Pattern Recognition Letters 166 (2023) 97–104

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

SIT-SR 3D: Self-supervised slice interpolation via transfer learning for 3D volume super-resolution

Muhammad Sarmad[a,*], Leonardo Carlos Ruspini[b], Frank Lindseth[a]

[a] Department of Computer Science, NTNU, Trondheim, Norway
[b] Petricore, Norway

ARTICLE INFO

Article history:
Received 18 December 2021
Revised 29 November 2022
Accepted 10 January 2023
Available online 12 January 2023

Edited by: Jiwen Lu

Keywords:
Super-resolution
Digital rock analysis
Self-supervised learning

ABSTRACT

We present SIT-SR 3D, a novel self-supervised method for 3D single image super-resolution (SISR). Scaling 2D SISR networks to 3D SISR requires code redesign, high computing resources, and 3D ground-truth. However, we circumvent this by (1) using a pre-trained 2D SISR for indirect supervision and (2) using a novel consistency loss to learn frame interpolation. Any pre-trained state of the art 2D SISR method can replace the 2D SISR used in SIT-SR 3D, thus transferring the merits of 2D to 3D and ensuring modularity. We trained two end-to-end 3D baselines in a supervised setting; a 3D RRDBNet trained only with L1 loss and a 3D ESRGAN trained with adversarial and perceptual loss. We show that the proposed pipeline's self-supervised version is qualitatively better than the baselines. When trained in a supervised setting, SIT-SR 3D achieves better PSNR than its counterparts. Furthermore, our pipeline uses fewer parameters compared to the baselines. We demonstrate our results on an open-source digital rock CT dataset. Our code and pre-trained models will be made publicly available.

1. Introduction

The future depends on our capacity to minimize the emissions of CO2 in the atmosphere while keeping the economy's wheel in motion to reduce poverty and improve the quality of life in developing countries. Upcoming technologies such as Carbon dioxide Capture and Storage (CCS) and more efficient oil and gas production will play a significant role in achieving carbon neutrality. The derivation of rock properties from high-resolution CT images (Digital Rock) is a disruptive technology in that it can fundamentally change the way we characterize rocks. High-resolution photos can help characterize the properties of rocks and minerals, such as porosity, permeability, and flow [1]. Often it is beneficial to enhance or super-resolve a 3D image before usage in other domains, e.g., medical CT [2,3].

Super-resolving a 3D image presents unique challenges compared to a 2D image because of three prominent reasons. The first reason is the lack of high and low-resolution ground-truth image pairs in 3D. It is often costly and time-consuming to obtain such images. However, recent advances in deep learning [4,5] and single image super-resolution (SISR) [6] rely heavily on such paired data, which is not always available. While 2D image pairs are available in abundance, they cannot be used directly to train 3D pipelines.

Secondly, the advances in 2D image super-resolution techniques are not always scalable to 3D images. e.g., a 3D equivalent of perceptual loss does not exist to the best of our knowledge [7]. Similarly, a 3D version of SRGAN requires a 3D convolutional discriminator, which adds many training parameters [8]. Some recent advances in 2D SISR have custom operators and layers [9] that are not easily adaptable to the 3D domain and require low-level code redesign. Similarly, transfer learning from well-known architectures, e.g., ResNet50 [10] is not possible as the state-of-the-art encoders consist of 2D convolution layers. The only option to train 3D variants of deep networks on 3D data is to design from scratch.

Thirdly, if we design the 3D variants of novel operators from the 2D domain by writing custom code, the training cost increases cubically with the resolution. For example, the cost of training increases when enhancing the input size of the 3D convolutional discriminator in an adversarial setting. Some of the custom loss functions, such as perceptual loss, need to be applied to individual 2D slices of the 3D image and cause additional computation and latency overhead. To overcome these challenges, we propose an architecture that transfers the benefits of advances in 2D SISR to 3D SISR without engineering a 3D version of 2D SISR. We also do not require any paired 3D ground-truth data.

* Corresponding author.
E-mail addresses: muhammad.sarmad@ntnu.no (M. Sarmad), Leonardo.ruspini@petricore.com (L.C. Ruspini), frankl@ntnu.no (F. Lindseth).

