732	0.733	0.773	0.695

TABLE 3. Performance comparison of teacher and student models.

Backbone SR models	PSNR / SSIM
PAN	T) 34.23 / 0.916
	S) 29.12 / 0.759
RFDN	T) 35.98 / 0.935
	S) 28.90 / 0.755
RLFN	T) 35.00 / 0.926
	S) 28.81 / 0.752

was cropped with a patch with a size of $192/s \times 192/s$ corresponding to the position to proceed with learning. Here, s corresponds to the scale factor of SR. For example, when s is 2, the patch size of the LR image becomes 96×96. For data augmentation, we used horizontal/vertical flipping, random rotation, and RGB alpha blending. The hyperparameters of loss functions were set as follows: $\lambda^{\mathcal{T}} = 10^{-2}$, $\lambda^{\mathcal{A}} = 10^{-4}$, and $\lambda^{\mathcal{S}} = 10^{-4}$. These hyperparameters are the values that show the best performance in grid search on the Spring dataset.

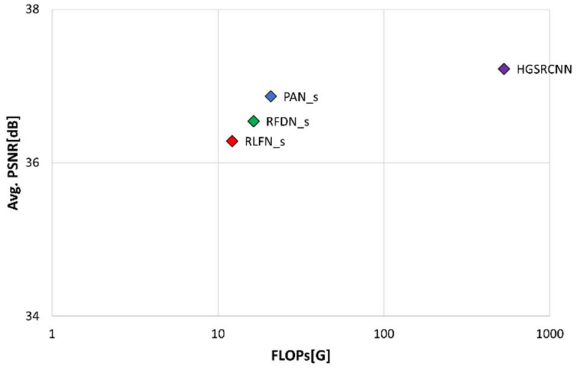

FIGURE 2. Avg. PSNR vs FLOPs plots of several student SR models with TAKDSR framework for BMFR and Tungsten datasets. The proposed framework shows similar performance to the high-performance SR model, i.e., HGSRCNN even with an SR model that has a much smaller amount of computation than HGSRCNN.

C. PERFORMANCE EVALUATION

First, we quantitatively evaluate the performance difference of the predicted output between the teacher and student

TABLE 4. PSNR results for BMFR dataset.

Scale	Models	FLOPs [G]	Classroom	Livingroom	SanMiguel	Sponza	Sponza-glossy	Sponza-movlight	Avg.
2x	PAN_s	20.81	35.55	40.76	29.12	31.82	33.36	31.21	33.64
	PAN_s-PISR′	20.81	35.64	40.88	29.18	31.93	33.43	31.26	33.72
	PAN_s-TAKDSR	20.81	**35.82**	**41.14**	**29.25**	**31.98**	**33.50**	**31.36**	**33.84**
	RFDN_s	16.40	34.98	39.64	28.90	31.73	33.13	31.03	33.24
	RFDN_s-PISR′	16.40	35.07	39.77	28.91	31.82	33.20	31.09	33.31
	RFDN_s-TAKDSR	16.40	**35.49**	**40.34**	**29.11**	**31.94**	**33.36**	**31.23**	**33.58**
	RLFN_s	12.15	34.83	39.44	28.81	31.70	33.04	30.97	33.13
	RLFN_s-PISR′	12.15	34.96	39.56	28.88	31.72	33.10	31.01	33.21
	RLFN_s-TAKDSR	12.15	**35.21**	**40.02**	**28.98**	**31.75**	**33.23**	**31.16**	**33.39**
	PAN [9]	70.86	36.15	41.68	29.35	32.01	33.60	31.47	34.04
	RFDN [10]	91.55	35.96	41.36	29.31	32.04	33.61	31.45	33.96
	RLFN [11]	99.00	35.96	41.40	29.30	32.03	33.62	31.44	33.96
	HGSRCNN [5]	529.66	36.36	41.76	29.42	32.19	33.75	31.56	34.17

TABLE 5. SSIM results for BMFR dataset.

