

## Article

# Pill Detection Model for Medicine Inspection Based on Deep Learning

Hyuk-Ju Kwon <sup>1</sup>, Hwi-Gang Kim <sup>2</sup> and Sung-Hak Lee <sup>1,\*</sup>

<sup>1</sup> School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 702-701, Korea; olin1223@knu.ac.kr

<sup>2</sup> Medical IT Convergence Laboratory, Electronics and Telecommunications Research Institute Daegu-Gyeongbuk Research Center, Daegu 42994, Korea; hwigangkim@etri.re.kr

\* Correspondence: shak2@ee.knu.ac.kr; Tel.: +82-53-950-7216

**Abstract:** This paper proposes a deep learning algorithm that can improve pill identification performance using limited training data. In general, when individual pills are detected in multiple pill images, the algorithm uses multiple pill images from the learning stage. However, when there is an increase in the number of pill types to be identified, the pill combinations in an image increase exponentially. To detect individual pills in an image that contains multiple pills, we first propose an effective database expansion method for a single pill. Then, the expanded training data are used to improve the detection performance. Our proposed method shows higher performance improvement than the existing algorithms despite the limited imaging and data set size. Our proposed method will help minimize problems, such as loss of productivity and human error, which occur while inspecting dispensed pills.

**Keywords:** deep learning; Mask R-CNN; pill detection; data augmentation; object region; object class



**Citation:** Kwon, H.-J.; Kim, H.-G.; Lee, S.-H. Pill Detection Model for Medicine Inspection Based on Deep Learning. *Chemosensors* **2022**, *10*, 4. <https://doi.org/10.3390/chemosensors10010004>

Academic Editor: Eleonora Alfinito

Received: 30 November 2021

Accepted: 21 December 2021

Published: 24 December 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Drug prescription and inventory management are very important tasks for safe drug dispensing, while promptness and accuracy are also very essential. Approximately 1000 types of pills are handled in large hospitals. The pills used by patients are changed depending on the patient's degree of improvement. In many existing hospitals and pharmacies, the pharmacist manually sorts and packs the pills according to the prescription, which is a time-consuming process. In addition, simple repetitive tasks can cause fatigue leading to mistakes being made during pill sorting; such situations can lead to medical accidents.

In recent years, automated equipment, such as automated medication dispensing machines [1–3], have rapidly spread in pharmacies and hospitals where multiple dispensing tasks need to be performed, such as sorting and packaging pills. An automated medication dispensing machine is a device that sorts and packs drugs based on a prescription that is input from a computerized program. However, the automatic dispensing machine also requires a function to inspect the prepared product because there is a risk of erroneous formulation. A vision inspection method using a digital camera is a widely used. The vision inspection method uses two forms of analysis. First, there is a rule-based analysis method that compares and analyzes product characteristics [4]. The second method for analysis involves a template that compares a similarity with a reference image [5–7]. Recently, deep learning-based object detection algorithms have been developed and investigated [8–11].

The template matching method is a method for finding a region with the highest similarity to a reference image in an input image. The methods used for comparing the input image with the reference image are divided into two categories: pixel-based and shape-based matching methods. The pixel-based matching method calculates the difference between the pixels of the reference image and the input image. Its representative methods include the sum of squared difference and normalized cross correlation [12,13].

The pixel-based matching method is robust against distortions such as blurring caused by the shaking of the camera during capturing. However, the pixel-based method is not effective for changes in the size and rotation of the inspection object because it calculates the difference between the pixels.

The shape-based matching method is a method for extracting a region of interest (ROI) of a test subject from a reference image and comparing it with an input image. The shape-based matching method does not use all the pixels of the reference image. Rather, it uses only the representative features and is effective in changing the size and rotation of an object. The shape-based matching method is superior to the pixel-based matching method in relation to lighting changes. Its representative methods include scale-invariant feature transform and shape-based matching included in the MVTec HALCON library [14–16].

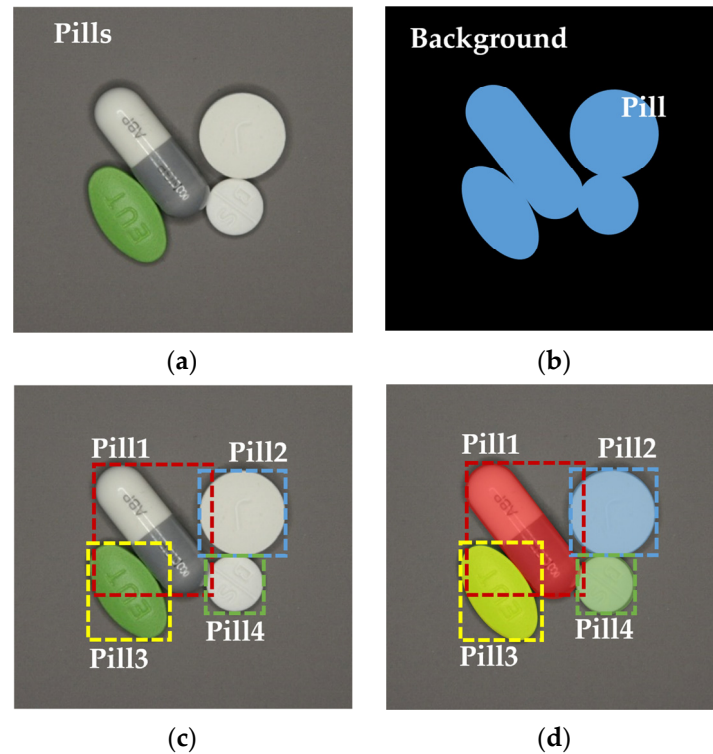
Deep learning technology is an artificial neural network technology that can learn and make judgments on its own based on data. This technology shows excellent performance in the field of object detection. Object detection refers to a method that specifies not only the presence of an object in an image but also the type and location of the object. Representative deep learning methods include you only look once (YOLO) and region-based convolutional neural network (R-CNN) [17,18]. To improve the detection performance in deep learning, scientists use various methods, such as data processing, loss function improvement, convolution layer control, and activation functions [19–23]. Instead of using the structure of a single convolutional neural network (CNN), a complex structure involving color, shape, and difference images is used [24]. However, the deep learning method requires a large amount of data for training despite its excellent detection performance. Substantial effort is required to create new data that are not public. Therefore, it is important to investigate how to effectively extend a small amount of data.

In this paper, we propose a regional deep learning algorithm that can improve detection performance by using limited training data when learning for pill detection. The proposed method aims to detect the location and type of individual pills in an image containing multiple pills. In general, when an individual pill is detected in an image that includes multiple pills, that image is also used in the learning stage. However, in the case of dispensed drugs, the number of cases that can be combined in one image increases exponentially as the types of detection targets increase. To solve this problem, the proposed method limits the data by capturing an image of only a single pill when generating the training data. To improve local detection performance, a two-step detection method based on Mask R-CNN is used. In the first step, we aim to detect only the number and area of the pills included in the image, regardless of the type of pill. In the second step, after separating the pill detected in the first step from the background, the type of the corresponding pill is detected. The training data consisting of a single pill can be used for the second-step learning because the pill was separated from the background. Consequently, even if the types of pills increase, it is easy to acquire training data because only the data on individual pills need to be considered. Finally, post-processing algorithms, such as image rotation, were applied to further improve performance in detecting the second-stage pill. The proposed pill learning and detection algorithm demonstrated higher detection performance improvement than the existing algorithms despite the limited imaging and data set size. This is expected to improve the performance of the automated devices (e.g., automated medication dispensing machine) and to minimize problems (e.g., loss of productivity and human errors).

