



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	holothurianechinus	scallop	starfish	mean AP	Median AP	
	原图	0.8	0.928	0.933	0.761	0.954	0.894	0.931
clahe		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
	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	dcp_guided	0.8	0.931	0.938	0.77	0.956	0.899	0.935
	dcp+clahe	0.8	0.929	0.935	0.748	0.965	0.894	0.932
deep	cyclegan sim	0.8	0.861	0.914	0.659	0.88	0.828	0.87
	cyclegan sim5	0.8	0.87	0.893	0.629	0.896	0.822	0.882
dcp(red)	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)+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 models 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.

```

1 def demo(sess, net, image_name):
2     """Detect object classes in an image using pre-computed object proposals."""
3
4     all_name = image_name + '.jpg'
5     im_file = os.path.join(cfg.DATA_DIR, 'demo', all_name)
6     im = cv2.imread(im_file)
7
8     timer = Timer()
9     timer.tic()
10    scores, boxes = im_detect(sess, net, im)
11    timer.toc()
12    print('Detection took {:.3f}s for {:.1d} object proposals'.format(timer.total_time, boxes.size))
13
14    CONF_THRESH = 0.01
15    NMS_THRESH = 0.3
16    im = im[:, :, (2, 1, 0)]
17    fig, ax = plt.subplots(figsize=(12, 12))
18    ax.imshow(im, aspect='equal')
19
20    for cls_ind, cls in enumerate(CLASSES[1:]):
21        cls_ind += 1 # because we skipped background
22        cls_boxes = boxes[:, 4*cls_ind:4*(cls_ind + 1)]
23        cls_scores = scores[:, cls_ind]
24        dets = np.hstack((cls_boxes,
25                          cls_scores[:, np.newaxis])).astype(np.float32)
26        keep = nms(dets, NMS_THRESH)
27        dets = dets[keep, :]

```

```
yuantu.txt (~/Files/tf-faster-rcnn-contest-2018/2018URPC) - gedit
打开(O) 保存(S)

1 2 0.9979261 278 33 326 88
1 2 0.997726 341 18 389 72
1 2 0.9950245 486 0 550 39
1 2 0.9949935 131 337 174 397
1 2 0.99457836 396 88 485 156
1 2 0.9938338 129 150 172 207
1 2 0.991653 164 2 203 40
1 2 0.99053895 378 59 422 112
1 2 0.98987234 538 203 585 262
1 2 0.9416136 556 87 585 150
1 2 0.33482745 53 99 102 130
1 2 0.19796725 72 51 100 80
1 2 0.0147540085 298 4 334 31
2 2 0.9968736 510 432 565 479
2 2 0.99639195 283 70 322 112
2 2 0.9951084 412 196 469 246
2 2 0.99419653 289 197 327 240
2 2 0.9938154 258 107 288 147
2 2 0.99377537 251 187 287 227
2 2 0.993654 524 366 574 417
2 2 0.99293447 414 248 474 311
2 2 0.9928657 308 347 348 393
2 2 0.9926803 203 344 242 384
2 2 0.99226475 465 426 519 473
2 2 0.99215347 506 233 548 281
2 2 0.9888053 192 309 226 341
2 2 0.986528 530 55 567 89
2 2 0.98511064 169 30 202 78
2 2 0.97635084 487 56 524 92
2 2 0.97336483 335 307 380 348
2 2 0.9723025 490 78 532 126
2 2 0.95543134 548 303 585 370
```

(a)

```
newhance.txt (~/Files/tf-faster-rcnn-contest-2018/2018URPC) - gedit
打开(O) 保存(S)

1 1 0.034670684 4 291 98 450
1 2 0.99352455 410 86 484 153
1 2 0.99347275 132 340 173 395
1 2 0.99197996 130 154 171 207
1 2 0.9902422 278 36 324 88
1 2 0.9882181 340 18 389 72
1 2 0.9855226 164 1 205 42
1 2 0.9771139 489 0 554 44
1 2 0.9758631 377 53 426 113
1 2 0.9704105 535 199 585 262
1 2 0.9354023 558 93 585 151
1 2 0.6804663 301 5 335 39
1 2 0.53445894 453 119 489 162
1 2 0.029116783 449 0 486 24
1 2 0.018197674 451 0 486 6
1 2 0.012541891 50 97 105 128
1 3 0.15703551 543 158 585 207
1 3 0.052172817 238 13 282 59
1 3 0.033669837 233 61 283 115
1 3 0.015595727 8 294 88 427
1 4 0.011577544 233 32 274 67
2 2 0.996505 202 342 241 385
2 2 0.9955332 526 366 574 416
2 2 0.99552625 506 233 548 281
2 2 0.9949604 514 433 568 478
2 2 0.9949469 289 196 327 240
2 2 0.9906335 192 308 227 344
2 2 0.9881981 415 195 467 246
2 2 0.9875473 258 108 289 147
2 2 0.98640394 307 348 349 394
2 2 0.98536634 415 246 474 311
2 2 0.9838472 357 299 391 340
```