https://doi.org/10.1016/j.patrec.2023.01.008

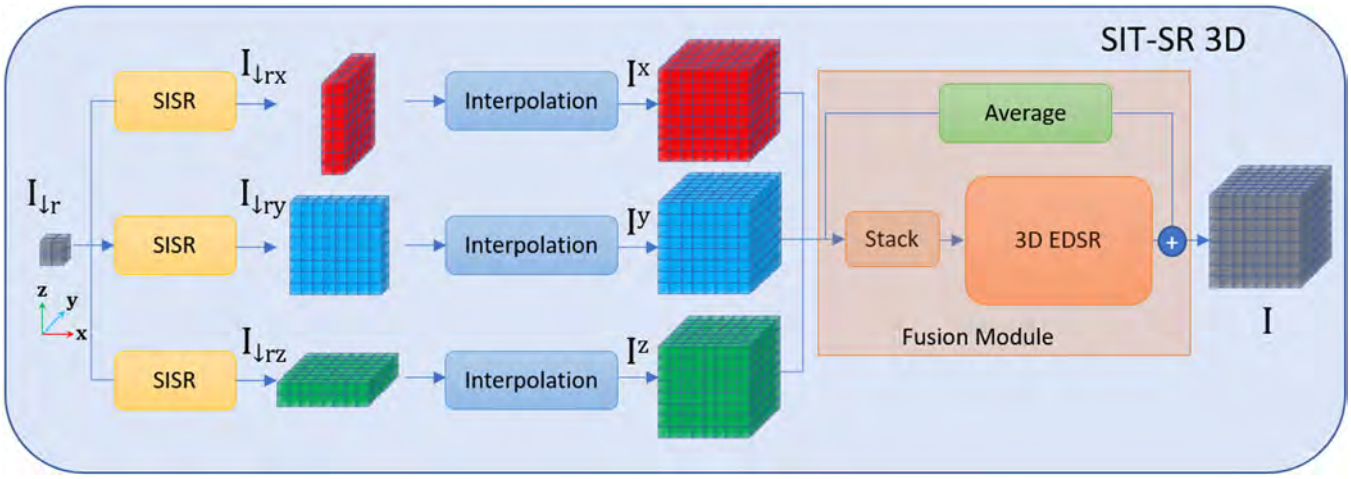

Fig. 1. SIT-SR 3D: The architecture of our proposed method. The low-resolution image $I_{\downarrow r}$ is upsampled along *x*, *y* and *z* respectively with the pre-trained 2D SISR, as a result, the volumes $I_{\downarrow r_x}$, $I_{\downarrow r_y}$ and $I_{\downarrow r_z}$ are obtained. I^x, I^y, and I^z are the corresponding volumes obtained after the interpolation operation. I^x, I^y, and I^z are passed through the Fusion module to get the output volume I.

In this work, we present SIT-SR 3D, a novel self-supervised 3D volume super-resolution technique. Our architecture is shown in Fig. 1. We employ transfer learning by utilizing a 2D SISR model trained on 2D feature-rich networks. We use this pre-trained 2D SISR to train the 3D SISR model on low-resolution 3D data only. We super-resolve the 3D image using this 2D SISR model along each of the three possible dimensions. We obtain the final volume by merging the three images using a Fusion module. This slice interpolation network's weights can be learned in an entirely self-supervised manner or with high-resolution ground-truth, if available. We introduce a consistency loss to train our method in a self-supervised manner. Our pipeline's self-supervised version compares well to the supervised baseline. We perform ablation studies and also compare the qualitative and quantitative results with multiple baselines. Our key contributions are the following:

- We present SIT-SR 3D, a novel self-supervised interpolation and transfer learning framework for 3D volume super-resolution.
- Our method can use any pre-trained 2D SISR model with desired qualities to transfer the merits of 2D SISR to the 3D pipeline.
- We propose a novel consistency loss for training SIT-SR 3D without 3D ground-truth.
- The approach is data-efficient, uses fewer parameters. Moreover, training converges fast and does not require 3D high-resolution ground truth.

2. Related works

2.1. Super resolution

The advent of convolutional neural networks (CNNs) [4] led to applications of deep learning for various computer vision problems. Single image super-resolution (SISR) is one such task that has benefited from the progress in deep learning. Initially, Dong et al. [11] proposed to perform end-to-end SISR using SR-CNN. They established state of the art by outperforming conventional methods such as the sparse coding-based method by Yang et al. [12]. Kim et al. [13] added skip connections and designed a lightweight recursive CNN architecture. Generative adversarial networks [14] can produce photo-realistic images but are hard to train due to the adversarial nature of training. Ledig et al. [15] utilized adversarial training in conjunction with the content loss to obtain photo-realistic SISR. Lim et al. [16] improved a ResNet based architecture [10] by removing batch normalization layers and introduced a multi-scale architecture to further enhance performance. Zhang et al. [17] applied an improved DenseNet [18] for SISR by removing batch normalization layers, pooling layers, and introduced a global feature fusion.

Various super-resolution quality measures have been developed, such as peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). There is no single metric that is considered better than the other, and both have their merits as shown in various work [19,20]. Both of these metrics have their shortcomings since they do not model human perception. Therefore, some works have employed human-based perceptual evaluation [15]. However, such measures are costly and time-consuming to obtain. We can extend state of the art 2D SISR methods for 3D images with significant customized code changes. However, we propose to utilize 2D SISR without extending them to 3D.

2.2. 3D super resolution

Previous works which super resolve 3D data are of two distinct types. The first one treats the entire *3D image as a collection of 2D slices* and then performs SISR on individual slices. Therefore, we can utilize any traditional 2D SISR methods applicable to colored images. However, such methods can only super-resolve a 3D image along two dimensions, and the third dimension is still of lower resolution. 3D images contain contextual information in all dimensions, and we must consider all three dimensions to upsample such images. There are many examples of using CT data to train 2D SISR networks for super resolving 2D slices of a 3D image in both the medical and digital rock domain [21–23].

The other type trains *end-to-end 3D networks* to super-resolve volumes. These methods are more challenging to design; hence only a few works exist, but they provide a complete solution [2,3]. Chen et al. [2] proposed mDSCRN for 3D volume super-resolution inspired by DenseNet [18]. Peng et al. [3] proposed SAINT and demonstrated that mDCSRN suffers from sub-optimal results and also has a higher memory and compute footprint. However, their method needs ground truth high-resolution data for supervised training. The approach has been applied to medical CT data that requires upsampling in one dimension. Therefore, they perform a frame interpolation method. However, our data requires upsam-

pling in all three dimensions. They do not provide any code for comparison.

3. Method

3.1. Problem formulation

The proposed work provides a solution to 3D volumetric super-resolution. We demonstrate our method on 3D CT Images. Consider a 3D image $I \rightarrow I(x, y, z) \in R^{X \times Y \times Z}$ which represents a densely sampled CT image. For I, the corresponding sparsely sampled volume $I_{\downarrow r}$ is defined as:

$$I_{\downarrow r} = I(r \cdot x, r \cdot y, r \cdot z) \tag{1}$$

where $I_{\downarrow r} \in R^{\frac{X \times Y \times Z}{r}}$, and r is the sparsity factor along the x, y and z axis from I to $I_{\downarrow r}$ and the up-sampling factor from $I_{\downarrow r}$ to I.

