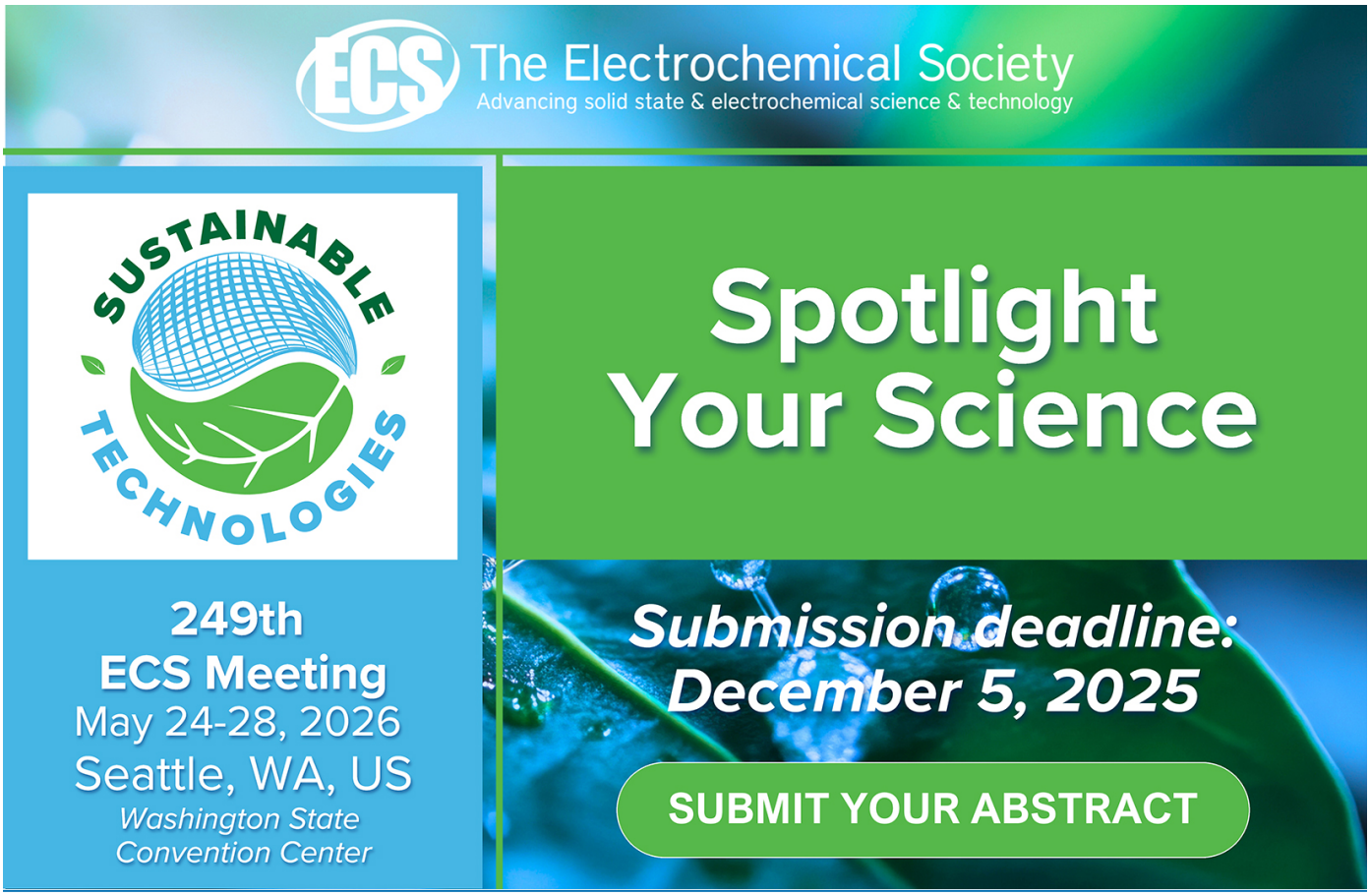
Japanese Journal of
Applied Physics

REGULAR PAPER

Automated measurement method based on deep learning for cross-sectional SEM images of semiconductor devices

To cite this article: Yutaka Okuyama and Takeshi Ohmori 2023 *Jpn. J. Appl. Phys.* **62** SA1016



Japanese Journal of Applied Physics **62**, SA1016 (2023) **REGULAR PAPER**
https://doi.org/**10.35848/1347-4065/ac923d**

Automated measurement method based on deep learning for cross-sectional SEM images of semiconductor devices

Yutaka Okuyama* and Takeshi Ohmori

Research & Development Group, Hitachi, Ltd., Kokubunji-shi, Tokyo 185-8601, Japan

*E-mail: yutaka.okuyama.tw@hitachi.com

Received May 20, 2022; revised August 31, 2022; accepted September 15, 2022; published online November 14, 2022

Feature length measurement in cross-sectional scanning electron microscopy images of modern semiconductor devices is time-consuming and laborious. We propose an automated measurement method based on deep learning technology and applied it to trench pattern images. The method combines two image-recognition tasks: (1) object detection for determining the coordinates of each unit of a pattern and (2) semantic segmentation for obtaining the boundaries of each area (mask, substrate, and background). By combining the results of these two tasks, typical feature lengths, such as width and depth, are precisely and immediately measured. The extraction speed of the proposed method is 240 times faster than manual measurement and provides measurement results independent of the engineer's skills.

1. Introduction

Nano-level processing for semiconductor devices has been increasing in complexity due to a structural change of device elements from two-dimensional (2D) to 3D,[1–4] which has increased process-development cost and time. Plasma etching is one such time-consuming and costly process, but it is an important process for nano-level fabrication of semiconductor devices. Etching-process development generally consists of a cycle of four steps: (i) etching recipe plan, (ii) etching test samples using planned recipes, (iii) observing cross-sectional scanning electron microscopy (SEM) images of etched samples, and (iv) measuring feature profiles in the SEM images. These four steps should be cycled until obtaining a target profile. The profile-measurement step should be semi-automatically or manually done by engineers with measurement software on a PC and typically requires several tens of minutes per image. For rapid development of the etching processes, measurement time of the feature profiles should be greatly reduced.

There are unique difficulties in measuring feature profiles in cross-sectional SEM images. Figure 1 is an example of the cross-sectional SEM images used in this study. Images have various profiles in accordance with the process recipes or a target profile. Brightness changes from image to image due to SEM-photography conditions. The white band is sometimes blurry or thick, about 8 pixels (or 8 nm), at the region signified with white arrows in Fig. 1. There are hindrance signals from the back structure signified with red arrows in Fig. 1. In the case of repeated line and space (*L*/*S*) patterns, four repetitions in Fig. 1, it is necessary to estimate the statistics of these patterns.