(b)

```
17+18.txt (~/Files/tf-faster-rcnn-contest-2018/2018URPC) - gedit
打开(O) 保存(S)

1 1 0.013726618 21 113 58 146
1 1 0.013068258 216 58 277 115
1 1 0.010497107 1 77 31 123
1 2 0.9883708 341 27 389 70
1 2 0.9856646 276 36 328 91
1 2 0.98487955 130 155 170 207
1 2 0.9844861 376 56 427 113
1 2 0.97731847 406 85 479 152
1 2 0.97006464 535 201 585 262
1 2 0.9687164 166 0 205 40
1 2 0.95798445 133 344 172 392
1 2 0.9517234 307 3 358 35
1 2 0.901074 498 2 544 36
1 2 0.78112334 555 90 584 154
1 2 0.7274874 455 130 487 162
1 2 0.6085122 1 119 37 158
1 2 0.57356876 48 96 100 132
1 2 0.2820728 272 106 308 138
1 2 0.26322153 333 163 359 189
1 2 0.22007272 543 13 578 47
1 2 0.1929161 324 14 367 56
1 2 0.08395462 65 50 99 84
1 2 0.079914704 298 19 344 64
1 2 0.0728235 109 95 141 131
1 2 0.041386478 1 78 29 124
1 2 0.038457435 522 185 565 242
1 2 0.03195945 488 170 554 215
1 2 0.031769704 0 1 41 33
1 2 0.031124748 210 51 247 85
1 2 0.02617488 542 102 575 189
1 2 0.022827018 0 420 39 479
1 2 0.017619325 521 1 562 23
```

(c)

```
dcp.txt (~/Files/tf-faster-rcnn-contest-2018/2018URPC) - gedit
打开(O) 保存(S)

1 2 0.99774593 403 85 482 155
1 2 0.99413884 541 208 585 262
1 2 0.99373806 343 26 388 70
1 2 0.9927898 132 337 171 396
1 2 0.98529726 374 59 426 110
1 2 0.9777238 280 34 324 83
1 2 0.9767386 165 1 203 41
1 2 0.9693266 557 88 585 150
1 2 0.95978343 132 154 171 205
1 2 0.9352527 70 49 100 85
1 2 0.68099964 453 133 487 162
1 2 0.17563431 489 3 530 27
1 2 0.052405033 297 20 336 56
1 2 0.029107343 308 5 343 36
1 2 0.0118042445 541 51 584 269
1 3 0.014163057 548 148 579 198
2 1 0.020280221 472 431 514 471
2 2 0.99809605 505 233 547 281
2 2 0.99709904 290 196 327 242
2 2 0.9970139 204 337 244 385
2 2 0.9964715 526 366 576 417
2 2 0.9950452 509 433 568 478
2 2 0.9947043 191 309 228 342
2 2 0.99402064 278 71 323 120
2 2 0.9937158 416 249 473 306
2 2 0.9935551 539 264 581 301
2 2 0.993233 255 189 284 225
2 2 0.99310184 305 345 354 395
2 2 0.992622 414 197 467 243
2 2 0.9917368 485 57 523 90
2 2 0.9904096 259 111 289 148
2 2 0.99025375 494 82 538 121
```

(d)

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

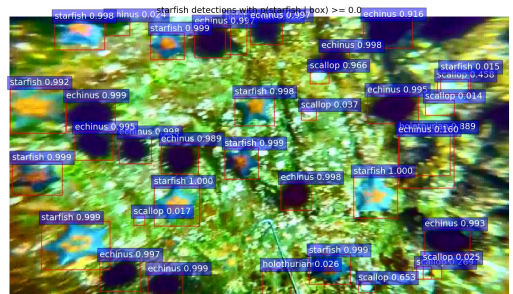
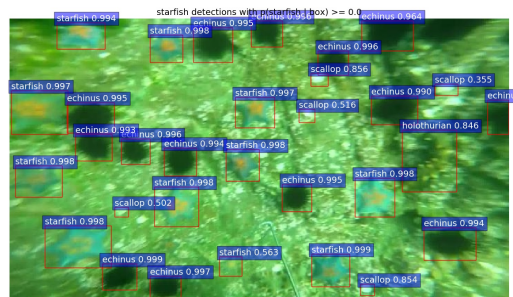


Figure 3: Pictures with bounding boxes which are tested by different models.

```

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.demonet
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_1
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