Brain Informatics

RESEARCH Open Access

Enhancing brain image quality with 3D U-net for stripe removal in light sheet fluorescence microscopy

Changshan Li[1], Youqi Li[2,4], Hu Zhao[3,4] and Liya Ding[1*]

Abstract

Light Sheet Fluorescence Microscopy (LSFM) is increasingly popular in neuroimaging for its ability to capture high-resolution 3D neural data. However, the presence of stripe noise significantly degrades image quality, particularly in complex 3D stripes with varying widths and brightness, posing challenges in neuroscience research. Existing stripe removal algorithms excel in suppressing noise and preserving details in 2D images with simple stripes but struggle with the complexity of 3D stripes. To address this, we propose a novel 3D U-net model for Stripe Removal in Light sheet fluorescence microscopy (USRL). This approach directly learns and removes stripes in 3D space across different scales, employing a dual-resolution strategy to effectively handle stripes of varying complexities. Additionally, we integrate a nonlinear mapping technique to normalize high dynamic range and unevenly distributed data before applying the stripe removal algorithm. We validate our method on diverse datasets, demonstrating substantial improvements in peak signal-to-noise ratio (PSNR) compared to existing algorithms. Moreover, our algorithm exhibits robust performance when applied to real LSFM data. Through extensive validation experiments, both on test sets and real-world data, our approach outperforms traditional methods, affirming its effectiveness in enhancing image quality. Furthermore, the adaptability of our algorithm extends beyond LSFM applications to encompass other imaging modalities. This versatility underscores its potential to enhance image usability across various research disciplines.

Keywords 3D U-net, Image denoising, 3D neural data, Nonlinear mapping

1 Introduction

Neural image data is primarily acquired using modern microscopy techniques, including light sheet fluorescence microscopy (LSFM), wide-field fluorescence microscopy, confocal light sheet microscopy, multiphoton microscopy [1–5], etc. LSFM is a popular choice for obtaining high-resolution fluorescence neural images as it minimizes the photobleaching and phototoxicity of specimens [6, 7]. However, artifacts like stripes can exist in LSFM images and greatly degrade the image quality. Due to the complex nature of the tissue structure, there remains a significant refractive index mismatch among different tissue types even after clearing. This mismatch

*Correspondence:
Liya Ding
dinglyosu@gmail.com
[1]Institute for Brain and Intelligence, Southeast University, Nanjing, China
[2]Department of Neurology, Beijing Tiantan Hospital, Capital Medical University, Beijing, China
[3]Academy for Advanced Interdisciplinary Studies, Peking University, Beijing, China
[4]Chinese Institute for Brain Research, Beijing, China

often results in the presence of numerous 3D stripes with varying widths and brightness in the images. The presence of these stripes poses challenges to data processing tasks such as image-stitching and automatic neuron morphology reconstruction.

In LSFM images, typical hardware-based methodologies for stripe reduction involve structured illumination techniques. For example, a vertically scanned LSFM method was developed to reduce stripes in unidirectional LSFM and improve image quality [8]. Similarly, multidirectional LSFM averages images from different light directions and eliminates stripes [9]. However, the stripes in the existing LSFM images cannot be eliminated using these approaches. Additionally, each neural sample utilized for LSFM neuroimaging is highly valuable and is not reusable, making it essential to find alternative methods to remove stripes from existing LSFM images. Numerous methods have been proposed for eliminating image stripes, which are primarily categorized into three categories: digital filtering, statistics-based techniques, and learning-based approaches [10]. Digital filtering is commonly used due to its simplicity [11]. In mBrainAligner [12], regular stripes caused by fluorescent bleaching during knife cutting and imaging can be effectively removed by eliminating corresponding frequency domain information through a Gaussian notch filter. Nevertheless, the diversity in width and brightness of stripes in LSFM images can complicate the application of direct filtering as it may negatively impact the original quality of the image. Midway histogram equalization (MHE) [13] is a statistics-based technique utilized for stripe noise removal. MHE mitigates stripe noise disturbance by introducing redundant information between adjacent columns. However, it is effective in removing only slight levels of stripe noise in the image. Furthermore, statistics-based methods are highly restricted due to their strong similarity assumptions [10]. Currently, deep learning-based methods have been widely explored in image processing applications and have shown remarkable improvement in performance, including interactive segmentation [14], multi-scale segmentation [15–17] and other image segmentation methods, pattern recognition [18], and neuronal tracking in brain images [19]. Additionally, CNNs are widely used for image noise removal [20]. Motivated by the outstanding performance of deep learning, a three-layer stripe noise removal convolution neural network (SNRCNN) was proposed [21]. The network aims to remove stripes directly, serving as an image-denoising and super-resolution tool without taking into account the specific characteristics of the images. Therefore, it is hard to completely remove the stripe and retain the high-frequency textures perfectly. DnCNN [22] is one of the classical network models for removing noise, which involves teaching residual networks to noise points, thus replacing the need to look for a mapping correlation between damaged and clean images. Guan et al. proposed a wavelet deep neural network (SNRWDNN) model for removing stripe noise [23], leveraging the properties of the wavelet domain and fusing the information of the image itself to eliminate stripe noise without compromising image details [24]. Regrettably, this technique has lower performance in removing high-noise stripes with varying widths and brightness. In 2006, Hinton et al. proposed and published the encoder-decoder structure [25]. The primary aim of this design was to attain image compression and denoising. Inspired by these techniques and recognizing that the U-net model can learn multiscale information effectively, an attention-based residual network (Att-ResNet) with 2D U-net [26] as the backbone was proposed to effectively eliminate stripe artifacts in LSFM images [27]. This application verified the efficacy of the U-net model in addressing different stripe patterns, however 2D U-net presents incompetence in learning spatial information of stripes in 3D.

To overcome this challenging problem, we propose a novel method USRL (3D U-net-based Stripe Removal for Light-sheet microscopy image). This method aims to effectively eliminate 3D stripes present in neural data while minimizing potential damage to biologically relevant information. The process of stripe removal is depicted in Fig. 1 and consists of three main modules: data pre-processing, stripe removal, and data post-processing. Neural images obtained through LSFM often exhibit optical distortion, large dynamic variation ranges and uneven distribution. In the pre-processing stage, as described in Sect. 3.3, flat field correction and nonlinear mapping are employed to transform the images, ensuring performance of subsequent model application, as illustrated in Fig. 1A. Additionally, we pad the sides of the image volume and dissect it into overlapping tiles, preparing them for next step. Stripe removal module serves as the core component of our proposed method. As illustrated in Fig. 1B and C, USRL utilizes a dual-resolution approach to remove stripes across a wide range of scales. Thin and wide stripes are removed separately from USRL S1 and USRL S2, respectively. This dual-resolution approach is inspired by the observation of varying stripe widths in neural data and represents the key innovation of our method. The details are explained in Sect. 3.1 and Sect. 3.2. Lastly, the data post-processing stage (Sect. 3.4) involves stitching the tiles together to reconstruction the whole volume and reversal processing of nonlinear mapping to restore the images to their original grayscale characteristic space.

The main contributions of this paper are summarized as follows:

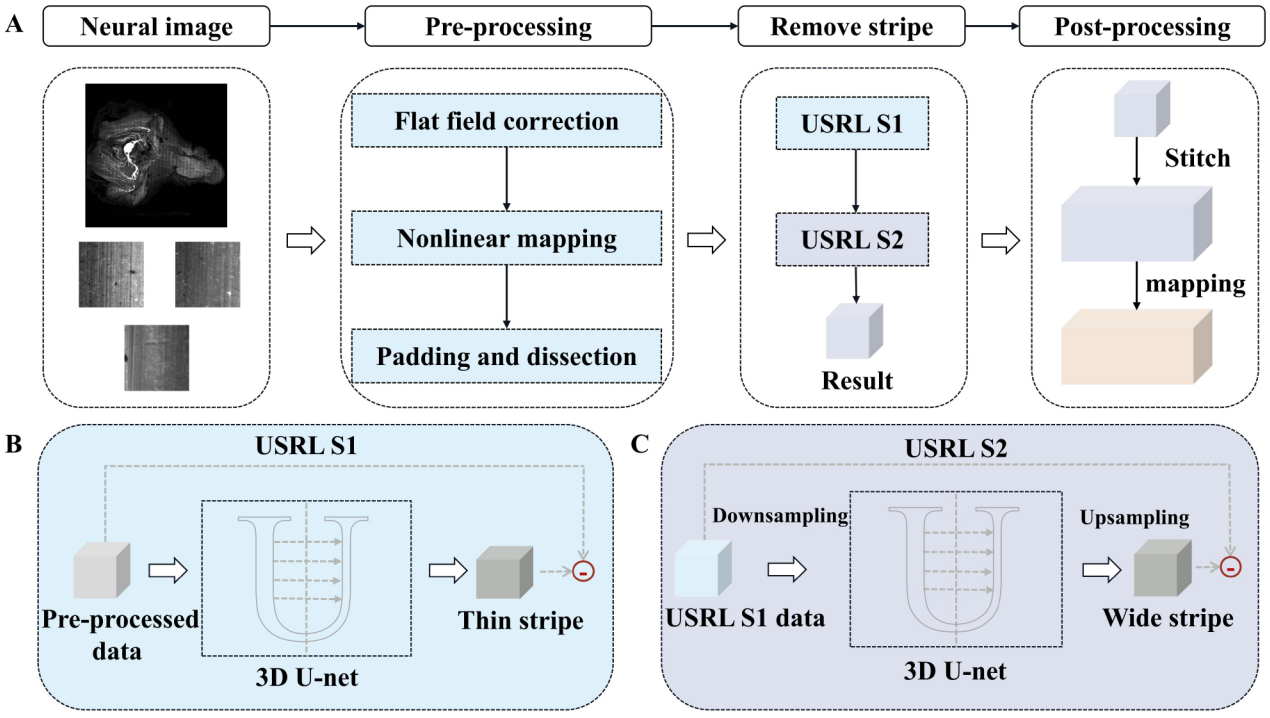

Fig. 1 Algorithm framework for removing stripe from mouse neural imaging data containing brain and surrounding tissue

1. Introducing the 3D U-net network model to effectively remove 3D stripe noise. Notably, our proposed method, USRL, pioneers the use of a 3D model for stripe noise removal in microscopy images. This advancement further substantiates the feasibility of employing a 3D encoder-decoder structure for noise removal.
2. The proposed dual-resolution approach demonstrates improvement of the performance of stripe removal from LSFM images, promoting the study of the anatomical structure and internal working of the brain and the use of LSFM.
3. Nonlinear mapping is employed to address the challenge of balancing the performance of stripe removal and the preservation of both details and light area information.

