Video Streaming and Tracking

Homework 2 - Object Detection

Deadline: 2024/10/21 23:55

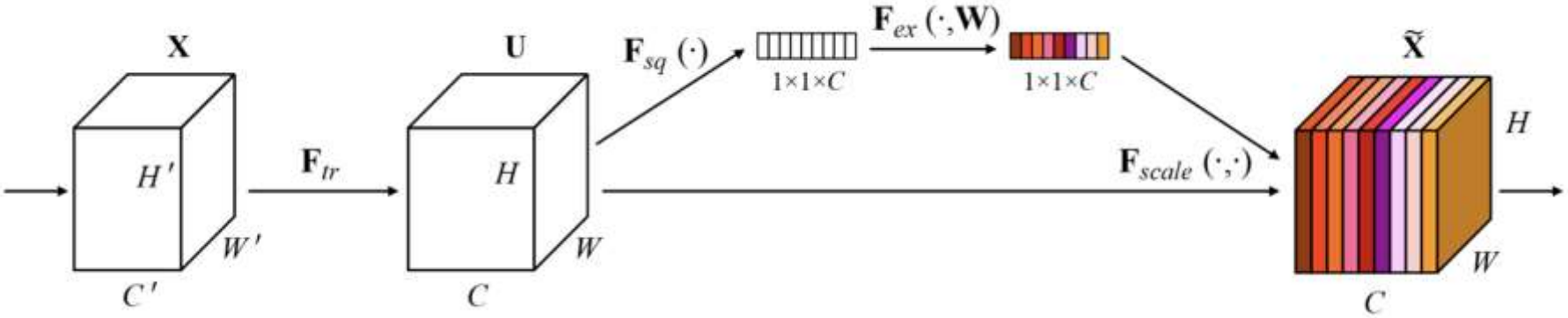
Outline

- Introduction
- Dataset
- Evaluation Metrics
- Hand in Rules
- Grading Policy

Introduction

- Train a neural network to do detection on our own dataset
- The pretrained model weights are available
- Model : object detection algorithms
  - YOLOX-s (we use the official code to set the baseline)
- Add SE module or Inception module to your network
- Framework : PyTorch

