



Weekly Work Report

Hongzhi Liu

VISION@OUC

August 26, 2018

1 Research problem

During this period of week, I spend time working about Faster R-CNN algorithm for off-line test and ROV equipment debugging in order to prepare URPC2018. Our team need to train 12 models that includes 11 kinds of image restoration and one image enhancement methods then output txt documents to evaluate algorithm performance with Yolo. Furthermore, I try to solve the error problems in running train code of flow-guided feature aggregation for video object detection.

Because of difficulty in code modification, I have difficulty in adding codes to realize mirror reversal image as data sets in order to add training image. However, we realize that what should be tested are four-fifths picture of all data set in chronological order. Then, I spend much time modifying train.txt and test.txt. At last, I should try to test contest models with different threshold value to evaluate Faster RCNN algorithm.

2 Research approach

In the process of research, I use the method of documentary analysis, comparative analysis and experimental research method. I read the thesis of Fast R-CNN [1], Faster R-CNN algorithm [2] and flow guided feature aggregation [3]. I try to understand core ideology in paper and learn about concept introduced by author.

Besides, I learn grammatical application of python on the one hand, and on the other hand, I try to write code files to achieve the goal. By this method, I can have a better understanding of python.

For deep learning, I watch the fifth course videos and write down the issues which I think are much important for further research. And then, I not only have learned the lessons of deep learning, but also put them into coding action.

3 Research progress

During preparation for URPC2018, I collated several data sets to compare results of different methods for testing original graphs and evaluate how good the restoration algorithm are with mAP values. I continue to learn about Faster R-CNN algorithm [2] and relevant theory of flow guided feature aggregation [3]. Besides, I change the original image size to 256×256 pixel and then resize to origin size. Then I use algorithm of python codes to complete the process of image mirror transformation. Furthermore, Our team has carried out commissioning of ROV underwater equipment. And I will list details about weekly work in Tab. 1 below.

Table 1: Weekly work progress.

URPC2018	Finish collating several data sets to train contest model. Finish comparing results of different methods. Finish training a lot of model evaluation data with different threshold. Finish comparing mAP value of different methods and judge which are better. Finish debugging ROV equipment.
Deep Learning	Finish learning Sequence Models which is the fifth course of deep learning.

4 Progress in this week

During preparation for URPC2018, I have collated several data sets to train contest model. Then I use algorithm to complete the process of image mirror transformation. Besides, I test origin image models and obtain mAP value with different thresholds. Furthermore, our team carry out commissioning of ROV underwater equipment.

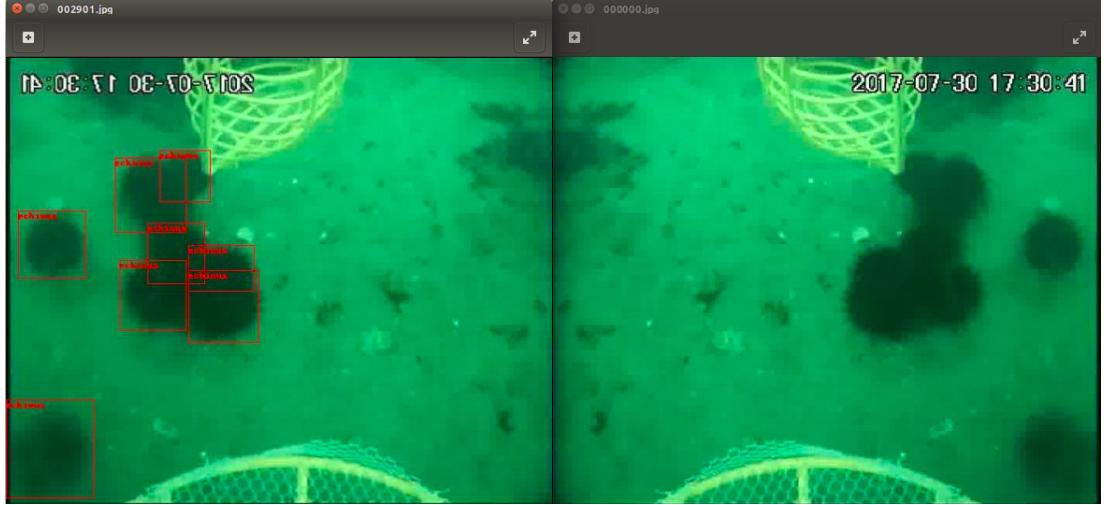


Figure 1: Data sets of picture mirror overturn method.

