J. ADV. SIMULAT. SCI. ENG.
Vol. 8, No. 2, 173-193.

Image processing method for automatic measurement of number of DNA breaks

Seiki Saito[1]*, Hiroaki Nakamura[2, 3], Takahiro Kenmotsu[4], Yasuhisa Oya[5], Yuji Hatano[6], Yuichi Tamura[7], Susumu Fujiwara[8], Hiroaki Ohtani[2, 9]

[1]Department of Informatics and Electronics, Faculty of Engineering, Yamagata University
[2]Fundamenal Physics Simulation Research Division, Department of Helical Plasma Research, National Institute for Fusion Science
[3]Department of Electrical Engineering, Graduate School of Engineering, Nagoya University
[4]Faculty of Life and Medical Sciences, Doshisha University
[5]Center for Radioscience Education and Research, Faculty of Science, Shizuoka University
[6]Hydrogen Isotope Research Center, Organization for Promotion of Research, University of Toyama
[7]Department of Intelligence and Informatics, Faculty of Intelligence and Informatics, Konan University
[8]Faculty of Materials Science and Engineering, Kyoto Institute of Technology
[9]Department of Fusion Science, The Graduate University for Advanced Studies, SOKENDAI

*saitos@yz.yamagata-u.ac.jp

Received: January 31, 2021; Accepted: July 1, 2021; Published: September 23, 2021

Abstract. The number of double-strand breaks can be evaluated from the change of average DNA length. The average DNA length is measured by the single-molecule observation method using fluorescence microscope. The measurement of DNA length in the microscope images is done manually by experienced operators and it is time consuming in many experiments. An image processing method using OpenCV library to measure length of DNA in fluorescence microscope images is developed in this paper. An automation of measurement using deep learning is also proposed.

Keywords: Image processing, DNA, Double-strand breaks, Fluorescence microscope, Pix2pix

1. Introduction

Damages of DNA induce serious problems for living things because DNA contains genetic instructions for the development, functioning, growth and reproduction of all known living organisms. Many research reports that DNA is damaged by chemical and physical reactions in environmental condition. For example, ultrasonic which is important for application in practical medicine causes double-strand breaks in genome-sized DNA [1]. Photo induced

damages in DNA are also investigated [2]. Quantitative evaluation of DNA damages induced by radiations such as γ ray irradiation [3-6] are key issue for radiation protection. The effects of tritium, which is planning to be used for fusion power generation, on DNA are also attracting the interest of researchers [7-10].

Single-molecule observation method is widely used for the investigation of double-strand breaks in DNA. In the observation, images of DNA molecules can be captured by fluorescence microscopes by using fluorescent dye such as YOYO-1 as a photosensitizer. By measuring changes in DNA length, it is possible to investigate how much the factors such as ultrasound, visible light, and radiation, cause double-strand breaks. More quantitatively, the number of double-strand breaks N can be calculated by the relation $N = (\langle L_0 \rangle - \langle L \rangle)/\langle L \rangle$, where $\langle L_0 \rangle$ and $\langle L \rangle$ are the average DNA length before and after the factors that cause double-strand breaks occur, respectively. $\langle L_0 \rangle$ or $\langle L \rangle$ can be obtained by measuring length and number of DNA segments in fluorescence microscope images.

Usually, the measurement is done manually by experienced operators and it is time consuming. In this paper, therefore, an image processing method is developed to measure the length of DNA segments in fluorescence microscope images. The automation of DNA length measurement can be achieved by the following two steps: 1) extracting only DNA segments from fluorescence microscopy images, and 2) measuring the length of the DNA segments by OpenCV. Step 2) is explained in section 2. Step 1) can be realized by deep learning, which is explained in section 3

2. Measurement method by image processing of OpenCV

An example of the original image captured by a fluorescence microscope is shown in Fig. 1 (a). Multiple DNA segments are shown in white in the image. To measure the length of a DNA segments, we convert the segments into a thin line one pixel wide (skeletonization). Using OpenCV library, the original images are skeletonized by the following steps before measuring the length and number of DNA segments in the image by the method explained later. In the steps, OpenCV functions: “equalizeHist”, “threshold”, “adaptiveThreshold”,

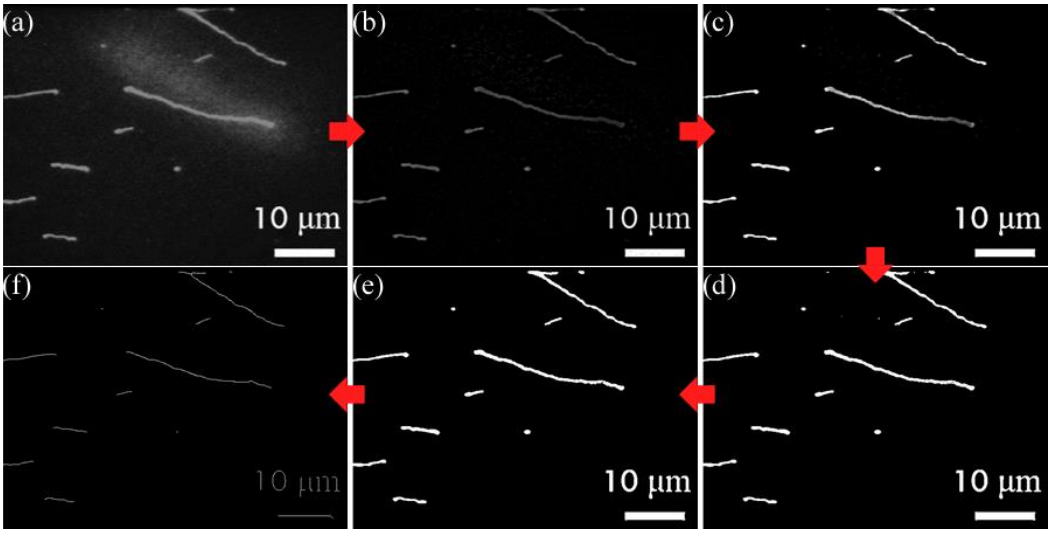

Figure 1: Example of a result of image processing by conventional method by OpenCV.

"opening", "findContours", and "arcLength" are used. "equalizeHist" is used to equalize the histogram to adjust the contrast of the original fluorescence microscope images. "threshold" is used for binarization of the images with constant threshold. "adaptiveThreshold" is used for binarization using threshold values that varies according to the local shading of the image. "opening" is used for the noise reduction by applying erosion and dilation. "findContours" is used to detect the outline of the DNA segments. "arcLength" is used to measure the length of the outline detected by "findContours".

step-A1 The histogram of the image is equalized by the function "equalizeHist" in OpenCV.
step-A2 The change in local shading is reduced by the unevenness coefficient as shown in Fig. 1 (b).
step-A3 The histogram is manually adjusted by changing tone curve by the function "LUT" as shown in Fig. 1 (c).
step-A4 The image is binarized by the function "threshold" or "adaptiveThreshold" as show in Fig. 1 (d).
step-A5 Noise reduction is performed by the function "opening" in OpenCV as shown in Fig. 1 (e).
step-A6 The white pixels are skeletonized by Zhang-Suen method [11] as shown in Fig. 1 (f).

The contrast and brightness of images depend on the experimental conditions. To support input images under various conditions, the procedure of image processing is manually controlled by the variables shown in Table 1 in addition to the tone curve in step-A3.

After the skeletonization (step-A6), the length and number of DNA segments in the image are measured. Figure 2 (a)-(c) show the process of measurement of the length of DNA segments. As shown in Fig 2 (b), the skeletonized image is firstly dilated by the function "dilate" in OpenCV with kernel size of 2×2. Then, as shown in red color in Fig. 2 (c), the contour of the area of white pixels is detected by the function "findContours" in OpenCV. Finally,

Table 1: controlled parameters used in the proposed image processing method.

Variables	Type	Range	Explanation
s_{uneven}	bool	True/False	Bool value of whether to carry out step-A2. If the value is false, step-A2 is skipped.
k_{uneven}	integer	0-50	Integer value of kernel size for calculating average brightness in calculation of the unevenness coefficient. Using k_{uneven}, the value of (i,j) pixel x_{ij} is updated to x'_{ij} with the unevenness coefficient r_{ij} of (i,j) pixel by the following equations: $x'_{ij} = 255 \times (1 - r_{ij}),\quad r_{ij} = (255 - x_{ij})/R_{ij},$ $R_{ij} = \frac{1}{(2k_{\mathrm{uneven}}+1)^2} \sum_{k,l=-k_{\mathrm{uneven}}}^{k_{\mathrm{uneven}}} (255 - x_{i+k,j+l}).$
s_{local}	bool	True/False	Bool value for adaptive thresholding in step-A4. The function "adaptiveThreshold" is applied instead of "threshold" if the value is true.
$b_{\mathrm{threshold}}$	integer	0-255	Integer value for the threshold of the binarization in step-A4.
k_{opening}	integer	0-20	Integer value of kernel size for the function "opening".

