



# Weekly Work Report

Hongzhi Liu

VISION@OUC

August 19, 2018

# 1 Research problem

During this period of week, I spend time studying deep learning courses and working about Faster R-CNN algorithm for off-line test and FGFA algorithm for video object detection method 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 the function which can read test\_list.txt then output a txt which includes information about picture ID, class, confidence and correct bounding box. However, we realize that what should be tested are picture excluding pictures from training sessions. Then, I spend much time deleting items that are not required in test\_list.txt. At last, I should try to train a contest model by using flow guided feature aggregation 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 unferstand core ideology in paper and learn about concept introduced by author.

Besides, I learn grammatical structure of python on the one hand, and on the other hand, I try to write script files to achieve batch processing commands. 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 receive an image enhancement datasets to train models and test 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]. Furthermore, I compare the results of 11 image restoration methods that are clahe2, clahe6, hsv, dcp, dcp\_guided, dcp(red), dcp(red)+guided, dcp+clahe, dcp(red)+clahe, ssim and ssim5 as well as 1 enhancement method after test with pictures from 2018URPC. By using the remote server, I try to train contest models with FGFA algorithm and solve the porblem encountered when running a program with help of senior student. I will list details about weekly work in Tab. 1 below.

Table 1: Weekly work progress.

URPC2018	Finish training and testing 11 kinds of image restoration methods.
	Finish preparing data sets and xml files of 20307 pictures for training the enhancement method.
	Finish resizing pictures and modified test_list file to evaluate performance of models.
	Finish comparing mAP value of 12 methods and judge which are better.
	Finish solving several error problems when running FGFA algorithm.
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 reordered four image restoration datasets upload to server. Then I spend time checking why output results of mAP value is too low and know the reason finally. After getting the correct result of mAP value of four image restoration methods, we compare them with origin picture to sum up experience. Besides, I modified code to make the output text is in correct method according to the official description. Furthermore, I can run a FGFA demo and solve the problem in running code.

**Step 1** Finish modifying test\_list.txt and delete items that are not required in it in order to choose test pictures for evaluation.

**Step 2** Finish training and testing 11 kinds of image restoration methods.

**Step 3** Finish preparing data sets and xml files of enhanced pictures for training the method.

**Step 4** Finish learning Sequence Models which is the fifth lesson.

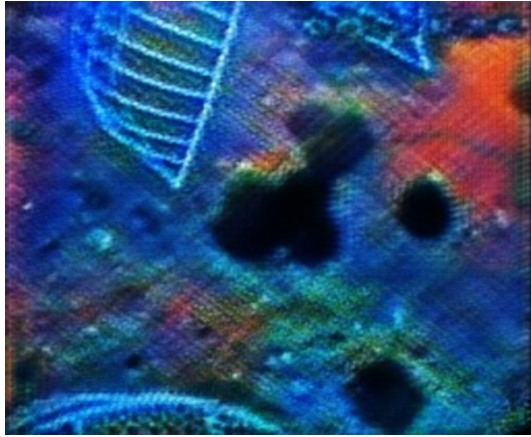
### 4.1 Data Sets

In this week, our team received an enhanced method which includes six data sets more than twenty thousand pictures as shown in Fig. 1. We train the contest model with the these datasets. Besides, I reordered B3, B4, B5, B6, green and blue pictures to train and test. Fig. 1(a) is the picture restored by B3 method, Fig. 1(b) is the picture restored by B4 method. Fig. 1(c) is the picture restored by B5 method. Fig. 1(d) is the picture restored by B6 method. Fig. 1(e) is the picture restored by green method. Fig. 1(f) is the picture restored by blue method. And we use Algorithm 4.1 to modified frame name in xml files to VOC format.

