Training Report

Jeff Gui

1/25/2021

Aim

The project aims at training a neural network which is able to,

- segment cell nucleus from PCNA signal,
- identify cell cycle phases according to PCNA signal features.

Framework

The baseline ResNet101-FPN_3x_maskRCNN model is obtained from detectron2 project. The model has been trained on COCO dataset with benchmark mask AP = 38.6, box AP = 42.9.

Schedule

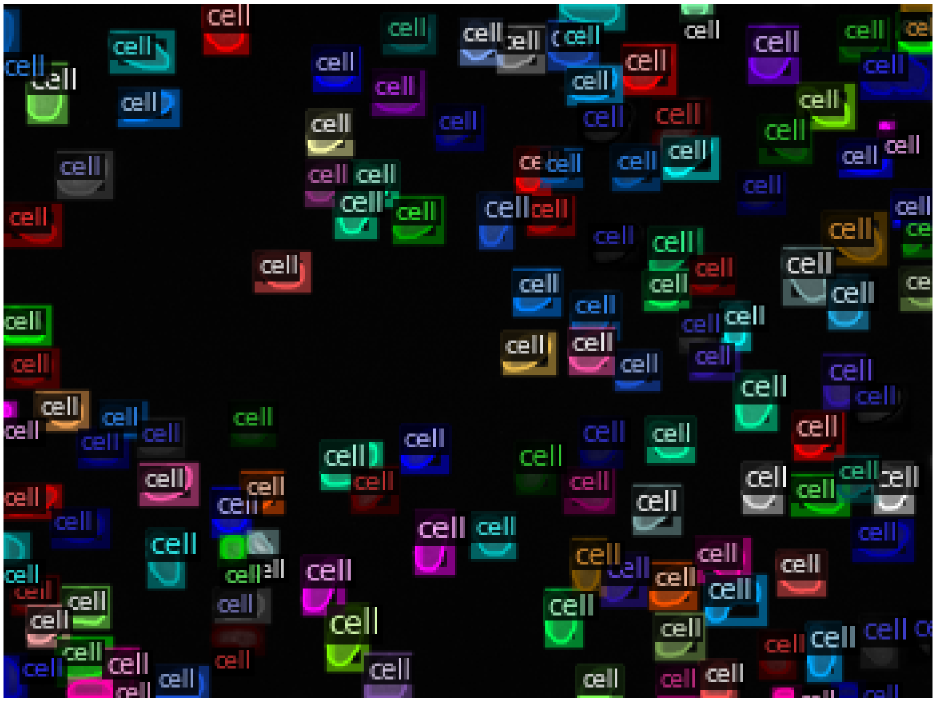
1. The model was first trained on one of Kaggle nucleus segmentation contests dataset containing common fluorescent microscopy images and immunohistochemistry images to identify cell nucleus. There were 670 images in total with various size.
   Augmentation: None, **+ Class number**: 1, **Weight decay**: Yes, **Learning rate warmup**: Yes
2. Final output model from step 1 was trained on deepcell dataset with fluroescent images from four cell lines. Since the dataset is in the unit of 30 tracked image frames, totally 2610 images can be grouped into 87 inter-correlated sets. Therefore, random shuffling was used when generating training input and validation data.
   Augmentation: None, **+ Class number**: 1, **Weight decay**: Yes, **Learning rate warmup**: Yes
3. Final outout model from step 2 was trained on 25 in-house labeled PCNA fluroescent images to identify cell nucleus and predict cell cycle phases. Because PCNA stably distributes in the nucleus during G1/G2, resembling nucleus signal in deepcell images and most images from Kaggle dataset, we labeled images with the same category ID as in the previous training. S phase and M phase were labeled as two new categories. We noted that during mitosis PCNA leaks into the cytoplasm therefore the signal is weak. Nevertheless, DIC feature of cell rounding during mitosis is obvious. Hence, two additional channels of the input were designed: 1. DIC image; 2. mask via lowered Otsu threshold that captures all true positive regions. See **Results.** for comparsion with single grayscale input.
   Augmentation: Yes, **+ Class number**: 3, **Weight decay**: Yes, **Learning rate warmup**: Yes