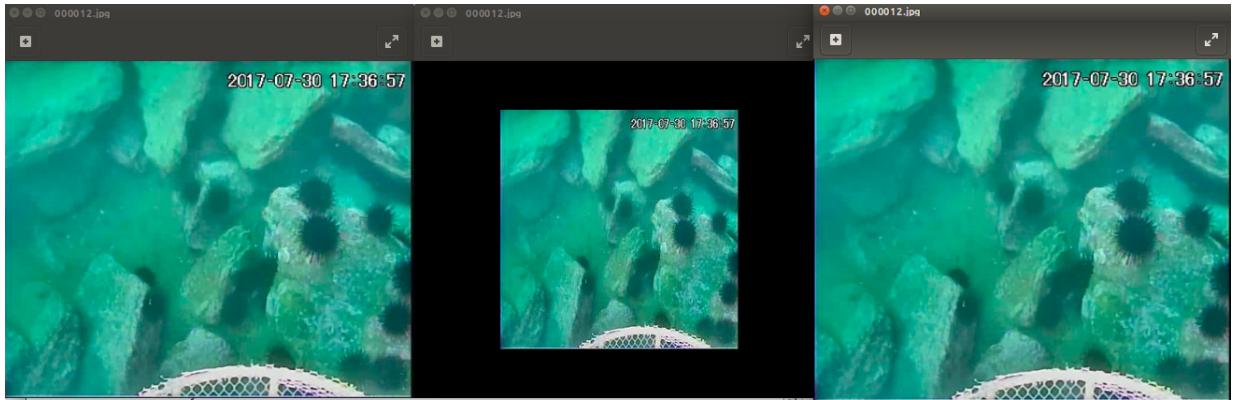


Figure 2: Data sets of resizing origin images.

Step 1 Finish collatting several data sets to train contest model.

Step 2 Finish training a lot of model evaluation data with different threshold.

Step 3 Finish comparing mAP value of different methods and judge which are better.

Step 4 Finish debugging ROV equipment.

4.1 Training Data Sets

In this week, I have collated several data sets to train contest model and compare results of different methods for testing original graphs, which can better evalute performance of Faster RCNN algorithm. According to the teacher's request, I should train origin images and their mirror convert pictures first as seen in Fig. 1. The two pictures are symmetrical in the axis of the mirror. This can be regarded as an effective way of image enhancement. Besides, I change the original image size to 256×256 pixel and then resize to origin size following recommendation of the teacher as seen in Fig. 2.

Coordinate position of bounding box after mirror transformation can be obtained by Eq. 1 if knowing the width and height of every picture in data sets. Then I use Algorithm 4.1 to complete the process of image mirror transformation. Besides, I realize that this conversion, in turn, would require changes to all xml files for training models, so I use Algorithm 4.1 to modified coordinates of bounding boxes such

as xmin and ymin.

$$\begin{cases} xmin = width - xmax \\ ymin = ymin \\ xmax = width - xmin \\ ymax = ymax \end{cases} \quad (1)$$

```

1 set -e # or use "set -o errexit" to quit on error.
2 set -x # or use "set -o xtrace" to print the statement
3 before you execute it.
4 FILES=*.jpg
5 for f in $FILES
6 do
7 echo "$f"
8 convert $f -flop $f
9 done

```

```

1 # coding: utf-8
2 import math
3 import numpy as np
4 import xml.etree.ElementTree as ET
5 import os
6
7 xmlpath = './5801xml/'
8 rotated_xmlpath = 'rotatedxml'
9 for i in os.listdir(xmlpath):
10 print(i)
11 a, b = os.path.splitext(i)
12 print(a, b)
13 tree = ET.parse(xmlpath + a + '.xml')
14 root = tree.getroot()
15 for ch in root.iter('size'):
16     width = float(ch.find('width').text)
17     for box in root.iter('bndbox'):
18         xmin = float(box.find('xmin').text)
19         ymin = float(box.find('ymin').text)
20         xmax = float(box.find('xmax').text)
21         ymax = float(box.find('ymax').text)
22         box.find('xmin').text = str(width-xmax)
23         box.find('ymin').text = str(ymin)
24         box.find('xmax').text = str(width+xmin)
25         box.find('ymax').text = str(ymax)
26     tree.write(rotated_xmlpath + a + '_' + d + '.xml')
27 print(str(a) + '.xml_has_been_rotated_for_' + d)

```

4.2 Test Contest Models

After a period of training and testing several models, I get a lot of model evaluation data with 0.8 threshold, but are far from enough. So I test origin image models and obtain mAP value with different thresholds which can be seen in Fig. 3. We can see from the picture that mAP value increases with decreasing threshold.

