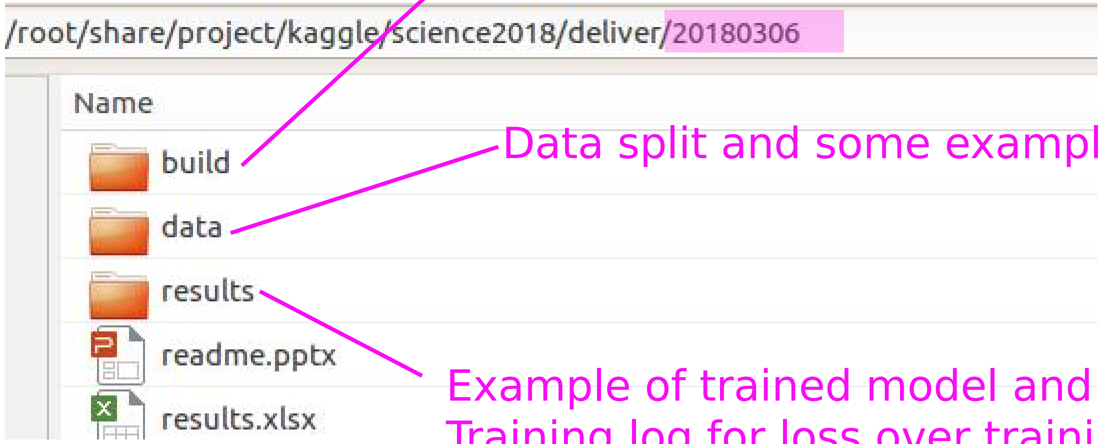
0. Deilvery

- What is in the downloaded delivery?



/root/share/project/kaggle/science2018/deliver/20180306

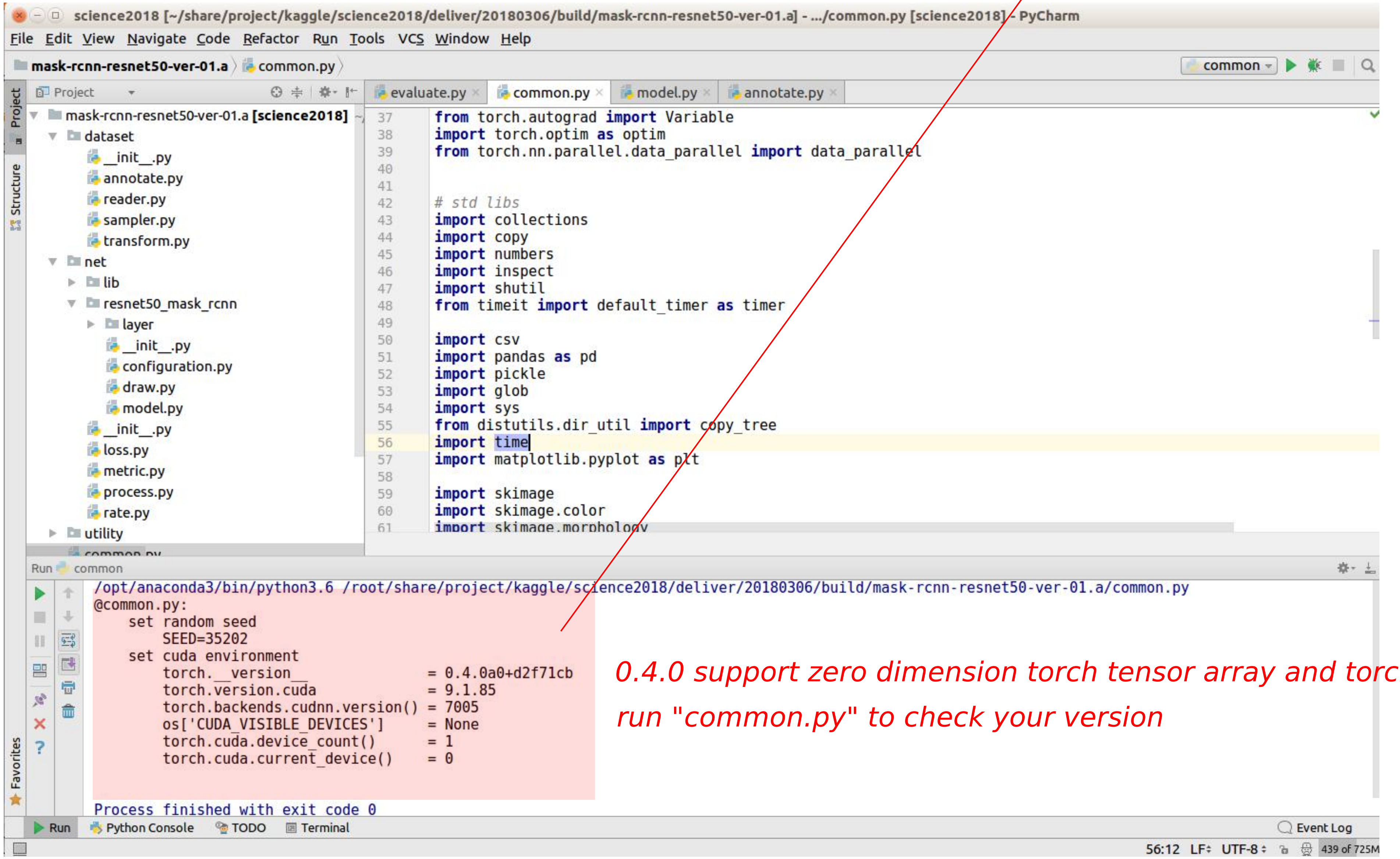
Python source code

Data split and some example of converted format.