Engineers currently use commercial tools to measure feature profiles in cross-sectional SEM images. These tools provide a semi-automatic measurement function based on an image-processing library and macro script. Since the image-processing library commonly uses only the luminous intensity for the edge/contour detection,[5] manual adjustment of the parameters, such as the threshold of the intensity, is required for each image. For example, the thin interface between the silicon dioxide (SiO_2) mask and silicon (Si) line (a pillar part of substrate) in Fig. 1 is easily recognized by the human eye but is difficult to detect for such a library. Therefore, fully automated measurement is not possible with these tools. To the best of our knowledge, there is no precedent for automated measurement of feature profiles in cross-sectional SEM images.

Deep learning has emerged in various fields such as computer vision, speech recognition, and natural language processing, etc.[6] Deep convolutional neural networks (DCNNs) have pushed the performance of computer vision systems to soaring heights on a broad array of high-level problems including image classification[7] and object detection.[8] Therefore, it is natural to expect that time-consuming and laborious tasks, such as profile measurement in SEM imaging, can be fully automated by using DCNNs.

We propose a measurement method that involves using image-recognition models based on DCNNs and apply it to etching profiles of typical *L*/*S* patterns used for the semiconductor manufacturing process.

2. Methods

2.1. Images and measured feature lengths

Samples we prepared had patterned SiO_2 masks with half pitches of *L*/*S* patterns of 50, 90, and 150 nm on a Si substrate. These *L*/*S* patterns were transferred to the Si substrate during plasma etching. A typical inductively coupled plasma (ICP) etcher was used for the etching. Cross-sectional images of the etched samples were observed by field emission (FE)-SEM SU9000 (Hitachi, Ltd.). To create a diverse dataset, various etching profiles were prepared by changing eight control parameters in a recipe for the ICP etcher. These recipe parameters are pressure, power, bias, gas-flow rates of sulfur hexafluoride (SF_6), carbon tetrafluoride (CF_4), trifluoro methane (CHF_3), oxygen (O_2), and etching time. We fabricated 57 etched samples for a half pitch (hp) of 50 nm (actual *L*/*S* of 70 nm and 30 nm), 33 for a hp of 90 nm (actual *L*/*S* of 135 nm and 45 nm), and 33 for a hp of 150 nm (actual *L*/*S* of 160 nm and 140 nm).

Figure 2(a) is an SEM image of the pre-etching structure for a hp of 50 nm. This is a state after the tetra ethoxy silane (TEOS) cut process. Note that the shape of the mask/substrate interface is not flat but curved. The white dotted curve indicating an interface was added for clarity in this figure. Figure 2(b) is an example of an SEM image for a hp of 50 nm sample. The image quality is 8-bit grayscale in JPG

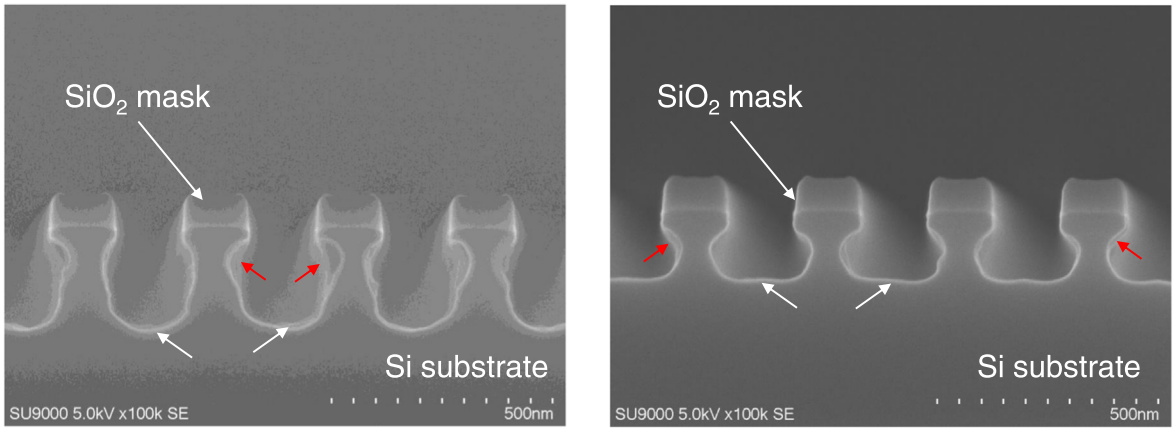


Fig. 1. (Color online) Typical cross-sectional SEM image with hp of 150 nm. White arrows point to a thick white band at the trench bottom and red arrows point to a hindrance signal stemming from the back structure.

