EDU

CSDN学院　IT实战派

Python 深度学习

深度学习实战篇：目标检测模型与应用

讲师：彭靖田

本章目录

EDU

CSDN学院　IT实战派

Python 深度学习

深度学习实战：目标检测问题定义与说明

讲师：彭靖田

目标检测问题

图像分类

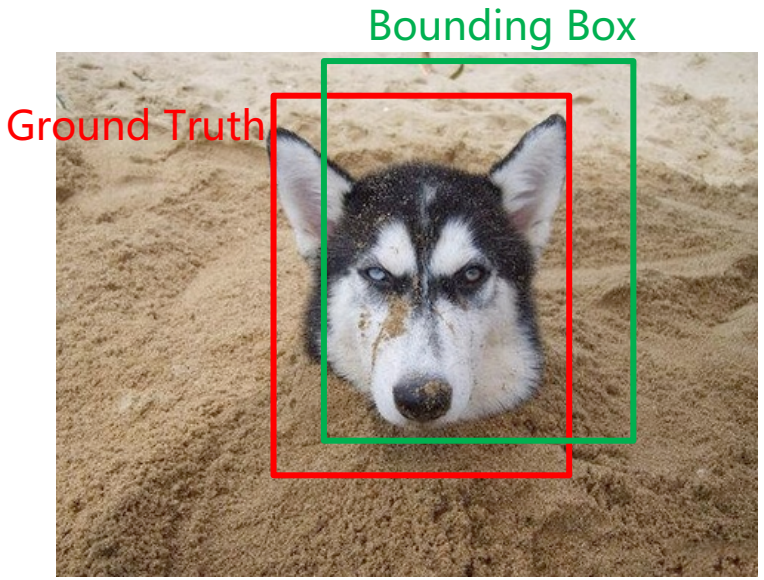
狗

目标定位

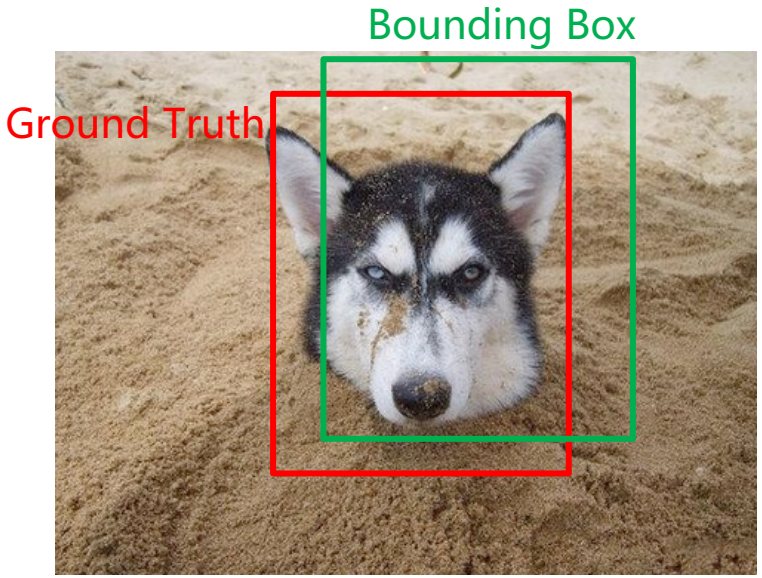
目标检测

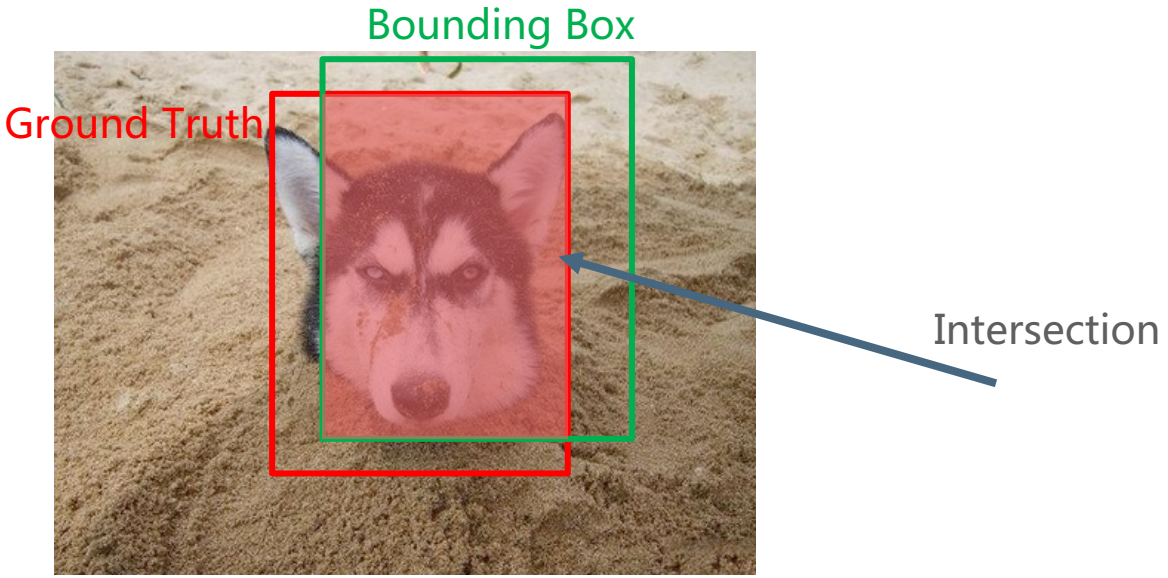
狗

目标检测评估：Ground Truth



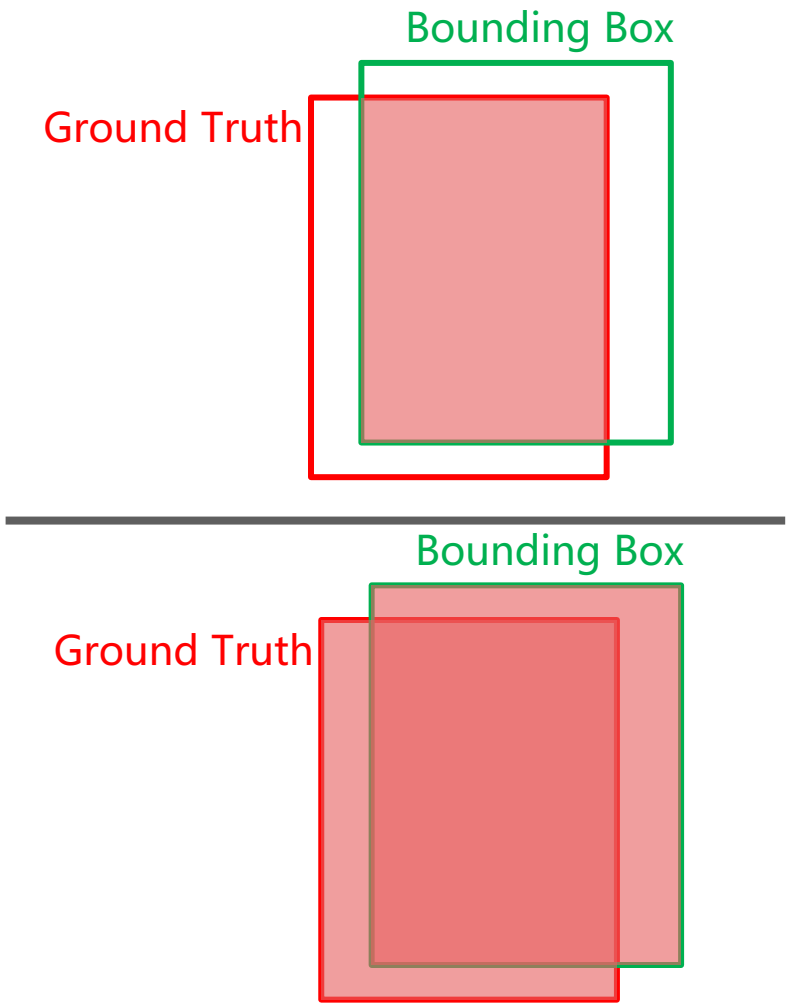
目标检测评估：Intersection over Union (IoU)





目标检测评估：Intersection over Union (IoU)

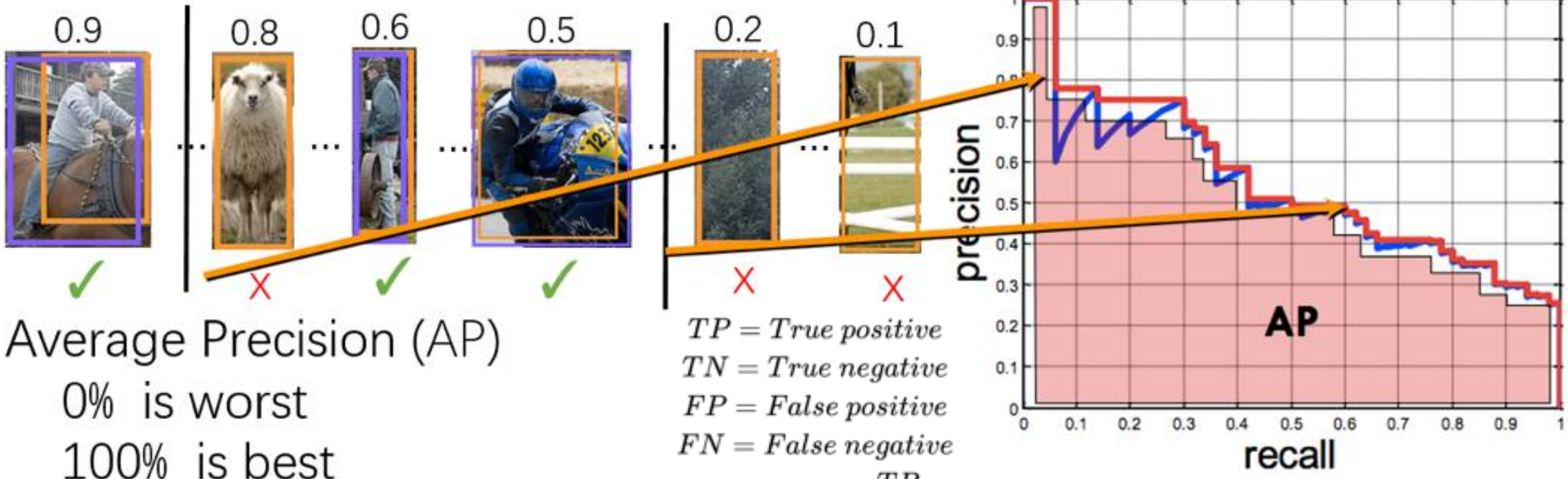
$$IoU = \frac{Intersection}{Union} =$$



目标检测评估：mean Average Precision（mAP）

- Evaluation metric

[text adopted from Ross Girshick]



Average Precision (AP)

0% is worst

100% is best

mean AP over classes (mAP)