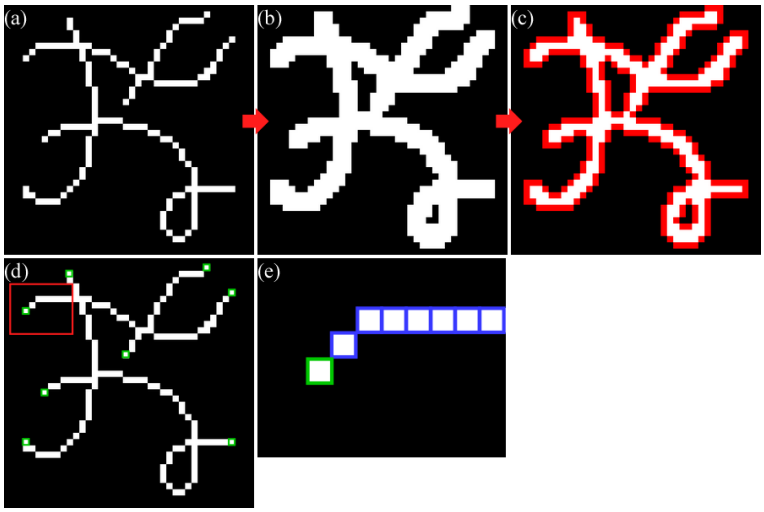

Figure 2: Method of measurement of total length and number of DNAs.

length of contour is calculated by the function “arcLength” in OpenCV. The length of DNA segments is obtained by dividing the length of the contour by 2. Figure 2 (d) and (e) explain the method of counting the number of DNA segments. The number can be counted by dividing the number of endpoints of the skeletonized segments by 2. The end points are emphasized in green in Fig. 2 (d). The pixel at an endpoint is detected by counting the number of white pixels in 8 adjacent pixels on the skeletonized DNA segments. Figure 2 (e) shows the enlarged image of the rectangle area shown by red frame in Fig. 2 (d). As shown in Fig. 2 (e), the number is unity for the pixels of endpoints as in the case of the pixel emphasized in green, although the number is two for the middle pixels of the segment emphasized in blue. We note that this measurement method provides the length of all intersecting segments as one segment. However, average length of DNA segments which is used for the calculation of number of double-strand breaks in DNA is obtained by dividing the total length of all DNA segments by the number of segments.

Figure 3 shows the image after the measurement method is applied to the image shown in Fig. 1 (a). In the image, the counters of the DNA segments are shown in yellow. The detected endpoints are shown by purple cross marks. Table 2 shows the comparison between DNA length measured by proposed image processing method and which obtained by manual measurement by an operator. In manual measurement, the operator clicks many times on the image to approximate the DNA segments with polygonal lines to obtain the DNA length. The segment numbers in Table 2 correspond to the numbers shown in Fig. 3. The result shows that the measurement by proposed method provides almost the same length as the result of manual measurement although the value is slightly longer in most of the cases. We note that the length of #2 and #8 is not measured in manual measurement because the length is too short. The difference is relatively large in the case of #1 because the part of the DNA segment protrudes from the image.

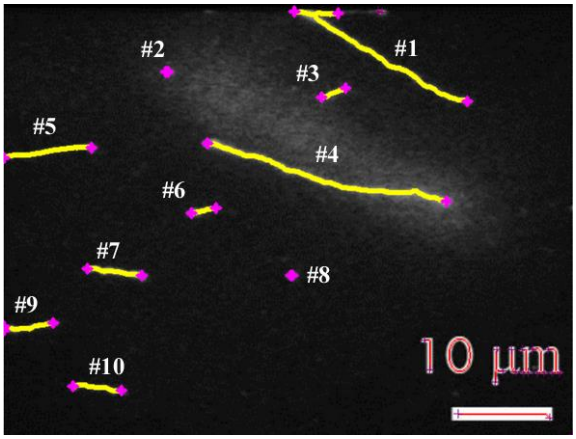

Figure 3: Example of the applying the measurement method of total length and number of DNAs to an actual fluorescence microscope image.

Table 2: Comparison between DNA length measured by proposed image processing method and which obtained by manual measurement.

Segment No.	Manual measurement [μm]	Measurement by image process. [μm]	Difference [μm]
#1	19.87	23.70	3.82
#2	-	0.42	-
#3	3.32	2.96	-0.35
#4	25.63	26.32	0.69
#5	8.76	9.40	0.64
#6	2.75	2.79	0.04
#7	5.72	5.96	0.24
#8	-	0.33	-
#9	5.54	5.68	0.14
#10	6.83	5.31	-1.52

3. Measurement automation by deep learning

3.1. Overview of automation method

When measure the length and number of DNA segments by the proposed image processing method explained in section 2, the operator must manually determine the tone curve in step-A3 and the control parameters shown in Table 1 to perform measurements on variety of images captured under different experimental conditions. In addition to that, operator must select DNA segments to measure because the images contain many images of impurities and curled DNA segments which should not include for the calculation of average DNA length as shown in Fig. 4.

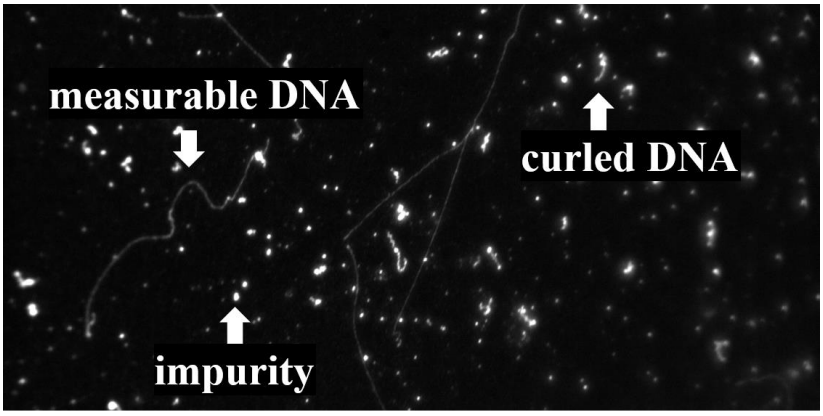

Figure 4: Example of fluorescence microscope images which contains many impurities and curled DNA segments.

To achieve automatic measurement, deep learning models are developed for determining all controlled parameters and selecting measurable DNA segments. Figure 5 shows the procedure of automatic measurement using deep learning models. The procedure uses five deep learning models: $\boldsymbol{M}_{\mathrm{param}}$, $\boldsymbol{M}_{\mathrm{conn}}$, $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$, $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$, and $\boldsymbol{M}_{\mathrm{noise}}$. The functions of these models are explained in Table 3. In step-B1, step-A1 to step-A6 are performed with predicted parameters by $\boldsymbol{M}_{\mathrm{param}}$. In step-B2, separate DNA segments, which should be considered as one DNA segment, are connected by $\boldsymbol{M}_{\mathrm{conn}}$ to improve the accuracy of DNA extraction in the next step. In step-B3a and step-B3b, the DNA extraction is performed by $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$ and $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$. To improve the accuracy, the two result images generated by two models $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$ and $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$ are superimposed in step-B4. To connect the DNA segments which are separated during the image processing, $\boldsymbol{M}_{\mathrm{conn}}$ is applied again after skeletonization and dilation in step-B5 to step-B7. In step-B8, the skeletonization is performed again. Then, the noise reduction $\boldsymbol{M}_{\mathrm{noise}}$ is applied. The training processes of the five deep learning models are explained in section 3.2 and 3.5 to 3.7.

3.2. Parameter prediction for image processing using OpenCV by CNN model: $\boldsymbol{M}_{\mathrm{param}}$

Figure 6 shows the architecture of $\boldsymbol{M}_{\mathrm{param}}$. Convolutional neural network (CNN) [12] is adopted for the model. $\boldsymbol{M}_{\mathrm{param}}$ received an equalized fluorescence microscope image (the

Table 3: Explanation of five deep learning models used for the developed procedure shown in Fig. 5.

Model	Functions
$\boldsymbol{M}_{\mathrm{param}}$	Prediction of the control parameters and the tone curve used in step-A1 to step-A6.
$\boldsymbol{M}_{\mathrm{conn}}$	Connection of DNA segments that are separated in binarization of step-A4 due to the problem of local shading.
$\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}, \boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$	Extraction of measurable DNA segments by removing impurities and curled DNA segments.
$\boldsymbol{M}_{\mathrm{noise}}$	Noise removal leaving the DNA segments.

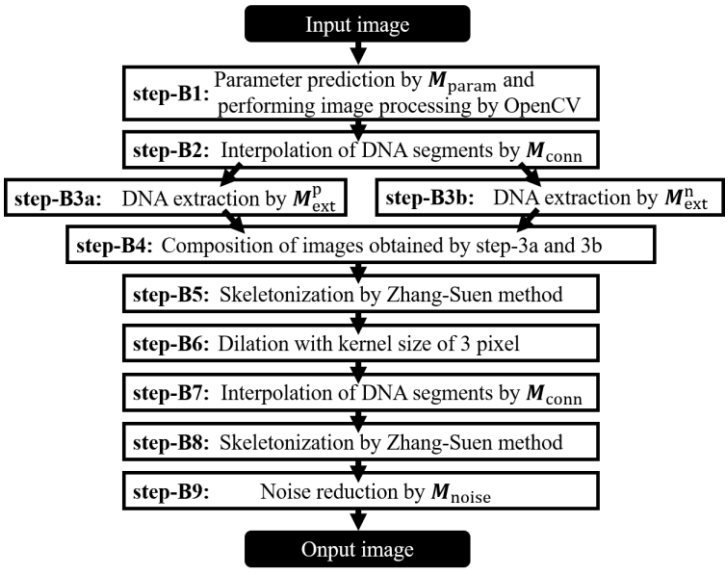

Figure 5: Procedure of automatic image processing of DNA extraction by five deep learning models: $\boldsymbol{M}_{\mathrm{param}}$, $\boldsymbol{M}_{\mathrm{conn}}$, $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$, $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$, and $\boldsymbol{M}_{\mathrm{noise}}$.

image after step-A1) and output a 261-dimensional vector. Input images are resized to 512×512 pixels. The 5 of 261 dimensions correspond to the 5 parameters shown in Table 1, and remaining 256 dimensions correspond to the tone-curve in step-A3. All values of the output vector corresponding to integer parameters are scaled to 0 to 1.0 by dividing maximum of the range of the parameter. The values corresponding to bool values are set to 1.0 for Ture and 0.0 for False.

