Neural image enhancement and restoration for time-lapse SPM images

Fuma Y asue†, Kota Shinjo†, Yuki Kondo, Kazutoshi Akita, Hibiki Mitsuboshi, Masamichi Yoshimura, and Norimichi Ukita*

Department of Advanced Science and Technology, Toyota Technological Institute, Nagoya, Aichi 468-8511 Japan

†E-mail: ukita@toyota-ti.ac.jp

Equally contributed authors.

Received June 9, 2022; revised July 21, 2022; accepted July 27, 2022; published online September 6, 2022

This paper presents methods for enhancing and restoring scanning probe microscopy (SPM) images. We focus on image super-resolution as enhancement and image denoising as restoration. Assumed that almost same time-lapse images are captured in the same area of each specimen. In contrast to a single image, our proposed methods using a recurrent neural network improve the enhancement and restoration of SPM images by merging the time-lapse images in order to acquire a single enhanced/restored image. However, subtle deformations between the time-lapse SPM images and degraded pixels such as noisy and blurred pixels in the SPM image disturb the network to successfully merge the images. For the successful merge, our methods spatially align the time-lapse images and detect degraded pixels based on the characteristic property of SPM images. Experimental results demonstrate that our methods can reconstruct sharp, super-resolved images, and clean noiseless images. © 2022 The Author(s). Published on behalf of The Japan Society of Applied Physics by IOP Publishing Ltd

1. Introduction

Scanning probe microscopy (SPM) allows us to inspect the surface of a specimen in nano scale.1,2 However, SPM has several problems for visual inspection. For example, the limited resolution of SPM makes it difficult to precisely inspect a specimen, while new scanning devices are developed for improving the resolution.3,4 In addition, SPM images are degraded by various reasons, such as noise and blur5 due to probe degradation. In order to resolve these problems, we enhance and restore the SPM images by machine learning such as deep neural networks. Specifically speaking, this paper proposes methods for image super-resolution (SR) as enhancement and image denoising and deblurring as restoration.

Image SR upscales an input low-resolution (LR) image to its high-resolution (HR) image. With upscaled SR images, a variety of applications can be realized. For example, distant-object detection,6–8 tiny-defect inspection,9 remote sensing,10 wide-angle image analysis,11 and cell image analysis.12 So with these examples, the upscaled SR image of a specimen allows us to visually inspect its detail and also develop automatic inspection systems. Similarly, in many image enhancement technologies,13–17 SR has been significantly improved by convolutional neural networks (CNNs).18–21 In general SR methods using CNNs,22 a SR network is trained so that a difference between a reconstructed SR image and its HR (ground-truth) image is minimized in the training process. A huge number of the training images are given so that the SR network is applicable to any new images whose image domain is similar to that of the training images.

However, such SR using a single input image is difficult because SR is a typically ill-posed problem where the number of pixels in the input LR image is less than that in the output SR image. In order to relieve this difficulty, multiple LR images are used as a set of input images in video SR23–25 and burst SR.26 Our proposed method also improves the quality of SR by employing time-lapse SPM images that capture the same location of a specimen.

Image denoising and deblurring reduce noisy and blurred pixels, respectively. Many of such degradations are caused by the degraded tip-apex of a cantilever. The cantilever of SPM is shown in Figs. 1(a) and 1(b). It is expected that its tip-apex is as sharp as possible as shown in Fig. 1(a). However, the tip-apex is worn out after repeated scanning, and materials are adhered to the tip-apex sometimes, as shown in Fig. 1(b). This produces pixel degradations in a SPM image, as shown in Figs. 1(c) and 1(d). Such noise and blur are specific to the SPM image. For restoring the image degraded by these SPM-specific noise and blur, it is better to design a restoration method optimized for these degradations rather than generic image restoration methods [e.g. denoising27–29 and deblur30–32]. To cope with this SPM-specific degradation problem, the 3D shape of the degraded tip-apex is estimated in Ref. 33. With this estimated shape of the tip-apex, we can restore the captured surface of a specimen. However, naive blind shape estimation using only the captured surface is not so reliable. In order to improve the reliability, machine learning is utilized with a number of extra training data for classifying each captured surface into a clean surface or a degraded surface in Ref. 34. However, the captured surfaces are just classified with no restoration in Ref. 34.