2 Data observation

To accurately estimate stripe noise in LSFM images, it is essential to learn the characteristics of such data. The Transparent Embedding Solvent System (TESOS) [28] neuroimaging data acquired by Inverted Selective Plane Illumination Microscopy, which we refer to as TESOS in this paper, is a representative LSFM dataset. The transparent embedding of cleared samples and imaging of adult mouse whole body sample using light sheet microscope for the acquisition of the original data is described in detail as in [28]. Stripes in LSFM images exhibit significant directional characteristics, as noted in [29]. Stripes in LSFM images are similar to other kinds of stripe noise, as visually illustrated in Fig. 2A, where stripe direction is within a minor deviation angle of 1 degree from the vertical. These stripes yield 3D negative light pillars, which are consistent with the artifacts generated in LSFM due to the high-absorption or scattering structures along the imaging light's pathway. We present a zoomed-in section of an image block mostly composed of stripes, displayed with intensity-inverted to offer a view of the 3D light pillars as presented in Fig. 2B. Stripe noise in LSFM images is diverse, as can be seen from the statistical analysis of a large number of TESOS data. This diversity primarily manifests in stripe width and brightness. In terms of width, we categorize stripes into thin stripes, which have strong characteristic information, and wide stripes, which have low characteristics and are contaminated with other stripes. Figure 2C depicts the distribution of stripe width in TESOS (voxel size at $0.58 \times 0.58 \times 1\ \mu m^3$), where the minimum width ranges from 1 to 2 pixels (as in X axis) and the maximum width exceeds 80 pixels. Most of the stripes are concentrated in 1 to 22 pixels (87.5 percentile of stripe width). Although the proportion of wide stripes is less, their impact cannot be ignored due to the broad coverage width. In terms of brightness, the range of intensity deviation from background is vast, ranging from -210 to -8 (negative due to dark stripes). Taking 95 percentile of the data intensity as reference I_R, they range up to 45% of I_R

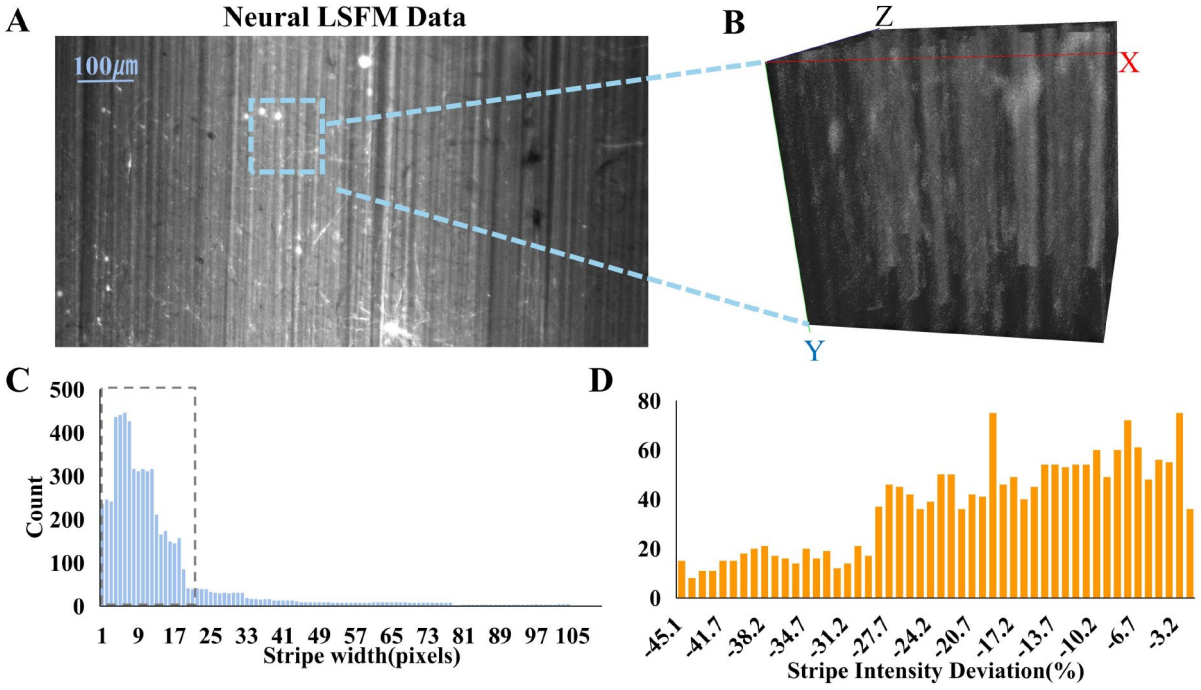

Fig. 2 Stripe statistics based on neural data observation. (**A**) Mouse neural data from TESOS (one *Z* slice). (**B**) 3D view of small volume displayed with inverted intensity. (**C**) Stripe width histogram. The box is the width ranges where the stripe is concentrated. (**D**) Stripe intensity deviation histogram

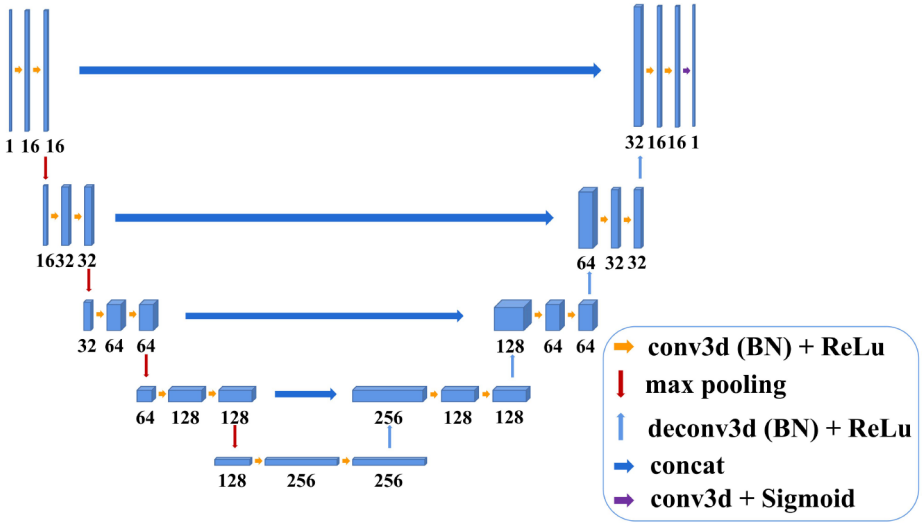

Fig. 3 Model architecture for removing stripe noise

, and the majority within 27% of I_R, as shown in Fig. 2D. In summary, the stripes in LSFM images take place in the form of 3D negative light pillars with varying width and brightness.

3 Methods

This section first outlines the network structure and the model targets and loss functions, as well as model training process. Then the data pre-processing including flat-field correction, nonlinear intensity mapping and padding and dissection are described in details.

3.1 USRL network architecture

The network structure employed for stripe noise removal is U-net [30, 31], which is a deep convolutional neural network with skip connections that has demonstrated its power in biomedical image processing tasks. The specific architecture of the model in proposed USRL is shown in Fig. 3. In contrast to existing deep learning-based stripe noise removal methods, USRL learns different scales of

stripes directly in 3D space. Furthermore, it includes an activation function in the output layer to improve the convergence speed.

In order to ensure that the parameters are reduced under the same sensing field, and the complexity of the model is reduced [32], the size of the convolution filter is set to $3 \times 3 \times 3$. In the encoding structure, we have 5 layers with each layer consisting of two $3 \times 3 \times 3$ convolutions and a rectified linear unit (ReLu), similar to general 3D U-net structure [30]. At the end of each layer, there is a $2 \times 2 \times 2$ max pooling with a stride of two in each dimension. The number of channels are doubled before the pooling process to avoid bottleneck [32, 33]. The input to the network is a $64 \times 128 \times 128$ voxels tile of the image with 1 channel. Our output in the final encoding layer is $4 \times 8 \times 8$ voxels in $\mathrm{Z} \times \mathrm{X} \times \mathrm{Y}$ directions respectively. In the decoding structure, each layer consists of $3 \times 3 \times 3$ up-convolution with a stride of two, followed by two $3 \times 3 \times 3$ convolutions, each of which is followed by a ReLu. In the decoding process, high-resolution features of the model at different scales are directly combined. The final layer consists of a $3 \times 3 \times 3$ convolution that reduces the number of output channels to the number of labels.

3.2 USRL training methods

Model training is the key to the performance of the model, this session systematically expounds the model training from the aspects of dataset generation, loss function and training strategy. Finally for optimal application in stripe removal, a concise explanation of the specific details involved in utilizing the trained model is provided.

3.2.1 Generate training set

Stripe noise in LSFM images is often understood as additive noise resulting from uneven illumination that is shown in Fig. 4. Degraded data (Fig. 4A) can be regarded as the sum of clean data (Fig. 4B) and stripe component (negative) (Fig. 4C), and the corresponding degradation model can be expressed as follows:

$$Y = X + S \tag{1}$$

where, $Y \in C^{D \times M \times N}$ represents image containing stripe noise, with D, M, N representing slice, row and column of the degraded data, respectively. The goal is to estimate the stripe component S from Y and subtract it from Y to obtain the desired clean data X, that is, $X = Y - S$.