Along each axis, there can be three kinds of slices which are referred to as follows:

- The down sampled slices along x-axis are given as: $I_{\downarrow r_x} = I(r \cdot x, y, z), \sim \forall x$. Interpolated version of $I_{\downarrow r_x}$ is given as I^x.
- The down sampled slices along y-axis are given as: $I_{\downarrow r_y} = I(x, r \cdot y, z), \sim \forall y$. Interpolated version of $I_{\downarrow r_y}$ is given as I^y.
- The down sampled slice along z-axis are given as: $I_{\downarrow r_z} = I(x, y, r \cdot z), \sim \forall z$. Interpolated version of $I_{\downarrow r_z}$ is given as I^z.

The *objective of SIT-SR 3D* is to find a mapping $\mathcal{F} : R^{\frac{X \times Y \times Z}{r}} \rightarrow R^{X \times Y \times Z}$ that can convert $I_{\downarrow r}$ back to I for a given resolution factor r.

3.2. Overview of the proposed approach

Fig. 1: shows the overview of SIT-SR 3D. The low-resolution image $I_{\downarrow r}$ is upsampled along x, y, and z respectively, using a pre-trained SISR network trained on 2D images. After this operation, we obtain $I_{\downarrow r_x}$, $I_{\downarrow r_y}$ and $I_{\downarrow r_z}$. These anisotropic 3D volumes are converted to isotropic volumes using an interpolation operation. After the interpolation, we obtain the volumes I^x, I^y, and I^z, which are stacked to form a single volume with three channels. We then process this volume by a 3D convolution-based 3D EDSR [16]. The average of I^x, I^y, and I^z is also added to the output of the 3D EDSR to form I. We can supervise the output of 3D EDSR using the high-resolution ground-truth I_{GT} with L_1 loss to learn how to combine I^x, I^y and I^z. We can also train it in a self-supervised manner using I^x, I^y and I^z and a novel consistency loss. One can also combine both loss functions to obtain a hybrid loss formulation that provides control over the network's output. We demonstrate results with all three loss function settings. Next, we will describe each of the modules in SIT-SR 3D in detail.

4. Single image super-resolution

This module's objective is to super-resolve 3D images efficiently using transfer learning based on models trained on 2D single image super-resolution (SISR). The proposed pipeline is modular, which means we can train it with any domain-specific dataset, e.g., 2D CT images, or it can also be trained on colored image datasets to learn useful features depending on the objective. Recent work of Asano et al. [24] demonstrated that a single image contains enough information to train the initial layers of a neural network given enough data augmentation. Our fundamental motivation is to enable rapid prototyping by selecting any 2D SISR trained on a custom dataset.

We process the low-resolution image $I_{\downarrow r}$ using a state-of-the-art 2D SISR pipeline. The selection of a 2D SISR pipeline is crucial for transfer learning and self-supervised learning of the subsequent module, i.e., the Fusion module. The Fusion module relies on the pre-trained 2D SISR model in the self-supervised case. We use ESRGAN [25] trained on 2D CT images to demonstrate that the properties such as realism and sharpness, inherited from a pre-trained 2D SISR, can be transferred to the Fusion module. ESRGAN is an encoder-decoder-based architecture trained using L_1, adversarial and perceptual loss.

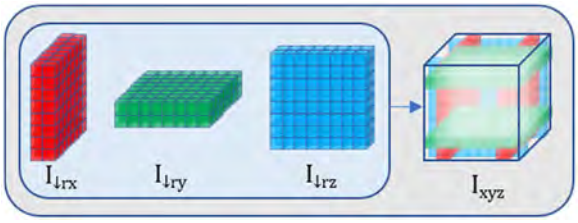

Fig. 2. 2D SISR slices overlap: $I_{\downarrow r_x}$, $I_{\downarrow r_y}$ and $I_{\downarrow r_z}$ can be superimposed to obtain I_{xyz}. The empty regions are missing information that SIT-SR 3D learns.

The input image $I_{\downarrow r}$ is processed along three axes x, y, and z using a pre-trained 2D SISR. Consider the dimensions of $I_{\downarrow r}$ given by Eq. (2).

$$I_{\downarrow r} : c \times X \times Y \times Z \tag{2}$$

Where c are the number of channels of greyscale or RGB image. We need to process $I_{\downarrow r}$ along each dimension x, y, and z. However, 2D SISR can process images of the form $I_{2D} : B \times c \times H \times W$ only, where B is the batch size, H and W are the dimensions of the 2D image. Therefore, in order to process $I_{\downarrow r}$, we first need to transpose it to obtain three copies $I_{\downarrow rcx}$, $I_{\downarrow rcy}$ and $I_{\downarrow rcz}$ as shown in Eqs. (3), (4) and (5).

$$I_{\downarrow rcx} : X \times c \times Y \times Z \tag{3}$$

$$I_{\downarrow rcy} : Y \times c \times X \times Z \tag{4}$$

$$I_{\downarrow rcz} : Z \times c \times X \times Y \tag{5}$$

We transpose $I_{\downarrow r}$ such that the considered axis is along the first dimension, whereas the remaining axes are on the third and the fourth dimension. We then process images $I_{\downarrow rcx}$, $I_{\downarrow rcy}$ and $I_{\downarrow rcz}$ by 2D SISR by considering the first dimension as the batch size. We again transpose the output of 2D SISR to restore the axes to normal for all three inputs. This operation's output gives us three upsampled anisotropic volumes $I_{\downarrow r_x}$, $I_{\downarrow r_y}$ and $I_{\downarrow r_z}$. All three inputs $I_{\downarrow rcx}$, $I_{\downarrow rcy}$ and $I_{\downarrow rcz}$ share the weights of 2D SISR, i.e., we use the same network for upsampling along axes x, y, and z.

