

Selected Topics in Visual Recognition using Deep Learning Homework 3

Github link of the codes

Project Repository: [VRDL_21au](#)

Reference

Github repository [Detectron2](#) and related pretrained models are used and modified in this project.

Introduction

The project is an experiment of implementing deep learning model to deal with instance segmentation task. The dataset used here is Nuclei dataset, whose images are all composed of densely distributed cell objects, provided by [2018 Data Science Bowl in Kaggle](#). For experimental sake, a subset of it is applied to our model, with 24 images as training data and 6 images as testing data. The main feature is that compared to typical instance classification problem, the resolution of the image is fixed and high (1000x1000), and the instance we're looking for is simple (only cell objects), small and densely distributed (~1,000/images). In this project, we analyze the predictability of several commonly-used models such as Mask-RCNN, Cascade Mask-RCNN, and also experiment the effectiveness of different backbone such as ResNet-50, ResNet-101 and ResNeXt-101.

Data Preprocessing

Some image preprocessing methodology, such as scaling, normalization and data augmentation, is shared with image classification tasks, so here we skip those detailed preprocessing description. For more information, [pytorch documentation](#) or [Detectron2 documentation](#) is the reference.

Another thing is that we do format transfer to label information. The raw label (location of each cell instance) is stored in mask image .png format. We found that due to the dense-object feature, the model requires lots of I/O operations and memory usage for each iteration before training. [The mask rcnn project](#) by default uses this method to preprocess the label information, which is not an efficient way to apply on a dense object detection problem. In

this project, the data is transformed into .json format in [Coco dataset style](#). It is a commonly-used format when training instance segmentation model and shall work well with those interface that Detectron2 or other libraries offer.

```
annotation{
  "id": int,
  "image_id": int,
  "category_id": int,
  "segmentation": RLE or [polygon],
  "area": float,
  "bbox": [x,y,width,height],
  "iscrowd": 0 or 1,
}

categories[{
  "id": int,
  "name": str,
  "supercategory": str,
}]
```

Basic Deep Learning Techniques

We apply transfer learning and learning rate scheduling techniques, by Detectron2 default settings, to our instance segmentation framework. Pretrained ResNet-50, ResNet-101 and ResNeXt-101 models are used as our initial backbone weights to utilize the domain knowledge learned from other tasks. For the scheduling part, StepLR is used so that smaller learning rate will be used in the last few epochs.

Thing should be noticed is that in our case, the training dataset only consists of 24 labeled images, which is relatively small. The training epochs hence should be carefully chosen to avoid overfitting issue. We've tried an range from 500 to 10,000 epochs and found that 500 epochs (i.e. 12,000 iterations) is enough here.

[Mask RCNN](#), [ResNeXt Backbone](#) and [Cascade RCNN](#)

Mask-RCNN, proposed in 2017 ICCV, is a deep learning model that deals with instance segmentation problem. Regarded as the successor of Faster-RCNN, Mask-RCNN is a region-based convolutional neural network that mainly consists of RCNN, RPN parts. We apply R50, R101 Mask RCNN framework by finetuning pretrained model with Nuclei dataset, and examine its capability of detecting cell objects. Besides, since the cell instance is usually small and dense, smaller anchor size is used here to better fit the possible region proposal. Also, according to the paper **Aggregated Residual Transformations for Deep Neural Networks** by **Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He**, a modified ResNet model

called ResNeXt is used to replace the backbone of mask-rcnn too. It is an multi-branch architecture that could capture more feature than ResNet in general, and we found that ResNeXt could largely improve the performance of detecting nuclei objects.

Another variants of rcnn model proposed in 2018 CVPR, Cascade Mask RCNN, is tested in this project. The main feature is that it is a muti-stage object detection architecture which consists of a sequence of detectors to be sequentially more selective against close false positives objects. In experiment result part, we compare the predictability of this with other models.

Experiment Result

We summarize the mAP and inference speed of all the models mentioned above, and run time is evaluated on Nvidia Tesla K80 GPU offered by Google Colab. Inference speed may fluctuate according to the enviornment setting of Colab, hence relative speed may be a better metric.

Model	mAP
Mask RCNN R-50	.232596
Mask RCNN R-101	.227455
Cascade Mask RCNN R-50	.229755
Mask RCNN X-101 small anchor size	.243117

Summary

As we could see above, the basic implementation of Mask RCNN model could achieve a decent mAP, with R50 backbone 23.26%, R101 backbone 22.75% respectively. If we apply Cascade Mask RCNN with R50 as backbone, the model performs almost equal to normal Mask RCNN. However, replacing the R101 with X101 plus using a smaller anchor size, mAP could increase to 24.31%.