Example of trained model and submission csv.
Training log for loss over training iterations

1. Software Setup (version: mask-rcnn-resnet50-ver-01.a)

- python 3.6 / Pycharm as IDE
- pytorch 0.4.0 (please build from source. Anaconda/Pip installation is only up to 0.30)



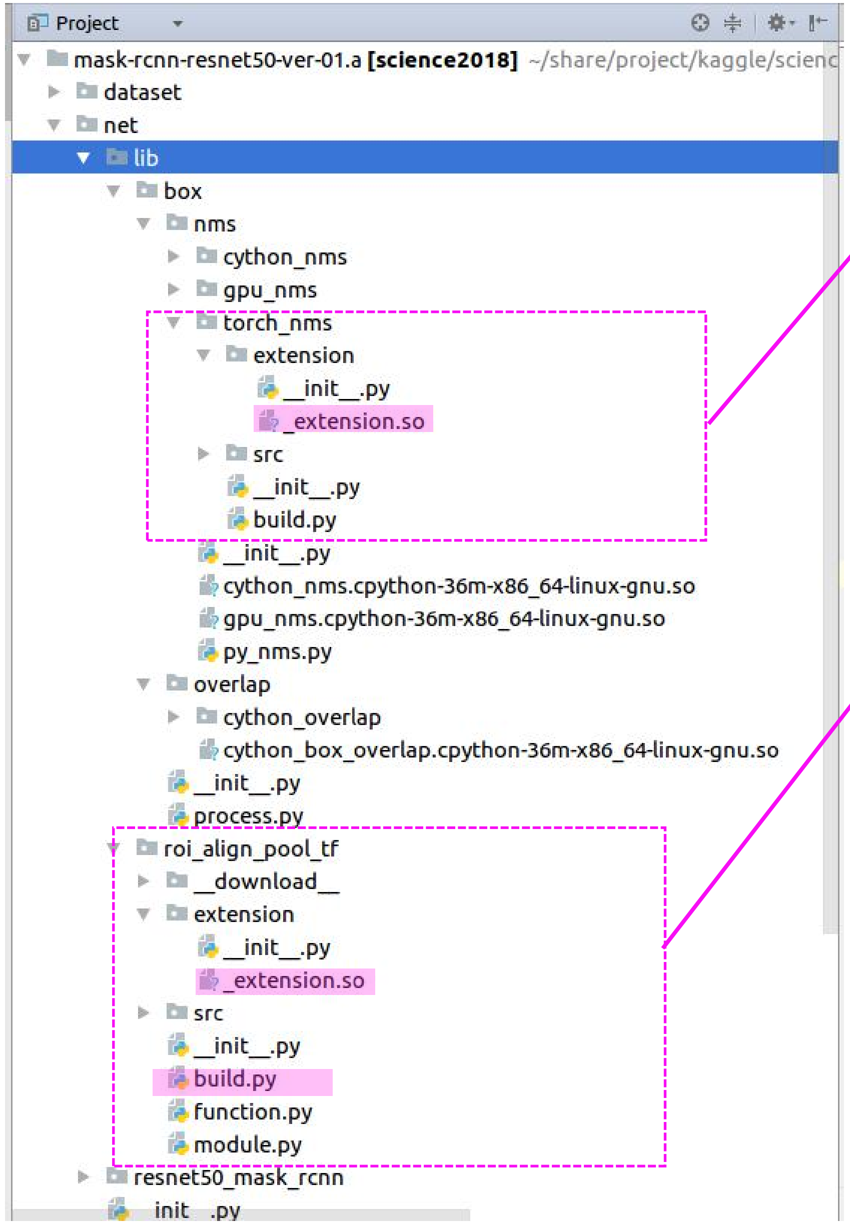
```python
from torch.autograd import Variable
import torch.optim as optim
from torch.nn.parallel.data_parallel import data_parallel


# std libs
import collections
import copy
import numbers
import inspect
import shutil
from timeit import default_timer as timer

import csv
import pandas as pd
import pickle
import glob
import sys
from distutils.dir_util import copy_tree
import time
import matplotlib.pyplot as plt

import skimage
import skimage.color
import skimage.morphology
```

```
/opt/anaconda3/bin/python3.6 /root/share/project/kaggle/science2018/deliver/20180306/build/mask-rcnn-resnet50-ver-01.a/common.py
@common.py:
    set random seed
        SEED=35202
    set cuda environment
        torch.__version__              = 0.4.0a0+d2f71cb
        torch.version.cuda             = 9.1.85
        torch.backends.cudnn.version() = 7005
        os['CUDA_VISIBLE_DEVICES']     = None
        torch.cuda.device_count()      = 1
        torch.cuda.current_device()    = 0

Process finished with exit code 0
```

0.4.0 support zero dimension torch tensor array and torch scalar

run "common.py" to check your version

- Software setup. You must build the torch *.so lib for your system



```
mask-rcnn-resnet50-ver-01.a [science2018] ~/share/project/kaggle/scienc
  dataset
  net
    lib
      box
        nms
          cython_nms
          gpu_nms
          torch_nms
            extension
              __init__.py
              _extension.so
            src
            __init__.py
            build.py
          __init__.py
          cython_nms.cpython-36m-x86_64-linux-gnu.so
          gpu_nms.cpython-36m-x86_64-linux-gnu.so
          py_nms.py
        overlap
          cython_overlap
          cython_box_overlap.cpython-36m-x86_64-linux-gnu.so
        __init__.py
        process.py
      roi_align_pool_tf
        __download__
        extension
          __init__.py
          _extension.so
        src
        __init__.py
        build.py
        function.py
        module.py
    resnet50_mask_rcnn
    __init__.py
```