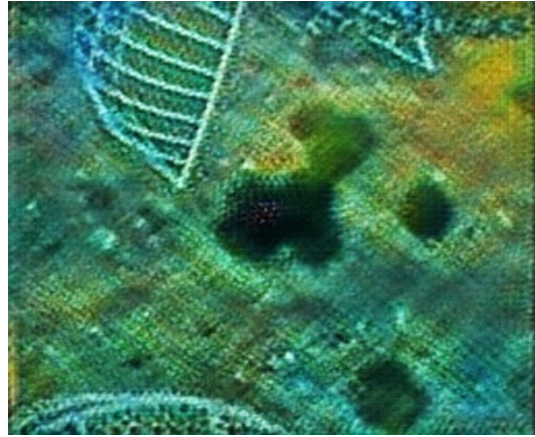
```
1 #coding=utf-8
2 import os
3 import os.path
4 import xml.dom.minidom
5 path="./enhance0815"
6 files=os.listdir(path)
7 s=[]
8 for xmlFile in files:
9     portion = os.path.splitext(xmlFile)
10    if not os.path.isdir(xmlFile):
11        dom=xml.dom.minidom.parse(os.path.join(path,xmlFile))
12        root=dom.documentElement
13        name=root.getElementsByTagName('frame')
14        for i in range(len(name)):
15            print(xmlFile)
16            if portion[1] == ".xml":
17                newname = portion[0]+".jpg"
18                print(newname)
19                name[i].firstChild.data=newname
20                print(name[i].firstChild.data)
21                with open(os.path.join(path,xmlFile), 'w', encoding='UTF-8') as fh:
22                    dom.writexml(fh)
23                print('filename_OK!')
```

### 4.2 Test a Contest Model

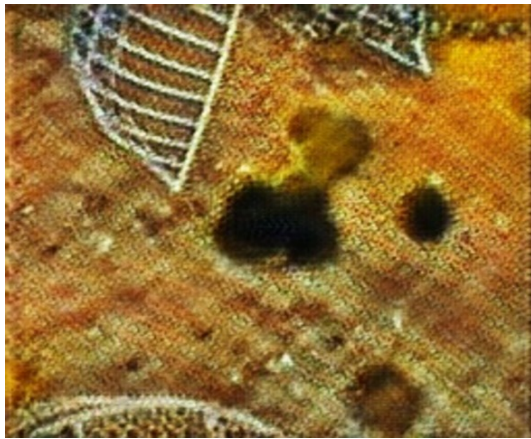
Accodting to the six data sets of image enhancement method, we train the model and get the results of it can be seen in Fig. 3. Each picture with a bounding box in Fig. 2, corresponding to the picture mentioned in Fig. 1 above.



(a)



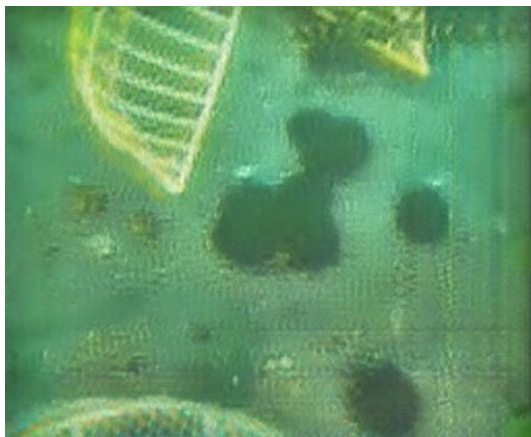
(b)



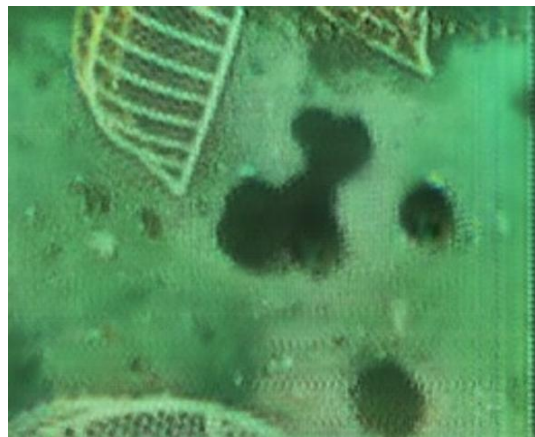
(c)