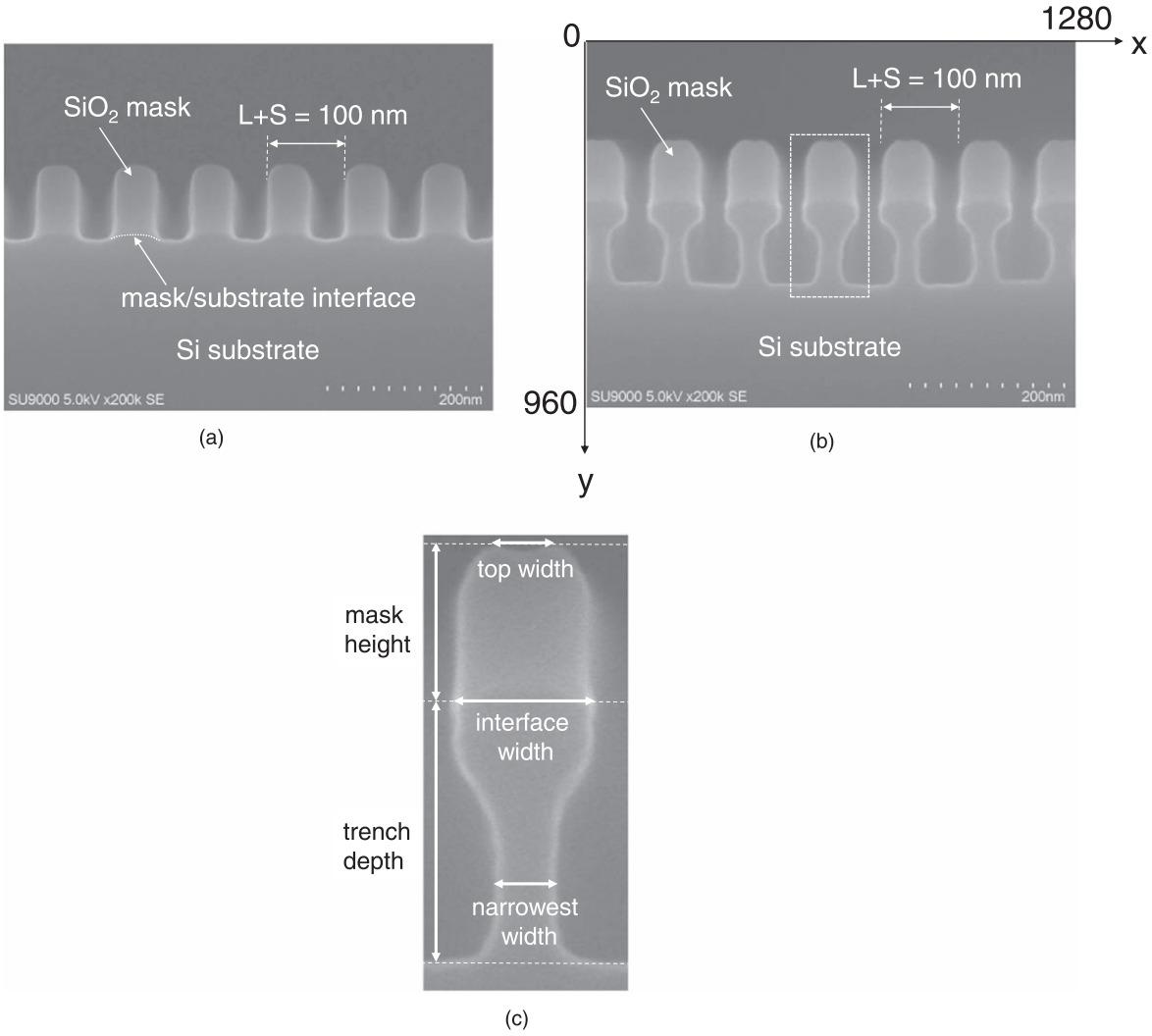


Fig. 2. (a) Mask and substrate structure with hp of 50 nm before etching process. White text in the figure is drawn for an explanation by authors. A white dotted curve is drawn to clarify curvature of mask/substrate interface. (b) Sample image for hp of 50 nm sample. Coordinate origin is set at the left corner. Horizontal axis is the *x*-direction and vertical axis is the *y*-direction. The region inside the white dotted rectangle is shown in (c). (c) Extracted unit of *L*/*S* pattern from (b). Five extracted feature lengths are mask height, and trench depth, mask top width, mask/Si line interface width, and narrowest Si line width.

format, and the size is 960×1280 pixels. The origin of coordinates was set at the left upper point of the image. The *X*-axis is horizontal, and the *y*-axis is vertical. Coordinates are measured in pixels. Figure 2(c) is the enlarged view inside the white dotted rectangle shown in Fig. 2(c) and represents five measured feature lengths in an *L*/*S* pattern: mask top width, mask/Si line interface width, narrowest Si line width, mask height, and trench depth. Mask top width is defined as a distance between two peaks appearing on the top part of a mask and is set to zero for a single peak. Interface width is defined as the distance between both end points of a curved interface of the mask and Si.

2.2. Measuring feature lengths

We first investigated how to use image recognition with DCNNs to measure feature lengths of etching profiles in cross-sectional SEM images. The following are the four major tasks for image recognition in computer vision.

(a) Image classification:[7,9,10] Classes of the objects contained in the image are inferred.
(b) Object detection:[8,11–14] Classes and positions (usually represented with a rectangle called a *bounding box*) of objects contained in images are inferred.
(c) Semantic segmentation:[15–18] Classes of objects in images are determined at pixel level. The background is considered as one class, and its pixels are also identified.
(d) Instance segmentation:[19–21] In addition to semantic segmentation, individuals in the same class are distinguished.

A straightforward method of measuring the feature lengths in cross-sectional SEM images is to detect contours or boundaries of each area or class such as the SiO_2 mask and Si substrate. Semantic segmentation is suitable for such a task because even the interface between the SiO_2 mask and Si line is easily distinguishable in segmented images. For the etching profiles shown in Fig. 2(b), the areas to be distinguished are the SiO_2 mask, Si substrate, and background. Therefore, there are three classes in our images. As described in the introduction, since an *L*/*S* pattern is repeated multiple times in one image, it is necessary to obtain statistics such as the mean and standard deviation of measured lengths. To calculate these statistics from the contours of an etching profile image, it is necessary to crop the contours for each unit of an *L*/*S* pattern shown in Fig. 2(c). The cropping region should be adapted in accordance with the etching profile in a unit. In other words, an object-detection method with high versatility is required that can respond to changes in an etching profile with various recipes. It is also desirable that a boundary (bounding box) for a unit of an *L*/*S* pattern be automatically obtained. This task is inside the scope of object detection. If the unit, as shown in Fig. 2(c), is defined as an object of the class *pattern* (hereafter, an italic *pattern* is used to indicate an object) for object detection, it is possible to automatically detect each region including the unit of *L*/*S* pattern by using an object-detection method with high versatility.

On the basis of the above considerations, to measure the feature lengths in etching profiles in cross-sectional SEM images efficiently, an optimal measurement method should use two image-recognition tasks: semantic segmentation and object detection.

Models for object detection and semantic segmentation based on DCNNs have been proposed since the development of the R-CNN[8,11] and FCN,[15] respectively, and their inference accuracy and speed have improved yearly. We use Faster R-CNN[12,13] as the object-detection model and SegNet[16] as the semantic-segmentation model for our method. The accuracies of these two models are slightly inferior compared with more recent models such as RepPoints v2[14] and CascadePSP,[18] but they are still frequently used.

2.3. Datasets

Dataset consists of SEM images and corresponding annotation data. All SEM images were shuffled and divided into training, validation, and test images in the ratio of 90:20:13. Each image was labeled with hp size and sample number such as 50 nm_8.jpg. Twelve images from the training dataset are shown in Fig. 3. The images of the first, second, and third rows represent cross-sections of the etched samples with a hp of 50, 90, and 150 nm, respectively. Various etching profiles were prepared using etching recipes with changing recipe parameters. Specifications of *L*/*S* patterns, imaging conditions, and the number of images are summarized in Table I.