4.1. Anisotropic volume interpolation

Three anisotropic volumes $I_{\downarrow r_x}$, $I_{\downarrow r_y}$ and $I_{\downarrow r_z}$ contain unique information. If they are overlayed on each other to fit in a isotropic cube, they result in a sparsely populated cube I_{xyz} which contains several empty and filled regions as shown in Fig. 2. One can treat the transformation from I_{xyz} to I as an inpainting problem. However, We first interpolate $I_{\downarrow r_x}$, $I_{\downarrow r_y}$ and $I_{\downarrow r_z}$ to form I^x, I^y and I^z. To achieve this, we employ a trilinear interpolation operation. The interpolated volumes, I^y and I^z are now isotropic and still contain all the useful information which was present in $I_{\downarrow r_x}$, $I_{\downarrow r_y}$ and $I_{\downarrow r_z}$. We feed these three volumes to the Fusion module as input.

4.2. Fusion module

We combine the information contained in the isotropic volumes I^x, I^y, and I^z into a single volume I using the Fusion module. The Fusion module's goal is to learn how to combine this information

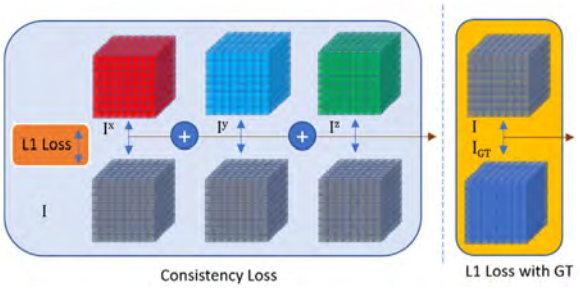

Fig. 3. Consistency Loss: I^x, I^y, and I^z are compared with I based on L_1 distance. This leads to unsupervised training compared to L_1 loss between I and I_{GT}.

in the best possible way. The Fusion module stacks the three inputs on the channel dimension such that we obtain a three-channel volume $I_{stack} : 3*c \times X \times Y \times Z$. This volume passes through a 3D convolutional neural network. We adapt EDSR [16] architecture to 3D for our purposes by converting all operations such as convolution and batch normalization to 3D to form 3D EDSR. We also take the average of the three input cubes and add it to the output of the 3D EDSR as shown in Eq. (6).

$$I_{avg} = \frac{I^x + I^y + I^z}{3} \tag{6}$$

Note that the input to the 3D EDSR has three channels, whereas this module's output is a 3D volume with a single channel. The input and output of the average module is a 3D volume with a single channel. The Fusion module's output is I with a single channel, which is the upsampled version of $I_{\downarrow r}$ by a factor r.

4.3. Loss function formulation

We train the Fusion module in a supervised or self-supervised manner. We achieve this by employing various loss functions. Next, we describe these loss functions in detail.

4.4.1. Supervised training with L_1 loss

If ground truth paired images are available, Fusion module can be trained in a supervised setting by using L_1 loss. We calculate the L_1 distance between I and the ground-truth high-resolution image I_{GT} by using the Eq. (7):

$$L_1 = \|I - I_{GT}\|_1 \tag{7}$$

The L_1 loss has a smoothing effect on the output, which is a known property of L_1 loss [15,25]. If we train 2D SISR on the adversarial loss, training the Fusion module with L_1 loss alone can lead to over smoothing of the output image I. This formulation leads to an inadequate transfer of properties of 2D SISR to I, but this method achieves the highest possible PSNR. This method also requires ground truth that is either not available or is expensive to obtain in medical CT or digital rock domains.

4.4.2. Self-Supervised training with consistency loss

We propose a unique formulation of loss to transfer the 2D SISR properties to the Fusion model. We call this formulation the consistency loss. The loss is given in Fig. 3 and Eq. (8):

$$L_c = \|I^x - I\|_1 + \|I^y - I\|_1 + \|I^z - I\|_1 \tag{8}$$

This loss calculates the L_1 distance of I from each of the three interpolated volumes I^x, I^y and I^z. It does not require ground truth image I_{GT}. In our experiments, we note that consistency loss maintains the desirable properties of preceding 2D SISR in the output image I. It can also filter out some high-frequency noise due to the L_1 nature of the three terms. The resulting image is sharp and also has lower noise levels than the real image. This noise filtering is an added advantage as CT images often suffer from high-frequency noise.

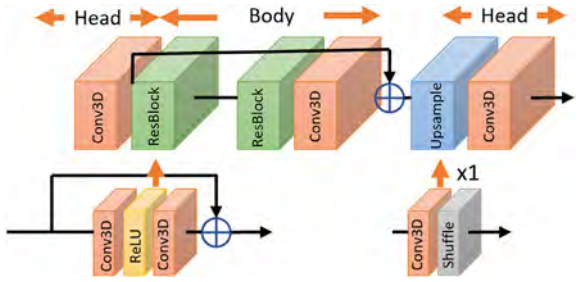

Fig. 4. 3D EDSR Architecture: We modify EDSR [16] and add 3D convolutional layers instead of 2D. This 3D EDSR is the part of the fusion module of our SIT-SR 3D model.

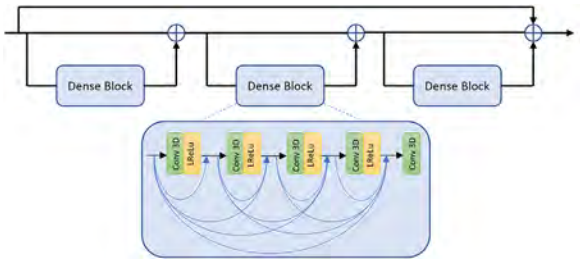

Fig. 5. 3D RRDB: The 3D residual in residual dense block contain 3D convolutional layers instead of 2D. We use this as the building block of our 3D RRDB-Net and 3D ESRGAN baselines [25].