1. lib/box/nms/torch_nms/extension/_extension.so

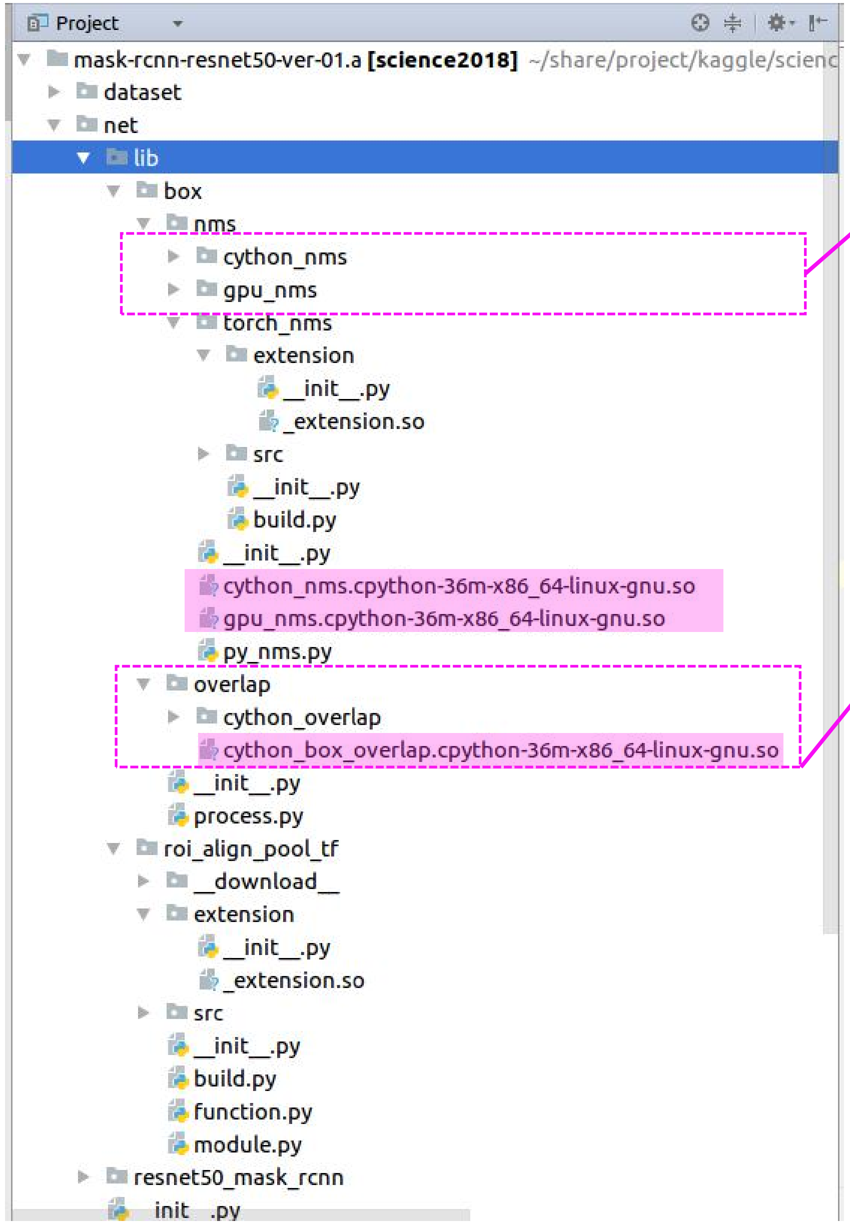
```
>> /usr/local/cuda-9.1/bin/nvcc -c -o nms_kernel.cu.o
nms_kernel.cu -x cu -Xcompiler -fPIC -arch=sm_52

>> python build
```

2. lib/roi_align_pool_tf/extension/_extension.so

This is porting of roi align pooling layer from tensorflow implementation. Note that this is not the same as mask-rcnn paper.

```
>> /usr/local/cuda-9.1/bin/nvcc -c -o
crop_and_resize_kernel.cu.o crop_and_resize_kernel.cu -x
cu -Xcompiler -fPIC -arch=sm_52

>> python build
```

- You must build the cython *.so lib for your system



```
Project
▼ mask-rcnn-resnet50-ver-01.a [science2018] ~/share/project/kaggle/scienc
  ▶ dataset
  ▼ net
    ▼ lib
      ▼ box
        ▼ nms
          ▶ cython_nms
          ▶ gpu_nms
          ▼ torch_nms
            ▼ extension
                __init__.py
                _extension.so
            ▶ src
              __init__.py
              build.py
            __init__.py
            cython_nms.cpython-36m-x86_64-linux-gnu.so
            gpu_nms.cpython-36m-x86_64-linux-gnu.so
            py_nms.py
        ▼ overlap
          ▶ cython_overlap
            cython_box_overlap.cpython-36m-x86_64-linux-gnu.so
          __init__.py
          process.py
      ▼ roi_align_pool_tf
        ▶ __download__
        ▼ extension
            __init__.py
            _extension.so
        ▶ src
          __init__.py
          build.py
          function.py
          module.py
    ▶ resnet50_mask_rcnn
      init .py
```

1. lib/box/nms/cython_nms_xxx.so

>> /opt/anaconda3/bin/python3 setup.py build_ext --inplace

2. lib/box/nms/gpu_nms_xxx.so

>> /opt/anaconda3/bin/python3 setup.py build_ext --inplace

To check mns, see "box/process.py" run_check_nms()

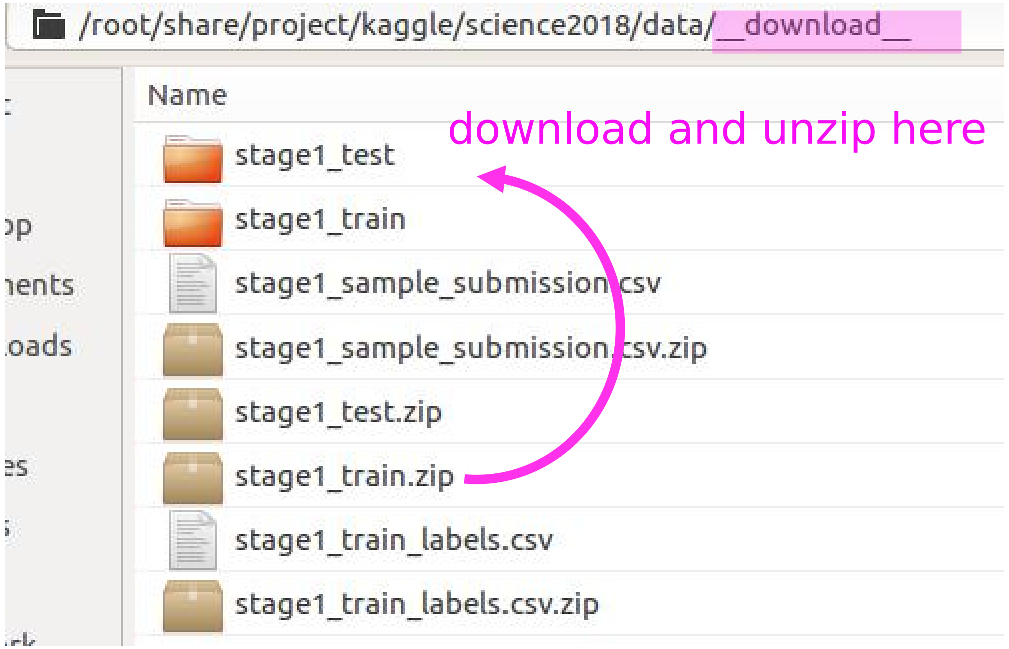
```
box.py: calling main function ...
gpu_nms    : [5, 55, 37, 27, 35, 20, 0, 45, 52, 19, 29, 24, 4, 11]
cython_nms : [5, 55, 37, 27, 35, 20, 0, 45, 52, 19, 29, 24, 4, 11]
torch_nms  : [ 5 55 37 27 35 20  0 45 52 19 29 24  4 11]
sucess!

Process finished with exit code 0
```