(d)



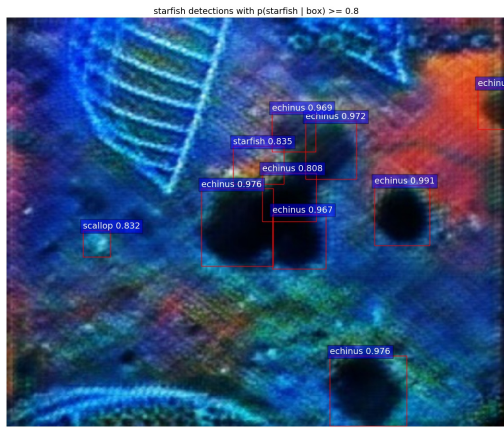
(e)



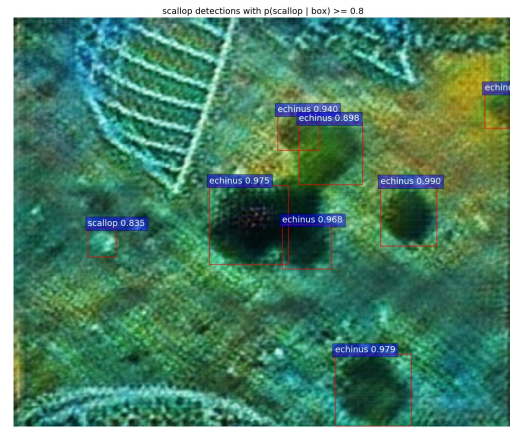
(f)

Figure 1: The same picture is enhanced by six kinds of image enhancement method.

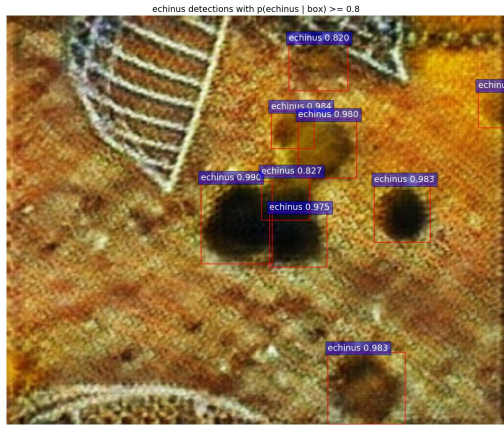




(a)



(b)



(c)



(d)



(e)



(f)

Figure 2: The bounding boxes in the same picture which is enhanced by six kinds of image enhancement method.

		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
图像增强		0.8	0.529	0.842	0.324	0.772	0.617	0.65

Figure 3: Output results of several methods.

During the model testing, our team spend time modifying test\_list.txt which includes 581 items. Then we use Algorithm 4.2 to run a train process to get the result of mAP value.

```

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 Training of FGFA Algorithm

There are some difficulties running the demo for us because of little issue reference in the github and blogs. So senior student and I try our best to solve the problems as 4.3.

```

1 In file included from src/operator/tensor/././sort_op.h:85:0,
2 from src/operator/tensor/./indexing_op.h:24,
3 from src/operator/tensor/indexing_op.cu:8:
4 src/operator/tensor/./././sort_op-inl.cuh:10:44:
5 fatal error: cub/device/device_radix_sort.cuh: No such file or directory
6 #include <cub/device/device_radix_sort.cuh>
7 ^
8 compilation terminated.

```

```
| 9  make: *** [build/src/operator/tensor/indexing_op_gpu.o] Error 1
```

We find the error about cub and then modify some codes to compile. Finally, we get over these difficulty but still have some to deal with.

## 5 Plan

**Objective:** Finish open ROV equipment debugging and FGFA training.

**Deadline:** 2018.08.25

2018.08.13—2018.08.19 Finish sequence models courses.

2018.08.20—2018.08.26 Finish evaluation of Faster RCNN.

2018.08.27—2018.09.02 Finish preparation of 2018URPC.

## 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](#)