The goal of our work is to restore images with machine learning such as CNNs. The basic strategy of denoising35 and deblurring36 using CNNs is same with that of SR. That is, a restoration network is trained so that error between an output restored image and its ground-truth image is minimized. In previous methods of microscopy image restoration,37–40 a single image is fed into a network in order to restore this image. On the other hand, in our proposed method, as with SR mentioned above, time-lapse SPM images are employed for improving the quality of the output image.

Different from SR, however, denoising and deblurring have their own technical issue. For any supervised machine learning, a set of pairs of input and its ground-truth data is required. For SR, a HR image as the ground-truth and its LR
image as the input are required as the paired data. For denoising/deblurring, a clean image as the ground-truth and its noisy/blurred image are required as the pair. Any HR SPM image can be easily and almost-uniquely downscaled to its LR image in order to make the pair for SR. On the other hand, it is difficult to assume that a clean image as the ground-truth is always available in each location. In the denoising method for STEM images, temporal images are captured and averaged to approximately produce a clean image in each location. However, it is known that such an averaged image is blurred. In addition, noise and blur produced by the degradation of a cantilever tend to be continuously observed in the same location in such temporal images. These noise and blur cannot be reduced by the averaging process.

Furthermore, noise and blur that are specific to SPM images are unknown and difficult to be simulated. Therefore, even if the clean image is available, it is difficult to synthesize realistic noisy and blurred images from this clean image. That is, it is difficult to have a pair of a clean image and its SPM-specific degraded image for training a restoration model. In previous methods for denoising microscopy images, empirically-produced synthetic degradations are given in order to artificially have the paired images. However, if these artificially-reproduced degradations are different from those observed in real SPM images, this difference makes it difficult to train the restoration model that is applicable to the real SPM images. Our proposed method avoids the unavailability of clean images in the training process by (i) selecting the cleanest image in each set of noisy time-lapse images as the ground-truth, if this cleanest image is clean enough, and (ii) using other images with real degradations as input images.

In this paper, time-lapse SPM images of carbon nanotube (CNT) are used for validating the effectiveness of our proposed methods for the aforementioned SR, denoising, and deblurring of time-lapse SPM images. In what follows, the details of “how to use time-lapse SPM images for SR” and “how to select the cleanest and other input images for denoising and deblurring” are described in Sects. 2.1 and 2.2, respectively.

2. Methods

2.1. Time-lapse image SR

As mentioned in Sect. 1, given a pair of a LR image and its HR image as training images, the LR image is fed into this network. Then its output SR image is compared with the HR image in order to train the SR network. The SR network is trained so that the difference between the SR and HR images (which are denoted by $S$ and $H$, respectively) is minimized. In our method, this difference is represented by the following mean square error (MSE) loss function:

$$L_{\text{MSE}}(H, S) = \sum_{x \in X, y \in Y} (H(x, y) - S(x, y))^2,$$

where $x \in X$ and $y \in Y$ denote $x$ and $y$ coordinates in an image, respectively. $H(x, y)$ and $S(x, y)$ are a pixel value at $(x, y)$ in HR and SR images, respectively. By minimizing this MSE for a number of training images, we can train the SR network properly.

In the aforementioned basic strategy for training the SR network, each HR image is given and downscaled to its LR image by a standard image scaling algorithm such as bicubic interpolation. While image upsampling such as SR is difficult because it is an ill-posed problem, it is easy to downscale any image with less error. It can be therefore assumed that we are able to easily collect each pair of LR and HR images for training, if any HR image is given.

For further improving the quality of SR, our proposed method employs a set of time-lapse images instead of a single input image. Our assumption is as follows. The time-lapse SPM images (e.g. 10 images) in each set are captured in the same area of each specimen. Under this capturing condition, all of these images should be almost same with each other. Among all the images, therefore, only one image is required to be super-resolved. The quality of this SR image is improved by merging all other images in the set.

Our proposed network for SR using time-lapse SPM images is a recurrent neural network shown in Fig. 2. The basic structure of this network is provided by our video SR network. Image features extracted in convolutional layers (indicated by “Conv” in the figure) are upscaled in the
projection modules. The upscaled features are concatenated and fed into the last convolutional layer for producing the output SR image. See the original paper\(^{41}\) for the details of each network component. Based on this video SR network, our network for SPM image SR is designed as follows:

- For machine learning, we need a number of training sets. Assume that we have \(N_s\) training sets of time-lapse SPM images. In our problem, each set consists of time-lapse images captured in the same location. Such training sets are captured in a variety of areas and specimens for collecting training images. Each training set is divided into (i) one ground-truth HR image [i.e. “Ground-truth” in Fig. 2] and (ii) other input images that are downsampled to their LR images and fed into the network [i.e. \(I_1, \ldots, I_N\) in Fig. 2].