Scale	Models	FLOPs [G]	Classroom	Livingroom	SanMiguel	Sponza	Sponza-glossy	Sponza-movlight	Avg.
2x	PAN_s	20.81	0.746	0.710	0.759	0.695	0.685	0.673	0.711
	PAN_s-PISR′	20.81	0.746	0.712	0.760	0.696	0.687	0.675	0.713
	PAN_s-TAKDSR	20.81	**0.751**	**0.716**	**0.762**	**0.698**	**0.688**	**0.677**	**0.715**
	RFDN_s	16.40	0.741	0.698	0.755	0.693	0.683	0.671	0.707
	RFDN_s-PISR′	16.40	0.741	0.700	0.756	0.693	0.684	0.674	0.708
	RFDN_s-TAKDSR	16.40	**0.746**	**0.701**	**0.759**	**0.696**	**0.686**	**0.675**	**0.711**
	RLFN_s	12.15	0.738	0.696	0.752	0.691	0.682	0.670	0.705
	RLFN_s-PISR′	12.15	0.738	0.698	0.753	0.692	0.682	0.670	0.706
	RLFN_s-TAKDSR	12.15	**0.742**	**0.707**	**0.755**	**0.693**	**0.684**	**0.671**	**0.709**
	PAN [9]	70.86	0.756	0.722	0.765	0.702	0.692	0.682	0.720
	RFDN [10]	91.55	0.753	0.710	0.763	0.702	0.691	0.681	0.717
	RLFN [11]	99.00	0.754	0.708	0.764	0.703	0.692	0.681	0.717
	HGSRCNN [5]	529.66	0.762	0.731	0.776	0.714	0.705	0.692	0.730

TABLE 6. Average inference time of each student model. All measurements were made on an NVIDIA Quadro RTX 8000.

Student models	Avg. inference time (ms)
RLFN_s	5.8
RFDN_s	8.4
PAN_s	12.1
HGSRCNN	93.7

models through the actual test dataset. Table 3 shows the quantitative performance between the teacher and student models of each backbone SR model. This results from the SanMiguel scene of the BMFR dataset, and shows the performance when KD is not applied. Table 3 showed that a large performance difference of about 5-7dB actually occurs between the teacher and student models. Tables 1-2 and Tables 4-5 show the quantitative performance for the BMFR and Tungsten dataset, respectively. Here, PISR' means the case where the TA model is excluded from the proposed TAKDSR and the discretized CF is not used. Note that the TAKDSR framework is implemented with the PISR model as a reference. Also, '_s' in the model name means that it is a small size student model and is the final target model.

Looking at Tables 1-2 and 4-5, when the PISR' framework was applied, the performance improved the most was the 'livingroom3' scene of RLFN_s. At this time, PSNR and SSIM were improved by 0.19dB and 0.017, respectively, compared to the baseline. This is just a marginal improvement. On the other hand, the improvement by TAKDSR was much larger. In the same scene, the PSNR of RFDN_s is improved by 0.91dB by TAKDSR, and the SSIM of RLFN_s is increased by 0.025. This proves that when the TAKDSR framework is applied to SR models, it demonstrates superior knowledge transfer capability than PISR′.