178 equalized fluorescence microscope images are prepared for the training and validation. 80 of 178 images are captured at University of Toyama using an inverted microscope (IX73, Olympus Co., Japan) equipped with a sCMOS camera (Zyla 4.3, Andor Technology Co., UK). 73 of 178 images are captured at Shizuoka university using an inverted microscope (IX73, Olympus Co., Japan) equipped with a sCMOS camera (Zyla 5.5, Andor Technology Co., UK). 25 of 178 images are captured at Doshisha University using an inverted microscope (Axiovert 135 TV, Carl Zeiss, Germany) equipped with an oil-immersed 100V objective lens. The 73 images captured at Shizuoka university are prepared by data augmentation from original 4 images. For the training and validation, corresponding output vectors for the 178 images are prepared manually by an operator. Figure 7 shows four examples of the input images used for the training. The left images are the equalized fluorescence microscope images, and the right images are the results of the image processing from step-A2 to step-A6 with manually adjusted parameters. The manually adjusted parameters and tone curve for the four examples are shown in Fig. 8 and Table 4.

80% of 178 images are used for the training and remaining 20% is used for the validation. Mean absolute error (MAE) is used for the loss function. Stochastic gradient descent (SGD) is used for the optimization. Figure 9 shows the evolution of training and validation loss during the training.

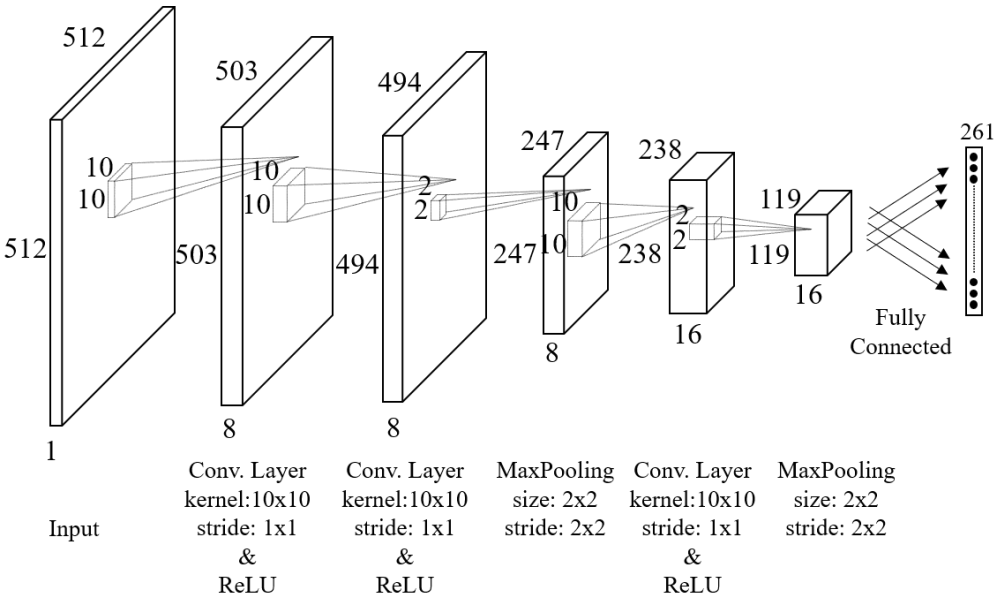

Figure 6: Architecture of CNN adopted for $\boldsymbol{M}_{\mathrm{param}}$.

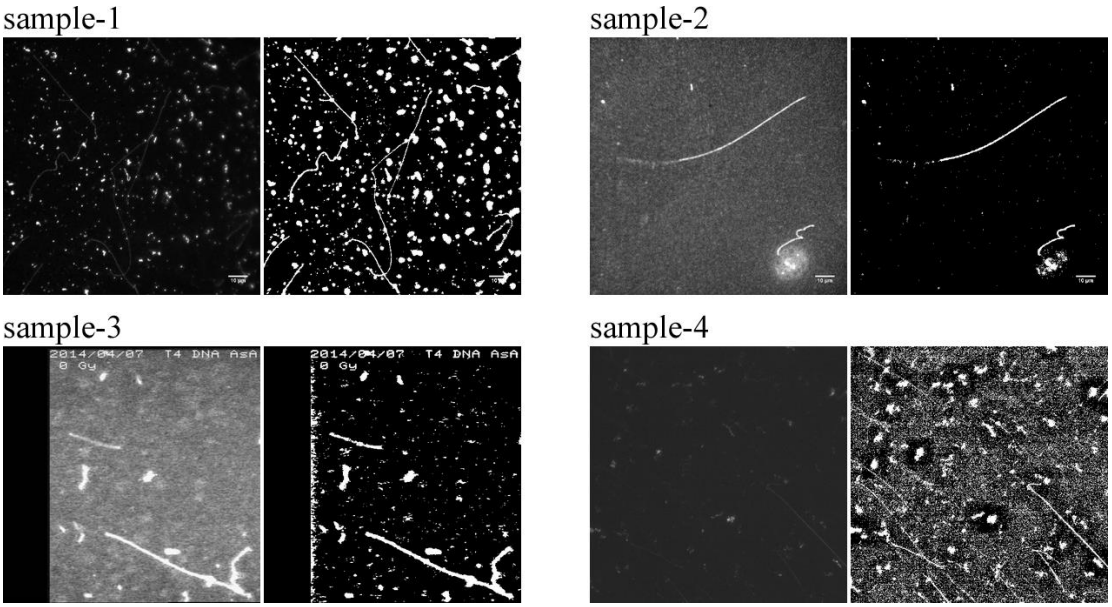

Figure 7: Four examples of pair of equalized input images (left), and the results (right) of the image processing from step-A2 to step-A6 with manually adjusted parameters for the training of CNN model $\boldsymbol{M}_{\mathrm{param}}$. Sample-1 and 2 are images captured at University of Toyama. Sample-3 is an image captured at Doshisha University. Sample-4 is an image captured at Shizuoka university.

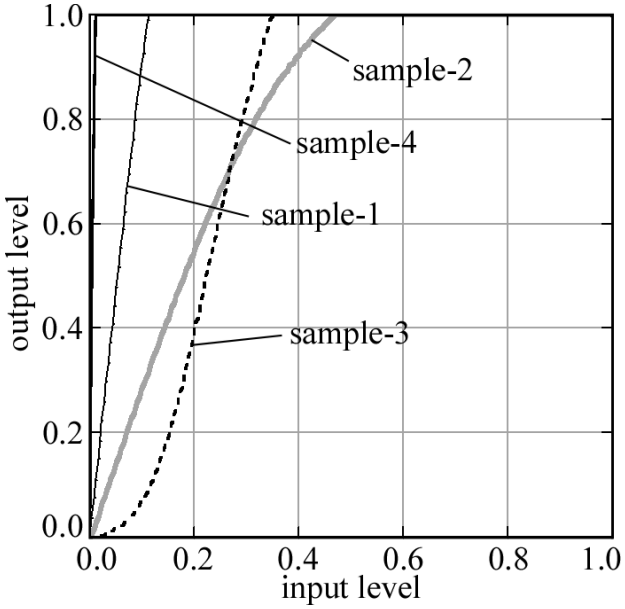

Figure 8: Manually adjusted tone-curves for the four examples shown in Fig. 7.

Table 4: Manually adjusted parameters for the four examples shown in Fig. 7.

Variable	s_{uneven}	k_{uneven}	s_{local}	$b_{\text{threshold}}$	k_{opening}
Type	bool	integer	bool	integer	integer
Range	True/False	0-50	True/False	0-255	0-20
sample-1	True	50	False	201	0
sample-2	True	50	False	66	0
sample-3	True	38	False	110	0
sample-4	True	50	False	79	0

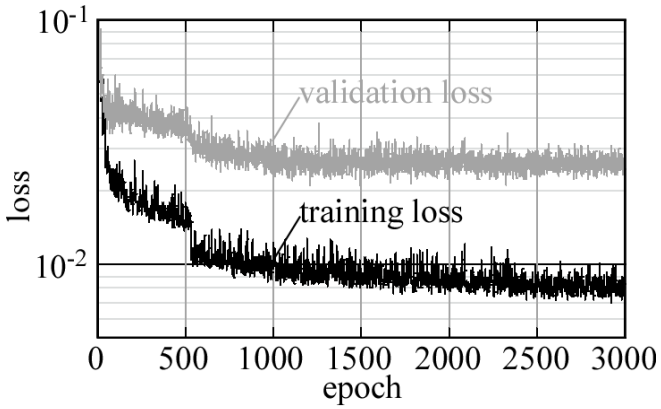

Figure 9: Evolution of training and validation loss during the training of CNN model $\boldsymbol{M}_{\text{param}}$.

3.3. Introduction to pix2pix

As explained in section 3.1, image conversion is performed by $\boldsymbol{M}_{\mathrm{conn}}$, $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$, $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$, and $\boldsymbol{M}_{\mathrm{noise}}$ after the fluorescence microscope images are binarized by step-A1 to step-A5 with predicted parameter provided by $\boldsymbol{M}_{\mathrm{param}}$. $\boldsymbol{M}_{\mathrm{conn}}$ interpolates the disconnected DNA segments. $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$ and $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$ extract DNA segments. $\boldsymbol{M}_{\mathrm{noise}}$ reduces noise. Pix2pix [13], which is a deep learning model which generates a converted image from an input image, is adopted for the models of $\boldsymbol{M}_{\mathrm{conn}}$, $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$, $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$, and $\boldsymbol{M}_{\mathrm{noise}}$.

