Weekly Work Report

Hongzhi Liu

VISION@OUC

September 2, 2018

1 Research problem

During this week's short and tense preparation time, our team take much efforts to deal with hardware and software problems. First, we need to counterweight for open ROV equipment in order to smooth the equipment in the water. Then we should try to make the Faster RCNN algorithm read the video stream from the device normally, object recognition and number counting when equipment is in the water. Besides, I prepare and test some model weights for contest that can help us to choose a better one with high mAP. Furthermore, I train two online model weights to detect holothurian, echinus, scallop and number one to six which representative six regions.

Because of difficulty in code modification for online algorithm version, I have little time for improving online recognition accuracy and performance. Until the day before the competition, we are still adjusting and modifying the algorithm codes in order to keep device in a better state and be able to get a ideal result. Fortunately, we run smoothly on the day of code running and the operation equipment as well.

2 Research approach

In the process of contest preparation, I use the method of documentary analysis, comparative analysis and experimental research method. I read the thesis of RCNN[2], Fast R-CNN [1] and Faster R-CNN algorithm [3]. I try to unferstand 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 have trained offline and online models for competition and evaluate how good the restoration algorithm are with mAP values. I continue to test origin pictures about Faster R-CNN algorithm [3]. Besides, I change different threshold value to test picture in order to get a better mAP. And I will list details about weekly work in Tab. 1 below.

Table 1: Weekly work progress.

URPC2018	Finish training offline and online models for competition. Finish comparing results of different threshold. Finish comparing mAP value of different methods and judge which are better. Finish online and offline tests on site.
Deep Learning	Finish learning Sequence Models which is the fifth course of deep learning.

4 Progress in this week

When we are in URPC2018 site, I have trained offline and online models for competition. And we finish online and offline testing on site smoothly.

Step 1 Finish training an online contest model.

Step 2 Finish testing four offline models.

Step 3 Finish comparing mAP value of different methods and decide which threshold value to use.

Step 4 Finish online and offline testing on site smoothly.

		Faster-CNN					
		阈值	holothurian	echinus	scallop	starfish	mean AP	Median AP
	原图	0.8	0.928	0.933	0.761	0.954	0.894	0.931
		0.8	0.847	0.9	0.663	0.904	0.828	0.873
	clahe2							
	clahe6	0.8	0.759	0.876	0.608	0.849	0.773	0.804
clahe	hsv	0.8	0.842	0.899	0.594	0.915	0.813	0.87
	dcp	0.8	0.916	0.939	0.73	0.954	0.885	0.928
	dcp_guided	0.8	0.931	0.938	0.77	0.956	0.899	0.935
dcp	dcp+clahe	0.8	0.929	0.935	0.748	0.965	0.894	0.932
	cyclegan_sim	0.8	0.861	0.914	0.659	0.88	0.828	0.87
deep	cyclegan_sim5	0.8	0.87	0.893	0.629	0.896	0.822	0.882
	dcp(red)	0.8	0.886	0.635	0.685	0.713	0.73	0.699
	dcp(red)+guided	0.8	0.919	0.927	0.74	0.958	0.886	0.923
dcp(red)	dcp(red)+clahe	0.8	0.929	0.949	0.751	0.965	0.899	0.939

Figure 1: Output test results of several models.

4.1 Contest Models Preparation

In order to prepare model weights for offline and online, I train several mdoels with cla2, cla6, hsv, dcp, dcp_guided, dcp with clahe, cycle ssim, cycle ssim5 and so on. On this basis, I test them with some threshold from 0.8 to 0.001 and get results as shown in Fig. 1. Finally, when we are in game site, I choose threshold equal to 0.01 with original model weight.

Except for what we need to see mAP value for evaluation, I try to write codes to output visualization results, for example, marking objects that are given to the game with Algorithm 4.1.

```
def demo(sess, net, image_name):
"""Detect object classes in an image using pre-computed object proposals."""

all_name = image_name + '.jpg'
im_file = os.path.join(cfg.DATA_DIR, 'demo', all_name)
im = cv2.imread(im_file)

timer = Timer()
timer.tic()
scores, boxes = im_detect(sess, net, im)
timer.toc()
print('Detection took {:.3f}s for {:d} object proposals'.format(timer.total_time, bo

CONF_THRESH = 0.01
NMS_THRESH = 0.3
im = im[:, :, (2, 1, 0)]
fig,ax = plt.subplots(figsize=(12, 12))
ax.imshow(im, aspect='equal')

for cls_ind, cls in enumerate(CLASSES[1:]):
cls_ind += 1 # because we skipped background
cls_boxes = boxes[:, 4*cls_ind:4*(cls_ind + 1)]
cls_scores = scores[:, cls_ind]
dets = np.hstack((cls_boxes,
cls_scores[:, np.newaxis])).astype(np.float32)
keep = nms(dets, NMS_THRESH)
dets = dets[keep, :]
```