After I train a contest model with on-line data sets, I find the output result is not very good as shown in Fig. 4. So we check xml files and label numbers to train again.

		0.8	0.919	0.929	0.732	0.94	0.88	0.924
		0.75	0.922	0.933	0.749	0.944	0.887	0.927
		0.7	0.923	0.938	0.754	0.947	0.891	0.931
		0.65	0.925	0.941	0.759	0.951	0.894	0.933
		0.6	0.928	0.942	0.762	0.953	0.896	0.935
		0.55	0.929	0.943	0.767	0.954	0.898	0.936
		0.5	0.93	0.944	0.774	0.955	0.901	0.937
		0.45	0.93	0.945	0.779	0.955	0.902	0.938
		0.4	0.932	0.946	0.786	0.956	0.905	0.939
原图+原图resize	原图	0.35	0.932	0.947	0.789	0.956	0.906	0.939
		0.3	0.932	0.948	0.799	0.958	0.909	0.94
		0.25	0.933	0.949	0.806	0.958	0.912	0.941
		0.2	0.934	0.95	0.806	0.958	0.912	0.942
		0.15	0.934	0.951	0.808	0.96	0.913	0.943
		0.1	0.936	0.951	0.81	0.96	0.914	0.943
		0.05	0.938	0.952	0.812	0.96	0.915	0.945
		0.01	0.95	0.953	0.824	0.96	0.922	0.952
		0.005	0.951	0.954	0.825	0.961	0.923	0.953
		0.001	0.952	0.954	0.826	0.962	0.924	0.953
原图+原图镜像	原图	0	0.952	0.954	0.826	0.963	0.924	0.953
		0.8	0.908	0.933	0.737	0.947	0.881	0.92
		0	0.95	0.951	0.809	0.961	0.918	0.95

Figure 3: Output results of two model. The first column is threshold. Then the four columns representative holothurian, echinus, scallop and starfish. The sixth column is mAP value.

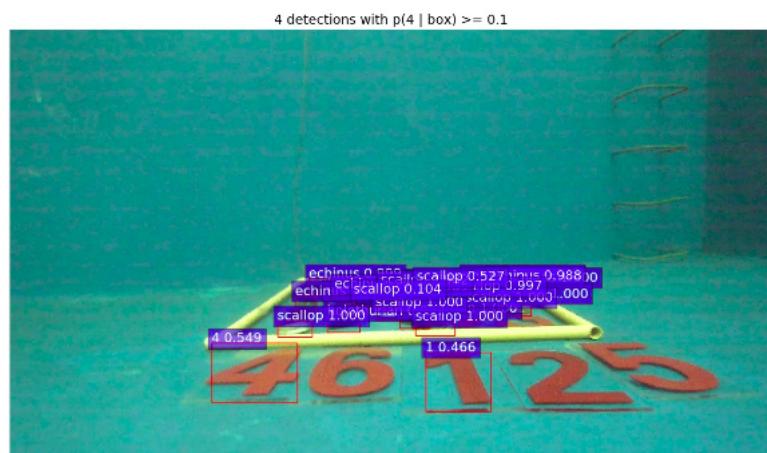


Figure 4: Output result picture of contest model with on-line data sets.