$TP = True\ positive$

$TN = True\ negative$

$FP = False\ positive$

$FN = False\ negative$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

EDU

CSDN学院　IT实战派

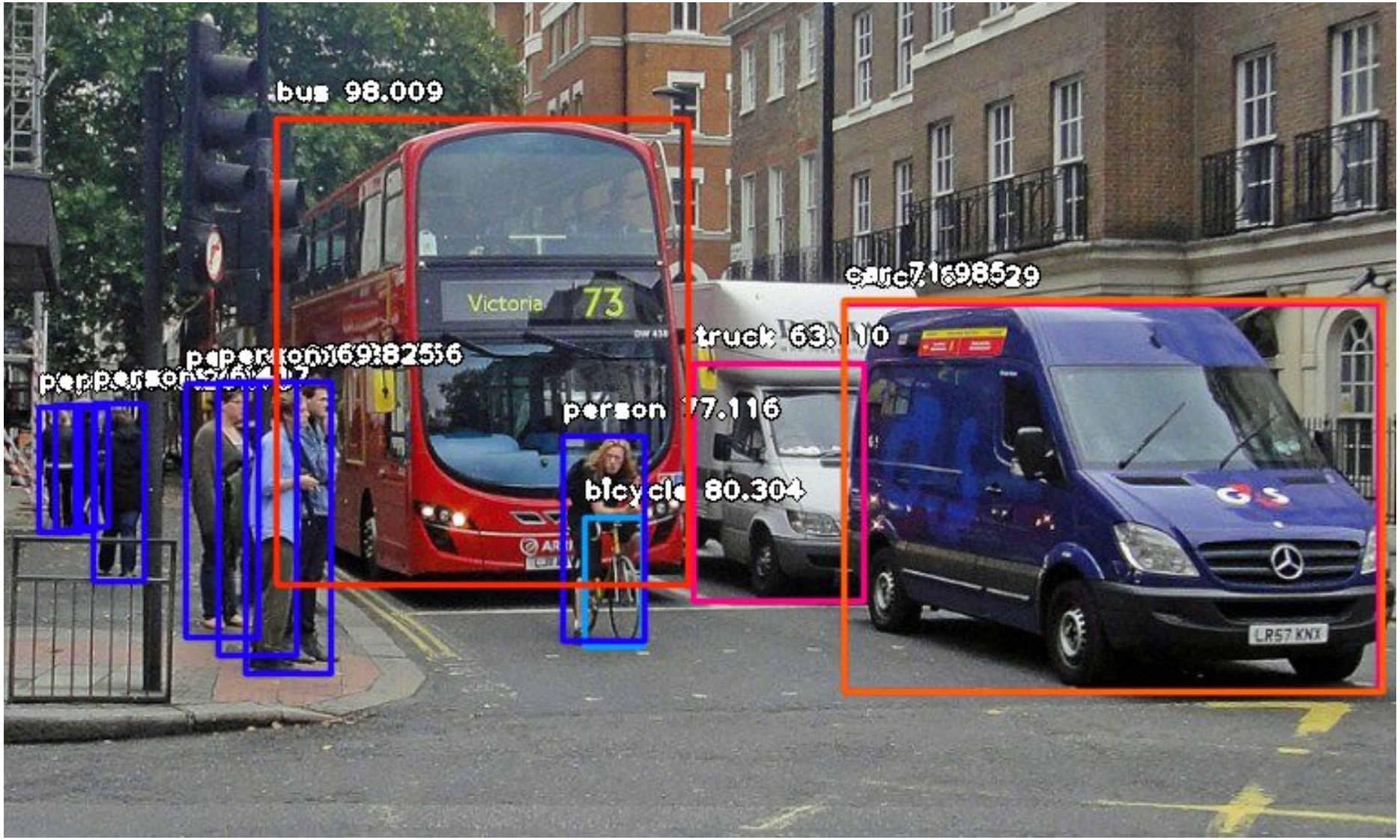
Python 深度学习

深度学习实战：目标检测模型的应用场景

讲师：彭靖田

目标检测应用：公共安全

目标检测应用：行人检测

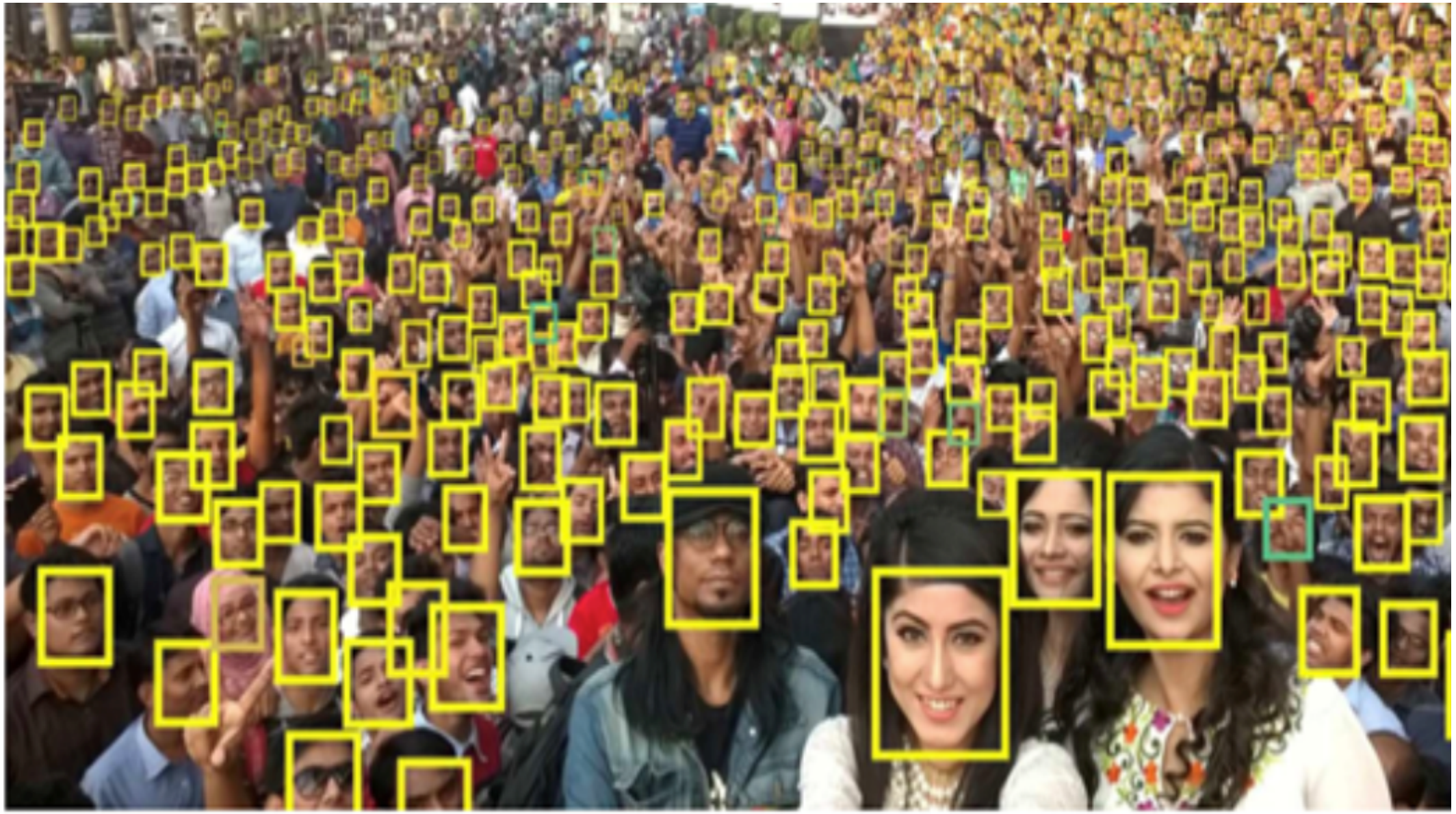

bus 98.009
Victoria 73
truck 63.110
person 77.116
bicycle 80.304
LR57 KNX

目标检测应用：货物盘点

目标检测应用：安全防护


保税零件区
BONDED MATERIAL
H
直送零件区
SHIP TO USE
成零件区
RAW MATERIAL
I
直送零件区
SHIP TO USE
J

目标检测应用：人脸识别



EDU

CSDN学院　IT实战派

Python 深度学习

深度学习实战：目标检测常用数据集介绍

讲师：彭靖田

通用目标检测数据集

Pascal VOC

- The PASCAL Visual Object Classes (VOC) Challenge

MS-COCO

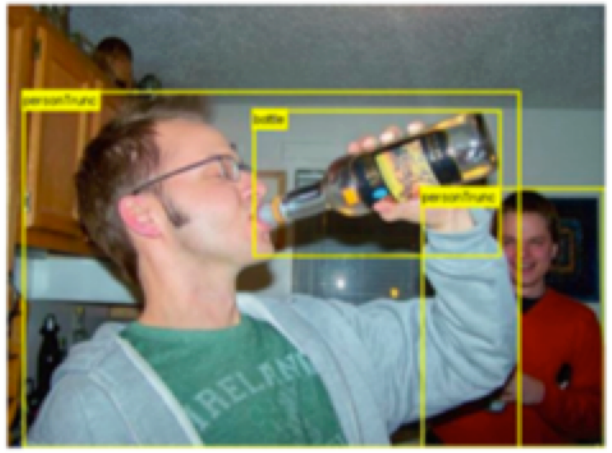
- Microsoft Common Objects in Context

ILSVRC

- The ImageNet Large Scale Visual Recognition Challenge

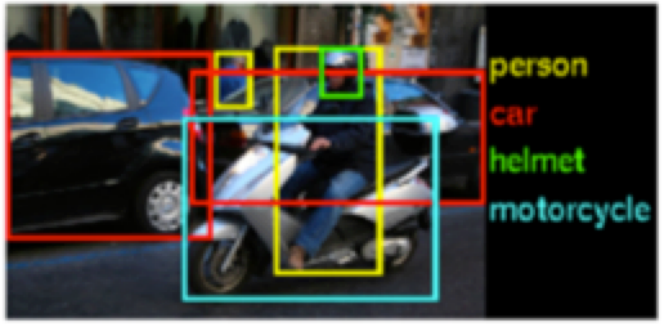
通用目标检测数据集

Pascal VOC



COCO

ILSVRC

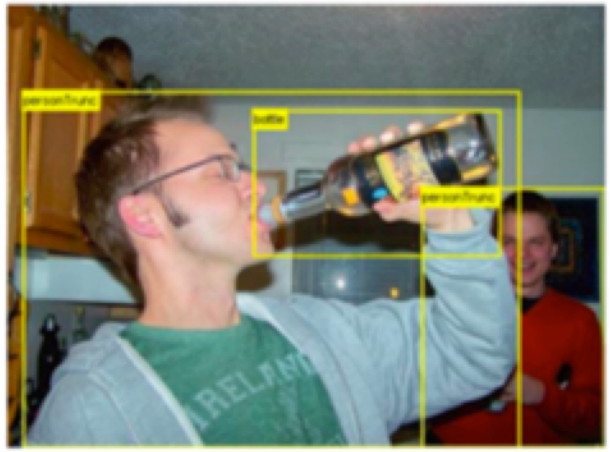


PASCAL VOC 数据集

4个大类：person, animal, vehicle, household

20个小类：

- person
- bird, cat, cow, dog, horse, sheep
- aeroplane, bicycle, boat, bus, car, motorbike, train
- bottle, chair, dining table, potted plant, sofa, tv/monitor



- VOC 2007：**9,963**张标注过的图片， 由train/val/test三部分组成， 共标注出**24,640**个物体。
- VOC 2012：trainval/test包含08-11年的所有对应图片。 trainval有**11,540**张图片共**27,450**个物体。

COCO 数据集

12个大类：

[‘appliance’ , ‘food’ , ‘indoor’ , ‘accessory’ , ‘electronic’ , ‘furniture’ , ‘vehicle’ , ‘sports’ , ‘animal’ , ‘kitchen’ , ‘person’ , ‘outdoor’]

91个小类：

[‘person’ , ‘bicycle’ , ‘car’ , ‘motorcycle’ , ‘airplane’ , ‘bus’ , ‘train’ , ‘truck’ , ‘boat’ , ‘traffic light’ , ‘fire hydrant’ , ‘stop sign’ , ‘parking meter’ , ‘bench’ , ‘bird’ , ‘cat’ , ‘dog’ , ‘horse’ , ‘sheep’ , ‘cow’ , ‘elephant’ , ‘bear’ , ‘zebra’ , ‘giraffe’ , ‘backpack’ , ‘umbrella’ , ‘handbag’ , ‘tie’ , ‘suitcase’ , ‘frisbee’ , ‘skis’ , ‘snowboard’ , ‘sports ball’ , ‘kite’ , ‘baseball bat’ , ‘baseball glove’ , ‘skateboard’ , ‘surfboard’ ,...]