Pix2pix is one of generative adversarial networks (GAN). The model consists of a generator $\boldsymbol{G}$ and a discriminator $\boldsymbol{D}$. In the case of pix2pix, $\boldsymbol{G}$ and $\boldsymbol{D}$ are trained by many of image pairs of $(\boldsymbol{x}, \boldsymbol{y}^{\mathrm{c}})$, where $\boldsymbol{x}$ is an original input image and $\boldsymbol{y}^{\mathrm{c}}$ is a corresponding converted image. $\boldsymbol{G}$ outputs an image $\boldsymbol{y} = \boldsymbol{G}(\boldsymbol{x})$ from an input image $\boldsymbol{x}$. $\boldsymbol{G}$ is trained to predicts $\boldsymbol{y}^{\mathrm{c}}$ from $\boldsymbol{x}$. The size of $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{y}^{\mathrm{c}}$ is set to 512×512 pixels in our model. $\boldsymbol{D}$ outputs 2-dimensional normalized vector $\boldsymbol{e} = \boldsymbol{D}(\boldsymbol{x}^{\mathrm{D}})$ from an input image $\boldsymbol{x}_{\mathrm{D}}$. $\boldsymbol{D}(\boldsymbol{x}^{\mathrm{D}})$ is trained to be a function $\boldsymbol{f}(\boldsymbol{x}^{\mathrm{D}})$ defined as follows:

$$\boldsymbol{f}(\boldsymbol{x}_{\mathrm{D}}) \equiv \begin{cases} (1,0) & \text{when } \boldsymbol{x}^{\mathrm{D}} \text{ is generator's output } \boldsymbol{y} \\ (0,1) & \text{when } \boldsymbol{x}^{\mathrm{D}} \text{ is training image } \boldsymbol{y}^{\mathrm{c}} \end{cases},$$

by minimizing the cross entropy $\mathcal{L}_{\log}\left(\boldsymbol{D}(\boldsymbol{x}^{\mathrm{D}}), \boldsymbol{f}(\boldsymbol{x}^{\mathrm{D}})\right)$. In the training process of $\boldsymbol{D}$, $\boldsymbol{x}^{\mathrm{D}}$ is set to a training image $\boldsymbol{y}^{\mathrm{c}}$ in half probability and set to $\boldsymbol{G}$'s output $\boldsymbol{y} = \boldsymbol{G}(\boldsymbol{x})$, otherwise. $\boldsymbol{G}$ is trained to maximizing the cross entropy $\mathcal{L}_{\log}(\boldsymbol{e}^{\mathrm{G}}, \boldsymbol{f}^{\mathrm{G}})$ and minimizing the L1 loss $\mathcal{L}_{\mathrm{L1}}(\boldsymbol{y}^{\mathrm{c}}, \boldsymbol{y})$ where $\boldsymbol{e}^{\mathrm{G}} \equiv \boldsymbol{D}(\boldsymbol{G}(\boldsymbol{x}))$ and $\boldsymbol{f}^{\mathrm{G}} \equiv \boldsymbol{f}(\boldsymbol{G}(\boldsymbol{x})) = (1,0)$. The definition of $\mathcal{L}_{\log}$ and $\mathcal{L}_{\mathrm{L1}}$ are as follows:

$$\mathcal{L}_{\log}(\boldsymbol{e}, \boldsymbol{f}) \equiv \sum_{k} f_k \log e_k, \quad \mathcal{L}_{\mathrm{L1}}(\boldsymbol{y}^{\mathrm{c}}, \boldsymbol{y}) \equiv \sum_{i,j} \left| y_{ij}^{\mathrm{c}} - y_{ij} \right|,$$

where, e_k and f_k are k-th element of vector $\boldsymbol{e}$ and $\boldsymbol{f}$, respectively. y_{ij}^{c} and y_{ij} are the i, j element of matrix $\boldsymbol{y}^{\mathrm{c}}$ and $\boldsymbol{y}$, respectively. Again, the training adversarially proceeds under minimizing $\mathcal{L}_{\log}(\boldsymbol{e}, \boldsymbol{f})$ by $\boldsymbol{D}$, and maximizing $\mathcal{L}_{\log}(\boldsymbol{e}^{\mathrm{G}}, \boldsymbol{f}^{\mathrm{G}})$ by $\boldsymbol{G}$. Instead of maximizing $\mathcal{L}_{\log}(\boldsymbol{e}^{\mathrm{G}}, \boldsymbol{f}^{\mathrm{G}})$, $\mathcal{L}_{\log}(\boldsymbol{e}^{\mathrm{G}}, \bar{\boldsymbol{f}}^{\mathrm{G}})$ is minimized to simplify the calculation, where $\bar{\boldsymbol{f}}^{\mathrm{G}} = (0,1)$. Therefore, $\boldsymbol{G}$ is trained by minimizing the following loss function $\mathcal{L}_{\mathrm{G}}$.

$$\mathcal{L}_{\mathrm{G}}(\boldsymbol{e}^{\mathrm{G}}(\boldsymbol{x}), \boldsymbol{y}(\boldsymbol{x}), \boldsymbol{y}^{\mathrm{c}}) \equiv \mathcal{L}_{\log}(\boldsymbol{e}^{\mathrm{G}}, \bar{\boldsymbol{f}}^{\mathrm{G}}) + h\mathcal{L}_{\mathrm{L1}}(\boldsymbol{y}^{\mathrm{c}}, \boldsymbol{y})$$

Here, $h = 10$ is a hyper parameter. For the optimization, Adam [14] is used with batch size of four.

Figure 10 (a) and (b) show the architecture of $\boldsymbol{G}$ and $\boldsymbol{D}$. $\boldsymbol{G}$ consists of a down-sampling part and an up-sampling part. In the down-sampling part, input image is downscaled by repeated convolutional layer L_i^{dn} shown in Fig. 10 (a). In the up-sampling part, the downscaled data is upscaled by repeated convolutional layer L_i^{up} shown in Fig. 10 (a). $\boldsymbol{D}$ consists of repeated convolutional layer L_i^{D} shown in Fig. 10 (b) for down-sampling. The size of parameters $W_i^{\mathrm{dn}} \times H_i^{\mathrm{dn}} \times C_i^{\mathrm{dn}}$, $W_i^{\mathrm{up}} \times H_i^{\mathrm{up}} \times C_i^{\mathrm{up}}$, and $W_i^{\mathrm{D}} \times H_i^{\mathrm{D}} \times C_i^{\mathrm{D}}$ of i-th layer are shown in Table 5.

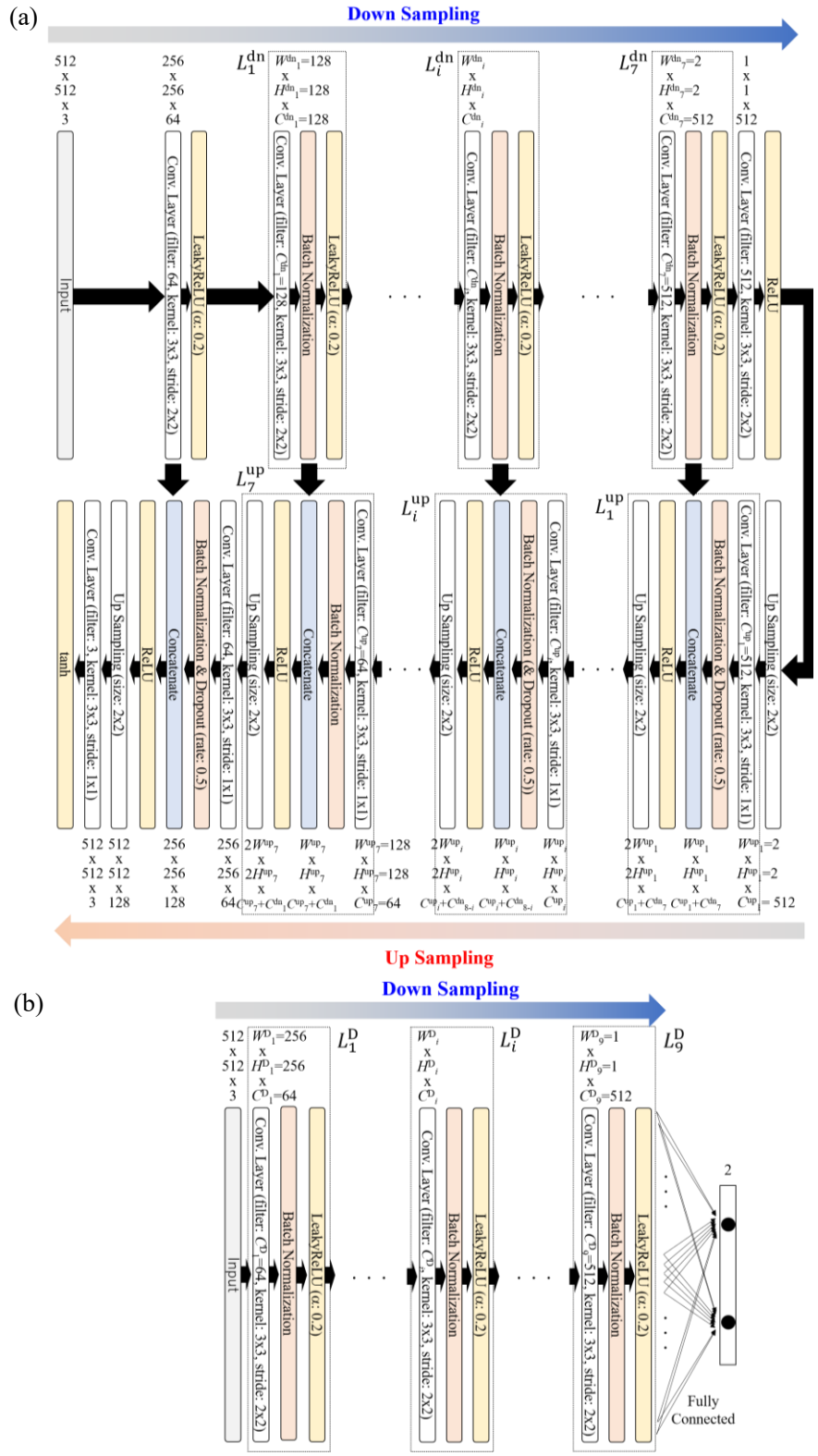

Figure 10: Architecture of pix2pix of (a) generator $\boldsymbol{G}$ and (b) discriminator $\boldsymbol{D}$ adopted for model of $\boldsymbol{M}_{\mathrm{conn}}$, $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$, $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$, and $\boldsymbol{M}_{\mathrm{noise}}$. The parameter α for the activation function: LeakyReLU is the slope when input value $x < 0$.