We use the PASCAL VOC detection dataset[22,23] format for creating annotation data to train Faster R-CNN to detect

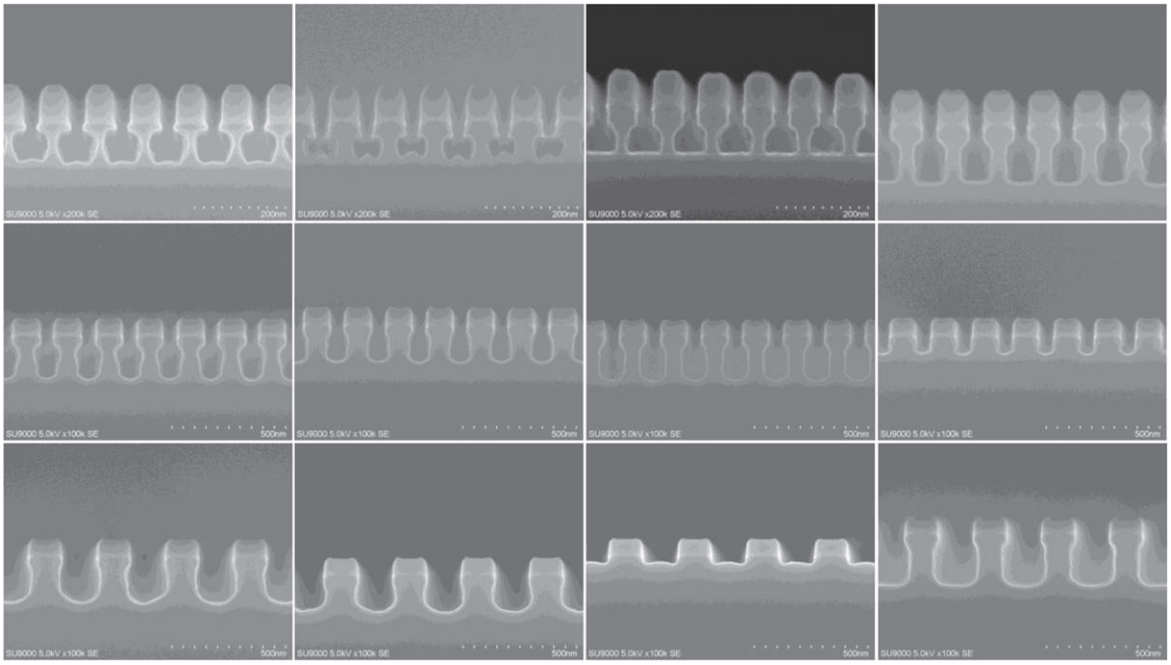


Fig. 3. Examples of SEM images included in the training dataset. Images of the first, second, and third rows represent cross-section of etched samples with hp of 50, 90, and 150 nm, respectively.

Table I. Specifications of prepared SEM images.

HP	Image size [pixels]	Magnification	Resolution [nm/pixel]	Number of images			
				Total	Training	Validation	Test
50	960 × 1280	200k	0.5	57	43	8	6
90		100k	1.0	33	25	6	2
150		100k	1.0	33	22	6	5

*pattern*s in an etching profile in a cross-sectional SEM image. This annotation data is an XML (extensible markup language) file, in which each bounding box is represented as two sets of coordinates: the upper left corner and the lower right corner of an object. Figure 4 shows the definition of bounding boxes in our study. This image contains five *pattern*s, and their bounding boxes are represented as five green rectangles. The *pattern* number is from left to right in the image. The origin position of the *x*-direction of each bounding box is set at the center of the trench width. The boundaries along the *y*-direction are set at a position of the mask top and trench bottom plus a margin of ten pixels; thus, the edges of the mask top and trench bottom are completely included in a bounding box. For the most left bounding box in Fig. 4, for example, the coordinates of the upper left and lower right corners are (126, 433) and (326, 667), respectively. In a normal object-detection task, each class is assigned to an isolated object such as a mask area in this image, which is clearly distinguished from the periphery. However, our original class *pattern* is assigned to the region consisting of the isolated mask and Si line. Since *L*/*S* patterns are periodic, the coordinates of bounding boxes are automatically calculated by detecting the rough contours of SiO_2 masks and Si substrate by using the edge detection library of OpenCV,[24] and the obtained coordinates are converted to the format of annotation data by Python script. Thanks to this script, this process finishes in about 5 min for all training and validation images.

We use the PASCAL VOC segmentation dataset[22,23] format for creating annotation data to train SegNet that can be utilized to detect contours of each area, namely three classes of SiO_2 mask, Si substrate, and background, in an etching profile in a cross-sectional SEM image. This annotation data is a PNG (portable network graphics) file that describes the class of each pixel by means of color or number. Figure 5 is an example of the annotation data. In our case, there are three classes: SiO_2 mask, Si substrate, and background. The pixels that belong to these classes are represented in black, red, and green in the image in Fig. 5, respectively. To create the annotation data, we use the graphical image annotation tool *lableme*.[25] This manual annotation takes around 16 h for all training and validation images. Unique class numbers 0, 1, and 2 are also assigned for each class. Table II summarizes the correspondence between the class name, class number, and color.

2.4. Proposed method of feature-length extraction based on object-detection and semantic-segmentation models

We explain the proposed method of extracting feature lengths in an etching profile in a cross-sectional SEM image by using both models of object-detection and semantic-segmentation models and by combining the results from both models. We first train the Faster R-CNN model and SegNet model with the datasets described above. The measurement flow of the proposed method is shown in Fig. 6. The flow consists of inference and measurement stages. At the inference stage, an SEM image is input into the two learned semantic-segmentation and object-detection models. From the semantic-segmentation model for detecting SiO_2 mask, Si substrate, and background area that belong to each class, a class-divided segmentation image and contours of each area are obtained. In parallel, from the object-detection model for detecting a unit of *L*/*S* pattern, the bounding boxes of detected *pattern*s are obtained. By transferring the coordinates of the detected bounding boxes into the segmentation image, every region containing a unit of *L*/*S* pattern in the segmentation image is

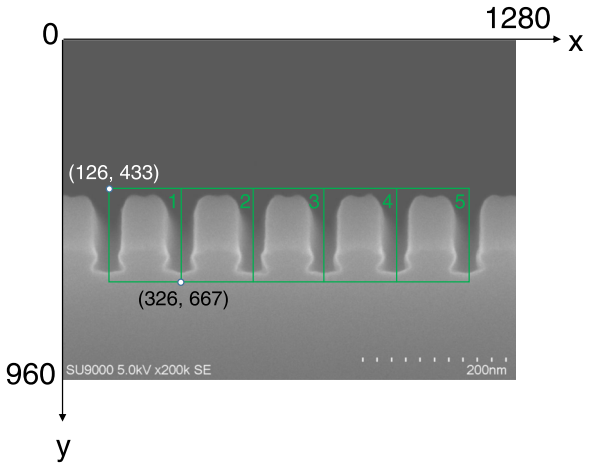


Fig. 4. (Color online) Example of bounding boxes used in annotation data for Faster R-CNN. Green lines represent bounding boxes. Bounding box is specified by two coordinates at the left upper point and right lower point in annotation data. Green numbers are *pattern* numbers.

Fig. 5. (Color online) Example of annotation data for SegNet. Three classes (background, mask, and substrate) are specified by color (black, red, green).