IMAGENET Large Scale Visual Recognition Challenge (ILSVRC)

ILSVRC 具体信息：

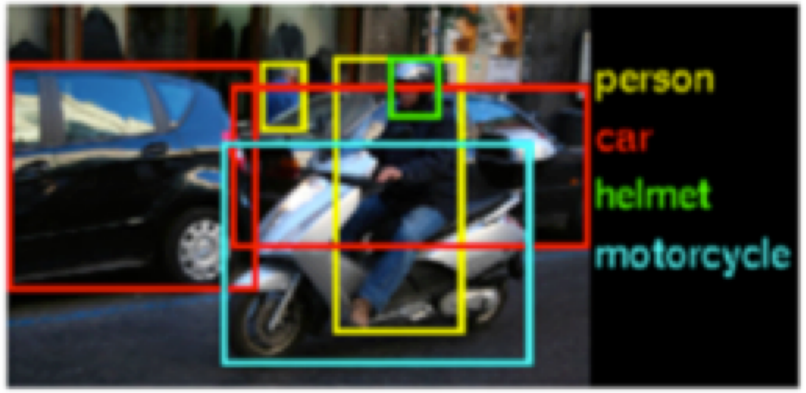
识别小类: 21841

图像总数: 1400万+

带有 Bounding box 的图像总数: 1,034,908

带有 SIFT 特征的识别小类: 1000

带有 SIFT 特征的图像总数: 1200万



EDU

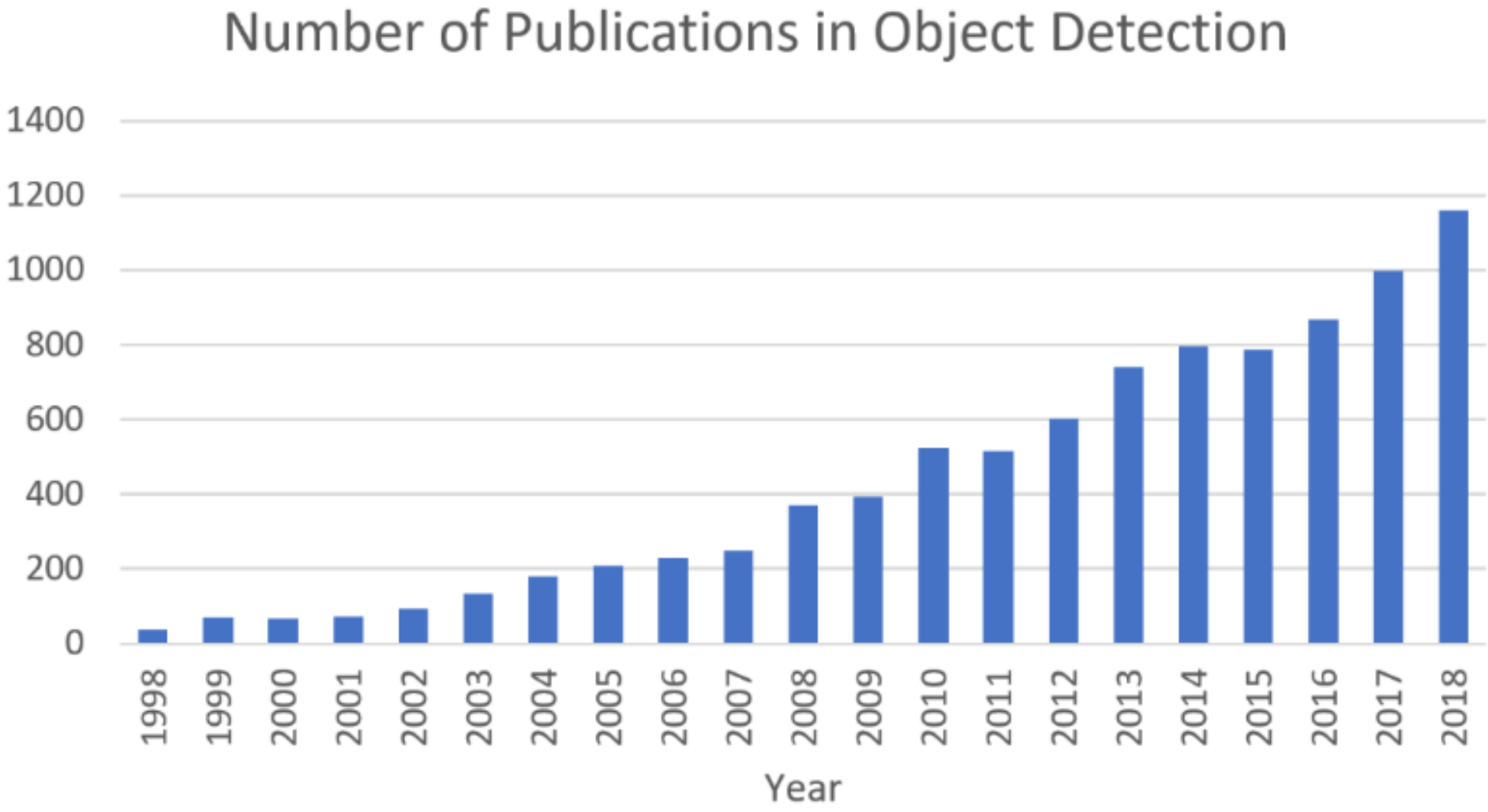
CSDN学院　IT实战派

Python 深度学习

深度学习实战：深度目标检测模型发展概述

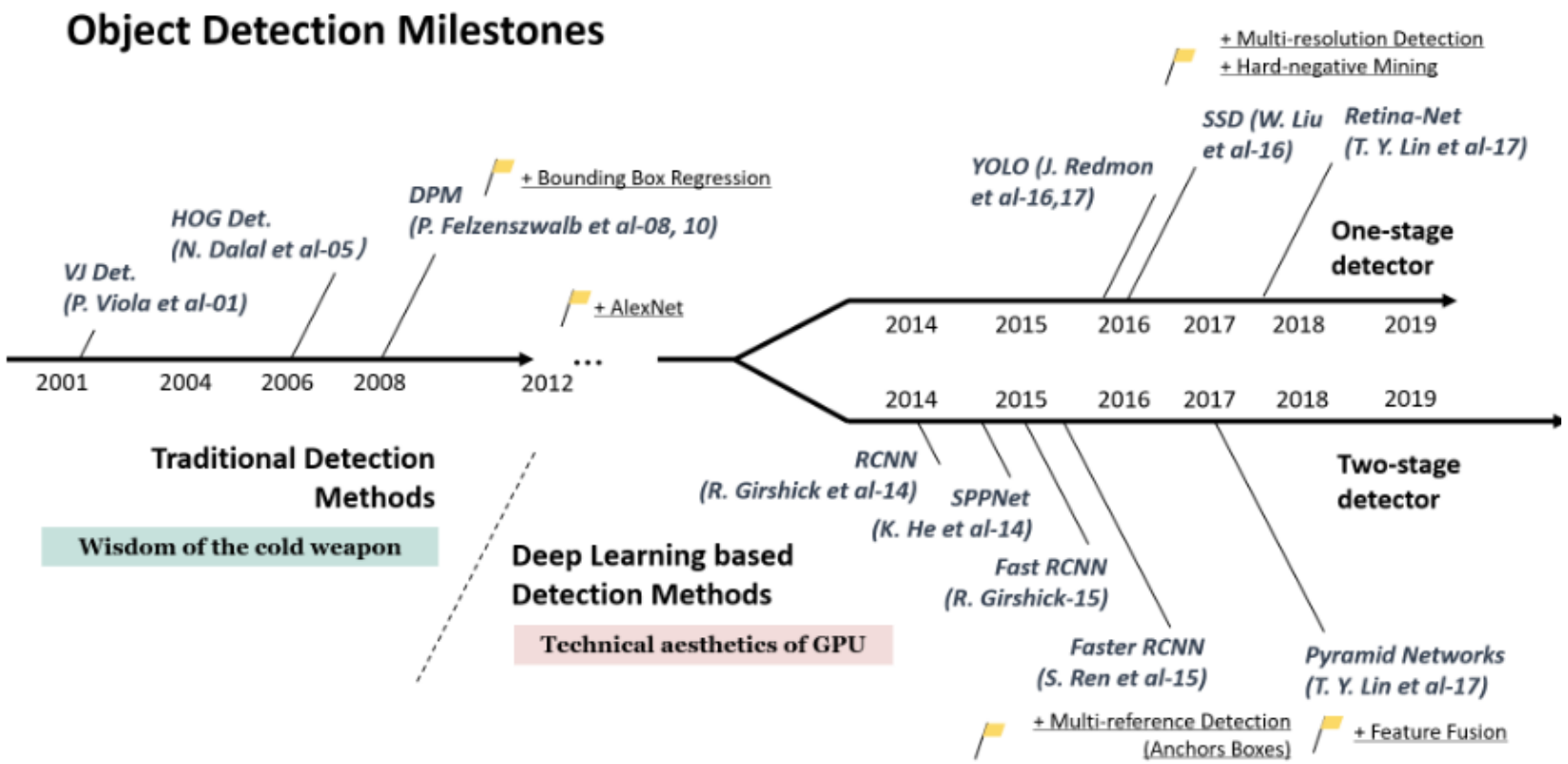
讲师：彭靖田

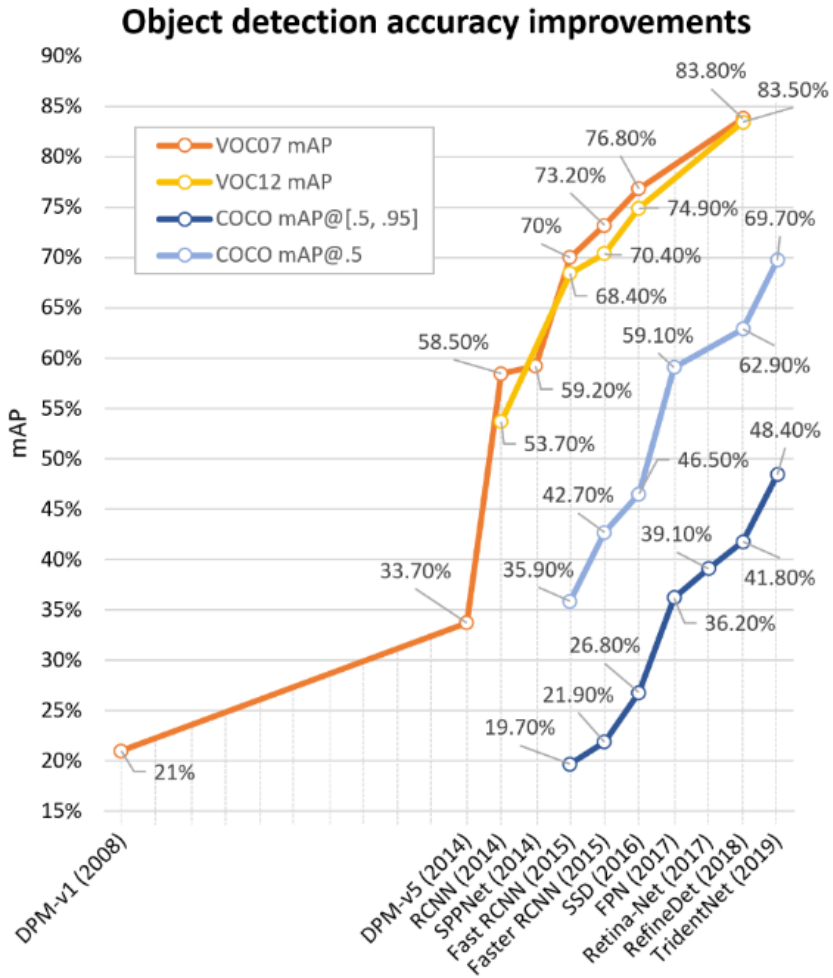
目标检测近20年发展



Ref: Zou, Z., Shi, Z., Guo, Y. and Ye, J., 2019. Object detection in 20 years: A survey. arXiv preprint arXiv:1905.05055.

目标检测近20年里程碑





Ref: Zou, Z., Shi, Z., Guo, Y. and Ye, J., 2019. Object detection in 20 years: A survey. arXiv preprint arXiv:1905.05055.