Table 5: Parameters of repeated convolutional layers L_i^{dn}, L_i^{up}, and L_i^{D} shown in Fig. 10.

i	1	2	3	4	5	6	7	8	9
W_i^{dn}	128	64	32	16	8	4	2		
H_i^{dn}	128	64	32	16	8	4	2		
C_i^{dn}	128	256	512	512	512	512	512		
W_i^{up}	2	4	8	16	32	64	128		
H_i^{up}	2	4	8	16	32	64	128		
C_i^{up}	512	512	512	512	256	128	64		
W_i^{D}	256	128	64	32	16	8	4	2	1
H_i^{D}	256	128	64	32	16	8	4	2	1
C_i^{D}	64	128	256	512	512	512	512	512	512

3.4. Method of imitation image generation for preparing training data

A software is developed to automatically generate enough sets of training data $(\boldsymbol{x}, \boldsymbol{y}^{\mathrm{c}})$. The software generates imitation images of fluorescence microscope images which binarized by step-A1 to step-A5. As shown in Fig. 11, in the software, the imitation images $\boldsymbol{x}$ and $\boldsymbol{y}^{\mathrm{c}}$ are generated by three steps: drawing DNA segments; drawing impurities including curled DNA; drawing noise. One DNA segment is drawn in white by connecting a series of points by third-order spline method. In addition to number of the points, distance and angle between adjacent points are controlled by random number to be closer to real DNA images. The thickness of DNA segments is also determined randomly. Number of DNA segments in an image are set randomly from 4 to 10. Impurities are drawn by pasting an image of impurities cut out from the actual fluorescence microscope images like a stamp. 200 stamps are prepared for the impurities. 22 examples of 200 stamps are shown in Fig. 12. These stamps are pasted at random positions after rotating or flipping randomly. The number of stamps in an image is randomly set from 20 to 100. Noise is drawn by randomly distributing white 1-pixel dots. The number of dots in an image are randomly set from 50 to 100000.

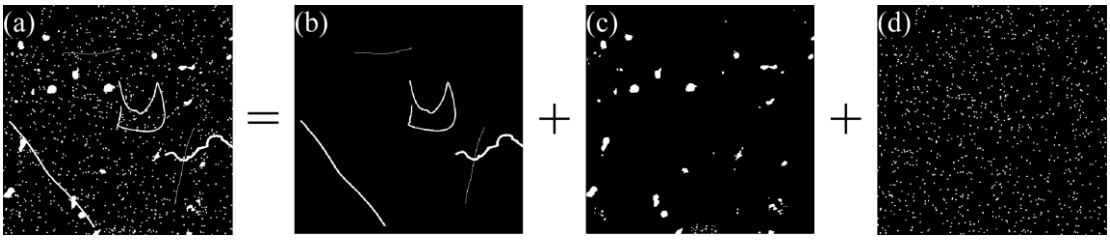

Figure 11: (a) Example of generated imitation image by developed generation software for training data $\boldsymbol{x}$ and $\boldsymbol{y}^{\mathrm{c}}$. The image (a) is generated by superimposing the images of (b) DNA segments, (c) impurities, and (c) noise. The noise in this figure is drawn in 2 pixels for display while actual images are drawn in 1 pixel.

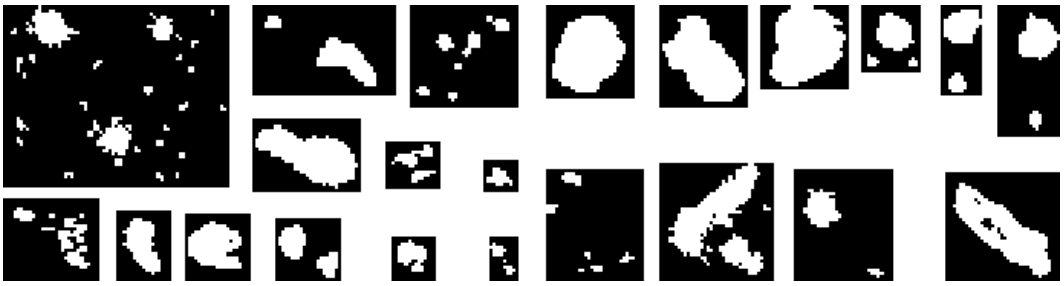

Figure 12: 22 examples of impurity images including curled DNA for generation of imitation images.

3.5. Training of $\boldsymbol{M}_{\mathbf{conn}}$ for interpolation of DNA segments

The model $\boldsymbol{M}_{\mathrm{conn}}$ is trained to connect DNA segments which are separated in binarization of step-A4 due to the problem of local shading. Therefore, images which are artificially cut DNA segments are generated for training images $\boldsymbol{x}$ corresponding to $\boldsymbol{y}^{\mathrm{c}}$. In the generation of training images $\boldsymbol{x}$, in the first, the DNA segments are drawn while changing the thickness of each part of that segments corresponding $\boldsymbol{y}^{\mathrm{c}}$ as shown in Fig. 13 (a). Then, blur effects are applied the segments with a thickness of 2 pixels or more as shown in Fig. 13 (b). Finally, the image is binarized as shown in Fig. 13 (c). Figure 14 shows four examples of training data set $(\boldsymbol{x}, \boldsymbol{y}^{\mathrm{c}})$ generated by the software.

The model $\boldsymbol{M}_{\mathrm{conn}}$ is trained to predict $\boldsymbol{y}^{\mathrm{c}}$ from $\boldsymbol{x}$. Figure 15 shows the history of training loss during the training. The values of three loss functions: cross entropy $\mathcal{L}_{\log}\left(\boldsymbol{D}(\boldsymbol{x}^{\mathrm{D}}), \boldsymbol{f}(\boldsymbol{x}^{\mathrm{D}})\right)$ for training of discriminator $\boldsymbol{D}$; cross entropy $\mathcal{L}_{\log}\left(\boldsymbol{D}(\boldsymbol{G}(\boldsymbol{x})), \boldsymbol{f}^{\mathrm{G}}\right)$ for training of generator $\boldsymbol{G}$; L1 loss $h\mathcal{L}_{\mathrm{L1}}(\boldsymbol{y}^{\mathrm{c}}, \boldsymbol{G}(\boldsymbol{x}))$ are shown separately. $\mathcal{L}_{\log}\left(\boldsymbol{D}(\boldsymbol{x}^{\mathrm{D}}), \boldsymbol{f}(\boldsymbol{x}^{\mathrm{D}})\right)$ and $\mathcal{L}_{\log}\left(\boldsymbol{D}(\boldsymbol{G}(\boldsymbol{x})), \boldsymbol{f}^{\mathrm{G}}\right)$ reach equilibrium. As a result, $\mathcal{L}_{\mathrm{L1}}(\boldsymbol{y}^{\mathrm{c}}, \boldsymbol{G}(\boldsymbol{x}))$ becomes smaller.

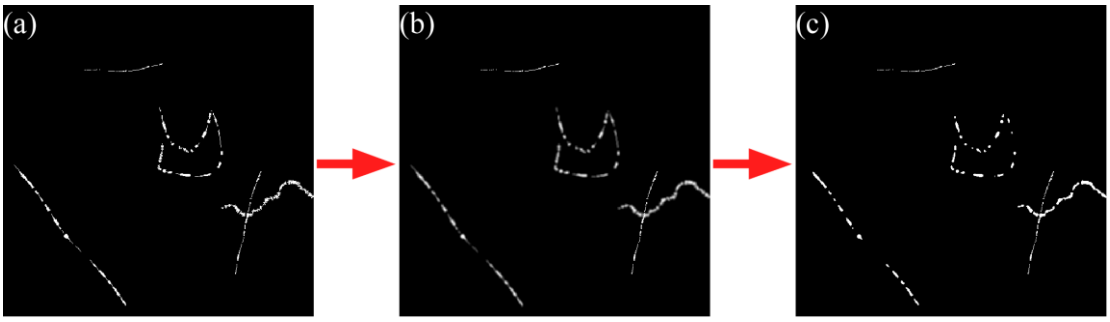

Figure 13: Generation process of imitation DNA segments which are artificially cut for generation of training input images $\boldsymbol{x}$ of $\boldsymbol{G}$ for training of $\boldsymbol{M}_{\mathrm{conn}}$.

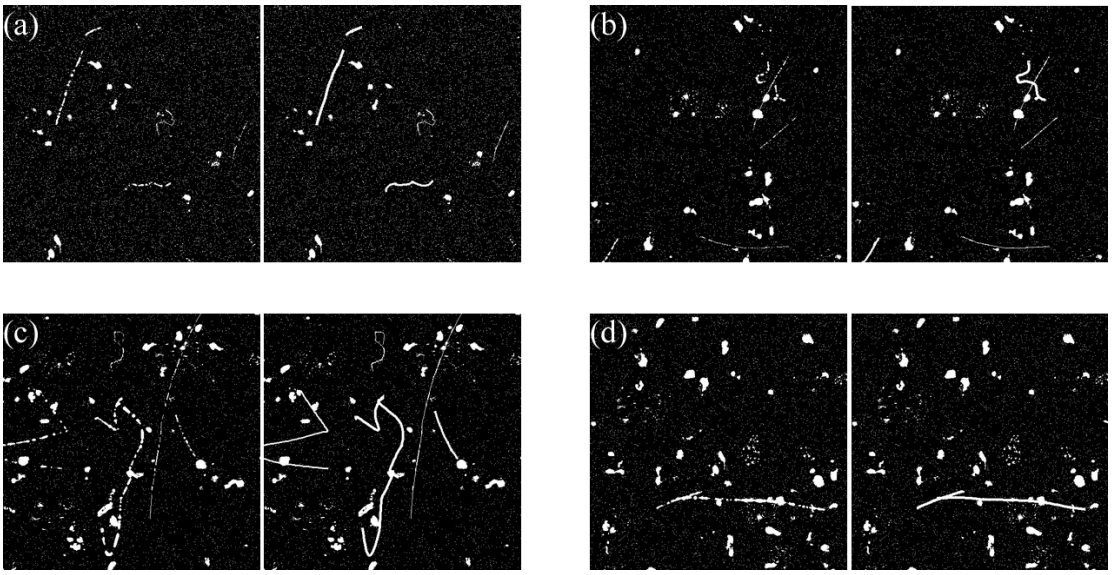

Figure 14: Four examples of input images $\boldsymbol{x}$ of $\boldsymbol{G}$ (left) and expected output images $\boldsymbol{y}^{\mathrm{c}}$ of $\boldsymbol{G}$ (right) for the training of $\boldsymbol{M}_{\mathrm{conn}}$.