3. lib/box/overlap/cython_box_overlap_xxx.so

>> /opt/anaconda3/bin/python3 setup.py build_ext --inplace

2. Data Setup

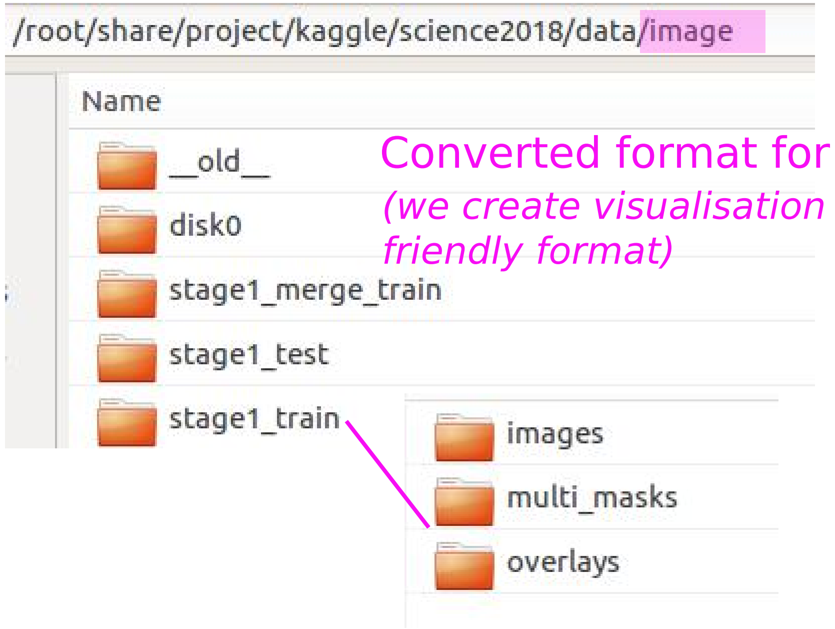
- run the functions at dataset/annotate.py:
  "run_make_train_annotation()" and "run_make_test_annotation()"



/root/share/project/kaggle/science2018/data/__download__

Name
- stage1_test
- stage1_train
- stage1_sample_submission.csv
- stage1_sample_submission.csv.zip
- stage1_test.zip
- stage1_train.zip
- stage1_train_labels.csv
- stage1_train_labels.csv.zip

download and unzip here

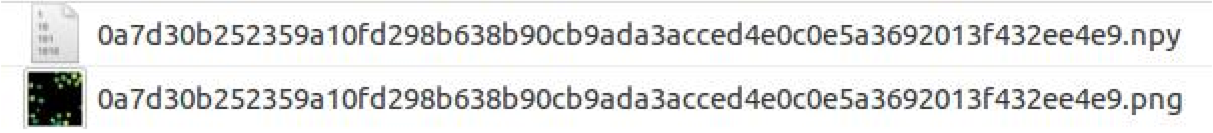


/root/share/project/kaggle/science2018/data/image

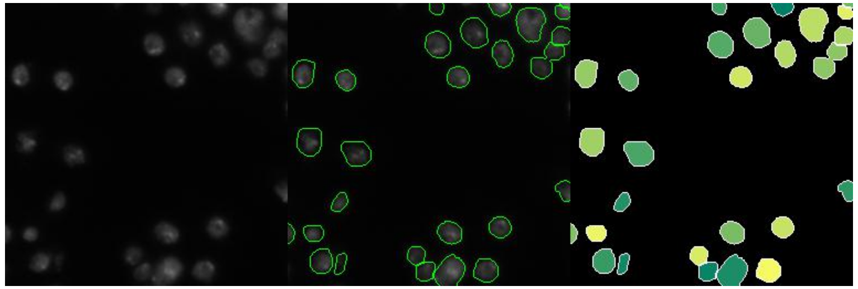
Name
- __old__
- disk0
- stage1_merge_train
- stage1_test
- stage1_train
  - images
  - multi_masks
  - overlays

Converted format for training

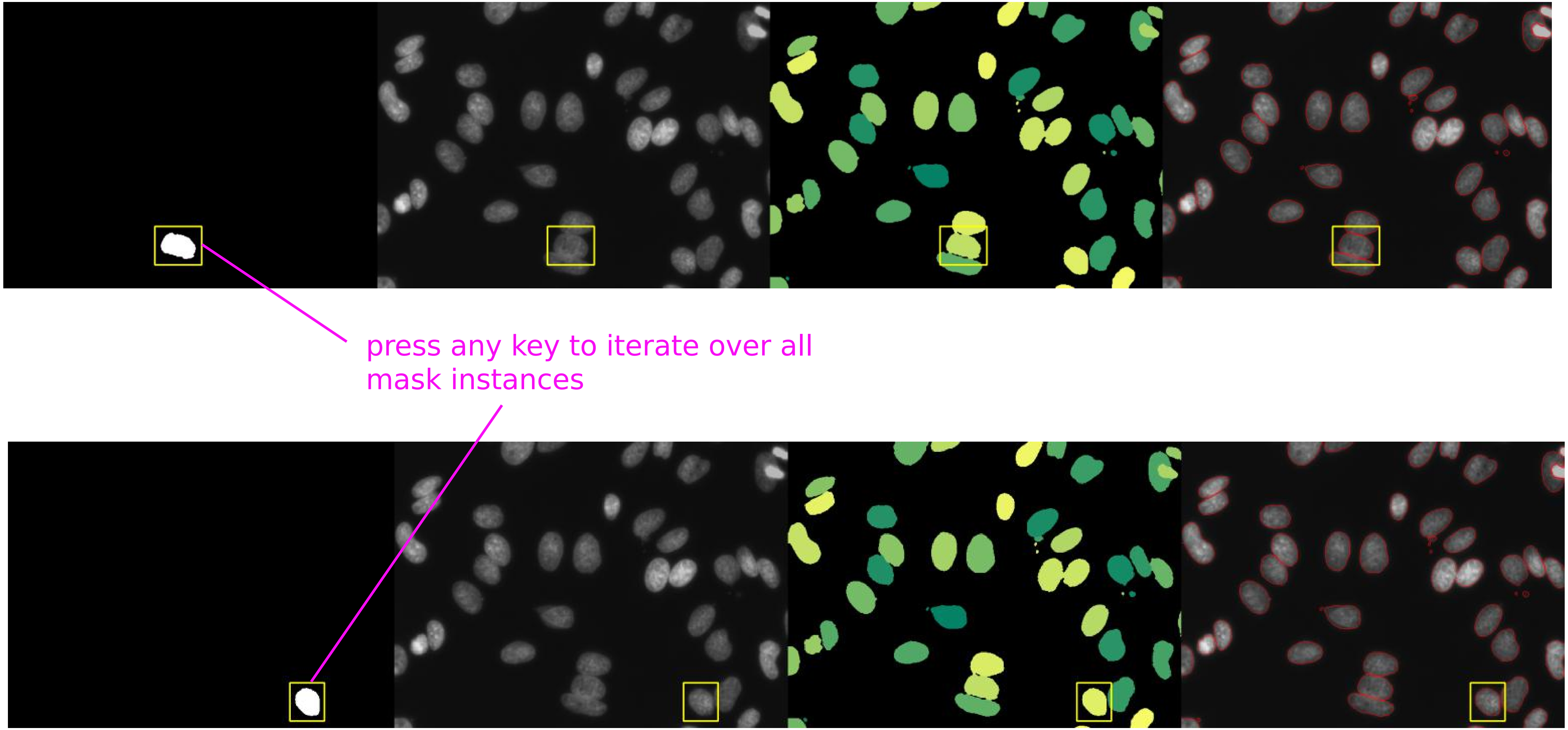
(we create visualisation and more user friendly format)



