Plant Objects Detection Considering Environmental Noise

Kakeru Abe s1260085 Supervised by Prof. Keita Nakamura

Abstract

At disaster sites, factories, and other dangerous places, there is a very high demand for automatic inspection by robots. For this purpose, automatic operation of valves and levers is indispensable. However, the camera images from the robot are affected by environmental noise such as sunlight and background color, and the detection results is depending on the situation.

In this study, we aimed to realize object detection independent of environmental noise using two deep learning models. As a result of our experiments, we could implement high accuracy of object detection and orientation detection for levers, however, the accuracy of detection of valves is not sufficient. In the future, we would like to improve this and implement this in disaster robot.

1 Introduction

In disaster sites and factories, there are places where it is dangerous for people to enter, such as rubble and pipes, and disaster response robots are expected to perform an active role in such places. Through the Great Hanshin Earthquake and the Great East Japan Earthquake, a variety of disaster robots were developed and deployed in the field [1]. To be effective at such sites, it is essential to operate the robots remotely and obtain information from the attached cameras. In the World Robot Summit (WRS) [2], which is a competition to evaluate disaster robots, it is prohibited to operate a robot while looking at it from the outside for all tasks, so participants should operate it remotely.

If we can automate tasks such as valve or lever operation in these situations, it can be applied to automatic inspections in factories, and to realize it, automatic object detection is needed. In previous studies, color-based segmentation is used for valve detection [3]. However, the images from cameras change easily depending on the situation such as light reflection or background color, therefore it is difficult to discriminate by previous method. Therefore, in this research, we aim to achieve object detection that is not affected by environmental noise using deep learning.

2 Proposed method

In this experiment, we apply YOLOv3 algorithm in Darknet [4] which is deep learning framework.

2.1 Generation for learning data

A lot of images are needed for deep learning, and it takes time to prepare them manually. Therefore, we apply data argumentation for several images by using OpenCV [5]. First, we took image of valve and lever from different angles. Figure 2.1 explains how to measure the degree. Then, changed the colors and scaled the images and saved them. Figure 2.2 is example of argumentation of images. In this experiment, we prepare valve and lever of 1 inch model sold by KITZ corporation [6][7].

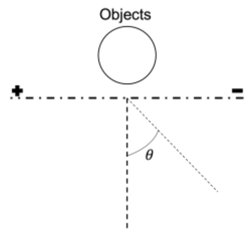


Figure 2.1: How to measure the degree

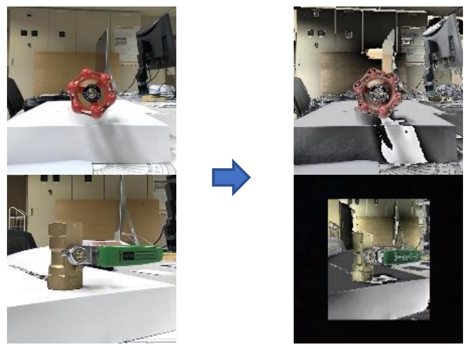


Figure 2.2: Example of image processing with OpenCV

For training YOLO, the text file that was created by the annotation process is also needed. This text file contains the class of the object, the coordinates of its center point, width and height. Figure 2.3 shows the example of text file. we copied and changed the contents of the file as the same time as argumentation.

```
0 0.498798 0.514423 0.305288 0.317308
```

$$[class, center\_X, center\_Y, width, height]$$

Figure 2.3: Example of annotation text file

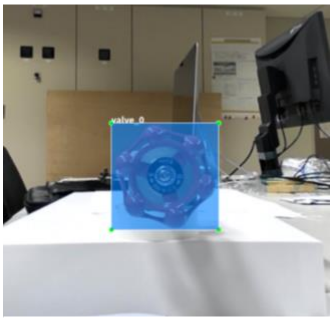


Figure 2.4: Annotation progress

2.2 Constructing YOLO DNN model

We classify the images as shown in Table2.1 and trained YOLO. The total number of images is 65,000 and the learning time is 9.5 hours. Figure 2.5 shows the relationship between iteration number and average training loss.