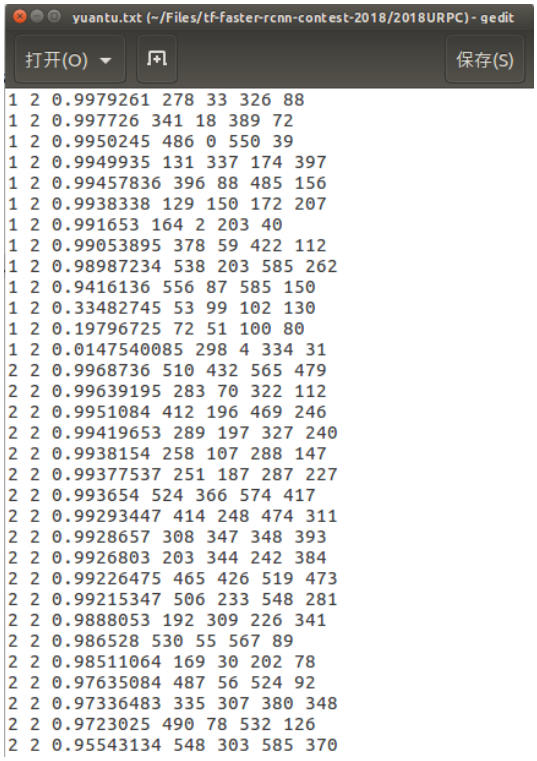
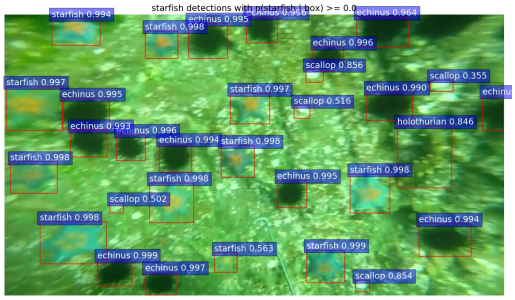
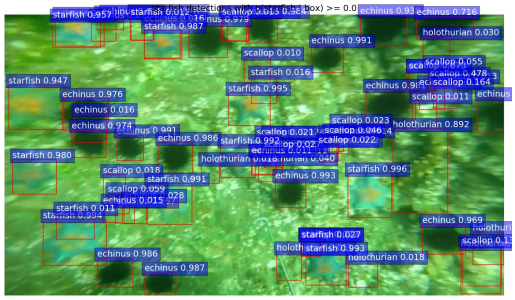


Figure 2: Contest result of different models that are origin model (or yuantu model), enhanced model, 17&18 model and dcp model.

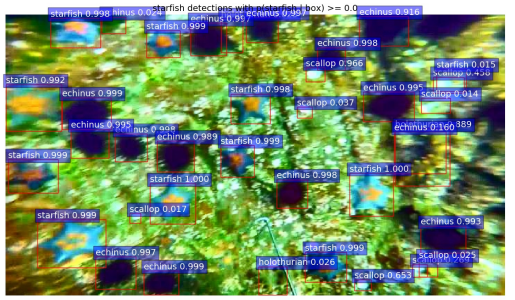
(a)



(b)



(c)



(d)

Figure 3: Pictures with bounding boxes which are tested by different models.

```
28
29  vis_detections(im, cls, dets,ax,thresh=CONF_THRESH)
30  plt.axis('off')
31  plt.tight_layout()
32  plt.draw()
33
34  if __name__ == '__main__':
35  cfg.TEST.HAS_RPN = True  # Use RPN for proposals
36  args = parse_args()
37  cfg.USE_GPU_NMS = False
38
39  demonet = args.demo_net
40  dataset = args.dataset
41  tfmodel = os.path.join('output', demonet, DATASETS[dataset][0], 'default',
42  NETS[demonet][0])
43
44
45  if not os.path.isfile(tfmodel + '.meta'):
46  raise IOError(('{:s} not found.\nDid you download the proper networks from '
47  'our server and place them properly?').format(tfmodel + '.meta'))
48
49  tfconfig = tf.ConfigProto(allow_soft_placement=True)
50  tfconfig.gpu_options.allow_growth=True
51  sess = tf.Session(config=tfconfig)
52
53  if demonet == 'vgg16':
54  net = vgg16()
55  elif demonet == 'res101':
56  net = resnetv1(num_layers=101)
57  else:
58  raise NotImplementedError
59  net.create_architecture("TEST",5,
60  tag='default', anchor_scales=[8, 16, 32])
61  saver = tf.train.Saver()
62  saver.restore(sess, tfmodel)
63
64  print('Loaded network {:s}'.format(tfmodel))
65  fr = open('/home/ai/Liuhongzhi/tf-faster-rcnn-contest-2018/data/VOCdevkit2007/test_l
66  for im_name in fr:
67
68  im_name = im_name.strip()
69  im_name = im_name.split(' ')
70  print('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~')
71  print('mainDemo for data/demo/{}{}'.format(im_name[0], '.jpg'))
72  print('mainDemo for data/demo/{}{}'.format(im_name[1], '.jpg'))
73  demo(sess, net, im_name[0])
74  plt.savefig("testfigs/" + im_name[0] + '.jpg')
75  #plt.show()
```

4.2 2018URPC Competition Site

Our offline contest began at 1 o'clock in the afternoon on September 2nd. The result of submission as shown in Fig. 2. We test pictures with origin model first and result as Fig. 2(a). Then we quickly test with enhanced model as Fig. 2(b). Thirdly, we test with 17&18 model Fig. 2(c). At last, we test with dcp model Fig. 2(d). Thanks to our partners, I can get enhanced pictures to test and get results in time.

Furthermore, I can visualization of test results with bounding boxes in pictures as shown in Fig. 3. We test pictures with origin model first and result as Fig. 3(a). Then we quickly test with enhanced model as Fig. 3(b). Thirdly, we test with 17&18 model Fig. 3(c). At last, we test with dcp model Fig. 3(d). Thanks to our partners, I can get enhanced pictures to test and get results in time.

5 Plan

Objective: Finish thesis with senior students for CVPR2019.
Deadline: 2018.11.16

2018.09.03—2018.09.09 Finish reading CVPR2018 paper about AttnGAN [4].

2018.09.10—2018.09.16 Finish recurrenting results of table from CVPR2018 paper.

References

[1] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 1

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1

[3] 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

[4] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018. 6