4.4.3. Hybrid loss

We also propose to use L_1 loss in tandem with consistency loss that allows controlling the quality of the output. We, therefore, introduce a hybrid loss. The hybrid loss is given in Eq. (9).

$$L_{(} = \alpha L_1 + (1 - \alpha) L_{]} \tag{9}$$

The parameter α serves as a tuning parameter that can control the contribution of each loss. By changing the parameter α, we can obtain the desired quality in the output image I. We can set the α value closer to 1 to achieve a higher PSNR. On the other hand, if the output needs to be closer to the 2D SISR model, a lower value can be used.

5. Experiments

5.1. Data

We utilize the digital rock dataset provided by Wang et al. [21] since it is the largest dataset available for this study. This work uses the default train, validation, and test split. This dataset consists of paired 2D (12,000 images) and 3D (3000 images) low and high-resolution images of various rock types. We use the input (x4 downsampled) and output image pairs for this study while using all rock types they provide. We used the 2D image pairs to train 2D ESRGAN while using 3D image pairs to train the 3D baselines and the Fusion module in supervised mode. They provide low-resolution images by downsampling high-resolution images using various downsampling operations such as box, triangle, lanczos2, lanczos3, and Lanczos. We use this dataset to ensure that the model is robust to the type of down-sampling operation used.

5.2. Training and testing details

We used PyTorch for all our models, and experiments. We utilize two RTX Titan GPUs for training our models. We build upon open-source Github repositories [26,27].

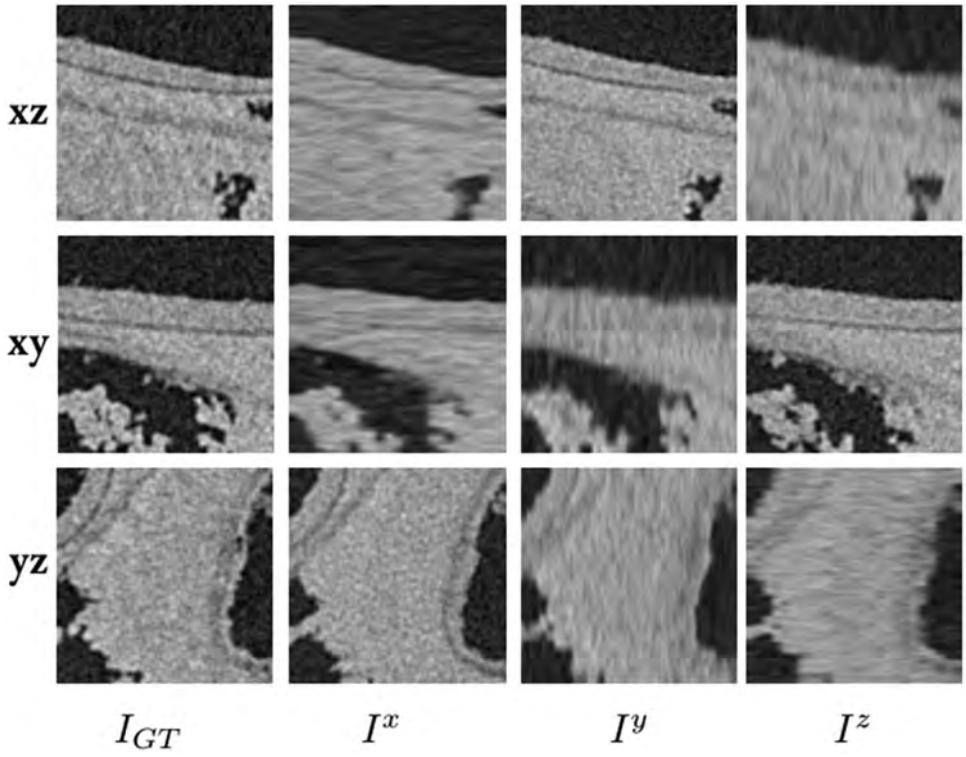

Fig. 6. Why the Fusion module is necessary? The first slice of cubes I_{GT}, I^x, I^y, I^z from all three faces, i.e., xz, xy, and yz, is shown. Only one of the frames for each I^x, I^y, and I^z contains sharp details. To form I, all three frames should be super-resolved. The Fusion module helps to combine the best of I^x, I^y, and I^z.

5.2.1. 2D SR image pipeline

We train a 2D ESRGAN [25] for the image super-resolution using 2D paired images provided by Wang et al. [21]. We modify the generator of ESRGAN to have 23 residual blocks. We use a batch size of 16. The image size used for training was 128 × 128. We use rotation and flipping to augment the training data. We first train the generator for 50 epochs with L_1 loss. Then we use the weights of this generator to train it further for 15 epochs with adversarial and VGG loss to obtain the final 2D ESRGAN configuration. More training details are in the supplementary section.

5.2.2. Interpolation of anisotropic 3D volumes

The anisotropic volumes produced after the 2D operation are converted to isotropic volumes using an interpolation operation. We used trilinear interpolation for this purpose due to its low computational cost. We also considered the consistency loss variant, which works with anisotropic volumes directly without the interpolation. However, this leads to checkerboard artifacts since the network is not motivated to learn anything meaningful or spatial consistent in the empty 3D regions.