深度目标检测网络

两阶段检测器（Two-stage Detectors）

- R-CNN
- Fast R-CNN
- Faster R-CNN
- R-FCN

一阶段检测器（One-stage Detectors）

- YOLO v1
- YOLO v2
- YOLO v3

EDU

CSDN学院　IT实战派

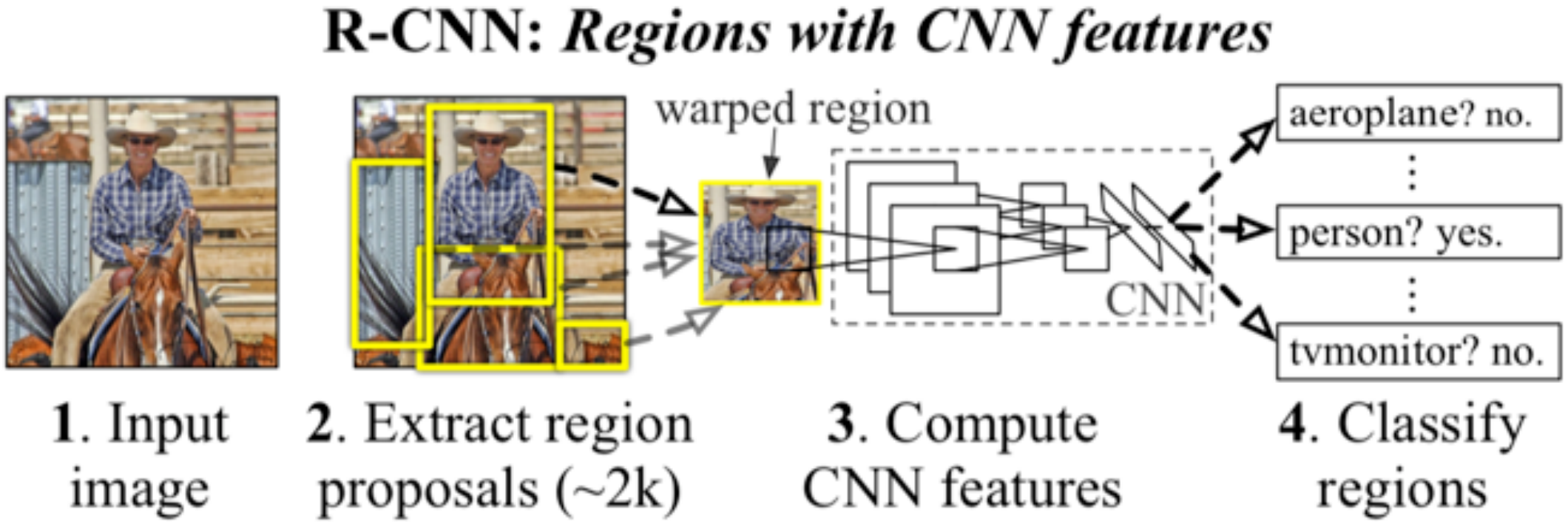
Python 深度学习

深度学习实战：二阶段（two-stage）目标检测模型概述

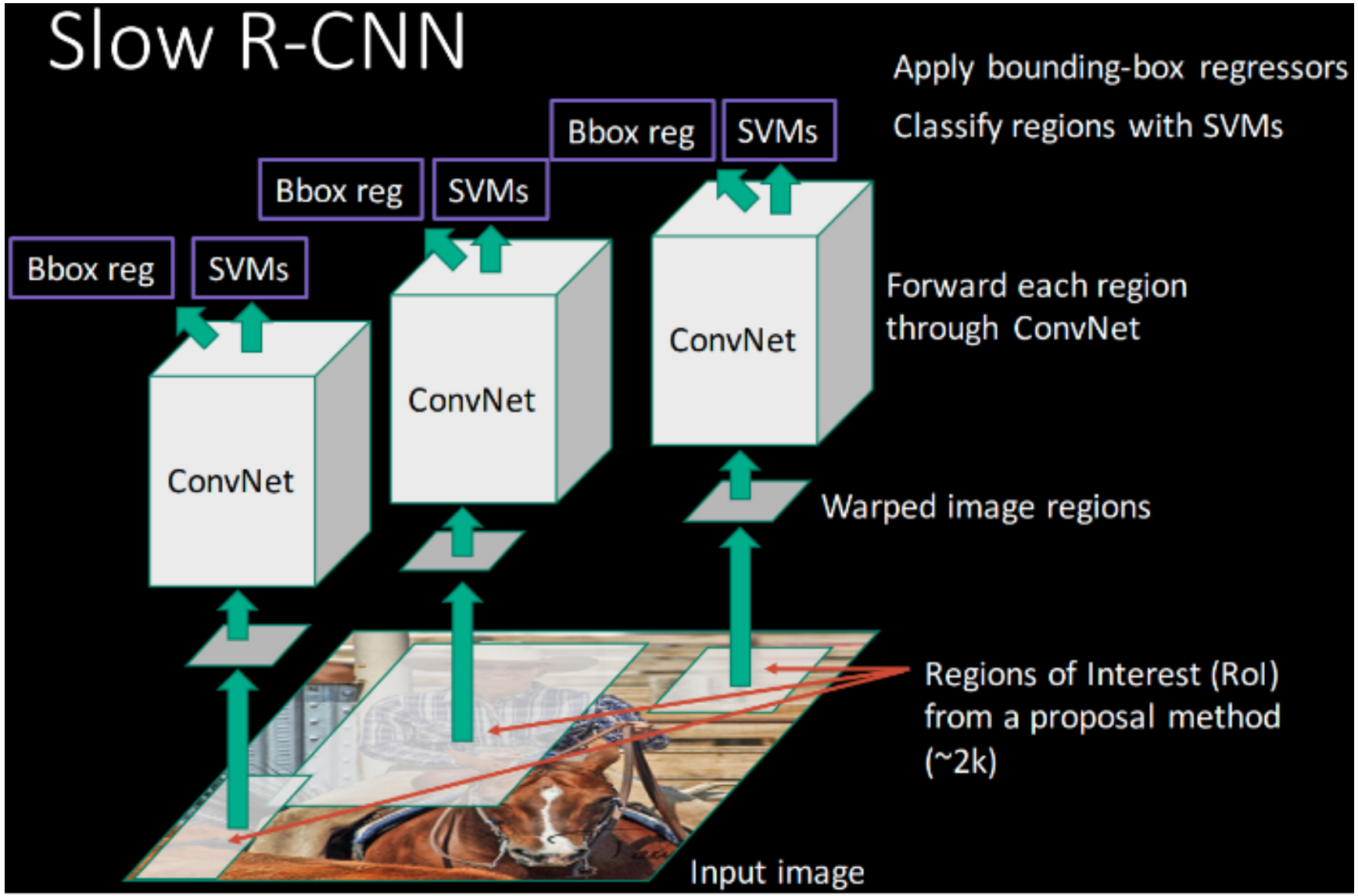
讲师：彭靖田

R-CNN

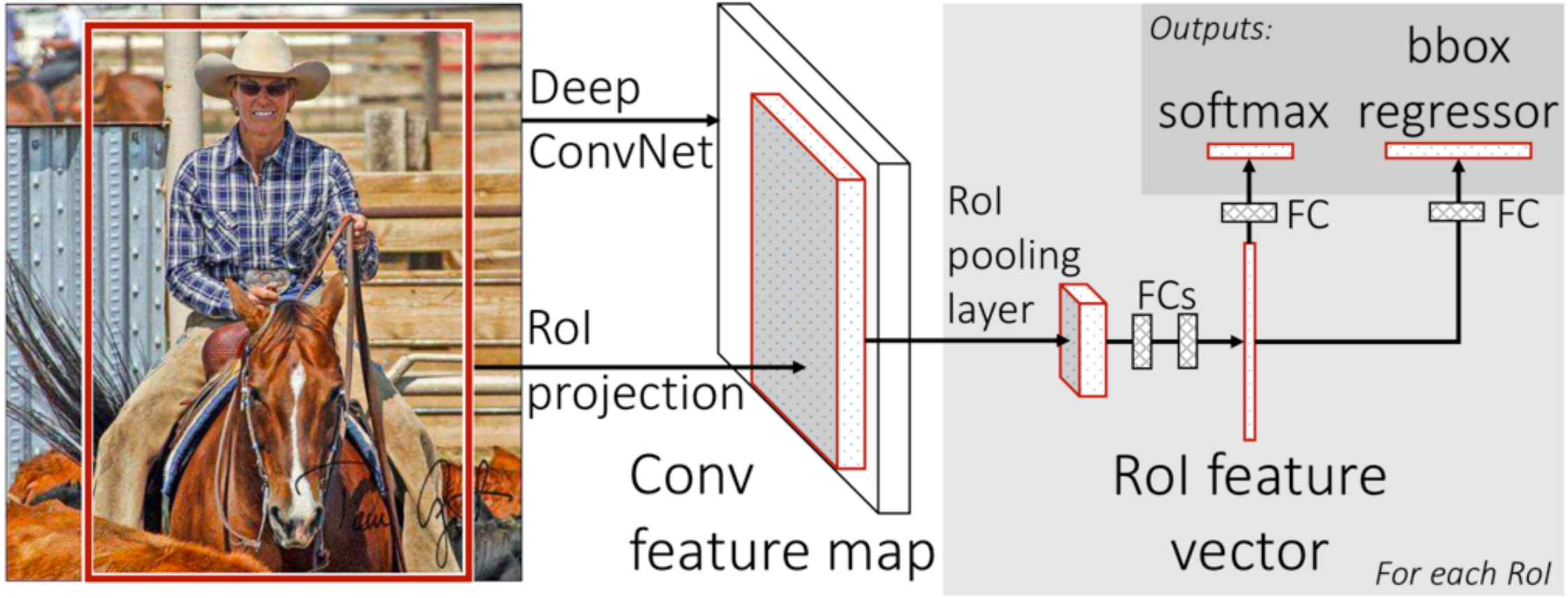
开启CNN+目标检测的大门



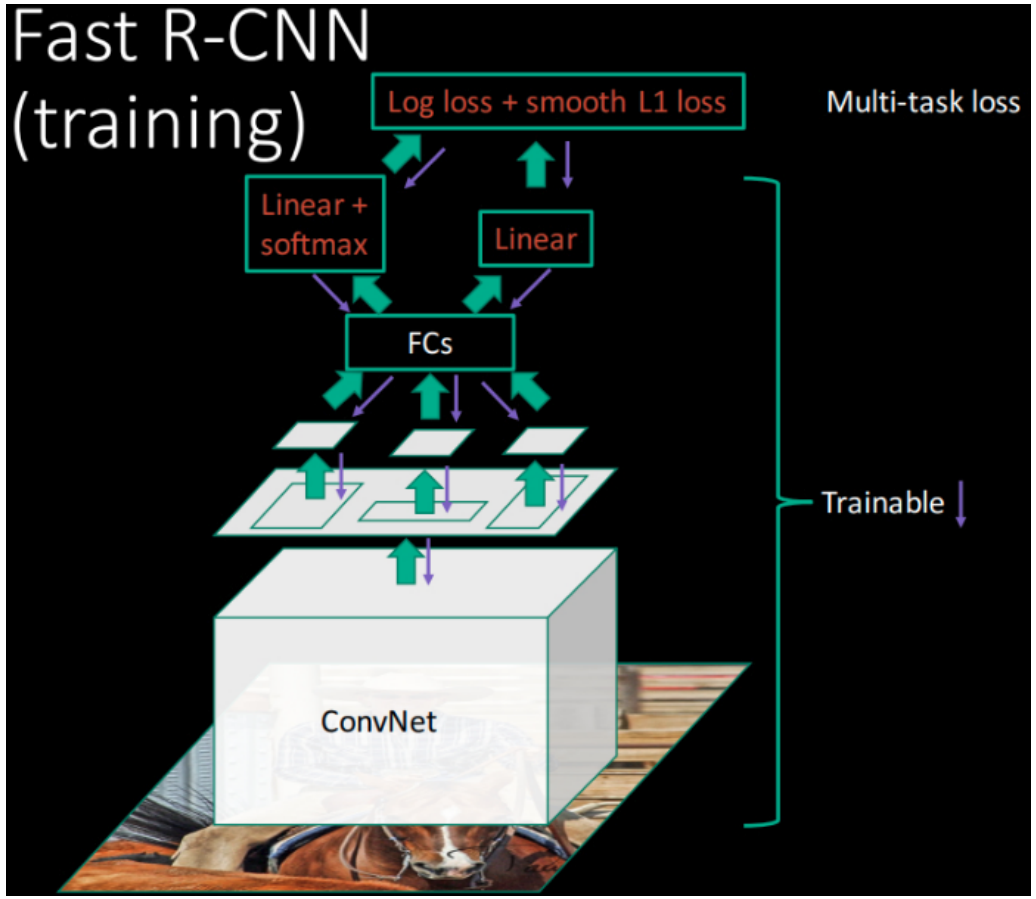
R-CNN

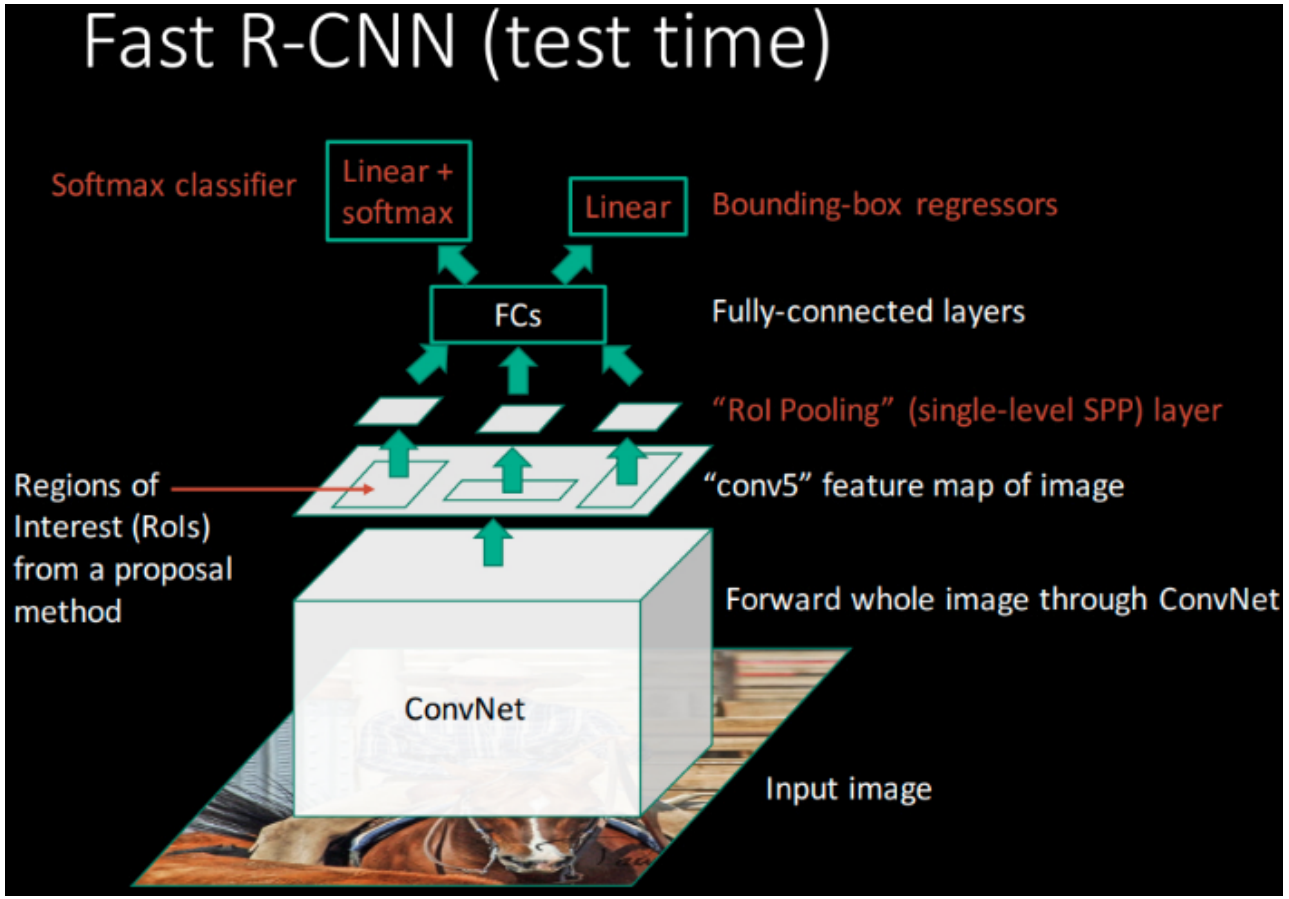

Slow R-CNN
Apply bounding-box regressors
Classify regions with SVMs
Bbox reg
SVMs
Bbox reg
SVMs
Bbox reg
SVMs
ConvNet
ConvNet
ConvNet
Forward each region through ConvNet
Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image

Fast R-CNN

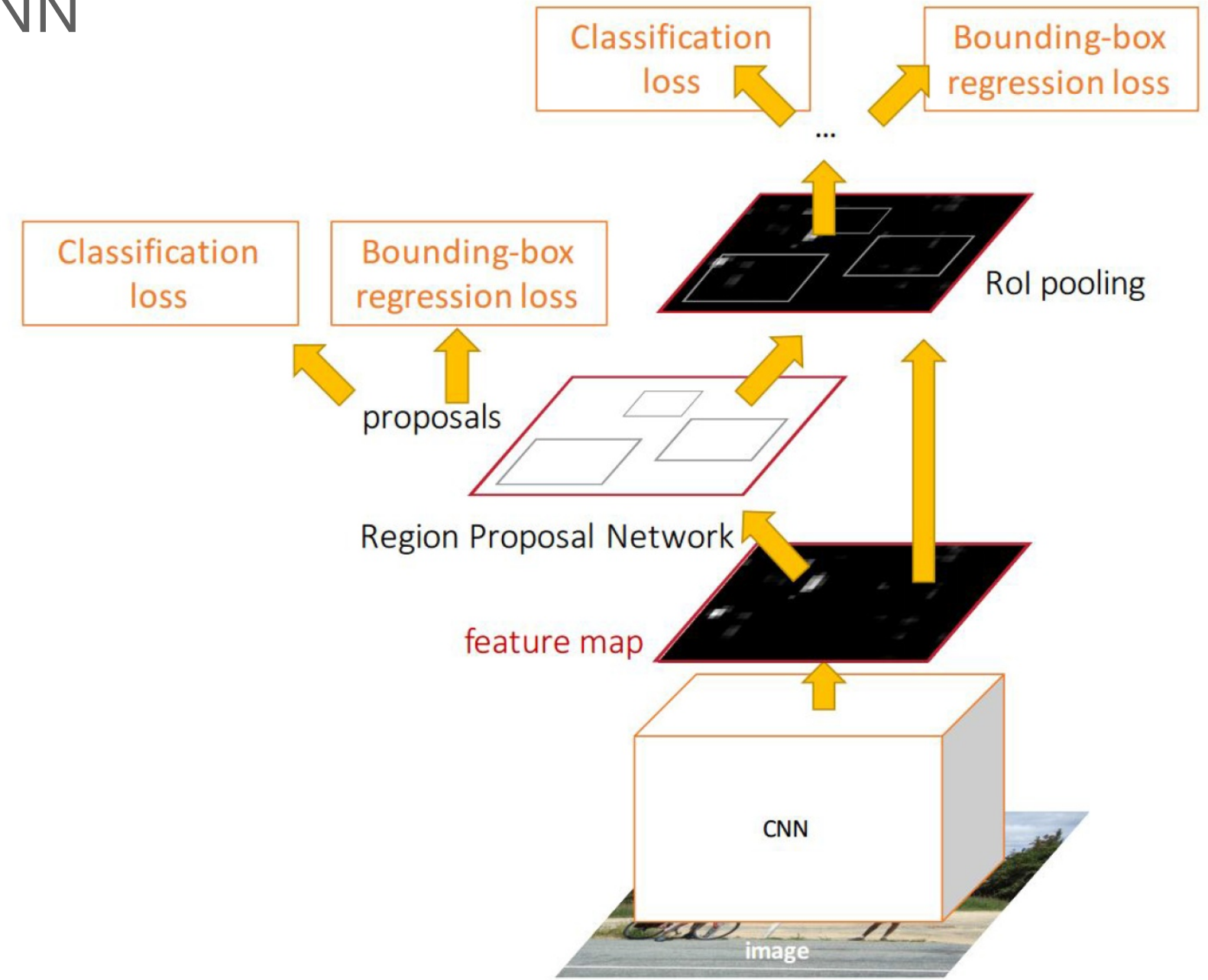

Deep
ConvNet
RoI
projection
Conv
feature map
RoI
pooling
layer
FCs
Outputs:
softmax
bbox
regressor
FC
FC
RoI feature
vector
For each RoI

Fast R-CNN


Fast R-CNN
(training)
Log loss + smooth L1 loss
Multi-task loss
Linear + softmax
Linear
FCs
Trainable
ConvNet


Fast R-CNN (test time)
Softmax classifier
Linear + softmax
Linear
Bounding-box regressors
FCs
Fully-connected layers
"RoI Pooling" (single-level SPP) layer
Regions of Interest (RoIs) from a proposal method
"conv5" feature map of image
Forward whole image through ConvNet
ConvNet
Input image

Faster R-CNN


Classification loss
Bounding-box regression loss
...
RoI pooling
Classification loss
Bounding-box regression loss
proposals
Region Proposal Network
feature map
CNN
image