## 2. Mask R-CNN

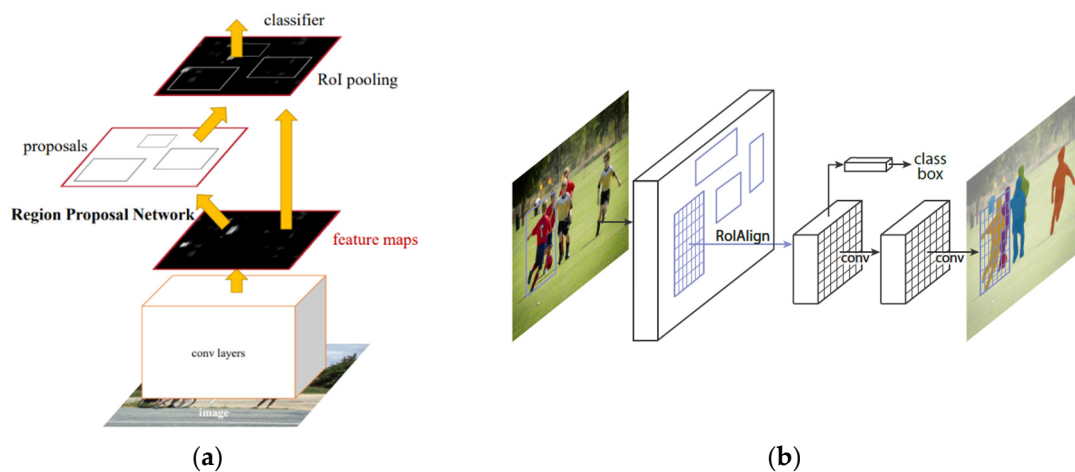
Mask R-CNN is an extended model of Faster R-CNN, which is an instance segmentation algorithm that simultaneously predicts the bounding box that informs the existing object location and the mask of the object area [25,26]. Figure 1 shows an example of image segmentation and object detection. Figure 1a is an input image and consists of four types of pills and a background. Figure 1b shows the result of a semantic segmentation. The picture is divided into two areas, the background and the pill. All pills belong to a single entity [27].

Figure 1c shows the result of an object detection, and it can be seen that the location of each pill is detected by the bounding box. Figure 1d shows the result of instance segmentation. Instance segmentation is a combination of semantic segmentation and object detection, and unlike the semantic segmentation, each pill is an individual object as a separate entity.



**Figure 1.** Examples of image segmentation and object detection: (a) Input image; (b) Semantic segmentation; (c) Object detection; (d) Instance segmentation.

Figure 2 shows the structures of Faster R-CNN and Mask R-CNN. Faster R-CNN consists of (i) a region proposal network that predicts the bounding boxes, (ii) the RoIPool that extracts the feature maps inside the bounding boxes, and (iii) multiclass classification and regression learning of the bounding boxes. Mask R-CNN is based on the structure of Faster R-CNN, and binary mask learning is performed for each ROI at the same time as the class classification and bounding box regression learning.

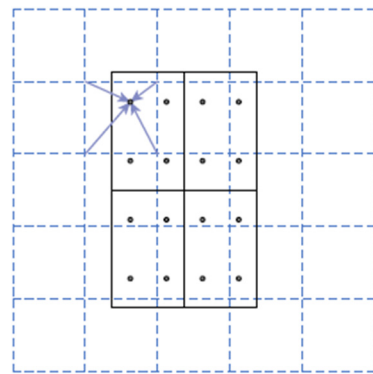


**Figure 2.** Structures of (a) Faster region-based convolutional neural network (R-CNN) and (b) Mask R-CNN.

Mask R-CNN's RoIAlign is an algorithm for resolving the discrepancy between the ROI and feature map positions that occur in the Faster R-CNN's RoIPool. Figure 3 shows the structure of RoIAlign. The loss calculated for each ROI during training is as follows:

$$L = L_{cls} + L_{box} + L_{mask} \quad (1)$$

where  $L_{cls}$  is the classifying loss,  $L_{box}$  is the bounding box loss, and  $L_{mask}$  is the average binary cross-entropy loss of a binary mask.

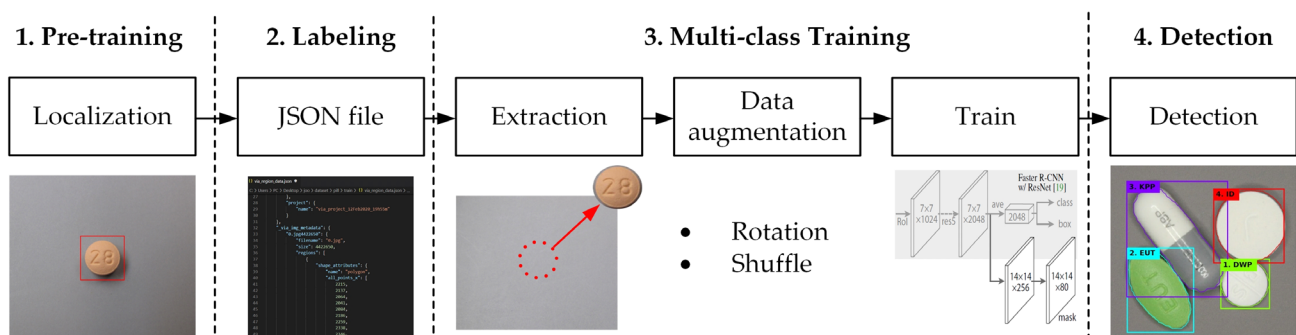


**Figure 3.** Structure of RoIAlign. The dashed grid represents the location of the feature map, and the solid grid represents the location of the quantized ROI. The dots inside the solid grid indicate the sample positions. RoIAlign computes the value of each sampling point by bilinear interpolation from the nearby grid points on the feature map.

### 3. Mask R-CNN–Based Pill Inspection Model

In general, to detect individual pills in multiple pill images, the training data images need to include multiple pills. In addition, the location and class of pills need to be defined for each image. However, as the types of pills increase, the number of possible combinations increases rapidly; therefore, the capturing of the training data and the labeling of the pill class in each image become complicated. Therefore, we propose a method for effectively detecting individual pills in an image that includes multiple pills. The individual pills will be detected by learning an image containing only one pill for each pill class.

Figure 4 shows the progress of the proposed pill learning and detection method. The proposed method consists of four steps. The first step is preprocessing learning, which is a single class of pill area learning for detecting the area of a pill. The second step is the data labeling process. The third step is multi-class pill detection learning to determine the types of pills detected in the first step. The fourth step is the pill detection process. The learning of pill detection is divided into two steps. In Step 1, an image of multiple pills was used as the training data for detecting the pill area. In Step 2, an image of a single pill was used as the multi-class training data.