To train a network model for stripe removal, we need training data containing the noisy data and also the clean data or the stripes. There is no 3D data set publicly available for such training tasks. Since stripe noise in LSFM images is additive noise that exhibits negative 3D pillars, our approach is to generate various pillar-shaped stripes and impose them to clean data to form the training and validation sets. The LSFM images from Osten lab [34] (the data set is referred to as OLST in this paper) is used as the clean data. To synthesize 3D light pillar as we observed, the first step is to synthesize the 2D section with uneven illumination using ellipses of varying shapes. The center locations and long and short axes are randomly generated for a variety of ellipses. Different centers are chosen to ensure the stripes to be spaced across our data set. The lengths of the long and short axis are randomly selected from 1 to 22 pixels. This range is chosen for two reasons. Firstly, stripe width is concentrated in this range (as shown in Fig. 2C). With model trained using synthetic data of this width range, thin stripes can be removed directly in the USRL S1 process. Additionally, the width of the wide stripes will also be transformed into the range of 1 to 22 after downsampling, so we can remove the wide stripes in the USRL S2 after removing the thin stripes. Secondly, the input to the network is $64 \times 128 \times 128$ voxels in $Z \times X \times Y$ directions during training, so if stripes with larger width are used for

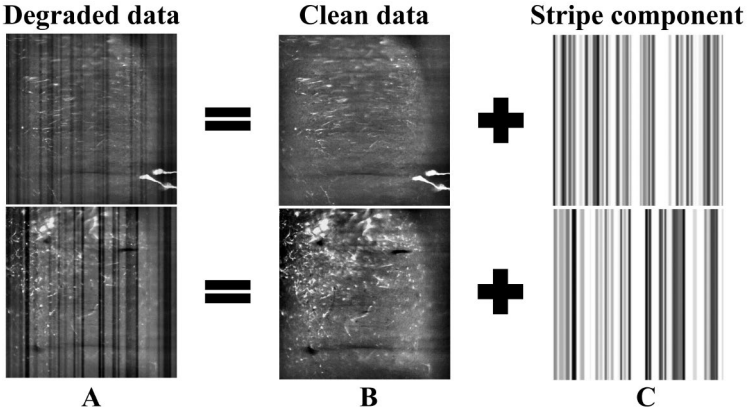

Fig. 4 (**A**) Degraded data. (**B**) Clean data. (**C**) Stripe component. **A** can be regarded as the sum of **B** and **C**

training, the portion of wide stripes occupation in the image is quite large, and also, we noticed the trained model may mistakenly classify a narrow clean area as a stripe.The intensity of the synthetic ellipses deviates from the background at the ranges from −72 to 0. With the intensity lower than 188 (95 percentile of the data) in general, this range is up to about 40% of the intensity. This is in analog to the data observation of TESOS data. Note that for both TESOS data and OLST, pre-processing is done before images being input to the network, as shown in Fig. 1. Intensity piecewise linear mapping changes the data distribution without changing the portion of stripe value in general signal. The principle and function of nonlinear mapping will be described in Sect. 3.3.2.

The preliminary 3D stripe component is constructed by duplicating the above 2D sections as Y slices so that the stacked 3D volume reaches the same Y slice count as the samples of neural data from OLST. To better conform to the characteristics of actual stripe noise, we apply Gaussian smoothing filters to generate the final reference stripes. Figure 4 displays sample stripes and the resulting degraded data in 2D Z slice form.

3.2.2 Model target and loss function

Clean data imposed with generated stripes gives the degraded data Y, which serves as the input of the network model. Instead of predicting the underlying clean image as other denoising models [35], USRL learns the stripe noise by training the residual mapping $R(Y) \approx S$, and hence obtain denoised image by $X = Y - R(Y)$. This approach has the advantage of focusing directly on the stripe information, including stripes that deviate only slightly from the background intensity. On the other hand, LSFM with sparse labelling has foreground signals which are much brighter than the rest of the regions, hence learning directly the clean data degrades model's ability to learn from darker regions due to brighter areas being relatively more influential than darker regions. Learning the clean data also results in limited performance in the removal of the stripes with small deviation from background intensity.

The average mean squared error (MSE) between the expected S and the estimated data $R(Y)$ from the noisy input can be calculated as follows:

$$l(\theta) = \frac{1}{N^2} \sum_{i=1}^{N} ||R(Y_i; \theta) - S_i||^2 \quad (2)$$

where N represents the number of pairs of stripe-clean data in the training data set, which is the amount of data inputted into the network in each batch. This MSE is used as the loss function to learn the trainable parameters θ in the model.

3.2.3 Training strategies

The training data consists of 40 sets of data generated as above, and the validation data consists of 10 sets of data. Each set of data has dimensions of $Z \times X \times Y$ corresponding to $64 \times 512 \times 512$ voxels. During training, the data directly input into the model has $64 \times 128 \times 128$ voxels. We perform data augmentation, including rotation and flipping, to enhance the generalization ability of model, which results in 25,000 $patches$. The $batch\_size$ is set to 10. For each training iteration, we select a random set of ten pairs of stripe and clean data to input into the model. We conduct 50 $epochs$ of training, with an initial learning rate of 0.001 that is reduced to 0.0001 after the first ten epochs. The training process, completed on a single 3090 GPU, lasted approximately 1.5 days.

3.2.4 Model application

In our proposed algorithm, the trained 3D U-net model is applied to data with the dual-resolution approach. The data with a size of $64 \times 512 \times 512$ is input to the model. The thin stripes and wide stripes in the data will be sequentially removed in USRL S1 and USRL S2. As shown in Fig. 1, USRL S1 outputs an intermediate result with thin stripes removed and wide stripes remains. In USRL S2, the data is downsampled by a factor of 4, resulting in the transformation of the width of remaining stripes to a range of 1 to 22. Subsequently, the U-net model estimates the wide stripes accordingly, which are then upsampled to match the original input size and removed from the outputs of USRL S1 to generate the final clean LSFM image.

3.3 Data pre-processing

The significance of data pre-processing is evident in shaping the features of the Unet model within USRL. This section systematically explains the principles and specifics of key techniques, including flat-field correction, nonlinear intensity mapping, as well as padding and dissection, employed in the data pre-processing stage.

3.3.1 Flat field correction

Vignetting [36] is a phenomenon that commonly occurs with LSFM images resulting from the large angle between the optical axis of the lens and imaging light in the corner. It is commonly accompanied by stripes and degrades the image quality by causing darker corner brightness compared to the center [37]. Figure 5A shows the impact of vignetting on TESOS data. To ensure accurate training of the network model, it is important to correct the contrast imbalance of data before removing stripe noise. Vignetting can lead to the loss of stripe feature information in the vignetting region due to lower intensity deviation, while the variation in stripe widths may cause the network model to mistakenly learn vignetting as a potential

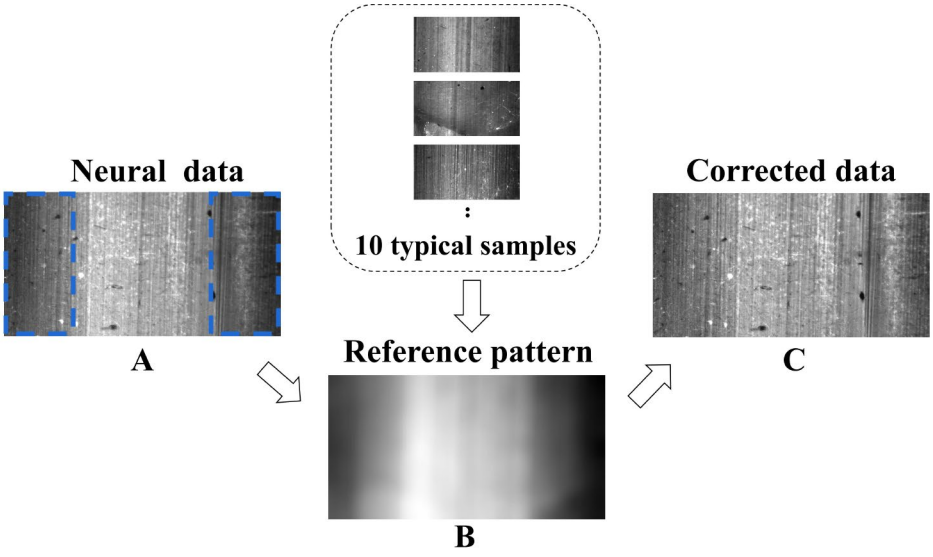

Fig. 5 The process of flat field correction. (**A**) Neural data. The blue boxes show the main area of vignetting influence. (**B**) Reference pattern image is constructed from 10 typical samples. (**C**) The data after flat field correction

source of wide stripes [38]. The LSFM data has diverse distribution, hence if each image goes through vignetting removal independently with different parameters, the resulting adjacent image tiles could inconsistency and the performance on sample corners might be poor. To solve this problem, we selected a set of typical samples and compute the unique reference pattern image R by averaging their Gaussian smooth filtering results, as shown in Fig. 5B. It can be expressed by the following:

$$R = \frac{1}{M}\sum\nolimits_{i=1}^{M} G(I\prime_i, K, \sigma) \tag{3}$$

where G is a Gaussian smoothing function. K and σ are Gaussian kernel and the standard deviation, which are taken as 255 and 64 respectively, to ensure a strong smoothing effect. M is the total number of typical samples, which we set as 10. I_i represent the 2D center z-slice image of the mouse neural data and the size of I_i is 2048×1024. Furthermore, data intensity distribution is uneven, with very bright voxels in the sparsely labeled signal regions (which will be described in detail in Sect. 3.3.2), flat field correction causes excessive sharpening of data due to the influence of large intensity regions. Therefore, I_i is adapted into $I\prime_i$, where voxels with intensity greater than 95 percentile of the data is assigned new intensity as the mode intensity of the data. With reference pattern image calculated, flat-field correction was performed with a division for all 2D slices of the image volumes. Figure 5C shows example result of flat field correction.

3.3.2 Nonlinear intensity mapping

The neural data collected from TESOS demonstrates a wide range of dynamic changes and an uneven distribution. Figure 6A illustrates the extensive dynamic range of brightness levels, spanning from a minimum value of 140 to a maximum value of 13,917. However, the majority of brightness values fall within the range of 146 to 460. Consequently, this concentration of low values diminishes the feature information of the stripes after linear normalization and redirects the focus of learning towards the brighter regions, rather than the noise present in the stripes. Additionally, the wide range of dynamic changes in the data contributes to the significant variations in stripe brightness, thus further exacerbating the challenge of training models for the analysis of stripes.