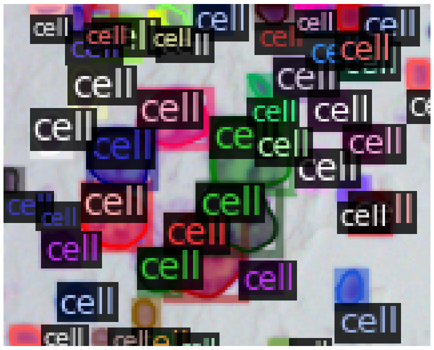
   (see Results. for detailed training configuration)

Results

Kaggle Nucleus

670 images were separated into 570 training set and 100 validation set. After training for 3000 iterations / ~**10 epochs**, total loss converges at `0.72` with `AP50 = 83.46`

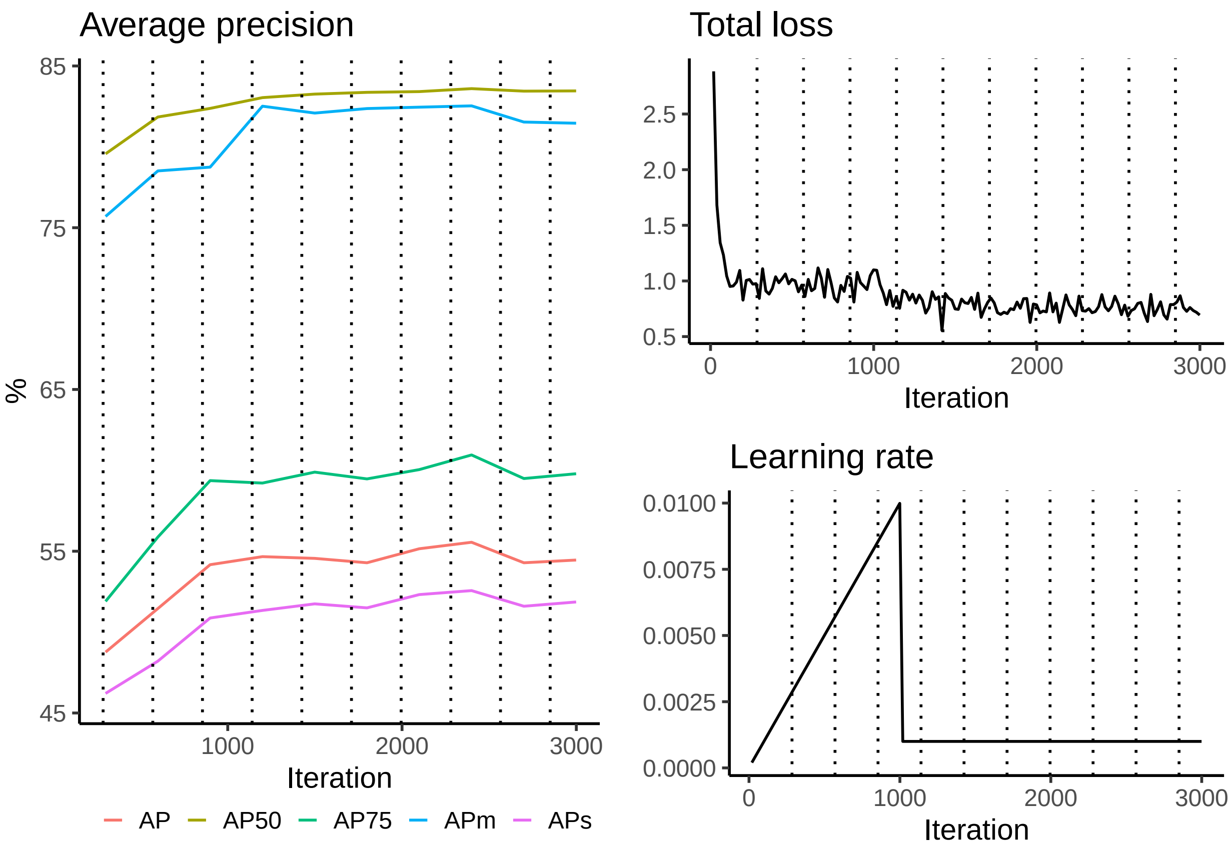




Configurations

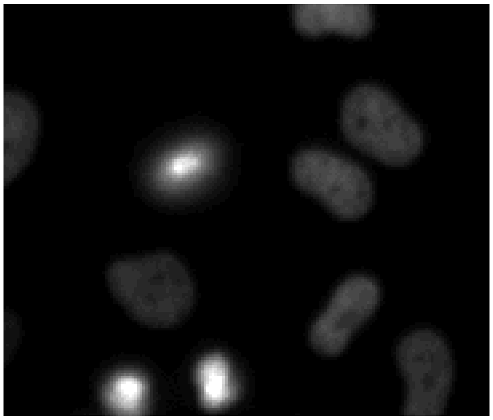
```
cfg.SOLVER.IMS_PER_BATCH = 2
cfg.SOLVER.BASE_LR = 0.01
cfg.SOLVER.WEIGHT_DECAY = 0.0001
cfg.SOLVER.WEIGHT_DECAY_NORM = 0.0
cfg.SOLVER.GAMMA = 0.1
cfg.SOLVER.STEPS = (1000,)
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 256
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 1
```

