Selected Topics in Visual Recognition using Deep Learning
HW4 Instance Segmentation

0856049 吳毓軒

GitHub repository link

Link: https://github.com/yuhsuan1203/VRDL/tree/master/HW4

Reference from GitHub

Link: https://github.com/dbolya/yolact (v1.1) (It has been updated to v1.2 on 12/16)

Introduction

In this homework, we need to train a convolutional neural network to segment the common objects in the images in Tiny PASCAL VOC dataset.

Instance segmentation is the task of identifying object at the pixel level. One of the popular approaches to solve this problem is Mask-RCNN. Mask-RCNN is a two-stage approach that first scans the image and generates proposals and then classifies those ROIs and generates bounding boxes and masks.

In my work, I used the framework, **YOLACT**, proposed by Bolya et al. in 2019. It breaks instance segmentation into two parallel subtasks, including generating a set of prototype masks and predicting mask coefficients per instance. Then we can produce the instance segmentation for the entire image by using these two components.

Methodology

- Data Preprocess
    - Data augmentation (SSDAugmentation)

      A set of data augmentation technique is randomly applied to the input images, such as hue, mirroring the image, cropping, and so on.

- Model Architecture

  Below is the model architecture of YOLACT.

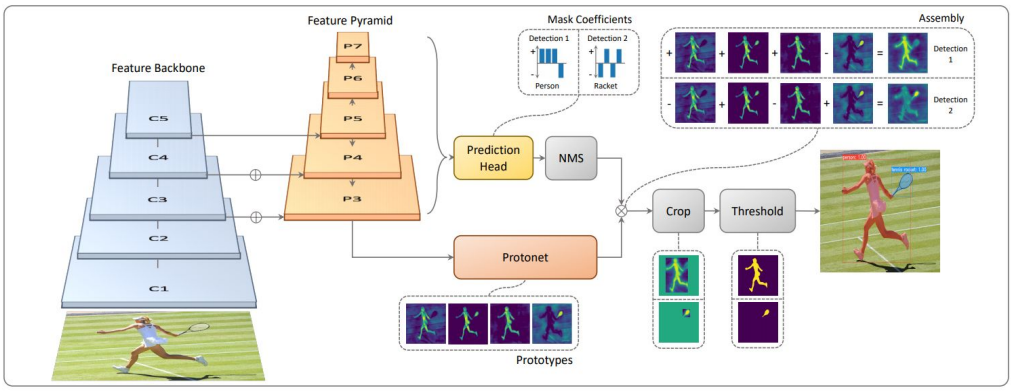

Figure 2: **YOLACT Architecture** Blue/yellow indicates low/high values in the prototypes, gray nodes indicate functions that are not trained, and $k = 4$ in this example. We base this architecture off of RetinaNet [27] using ResNet-101 + FPN.

D Bolya et al. YOLACT Real-time Instance Segmentation 2019
https://arxiv.org/pdf/1904.02689.pdf

Instead of image size of 550 x 550 used in this paper, I used 700 x 700 and ResNet-101 with FPN as the feature backbone detector.

- Hyperparameters
  - total epochs: about 450
  - batch size: 8
  - Optimizer: SGD with learning rate=0.001, momentum=0.9 and decay=0.0005

Summary & Findings

The result is quite good (mAP: 0.47) even training for only about 50 epochs if weights pre-trained on ImageNet is used. I have also tried using ResNet-50 with input image size of 550 x 550 as the feature extractor but ResNet-101 with 700 x 700 is better. Finally, the best result I could get is training for about 450 epochs.

mAP_0.55767_0856049.json

The original purpose in YOLACT is for real-time instance segmentation. Although it worked well enough, it still has localization failure and leakage problem. However, it is still not bad after all for this homework.