Hence, it is essential to perform nonlinear mapping subsequent to flat field correction in order to ensure the preservation of complete stripe characteristic information after normalization. This approach not only enhances the learning process of information from relatively dark regions but also ensures the preservation of intricate details. As depicted in Fig. 6B, nonlinear mapping is accomplished by utilizing piecewise linear mapping. This mapping technique assigns the first 95 percentile of the voxel intensity T to the range of $[0,255 \times 0.95]$, while the remaining 5% of the voxel intensities are mapped to the range of $[255 \times 0.95,255]$. The range cutoff of 95 percentile is selected due to the observation that the majority of sparsely labeled neural data tends to have a constrained amount of bright signal. This transformation serves the purpose of converting the uneven distribution of the data to a more even normal distribution without causing any harm to the original data. The outcome of this process

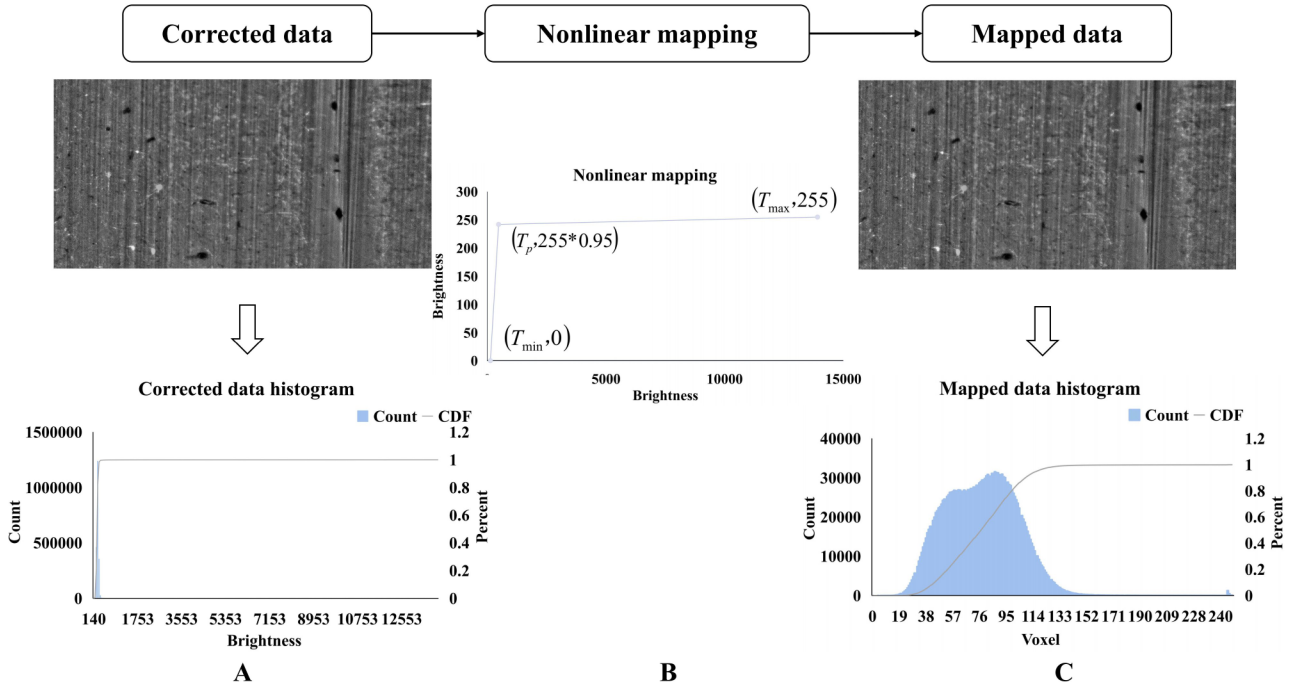

Fig. 6 Nonlinear mapping process. (**A**) The data after flat field correction and its brightness histogram. (**B**) The description of nonlinear mapping, which is realized by piecewise linear mapping methods. (**C**) Nonlinearly mapped data and its brightness histogram

is the generation of mostly normally distributed nonlinear mapped data, which can be observed in Fig. 6C. The inverse nonlinear mapping procedure can be employed to restore the data within the aforementioned intervals back to the original data interval, which is achieved by applying the inverse nonlinear mapping function. The specific mathematical formula for the nonlinear mapping is presented below:

$$T_i\prime = \begin{cases} \frac{T_i - T_{min}}{T_p - T_{min}} \times (255 \times 0.95), T_{min} \leq T_i \leq T_p \\ \frac{T_i - T_p}{T_{max} - T_p} \times (255 \times 0.05) \\ \quad + (255 \times 0.95) \end{cases}, T_p < T_i \leq T_{max} \quad (4)$$

where $T\prime$ is the mapped neural data and T is neural data from TESOS. T_{min}, T_{max}, T_p correspond to the smallest brightness, largest brightness, and the 95 percentiles of all brightness in T. It is noteworthy that the application of nonlinear mapping expands the versatility of our method, making it applicable to LSFM images with different variations.

3.3.3 Padding and dissection

When the trained 3D U-net is utilized to process large 3D volumes as TESOS data, two additional processes are required. Convolutional operations on input images often lead to a reduced number of voxeloperations at the boundary region, resulting in decreased recognition performance in that area. This limitation is particularly pronounced in top and bottom Z slices. To improve the removal of stripes in boundary regions, images are preprocessed using a filling technique prior to the actual processing [39]. This technique, as depicted in Fig. 7A, involvessymmetrically padding boundary voxels of the volumes. Although convolution reduces the resolution of feature maps, it is expected that the boundary points of the data can be preserved until the final layer of feature maps. Taking into account the number of pooling operations in the model, computational performance, and the effect of padding size on improving the performance of removing boundary stripes in practical operations, padding using mirroring with width of 128 voxels are symmetrically made in the X and Y direction, followed by padding with width of 32 voxels in the Z direction. Figure 7A displays the padding used to obtain additional voxel values outside of the original data in the grey box.

Additionally, considering the efficiency of data processing and the available memory capacity for computing, the neural data is processed in tiles of $64 \times 512 \times 512$. Since the voxel of the boundary region generated by partitioning will face the same problem as the original data as stated above, the method of overlapping tiles is adopted to solve this problem. Padding and dissection with overlapping tiles in the X axis is demonstrated in Fig. 7A and B. With the Z axis $stride$ set as 32, and X/Y axis $stride$ set as 256, the whole volume is dissected and the resulting tiles have 50% overlap in each direction. The tiles are put through USRL and the center region of the prediction

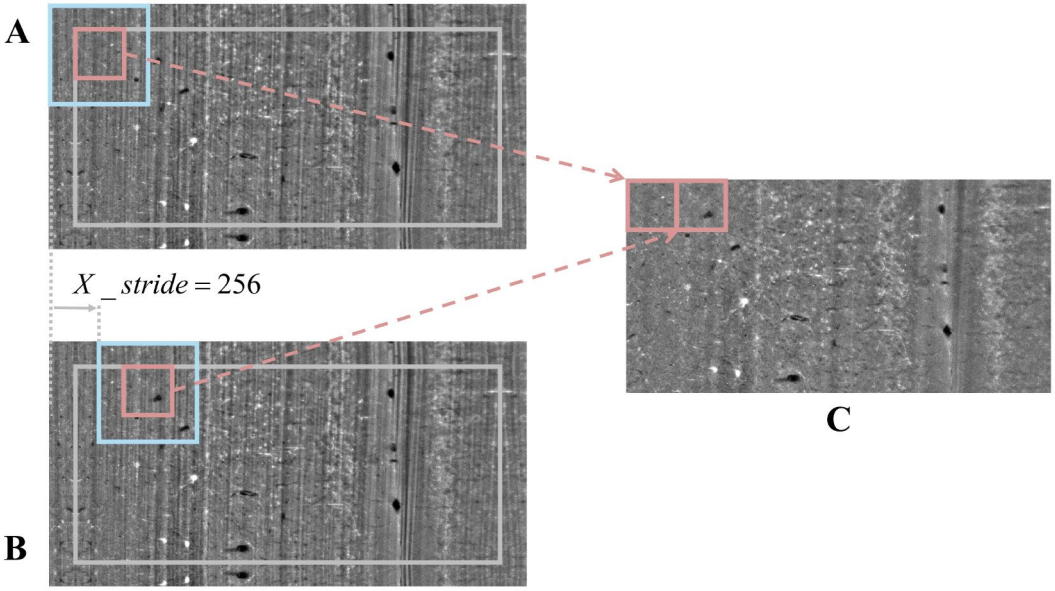

Fig. 7 The padding and dissection process. The 3D volume (in light grey box) is padded on all boundaries (shown in the XY-slice here). (**A**) The block in the blue box is cropped out for stripe removal in the center region (in the red box). (**B**) Another block adjacent to the block in **A** is dissected out for model application. (**C**) The outputs are stitched together to form the complete USRL results

(50% in the center in all directions and within the original volume region) are kept and are sequentially seamlessly stitched together into the whole output volume, as depicted in Fig. 7C.

3.4 Data post-processing

The data post-processing module encompasses two key steps: image stitching and the reversal of nonlinear mapping.

Image stitching involves reassembling the image, which has undergone stripe removal, to match the original data alignment. This is done by following the same dissection sequence used during pre-processing. Because of the carefully designed padding and dissection strategy, simple stitching without blending techniques suffices to achieve accurate alignment.Then the inverse of the nonlinear mapping applied during pre-processing is applied to the stitched images. This step is crucial for restoring the image to its original intensity distribution. Together, these post-processing steps complete the algorithmic process.

4 Experiments

4.1 Analysis of feasibility

Prior to comparing the proposed method USRL with other classical algorithms, the feasibility of USRL is first assessed on the validation set. Specifically, the consistency of the data before and after stripe removal is analyzed, as well as the amount of stripe remaining in the results. These evaluations provide an intuitive assessment of the performance of the proposed algorithm.

From the results obtained on the validation set, we can see that the removal of stripes by USRL leads to a satisfactory improvement in the quality of the images. Figure 8AB show two sets of visual examples of the data passing through the dual-resolution stripe removal framework. After the application of USRL S1, the majority of the thin stripes are removed, with some residue stripes primarily associated with wide stripes, especially ones with high magnitude, as shown in the red boxes in Fig. 8AB. With the subsequent application of USRL S2, the remaining stripes are eliminated in the final results. The results provided visual evidence for the effectiveness of the proposed dual-resolution approach.

In quantitative evaluation and analysis, data distribution similarity and changes are investigated. Figure 8C provides the comparison of USRL S1 and USRL S2 processes in terms of how they affected the average brightness of the stripy image. With the USRL S1 and USRL S2 process, the processed data's average brightness converges toward that of the clean most of the samples (9 out of 10, with the exception of validation 10 data). As shown in Fig. 8D, the standard deviation of most of data after the removal of stripes also gets much closer to that of the clean data when compared to the degraded data, which suggests that the data after USRL is remarkably similar to the clean data distribution. These quantifications suggest that the damage sustained from full USRL is at an acceptable level. Processing thin and wide stripes sequentially has proven to be credible.