0a7d30b252359a10fd298b638b90cb9ada3acced4e0c0e5a3692013f432ee4e9.npy
0a7d30b252359a10fd298b638b90cb9ada3acced4e0c0e5a3692013f432ee4e9.png



- run the function "run_check_dataset_reader() at dataset/reader.py to see if your data is setu correctly



press any key to iterate over all mask instances

4. Make a submission with the given trained model (00016500_model.pth)

- This will test if the mask rcnn can run correctly (at inference)

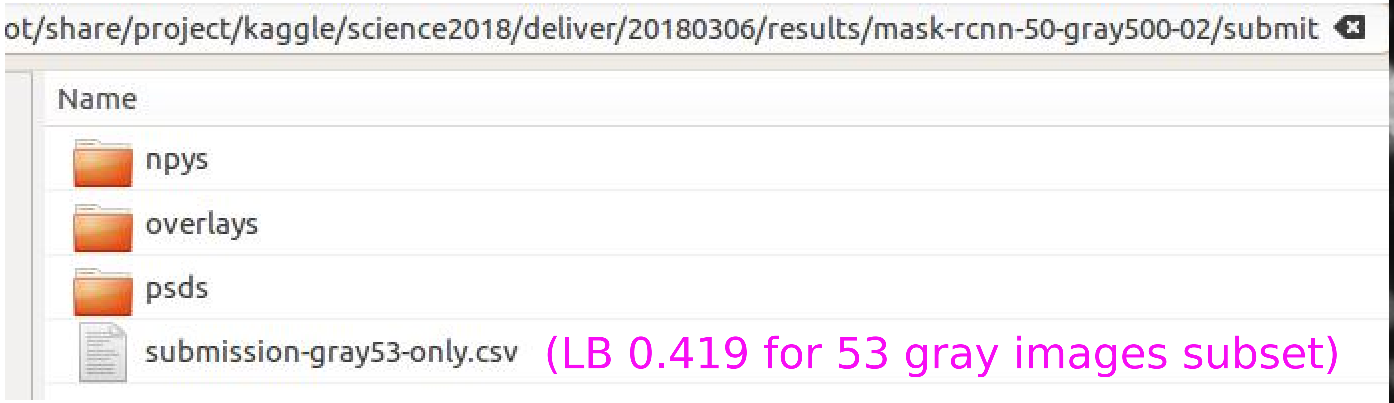
- Run the functions of "submit.py":

```
run_submit()
run_npy_to_sumbit_csv()
```

this only make prediction and save results as images and npy.

this read the npy and do post processing to make submission csv

- Example results (see folder "results/mask-rcnn-50-gray500-02/submit")



ot/share/project/kaggle/science2018/deliver/20180306/results/mask-rcnn-50-gray500-02/submit

Name

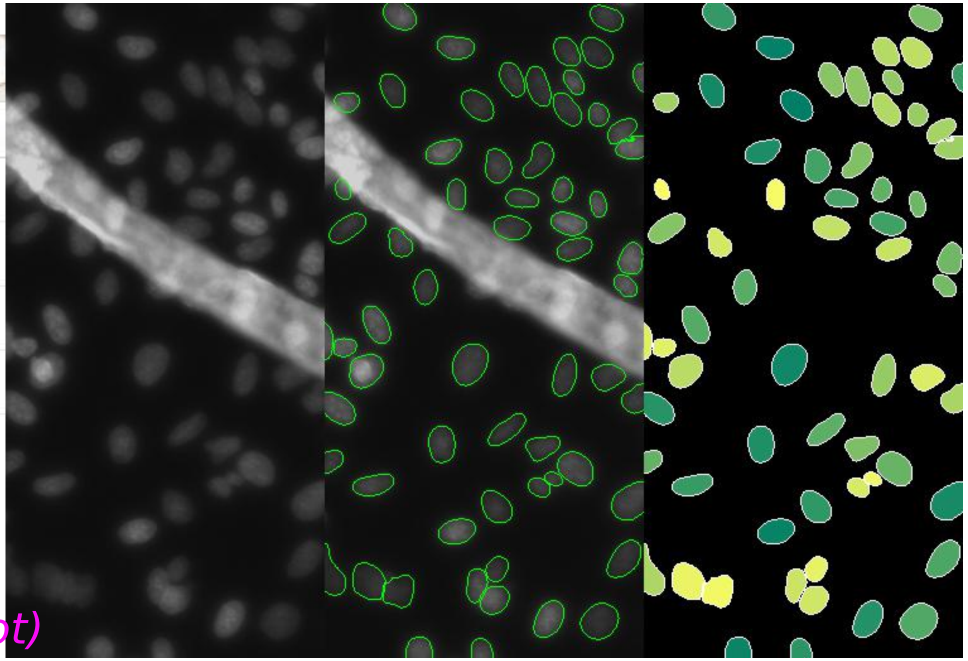
npys

overlays

psds

submission-gray53-only.csv

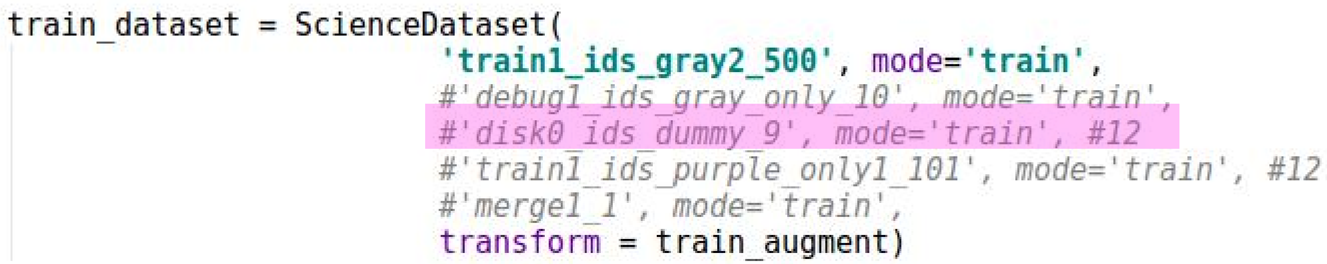
(LB 0.419 for 53 gray images subset)