- Unlike single-image SR, not only one input image (i.e. \(I_1\) in Fig. 2) but also other input images in the same time-lapse set (i.e. \(I_2, \ldots, I_N\) in Fig. 2) are fed into the network one by one. The outputs from \(I_2, \ldots, I_N\) (i.e. \(H_2, \ldots, H_N\) in Fig. 2) are integrated in the last convolution layer (i.e. the rightmost “Conv” in Fig. 2) in order to reconstruct the output image (i.e. SR image).

In addition to the aforementioned processes, we have to resolve the following problem for employing time-lapse SPM images for SR. SPM is often employed for observing a variety of deformable specimens such as CNT and biological samples. The time-lapse images of such deformable objects are changed. It is more difficult to merge the deformed images than similar images. To make it easier to merge the deformed images, we add the following processes:

- The deformations between images [i.e. \(F_i\) where \(i \in \{2, \ldots, n\}\) in Fig. 2] are estimated and fed into the network, as shown in Fig. 2. \(F_i\) represents the spatial pixel displacements from \(I_1\) to \(I_i\), which is called an optical flow image in general. We expect that \(F_i\) fills a gap between \(I_1\) to \(I_i\). In our method, \(F_i\) is estimated by a simple and fast optical flow estimation algorithm.\(^{42}\)

While previous methods\(^{43,44}\) also cope with deformations in SPM images, the interpolation scheme\(^{43}\) produces a blurry image, and highly non-convex minimization\(^{44}\) is slow and unstable.

- Even with the aforementioned optical flow, it is not easy to fill a large gap between \(I_1\) to \(I_i\). If the deformation is too large, it makes it too difficult to merge the images. To avoid this problem, such over-deformed images are removed from each time-lapse set. This image removal is achieved so that (i) the mean over all time-lapse images in each set is computed and (ii) we remove images that are significantly different from the mean image. This difference is expressed as the MSE between each image and the mean image. If the MSE is larger than a pre-defined threshold (denoted by \(T_r\)), this image is removed.

### 2.2. Time-lapse image denoising and deblurring

#### 2.2.1. Basic strategy for time-lapse image restoration.

As with for SR, our proposed method for image restoration employs time-lapse SPM images. The basic structure of our neural network for this restoration is equal to Fig. 2. The training of the image restoration network is also same with the one of the SR network so that the MSE loss, Eq. (1), is minimized. On the other hand, different from SR where an input image is upscaled in the projection modules, image restoration does not change the image size in its output. In order to hold the image size, the projection module is designed to hold the size of its output feature map.

Another important difference is that image denoising and deblurring should explicitly cope with degraded pixels. This is because image denoising and deblurring have to reduce

---

**Fig. 2.** (Color online) Our proposed recurrent network for SR and image restoration using time-lapse SPM images.

**Fig. 3.** (Color online) Time-lapse images that are partially degraded due to the degradation of a cantilever. All red and green circles are overlaid in the same location.
these degraded pixels, while SR processes degraded pixels as they are. Although many time-lapse SPM images are degraded, several pixels are degraded in some images, and those pixels can be captured with less noise in other images. In sample time-lapse images shown in Fig. 3, red and green circles indicate clean and degraded pixels, respectively. Our goal is to merge all clean pixels in order to restore one of the input time-lapse images.

The basic training strategy mentioned in Sect. 2.1 is not only for SR but also for most of any other image enhancement and restoration methods. That is, given a pair of input and its ground-truth images as training data, the network is trained so that its output gets close to the ground-truth image. To train a neural network for denoising and deblurring, degraded and clean images are input with its ground-truth images, respectively.

2.2.2. Image selection depending on the degradation score. While our SR and restoration problems are similar to each other as mentioned in Sect. 2.2.1, there are several differences between our SR and image restoration tasks. Due to these differences, we have the following three problems:

1. Training sets: In our SR task, all training sets can be used in each set. In our image restoration task, however, a training set cannot be used if all images in this set have a number of degraded pixels. This is because at least one clean image is required as the ground-truth for the training process.