EDU

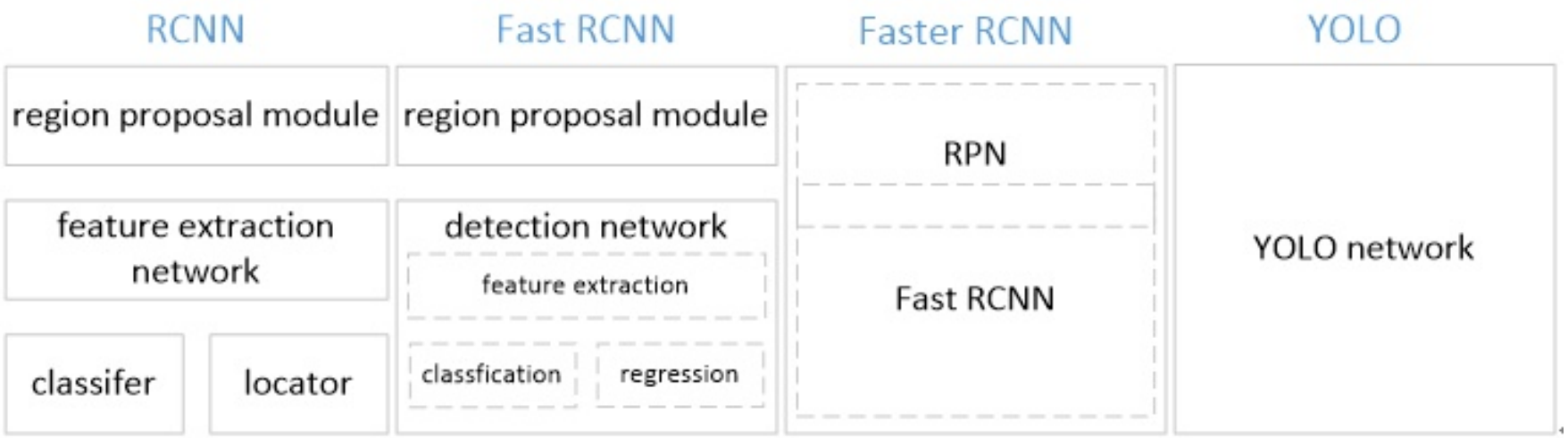
CSDN学院　IT实战派

Python 深度学习

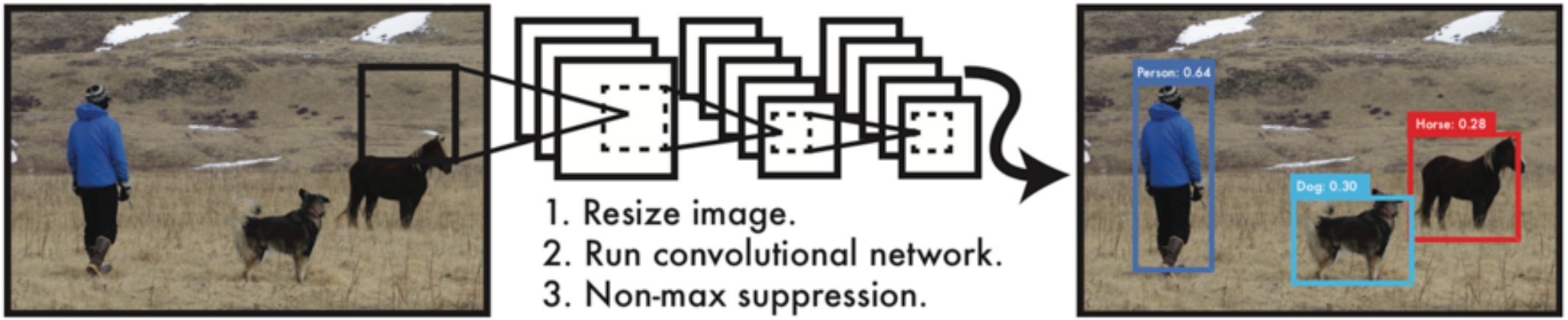
深度学习实战：目标检测 YOLO 系列模型发展

讲师：彭靖田

YOLO 与 RCNN 系列对比

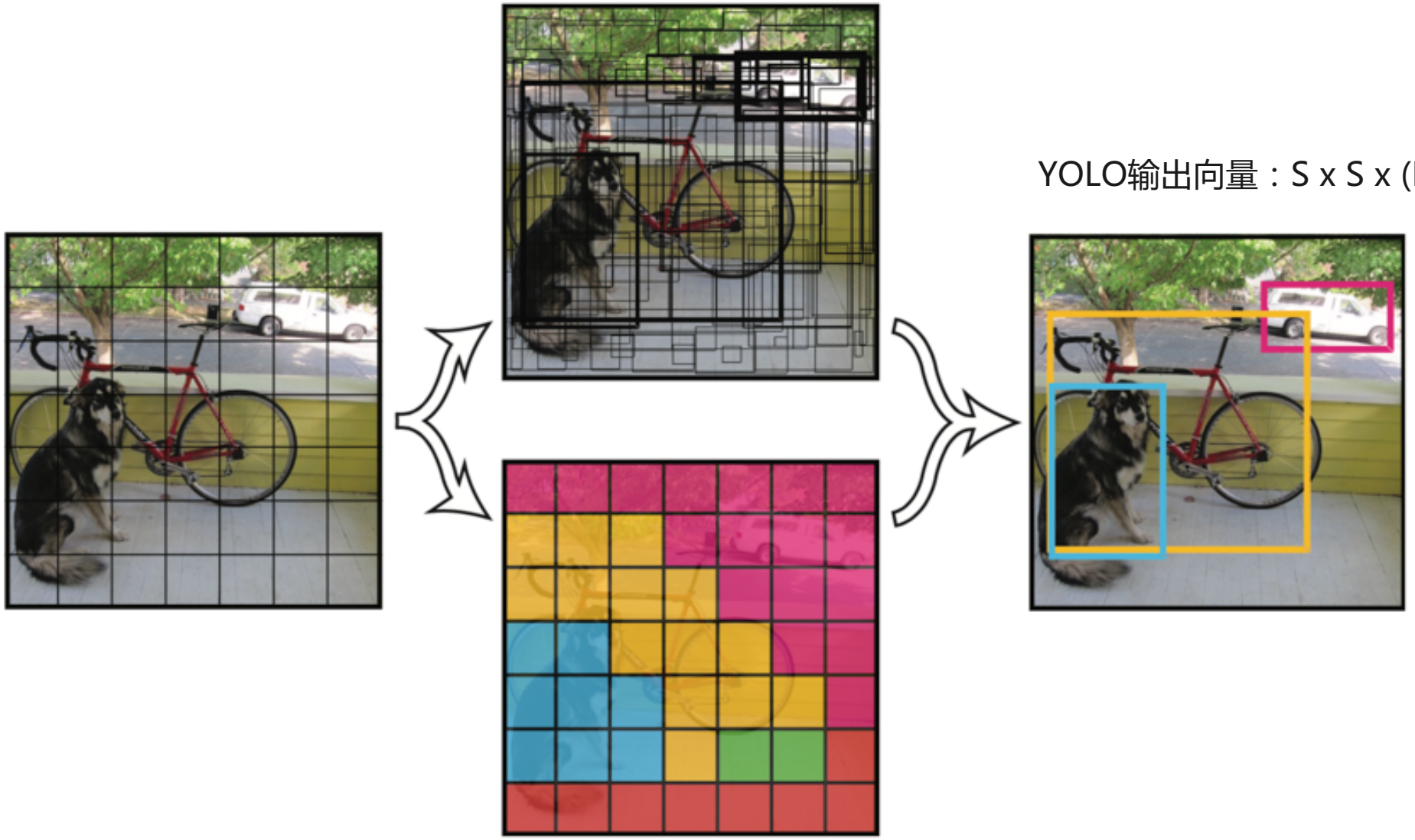


YOLOv1：首个深度学习的一阶段检测器



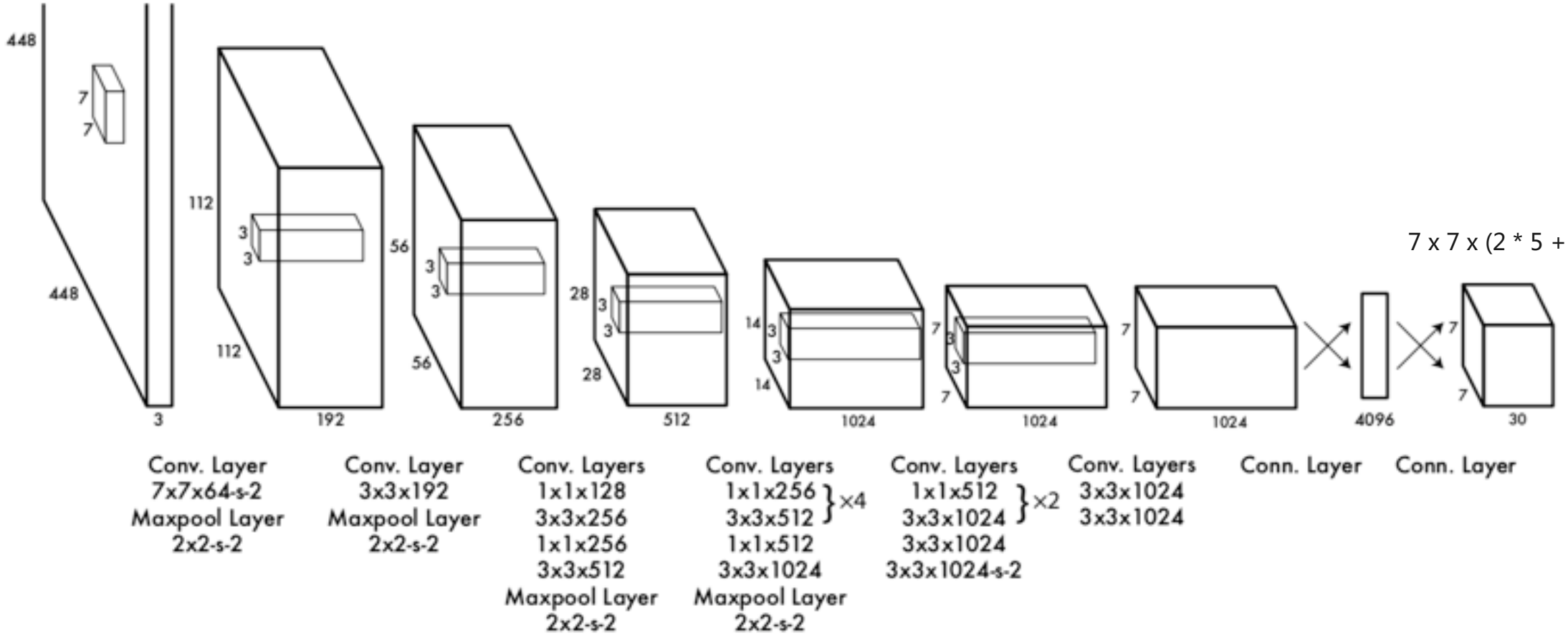
YOLOv1：首个深度学习的一阶段检测器

YOLO输出向量：S x S x (B * 5 + C)