*** you need not submit.*
Just make your csv file and see if your file is the same as mine or not)

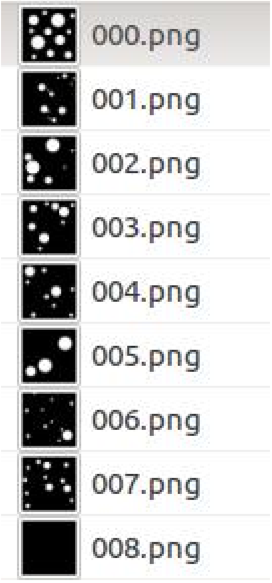
5. Train a model on the dummy images

- Run function "run_train()" of train_0.py
- Set both train and validation split to 'disk0_ids_dummy_9'



```
train_dataset = ScienceDataset(
                         'train1_ids_gray2_500', mode='train',
                         #'debug1_ids_gray_only_10', mode='train',
                         #'disk0_ids_dummy_9', mode='train', #12
                         #'train1_ids_purple_only1_101', mode='train', #12
                         #'merge1_1', mode='train',
                         transform = train_augment)
```

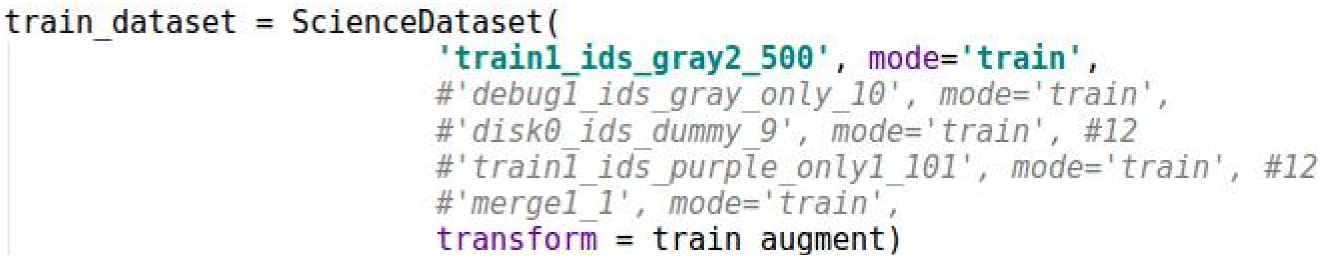
- This is simple dummy data. You should be able to train all loss to 0!
  You can see some training visualisation: rpn precision, rcnn precision and mask precision



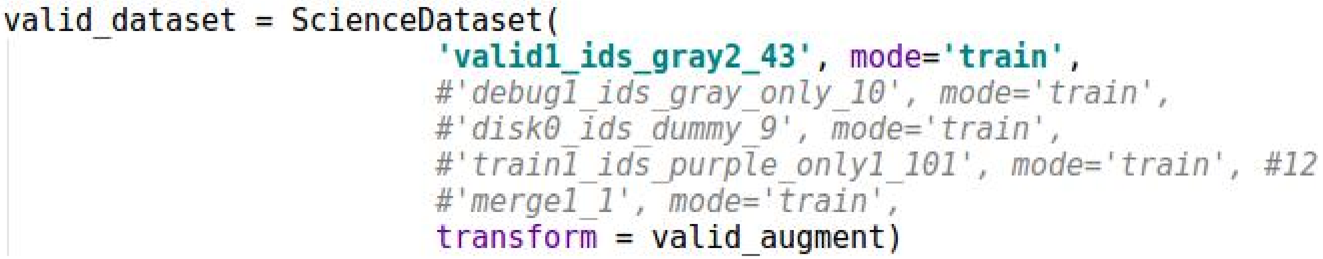
000.png
001.png
002.png
003.png
004.png
005.png
006.png
007.png
008.png

5. Train a model on the gray train images

- Set both train and validation split as follows



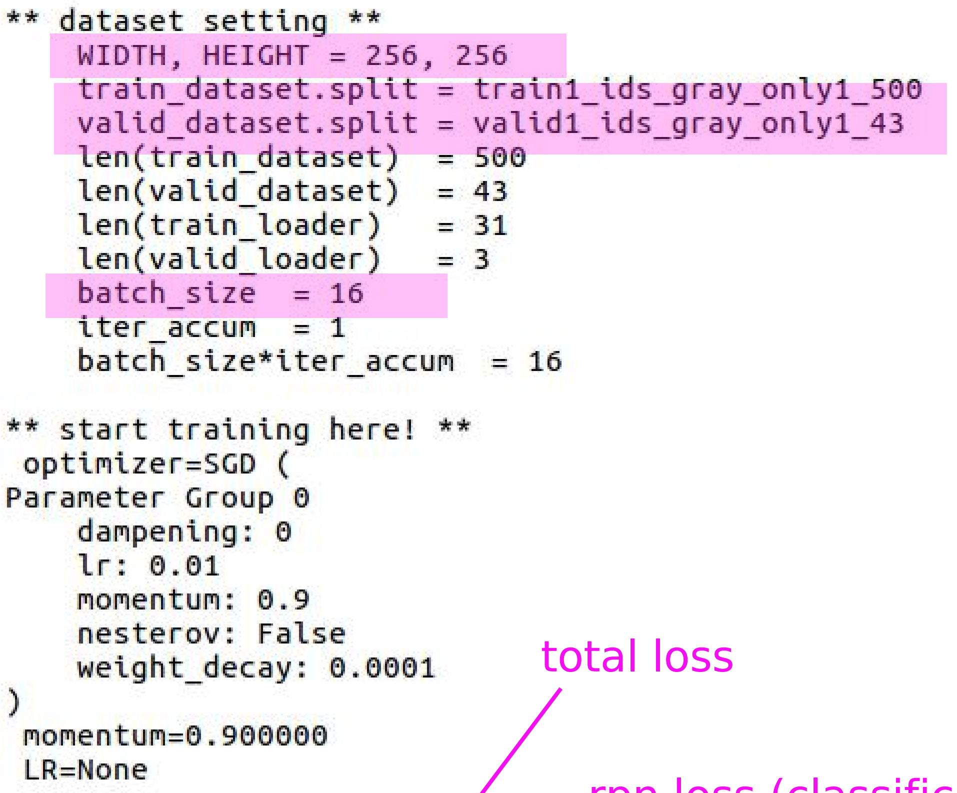
```
train_dataset = ScienceDataset(
                                'train1_ids_gray2_500', mode='train',
                                #'debug1_ids_gray_only_10', mode='train',
                                #'disk0_ids_dummy_9', mode='train', #12
                                #'train1_ids_purple_only1_101', mode='train', #12
                                #'merge1_1', mode='train',
                                transform = train_augment)
```