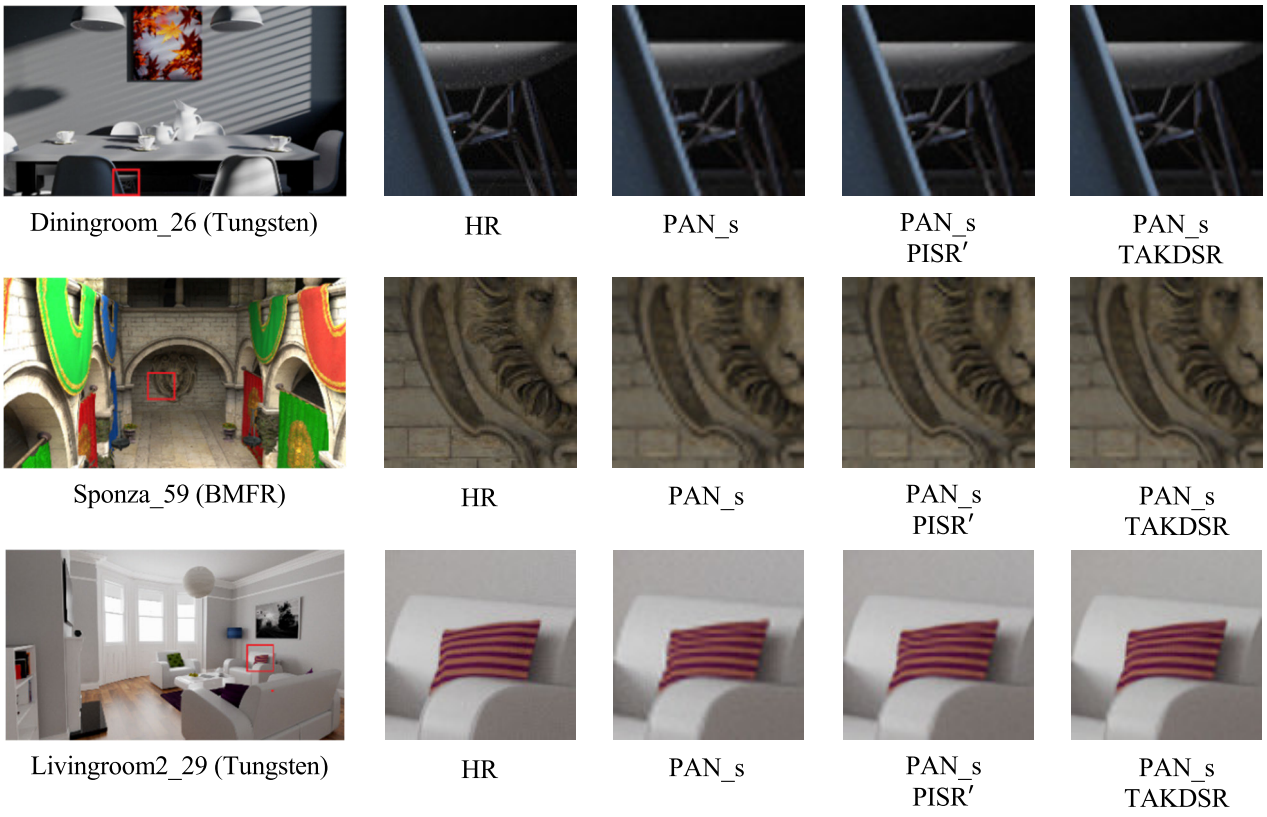

FIGURE 3. Qualitative evaluation of PAN_s model.

TAKDSR achieves a good tradeoff between computation and performance. Figure 2 plots FLOPs vs. average PSNR. For relative comparison, HGSRCNN, one of the high-performance SR models, is also shown. We can observe that while FLOPs are reduced by up to 44 times, PSNR degradation is not significant compared to HGSRCNN.

The actual inference time of TAKDSR is also significantly smaller than that of HGSRCNN. Table 6 shows the average inference time per frame for each student model. This experiment was performed with the Tungsten dataset. As a result, SR models applied with TAKDSR are 8 to 16 times faster than HGSRCNN while providing reasonable visual quality, and their inference time itself is absolutely small, so they can be a suitable solution for CG applications that require real-time operation.

Finally, the qualitative results for each backbone SR model can be found in Figures 3-5. Qualitatively, it is demonstrated that TAKDSR provides excellent visual quality. Specifically, when KD is not applied, a lot of artifacts are generally observed in the output images. In addition, scenes such as BMFR's classroom or Tungsten's livingroom2 and staircase show a phenomenon in which certain patterns are distorted. The PISR' framework shows less artifacts than this case when KD is not applied. However, the degree of improvement is weak, and the distortion of the pattern is not well improved. On the other hand, we can observe that TAKDSR not only significantly reduces artifacts but also significantly improves pattern distortion.

TABLE 7. PSNR of student model according to TA performance.

Student model	$\lambda_1 : \lambda_2$	PSNR of TA model	PSNR of the student model
RFDN_s	0.5 : 0.5	33.72	29.01
	0.3 : 0.7	31.85	**29.11**
	0.1 : 0.9	29.91	29.03

D. ABLATION STUDY

First, by analyzing the performance of the student model according to the TA performance targeting the RFDN model, the configuration of the best TA for the TAKDSR framework is determined. When configuring the mixed input, the performance of the TA model was adjusted by the ratio between discretized CF and LR images, that is, λ_1 and λ_2. Table 7 shows the performance of the student model according to the performance of the TA model. For this experiment, the SanMiguel scene from the BMFR dataset was used.