Table II. Correspondence of class name, class number, and color used in annotation data for SegNet.

Class	Number	Color (R, G, B)
Background	0	Black (0, 0, 0)
Mask	1	Red (128, 0, 0)
Substrate	2	Green (0, 128, 0)

cropped. At the measurement stage, seven key points corresponding to endpoints of five feature lengths on contours are identified with Python script, in which the corresponding key points are calculated on the basis of the curvature of contours. Feature lengths are calculated by taking into account the image resolution (nm/pixel) determined from the hp widths (50, 90, and 150 nm) included in the file name of the input image.

3. Results and discussion

We used the deep learning frameworks Chainer[26] and ChainerCV,[27] in which both Faster R-CNN and SegNet were implemented. Annotation and measurement steps were carried out with a CPU and training and inference steps were carried out with a GPU.

3.1. Results

In the training step of Faster R-CNN, data augmentation with a random horizontal flipping was done for the training and validation data. For the backbone VGG-16[9] in Faster R-CNN, the pre-trained model for the PASCAL VOC 2007 dataset[22,23] was used and then fine-tuned with our dataset. In accordance with the default setting of the implemented Faster R-CNN model, stochastic gradient descent with a learning rate of 10^{-3}, momentum of 0.9, and weight decay of 5×10^{-4} was used for optimization. The input image size was 960×1280 pixels, which is the original size obtained from SEM observation. The training of Faster R-CNN was done up to 50 epochs with a batch size of 1 and took around 20 min. In the training step of SegNet, we carried out data augmentation with random cropping to the size of 512×512 pixels for the training and verification data. We also carried out random horizontal flipping for the training data only. The optimizer and its parameters were the same as for the Faster R-CNN model. The model parameters were trained from scratch with our dataset. The training was done up to 500 epochs with a batch size of 3 and took around 8 h.

Figure 7 shows the inference results for object detection [Fig. 7(a)] and semantic segmentation [Fig. 7(b)] for the image 150 nm_5. The inference times for both Faster R-CNN and SegNet models were around 5 and 40 s for all 13 test images, respectively. The scores shown in Fig. 7(a) are the intersection-over-union (IoU) overlap with a ground-truth bounding box. A score of 1.00 means complete overlap. To exclude a partial unit of *L*/*S* pattern that may appear at the left and/or right edge of an image in which a part of feature lengths can be extracted, we only selected the *pattern*s with a score of 1.00 as detected results. There was no false detection of the *pattern* by the object detection in all 13 test images. As shown in Fig. 7(b), the thin mask/Si-line interface was accurately detected while ignoring the hindrance signal from the back structure. As expected, the unique difficulties in measuring cross-sectional SEM images mentioned in the introduction, i.e. detection of the thin interface and ignoring the signals from back structure, are resolved using semantic-segmentation model.

The results of extracted feature lengths for the two images (50 nm_38 and 150 nm_5) are shown in Figs. 8 and 9. The measurement time was around 5 s for all 13 test images, respectively. The extraction was successful in all 13 test images, including those not shown here. In Fig. 8, with respect to the mask top width, arrows and dimensions are not displayed because of a single peak of the mask top contour that is mentioned in Sect. 2.1.

It is difficult to obtain perfectly accurate feature lengths because the white band of the edge contours has a width and obtained lengths depend on which pixels on the white band

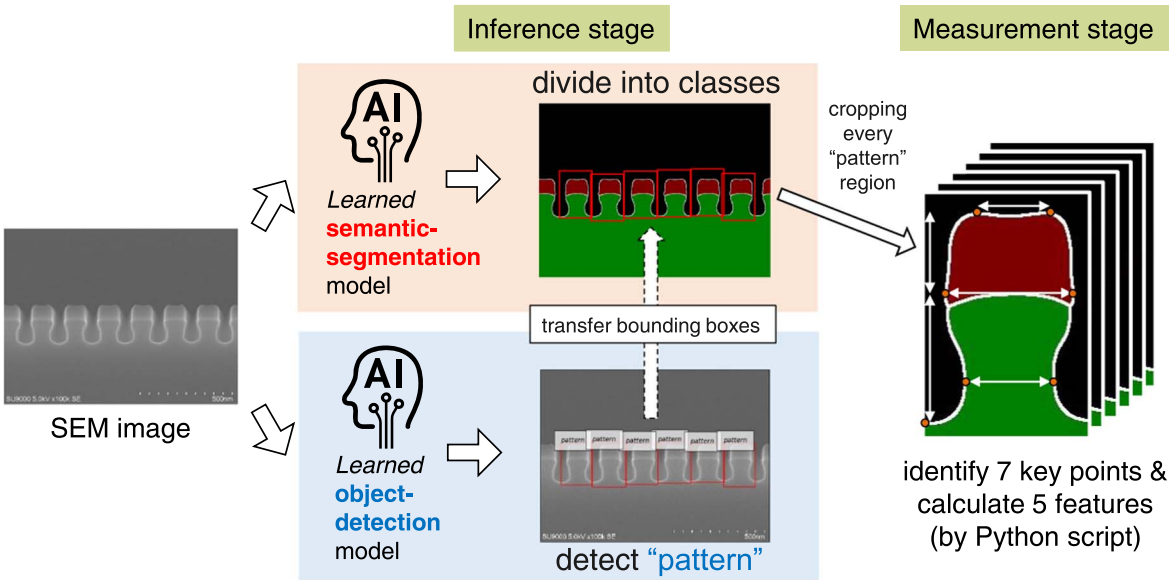


Fig. 6. (Color online) Flow of prediction and extraction steps of proposed method. At the prediction stage, the SEM image is input into two learned semantic-segmentation and object-detection models. Class-divided image and contours are obtained with the semantic-segmentation model. Bounding boxes of detected *pattern*s are obtained with the object-detection model. By transferring coordinates of detected bounding boxes into the segmented image, every *pattern* region in the segmented image is cropped. At extraction stage, seven key points on contours are identified then five feature lengths are calculated from their coordinates.