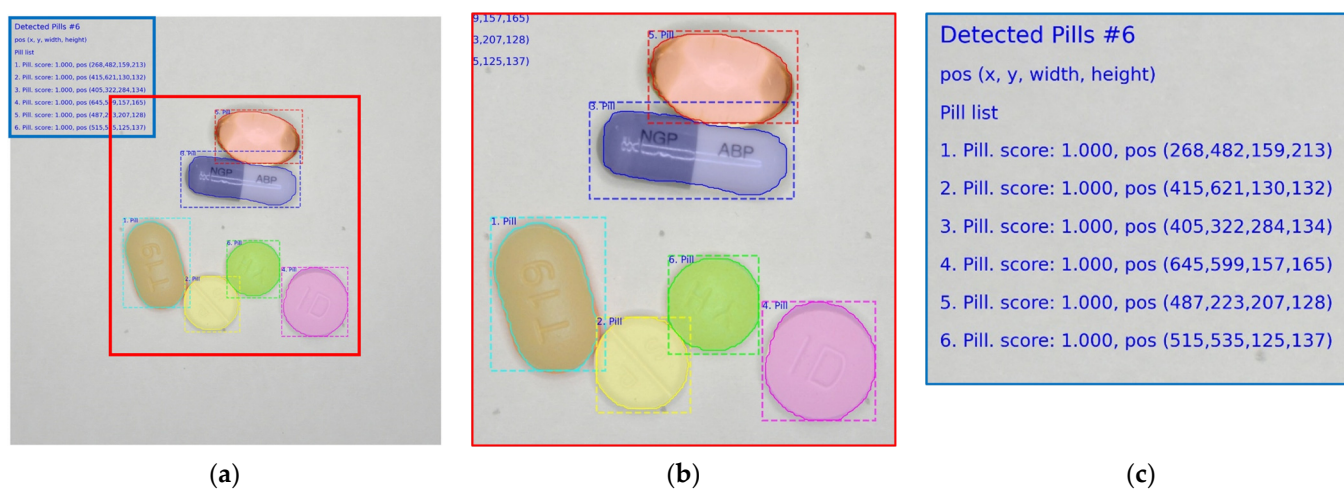


**Figure 4.** Process of the proposed method.

### 3.1. Single-Class—Based Pill Area Detection Learning

Single-class pill area detection learning is conducted to accurately detect the location of pills in an image. This learning model is used to separate pills from the background during label automation and detection of multiple classes of pills. To accurately detect the various positions of the pills in the image, we used an image containing multiple pills and a binary mask image corresponding to each image. The class was matched as one class (“Pill”) regardless of the type of pill.

Figure 5 shows the resultant image. The area of the pill is indicated in units of pixels regardless of the color and shape of the pill. Training for pill area detection is performed to detect the location and area of individual pills when detecting pills. Detection is performed once as pre-training for pill detection.

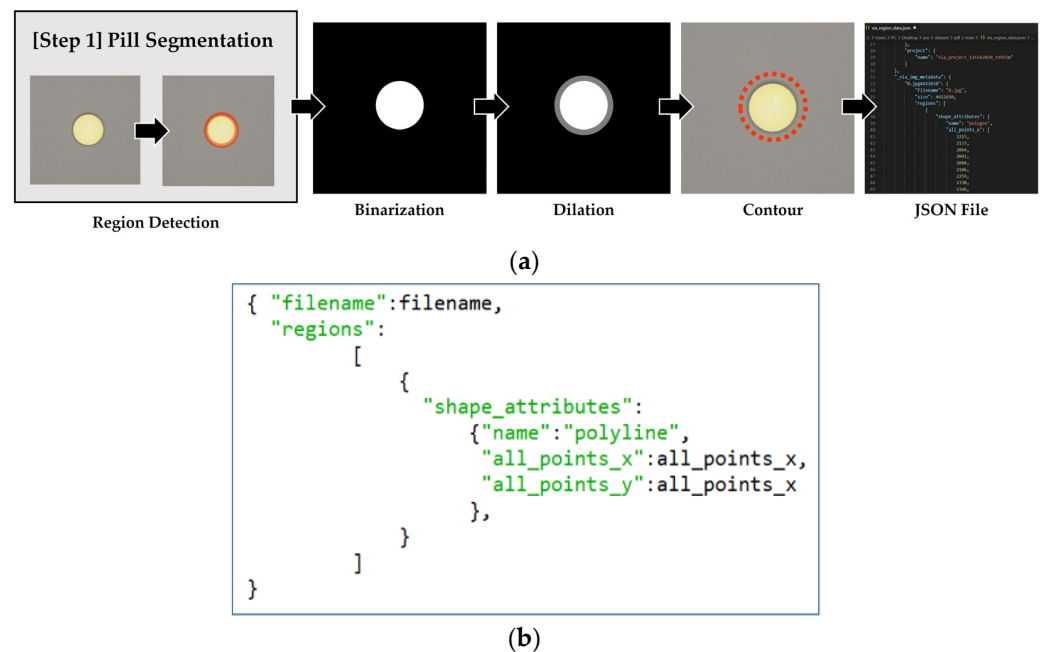


**Figure 5.** Result of the pill area detection: (a) Detection result image; (b) Cropped image of instance segmentation. Outer rectangle is a bounding box and inner solid line indicates a detected pill area; (c) Cropped image of detection information consisting of the number of pill, detection scores, and bounding box positions.

### 3.2. Data Labeling and Automatic Generation of JSON Files

Data labeling refers to the work of classifying and transforming data using data processing tools to train a deep learning model. For image-based object detection, the training image and the position coordinates of the object corresponding to each image are required. Mask R-CNN requires polygonal coordinates that represent the shape of the object and the position coordinates of the object. To create this polygonal coordinate, it is necessary to use the video annotation tool for displaying the polygonal coordinates and class names for each object in the image. However, the use of these tools require considerable time and effort. To reduce losses, we need a way to automate data labeling. For automation, we propose a method for detecting the area of a pill using the single-class pill area learning model given in Section 3.1, and we change the detected area into polygonal coordinates. We also propose a method for automatically converting the coordinates and image information into a JSON file.

Figure 6 shows the proposed data labeling and JSON file generation process. The data stored in the JSON file include the file name of each image and the polygonal coordinates of the pill area.



**Figure 6.** Process of data labeling and JavaScript Object Notation file creation: (a) Process of data labeling; (b) Structure of JavaScript Object Notation.

The process of automation consists of the following steps: region detection, binarization, region dilation, contour extraction, and JavaScript Object Notation (JSON) file generation. The region detection process selects the region of the pill using a one-step learning model, and the binary step is used to expand the detected region and extract the contour. The dilation process of the pill area was used to improve the detection performance by including more edge information (shade regions) of the pill during training. To match the center of the expanded area with the center of the actual pill when the area is enlarged, the image was enlarged based on the centroid of the detected pill area using the following formula:

$$D(x, y) = \begin{bmatrix} s_x & 0 & -x_c \\ 0 & s_y & -y_c \\ 0 & 0 & 1 \end{bmatrix} T(x, y) + \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix}, \quad (2)$$

$$x_c = \frac{|x|T(x, y) = 255|}{N}, \quad (3)$$

$$y_c = \frac{|y|T(x, y) = 255|}{N}, \quad (4)$$

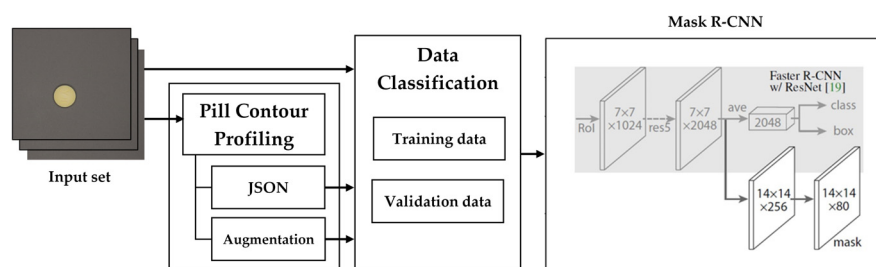
where  $D(x, y)$  denotes an enlarged image, and the size of the image is the same as the size of  $T(x, y)$ , which is the input image. The parameters  $s_x$  and  $s_y$  denote the ratios applied to the dilation; the dilation ratio was set to 1.2 for both  $s_x$  and  $s_y$ . The parameters  $x_c$  and  $y_c$  denote the center of gravity of the pill area.  $N$  indicates the number of pixels satisfying  $T(x, y) = 255$ .