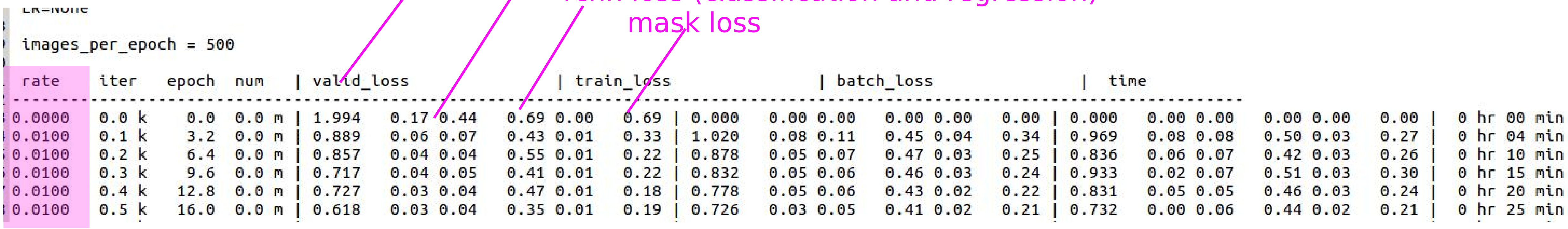
```
valid_dataset = ScienceDataset(
                                'valid1_ids_gray2_43', mode='train',
                                #'debug1_ids_gray_only_10', mode='train',
                                #'disk0_ids_dummy_9', mode='train',
                                #'train1_ids_purple_only1_101', mode='train', #12
                                #'merge1_1', mode='train',
                                transform = valid_augment)
```

- this is used to produce model **00016500_model.pth**

- An example of training loss is given at: "20180306/results/mask-rcnn-50-gray500-02/log.train.txt"



```
** dataset setting **
    WIDTH, HEIGHT = 256, 256
    train_dataset.split = train1_ids_gray_only1_500
    valid_dataset.split = valid1_ids_gray_only1_43
    len(train_dataset)  = 500
    len(valid_dataset)  = 43
    len(train_loader)   = 31
    len(valid_loader)   = 3
    batch_size  = 16
    iter_accum  = 1
    batch_size*iter_accum  = 16

** start training here! **
 optimizer=SGD (
Parameter Group 0
    dampening: 0
    lr: 0.01
    momentum: 0.9
    nesterov: False
    weight_decay: 0.0001
)
 momentum=0.900000
 LR=None
```



total loss

rpn loss (classification and regression)

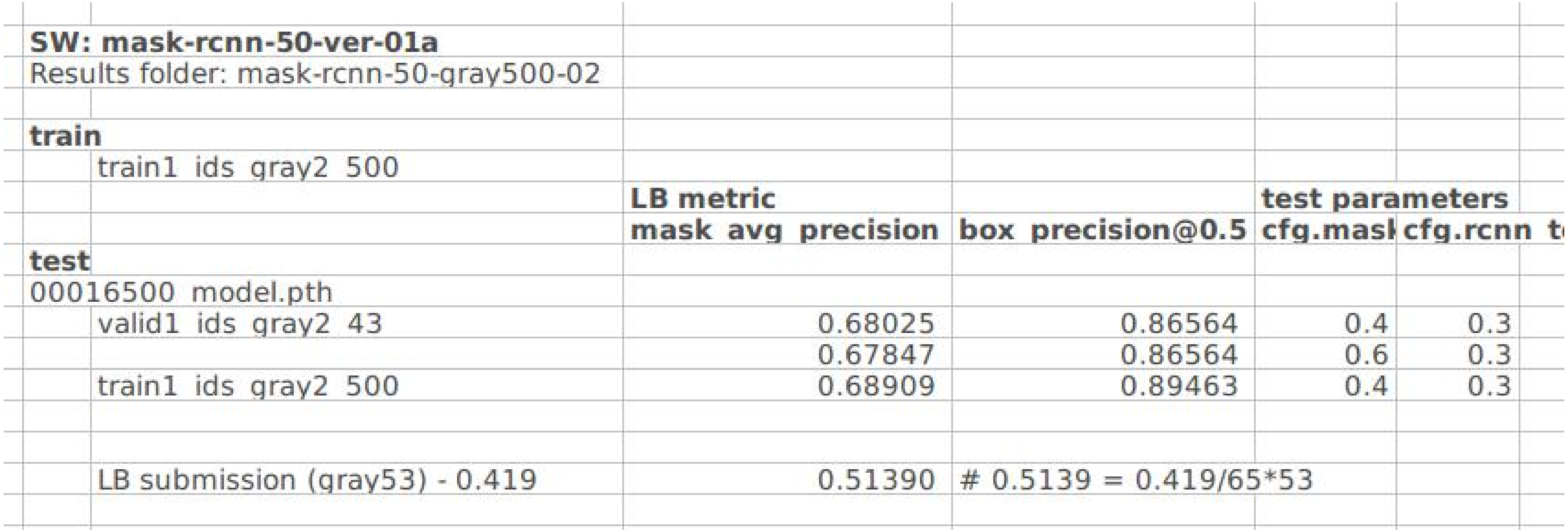
rcnn loss (classification and regression)