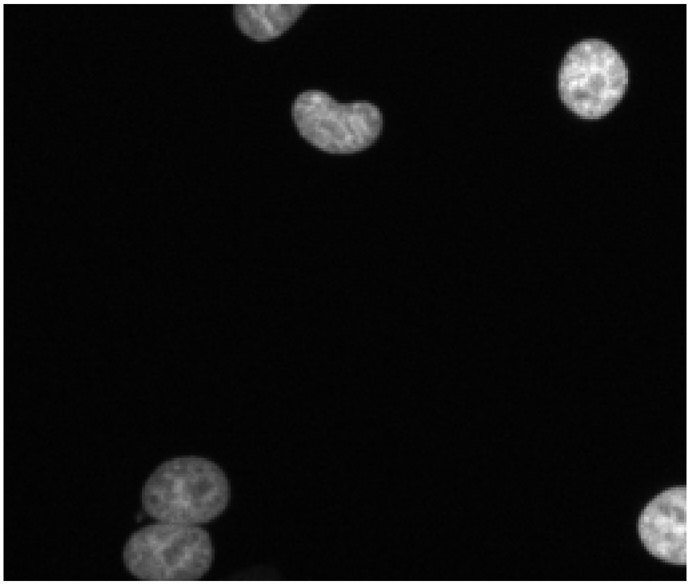
Outputs Outputs generated by detectron2 logging matrix.


Average precision
85
75
65
55
45
%
1000
2000
3000
Iteration
AP
AP50
AP75
APm
APs
Total loss
2.5
2.0
1.5
1.0
0.5
0
1000
2000
3000
Iteration
Learning rate
0.0100
0.0075
0.0050
0.0025
0.0000
0
1000
2000
3000
Iteration

Deepcell Nucleus

2610 images were separated into 2220 training set and 390 validation set with random shuffling to avoid inter-correlation among images. After training for 1800 iterations / ~**13 epochs**, total loss converges at `0.60` with `AP50 = 95.50`