We can find that the PSNR of the student model is the best when λ_1 and λ_2 are 0.3 and 0.7. At this time, referring to Table 3, the PSNR of the TA model is closest to the median of the teacher and student models. On the other hand, if the

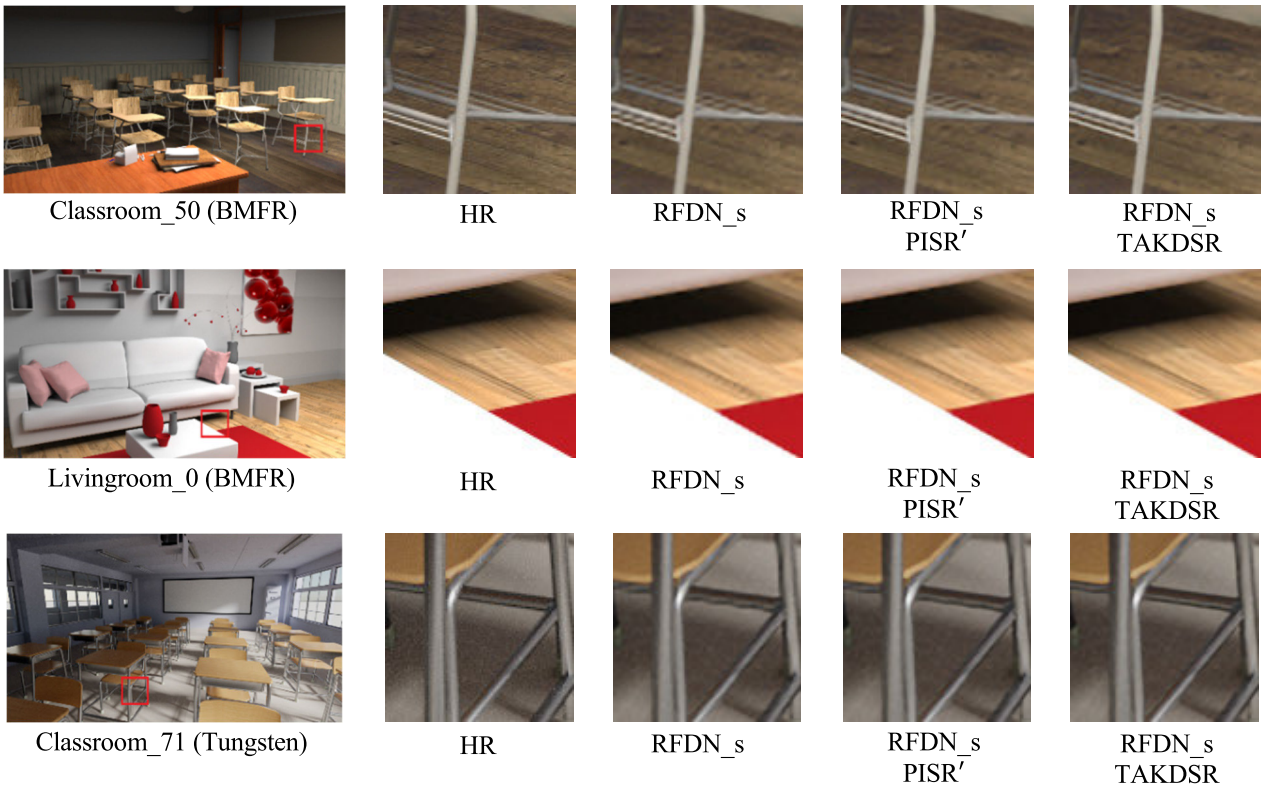

FIGURE 4. Qualitative evaluation of RLDN_s model.

TABLE 8. PSNR of the student model according to the number of TAs.

Student model	# of TA	$\lambda_1 : \lambda_2$	PSNR of TA models	PSNR of the student model
RLFN_s	2	0.6 : 0.4	33.19	**29.08**
		0.2 : 0.8	30.77	
	1	0.3 : 0.7	31.45	28.98

TABLE 9. Performance comparison of student models before and after applying CF discretization.

Student models	Distribution of CF values	PSNR/SSIM
RLFN_s	Discrete	38.10 / 0.688
	Continuous	37.97 / 0.687
RFDN_s	Discrete	38.28 / 0.686
	Continuous	38.11 / 0.684

PSNR of the TA model is close to that of the teacher model, as in the case of $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$, or if the PSNR of the TA model is close to that of the student model, as in the case of $\lambda_1 = 0.1$ and $\lambda_2 = 0.9$, the performance of the student model is slightly lowered. Therefore, similar to [18], the TA model with performance corresponding to the median of the teacher and student models can be said to be the best TA that makes the best student model.

