Baseline，训练未改动的ssd在voc0712数据集上map=0.7749

具体数据

AP for aeroplane = 0.8207

AP for bicycle = 0.8568

AP for bird = 0.7546

AP for boat = 0.6952

AP for bottle = 0.5019

AP for bus = 0.8479

AP for car = 0.8584

AP for cat = 0.8734

AP for chair = 0.6136

AP for cow = 0.8243

AP for diningtable = 0.7906

AP for dog = 0.8566

AP for horse = 0.8714

AP for motorbike = 0.8403

AP for person = 0.7895

AP for pottedplant = 0.5069

AP for sheep = 0.7767

AP for sofa = 0.7894

AP for train = 0.8623

AP for tvmonitor = 0.7670

Mean AP = 0.7749

~~~~~~~~

Results:

0.821

0.857

0.755

0.695

0.502

0.848

0.858

0.873

0.614

0.824

0.791

0.857

0.871

0.840

0.790

0.507

0.777

0.789

0.862

0.767

0.775

~~~~~~~~

--------------------------------------------------------------

Results computed with the \*\*unofficial\*\* Python eval code.

Results should be very close to the official MATLAB eval code.

[Speed]: 0.0352 s, 28.4 fps

在voc数据集上训练的basic ssd

iter [9010/12000] || Loss: 4.1331

iter [9020/12000] || Loss: 4.6613

iter [9030/12000] || Loss: 4.4244

iter [9040/12000] || Loss: 4.5167

iter [9050/12000] || Loss: 4.0180

iter [9060/12000] || Loss: 5.3055

iter [9070/12000] || Loss: 3.9833

iter [9080/12000] || Loss: 4.8178

iter [9090/12000] || Loss: 4.4742

iter [9100/12000] || Loss: 3.6661

iter [9110/12000] || Loss: 4.1763

iter [9120/12000] || Loss: 4.5916

iter [9130/12000] || Loss: 4.8381

iter [9140/12000] || Loss: 4.4131

iter [9150/12000] || Loss: 3.9451

iter [9160/12000] || Loss: 4.7541

iter [9170/12000] || Loss: 3.6682

iter [9180/12000] || Loss: 4.1446

iter [9190/12000] || Loss: 4.7649

iter [9200/12000] || Loss: 3.7275

iter [9210/12000] || Loss: 5.0252

iter [9220/12000] || Loss: 4.0646

iter [9230/12000] || Loss: 3.4723

iter [9240/12000] || Loss: 4.5796

iter [9250/12000] || Loss: 4.7237

iter [9260/12000] || Loss: 4.7627

iter [9270/12000] || Loss: 4.1203

iter [9280/12000] || Loss: 3.9745

iter [9290/12000] || Loss: 4.5499

iter [9300/12000] || Loss: 4.5299

iter [9310/12000] || Loss: 4.2996

iter [9320/12000] || Loss: 4.6204

iter [9330/12000] || Loss: 4.8062

iter [9340/12000] || Loss: 4.7765

iter [9350/12000] || Loss: 4.4410

iter [9360/12000] || Loss: 4.6436

iter [9370/12000] || Loss: 5.0264

iter [9380/12000] || Loss: 4.6492

iter [9390/12000] || Loss: 4.3095

iter [9400/12000] || Loss: 4.1487

iter [9410/12000] || Loss: 4.1966

iter [9420/12000] || Loss: 4.8119

iter [9430/12000] || Loss: 4.4698

iter [9440/12000] || Loss: 4.4833

iter [9450/12000] || Loss: 4.1502

iter [9460/12000] || Loss: 4.2902

iter [9470/12000] || Loss: 4.4851

iter [9480/12000] || Loss: 4.7534

iter [9490/12000] || Loss: 4.3837

iter [9500/12000] || Loss: 4.0675

iter [9510/12000] || Loss: 5.1883

iter [9520/12000] || Loss: 4.2238

iter [9530/12000] || Loss: 3.7498

iter [9540/12000] || Loss: 4.3480

iter [9550/12000] || Loss: 4.2786

iter [9560/12000] || Loss: 4.5116

iter [9570/12000] || Loss: 4.5150

iter [9580/12000] || Loss: 4.8275

iter [9590/12000] || Loss: 4.5455

iter [9600/12000] || Loss: 4.8288

iter [9610/12000] || Loss: 5.1905

iter [9620/12000] || Loss: 4.2839

iter [9630/12000] || Loss: 4.0937

iter [9640/12000] || Loss: 3.7542

iter [9650/12000] || Loss: 4.3918

iter [9660/12000] || Loss: 4.4288

iter [9670/12000] || Loss: 4.3572

iter [9680/12000] || Loss: 4.9142

iter [9690/12000] || Loss: 3.7434

iter [9700/12000] || Loss: 3.8816

iter [9710/12000] || Loss: 4.1380

iter [9720/12000] || Loss: 4.1023

iter [9730/12000] || Loss: 5.7598

iter [9740/12000] || Loss: 4.5463