Furthermore, a direct comparison is conducted between the remaining stripe information before and

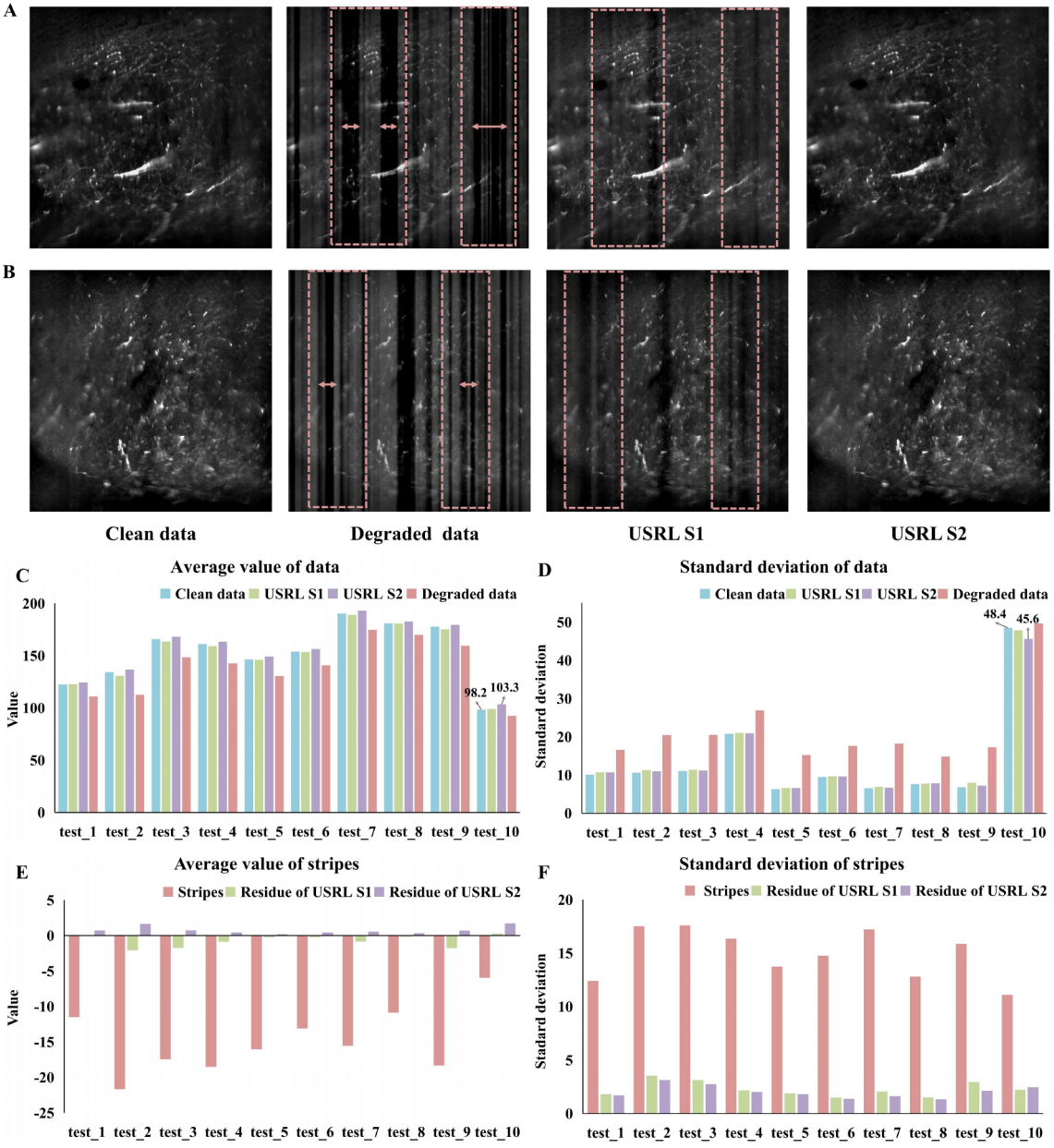

Fig. 8 The similarity and progress of the data before and after stripe removal on validation set. (**A**, **B**) are visual examples. The first column is clean validation data, the second column is degraded data, the third column is the processing result of USRL S1, and the fourth column is the processing result of USRL S2. The red boxes are wide stripes contaminated with other thin stripes. (**C**) Average brightness before and after stripe removal. (**D**) Standard deviation before and after stripe removal. (**E**) Average value of stripe information. (**F**) Standard deviation of stripe information

after stripe removal. The average value of the stripe information is calculated and present in Fig. 8E. Before removal, the average value of the stripe information is significantly negative, due to the strong negative light pillars. Theoretically, successful stripe removal would result in the value of the stripes approaching zero. As depicted in Fig. 8E, the value of residue in the USRL S2 results is very close to the theoretical value, 0. Across all test sets, the overall average value of the residual in the USRL S2 results is closer to zero than the residue in the USRL S1 results. Figure 8F provides a more distinct illustration of the residue in the USRL S2 results, where the standard deviation of stripes is closer to 0 than in the USRL S1 results. The standard deviation of the stripe decreases progressively through the processes, and the direct depiction of the stripe information confirms the feasibility of sequentially removal of thin and wide stripes.

4.2 Analysis of USRL structure

There are generally two perspectives for analyzing the rationality of an algorithm. One approach involves conducting ablation experiments [22, 23], while the other entails feature analysis, as exemplified in [40]. For USRL, image pre-processing procedures introduce alterations to the features in the U-net model. The analysis of the USRL structure incorporates the effects of these pre-processing procedures, coupled with a standard ablation experiment focusing on the skip connection within themodel structure. This dual-perspective analysis is discussed in-depth and assessed through experiments to ascertain the optimal preprocessing steps and model structure.

4.2.1 Effectiveness of nonlinear mapping

Performing direct linear normalization on LSFM images with a large dynamic variation range and uneven distribution causes the model to focus on brighter areas during the learning process. A notable observation is the increase in damage of useful information while removing stripe noise, particularly in the detailed structure of bright or dark areas. Bright areas tend to spread out after stripes removal, and due to reduced stripe characteristics, whereas dark structural areas are blurred or even removed as potential stripes. To further substantiate the necessity of non-linear mapping, we make a comparison of the reconstruction results with and without nonlinear mapping.

As depicted in Fig. 9, we present a comparison between the local structure of the data from TESOS and the outcomes of stripe removal with and without nonlinear mapping processing. Specifically, when examining the zoomed-in 3D tri-views of the region within the blue box, it can be observed that the data after stripe removal with nonlinear mapping exhibits a high resemblance in terms of 3D spatial characteristics to the bright areas in the original TESOS data. Conversely, stripe removal without mapping leads to significant damage to the bright areas, exhibiting a diffusion, which can potentially adversely impact subsequent image processing tasks, such as image segmentation and data collection evaluation. Additionally, as shown in the red boxes in Fig. 9, nonlinear mapping not only promotes the uniform distribution of data and reduces the attention of model to bright areas, but also enhances model's ability to learn information from relatively dark regions and effectively preserves image detail information.

4.2.2 Effectiveness of padding and dissection

The inclusion of padding and dissection plays a crucial role in improving the performance of boundary stripe removal. To examine the effectiveness of this module, a comparison is conducted between modules with and without padding and overlapping tile. As described in Sect. 2, stripes exhibit distinct directional characteristics, primarily appearing as vertical stripes in the XY plane and elliptical-shaped dark areas in the XZ plane. When using unpadded data, the stripe noise in the boundary areas is not removed entirely, as indicated by Fig. 10A. More specifically, the striped area within the red region still exhibits degraded image quality, and in the blue region, the remaining stripes create distinct shadows in the XZ plane, resulting in irregular local brightness variations. Conversely, when utilizing padded data, the performance of stripe removal in the boundary area is markedly enhanced, as depicted in Fig. 10B.This

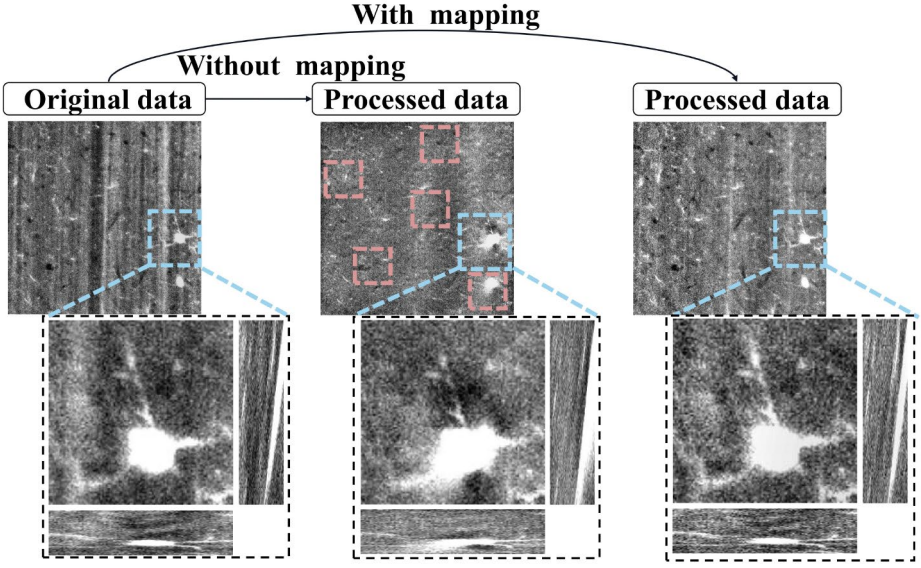

Fig. 9 Comparison of processed data obtained by model with and without nonlinear mapping. The red boxes are for several areas where the data structure is damaged. The blue box is for region with detailed and bright structure. Images in blue box are displayed in zoom-ins with 3D tri-views

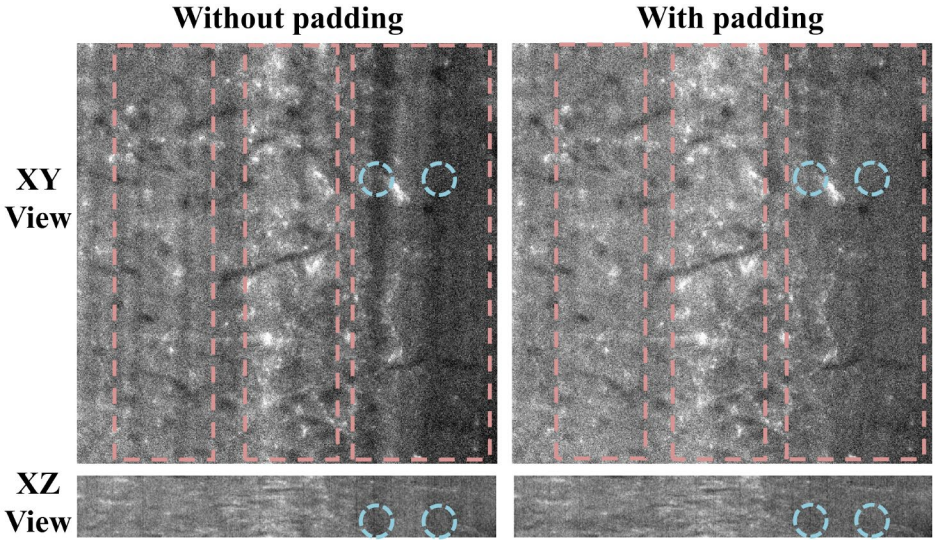

Fig. 10 Padding and overlapping tiles improve the performance of removing boundary stripes. (**A**) Stripe removal results without padding. (**B**) Stripe removal results without padding. The red boxes are for the regions influenced by stripes on the bottom z-slices as shown in the XY view. The blue circles centered at locations for regions with obvious phenomenon of stripe projected on the XZ view

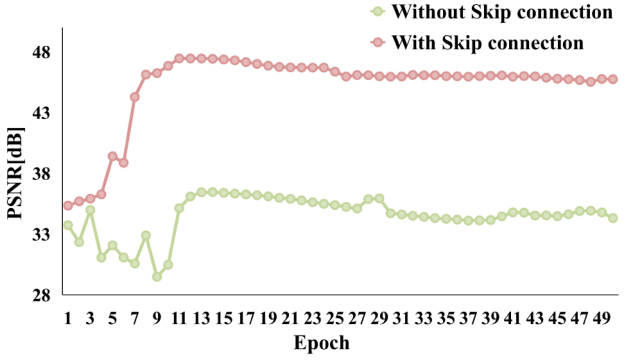

Fig. 11 Effectiveness of skip connection. The PSNR of the models with stripe removal in each epoch are plotted. The red line for the model with skip connection and green line for model without skip connection

improvement has successfully reduced the interference caused by stripes, seen both in the XY and XZ planes.