The next step is to extract the contour of the enlarged area and then convert it into polygonal coordinates. To extract the polygonal coordinates from the binary image, we used the `findContours` function of an OpenCV library. Finally, we saved the data labels of each image as a JSON file. In this step, the data capacity was reduced by converting the training data into a JSON file and storing this file. The image and the location information of the pill existing in the image were read efficiently during the learning process.

### 3.3. Multi-Class—Based Pill Label Detection Learning

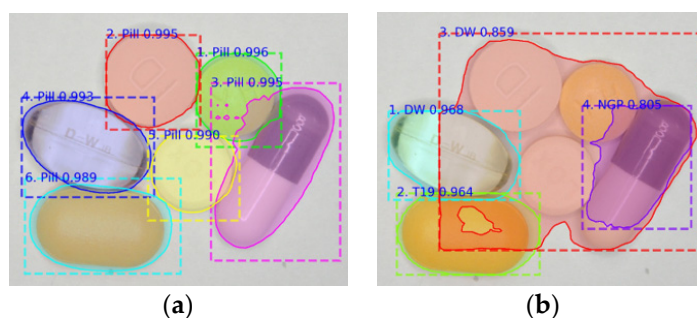
A model for pill label detection requires a model specialized in classification and detection. Mask R-CNN is a successor model to Faster R-CNN and has the best performance among the pill detection models analyzed in [28]. Mask R-CNN has an instance segmentation function that can express the area of the detected object in pixels. The proposed learning model uses a training image in which only one pill exists per image, and a JSON file with polygonal coordinates of the pill area is used as the input data. Data for the pill area are obtained using the pill region detection model given in Section 3.1. The obtained data are converted into a JSON file using data labeling and the JSON file automatic generation algorithm described in Section 3.2.

In addition, exposure and rotation augmentation were performed to supplement the insufficient data during training. For data augmentation, “imgaug” of a python library [29] was used, and the image was rotated at an arbitrary angle between  $-180^\circ$  and  $+180^\circ$  during training. Finally, multi-class learning using individual pill images was performed. Figure 7 shows the training process for multi-class pill detection.

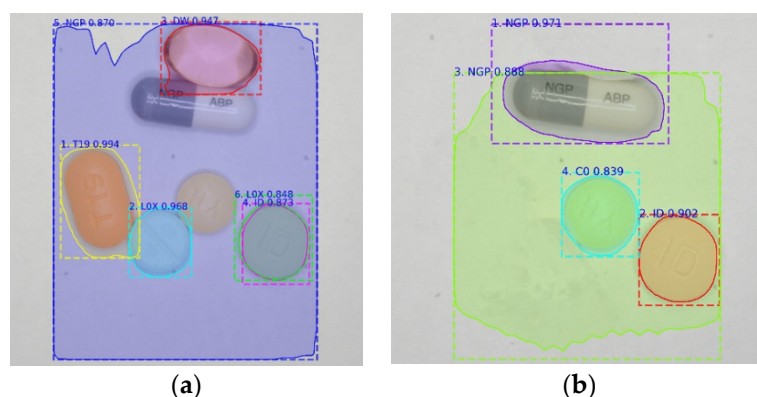


**Figure 7.** Training process of pill detection using mask region-based convolutional neural network: Pill contour profiling makes labeling data using automatic data labeling algorithm and rotated images. The rotated images are used for validation data. In the data classification step, the image data are classified into training and validation data set.

Figures 8 and 9 show the need for the proposed two-step model. Figure 8 shows the results of learning an image in which only one pill exists as the training data. It also shows the result of detecting a pill in an image that includes multiple pills using this model. Figure 8a is the result of setting only one class (“Pill”) during learning regardless of the type of pill, and Figure 8b is the result of learning with multiple classes using the same training data. In Figure 8a, although the image was learned with only one pill, it is composed of one class; therefore, it seems that the individual pill area can be detected even in the image that contains multiple pills. However, in multi-class learning in Figure 8b, we can see that many miss-detection, over-detection, and non-detection phenomena appear during testing.



**Figure 8.** Detection results of multi-pill image using a single pill trained model: (a) Result of a model trained with single class; (b) Result of a model trained with multiple classes.



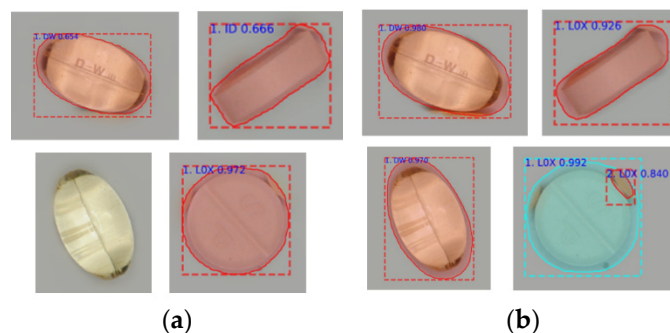
**Figure 9.** Results with and without adjacent pills: (a) Image in which the pills were undetected due to adjacent pills; (b) Image in which the undetected pills were detected after the adjacent pills were removed.

Figure 9 shows the test results for the effect of adjacent pills. Figure 9a clearly shows that non-detection was observed for two pills and over-detection was observed the other pills. At this time, to check the effect on the adjacent pills, we removed the three pills DW, LOX, and T19, as shown in Figure 9a, and performed the test again. Figure 9b shows the results. Consequently, the two tablets CO and NGP, which were not detected earlier (see Figure 9a), were detected. Therefore, to effectively adopt the model learned by using the image in which only one pill exists, it is necessary to create an image in which only one pill exists by removing the surrounding pills during detection. For this purpose, we used a two-step method of single-class pill area detection and multi-class pill detection.

### 3.4. Pill Detection Process

#### 3.4.1. Optimization of Area Dilation in Detecting Multi-Class Pills

This part is foreground-background segregation. When the pill was separated from the background, the detected area was enlarged by a certain percentage to include more edge information of the pill. Figure 10 shows the resulting image according to the area dilation ratio. Figure 10a,b show the detection results in the multi-class pill detection model after dilation of each detected area by 10% and 20%. In Figure 10a, a few pills are smaller than the area used for learning. Therefore, non-detection occurred. In the bottom right image of Figure 10b, when the area is enlarged at the rate of 20%, a portion of the adjacent pill is included, which results in false detection. An additional post-processing algorithm is required to solve this problem.



**Figure 10.** Resultant images according to the dilation ratio in the second-stage detection: (a) Results of the 10% area dilation ratio; (b) Results of 20% area dilation ratio.