Next, let's look at the performance of the student model according to the number of TA models when constructing the framework. In this experiment, the RLFN model was used. As mentioned in Section III, the TAKDSR framework can be composed of multiple TAs as well as a single TA. Table 8 compares the performance of the student model according to the number of TAs. In this experiment, the SanMiguel scene from the BMFR dataset was used. Since there was no significant difference between the case where the number of TAs was 3 or more and the case where the number of TAs was 2, only the case of 2 is dealt with here. In addition, referring to the experimental result in Table 7, this experiment was conducted by setting appropriate λ_1 and λ_2. According to Table 8, multiple TA improves PSNR by about 0.1dB more than single TA. However, an increase in the number of TAs inevitably causes an increase in the time required for learning. This means that the amount of resources used for learning increases. Therefore, since there is little justification for adopting multiple TAs, the number of TAs is fixed at 1 in this paper.

Finally, to verify the effect of discretization, the performance of the student model was compared when CF had a continuous distribution and a discrete distribution, respectively. For this experiment, the RLFN and RFDN models were used, and the bedroom scene of the Tungsten dataset was adopted. Table 9 shows that the discrete CF distribution improves the PSNR by 0.13∼0.16 dB over the continuous CF distribution. Therefore, in order to use the knowledge of the teacher model more effectively, it is necessary to add