4.2.3 Effectiveness of skip connection

The skip connection merges feature maps at corresponding positions in the channel. By fusing the low-level and high-level feature maps, the network preserves more high-resolution details contained in the high-level feature maps, which helps to extract features of various resolutions. To evaluate the effectiveness of the skip connection, we made a version with skip connection removed from the proposed model [41]. Then, the same stripe noise removal task was performed on the same test dataset to compare the performance of the models with and without skip connection. To access the stripe noise removal performance, a number of metrics are calculated to evaluate image quality. Commonly used metrics include mean square error (MSE) [42], peak signal-to-noise ratio (PSNR) [43] and structured similarity indexing method (SSIM) [44], among which PSNR is the most common and widely used objective measurement method to evaluate image quality after noise removal. The curves of PSNR for the two models are used as the performance evaluation metric. Figure 11 presents a comparison of the learning curves for each epoch. As can be observed, the model without skip connection has greater fluctuation in learning performance and converges at a slower rate compared to the model with skip connection, which converges ore stably and faster. Furthermore, the model with the connection structure demonstrates a 10dB improvement in the corresponding evaluation metric throughout the entire training process. In conclusion, skip connection is a powerful and effective way to achieve stable convergence.

To further demonstrate the necessity of the skip connection, we test the stripe noise removal ability of the two models on the validation set. As shown in Table 1, the stripe removal performance of the model with skip connection is better than the model without skip connection for each test data, with the minimum PSNR improvement of 6dB, demonstrating the significant impact of the

Table 1 PSNR values of stripe noise removal results in validation set obtained by model with and without skip connection

	test_1	test_2	test_3	test_4	test_5	test_6	test_7	test_8	test_9	test_10
USRL	**44.45**	**37.00**	**39.93**	**41.30**	**42.64**	**45.75**	**43.62**	**45.19**	**41.02**	**34.33**
w/o connection	33.70	26.88	25.92	35.08	28.29	34.32	34.43	34.89	27.65	24.60

Table 2 PSNR of the results of different stripe removal methods for the validation set

	mBA*	SNRCNN	SNRWDNN	DnCNN	2D U-net	USRL
test_1	26.87	27.54	27.81	35.41	42.02	**44.45**
test_2	20.30	26.87	30.27	33.65	**39.43**	37.00
test_3	21.72	27.58	32.32	35.79	39.31	**39.93**
test_4	21.13	30.60	32.03	34.84	39.67	**41.30**
test_5	23.76	28.60	32.29	36.44	38.97	**42.64**
test_6	25.07	27.04	30.71	35.31	41.02	**45.75**
test_7	23.03	31.72	32.72	37.71	26.90	**43.62**
test_8	25.57	30.90	29.81	33.51	40.58	**45.19**
test_9	24.60	29.05	33.81	34.90	**42.94**	41.02
test_10	23.99	22.70	24.16	24.28	27.02	**34.33**

*: mBA is short for mBrainAligner

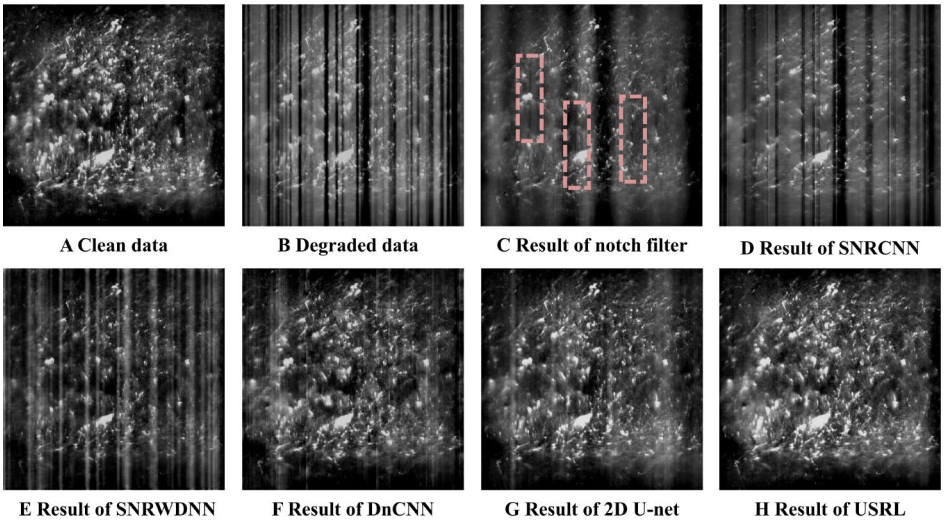

Fig. 12 Effectiveness of different striping methods in validation data. (**A**) Clean data. (**B**) Degraded data. (**C**) Result of notch filter. Image information in the red boxes is reduced. (**D**) Result of SNRCNN. (**E**) Result of SNRWDNN. (**F**) Result of DnCNN. (**G**) Result of 2D U-net. (**H**) Result of USRL

skip connection module in improving the performance of stripe noise removal.

4.3 Comparison with previous methods

In this section, quantitative and qualitative evaluations of the performance of the proposed model and existing methods are presented.

4.3.1 Comparison on synthetic dataset

In this study, we evaluated the effectiveness of using USRL for removing synthetic stripe noise in the neural data and compared it with the performance of various algorithms, including mBrainAligner [12], SNRCNN [21], DNCNN [22], SNRWDNN [23], and 2D U-net [26]. The PSNR values of the results obtained by applying these algorithms to the validation set are presented in Table 2. In general, the performance of deep learning-based methods is better than that of digital filtering technique(mBrainAligner). The performance SNRCNN sometimes is lower than that of digital filtering due to its limited learning ability for complex and variable stripes. DnCNN performs better than SNRWDNN and SNRCNN. The stripe removal outcomes obtained by the U-net network frameworks outperformed all other algorithms, indicated by being highlighted in bold. It is worth noting that in two test cases, the performance of the 3D USRL model was slightly lower than that of 2D U-net-based methods. This variation can be attributed to the complexity of the spatial structure in certain instances. However, overall, the performance of USRL in stripe removal generally surpasses that of 2D methods.

A set of visual examples illustrating the outcomes achieved by different stripe removal algorithms is depicted in Fig. 12. Clean data (Fig. 12A) imposed with generated stripes gives the degraded data (Fig. 12B), which serves as the input of the network model. The digital filtering employed in mBrainAligner reduces image information that shares the same frequency as the stripes when eliminating irregular stripes in LSFM images. This reduction in structure is evident in the red boxes displayed in Fig. 12C, thereby diminishing the similarity between the cleaned data and the original clean data.

SNRCNN exhibits a limited capacity for generalization and is only capable of removing simple thin stripes, as illustrated in Fig. 12D. It struggles to handle stripes with diverse morphology and brightness. SNRWDNN shows good recognition of stripes; however, a closer examination of Fig. 12B and E reveals that its performance degrades when faced with denser and wider stripes. DnCNN exhibits less effective recognition of the directional characteristics of stripes, resulting in notable smoothing in the vertical direction, as illustrated in Fig. 12F.

U-net models have shown excellent performance in removing stripes in both 2D and 3D domains as in Table 2 and Fig. 12GH. When selecting the dimensionality of the model for the stripe removal task, we considered two essential factors. Firstly, LSFM captures neural data and stripes in a three-dimensional format. Therefore, the inclusion of 3D spatial information is critical for the model to effectively learn and comprehend the characteristics of the stripes within the data. Secondly, attempting to learn and remove three-dimensional stripes using 2D slices can result in the loss of important spatial information. To ensure the preservation of spatial details, it is imperative to employ a three-dimensional approach for efficiently addressing stripe removal.

To compare the performance of 2D and 3D U-net models directly, an investigation into the local spatial information of the data and an assessment of the resulting structure's continuity in the XZ views were conducted, as shown in Fig. 13A. When comparing the results obtained using 3D USRL for stripe removal with the clean data, a high level of resemblance was observed. The intensity consistency and continuity between adjacent slices were also evident. Conversely, the differences between the results of the 2D U-net-based stripe removal and the clean data were more pronounced in the XZ views. Notably, abrupt changes occurred between adjacent Z slices as highlighted by the blue circles in Fig. 13B. This discrepancy can be attributed to the fact that the 2D U-net model lacks training with 3D spatial information. As a result, while the 2D U-net model may achieve similar PSNR to the 3D U-net model during validation, it detrimentally affects the spatial structure of the data.