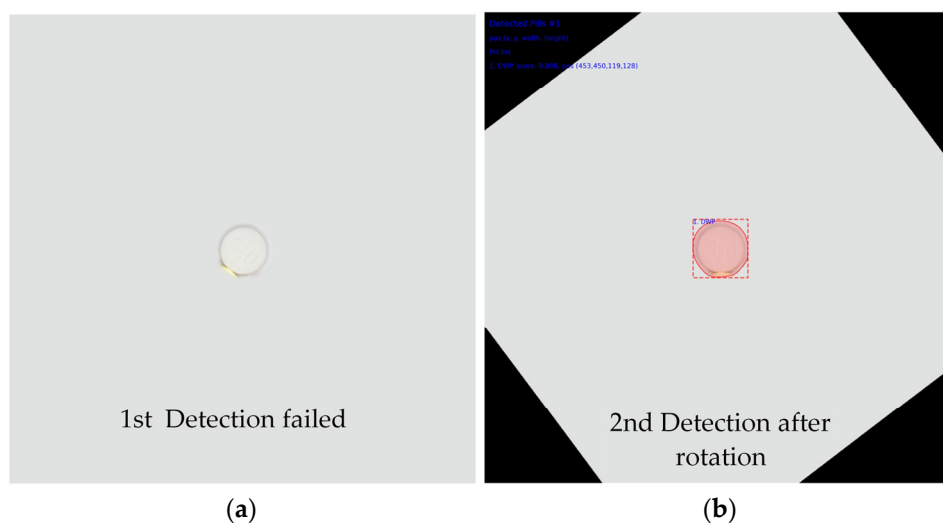
#### 3.4.2. Post-Processing Algorithm to Enhance the Multi-Class Detection Performance

When training the multi-class pill detection model with rotation augmentation, detection was not performed for some angles. To solve this problem, we added a method for



repeating detection by rotating the input image when the pill was not detected. Figure 11 shows the detection results before and after rotation of the input image. Figure 11a shows an image of a pill separated from the background after detecting the pill area, and it shows the result of a failure to detect during pill classification. Figure 11b shows the result of multi-class detection after rotating the image of Figure 11a by  $45^\circ$ . Unlike Figure 11a, we can see that the pill was accurately detected. If multiple results appear in Figure 10b, only the result with the largest area was selected. The post-processing steps to improve the multi-class detection performance are as follows:

- (1) After pill area detection, 20% dilation is performed.
- (2) When multiple pills are detected, the pill with the largest area is selected.
- (3) When the pill is not detected, stepwise rotation detection is performed from  $1^\circ$  to  $45^\circ$ .



**Figure 11.** Resultant images according to rotation in second-stage detection: (a) Image in which the pill (“DWP”) was not detected before rotation; (b) Image in which an undetected pill (“DWP”) was detected after rotation.

## 4. Results

### 4.1. Pill Area Detection Experiment

The first stage is a model for detecting the number of pills and the area of the pills in the image regardless of the type of pills. Therefore, when learning, the parameter is set as a single “Pill” class without class classification according to the type of pill.

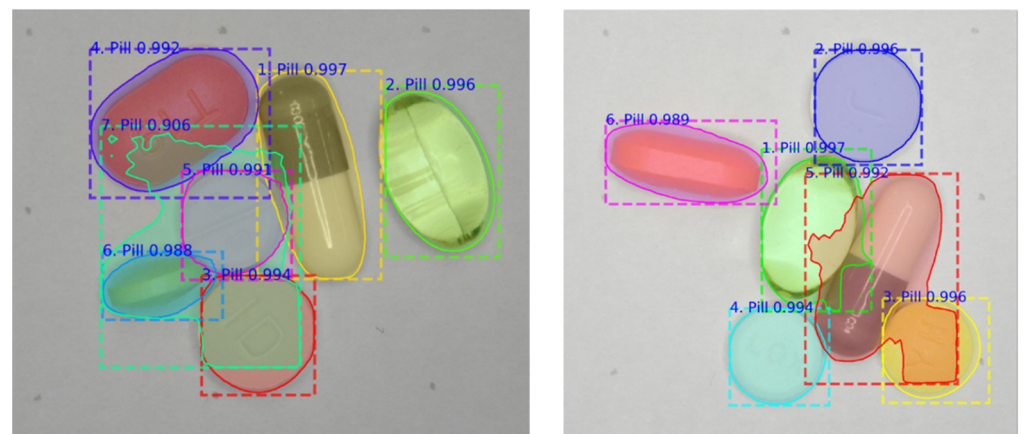
Figures 12 and 13 are experiments to improve the performance of the first-stage pill area detection. Figure 12 shows an image in which only one pill was used as the training data during the first-step model training. In Figure 12, we used six kinds of pills for the training data. Each image consists of a pill that is placed in various directions and positions. Each image was taken as two images with different exposures using the bracketing function of the camera, and 600 images were used for learning. Data augmentation was not used during training, and the epoch was fixed at 100. Figure 12a shows the training data set, and Figure 12b shows the result of area detection. Figure 12b shows over-detection, that is, detecting a larger area than the area of some pills. To solve this over-detection problem, the pill area detection model was retrained using images containing multiple pills (see Figure 13).

Figure 13a shows a part of the image used for training. The image includes six kinds of pills; each image contains five to six pills, and the location of the pills is variously arranged. Each image was taken as two images with different exposures using the bracketing function of the camera, and 50 images were used for training. The polygonal coordinates were manually indicated for each image. Data augmentation was not used during training, and the epoch was fixed at 20. Figure 13b is the detection result of the pill area. In the

image on the left of Figure 13b, all the pill positions are accurately displayed. In the image on the right of Figure 13b, the pill area is accurately displayed without over-detection or non-detection for pills not used for other backgrounds and for learning.