Squeeze-and-Excitation Networks



$F_{tr}()$: convolution operation

$F_{sq}()$: avg_pool2d

$F_{ex}()$: Linear → ReLU → Linear → Sigmoid

https://arxiv.org/pdf/1709.01507.pdf

Sample code

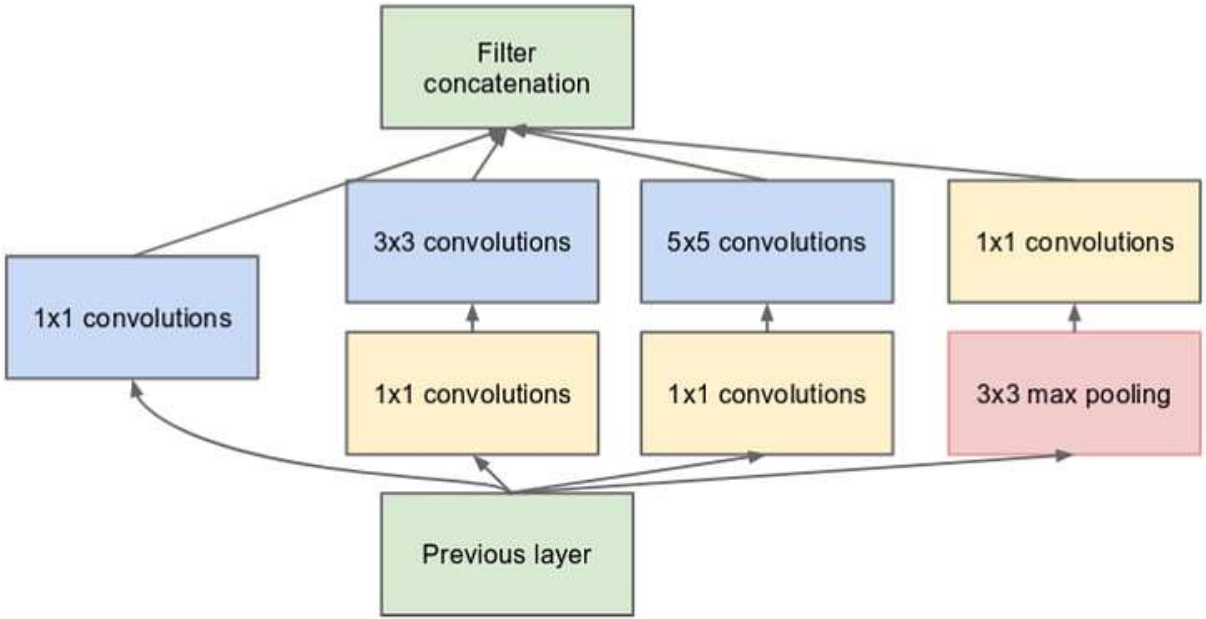
- Conv2d → SELayer → Conv2d

```python
from torch import nn

class SELayer(nn.Module):
    def __init__(self, channel, reduction=16):
        super(SELayer, self).__init__()
        self.avg_pool = nn.AdaptiveAvgPool2d(1)
        self.fc = nn.Sequential(
            nn.Linear(channel, channel // reduction),
            nn.ReLU(inplace=True),
            nn.Linear(channel // reduction, channel),
            nn.Sigmoid()
        )

    def forward(self, x):
        b, c, _, _ = x.size()
        y = self.avg_pool(x).view(b, c)
        y = self.fc(y).view(b, c, 1, 1)
        return x * y
```

Inception Module



https://arxiv.org/pdf/1709.01507.pdf

Sample code

- MaxPool2d → Inception → Inception → MaxPool2d

```
class Inception(nn.Module):
    def __init__(self,in_c,c1,c2,c3,c4):
        super(Inception,self).__init__()
        self.p1 = nn.Sequential(
            nn.Conv2d(in_c,c1,kernel_size=1),
            nn.ReLU(inplace=True)
        )
        self.p2 = nn.Sequential(
            nn.Conv2d(in_c,c2[0],kernel_size=1),
            nn.ReLU(inplace=True),
            nn.Conv2d(c2[0], c2[1], kernel_size=3,padding=1),
            nn.ReLU(inplace=True)
        )
        self.p3 = nn.Sequential(
            nn.Conv2d(in_c, c3[0], kernel_size=1),
            nn.ReLU(inplace=True),
            nn.Conv2d(c3[0], c3[1], kernel_size=5,padding=2),
            nn.ReLU(inplace=True)
        )
        self.p4 = nn.Sequential(
            nn.MaxPool2d(kernel_size=3,stride=1,padding=1),
            nn.Conv2d(in_c,c4,kernel_size=1),
            nn.ReLU(inplace=True)
        )
    def forward(self, x):
        p1 = self.p1(x)
        p2 = self.p2(x)
        p3 = self.p3(x)
        p4 = self.p4(x)
        return torch.cat((p1,p2,p3,p4),dim=1)
```

Dataset

- GTA video dataset
- You only need to detect car (Only one class)
- 1596 training images, labels
- 227 validation images, labels
- 456 testing images

Labels

- ./HW2_ObjectDetection_2024/{train, val}_labels/
- Each row is [class x_center y_center width height] (0~1 range) format (use 0 to represent car)
- Converting the dataset to the YOLOX-compatible format(e.g. COCO)

Open ▾ 8.txt
~/Desktop/HW2_ObjectDetection_2022/train_labels

```
0 0.59453125 0.8930555555555556 0.08802083333333334 0.213888888888
0 0.7171875 0.5055555555555555 0.1125 0.08148148148148149
0 0.5263020833333333 0.4310185185185185 0.0515625 0.076851851851851
0 0.54375 0.3023148148148148 0.03333333333333333 0.060185185185185
0 0.55234375 0.27037037037037037 0.028645833333333332 0.035185185185
0 0.09869791666666666 0.5041666666666667 0.12447916666666667 0.0953
0 0.6130208333333333 0.300462962962963 0.036458333333333336 0.043518
0 0.6296875 0.16805555555555557 0.06875 0.09166666666666666
```

class x_center y_center width height

Evaluation Metrics: mAP (mean Average Precision)