4.3.2 Validations on real mouse neural data

To assess the practical effectiveness of USRL, we performed evaluations on real neural data and compared its performance with other stripe removal methods. Figure 14 provides a visual comparison of the results obtained from various stripe removal techniques. Note that Table 2 indicates that deep learning-based methods outperform digital filtering techniques, hence our focus in this experiment centered exclusively on comparing deep learning-based methods. In Fig. 14A-E, we present the original data and the results of stripe removal using SNRCNN, SNRWDNN, DnCNN, and our USRL method, respectively. It is evident that SNRCNN, SNRWDNN, and DnCNN are capable of identifying stripes within the data. However, SNRCNN's recognition capabilities are limited due to its shallow architecture as it can only learn thin stripes, as illustrated in Fig. 14B. While SNRWDNN exhibits strong suppression of stripes with accurate localization, the visual effects of its results are poor. This is attributed to its unfavorable learning of stripe intensities, resulting in increased brightness within the stripe areas beyond the background brightness, as shown in Fig. 14C. DnCNN performs better than SNRCNN and SNRWDNN but still tends to smooth the vertical direction and exhibit

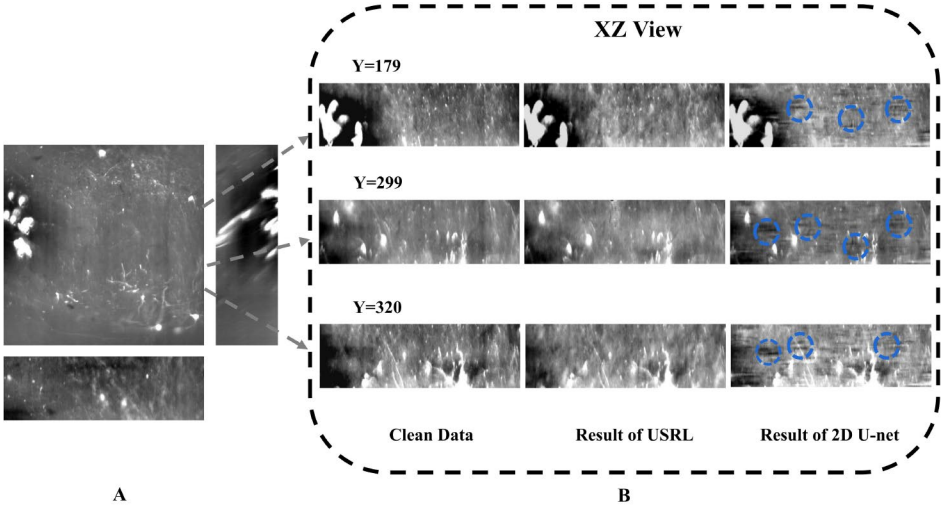

Fig. 13 Comparison of spatial information between the USRL result and the 2D U-net-based stripe removal result. (**A**) Tri-view of an image tile. Three y locations are selected to be viewed in (**B**). (**B**) XZ views of the clean data, the result of USRL, and the 2D U-net-based stripe removal result at the three y locations indicated in (**A**). The blue circles are for regions that display spatial information incoherence in the 2D U-net-based stripe removal result

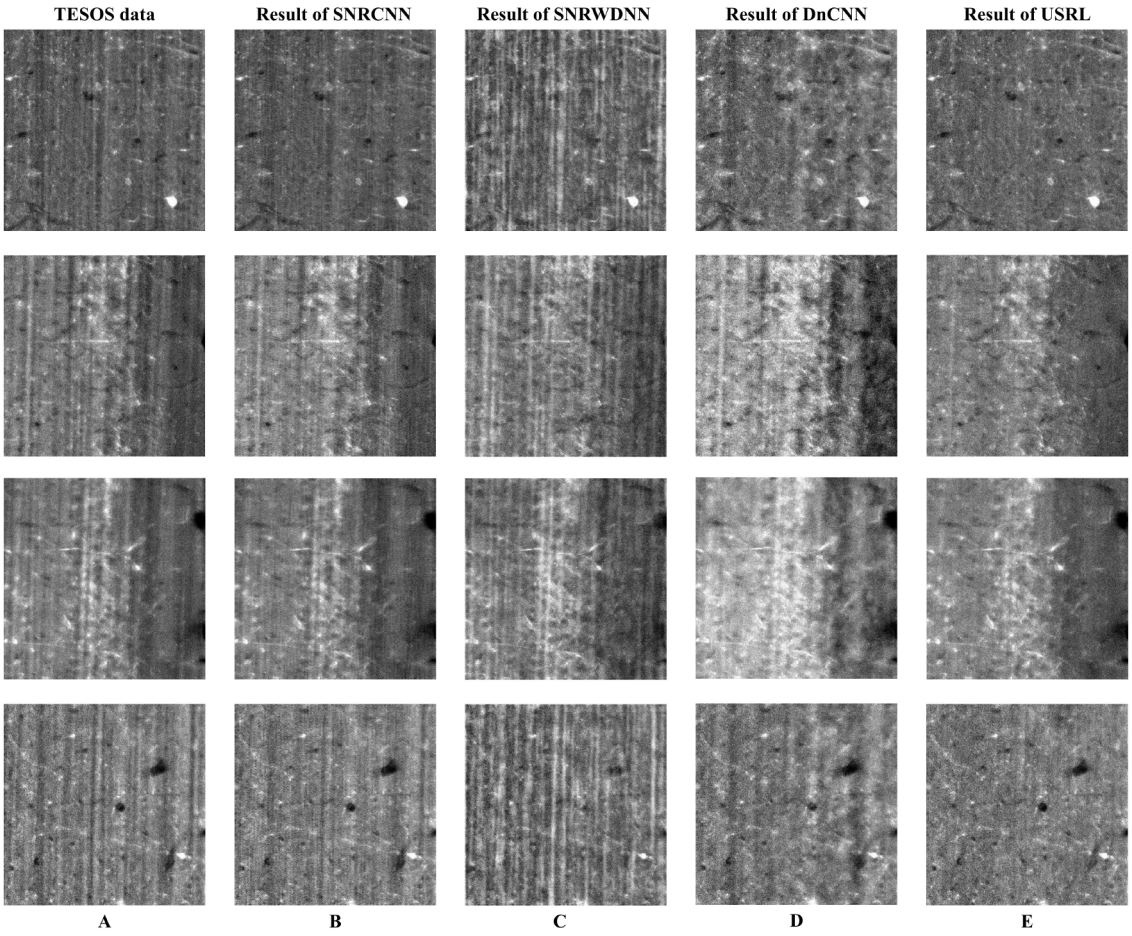

Fig. 14 Effectiveness of different striping methods in real data. (**A**) Neural data from OSLT_TESOS. (**B**) Result of SNRCNN. (**C**) Result of SNRWDNN. (**D**) Result of DnCNN. (**E**) Result of USRL

pixelated artifacts, as shown in Fig. 14D. This is primarily due to its design as a Gaussian noise removal model, lacking directionality in stripe removal. In contrast, our proposed USRL method directly learns and captures 3D stripes of varying scales and intensities, leading to superior reconstruction results. By utilizing this approach, we achieve optimal performance in stripe removal while preserving the quality of the image.

Hence, it can be concluded that the 3D U-net model is superior to the 2D U-net model in effectively removing 3D stripes from mouse neural data.

5 Conclusion

In this study, a novel dual-resolution approach based on the 3D-Unet model is proposed to effectively remove stripes in LSFM images. Unlike existing methods, the proposed method USRL directly learns and addresses stripes in the 3D space. Moreover, the approach utilizes pooling operations to model stripes at various resolutions. The sequential concatenation of two models at two resolutions boosted the usability of the removal for stripes with big range of width. Our method incorporates a nonlinear mapping technique to normalize data with high dynamic range and uneven distribution. This pre-processing step expanded the applicability of the USRL. Experimental results demonstrated the efficacy of our proposed method in effectively removing stripe noise and enhancing visual perception. Considering the promising outcomes, the proposed algorithm framework can serve as a valuable preprocessing tool to improve the performance of various subsequent applications such as neuron reconstruction, image classification, and signal processing [45, 46].

Currently, the limitation of the model is that the thin and wide stripes must be removed separately due to constraints posed by a restricted training dataset and limited model input data sizes. In addition, the performance of the stripe removal model decreases when applied to darker images. In the future, the research team will enhance the performance of the model for wide stripes and dark images by increasing the diversity of training data and incorporating additional normalization

and masking techniques. On the other hand, the proposed USRL model's utility can be extended beyond stripe removal in LSFM. It could be adopted to remove noise from various types of microscopic images, such as fMOST, and remove stripes from images in other fields, such as infrared images and medical images. USRL can thus facilitate biological research on a broader range of data.

Acknowledgements
Southeast University supported the algorithm development at the Institute for Brain and Intelligence. We thank Dr. Hanchuan Peng for his support and making the connection of the collaborators of the work. We thank Osten lab in Cold Spring Harbor Laboratory, USA, for providing OLST image data set as our training data. We thank Lei Qu and Yuanyuan Li at Anhui University for the discussion on existing methods.

Author contributions
L.D. conceptualized the project and developed the algorithm with help from the group. C.L. was the primary contributor to software implementation, experiments and manuscript writing. Y.L. and H.Z. provided the image data and contributed to algorithm design and discussion. L.D. provided guidance and provisions during the manuscript writing process. All authors have read and approved the final manuscript.

Funding
This work was funded by NSFC-Guangdong Joint Fund-U20A6005 and Fundamental Research Funds for the Central Universities (2242022R10089) to Liya Ding. The work was also supported by a Southeast University neuroscience initiative awarded to Dr. Hanchuan Peng.

Data availability
USRL is written in Python and is available on GitHub for public access and utilization, and the user can obtain the code from https://github.com/ChangshanLi/StripeRemoval/tree/master. Data for training and testing available https://github.com/ChangshanLi/StripeRemoval/tree/master/data. Original whole TESOS data is available upon request due to the very large data size of ~70 TB. Users need to ensure the existence of Pytorch 1.12.1, Python 3.9 including packages such as SimpleITK, Aicsimageio, and Numpy to guarantee USRL execution.

Declarations

Ethics approval and consent to participate
All animal experiments were approved by the Institutional Animal Care and Use Committee of the Chinese Institute for Brain Research.

Competing interests
The authors declare no competing interests.