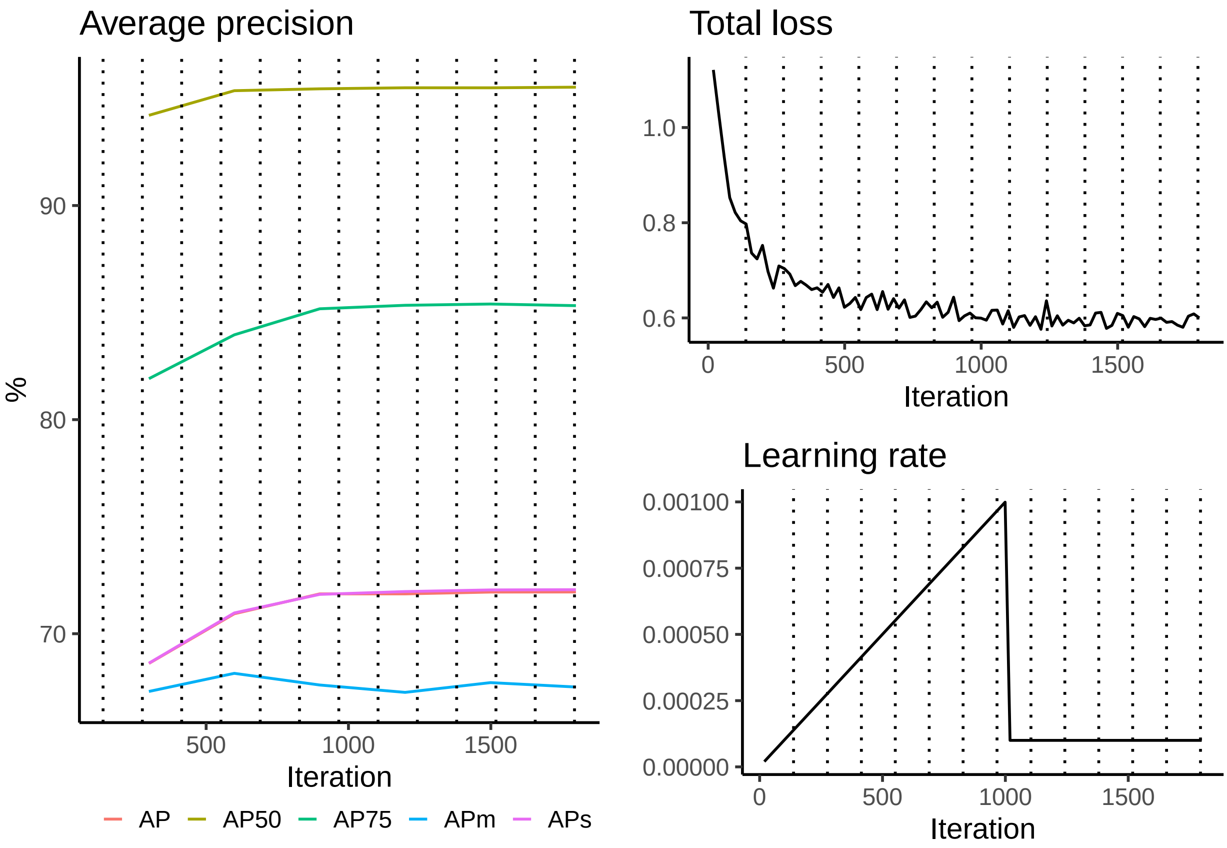




Configurations

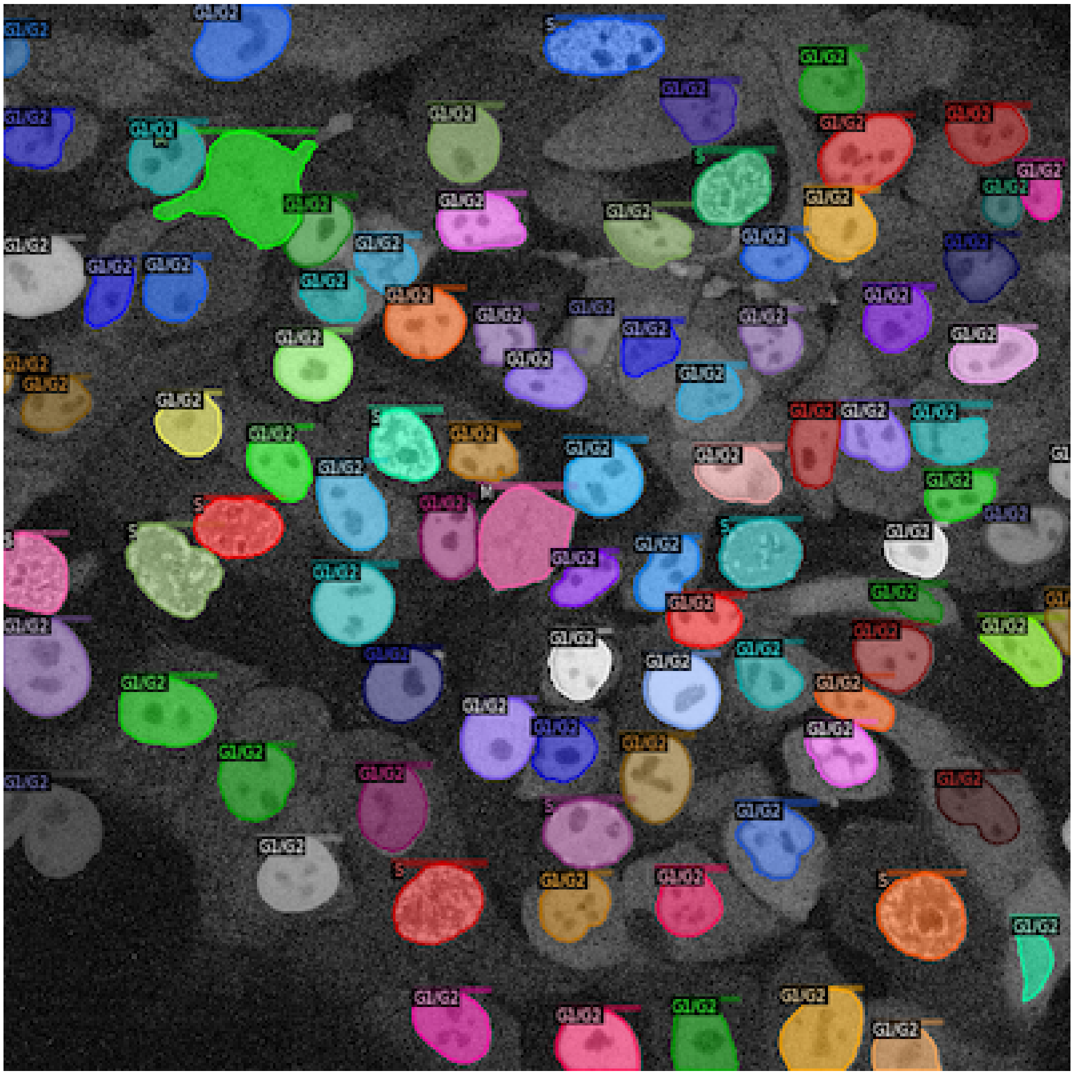
```
cfg.SOLVER.IMS_PER_BATCH = 16
cfg.SOLVER.BASE_LR = 0.001
cfg.SOLVER.WEIGHT_DECAY = 0.0001
cfg.SOLVER.WEIGHT_DECAY_NORM = 0.0
cfg.SOLVER.GAMMA = 0.1
cfg.SOLVER.STEPS = (1000,)
cfg.SOLVER.MAX_ITER = 1800
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 256
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 1
```