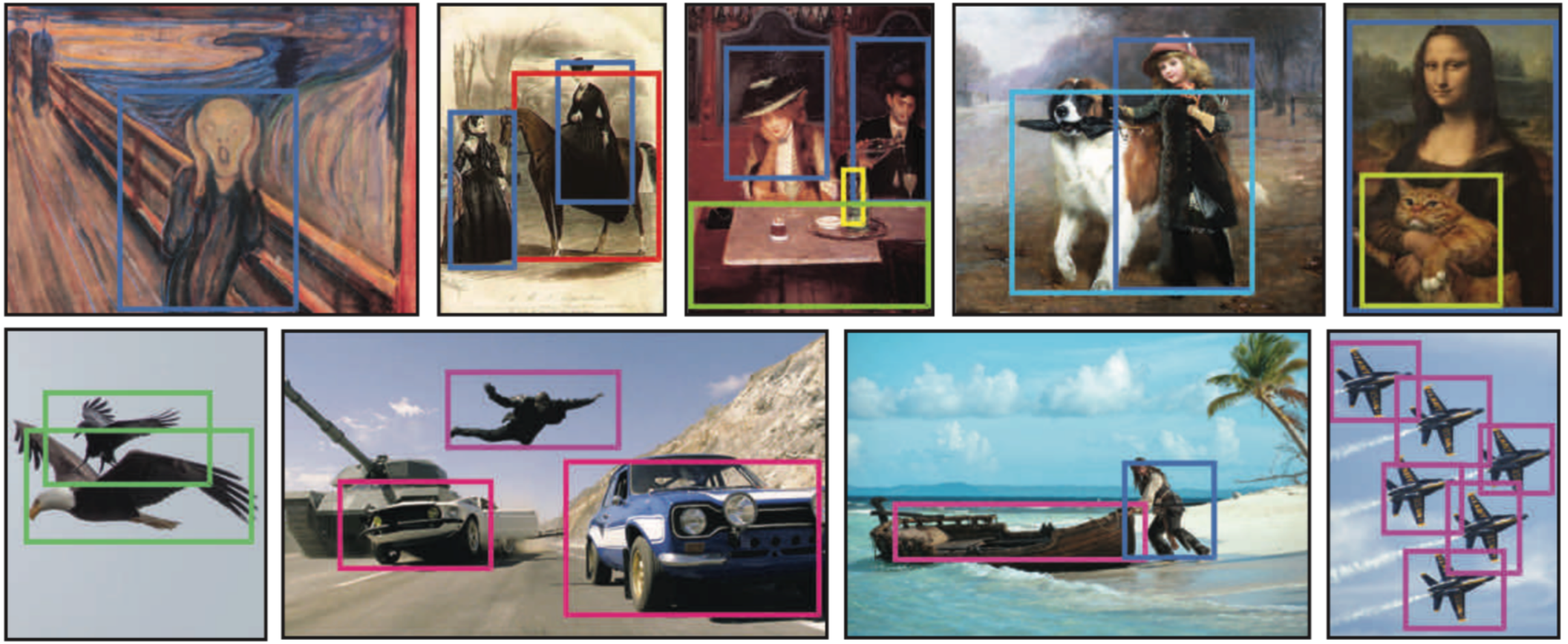


YOLOv1：首个深度学习的一阶段检测器

YOLO检测网络包括24个卷积层和2个全连接层，如下图所示。



YOLOv1 检测效果展示



YOLOv2：更快、更好、更强

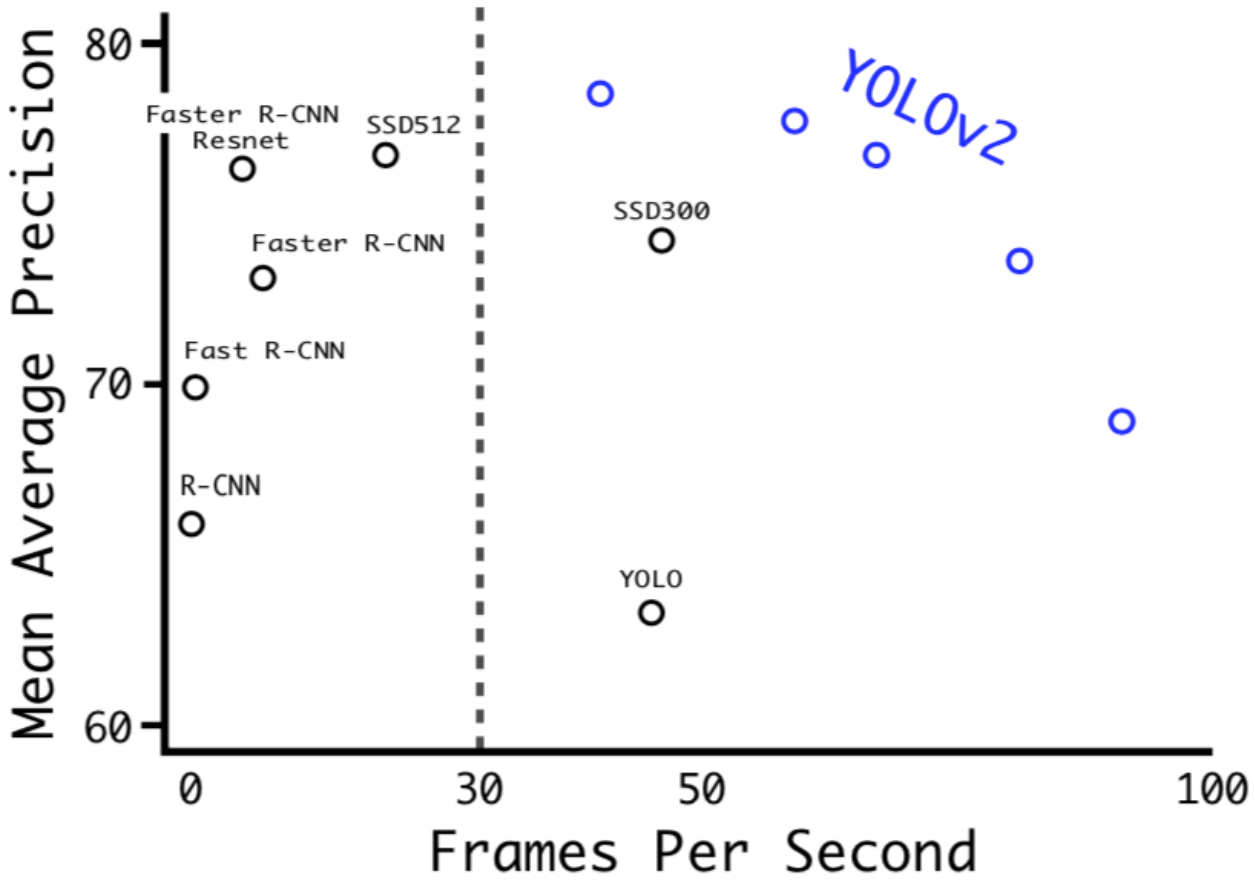


Figure 4: Accuracy and speed on VOC 2007.

YOLOv2：更快、更好、更强

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast R-CNN [5]	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
Faster R-CNN [15]	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
YOLO [14]	07++12	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
SSD300 [11]	07++12	72.4	85.6	80.1	70.5	57.6	46.2	79.4	76.1	89.2	53.0	77.0	60.8	87.0	83.1	82.3	79.4	45.9	75.9	69.5	81.9	67.5
SSD512 [11]	07++12	74.9	87.4	82.3	75.8	59.0	52.6	81.7	81.5	90.0	55.4	79.0	59.8	88.4	84.3	84.7	83.3	50.2	78.0	66.3	86.3	72.0
ResNet [6]	07++12	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6
YOLOv2 544	07++12	73.4	86.3	82.0	74.8	59.2	51.8	79.8	76.5	90.6	52.1	78.2	58.5	89.3	82.5	83.4	81.3	49.1	77.2	62.4	83.8	68.7

Table 4: PASCAL VOC2012 `test` detection results. YOLOv2 performs on par with state-of-the-art detectors like Faster R-CNN with ResNet and SSD512 and is $2-10\times$ faster.

		0.5:0.95	0.5	0.75	S	M	L	1	10	100	S	M	L
Fast R-CNN [5]	train	19.7	35.9	-	-	-	-	-	-	-	-	-	-
Fast R-CNN[1]	train	20.5	39.9	19.4	4.1	20.0	35.8	21.3	29.5	30.1	7.3	32.1	52.0
Faster R-CNN[15]	trainval	21.9	42.7	-	-	-	-	-	-	-	-	-	-
ION [1]	train	23.6	43.2	23.6	6.4	24.1	38.3	23.2	32.7	33.5	10.1	37.7	53.6
Faster R-CNN[10]	trainval	24.2	45.3	23.5	7.7	26.4	37.1	23.8	34.0	34.6	12.0	38.5	54.4
SSD300 [11]	trainval35k	23.2	41.2	23.4	5.3	23.2	39.6	22.5	33.2	35.3	9.6	37.6	56.5
SSD512 [11]	trainval35k	**26.8**	**46.5**	**27.8**	**9.0**	**28.9**	**41.9**	**24.8**	**37.5**	**39.8**	**14.0**	**43.5**	**59.0**
YOLOv2 [11]	trainval35k	21.6	44.0	19.2	5.0	22.4	35.5	20.7	31.6	33.3	9.8	36.5	54.4

Table 5: Results on COCO `test-dev2015`. Table adapted from [11]

YOLOv2：更快、更好、更强

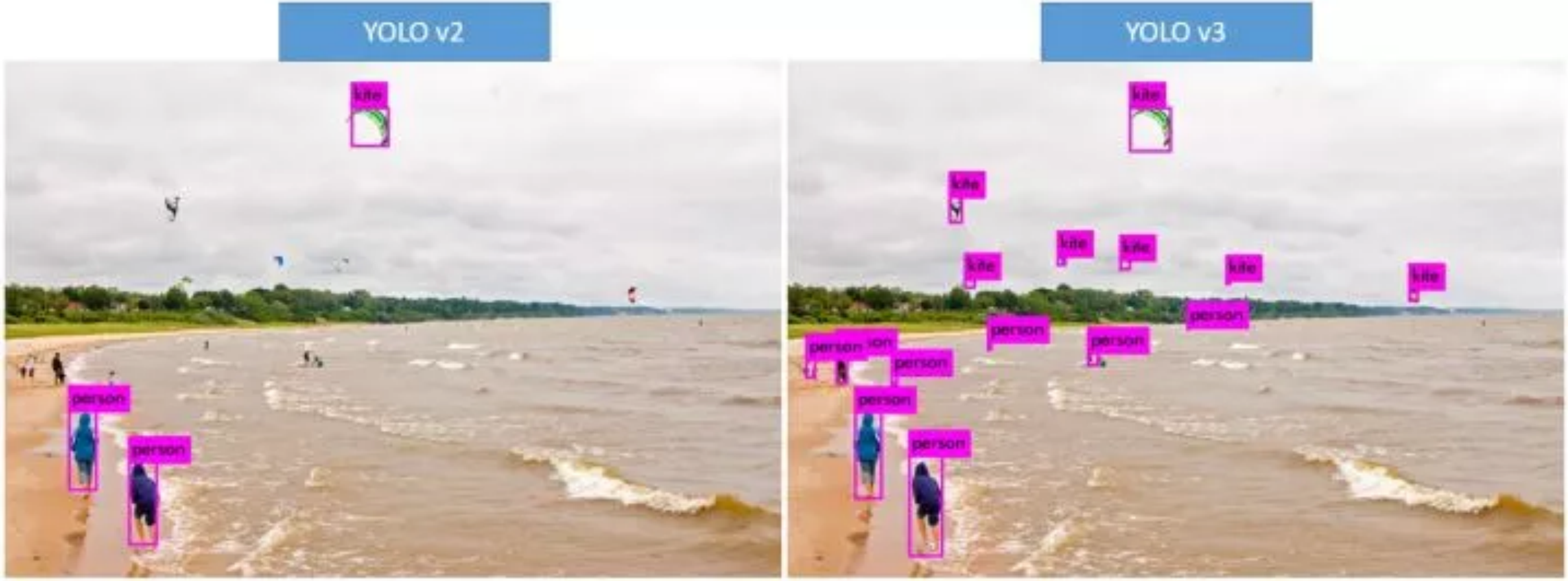
	YOLO							YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓	✓	✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓
anchor boxes?				✓	✓			
new network?					✓	✓	✓	✓
dimension priors?						✓	✓	✓
location prediction?						✓	✓	✓
passthrough?							✓	✓
multi-scale?								✓
hi-res detector?								✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	**78.6**

YOLOv2：更快、更好、更强

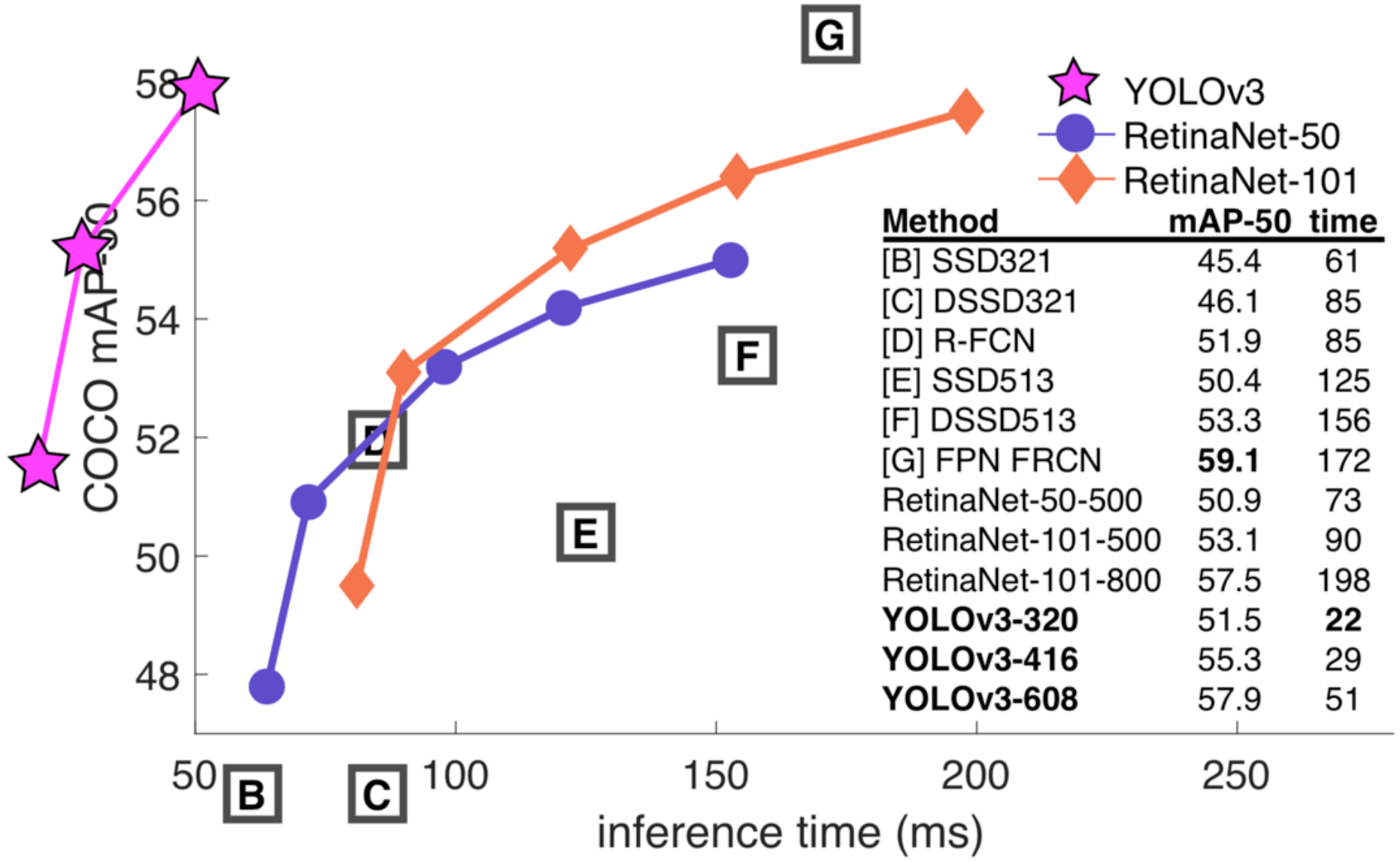
Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

Table 6: Darknet-19.