Table 2.1: Relationship between valve degree and class

valve / lever degree	class
[-90, -80)	valve_-90 / lever_-90
[-80, -50)	valve_-60 / lever_-60
[-50, -20)	valve_-30 / lever_-30
[-20, 20)	valve_0 / lever_0
[20, 50)	valve_30 / lever_30
[50, 80)	valve_60 / lever_60
[80, 90]	valve_90 / lever_90

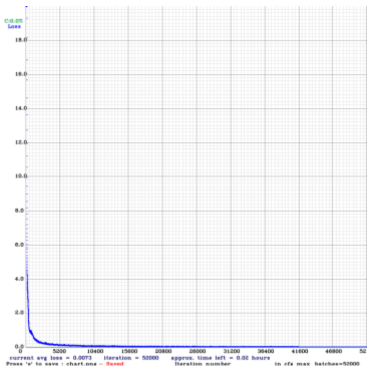


Figure 2.5: Relation between iteration and average loss

3 Experiment

3.1 Verification of the effectiveness of the proposed method

In this verification, it is important to indicate the effectiveness to environmental noise. Therefore, we prepared different images from the ones used for training section and adopted argumentation. Figure 3.1 shows the example of images. We apply object detection with learned YOLO model for 30 images for each angle and summarize to Tables 3.1. In these tables, count the number of image that nothing is detected as “NONE”. Figure 3.2 shows the flow of verification.

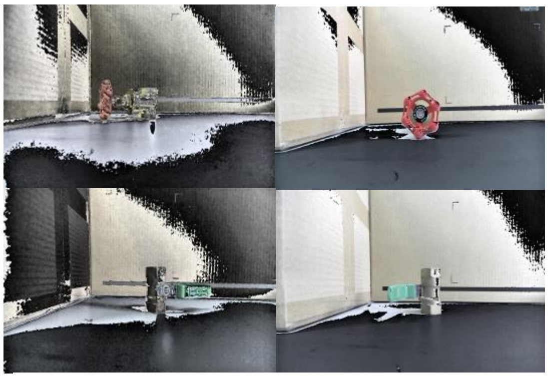


Figure 3.1: Example of images using in verification

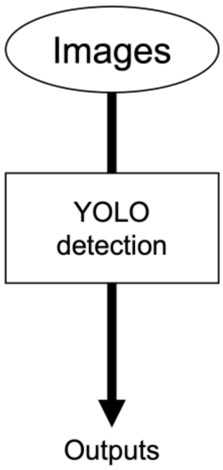


Figure 3.2: Flow of the verification

Expected degree
Nearest degree
Opposite degree

Table 3.1: Results of detection of YOLO model

(a) Results of levers

lever		Detection result							
		90	60	30	0	-30	-60	-90	NONE
Input	90	15	0	0	0	0	0	0	15
	60	28	0	0	0	0	0	0	2
	30	0	0	0	3	27	0	0	0
	0	0	0	0	30	0	0	0	0
	-30	0	0	0	0	25	0	0	5
	-60	0	0	0	0	0	1	0	29
	-90	11	0	0	0	11	0	4	4

(b) Results of valves

valve		Detection result							
		90	60	30	0	-30	-60	-90	NONE
Input	90	29	0	0	0	0	0	1	0
	60	0	30	0	0	0	0	0	0
	30	0	0	0	0	30	0	0	0
	0	0	0	0	30	0	0	0	0
	-30	0	0	0	0	30	0	0	0
	-60	8	2	0	0	0	0	0	20
	-90	4	0	0	0	0	0	15	11

3.2 Discussion

From the experimental results, the probability of detection at 0° is very high. However, when the lever is 30° or -90° and valve is 30°, the result shows the opposite degree, it means it is not able to discriminate whether the orientation is right or left. In addition, when the lever is -60° or -90° and the valve is -60° or -90°, almost half results show NONE.