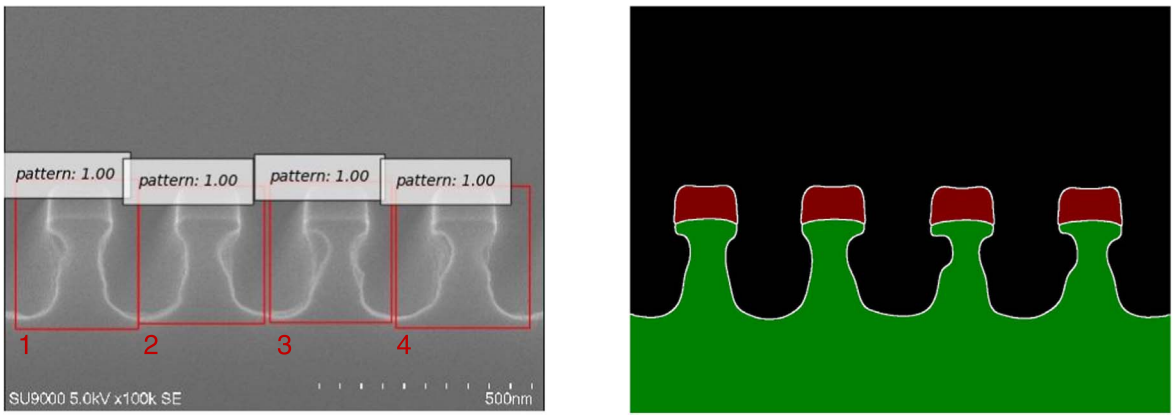


Fig. 7. (Color online) Inferred results for image 150 nm_5. (a) Result from object detection. (b) Result from semantic segmentation with detected contours.

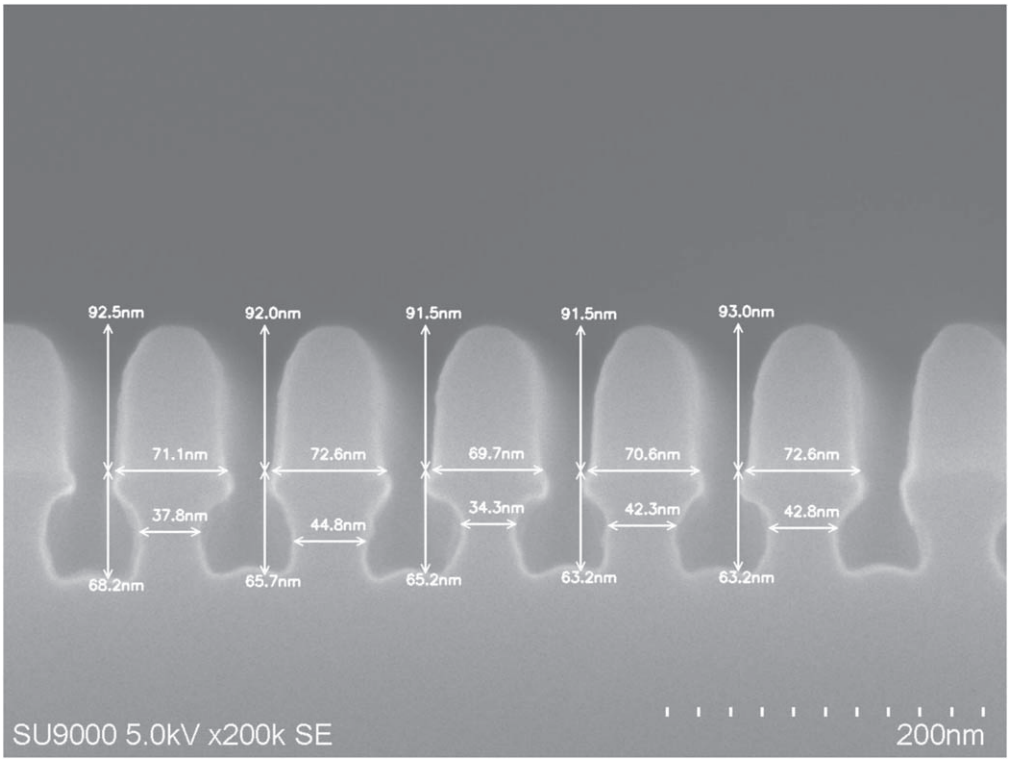


Fig. 8. Measured results for image 50 nm_38.

are selected as the edge points. Therefore, we evaluated the accuracy of the proposed method by comparing it with manual measurement by an expert measurement engineer. Tables III and IV show the comparison between manually measured feature lengths and automatically measured feature lengths using the proposed method for images 50 nm_38 and 150 nm_5, respectively. The left and right values separated by a slash represent manually measured and automatically measured feature lengths, respectively (unit of values is nm). The mean and standard deviation are also shown. The agreement percentage between manually measured and automatically measured feature lengths is shown in parentheses. The agreement was fairly good. The maximum difference (in bold) is at the narrowest width in both images. It was also found that the narrowest width had large values of standard deviations in both manually measured and automatically measured feature lengths.

3.2. Discussion

In this section we discuss the pros and cons of the proposed method. Table V shows the maximum absolute errors between the proposed method and manual measurement for all test images. The majority ranged from about 1 to 8 nm, which is acceptable from the image quality point of view because the contours in all SEM images have a white band with a maximum width of about 8 nm. The table shows that the proposed method can replace the manual measurement. However, there are two exceptional cases, i.e. 150 nm_7 and 150 nm_26 with a hp of 150 nm, in which the maximum difference is 10.6 nm at mask top width. Figure 10 shows a cropped region by a bounding box for a unit of *L*/*S* pattern (above) and the segmented image corresponding to the cropped region (below) of image 150 nm_7. It is clear that the automatically measured width is an underestimation. This is due to the two sharp peaks of the mask top becoming dull