2. Ground-truth image for training: In each training set, both input and its ground-truth images are required. In our SR task, any image can be the ground-truth in each set. In our image restoration task, however, the ground-truth image is required to be clean.

3. Input images: Input images should be also as clear as possible for better restoration. In our image restoration task, however, all images are degraded in some time-lapse sets. In what follows, our solution for each problem is described.

1. Training sets: In order to successfully train the network, we propose to split all training sets to clean sets and others. In each of the clean sets, at least one clean image is included. This clean image can be regarded as the ground-truth image. If no clean image is captured in a set, this set is not used for training the network.

2. Ground-truth image for training: In each training set, among all clean images, the cleanest image is selected and regarded as the ground-truth image used for training the network.

3. Input images: Among all input images, the cleanest image is selected as I₁ shown in Fig. 2. This is because a primary input in our network is I₁ so that I₁ is improved by merging clean pixels in other input images, I₂, …, IN.

In all the three solutions mentioned above, we assume that the cleanness level of each image can be measured. For example, the clearest image is selected in each training set. In our proposed method, the noise level, which is the opposite measure of the cleanness level, is defined by using image derivatives. The discrete image derivatives are approximately computed by subtraction between neighboring pixel values as follows:

![Fig. 4.](image-url) (Color online) Noise-level measurement using image derivatives. From clean and noisy images, (a) and (d), their horizontal image derivatives, (b) and (e), and vertical image derivatives, (c) and (f), are computed, respectively. The directions of these stripe noise and blur depend on the scanning direction of SPM.

© 2022 The Author(s). Published on behalf of Jpn. J. Appl. Phys. 61, 107001 (2022) F. Yasue et al.
\[
\frac{\partial I(x, y)}{\partial x} = I(x + 1, y) - I(x, y), \\
\frac{\partial I(x, y)}{\partial y} = I(x, y + 1) - I(x, y),
\]

where \( I(x, y) \) denotes a pixel value at \((x, y)\) in image \( I \). In the image derivatives, boundary lines are activated as shown in Fig. 4. Figures 4(b) and 4(c) are the horizontal and vertical image derivatives of a SPM image shown in Fig. 4(a), respectively. We assume that boundary lines in a microscopy image are uniformly random. If such an image with random boundary lines has no noise, “the sum of vertical derivatives, Eq. (3), in all pixels (denoted by \( D_v \)) minus “the sum of horizontal derivatives, Eq. (2), in all pixels (denoted by \( D_h \))” is expected to be zero. In SPM images, however, stripe noise is observed horizontally along the scanning direction of a microscope, shown in examples in Figs. 4(d), 4(e), and 4(f). Due to this property, the sum of vertical derivatives is larger than the sum of horizontal derivatives. This subtraction, \( D_v - D_h \), is considered to be the noise score, \( S_n \), in our method as follows:

\[
S_n = D_v - D_h = \sum_{x \in X} \sum_{y \in Y} \left| \frac{\partial I(x, y)}{\partial y} \right| - \sum_{x \in X} \sum_{y \in Y} \left| \frac{\partial I(x, y)}{\partial x} \right|.
\]

### 2.2.3 Image division for data augmentation

As shown in Fig. 3, many images might be degraded by noise and blur. If all images in a set of time-lapse images are degraded significantly, this set cannot be used in the training process of our method. This is critical because machine learning cannot work well if the amount of training data is insufficient.

However, these noise and blur are observed partially in general, as indicated by red circles in Fig. 5. With this property, we propose to divide each SPM image to sub-images and to feed the sub-image to our restoration network. In an example shown in Fig. 5, each SPM image is divided to four sub-images, each of which is enclosed by colored rectangles. Green and blue rectangles contain no and several degraded pixels, respectively. While both the SPM images 1 and 2 in Fig. 5 cannot be used as the ground-truth image due to noise pixels, the sub-images with no noise pixels can be employed as the clean ground-truth data.

Since the aforementioned method feeds a sub-image to the network, its output is also a sub-image. If one requires the output restored image to be its original size, all sub-images generated from each SPM image are tiled to be one image.

### 3. Results

#### 3.1. Dataset

We captured the following specimen for our experiments:

- **Specimen**: CNT.
- **Substrate**: SiO2 (300 nm)/Si, which was cleaned by ultrasonic (10 mins).
- **Substrate dimension**: 5 \( \times \) 5 mm.
- **UV ozonation**: 1 h.