5.2.3. Fusion module

Fig. 6 shows the importance of the Fusion module to combine I^x, I^y and I^z to formulate I. We train SIT-SR 3D in three different configurations (supervised, self-supervised and hybrid) as described in the method section. We can train the hybrid configuration with various values of the parameter α. We document the result with α set to 0.5, but in our experiments, we found that this parameter can control the output quality. Choosing a high value for α leads to smooth output and high PSNR, while low values drive the results closer to the underlying pipeline's properties. The training converges in about 12 h with a batch size of 2. We use a 3D EDSR architecture for the Fusion module as shown in Fig. 4. This custom 3D EDSR contains 16 residual blocks. 3D EDSR is the only component in the Fusion module that needs to be learned.

5.2.4. 3D baseline

To perform a comparison, we create our own 3D baseline networks. The baseline of choice is ESRGAN [25] due to its good performance in 2D domain. We designed the baseline network by converting all 2D operations such as convolution, batch normalization, etc., into 3D versions. The basic building block of our baselines is the 3D residual in residual dense block (3D RRDB) as shown in Fig. 5. Similarly, the 3D version of adversarial loss is obtained by converting the 2D convolution-based discriminator model to a 3D convolution-based model. However, a 3D version of VGG loss [7] does not exist. Therefore, we apply a 2D VGG loss frame by frame on the complete 3D volume along all three dimensions such that we obtain three-loss values along each dimension. We take the average of the three VGG loss terms to ensure the best possible training for the network.

3D ESRGAN uses 3D RRDBNet as the generator architecture. We first train 3D RRDBNet with L_1 loss. We then use this 3D RRDBNet as one of the baselines. We train 3D RRDBNet with the L_1 loss as 3D RRDBNet for reference in the quantitative and qualitative results. We then train the 3D ESRGAN by using weights of pretrained 3D RRDBNet as initial weights. This network formulates the second baseline, and we call it 3D ESRGAN in results. This work uses the default test, and train splits provided by [21]. 3D RRDBNet was trained with the batch size 6, while we train the 3D ESRGAN with the batch size two due to the GPU memory constraint caused by an additional discriminator network. The input channel size for both networks was one instead of three due to the training set's greyscale nature. We flip and rotate the training images for data augmentation. The size of the high-resolution image used for training was 92^3.

5.3. Quantitative results

Table 1 summarizes the quantitative results. SIT-SR 3D performs well in both supervised and self-supervised settings. It out-

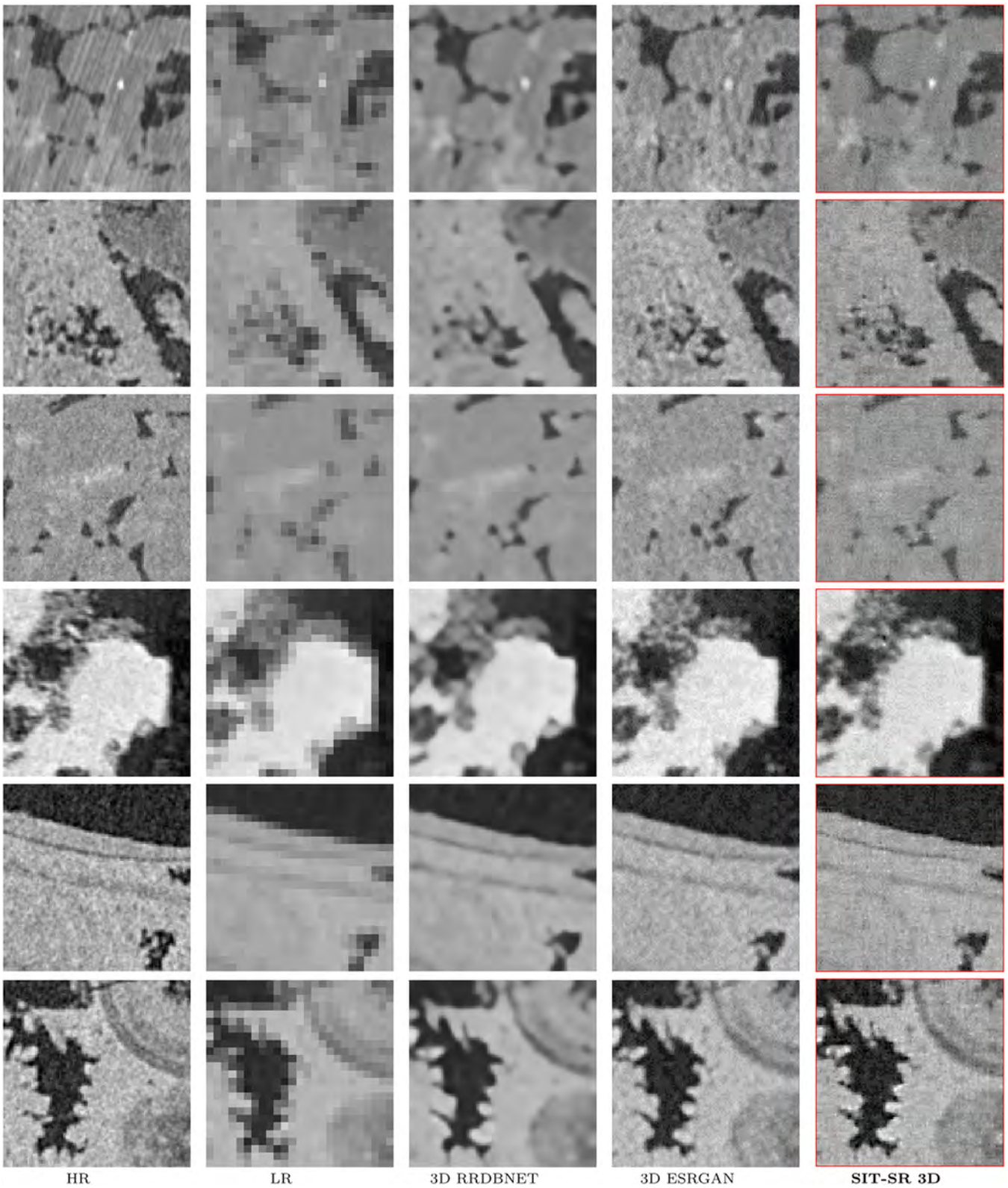

Fig. 7. This figure shows a visual comparison of different methods. HR, LR indicates high-resolution and low-resolution images. 3D RRDBNet is a 3D convolution-based network supervised with L_1 loss. 3D ESRGAN is trained with GAN and VGG loss using pre-trained weights of 3D RRDBNet. SIT-SR 3D is trained in an utterly self-supervised setting using only the consistency loss.