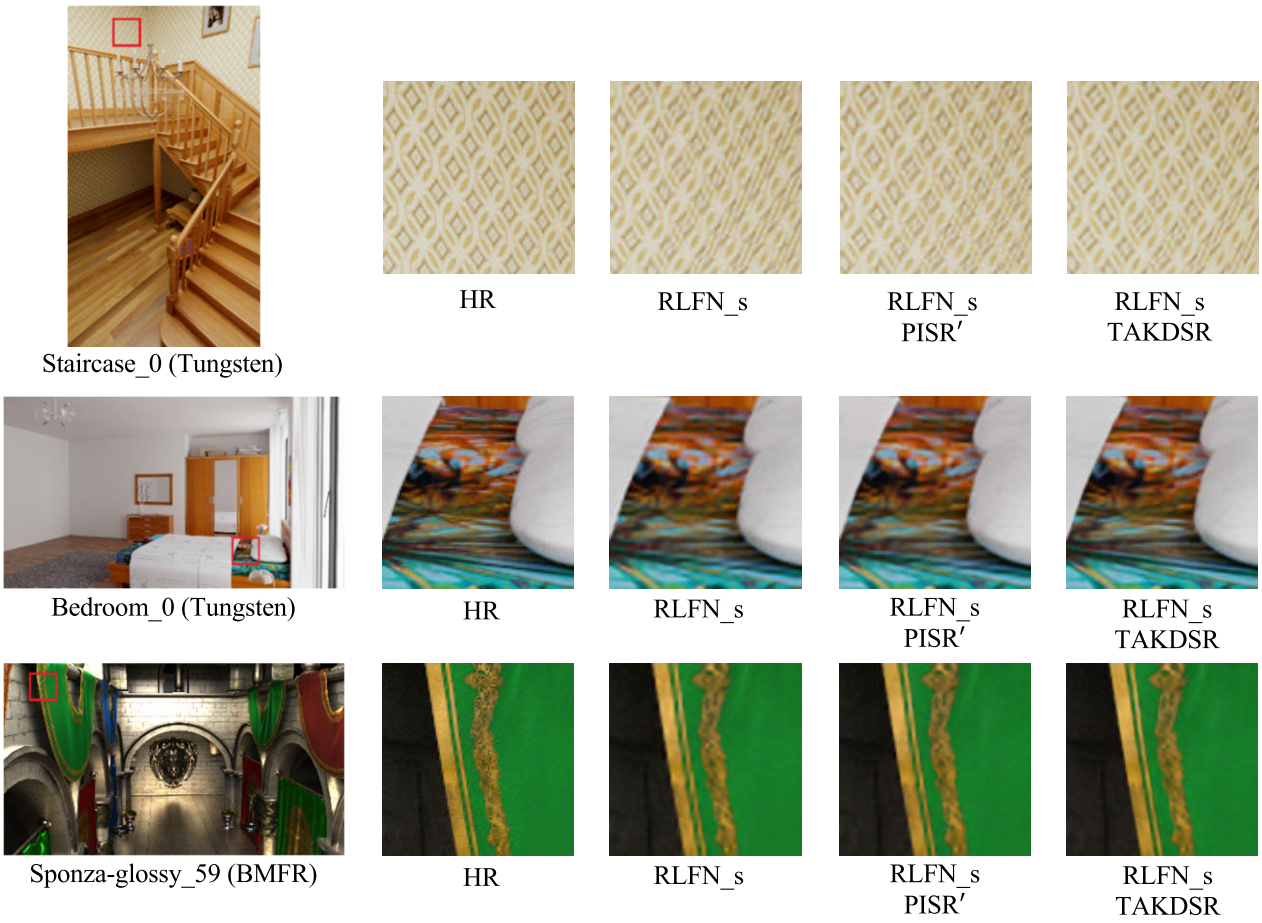

FIGURE 5. Qualitative evaluation of RLFN_s model.

discretization so that the components of the CF can have the same discrete distribution as the LR image.

V. LIMITATION

As mentioned earlier in the dataset section of Section IV, there is no publicly available dataset composed of CG images in the SISR field. The experiments in this paper also used CG image datasets, which are used in other tasks other than SR, for the experiments, and the amount of the datasets is rather insufficient. Therefore, if more certified CG image datasets for SISR are proposed, better algorithm design will be possible.

In addition, when designing the distillation loss of the TAKDSR framework, a simple comparison of the intermediate features of the two models with pixel-wise L1 loss was adopted. However, there is still the potential for more effective KD based on new metrics. We need to discuss this. This will be our further work.

VI. CONCLUSION

In this paper, a new framework that applies knowledge distillation to a lightweight single image super-resolution model for real-time inference is proposed. In order to alleviate the problem of inconsistent knowledge transfer from teacher to student model, a teacher assistant model specialized for SR was introduced. In addition, a discretized CF is created so that the student model can effectively utilize the knowledge generated by the teacher model. Experimental results prove the outstanding cost performance of the proposed method. Also, the proposed method is expected to be applicable in various industries that require real-time SR inference of graphics images. In particular, thanks to the drastically small amount of computation of the student model, it can be used in low-power devices such as mobile devices.

REFERENCES

[1] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 184–199.
[2] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1646–1654.
[3] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1132–1140.
[4] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11057–11066.