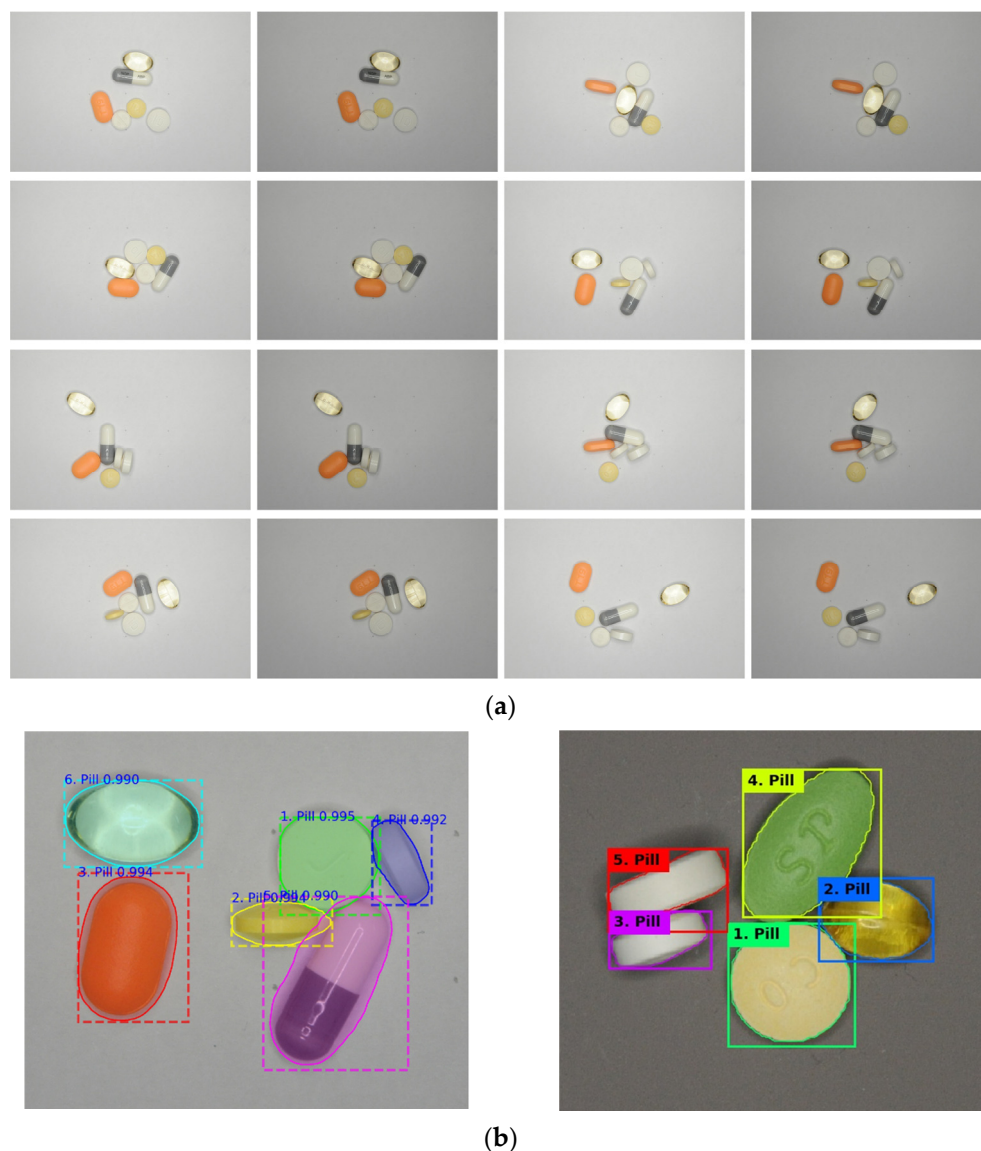


(a)



(b)

**Figure 12.** Detection results using single pill image training data set: (a) Training data set consisting of single pill images; (b) Detection results of multiple pills.



**Figure 13.** Detection results using image training data set for multiple pills: (a) Training data set consisting of multiple pill images; (b) Detection results of multiple pills.

#### 4.2. Pill Label Detection Experiment

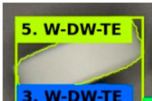
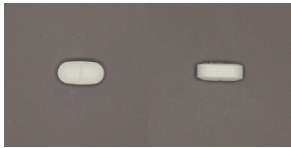
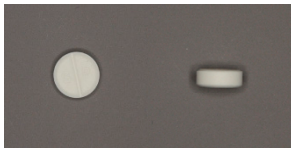
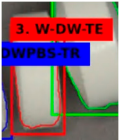
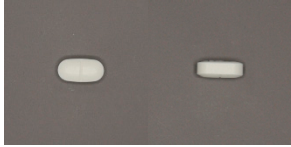
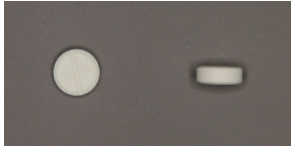
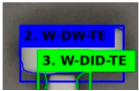
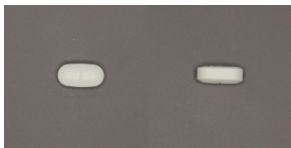

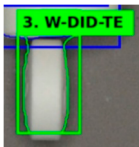
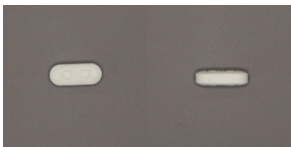

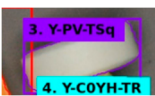
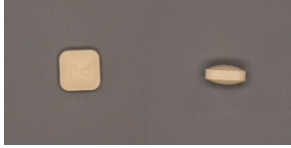
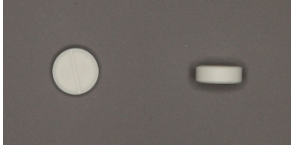

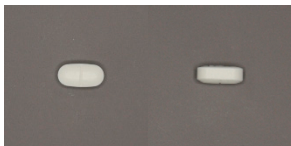

For two-step multi-class pill detection, the training data for 27 kinds of pills (including 10 kinds of white pills) were generated. The training data consisted of a single pill image and polygonal coordinates that represented the pill area of each image. When shooting a learning video, the shutter speed of the camera was adjusted to shoot images simultaneously with different amounts of exposure, and the position of the pill was changed for each shot. The position of the pill was physically changed, such as left–right inversion,  $45^\circ$  unit rotation, and eight-direction movement along the up, down, left, and right diagonals. To compensate insufficient data during learning, the images were rotated at an arbitrary angle ranging from  $-180^\circ$  to  $+180^\circ$ , while 30% of the total data were used for validation.

The two-step learning model is based on Mask R-CNN [30]. The input image size was  $1024 \times 1024$ , and the color space was RGB. The batch size was 4, and the learning rate was 0.01. Backbone ResNet50 was used as the network. We also used Python 3.6, Tensorflow 1.14, and Keras 2.1.3 frameworks on the Windows 10 operating system.

Unlike the case for chromatic pills, the number of false positives for white pills increases as the class increases. Table 1 shows false detection result. In the case of white

pills photographed in the lateral direction, it is difficult to distinguish between the round and oval shapes even with the naked eye. Overfitting occurred due to relatively insufficient data. Therefore, to improve the detection performance of white pills, additional images were taken in the lateral direction when photographing white pills. In addition, the shooting direction is in four diagonal directions and a 45° angle, and the number of side training data doubled as compared with the previous case. By changing the shooting background to gray (N5) with a reflectance of 50%, the boundary between the white pills was well distinguished. Table 2 shows the final capturing conditions as well as training and test settings.

**Table 1.** False detection results of the sideways pill images.

Failure Case	Failure Information		Ground Truth	
	Class	Registered Image	Class	Registered Image
	W-DW-TE		W-L0XSP-TR	
	W-DW-TE		W-L0XSP-TR	
	W-DW-TE		W-L0XSP-TR	
	W-DID-TE		W-DWPBS-TR	
	Y-PV-TSq		W-L0XSP-TR	
	W-DW-TE		W-DWPBS-TR	

Figures 14 and 15 show the performance of the proposed two-step learning model. Figure 14 shows the results of not using post-processing when detecting multi-class pills, and Figure 15 shows the results of applying post-processing. Table 3 shows the numerical results of Figures 14 and 15. The precision and accuracy values were improved by 10–16% in the learning range of less than 500 epochs depending on the area expansion and detection improvement post-processing. The post-processing model shows the best performance at 300 epochs. Accuracy is lower than precision at 60 epochs because FN cases occur, and FN cases do not happen above 100 epochs, so accuracy and precision have the same score. In Table 3, we used two evaluation metrics, namely precision and accuracy, to compare the performance according to the epoch.

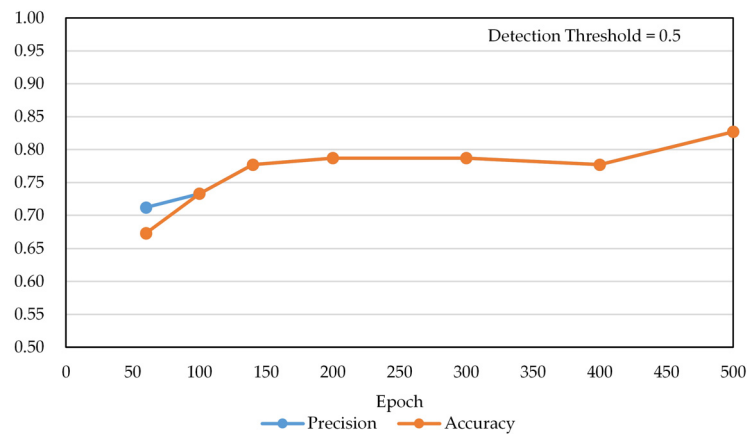
**Table 2.** Capturing conditions and training and test settings for simulations.