This specimen was captured in the following preparation and measurement conditions:

- **Microscopy**: OXFORD INSTRUMENTS Asylum Research, MFP-3D Origin.
- **Cantilever**: Tap300Al-G.
- **Cantilever preparation**: No special preparation; neither cleaning nor deposition was provided to the cantilever.
- **Scanning mode**: Dynamic Force Mode.
- **Scan rate**: 0.8 Hz.
- **Capturing range**: 2 \( \times \) 2 \( \mu \)m.
- **Image resolution**: 256 \( \times \) 256 pixels.
- **Tilt correction software**: Gwyddion.
- **Time-lapse**: 10 images/location.
- **Captured locations**: 162 locations.
- **Capturing time per a cantilever**: 16 h and 4 min for capturing 2 \( \times \) 2 \( \mu \)m 160 times.

These 162 sets of time-lapse images are divided into 143 training sets and 19 test sets.

**Training Sets**: For verifying the noise robustness of the proposed method, these 143 training sets are divided into 67 and 76 sets so that, in each of the latter 76 sets, the cleanest time-lapse image is less noisy. That is, with the latter set, our network can be trained properly for reducing the noise more. We call the latter 76 training set the Dataset-NL. All training sets are used in all experiments except for the one for Table I(f) in which only the Dataset-NL is trained.
If the amount of training data is not enough, the network cannot perform well. In reality, 143 training sets in our experiments are much smaller than general datasets for image processing and recognition. It is difficult to train a huge non-linear model such as a neural network with only such a small amount of data. To relieve such difficulty in training with small amount of data, fine-tuning from a pre-trained model is employed in general. For fine-tuning, the pre-trained model is trained with a huge amount of data, whose domain is different from the target domain (i.e. SPM images in our case) in advance. By using this pre-trained model as an initial model, the network is trained with a small amount of data of the target domain. In all experiments shown in this paper, the network is trained with this fine-tuning strategy. In our experiments, the network is pre-trained with general-purpose video datasets [i.e. Vimeo90K, SPMCS, and Vid4] in which general landscape photos are included.

Test Sets: For evaluating the performance of noise removal with a test set, the noisefree image of the images in this test set is required. It is, however, difficult to capture such a perfectly-noisefree SPM image. For quantitative evaluation in this paper, in each test set, the most noiseless image is selected and regarded as the ground-truth. For this selection, the amount of noise is defined by Eq. (4).

### Table I. Quantitative evaluation results of three metrics, namely the noise score $\downarrow$, Eq. (4), SSIM $\uparrow$, Eq. (6), and sharpness score $\uparrow$, Eq. (8). In the metrics that are marked with $\uparrow$ and $\downarrow$, higher and lower values are better, respectively. Each score is the average over all test images.

|                | Noise $S_n \downarrow$ | SSIM $\uparrow$ | Sharpness $S_s \uparrow$ |
|----------------|------------------------|------------------|--------------------------|
| (a) Cleanest image | 0.0328 | 1.000 | 936 |
| (b) Our method 7 input images | 0.0157 | 0.474 | 266 |
| All training sets | | | |
| Sub-image size 64 × 64 | | | |
| (c) 9 input images | 0.0121 | 0.584 | 197 |
| (d) 4 input images | 0.0175 | 0.305 | 258 |
| (e) 1 input image | 0.0192 | 0.338 | 272 |
| (f) Dataset-NL | 0.0175 | 0.590 | 231 |
| (g) Sub-image size 32 × 32 | 0.0135 | 0.545 | 170 |
| (h) Sub-image size 96 × 96 | 0.0185 | 0.482 | 138 |

### 3.2. Implementation details

The implementation details of our networks are as follows. The sub-image size is 64 × 64 pixels. The number of input frames is seven.

In addition to the proposed image division process for data augmentation, as general data augmentation, the training time-lapse images are also augmented by (i) random brightness adjustment using the Gamma correction with $0.8 \leq \gamma \leq 1.1$ and (ii) random horizontal and vertical image flipping. Note that all time-lapse images in each set are augmented in the same manner so that these images are temporally consistent.

In the training process, Adam (adaptive moment estimation) is used as an optimizer. Hyper parameters used in our neural network training are as follows. The learning rate is $10^{-4}$, and the mini-batch size is 12.