4 Additional experiment

4.1 Reconstructing YOLO model and finetuning ResNet50 model

To solve previous problem, we attempted another way to determine the orientation. In previous study, QR code is places to determine the orientation[8]. However, there is no QR code in real situation. Therefore, we decided to use the YOLO model to detect only the angle of the object and use other deep learning models to determine the left or right orientation with only its objects. We finetuned ResNet50 [9] pretrained by ImageNet [10]. Classified the images used in the Sec.2.2 experiment as Tables 4.1, 4.2, and 4.3, and retrained YOLO and ResNet model. The total image number is 14,000 for YOLO and 6,000 for ResNet per training. The training time was 4.1 hours for YOLO and 30 minutes for ResNet. Figures 4.1 shows the relationship between training accuracy or loss and epoch in ResNet, and Figure 4.2 shows relationship between iteration number and loss in YOLO.

Table 4.1: Relationship between valve / lever degree and class

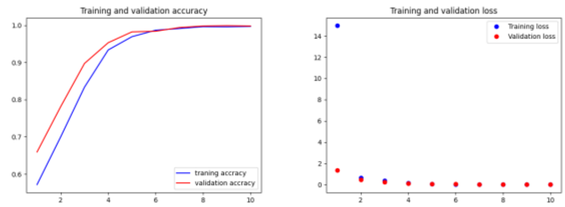
valve / lever degree		class
[0, 20)	[-20, 0)	valve_0 / lever_0
[20, 50)	[-50, -20)	valve_30 / lever_30
[50, 80)	[-80, -50)	valve_60 / lever_60
[80, 90]	[-90, -80]	valve_90 / lever_90

Table 4.2: Relationship between lever degree and class

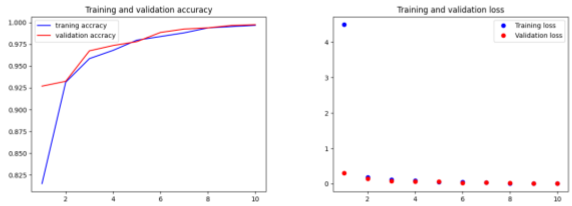
lever degree	class
[-90, 0]	left
(0, 90]	right

Table 4.3: Relationship between valve degree and class

valve degree	class
[-90, 0]	left
(0, 90]	right



(a) Training for levers



(b) Training for valves

Figure 4.1: Relation between accuracy or loss and epoch

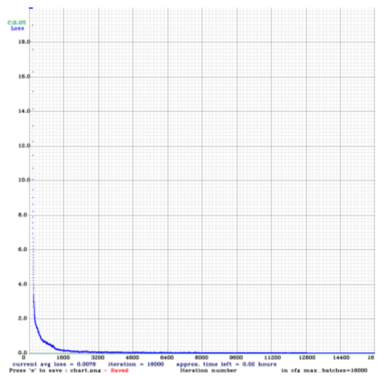


Figure 4.2: Relation between iteration

4.2 Verification of the effectiveness of the additional experiment

In this verification. the images used in Sec.3.1 verification is used. Figure 4.3 is example of detection of ResNet. We collect the result of detection and summarize to Table 4.4. Figure 4.4 shows the flow of verification.