Environment	Option	Description
Capturing conditions	Count	728 (27 pills, 20–40 captures per pill)
	Background	Gray (N5, 50% reflectivity)
	Position	[Set 1] center, 4-shift, rotation (0°, 90°), 2-exposure [Set 2] Set 1 positions, 8-shift, rotation (0°, 45°, 90°) for white pills
Training setting	Batch	4
	Step size	182
	Learning rate	0.01
	Transfer learning	ResNet50
	Training set	728
	Validation set	226 (30% random selections per class and rotation)
	Augmentation	From −180° to +180° random rotation per epoch
Test setting	Test set	50
	Pill count	202
	Threshold	0.5

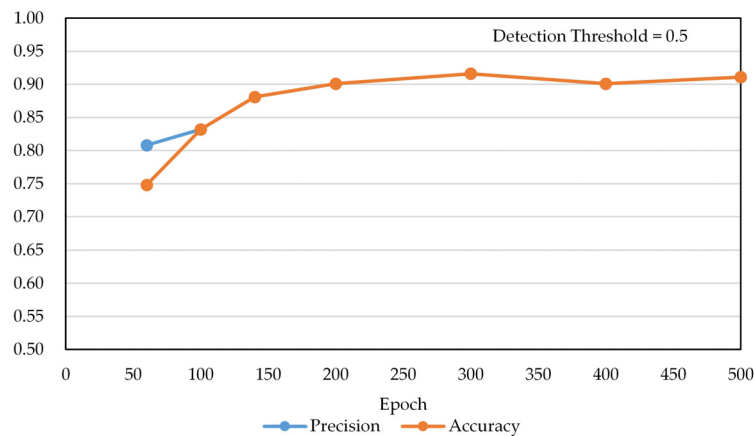
$$\text{Precision} = \frac{TP}{TP + FP'} \tag{5}$$

$$\text{Accuracy} = \frac{TP}{TP + FP + FN'} \tag{6}$$

where TP, FP, and FN symbolize the true positive, false positive, and false negative, respectively.



**Figure 14.** Test results of the two-step detection of 27 pills (without post-processing, 728 training images).



**Figure 15.** Test results of the two-step detection of 27 pills (with post-processing, 728 training images).

**Table 3.** Comparison of the detection performance with and without post-processing to improve pill detection.

Epoch	Without Post-Processing		With Post-Processing	
	Precision	Accuracy	Precision	Accuracy
60	0.712	0.673	0.808	0.748
100	0.733	0.733	0.832	0.832
140	0.777	0.777	0.881	0.881
200	0.787	0.787	0.901	0.901
300	0.787	0.787	0.916	0.916
400	0.777	0.777	0.901	0.901
500	0.827	0.827	0.911	0.911

Tables 4 and 5 show the detection results for the front and sideways position of the pills with and without post-processing. Overall, we can see that the detection rate for the frontal image of a pill is higher than that for the sideways image. The results of Table 5 with post-processing show better performances than the results of Table 4 without post-processing. Finally, Table 6 shows the detection rate of the sideways image according to the conversion of the capturing conditions 1 and 2 (Set 1 and Set 2). By additionally taking data for the side of the white pill, we confirmed that the detection performance of Set 2 was higher than that of Set 1. The detection ratio is represented by  $OK / (OK + NG)$ . Table 7 represents the computational time of the proposed method without and with post-processing. The number of images used for the test is 50, and the number of pills is 202. The computational speed of 202 pills is 98.70 s without post-processing and 108.11 s with post-processing. The average per pill is 0.49 s without post-processing and 0.54 s with post-processing.

**Table 4.** Detection results for 27 pills (without post-processing).

Epoch	Front			Side (Set 2 Condition)		
	OK	NG	Detection Ratio	OK	NG	Detection Ratio
60	109	47	0.699	27	19	0.587
100	119	37	0.763	29	17	0.630
140	127	29	0.814	30	16	0.652
200	130	26	0.833	29	17	0.630
300	131	25	0.840	28	18	0.609
400	130	26	0.833	27	19	0.587
500	137	19	0.878	30	16	0.652

**Table 5.** Detection results for 27 pills (with post-processing).

Epoch	Front			Side (Set 2 Condition)		
	OK	NG	Detection Ratio	OK	NG	Detection Ratio
60	122	34	0.782	29	17	0.630
100	138	18	0.885	30	16	0.652
140	147	9	0.942	31	15	0.674
200	149	7	0.955	33	13	0.717
300	152	4	0.974	33	13	0.717
400	146	10	0.936	36	10	0.783
500	149	7	0.955	35	11	0.761

**Table 6.** Side pill detection results for 27 pills.

Epoch	Side (Set 1 Condition)			Side (Set 2 Condition)		
	OK	NG	Detection Ratio	OK	NG	Detection Ratio
60	30	16	0.652	29	17	0.630
100	31	15	0.674	30	16	0.652
140	26	20	0.565	31	15	0.674
200	25	21	0.543	33	13	0.717
300	23	23	0.500	33	13	0.717
400	23	23	0.500	36	10	0.783

**Table 7.** Computational time of the proposed method without and with post-processing.

	Without Post-Processing	With Post-Processing
202 pills	98.70 s	108.11 s
Average per pill	0.49 s	0.54 s

We compared the results of the proposed method with YOLOv3 [31]. The training data of YOLOv3 consist of single pill images, and multi pill images are used for testing without proposed foreground-background segregation and post-processing. The images used for training and testing of YOLOv3 are the same as those of the proposed method. The input image size is  $416 \times 416$ , and the color space is RGB. The batch size is 64, and the learning rate is 0.001. Other parameters and data augmentation options use default values. The training is set to 10,000 iterations and this iteration can be calculated with approximately 880 epochs. Table 8 shows the results of YOLOv3 and the proposed method. The precision of YOLOv3 is higher than that of the proposed method, but the accuracy of YOLOv3 is lower than that of the proposed method because the FN of the proposed method is 0. Therefore, the proposed method is better than YOLOv3 in terms of detection performance.

**Table 8.** Comparison results for each method.

Method	TP	FP	FN	Precision	Accuracy
YOLOv3	148	12	42	0.925	0.733
Proposed method	187	15	0	0.916	0.916

## 5. Discussion

The detection of the pill area was used to confirm the number and location of pills in the image to be examined using a learning model that consisted of a single class. For effective learning in a single class, the training data consist of images with different exposure levels obtained from images composed of multiple random pills. The results show that the positions and numbers of various types of pills were accurately detected.

The second-stage learning model classifies the class of pills detected by the first-stage learning. Unlike the first-stage learning model, the second-stage training data used an image composed of one pill for each shooting. This can minimize the number of data sets that increased in proportion to the number of classes because the number of classes increased when shooting with multiple pill combinations.

In a classification detection experiment using an actual image of multiple pills, it was confirmed that a single pill extraction and detection post-processing algorithm can solve a number of non-detection, erroneous detection, and over-detection phenomena that occurred because of differences between the multiple pill data and the training data captured with a single pill.

## 6. Conclusions

In this paper, we proposed a deep learning algorithm that can improve the detection performance based on limited training data and an effective database expansion method for

the additional identification of pills. The proposed algorithm aims to detect individual pills among multiple pills. To minimize the data required for learning, an image including only one pill (not multiple pills) was taken when generating the training data. For pill detection, we proposed a model with a two-step structure for the pill area detection and multi-class pill detection. Moreover, we added single pill extraction and detection post-processing to improve the detection ratio. This study proposes a limited pill identification method that can be applied to various object detection techniques in environmental conditions that lack training data in various fields.