#### 3.3. Time-lapse image SR

In the literature (e.g. Refs. 18, 49), the performance of SR is evaluated by peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). PSNR compares HR and SR images (denoted by $H$ and $S$) pixelwise as follows:

$$
\text{PSNR} = \frac{20 \log_{10} \text{Max}_H}{\sqrt{\frac{1}{WH} \sum_{i=1}^{W} \sum_{j=1}^{H} [H(i,j) - S(i,j)]^2}}
$$

where $W$ and $H$ denote the width and height of each image. Max$_H$ is the maximum pixel value in $H$. In the case of a 8-bit gray scale image, Max$_H$ = 255.

On the other hand, SSIM evaluates the appearance correlation between images with small deformations. SSIM between two image windows $x$ and $y$ are defined as follows:

$$
\text{SSIM} = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{\mu_x^2 + \mu_y^2 + c_1(\sigma_x^2 + \sigma_y^2 + c_2)}
$$

where $(\mu_x, \mu_y)$, $(\sigma_x^2, \sigma_y^2)$, and $\sigma_{xy}$ denote the averages of $x$ and $y$, the variances of $x$ and $y$, and the covariance of $x$ and $y$, respectively. $c_1 = (0.01 \times 255)^2$ and $c_2 = (0.03 \times 255)^2$ denote regularization constants. SSIM has the maximum value (i.e. 1), if $x = y$. 

---

Fig. 6. (Color online) PSNR and SSIM scores of SR images. The horizontal axis indicates the threshold, $T_s$, for removing time-lapse images from each training set. Different bar colors indicate different thresholds, $T_s$, for each test set of time-lapse images. “All” indicates that all images are used without removing any image.
Experimental results are shown in Fig. 6. We conducted experiments with several thresholds, \( T_r \), for removing time-lapse images from each training and test set. In both results [i.e. (a) PSNR and (b) SSIM], the following observations can be seen:

- In terms of the threshold for the test images, a lower threshold achieves better results in all cases. That is, among all bars in the same cluster (i.e. six different color bars that clump together along the horizontal axis), a lower threshold achieves better performance than a higher threshold.
- Against our expectations, on the other hand, a lower threshold for the training images cannot improve the SR quality. That is, among all bars with the same color, a lower threshold along the horizontal axis achieves better performance than a higher threshold.
- The PSNR scores distribute between around 43.0 and 43.5. This difference is not so remarkable, but sensible. On the other hand, the distribution of the SSIM scores are tiny. This might mean that SSIM can compare the similar HR and SR images without being disturbed by the physical deformations of a specimen in each set of time-lapse images.

3.4. Time-lapse image denoising and deblurring

3.4.1. Quantitative evaluation metrics. a. Noise removal. The performance of noise removal is evaluated with the amount of noise expressed by \( S_n \), given in Eq. (4).

In addition, the performance is also quantified with the difference between the noise-removed image and its ground-truth clean image. The image difference is, in general, evaluated by PSNR and SSIM as mentioned in Sect. 3.3. Since a pixelwise metric such as PSNR is inappropriate for evaluating the SPM image of a deformable specimen because the different images of the same specimen are deformed, only SSIM is used for evaluating noise removal.

b. Deblur. The result of deblur is considered to be better if the result image is sharper. The image sharpness is evaluated by the following Laplacian filter:

\[
L(I_{3 \times 3}(x, y)) = -I(x, y-1) - I(x - 1, y) - I(x + 1, y) - I(x, y+1) + 4I(x, y),
\]

(7)

where \( I_{3 \times 3}(x, y) \) denotes the output value of the Laplacian filter with \( 3 \times 3 \) pixels at \((x, y)\) in an image. The Laplacian filter is a standard image processing filter used for edge detection. Since the Laplacian filter outputs are larger and smaller values in sharper edge pixels with less blur and blurry pixels, respectively (as shown in Fig. 7), this filter can be used for evaluating how much each image is deblurred. The sharpness score is expressed with Eq. (7) as follows:

\[
S_s = \sum_{x \in X} \sum_{y \in Y} L(I_{3 \times 3}(x, y)).
\]

(8)

3.4.2. Evaluation. The performance of our method is quantitatively evaluated with the noise score, Eq. (4), SSIM, Eq. (6), and sharpness score, Eq. (8), as shown in Table I. The results of our method are shown in Table I as (b). For comparison, the scores of the ground-truth cleanest image are shown in Table I as (a).