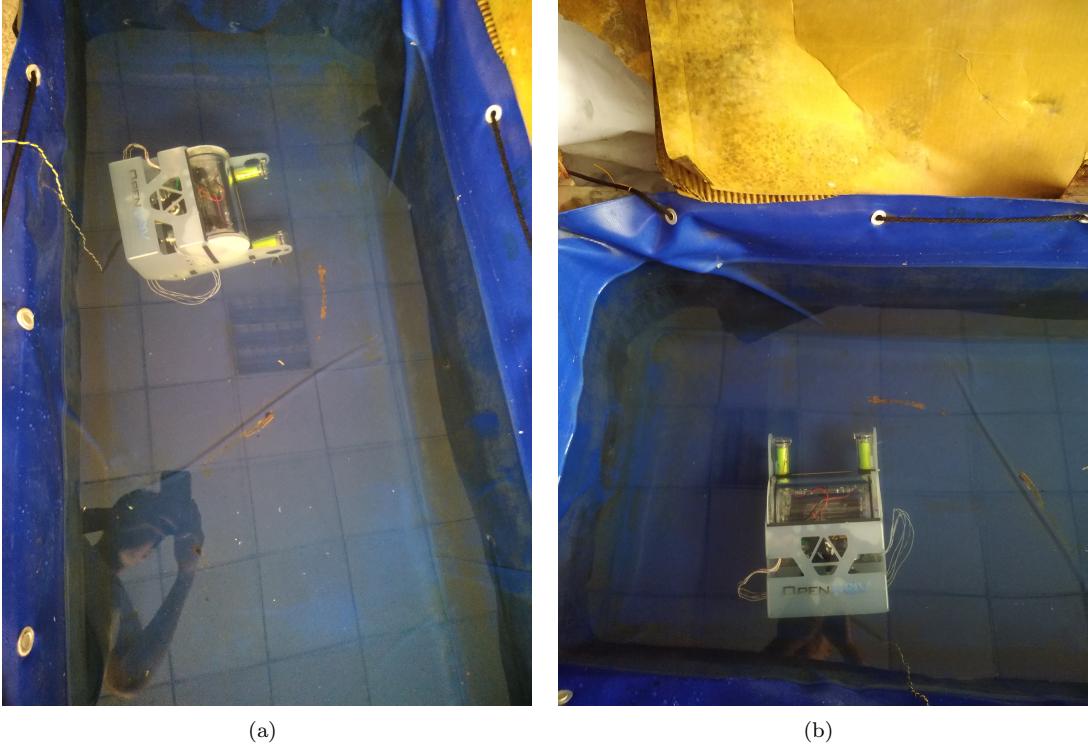


Figure 5: ROV equipment on-site commissioning process.

```

1  def demo(sess , net , image_name):
2      # Load the demo image
3      print( 'demoDemo_for_data/demo/{}{}'.format(im_name[0] , '.jpg' ))
4      print( 'demoDemo_for_data/demo/{}{}'.format(im_name[1] , '.jpg' ))
5      print( '\n' )
6
7      all_name = image_name+'.jpg'
8      im_file = os.path.join(cfg.DATA_DIR, 'demo' , all_name)
9      im = cv2.imread(im_file)
10
11     fr = open( '/home/henry/Files/tf-faster-rcnn-contest/data/
12     VOCdevkit2007/test_list.txt' , 'r' )
13     for im_name in fr :
14         im_name = im_name.strip()
15         im_name = im_name.split(',')
16         print( '~~~~~' )
17         print( 'mainDemo_for_data/demo/{}{}'.format(im_name[0] , '.jpg' ))
18         print( 'mainDemo_for_data/demo/{}{}'.format(im_name[1] , '.jpg' ))
19         print( '\n' )
20     demo(sess , net , im_name[0])

```

4.3 ROV Equipment Debugging

Our team has carried out commissioning of ROV underwater equipment this week as Fig. 5. We encountered many problems when we debugged the device. First of all, steering gear turns opposite and we change the battery and work well. Then we adjust the direction of the camera and can get image from camera. Furthermore, what we need to do next step is equipment and program docking, which

decides whether to win or lose the contest.

5 Plan

Objective: Finish off-line and on-line test for 2018URPC.

Deadline: 2018.08.31

2018.08.27—2018.09.02 Finish preparation for 2018URPC.

2018.09.03—2018.09.09 Finish literature review about image synthesis.

2018.09.10—2018.09.16 Finish reading reference documentation.

References

- [1] R. Girshick. Fast R-CNN. In *ICCV*, 2015. [1](#)
- [2] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. [1](#)
- [3] X. Zhu, Y. Wang, J. Dai, L. Yuan, and W. Yichen. Flow-guided feature aggregation for video object detection. In *ICCV*, 2017. [1](#)