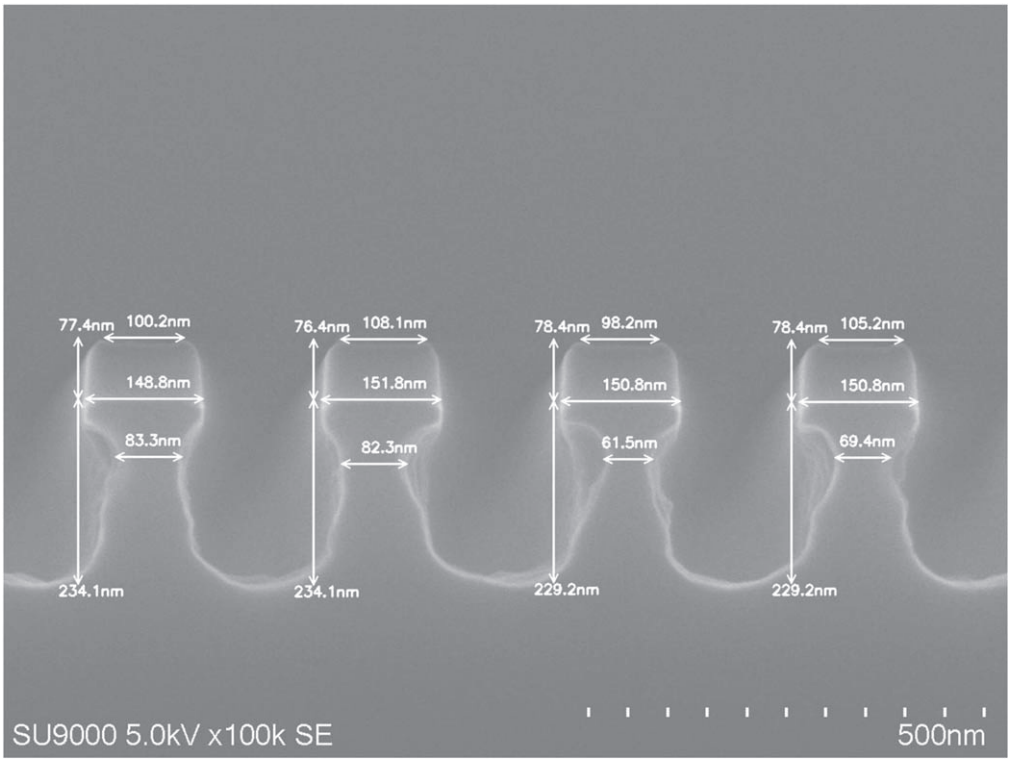


Fig. 9. Measured results for image 150 nm_5.

Table III. Comparison between manually measured and automatically measured feature lengths for image 50 nm_38 shown in Fig. 8. Mean and standard deviation are also shown. *Pattern* number is numbered from left to right in the image. Left and right values separated by slash represent manually measured and automatically measured feature lengths, respectively. The unit of values is nm. The number in parentheses is an agreement percentage. The maximum of difference is indicated in bold.

Pattern no.	Mask height	Trench depth	Interface width	Narrowest width
1	91.5/92.5 (99)	66.1/68.2 (97)	69.3/71.1 (97)	**35.2/37.8 (92)**
2	92.3/92.0 (100)	64.1/65.7 (97)	71.3/72.6 (98)	43.9/44.8 (98)
3	90.0/91.5 (98)	64.1/65.2 (98)	70.6/69.7 (99)	32.4/34.3 (94)
4	91.5/91.5 (100)	61.8/63.2 (98)	69.1/70.6 (98)	41.6/42.3 (98)
5	93.5/93.0 (99)	64.1/63.2 (99)	72.8/72.6 (100)	42.6/42.8 (97)
Mean	91.8/92.1 (100)	64.0 /65.1 (98)	70.6/71.3 (99)	39.0/40.4 (96)
Std. dev.	1.1 /0.6	1.3 /1.9	1.4/1.1	4.4/3.8

Table IV. Comparison between manually measured and automatically measured feature lengths for image 150 nm_5 shown in Fig. 9.

Pattern no.	Mask height	Trench depth	Top width	Interface width	Narrowest width
1	78.0/77.4 (99)	233.1/234.1 (99)	100.9/100.2 (99)	148.6/148.8 (100)	**79.5/83.3 (95)**
2	75.5/76.4 (99)	233.1/234.1 (99)	107.9/108.1 (100)	148.6/151.8 (98)	79.5/82.3 (97)
3	76.5/78.4 (98)	226.1/229.2 (99)	99.9/98.2 (98)	150.6/150.8 (100)	60.1/61.5 (98)
4	79.0/78.4 (99)	229.1/229.2 (100)	105.9/105.2 (99)	147.6/150.8 (98)	68.1/69.4 (98)
Mean	77.3/77.7 (100)	229.4/231.7 (99)	103.6/102.9 (99)	148.9/150.6 (99)	71.8/74.1 (97)
Std. dev.	1.3/0.8	2.0/2.5	3.3/3.9	1.1/1.1	8.2/9.1

in the segmented image. Therefore, this measurement error stems from the precision of semantic segmentation. This may be improved using a newer semantic-segmentation model[18] and/or a larger training dataset.

The required measurement times with the proposed method and manual measurement are compared in Table IV. Annotation and training times are for the 110 training and validation images. Inference and measurement times are for the 13 test images. Of course, annotation, training, and inference steps are not necessary for manual measurement. The total measurement and inference times were 50 s with the proposed method and about 200 min with manual measurement. Therefore, the proposed method's measurement speed is 240 times faster than the manual measurement. The effectiveness of the proposed method increases with the number of images for the measurement.

The proposed method, however, requires the creation of semantic-segmentation annotation data for training a semantic-segmentation model. The annotation data of all images should be provided by humans, which is still time-consuming. About 16 h were required to make about 100 images in this study. To reduce this annotation cost, it would

Table V. Maximum absolute errors of automatic measurement for 13 test images. Error was measured from manually measured feature lengths. The maximum difference is indicated in bold.

Image	Mask height	Trench depth	Top width	Interface width	Narrowest width
50 nm_8	2.5	5.0	3.5	3.8	2.0
50 nm_17	2.1	4.9	—	5.6	2.7
50 nm_20	4.0	3.0	—	6.8	2.3
50 nm_24	4.8	3.5	—	6.4	2.4
50 nm_29	3.8	2.1	—	4.4	1.1
50 nm_38	1.0	1.0	—	1.0	1.0
90 nm_17	3.6	3.2	7.8	6.7	4.3
90 nm_28	1.6	2.3	4.3	3.2	2.6
150 nm_5	1.0	1.0	1.0	1.0	1.0
150 nm_7	3.3	4.2	**10.6**	3.2	3.7
150 nm_13	6.3	1.7	8.2	5.2	2.7
150 nm_26	4.8	6.9	**10.6**	3.2	5.7
150 nm_29	5.8	4.6	4.3	3.2	5.2

Table VI. Comparison between times required for automatic and manual measurement. Annotation and training were for 110 training and validation images, whereas inference and measurement were for 13 test images. Annotation for semantic segmentation was carried out manually with a tool on a PC. Although training and inference times were on GPU, annotation and measurement times were on CPU.

Method		Annotation	Training	Inference	Measurement
DCNNs	Object detection	5 min	20 min	5 s	5 s
	Semantic segmentation	∼16 h	∼8 h	40 s	
	Manual	—	—	—	∼200 min