However, there are fixed limitations in the experimental environment for acquiring learning and test images, and it is necessary to experiment in various environments using easy-to-use shooting tools such as mobile phones. Moreover, in addition to Mask R-CNN, it is necessary to conduct an experiment by applying a recent transformer-based technique, such as MaskFormer and Trans4Trans, that can simplify the mask classification task [32,33].

**Author Contributions:** Conceptualization, S.-H.L.; methodology, S.-H.L. and H.-J.K.; software, H.-J.K.; validation, S.-H.L., H.-J.K. and H.-G.K.; formal analysis, S.-H.L. and H.-J.K.; investigation, S.-H.L. and H.-J.K.; resources, S.-H.L. and H.-J.K.; data curation, S.-H.L., H.-J.K. and H.-G.K.; writing—original draft preparation, H.-J.K.; writing—review and editing, S.-H.L.; visualization, H.-J.K.; supervision, S.-H.L.; project administration, S.-H.L.; funding acquisition, S.-H.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) and the BK21 FOUR project funded by the Ministry of Education, Korea (NRF-2021R1I1A3049604, 4199990113966) and Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [21ZD1140, Development of ICT Convergence Technology for Daegu-Gyeongbuk Regional Industry].

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare that there is no conflict of interest regarding the publication of this paper.

## References

1. Gordon, J.O.; Hadsall, R.S.; Schommer, J.C. Automated medication-dispensing system in two hospital emergency departments. *Am. J. Health Pharm.* **2005**, *62*, 1917–1923. [CrossRef]
2. Fung, E.Y.; Leung, B.; Hamilton, D.; Hope, J. Do Automated Dispensing Machines Improve Patient Safety? *Can. J. Hosp. Pharm.* **2009**, *62*, 516–519. [CrossRef] [PubMed]
3. Craswell, A.; Bennett, K.; Hanson, J.; Dalglish, B.; Wallis, M. Implementation of distributed automated medication dispensing units in a new hospital: Nursing and pharmacy experience. *J. Clin. Nurs.* **2021**, *30*, 2863–2872. [CrossRef] [PubMed]
4. Wen, Z.; Tao, Y. Building a rule-based machine-vision system for defect inspection on apple sorting and packing lines. *Expert Syst. Appl.* **1999**, *16*, 307–313. [CrossRef]
5. Chantara, W.; Mun, J.-H.; Shin, D.-W.; Ho, Y.-S. Object Tracking using Adaptive Template Matching. *IEIE Trans. Smart Process. Comput.* **2015**, *4*, 1–9. [CrossRef]
6. Zhou, X.; Wang, Y.; Xiao, C.; Zhu, Q.; Lu, X.; Zhang, H.; Ge, J.; Zhao, H. Automated Visual Inspection of Glass Bottle Bottom With Saliency Detection and Template Matching. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 4253–4267. [CrossRef]
7. Chen, J.-Y.; Hung, K.-F.; Lin, H.-Y.; Chang, Y.-C.; Hwang, Y.-T.; Yu, C.-K.; Hong, C.-R.; Wu, C.-C.; Chang, Y.-J. Real-time FPGA-based template matching module for visual inspection application. In Proceedings of the 2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Kaohsiung, Taiwan, 11–14 July 2012; pp. 1072–1076.
8. Wong, Y.F.; Ng, H.T.; Leung, K.Y.; Chan, K.Y.; Chan, S.Y.; Loy, C.C. Development of fine-grained pill identification algorithm using deep convolutional network. *J. Biomed. Inform.* **2017**, *74*, 130–136. [CrossRef] [PubMed]
9. Chupawa, P.; Kanjanawanishkul, K. Pill Identification with Imprints Using a Neural Network. *Maharakham Int. J. Eng. Technol.* **2015**, *1*, 30–35.
10. Wang, Y.; Ribera, J.; Liu, C.; Yarlagadda, S.; Zhu, F. Pill Recognition Using Minimal Labeled Data. In Proceedings of the 2017 IEEE Third International Conference on Multimedia Big Data (BigMM), Laguna Hills, CA, USA, 19–21 April 2017; pp. 346–353.
11. Larios Delgado, N.; Usuyama, N.; Hall, A.K.; Hazen, R.J.; Ma, M.; Sahu, S.; Lundin, J. Fast and accurate medication identification. *Npj Digit. Med.* **2019**, *2*, 10. [CrossRef] [PubMed]



12. Briechle, K.; Hanebeck, U.D. Template matching using fast normalized cross correlation. In Proceedings of the Optical Pattern Recognition XII, Orlando, FL, USA, 16–20 April 2001; Volume 4387, pp. 95–102.
13. Hisham, M.B.; Yaakob, S.N.; Raof, R.A.; Nazren, A.A.; Wafi, N.M. Template Matching using Sum of Squared Difference and Normalized Cross Correlation. In Proceedings of the 2015 IEEE Student Conference on Research and Development (SCoReD), Kuala Lumpur, Malaysia, 13–14 December 2015; pp. 100–104.
14. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
15. Xu, E.; Zhang, X.; Han, J.; Wu, C. HALCON application for shape-based matching. In Proceedings of the 2008 3rd IEEE Conference Industry Electronics and Applications, Singapore, 3–5 June 2008; pp. 2431–2434. [[CrossRef](#)]
16. Belongie, S.; Malik, J.; Puzicha, J. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 509–522. [[CrossRef](#)]
17. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A.; Impiombato, D.; Giarrusso, S.; Mineo, T.; Catalano, O.; Gargano, C.; La Rosa, G.; et al. You Only Look Once: Unified, Real-Time Object Detection. *Nucl. Instrum. Methods Phys. Res. Sect. A Accel. Spectrom. Detect. Assoc. Equip.* **2015**, *794*, 185–192.
18. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
20. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings, San Diego, CA, USA, 7–9 May 2015.
21. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
23. Sharma, O. A New Activation Function for Deep Neural Network. In Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 14–16 February 2019; pp. 84–86.
24. Zeng, X.; Cao, K.; Zhang, M. MobileDeepPill: A small-footprint mobile Deep learning system for recognizing unconstrained pill images. In Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services, Niagara Falls, NY, USA, 19–23 June 2017; pp. 56–67. [[CrossRef](#)]
25. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [[CrossRef](#)] [[PubMed](#)]
26. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
27. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
28. Yi, G.; Kim, Y.; Kim, S.; Kim, H.; Kim, K. Comparison and Verification of Deep Learning Models for Automatic Recognition of Pills. *J. Korea Multimed. Soc.* **2019**, *22*, 349–356.
29. Alexander Jung Imagaug. Available online: <https://imgaug.readthedocs.io/en/latest/> (accessed on 15 December 2021).
30. Mask R-CNN for Object Detection and Segmentation. Available online: [https://github.com/matterport/Mask\\_RCNN](https://github.com/matterport/Mask_RCNN) (accessed on 14 December 2021).
31. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
32. Cheng, B.; Schwing, A.G.; Kirillov, A. Per-Pixel Classification is Not All You Need for Semantic Segmentation. *arXiv* **2021**, arXiv:2107.06278.
33. Zhang, J.; Yang, K.; Constantinescu, A.; Peng, K.; Muller, K.; Stiefelwagen, R. Trans4Trans: Efficient Transformer for Transparent Object Segmentation to Help Visually Impaired People Navigate in the Real World. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021; pp. 1760–1770.