mask loss

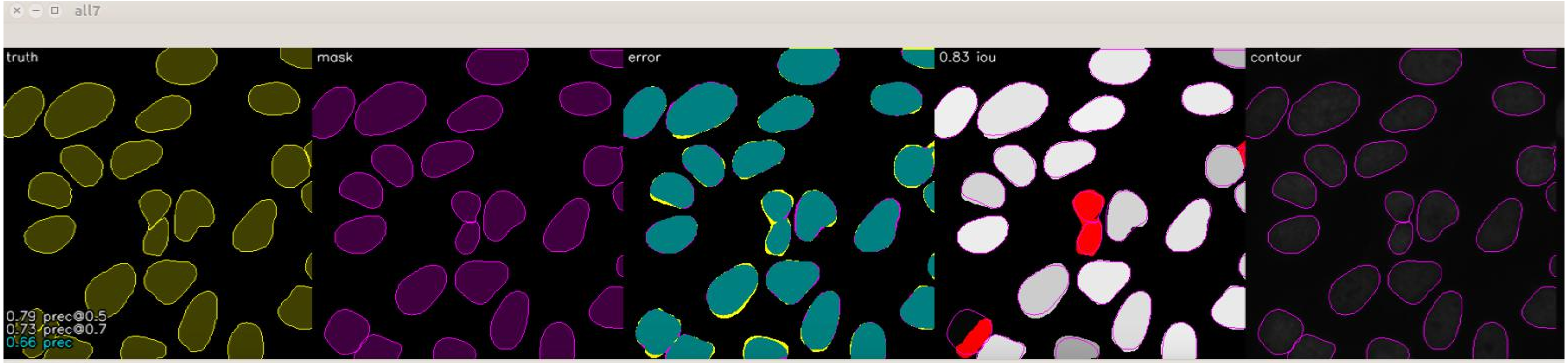
```
LR=None

images_per_epoch = 500

 rate     iter   epoch  num    | valid_loss                | train_loss                | batch_loss                |  time
------------------------------------------------------------------------------------------------------------------------------
0.0000    0.0 k    0.0  0.0 m | 1.994   0.17 0.44   0.69 0.00   0.69 | 0.000   0.00 0.00   0.00 0.00   0.00 | 0.000   0.00 0.00   0.00 0.00   0.00 |  0 hr 00 min
0.0100    0.1 k    3.2  0.0 m | 0.889   0.06 0.07   0.43 0.01   0.33 | 1.020   0.08 0.11   0.45 0.04   0.34 | 0.969   0.08 0.08   0.50 0.03   0.27 |  0 hr 04 min
0.0100    0.2 k    6.4  0.0 m | 0.857   0.04 0.04   0.55 0.01   0.22 | 0.878   0.05 0.07   0.47 0.03   0.25 | 0.836   0.06 0.07   0.42 0.03   0.26 |  0 hr 10 min
0.0100    0.3 k    9.6  0.0 m | 0.717   0.04 0.05   0.41 0.01   0.22 | 0.832   0.05 0.06   0.46 0.03   0.24 | 0.933   0.02 0.07   0.51 0.03   0.30 |  0 hr 15 min
0.0100    0.4 k   12.8  0.0 m | 0.727   0.03 0.04   0.47 0.01   0.18 | 0.778   0.05 0.06   0.43 0.02   0.22 | 0.831   0.05 0.05   0.46 0.03   0.24 |  0 hr 20 min
0.0100    0.5 k   16.0  0.0 m | 0.618   0.03 0.04   0.35 0.01   0.19 | 0.726   0.03 0.05   0.41 0.02   0.21 | 0.732   0.00 0.06   0.44 0.02   0.21 |  0 hr 25 min
```

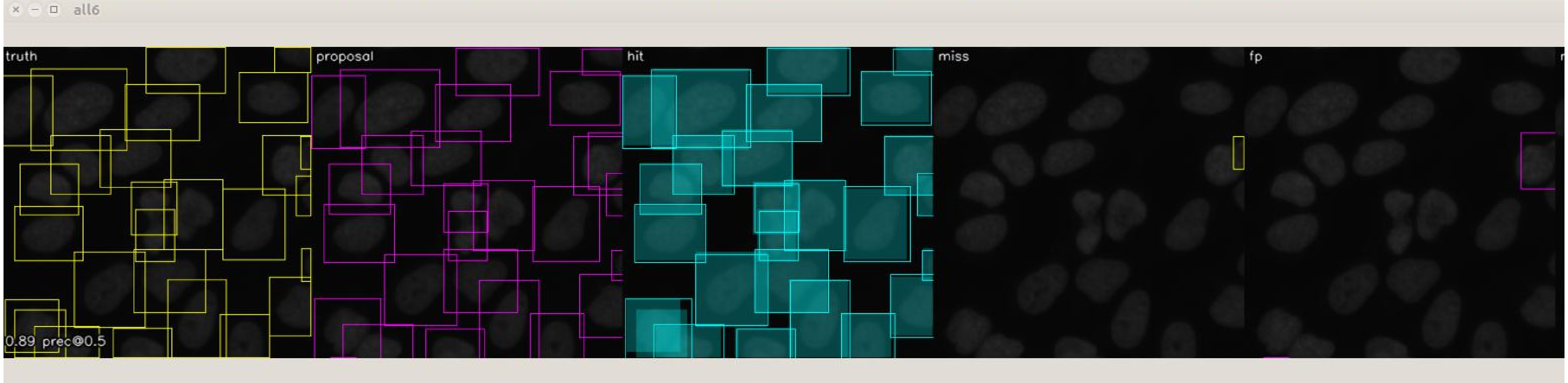
6. Other Evaluation

- Run function "run_evaluate()" of evaluate.py
- You can measure detection box precision at 0.5, and mask average precision from 0.5 to 1.0
- Results of **00016500_model.pth** is given at "results-1.xlsx"



SW: mask-rcnn-50-ver-01a						
Results folder: mask-rcnn-50-gray500-02						
train						
	train1 ids gray2 500					
		LB metric		**test parameters**		
		mask avg precision	**box precision@0.5**	**cfg.masl**	**cfg.rcnn**	**t**
test						
00016500 model.pth						
	valid1 ids gray2 43	0.68025	0.86564	0.4	0.3	
		0.67847	0.86564	0.6	0.3	
	train1 ids gray2 500	0.68909	0.89463	0.4	0.3	
	LB submission (gray53) - 0.419	0.51390	# 0.5139 = 0.419/65*53			


all7
truth
mask
error
0.83 iou
contour
0.79 prec@0.5
0.73 prec@0.7
0.66 prec


all6
truth
proposal
hit
miss
fp
0.89 prec@0.5