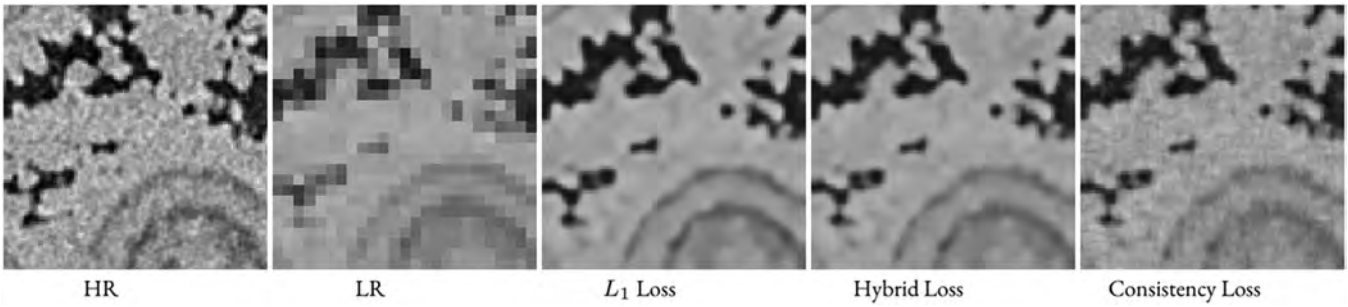

Fig. 8. Effect of α: This figure shows the ablation study with various values of α in Eq. (9). The values are 1, 0.5, and 0 for supervised (L_1 Loss), hybrid, and self-supervised (consistency loss) respectively of SIT-SR 3D.

Table 1
Quantitative comparison of SIT-SR 3D with 3D baselines. The best supervised case is underlined. The self-supervised case is shown in **bold**. We perform all experiments with x4 downsampled image as input.

Arch	Loss	Params	PSNR (dB)	SSIM
3D RRDBNet	L_1	50.05 M	30.84	0.71
3D ESRGAN	GAN	101.15 M	28.41	0.60
SIT-SR $3D_{sup}$	L_1	31.44 M	30.98	0.69
SIT-SR $3D_{self}$	L_c	31.44 M	**29.78**	**0.64**

Table 2
Effect of α and Average Module: We performed an ablation study to see the effect of α and average module on the PSNR and SSIM. The value of α is 1, 0.5 and 0. The best supervised case is underlined, whereas the best self-supervised case is in **bold**.

Scale	Loss	Average Module	PSNR (dB)	SSIM
x4	L_1	Yes	30.987	0.6921
	L_c	Yes	**29.789**	**0.6441**
	L_h	Yes	30.932	0.6846
	L_1	No	30.932	0.6935
	L_c	No	29.788	0.6440
	L_h	No	30.90	0.6844

performs the end-to-end 3D learning methods such as 3D RRDBNet in purely supervised settings. In the self-supervised setting, it has a higher PSNR than the 3D ESRGAN. The number of training parameters used by SIT-SR 3D are lesser than both 3D RRDBNet and 3D ESRGAN, demonstrating the merit of this approach.

5.4. Qualitative results

Fig. 7 shows the qualitative performance of SIT-SR 3D on the test set of 3D image pairs provided by Wang et al. [21]. The figure shows that SIT-SR 3D learns sharp details in the self-supervised setting well, and it also transfers the properties of the underlying 2D ESRGAN to the 3D SISR. The output of SIT-SR 3D is sharper than the one produced in a wholly supervised 3D RRDBNet (with L_1 loss). The produced result has less noise and artifact than the GT data and 3D ESRGAN's output due to the L_1 loss formulation of the consistency loss. Unfortunately, all the slices of the 3D image cannot be shown here due to presentation constraints. We have attached 3D images in the supplementary with the section and instructions on how to view them.

5.5. Ablation study

We performed ablations as shown in Fig. 8 and Table 2 to see the effect of α and the average module on the performance of SIT-SR 3D. The hybrid loss allows us to control the quality of the output, i.e., higher α leads to more blurry images with high PSNR results, and lower α favors the consistency loss and produces sharper results with lower PSNR. In this work, we only used α value of 0.5, which can be changed based on the required output. We also note that the average module has a relatively small effect on the PSNR, but since it consistently improves PSNR values in all experiments, we choose to keep it in the pipeline.

6. Discussion and future work

We have presented SIT-SR 3D, a modular and efficient network for super-resolution of 3D images. The proposed approach learns to super-resolve 3D low-resolution images in a self-supervised manner. We achieved this task by utilizing a 2D SISR pipeline trained with adversarial and VGG loss on 2D image pairs. We applied this 2D SISR along three dimensions of the 3D image to obtain three asymmetric cubes. We interpolated these anisotropic volumes using trilinear interpolation to obtain isotropic volumes. Then these symmetric cubes were fused to form a single cube. The fusion operation was learned by a 3D CNN using a novel consistency loss. The proposed approach outperformed the end-to-end 3D baseline when trained in a supervised manner in quantitative and qualitative metrics while using fewer parameters. We can use any 2D SISR pipeline depending on the desired output characteristics. SIT SR-3D is especially useful when 3D ground-truth is not available, but 2D ground-truth is available. In the case of digital rocks, it is often the case that 3D high-resolution ground truth is not available. We can go beyond a particular resolution using backscattered electrons in a scanning electron microscope (BSEM) [28] to obtain 2D images. We can then train a 2D SISR on 2D SEM images in these scenarios and transfer the knowledge to upsample low-resolution 3D CT images. However, this will be a subject of future study.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was partially supported by the Norwegian Research Council (grant number 296093) and the members of the SmartRocks joint industry project (ENI AS, Repsol AS, and Chevron Corporation).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patrec.2023.01.008.