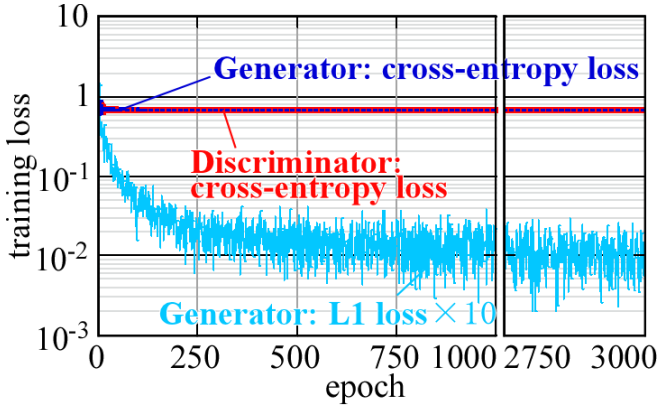

Figure 15: Evolution of training loss during the training of $\boldsymbol{M}_{\mathrm{conn}}$.

3.6. Training of $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$ and $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$ for extraction of DNA

$\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$ is trained to extract DNA segments from input images. Figure 16 shows four examples of the training data set $(\boldsymbol{x}, \boldsymbol{y}^{\mathrm{c}})$ for $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$ generated by the software. The input image $\boldsymbol{x}$ is generated by almost the same procedure of generation of $\boldsymbol{x}$ for the training of $\boldsymbol{M}_{\mathrm{conn}}$ except for mixing cut DNA segments with uncut segments. By drawing pieces of cut DNA segments in $\boldsymbol{x}$ with uncut segments, $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$ is expected to learn the difference of the pieces of cut DNA segments from impurities which should be removed. The expected output images $\boldsymbol{y}^{\mathrm{c}}$ is generated by drawing impurities and noise in gray. We note that the training fails if any impurities and noise are not drawn in $\boldsymbol{y}^{\mathrm{c}}$, because $\boldsymbol{G}$ outputs an image filled with black in the beginning in that case, and $\boldsymbol{D}$ cannot distinguish the difference between $\boldsymbol{y}^{\mathrm{c}}$ and $\boldsymbol{y} = \boldsymbol{G}(\boldsymbol{x})$ in the early stage of training when the performance of $\boldsymbol{D}$ is poor. By drawing impurities and noise in gray, it is possible to suppress $\boldsymbol{G}$ from outputting the black image.

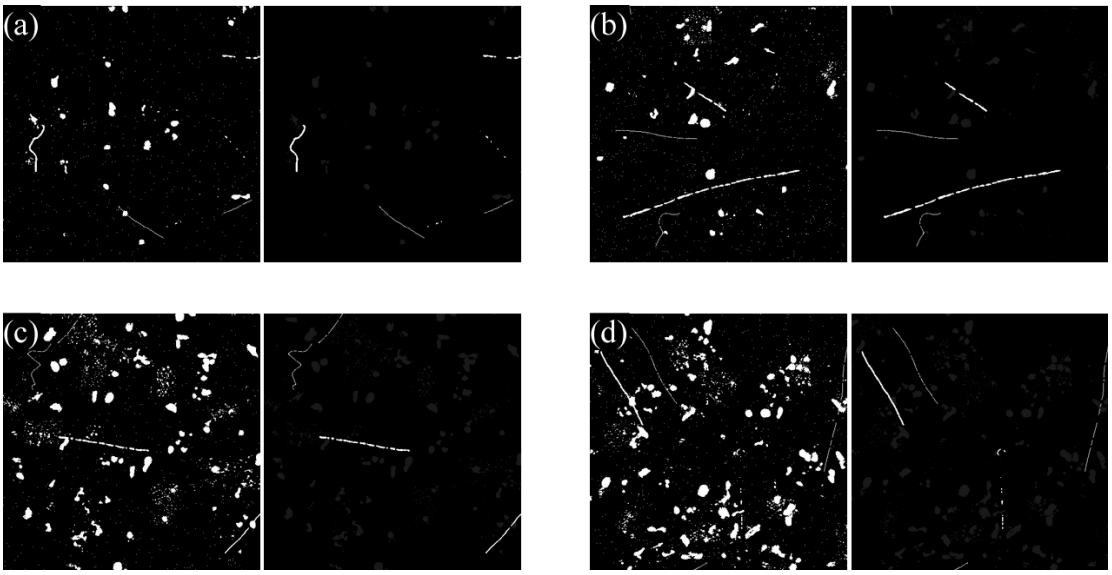

Figure 16: Four examples of input images $\boldsymbol{x}$ of $\boldsymbol{G}$ (left) and expected output images $\boldsymbol{y}^{\mathrm{c}}$ of $\boldsymbol{G}$ (right) for the training of $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$.

In addition to $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$, $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$ is also trained to extract DNA segments from input images. Figure 17 shows two examples of training data set $(\boldsymbol{x}, \boldsymbol{y}^{\mathrm{c}})$ for $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$ generated by the software. The images below are the enlarged images of the region surrounded by red square frames. The input image $\boldsymbol{x}$ is generated by the same procedure of $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$. The expected output images $\boldsymbol{y}^{\mathrm{c}}$ is generated by drawing DNA segments in black after impurities and noise is drawn in white. As shown in Fig. 18, using $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$, images of DNA segments are obtained by subtracting the output images $\boldsymbol{y} = \boldsymbol{G}(\boldsymbol{x})$ from the original input images $\boldsymbol{x}$.

At the step-B4, two DNA extracted images obtained by $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$ and $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$ are superimposed. Both $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$ and $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$ have the probability to eliminate parts of DNA segments by misjudging as impurities. By superimposing two images obtained by different models, the probability of misjudgment can be reduced. Figure 19 shows the evolution of training loss during the training of $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$ and $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$.

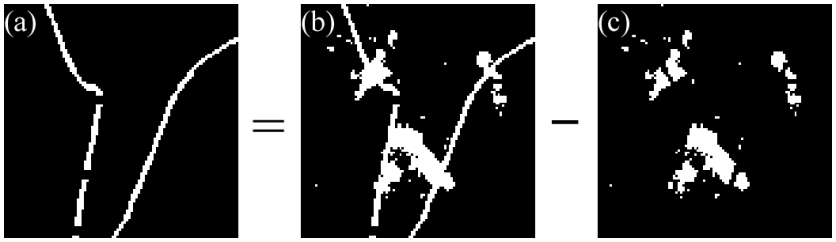

Figure 18: Method of DNA extraction by $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$. (a) Image of DNA segments. (b) Input images $\boldsymbol{x}$ of $\boldsymbol{G}$. (c) Output images $\boldsymbol{y} = \boldsymbol{G}(\boldsymbol{x})$. Image (a) is obtained by subtracted the image (c) from the image (b).

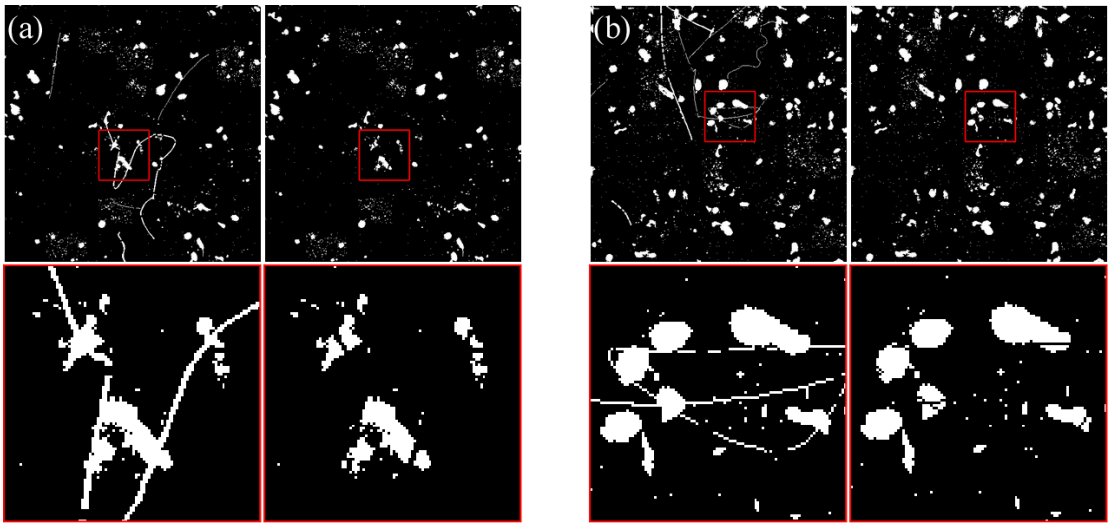

Figure 17: Two examples of input images $\boldsymbol{x}$ of $\boldsymbol{G}$ (left) and expected output images $\boldsymbol{y}^{\mathrm{c}}$ of $\boldsymbol{G}$ (right) for the training of $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$. The images below are the enlarged images of the region surrounded by red square frames.