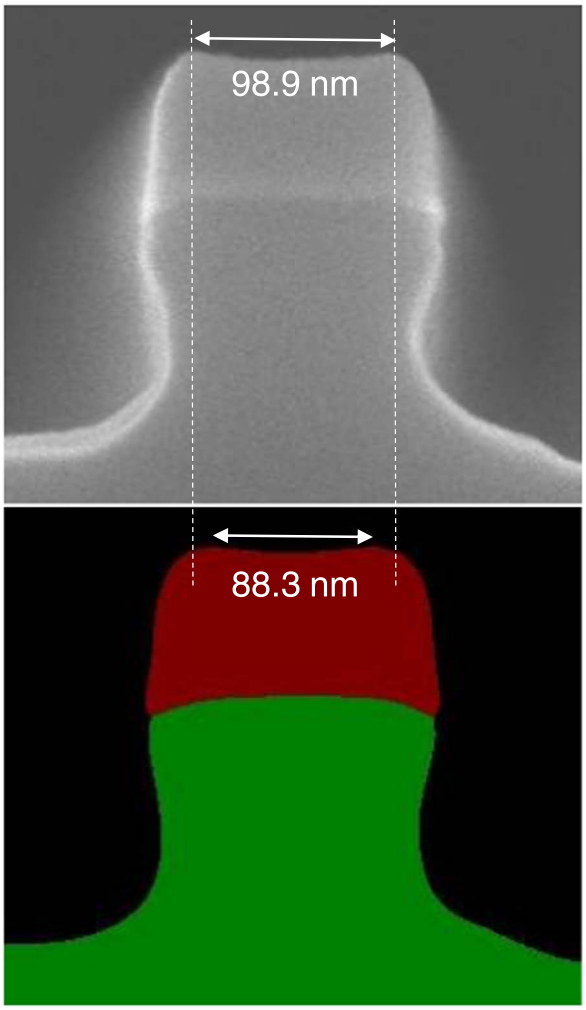


Fig. 10. (Color online) Extracted image for *pattern* No. 2 for image 150 nm_7 (above) and corresponding segmented image (below). Arrows in figures show mask top width measured manually (above) and automatically (below).

be interesting to use a data-efficient semantic-segmentation method such as a semi-supervised[28] or self-supervised[29,30] one for future work.

From the point of view of the training of DCNNs, it seems that the training dataset of size 90 used in our study is too small to train the model for input images, 512 × 512 pixels for the SegNet model and 960 × 1280 pixels for the Faster R-CNN model. Usually, a larger dataset is used to train the DCNNs even for a smaller pixel size of input images. The small diversity of datasets in this study is the reason the proposed method can be trained well. Various profiles are prepared by changing recipe parameters, but the variation in image diversity in this study was small as compared with that in other typical problems for generic object recognition.

Of course, the proposed method is not the only solution for automated measurement. An alternative solution may be a method with which key points of feature lengths are detected directly using a key-point detection model[31] or a multi-person human-pose estimation model.[32] With such a method, only one model should be trained, but the model must be trained again when the number and/or types of measured feature lengths are changed during etching-process development. We plan to investigate such a method for future work.

4. Conclusions

Feature-length measurement in cross-sectional SEM images of modern semiconductor devices is time-consuming and laborious. We proposed an automated measurement method for cross-sectional SEM images based on deep learning technology and applied it to trench patterns. This method combines two image-recognition tasks: (1) object detection is for determining the coordinates of each unit of a pattern and (2) semantic segmentation is for obtaining the boundaries of each area (mask, substrate, and background). The method

was successful for all test images. Our results show that the proposed method enables the measuring of feature lengths 240 times the speed of manual measurement with acceptable precision and independent of an engineer's skills. We hope this study will contribute to more rapid process development of advanced semiconductor devices in the future.

Acknowledgments

The authors would like to thank Yasutaka Toyoda, Masayoshi Ishikawa, and Kohei Matsuda for their valuable discussions on methods and results and Tatehito Usui for their experimental support.

1) C. Kim et al., IEEE J. Solid-State Circuits **53**, 124 (2018).
2) G. Bae et al., IEEE Int. Electron Devices Meeting, 2018, p. 338.
3) A. Goda, IEEE Trans. Electron Devices **67**, 1373 (2020).
4) International Roadmap for Devices and Systems 2021, Edition: More Moore https://irds.ieee.org/images/files/pdf/2021/2021IRDS_MM.pdf.
5) M. A. Groeber, B. K. Haley, M. D. Uchic, D. M. Dimiduk, and S. Ghosh, Mat. Char. **57**, 259 (2006).
6) I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, 2016).
7) A. Krizhevsky, I. Sutskever, and G. E. Hinton, Proc. Int. Conf. Neural Inf. Process. Syst., 2012, p. 1097.
8) R. Girshick, J. Donahue, T. Darrell, and J. Malik, Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., 2014, p. 580.
9) K. Simonyan and A. Zisserman, Proc. Int. Conf. Learn. Repr., 2015, p. 1.
10) K. He, X. Zhang, S. Ren, and J. Sun, Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, p. 770.
11) R. Girshick, J. Donahue, T. Darrell, and J. Malik, IEEE Trans. Pattern Anal. Mach. Intell. **38**, 142 (2016).
12) S. Ren, K. He, R. Girshick, and J. Sun, Proc. Int. Conf. Neural Inf. Process. Syst., 2015, p. 91.
13) S. Ren, K. He, R. Girshick, and J. Sun, IEEE Trans. Pattern Anal. Mach. Intell. **39**, 1137 (2017).
14) Y. Chen, Z. Zhang, Y. Cao, L. Wang, S. Lin, and H. Hu, Proc. Conf. Neural Inf. Process. Syst., 2020, p. 5621.
15) J. Long, E. Shelhamer, and T. Darrell, Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, p. 3431.
16) V. Badrinarayanan, A. Kendall, and R. Cipolla, IEEE Trans. Pattern Anal. Mach. Intell. **39**, 2481 (2017).
17) L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, IEEE Trans. Pattern Anal. Mach. Intell. **40**, 834 (2018).
18) H. K. Cheng, J. Chung, Y.-W. Tai, and C.-K. Tang, Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, p. 8890.
19) K. He, G. Gkioxari, P. Dollár, and R. Girshick, Proc. IEEE Int. Conf. Comput. Vis., 2017, p. 2961.
20) D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, Proc. IEEE Int. Conf. Comput. Vis., 2019, p. 9157.
21) E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, and P. Luo, Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, p. 12193.
22) M. Everingham et al., J. Comput. Vis. **88**, 303 (2010).
23) (http://host.robots.ox.ac.uk/pascal/VOC/).
24) A. Kaehler and G. Bradski, *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library* (O'Reilly Media, Sebastopol, 2017) 1st ed.
25) (https://github.com/wkentaro/labelme).
26) S. Tokui, K. Oono, S. Hido, and J. Clayton, Proc. Workshop on Machine Learning Systems (LearningSys) in the 29th Ann. Conf. Neural Info. Process. Syst., 2015.
27) (https://github.com/chainer/chainercv).
28) Y. Ouali, C. Hudelot, and M. Tami, Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, p. 12674.
29) O. J. Hénaff, S. Koppula, J.-B. Alayrac, A. van den Oord, O. Vinyals, and J. Carreira, Proc. IEEE Int. Conf. Comput. Vis, 2021, p. 10086.
30) N. Araslanov and S. Roth, Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2021, p. 15384.
31) X. Zhou, J. Zhuo, and P. Krähenbühl, Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, p. 850.
32) Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, p. 7291.