References

[1] H. Andrä, N. Combaret, J. Dvorkin, E. Glatt, J. Han, M. Kabel, Y. Keehm, F. Krzikalla, M. Lee, C. Madonna, M. Marsh, T. Mukerji, E.H. Saenger, R. Sain, N. Saxena, Digital rock physics benchmarks—Part I: imaging and segmentation, Comput. Geosci. 50 (2013) 25–32.

[2] Y. Chen, F. Shi, A.G. Christodoulou, Z. Zhou, Y. Xie and D. Li, *Efficient and accurate MRI super-resolution using a generative adversarial network and 3D multi-level densely connected network*, arXiv, 2018.

[3] C. Peng, W.-.A. Lin, H. Liao, R. Chellappa, S.K. Zhou, SAINT: spatially aware interpolation network for medical slice synthesis, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 6, 2020.

[4] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Proceedings of the 25th International Conference on Neural Information Processing Systems, 1, 2012, pp. 1097–1105. -.

[5] Y. LeCun, B.E. Boser, J.S. Denker, D. Henderson, R.E. Howard, W.E. Hubbard, L.D. Jackel, Handwritten digit recognition with a back-propagation network, Adv. Neural Inf. Process. Syst. 2 (1990) 396–404.

[6] W. Shi, J. Caballero, F. Huszár, J. Totz, A.P. Aitken, R. Bishop, D. Rueckert, Z. Wang, Real-time single image and video super-resolution using an efficient, CoRR abs/1609.05158 (2016).

[7] J. Johnson, A. Alahi, F.-.F. Li, Perceptual losses for real-time style transfer and super-resolution, CoRR abs/1603.08155 (2016).

[8] Y. Wang, Q. Teng, X. He, J. Feng, T. Zhang, CT-image of rock samples super resolution using 3D convolutional neural network, Comput. Geosci. 133 (2019) 104314 12.

[9] K. Nazeri, H. Thasarathan, M. Ebrahimi, Edge-Informed Single Image Super-Resolution, arXiv, 2019.

[10] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CoRR abs/1512.03385 (2015).

[11] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, CoRR abs/1501.00092 (2015).

[12] J. Yang, J. Wright, T.S. Huang, Y. Ma, Image super-resolution via sparse representation, IEEE Trans. Image Process. 19 (11) (2010) 2861–2873.

[13] J. Kim, J.K. Lee, K.M. Lee, Deeply-recursive convolutional network for image super-resolution, CoRR abs/1511.04491 (2015).

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014) 2672–2680.

[15] C. Ledig, L. Theis, F. Huszar, J. Caballero, A.P. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, Photo-realistic single image super-resolution using a generative adversarial, CoRR abs/1609.04802 (2016).

[16] B. Lim, S. Son, H. Kim, S. Nah, K.M. Lee, Enhanced deep residual networks for single image super-resolution, CoRR abs/1707.02921 (2017).

[17] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, Residual dense network for image super-resolution, CoRR abs/1802.08797 (2018).

[18] G. Huang, Z. Liu, K.Q. Weinberger, Densely connected convolutional networks, CoRR abs/1608.06993 (2016).

[19] Y. Lu, The Level Weighted Structural Similarity Loss: A Step Away from the MSE, arXiv, 2019.

[20] A. Horé, D. Ziou, Image quality metrics: PSNR vs. SSIM, in: 2010 20th International Conference on Pattern Recognition, 2010, pp. 2366–2369.

[21] Y.D. Wang, R.T. Armstrong, P. Mostaghimi, Enhancing resolution of digital rock images with super resolution convolutional neural networks, J. Pet. Sci. Eng. 182 (2019).

[22] H. Chen, X. He, Q. Teng, R.E. Sheriff, J. Feng, S. Xiong, Super-resolution of real–world rock micro-computed tomography images using cycle-consistent generative adversarial networks, Phys. Rev. E 101 (2) (2020).

[23] C. You, W. Cong, M.W. Vannier, P.K. Saha, E.A. Hoffman, G. Wang, G. Li, Y. Zhang, X. Zhang, H. Shan and, et al., CT super-resolution GAN constrained by the identical, residual, and cycle learning ensemble (GAN-CIRCLE), IEEE Trans. Med. Imaging 39 (1) (2020) 188–203 1.

[24] Y. Asano, C. Rupprecht, A. Vedaldi, A critical analysis of self-supervision, or what we can learn from a single image, International Conference on Learning Representations, 2020.

[25] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C.C. Loy, Y. Qiao, X. Tang, ESRGAN: enhanced super-resolution generative adversarial networks, CoRR abs/1809.00219 (2018).

[26] X. Wang, K. Yu, K.C.K. Chan, C. Dong and C.C. Loy, *BasicSR*, https://github.com/xinntao/BasicSR, 2020.

[27] H.T. Kamyar Nazeri and M. Ebrahimi, Edge-informed-sisr, https://github.com/knazeri/edge-informed-sisr, 2019.

[28] P. Wandrol, J. Matějková, A. Rek, High resolution imaging by means of backscattered electrons in the scanning electron microscope, Mater. Struct. Micromech. Fract. V 567 (2008) 313–316 11.