[5] C. Tian, Y. Zhang, W. Zuo, C.-W. Lin, D. Zhang, and Y. Yuan, "A heterogeneous group CNN for image super-resolution," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 13, 2022, doi: 10.1109/TNNLS.2022.3210433.
[6] (Mar. 23, 2020). NVIDIA. *NVIDIA DLSS 2.0: A Big Leap in AI Rendering*. [Online]. Available: https://www.nvidia.com/engb/geforce/news/nvidia-dlss-2-0-a-big-leap-in-ai-rendering/
[7] (2020). AMD. *AMD FidelityFX*. [Online]. Available: https://www.amd.com/en/technologies/radeon-software-fidelityfx
[8] Z. Hui, X. Gao, Y. Yang, and X. Wang, "Lightweight image super-resolution with information multi-distillation network," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 2024–2032.
[9] H. Zhao, X. Kong, J. He, Y. Qiao, and C. Dong, "Efficient image super-resolution using pixel attention," in *Proc. Eur. Conf. Comput. Vis. Workshops (ECCVW)*, Aug. 2020, pp. 56–72.
[10] J. Liu, J. Tang, and G. Wu, "Residual feature distillation network for lightweight image super-resolution," in *Proc. Eur. Conf. Comput. Vis. Workshops (ECCVW)*, Aug. 2020, pp. 41–55.
[11] F. Kong, M. Li, S. Liu, D. Liu, J. He, Y. Bai, F. Chen, and L. Fu, "Residual local feature network for efficient super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 765–775.
[12] H. Li, C. Yan, S. Lin, X. Zheng, B. Zhang, F. Yang, and R. Ji, "PAMS: Quantized super-resolution via parameterized max scale," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 564–580.
[13] C. Hong, H. Kim, S. Baik, J. Oh, and K. M. Lee, "DAQ: Channel-wise distribution-aware quantization for deep image super-resolution networks," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 913–922.
[14] H. Wang, P. Chen, B. Zhuang, and C. Shen, "Fully quantized image super-resolution networks," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 639–647.
[15] Q. Gao, Y. Zhao, G. Li, and T. Tong, "Image super-resolution using knowledge distillation," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, Dec. 2020, pp. 527–541.
[16] W. Lee, J. Lee, D. Kim, and B. Ham, "Learning with privileged information for efficient image super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 465–482.
[17] X. Luo, Q. Liang, D. Liu, and Y. Qu, "Boosting lightweight single image super-resolution via joint-distillation," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 1535–1543.
[18] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2020, vol. 34, no. 4, pp. 5191–5198.
[19] W. Son, J. Na, J. Choi, and W. Hwang, "Densely guided knowledge distillation using multiple teacher assistants," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9375–9384.
[20] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "SwinIR: Image restoration using Swin transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 1833–1844.
[21] X. Chen, X. Wang, J. Zhou, Y. Qiao, and C. Dong, "Activating more pixels in image super-resolution transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22367–22377.
[22] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 391–407.
[23] N. Ahn, B. Kang, and K. A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 252–268.
[24] T. Dong, H. Yan, M. Parasar, and R. Krisch, "RenderSR: A lightweight super-resolution model for mobile gaming upscaling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 3086–3094.
[25] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.
[26] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2006, pp. 535–541.
[27] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2014, pp. 2654–2662.
[28] C. Yang, L. Xie, C. Su, and A. L. Yuille, "Snapshot distillation: Teacher–student optimization in one generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2854–2863.
[29] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4320–4328.
[30] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," 2019, *arXiv:1910.10699*.
[31] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3962–3971.
[32] A. Romero, N. Ballas, S. Ebrahimi Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," 2014, *arXiv:1412.6550*.
[33] B. Peng, X. Jin, D. Li, S. Zhou, Y. Wu, J. Liu, Z. Zhang, and Y. Liu, "Correlation congruence for knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5006–5015.
[34] L. Mehl, J. Schmalfuss, A. Jahedi, Y. Nalivayko, and A. Bruhn, "Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 4981–4991.
[35] M. Koskela, K. Immonen, M. Mäkitalo, A. Foi, T. Viitanen, P. Jääskeläinen, H. Kultala, and J. Takala, "Blockwise multi-order feature regression for real-time path-tracing reconstruction," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–14, Oct. 2019.
[36] X. Meng, Q. Zheng, A. Varshney, G. Singh, and M. Zwicker, "Real-time Monte Carlo denoising with the neural bilateral grid," in *Proc. EGSR* , Jun. 2020, pp. 13–24.
[37] Y. Jiang, Y. Liu, W. Zhan, and D. Zhu, "Lightweight dual-stream residual network for single image super-resolution," *IEEE Access*, vol. 9, pp. 129890–129901, 2021.
[38] Y. Yan, X. Xu, W. Chen, and X. Peng, "Lightweight attended multi-scale residual network for single image super-resolution," *IEEE Access*, vol. 9, pp. 52202–52212, 2021.
[39] Z. Wang, Y. Liu, R. Zhu, W. Yang, and Q. Liao, "Lightweight single image super-resolution with similar feature fusion block," *IEEE Access*, vol. 10, pp. 30974–30981, 2022.
[40] S. Park and N. Kwak, "Local-selective feature distillation for single image super-resolution," 2021, *arXiv:2111.10988*.

MIN YOON received the B.S. degree in electronic engineering from Inha University, Incheon, South Korea, in 2022, where he is currently pursuing the M.S. degree in electrical and computer engineering. His research interests include computer vision and deep learning.

SEUNGHYUN LEE (Associate Member, IEEE) received the B.S. and Ph.D. degrees in electronic engineering from Inha University, Incheon, South Korea, in 2017 and 2023, respectively. His research interests include computer vision and machine learning.

BYUNG CHEOL SONG (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1994, 1996, and 2001, respectively. From 2001 to 2008, he was a Senior Engineer with the Digital Media Research and Development Center, Samsung Electronics Company Ltd., Suwon, South Korea. In 2008, he joined the Department of Electronic Engineering, Inha University, Incheon, South Korea, where he is currently a Professor. His research interests include image processing and computer vision.