The visual results of our method are shown in Fig. 8. We can see that (i) stripe noise observed in the input images are relieved in the output image and (ii) the output image gets close to the cleanest image in Fig. 8(a). In Fig. 8(b), on the other hand, our method fails to relieve noise in the input images. This is because there are only noise in all input images. In such a case, it is impossible to select clean images from the input images.

In Table I, (c), (d), and (e), gives the effects of the number of input images in our method which are verified. While our network whose results are shown in (b) accepts seven input images in experiments shown above, nine, four, and one images are fed into our network for comparison. In Table I, (f) shows the results of our network trained by the Dataset-NL. While the number of training sets in the Dataset-NL is less than all the training sets (i.e. 76 versus 162), all training images in the Dataset-NL are cleaner. In Table I, (g) and (h), shows the effects of the size of sub-images which are verified. While the default size is 64 \times 64 pixels, smaller and larger sizes (i.e. 32 \times 32 pixels and 96 \times 96 pixels) are used for evaluation. We interpret these results in Sect. 4.

4. Discussion

4.1. Time-lapse image SR

As shown in Fig. 6, lower thresholds \( T_r \) in test and training image sets can and cannot improve the SR quality, respectively. It is a natural consequence because the test images selected by the lower threshold are more similar to each other than all the test images in each set. The similar images are easier to be merged by our recurrent network for improving the output quality. We expect that the lower threshold also in the training images can intrinsically improve the performance. However, the SR quality is not improved in our experiments probably because we used only a small number of training images. For verifying this assumption, experiments with more training images are required.

4.2. Time-lapse image denoising and deblurring

a. Effect of our method. While the noise score \( S_n \) of (b) is better than (a) as we aim at noise reduction, the sharpness score \( S_s \) of (b) is worse than (a). In terms of \( S_n \), not only (b) but also all of other results (c)–(h) are inferior to (a). This might be because the MSE loss, Eq. (1), tends to produce blurry results, as validated in the literature.18

In Table I(b), SSIM = 0.474 is not high enough as a score of image similarity. This might be because of large deformation between time-lapse images. It is well-known that a sharper image can be reconstructed by additional constraints.
(e.g. minimization between not only images but also deep features) on training a neural network. However, these constraints tend to reconstruct fake images. For looking at images as art appreciation, images should be sharp (i.e. should be visually better), while realistic fakes are not problematic. On the other hand, such fakes are not acceptable for verifying experimental results on material science. Due to this difference between general image reconstruction tasks for art appreciation and verification of the experimental results, in future work, we aim at a trade-off between “noise removal and image sharpening” and “fake removal.”

b. Effect of the number of input images. While the noise score $S_n$ is best in (c) (i.e. with the maximum number of input images), the sharpness score $S_s$ is best in (e) (i.e. with the minimum number of input images). This might be because (i) noise can be better reduced by more input images and (ii) more input images degrade the aforementioned blurry effect.

c. Effect of noiseless training images. No significant difference is observed in the noise and sharpness scores between (b) and (f), while SSIM is relatively improved in (f). That is, SSIM (i.e. similarity between input images and their ground-truth image) is improved even with a small number of training images. This might be because of overfitting, which is a well-known effect that improves the performance of a machine learning model only for test data that are similar to training data, if this model is trained with a limited number of training data. For clearly verifying the effect of noiseless training images, experiments with more noiseless training images are required.

d. Effect of sub-image size. The noise score is better in smaller sub-images. This might be because it is more difficult to remove stripe noise (in other words, to produce a realistic noiseless image) in a larger sub-image. On the other hand, the sharpness score gets worse in smaller and larger sub-images. Compared with the score difference in the noise score, the sharpness score largely differs depending on the sub-image size. Therefore, the default sub-image size in our method is determined to be $64 \times 64$ pixels in our experiments so that the sharpness score is emphasized.

5. Conclusions
This paper proposed image SR, noise removal, and deblurring methods using time-lapse SPM images. These methods are designed with deep neural networks. Different from similar methods, our methods employ multiple input images (i.e. time-lapse images) for one output image.

Future work includes more realistic image reconstruction with less fakes and experiments with more training images. Since it is not easy to capture a huge number of time-lapse SPM images due to mechanical reasons of SPM, efficient time-lapse image capturing is also an important issue.

Acknowledgments
This work was partly supported by JSPS KAKENHI (Grant Nos. 19K12129 and 22H03618).