Outputs Outputs generated by detectron2 logging matrix.

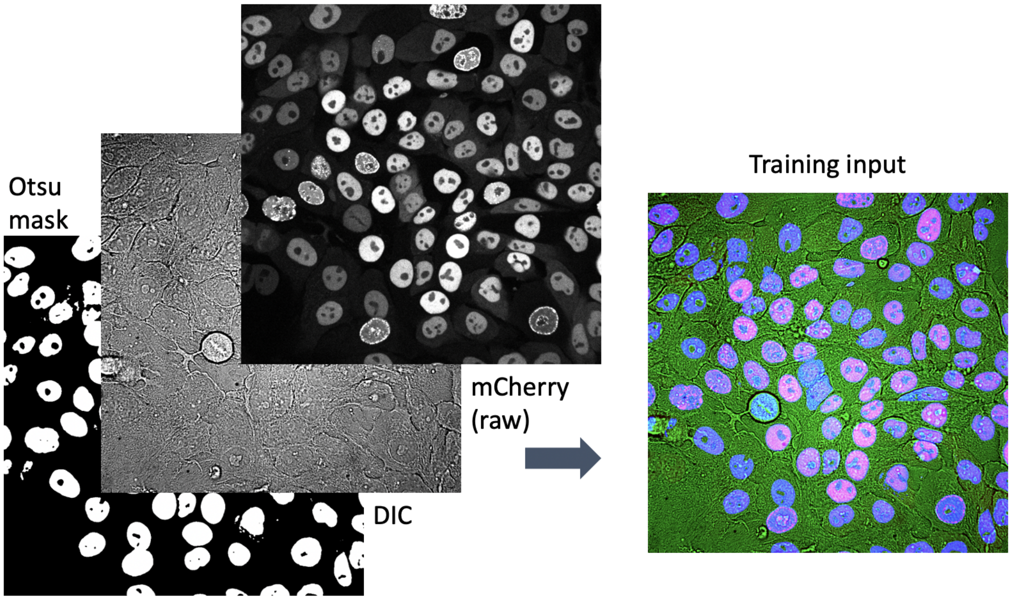

Average precision
%
90
80
70
500
1000
1500
Iteration
AP
AP50
AP75
APm
APs
Total loss
1.0
0.8
0.6
0
500
1000
1500
Iteration
Learning rate
0.00100
0.00075
0.00050
0.00025
0.00000
0
500
1000
1500
Iteration

In-house PCNA Nucleus

25 fluorescent images were used as the training set. Random cropping and flipping augmentation was applied. Unlike previous two steps, detection class is sub-divided into G1/G2, S and M cell cycle phases. After training for 10000 iterations / ~**400 epochs**, total loss converges at `1.38`.



Expanding gray scale input with DIC and Otsu mask channels significantly enhances learning performance with total loss converges at `0.25`.