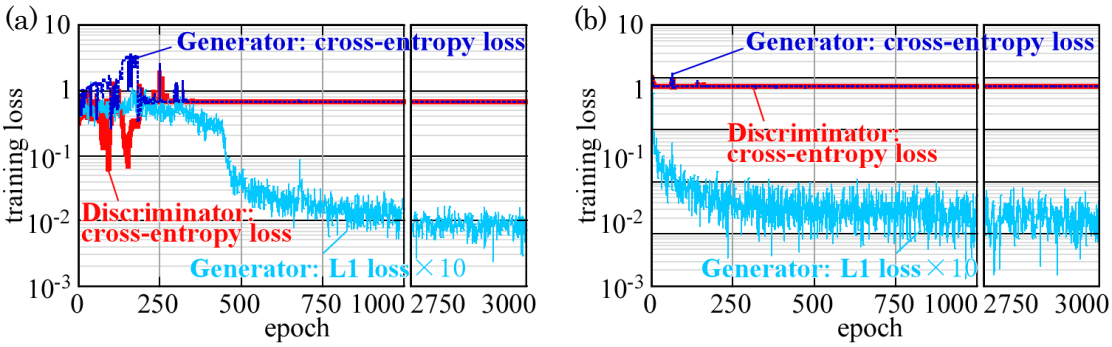

Figure 19: Evolution of training loss during the training of (a) $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$ and (b) $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$.

3.7. Training of $\boldsymbol{M}_{\mathrm{noise}}$ for noise reduction

$\boldsymbol{M}_{\mathrm{noise}}$ is trained to remove noise while remaining DNA segments. The opening method is known as a classical noise reduction method. However, in the method, skeletonized lines one pixel width are also considered as noise and be removed. In this study, deep learning model is trained to separate DNA lines of one pixel width from noise and remove only the noise. Figure 20 shows four examples of training data set $(\boldsymbol{x}, \boldsymbol{y}^{\mathrm{c}})$ for $\boldsymbol{M}_{\mathrm{noise}}$ generated by the software. The input image $\boldsymbol{x}$ is generated by the same procedure of $\boldsymbol{M}_{\mathrm{conn}}$. Expected output images $\boldsymbol{y}^{\mathrm{c}}$ are generated as images as no noise. We note $\boldsymbol{M}_{\mathrm{noise}}$ can apply even for noise removal with skeletonized lines whose thickness is 1 pixel while opening method which is a conventional noise removal method cannot remain skeletonized lines. Figure 21 shows the evolution of training loss during the training of $\boldsymbol{M}_{\mathrm{noise}}$.

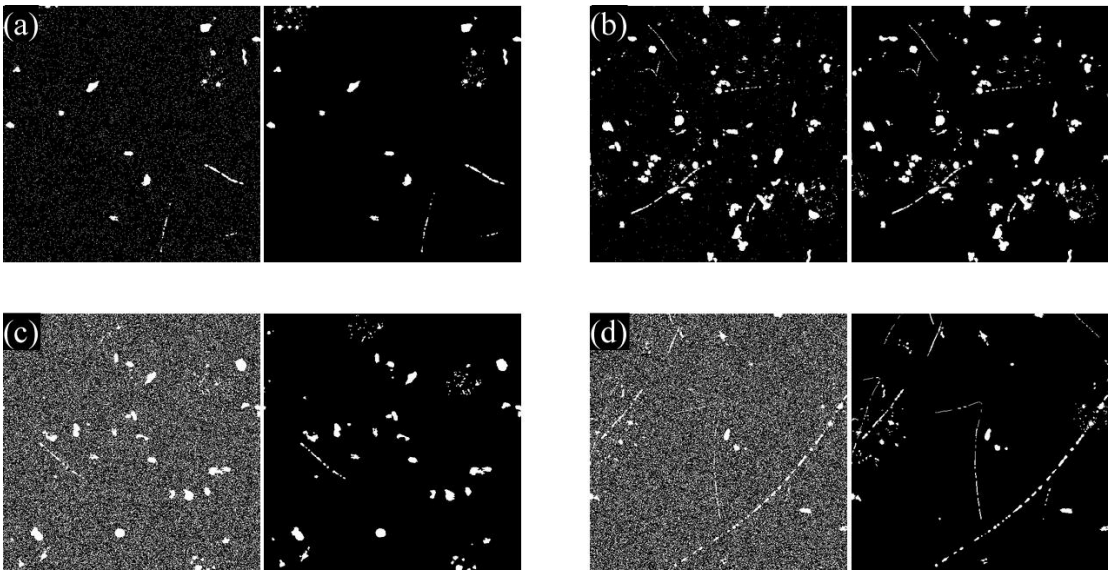

Figure 20: Three examples of input images $\boldsymbol{x}$ of $\boldsymbol{G}$ (left) and expected output images $\boldsymbol{y}^{\mathrm{c}}$ of $\boldsymbol{G}$ (right) for the training of pix2pix model $\boldsymbol{M}_{\mathrm{noise}}$.

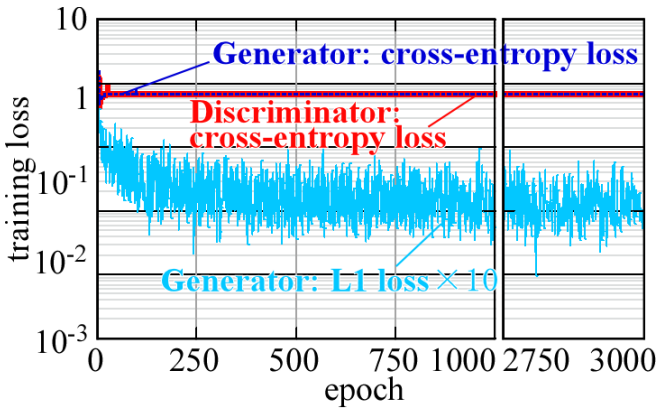

Figure 21: Evolution of training and validation loss during the training of $\boldsymbol{M}_{\mathrm{noise}}$.

3.8. Results

Figure 22 shows an example of evolution of an actual fluorescence microscope image from step-B1 to step-B9. As explained in section 3.1, $\boldsymbol{M}_{\mathrm{param}}$ is applied to predict the parameters for image processing by OpenCV (step-A1 to step-A6) in step-B1. It is confirmed that DNA segments are clearly binarized by the image processing with predicted parameters although some DNA segments are separated due to the problem of local shading of the input image. $\boldsymbol{M}_{\mathrm{conn}}$ is applied to connect separated DNAs segments in step-B2. After that, $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$ and $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$ are applied to remove impurities and curled DNA segments, and extract measurable DNA segments in step-B3a and step-B3b. It is seen that some DNA segments that are extracted in one model are not extracted in the other, or vice versa. At step-B4, both images obtained by $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$ and $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$ are superimposed. Skeletonization is performed in step-B5. In

this image, it is seen that some DNA segments are disconnected. To connect the disconnected segments, $\boldsymbol{M}_{\mathrm{conn}}$ is applied again in step-B7 after skeletonized segments are dilated at step-B6. The image of step-B7 shows that many of the cut DNA segments are connected. In step-B8, skeletonization is applied again. Finally, $\boldsymbol{M}_{\mathrm{noise}}$ is applied for noise reduction. It is seen that the skeletonized segments are remained even though noise is removed in step-B9.

Figure 23 shows the six examples of results of automatic image processing. Left images show the original input images obtained by actual fluorescence microscope. Middle images are the image after step-B1 is applied. Right images show the final images after step-B9 is applied. Even for various input images with different contrast and brightness, DNA segments are almost correctly extracted after the parameters are automatically estimated, although some DNA segments that should be separated are connected, or some of faint parts of DNA segments are eliminated. By applying the measurement method explained in section 2, total length of DNA segments can be measured. In the near future, we are planning to reveal the accuracy of the automatic measurements by comparing them with data already measured manually. We confirmed that DNA segments can be extracted by the procedure in most cases. However, in some special cases, we found some additional treatment is necessary to apply manually for the DNA extraction. Fig. 23 (f) is one of the special cases. In this case, signal-to-noise ratio of input image is too low to extract DNA segments. Therefore, to reduce the noise, $\boldsymbol{M}_{\mathrm{noise}}$ is manually applied after step-B1 is applied as shown in Fig. 23 (f2). By applying the procedure from step-B2 to step-B9 to the image of Fig. 23 (f2), DNA segments is successfully extracted.

4. Summary

To measure DNA length and number of DNA segments in fluorescence microscope images, image processing method using OpenCV is developed. The method has control parameters to measure for various input images which is captured under different experimental conditions. To realize measurement automation, a deep leaning model $\boldsymbol{M}_{\mathrm{param}}$ using CNN is developed. The model predicts the control parameters. Moreover, to extract DNA segments and remove images of impurities and noise from the input images, four deep learning models $\boldsymbol{M}_{\mathrm{conn}}$, $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{p}}$, $\boldsymbol{M}_{\mathrm{ext}}^{\mathrm{n}}$, and $\boldsymbol{M}_{\mathrm{noise}}$ using pix2pix are developed. Applying these models, we succeeded in automatically measuring the length of the DNA segments for most input images. It is also confirmed that some input images with a low signal-to-noise ratio can be measured by manually removing noise by applying $\boldsymbol{M}_{\mathrm{noise}}$ in the appropriate steps.