iter [9750/12000] || Loss: 4.7704

iter [9760/12000] || Loss: 3.7874

iter [9770/12000] || Loss: 4.6828

iter [9780/12000] || Loss: 4.3589

iter [9790/12000] || Loss: 4.8938

iter [9800/12000] || Loss: 4.2800

iter [9810/12000] || Loss: 4.4586

iter [9820/12000] || Loss: 4.0567

iter [9830/12000] || Loss: 4.1475

iter [9840/12000] || Loss: 4.2152

iter [9850/12000] || Loss: 4.1977

iter [9860/12000] || Loss: 4.6082

iter [9870/12000] || Loss: 4.0297

iter [9880/12000] || Loss: 4.4461

iter [9890/12000] || Loss: 4.6072

iter [9900/12000] || Loss: 4.3564

iter [9910/12000] || Loss: 4.2659

iter [9920/12000] || Loss: 4.3708

iter [9930/12000] || Loss: 3.9584

iter [9940/12000] || Loss: 4.0858

iter [9950/12000] || Loss: 4.2318

iter [9960/12000] || Loss: 2.8292

iter [9970/12000] || Loss: 4.2123

iter [9980/12000] || Loss: 3.9682

iter [9990/12000] || Loss: 4.3130

iter [10000/12000] || Loss: 4.1018

在visdrone2018数据集上训练basic ssd

SSD 300 on drone full valid set

AP for pedestrian = 0.0686

AP for person = 0.0151

AP for bicycle = 0.0081

AP for car = 0.1891

AP for van = 0.0605

AP for truck = 0.0055

AP for tricycle = 0.0159

AP for awning-tricycle = 0.0604

AP for bus = 0.0911

AP for motor = 0.0149

AP for others = 0.0000

Mean AP = 0.0481

~~~~~~~~

Results:

0.069

0.015

0.008

0.189

0.061

0.006

0.016

0.060

0.091

0.015

0.000

0.048

~~~~~~~~

--------------------------------------------------------------

Results computed with the \*\*unofficial\*\* Python eval code.

Results should be very close to the official MATLAB eval code.

--------------------------------------------------------------

[Speed]: 0.3321 s, 3.0 fps

SSD 300 on drone mini valid set

AP for pedestrian = 0.0947

AP for person = 0.0141

AP for bicycle = 0.0084

AP for car = 0.1897

AP for van = 0.0607

AP for truck = 0.0055

AP for tricycle = 0.0160

AP for awning-tricycle = 0.0599

AP for bus = 0.0911

AP for motor = 0.0151

AP for others = 0.0000

Mean AP = 0.0505

~~~~~~~~

Results:

0.095

0.014

0.008

0.190

0.061

0.006

0.016

0.060

0.091

0.015

0.000

0.050

~~~~~~~~

--------------------------------------------------------------

Results computed with the \*\*unofficial\*\* Python eval code.

Results should be very close to the official MATLAB eval code.

--------------------------------------------------------------

[Speed]: 0.3449 s, 2.9 fps

SSD 512 on mini valid set

AP for pedestrian = 0.1249

AP for person = 0.1233

AP for bicycle = 0.1193

AP for car = 0.2941

AP for van = 0.1709

AP for truck = 0.0916

AP for tricycle = 0.0684

AP for awning-tricycle = 0.1286

AP for bus = 0.0077

AP for motor = 0.0585

AP for others = 0.0000

Mean AP = 0.1079

~~~~~~~~

Results:

0.125

0.123

0.119

0.294

0.171

0.092

0.068

0.129

0.008

0.058

0.000

0.108

~~~~~~~~

--------------------------------------------------------------

Results computed with the \*\*unofficial\*\* Python eval code.

Results should be very close to the official MATLAB eval code.

--------------------------------------------------------------

[Speed]: 0.2914 s, 3.4 fps

FGFA default

val/uav0000117\_02622\_v

testing 336.JPEG 0.2755s, 3.63 fps

testing 337.JPEG 0.2753s, 3.63 fps

testing 338.JPEG 0.2751s, 3.63 fps

testing 339.JPEG 0.2750s, 3.64 fps

testing 340.JPEG 0.2748s, 3.64 fps

testing 341.JPEG 0.2746s, 3.64 fps

testing 342.JPEG 0.2745s, 3.64 fps

testing 343.JPEG 0.2743s, 3.65 fps

testing 344.JPEG 0.2741s, 3.65 fps

testing 345.JPEG 0.2739s

testing 346.JPEG 0.2734s

testing 347.JPEG 0.2729s

testing 348.JPEG 0.2724s