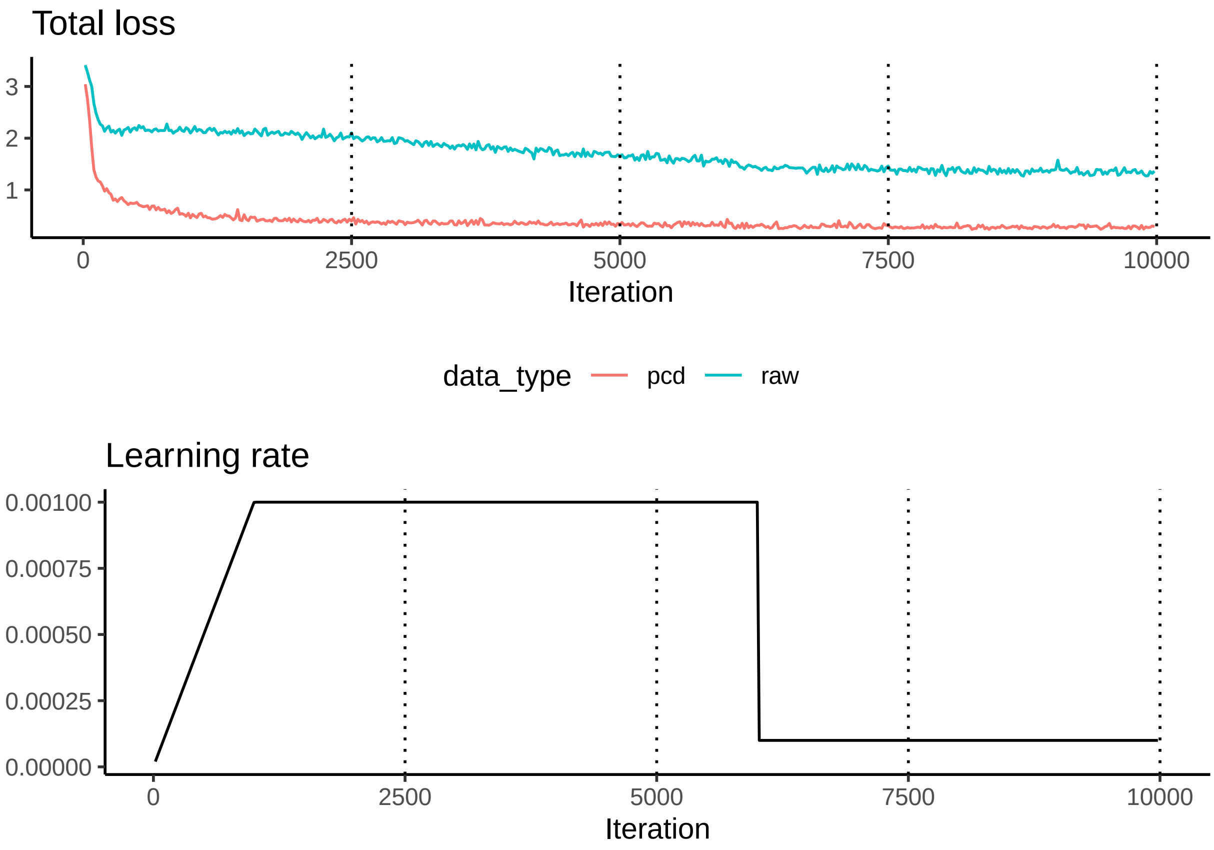


Configurations

```
cfg.SOLVER.IMS_PER_BATCH = 1
cfg.SOLVER.BASE_LR = 0.001
cfg.SOLVER.WEIGHT_DECAY = 0.0001
cfg.SOLVER.WEIGHT_DECAY_NORM = 0.0
cfg.SOLVER.GAMMA = 0.1
cfg.SOLVER.STEPS = (6000,)
cfg.SOLVER.MAX_ITER = 10000
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 256
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 3

# Augmentation
cfg.INPUT.MIN_SIZE_TRAIN = 1000
cfg.INPUT.MAX_SIZE_TRAIN = 1200
cfg.INPUT.MIN_SIZE_TRAIN_SAMPLING = 'choice'
cfg.INPUT.CROP.ENABLED = True
cfg.INPUT.CROP.TYPE = 'relative'
cfg.INPUT.CROP.SIZE = [0.9,0.9]
```

Outputs Outputs generated by detectron2 logging matrix.



Total loss

Learning rate