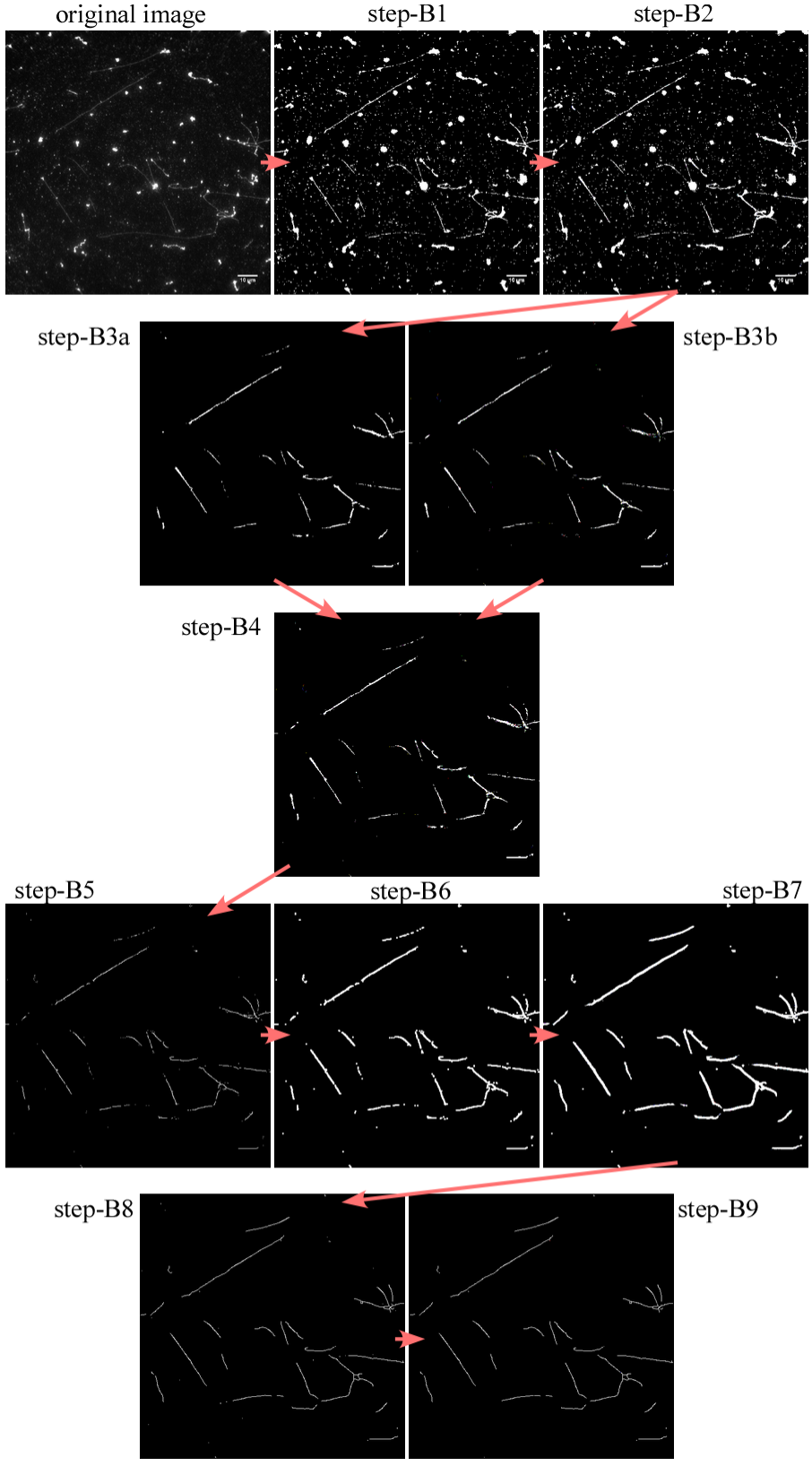

Figure 22: Example of evolution of an actual fluorescence microscope image from step-B1 to step-B9.

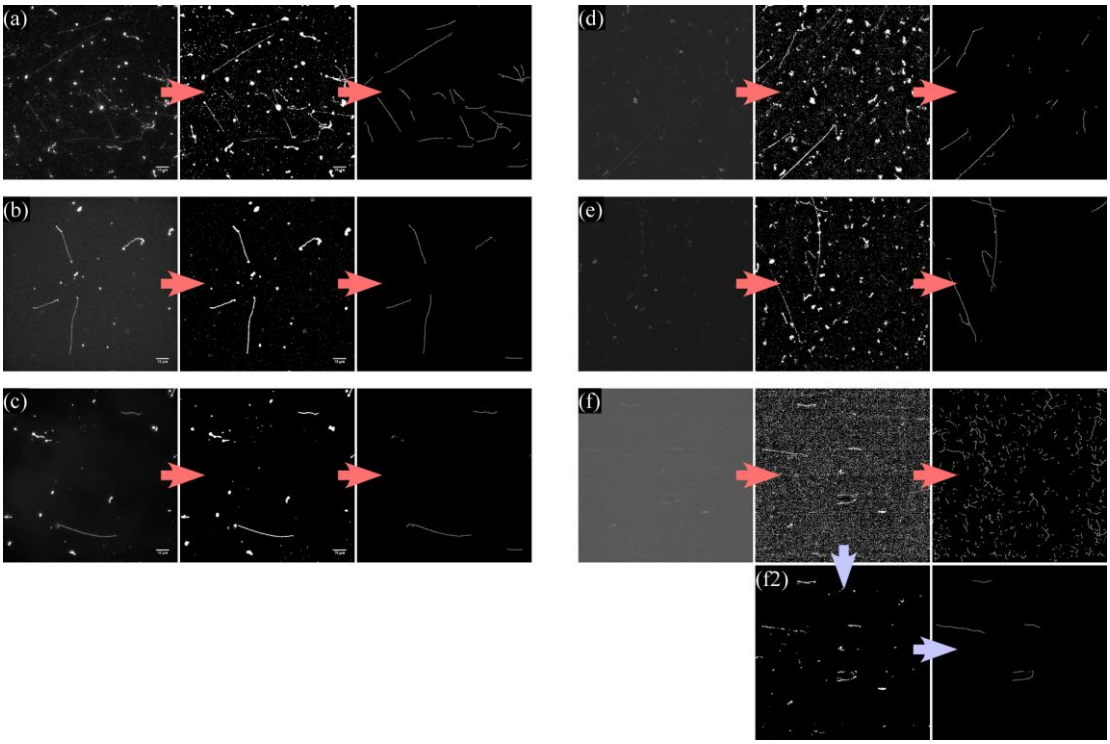

Figure 23: Six examples of results of automatic image processing. Left images are the original input images obtained by actual fluorescence microscopes. Middle images are the image after step-B1. Right images are the final images after the processing (step-B9).

Acknowledgement

The research was partially supported by Grant-in-Aid for Scientific Research, No.18K13528 from the Japan Society for the Promotion of Science, and by the NIFS Collaborative Research Program NIFS21KKGS030 and NIFS20KNSS149. The computations were performed using the JFRS-1 supercomputer system at Computational Simulation Centre of International Fusion Energy Research Centre (IFERC-CSC) in Rokkasho Fusion Institute of QST (Aomori, Japan), Research Center for Computational Science (Okazaki, Aichi, Japan) and Plasma Simulator of NIFS (Toki, Gifu, Japan) .

References

[1] R. Kubota, Y. Yamashita, T. Kenmotsu, Y. Yoshikawa, K. Yoshida, Y. Watanabe, T. Imanaka, K. Yoshikawa: *Double-Strand Breaks in Genome-Sized DNA Caused by Ultrasound*, ChemPhysChem, 18 (2017), 959.

[2] Y. Yoshikawa, M. Suzuki, N. Yamada, K. Yoshikawa: *Double-strand break of giant DNA: protection by glucosyl-hesperidin as evidenced through direct observation on individual DNA molecules*, FEBS Letters, 566 (2004), 39-42.

[3] Y. Ma, N. Ogawa, Y. Yoshikawa, T. Mori, T. Imanaka, Y. Watanabe, K. Yoshikawa: *Pro-*

tective effect of ascorbic acid against double-strand breaks in giant DNA: Marked differences among the damage induced by photo-irradiation, gamma-rays and ultrasound, Chemical Physics Letter, 638 (2015), 205-209.

[4] Y. Yoshikawa, T. Mori, N. Magome, K. Hibino, K. Yoshikawa: *DNA compaction plays a key role in radioprotection against double-strand breaks as revealed by single-molecule observation*, Chemical Physics Latter, 456 (2008), 80-83.

[5] Y. Yoshikawa, T. Mori, M. Suzuki, T. Imanaka, K. Yoshikawa: *Comparative study of kinetics on DNA double-strand break induced by photo and gamma-irradiation: Protective effect of water-soluble flavonoids*, Chemical Physics Latter, 501 (2010), 146-151.

[6] M. Noda, Y. Ma, Y. Yoshikawa, T. Imanaka, T. Mori, M. Furuta, T. Tsuruyama, K. Yoshikawa: *A single-molecule assessment of the protective effect of DMSO against DNA double-strand breaks induced by photo-and γ-ray irradiation, and freezing*, Scientific Reports, 7 (2017), 8557.

[7] Y. Hatano, Y. Konaka, H. Shimoyachi, T. Kenmotsu, Y. Oya, H. Nakamura: *Kinetics of double strand breaks of DNA in tritiated water evaluated using single molecule observation method*, Fusion Engineering and Design, 146 (2019), 100-102.

[8] T. Wada, A. Koike, S. Yamazaki, K. Ashizawa, F. Sun, Y. Hatano, H. Shimoyachi, T. Kenmotsu, T. Ikka, Y. Oya: *Protective behavior of tea catechins against DNA double strand breaks by radiations with different Linear Energy Transfer (LET)*, submitted to Fusion Engineering and Design.

[9] S. Fujiwara, H. Nakamura, H. Li, H. Miyanishi, T. Mizuguchi, T. Yasunaga, T. Otsuka, Y. Hatano, S. Saito: *Computational strategy for studying structural change of tritium-substituted macromolecules by a beta decay to helium-3*, Journal of Advanced Simulation in Science and Engineering, 6 (2019), 94-99.

[10] H. Nakamura, H. Miyanishi, T. Yasunaga, S. Fujiwara, T. Mizuguchi, A. Nakata, T. Miyazaki, T. Otsuka, T. Kenmotsu, Y. Hatano, S. Saito: *Molecular dynamics study on DNA damage by tritium disintegration*, Japanese Journal of Applied Physics, 59 (2020), SAAE01.

[11] T. Y. Zhang, C. Y. Suen: *A Fast Parallel Algorithm for Thinning Digital Patterns*, Communications of the ACM, 27:3 (1984), 236.

[12] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, L. Wang, G. Wang, J. Cai, T. Chen: *Recent advances in convolutional neural networks*, arXiv preprint arXiv: 1512.07108 (2015).

[13] P. Isola, J. Y. Zhu, T. Zhou, A. A. Efros: *Image-to-Image Translation with Conditional Adversarial Networks*, arXiv preprint arXiv: 1611.07004 (2016).

[14] D. Kingma, J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv preprint arXiv: 1412.6980 (2014).