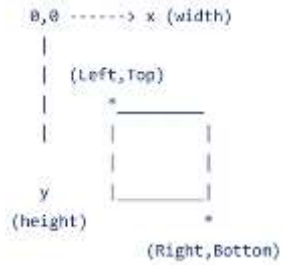
- Most common metric for object detection
- In this HW, mAP defined in the **PASCAL VOC 2012** competition is used
- We will use the following github repo to calculate your score

  https://github.com/rafaelpadilla/Object-Detection-Metrics

- It also contains some explanations about how to calculate it
- We will set IoU threshold greater than 0.85 to calculate the testing score

  E.g. python pascalvoc.py -t 0.85 -gtformat xyrb -detformat xyrb -np

Hand in Rules (1/3)



- You should hand in your result by detecting the testing data through your model
- Format **[class confidence left top right bottom]** in pixel wise (1920x1080)

Open

14.txt
~/Desktop/mAP/result

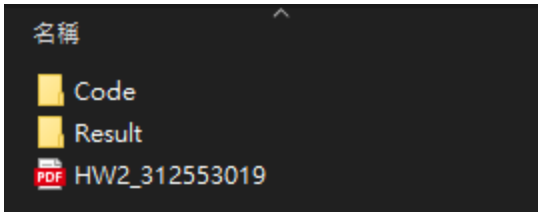
```
0 0.222222 676 215 770 357
0 0.111111 1345 443 1556 584
0 0.999999 753 55 815 118
```

Hand in Rules (2/3)

- Store each detection result in **[image_name].txt**
- You should have a model: with the SE module or Inception module
- Use this model to detect on the testing set
- Submit results in the folder
- You should hand in
  - **Result folder contain testing 456 results**
  - **Folder should be named: Result**

Hand in Rules (3/3)

- Your submission should contain
  - **Result folder contain testing 456 results**
  - **Report (in pdf)**
  - **Code (include your environment and checkpoints). Do not contain dataset and pretrained model weights.**
    - Please submit the code that can generate the prediction results.
- Compress them into one zip file name **HW2_[studentID].zip**



Grading Policy (1/2)

- Model implementation - 65 points
  - Implement on your own or clone from Github then run on our dataset and **pass the baseline** (mAP = **0.8**) by **using the code we provide (set IoU threshold to 0.85) to evaluate on the validation set** – 50 points
  - Add the **SE module** or **Inception module** to your model – 15 points
- Model performance - 15 points
  - The points will determined by the rank with your classmates
  - You can use **SE module or Inception module + other module** to improve your performance
  - You should use **YOLOX-s** as the baseline model
  - Ranking the **average** mAP **on testing set** - you will get **15 / 10 / 5 / 0 points** base on your rank in the class

Grading Policy (2/2)

- Report - 20 points
  - Experiment Setup (Data pre-process, Hyperparameters,...) - 5 points
  - Brief explain your code - 5 points
    - which layer you add modules
    - training / inference command line
  - Screenshot your validation results on your models - 10 points
- If you used code from GitHub, provide reference

Penalty

- Format penalty - 5 points
  - Submit the result in the wrong name, format, etc.
  - Submit the report not in pdf format
- No validation results are shown in the report - 10 points
- Deadline: 2024/10/21 23:59
- Late penalty - 20% per day
  - 1 day => 80%, 2 day => 60%...
- You can use any code from Github, but don’t copy from your classmate!

References

- https://arxiv.org/pdf/1709.01507.pdf
- https://github.com/Megvii-BaseDetection/YOLOX
- https://github.com/rafaelpadilla/Object-Detection-Metrics