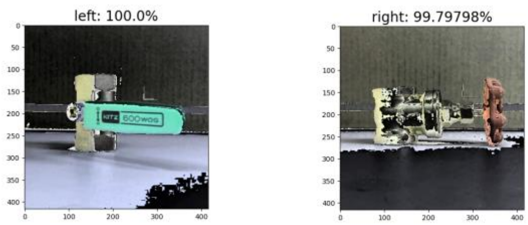


Figure 4.3: Example of ResNet detection

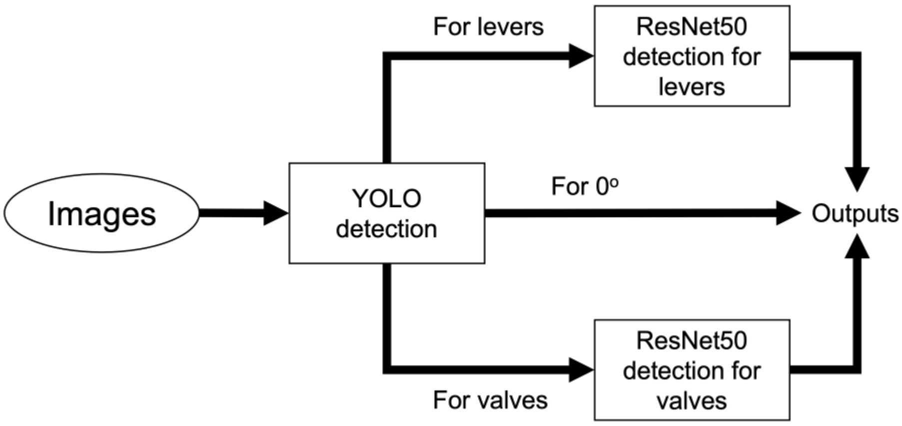


Figure 4.4: Flow of verification

Expected degree
Nearest degree
Opposite degree

Table 4.4: Results of detection with YOLO and ResNet

(a) Results of levers

lever			YOLO detection					ResNet detection	
			90	60	30	0	NONE	left (-)	right (+)
Input	right	90	30	0	0	0	0	4	26
		60	0	30	0	0	0	0	30
		30	0	0	30	0	0	0	30
		0	0	0	6	24	0	-	-
	left	-30	0	0	30	0	0	28	2
		-60	0	30	0	0	0	30	0
		-90	0	0	15	15	0	30	0

(b) Results of valves

valve			YOLO detection					ResNet detection	
			90	60	30	0	NONE	left (-)	right (+)
Input	right	90	30	0	0	0	0	5	25
		60	0	30	0	0	0	0	30
		30	0	0	30	0	0	24	6
		0	0	0	0	30	0	-	-
	left	-30	0	0	30	0	0	11	19
		-60	28	0	0	0	2	26	4
		-90	30	0	0	0	0	15	15

4.3 Additional discussion

As a result of this experiment, we succeeded in reducing the number of NONE and increasing the probability of object detection from various angles. However, when the lever is -90° or the valve is -60°, there are no correct detection results. On the other hand, for levers, there is high accuracy of orientation detection with ResNet. However, for valves, there is still room for improvement.

5 Conclusion and future work

In this research, we attempted to use two deep learning models to discriminate the orientation and the type of objects. For levers, the accuracy is efficient, however, for valves, it is needed to improve. In the future, we would like to improve the accuracy, and implement this in our disaster robot Spider2020.

Acknowledgement

I would thank Prof. Keita Nakamura, members of WRS2020, and all the people involved in this research.

References

[1] H.Asama, "R&D for Practical Robot Technology and Establishment of Its Operation System for Disaster Response",2014, Journal of the Robotics Society of Japan, Vol.32, No.1, pp. 37-41

[2] "World Robot Summit", https://wrs.nedo.go.jp/wrs2020/challenge/disaster/standard.html

[3] Shishiki Keito, "Object Recognition and Pose Estimation from RGB-D Data Using Active Sensing", Graduation research of University of Aizu, 2021.

[4] "darknet", https://github.com/AlexeyAB/darknet#yolo-v4-v3-and-v2-for-windows-and-linux

[5] "OpenCV", https://opencv.org

[6] https://www.kitz-valvesearch.com/products/detail.php?product_id=15819813

[7] https://www.kitz-valvesearch.com/products/detail.php?product_id=15817065#files

[8] Giorgia Pitteri, Aurelie Bugeau, Slobodan Ilic, Vincent Lepetit, "3D Object Detection and Pose Estimation of Unseen Objects in Color Images with Local Surface Embeddings"; Proceeding of the Asian Conference on Computer Vision (ACCV), 2020

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep residual learning for image recognition"; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778

[10] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255