Received: 27 June 2024 / Accepted: 19 August 2024
Published online: 26 September 2024

References

1. Dodt H-U et al (2007) Apr., Ultramicroscopy: three-dimensional visualization of neuronal networks in the whole mouse brain, *Nat Methods*, vol. 4, no. 4, pp. 331–336, https://doi.org/10.1038/nmeth1036
2. Kirst C et al (2020) Mapping the Fine-Scale Organization and plasticity of the Brain vasculature. Cell 180(4):780–795 .e25, Feb. https://doi.org/10.1016/j.cell.2020.01.028
3. Silvestri L, Bria A, Sacconi L, Iannello G, Pavone FS (2012) Confocal light sheet microscopy: micron-scale neuroanatomy of the entire mouse brain, *Opt. Express*, vol. 20, no. 18, p. 20582, Aug. https://doi.org/10.1364/OE.20.020582
4. Ueda HR, Dodt H-U, Osten P, Economo MN, Chandrashekar J, Keller PJ (May 2020) Whole-brain profiling of cells and circuits in mammals by tissue Clearing and Light-Sheet Microscopy. Neuron 106(3):369–387. https://doi.org/10.1016/j.neuron.2020.03.004
5. Verveer PJ, Swoger J, Pampaloni F, Greger K, Marcello M, Stelzer EHK (2007) High-resolution three-dimensional imaging of large specimens with light sheet–based microscopy, *Nat Methods*, vol. 4, no. 4, pp. 311–313, Apr. https://doi.org/10.1038/nmeth1017
6. Santi PA (2011) Light Sheet Fluorescence Microscopy: A Review, *J Histochem Cytochem.*, vol. 59, no. 2, pp. 129–138, Feb. https://doi.org/10.1369/0022155410394857
7. Ricci P et al (2022) Jan., Removing striping artifacts in light-sheet fluorescence microscopy: a review, *Progress in Biophysics and Molecular Biology*, vol. 168, pp. 52–65, https://doi.org/10.1016/j.pbiomolbio.2021.07.003
8. Dong D et al (Oct. 2014) Vertically scanned laser sheet microscopy. J Biomed Opt 19(10):1. https://doi.org/10.1117/1.JBO.19.10.106001
9. Huisken J, Stainier DYR (Sep. 2007) Even fluorescence excitation by multidirectional selective plane illumination microscopy (mSPIM). Opt Lett 32:2608. https://doi.org/10.1364/OL.32.002608
10. Chang Y, Yan L, Wu T, Zhong S (Dec. 2016) Remote sensing image stripe noise removal: from image decomposition perspective. IEEE Trans Geosci Remote Sens 54(12):7018–7031. https://doi.org/10.1109/TGRS.2016.2594080
11. Münch B, Trtik P, Marone F, Stampanoni M (May 2009) Stripe and ring artifact removal with combined wavelet—fourier filtering. Opt Express 17(10):8567. https://doi.org/10.1364/OE.17.008567
12. Qu L et al (Jan. 2022) Cross-modal coherent registration of whole mouse brains. Nat Methods 19(1):111–118. https://doi.org/10.1038/s41592-021-01334-w
13. Tendero Y, Gilles J, Landeau S, Morel JM (2010) Efficient single image non-uniformity correction algorithm, presented at the Security + Defence, D. A. Huckridge and R. R. Ebert, Eds., Toulouse, France, Oct. p. 78340E. https://doi.org/10.1117/12.864804
14. Wang G et al (2019) Jul., DeepIGeoS: A Deep Interactive Geodesic Framework for Medical Image Segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1559–1572, https://doi.org/10.1109/TPAMI.2018.2840695
15. Alam MS, Wang D, Liao Q, Sowmya A (2023) A Multi-Scale Context Aware Attention Model for Medical Image Segmentation, *IEEE J. Biomed. Health Inform.*, vol. 27, no. 8, pp. 3731–3739, Aug. https://doi.org/10.1109/JBHI.2022.3227540
16. Hosny KM, Khalid AM, Hamza HM, Mirjalili S (Nov. 2022) Multilevel segmentation of 2D and volumetric medical images using hybrid coronavirus optimization algorithm. Comput Biol Med 150:106003. https://doi.org/10.1016/j.compbiomed.2022.106003
17. Guo Z, Zhao L, Yuan J, Yu H (2022) MSANet: Multiscale Aggregation Network Integrating Spatial and Channel Information for Lung Nodule Detection, *IEEE J. Biomed. Health Inform.*, vol. 26, no. 6, pp. 2547–2558, Jun. https://doi.org/10.1109/JBHI.2021.3131671
18. Chen X, Yang Q, Wu J, Li H, Tan KC (2023) A hybrid neural coding Approach for Pattern Recognition with spiking neural networks. IEEE Trans Pattern Anal Mach Intell 1–15. https://doi.org/10.1109/TPAMI.2023.3339211
19. Park CF et al (2023) Automated neuron tracking inside moving and deforming C. Elegans using deep learning and targeted augmentation. Nat Methods Dec. https://doi.org/10.1038/s41592-023-02096-3
20. Zavala-Mondragon LA, De With PHN, Sommen FVD (2021) Image noise reduction based on a fixed Wavelet Frame and CNNs Applied to CT. IEEE Trans Image Process 30:9386–9401. https://doi.org/10.1109/TIP.2021.3125489
21. Kuang X, Sui X, Chen Q, Gu G (Aug. 2017) Single infrared image stripe noise removal using deep Convolutional Networks. IEEE Photonics J 9(4):1–13. https://doi.org/10.1109/JPHOT.2017.2717948
22. Zhang K, Zuo W, Chen Y, Meng D, Zhang L (2017) Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising, *IEEE Trans. on Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. https://doi.org/10.1109/TIP.2017.2662206
23. Guan J, Lai R, Xiong A (2019) Wavelet Deep Neural Network for stripe noise removal. IEEE Access 7:44544–44554. https://doi.org/10.1109/ACCESS.2019.2908720
24. Pande-Chhetri R, Abd-Elrahman A (Sep. 2011) De-striping hyperspectral imagery using wavelet transform and adaptive frequency domain filtering. ISPRS J Photogrammetry Remote Sens 66(5):620–636. https://doi.org/10.1016/j.isprsjprs.2011.04.003
25. Hinton GE, Salakhutdinov RR (2006) Reducing the Dimensionality of Data with Neural Networks, *Science*, vol. 313, no. 5786, pp. 504–507, Jul. https://doi.org/10.1126/science.1127647

26. Ronneberger O, Fischer P, Brox T U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv, May 18, 2015. Accessed: Mar. 16, 2023. [Online]. Available: http://arxiv.org/abs/1505.04597
27. Wei Z et al (Mar. 2022) Elimination of stripe artifacts in light sheet fluorescence microscopy using an attention-based residual neural network. Biomed Opt Express 13(3):1292. https://doi.org/10.1364/BOE.448838
28. Yi Y et al (Jan. 2024) Mapping of individual sensory nerve axons from digits to spinal cord with the transparent embedding solvent system. Cell Res 34(2):124–139. https://doi.org/10.1038/s41422-023-00867-3
29. Chen Y, Huang T-Z, Deng L-J, Zhao X-L, Wang M (2017) Group sparsity based regularization model for remote sensing image stripe noise removal, *Neurocomputing*, vol. 267, pp. 95–106, Dec. https://doi.org/10.1016/j.neucom.2017.05.018
30. Çiçek Ö, Abdulkadir A, Lienkamp SS, Brox T, Ronneberger O 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. arXiv, Jun. 21, 2016. Accessed: Mar. 16, 2023. [Online]. Available: http://arxiv.org/abs/1606.06650
31. Falk T (2019) U-Net: deep learning for cell counting, detection, and morphometry. Nat Methods, 16
32. Simonyan K, Zisserman A (2015) VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION
33. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z Rethinking the Inception Architecture for Computer Vision. arXiv, Dec. 11, 2015. Accessed: Apr. 21, 2023. [Online]. Available: http://arxiv.org/abs/1512.00567
34. Qi X et al (2023) High-throughput confocal airy beam oblique light-sheet tomography of brain-wide imaging at single-cell resolution. Neurosci Preprint Jun. https://doi.org/10.1101/2023.06.04.543586
35. Burger HC, Schuler CJ, Harmeling S, Image denoising: Can plain neural networks compete with BM3D? in (2012) *IEEE Conference on Computer Vision and Pattern* Recognition, Providence, RI: IEEE, Jun. 2012, pp. 2392–2399. https://doi.org/10.1109/CVPR.2012.6247952
36. Andersen CF, Farup I, Hardeberg JY (2023) Additivity Constrained Linearisation of Camera Calibration Data. IEEE Trans Image Process 32:3774–3789. https://doi.org/10.1109/TIP.2023.3287735
37. Kask P, Palo K, Hinnah C, Pommerencke T (2016) Flat field correction for high-throughput imaging of fluorescent samples, *Journal of Microscopy*, vol. 263, no. 3, pp. 328–340, Sep. https://doi.org/10.1111/jmi.12404
38. Cao H, Gu X, Zhang M, Zhang H, Chen X (2022) Vignetting correction based on a two-Dimensional Gaussian Filter with harmony for area array sensors. IEEE Trans Comput Imaging 8:576–584. https://doi.org/10.1109/TCI.2022.3188413
39. Saleh HM, Saad NH, Isa NAM (2019) Overlapping chromosome segmentation using U-Net: Convolutional Networks with Test Time Augmentation. Procedia Comput Sci 159:524–533. https://doi.org/10.1016/j.procs.2019.09.207
40. Zhao B-W, Su X-R, Hu P-W, Huang Y-A, You Z-H, Hu L (Aug. 2023) iGRLDTI: an improved graph representation learning method for predicting drug–target interactions over heterogeneous biological information network. Bioinformatics 39(8):btad451. https://doi.org/10.1093/bioinformatics/btad451
41. Huang G, Liu Z, van der Maaten L, Weinberger KQ (2018) Densely Connected Convolutional Networks. arXiv, Jan. 28, Accessed: May 05, 2023. [Online]. Available: http://arxiv.org/abs/1608.06993
42. Twogood RE, Sommer FG (1982) Digital Image Processing. IEEE Trans Nucl Sci 29(3):1075–1086. https://doi.org/10.1109/TNS.1982.4336327
43. Sheikh HR, Sabir MF, Bovik AC (2006) A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms, *IEEE Trans. on Image Process.*, vol. 15, no. 11, pp. 3440–3451, Nov. https://doi.org/10.1109/TIP.2006.881959
44. Preedanan W, Kondo T, Bunnun P, Kumazawa I (2018) A comparative study of image quality assessment, in *2018 International Workshop on Advanced Image Technology (IWAIT)*, Chiang Mai: IEEE, Jan. pp. 1–4. https://doi.org/10.1109/IWAIT.2018.8369657
45. Zhou G, Zhao Q, Zhang Y, Adali T, Xie S, Cichocki A, Linked Component Analysis From Matrices to High-Order Tensors: Applications to Biomedical, Data (2016) *Proc. IEEE*, vol. 104, no. 2, pp. 310–331, Feb. https://doi.org/10.1109/JPROC.2015.2474704
46. Lai R, Mo Y, Liu Z, Guan J (2019) Local and Nonlocal Steering Kernel Weighted Total Variation Model for Image Denoising, *Symmetry*, vol. 11, no. 3, p. 329, Mar. https://doi.org/10.3390/sym11030329

Publisher's note