YOLOv3：小目标识别大提升



YOLOv3：工业级高性能目标检测器



Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51

YOLOv3：Darknet-53 网络结构

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

EDU

CSDN学院　IT实战派

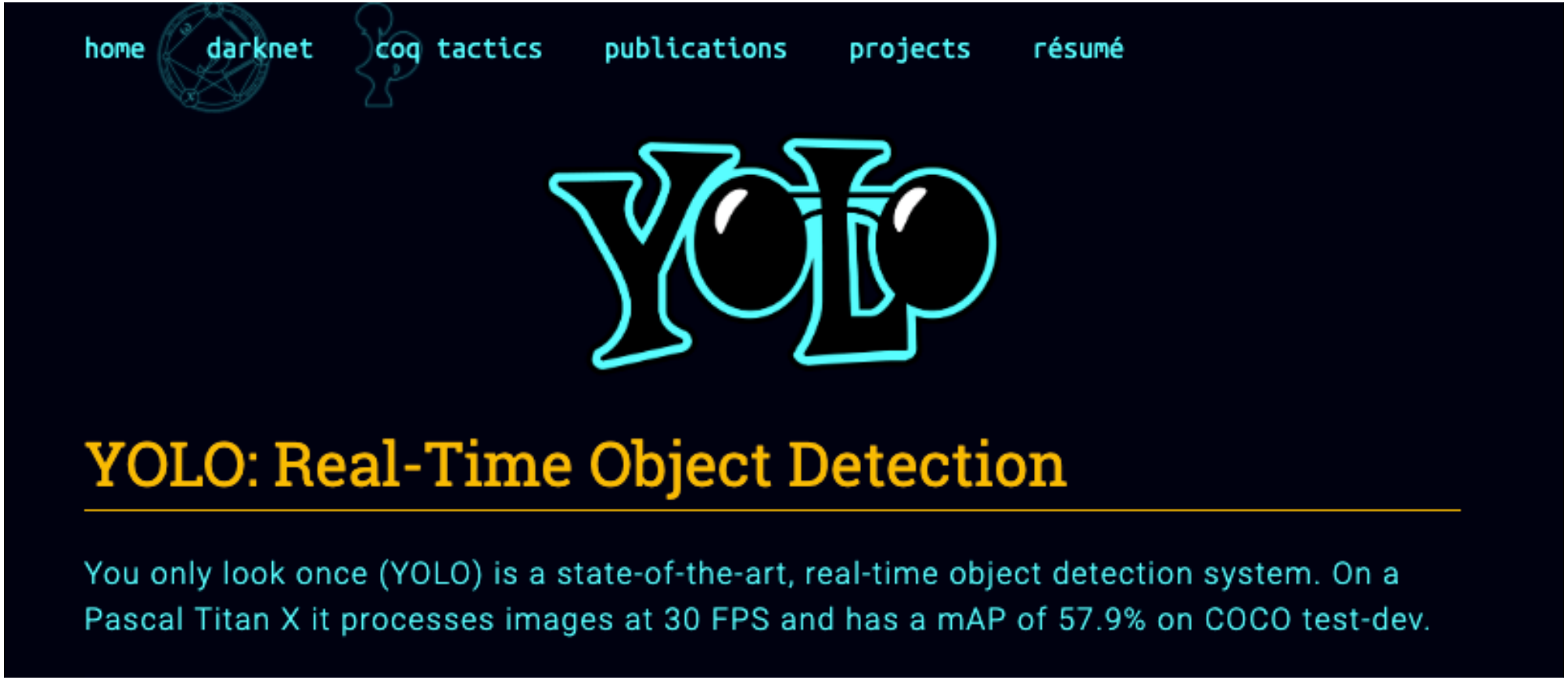
Python 深度学习

深度学习实战：实战 Darknet YOLOv3 目标检测

讲师：彭靖田

Darknet 高性能神经网络框架

- Darknet
- Darknet YOLO 实时目标检测框架



Darknet 使用预训练模型检测

```
git clone https://github.com/pjreddie/darknet
cd darknet
make
```

Easy!

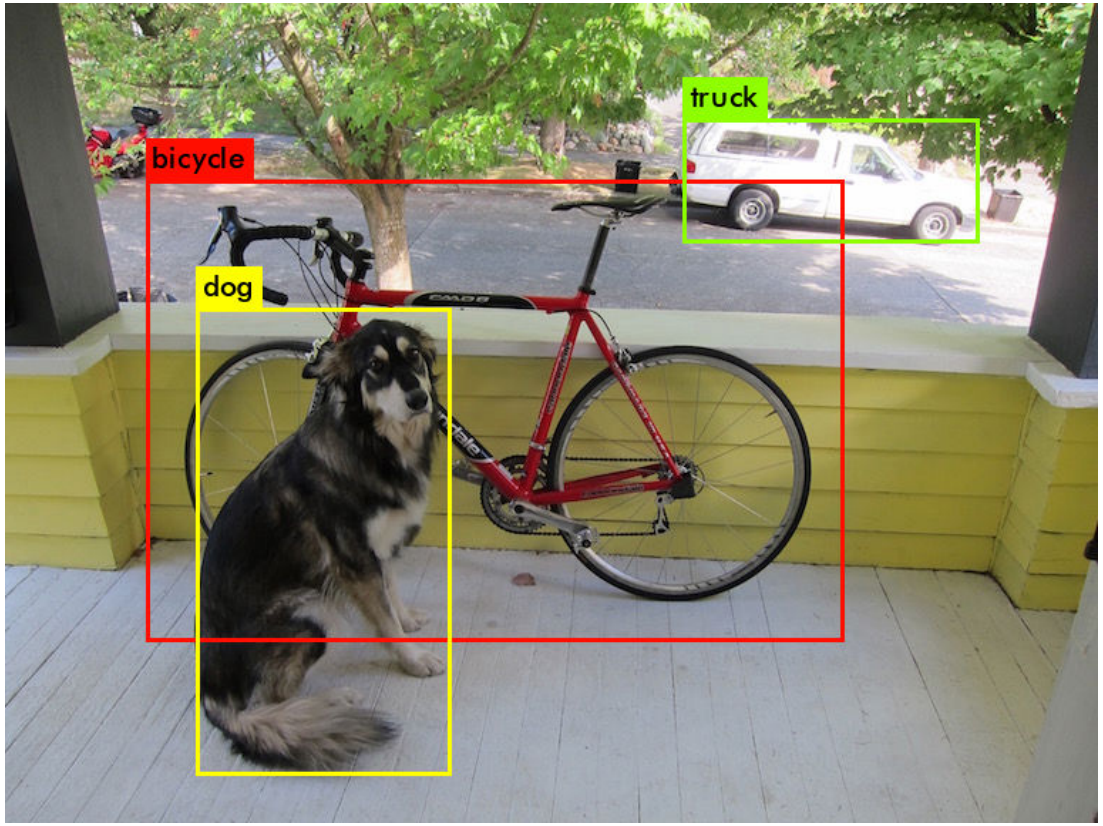
You already have the config file for YOLO in the `cfg/` subdirectory. You will have to download the pre-trained weight file here (237 MB). Or just run this:

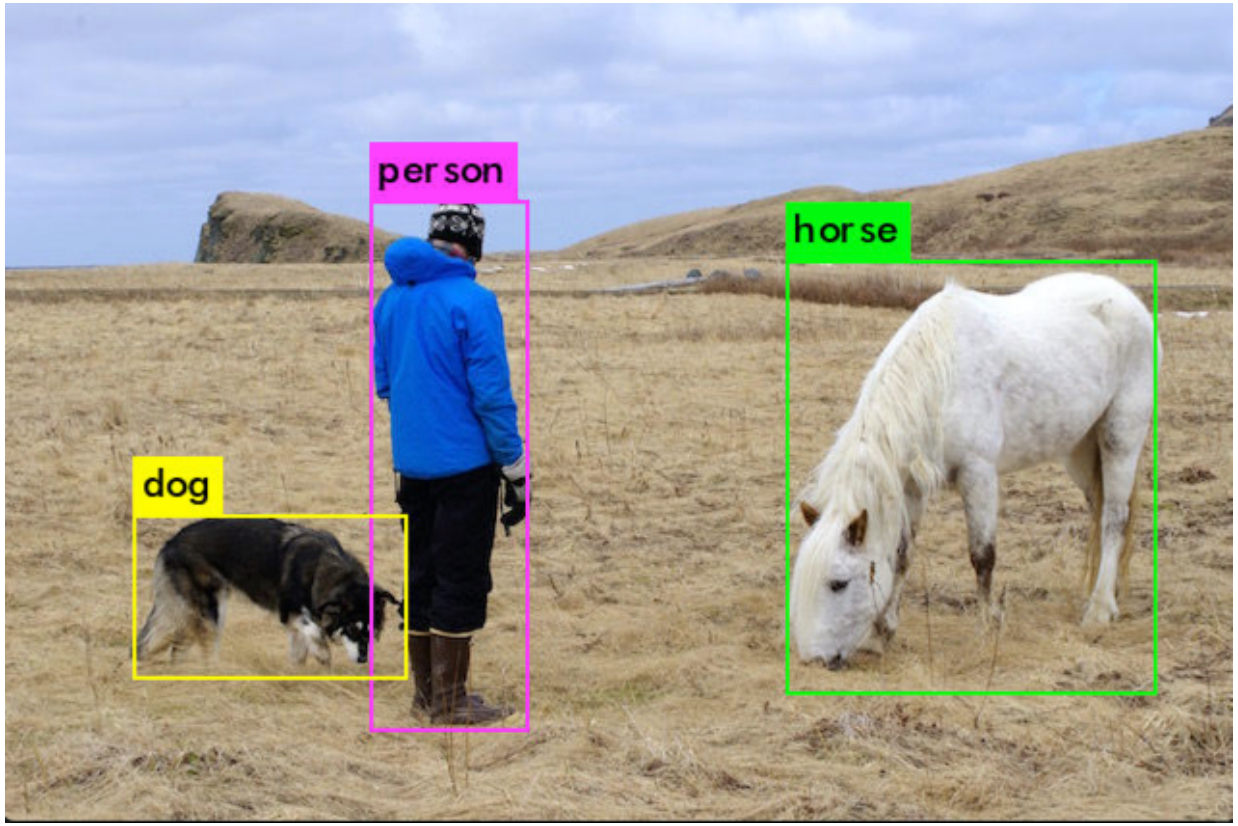
```
wget https://pjreddie.com/media/files/yolov3.weights
```

Then run the detector!

```
./darknet detect cfg/yolov3.cfg yolov3.weights data/dog.jpg
```

Darknet 使用预训练模型检测





Darknet 多张图像检测

Multiple Images

Instead of supplying an image on the command line, you can leave it blank to try multiple images in a row. Instead you will see a prompt when the config and weights are done loading:

```
./darknet detect cfg/yolov3.cfg yolov3.weights
layer     filters    size              input                output
    0 conv     32  3 x 3 / 1   416 x 416 x   3   ->   416 x 416 x  32  0.299 BFLOPs
    1 conv     64  3 x 3 / 2   416 x 416 x  32   ->   208 x 208 x  64  1.595 BFLOPs
    .......
  104 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256  1.595 BFLOPs
  105 conv    255  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 255  0.353 BFLOPs
  106 detection
Loading weights from yolov3.weights...Done!
Enter Image Path:
```

Enter an image path like `data/horses.jpg` to have it predict boxes for that image.

EDU

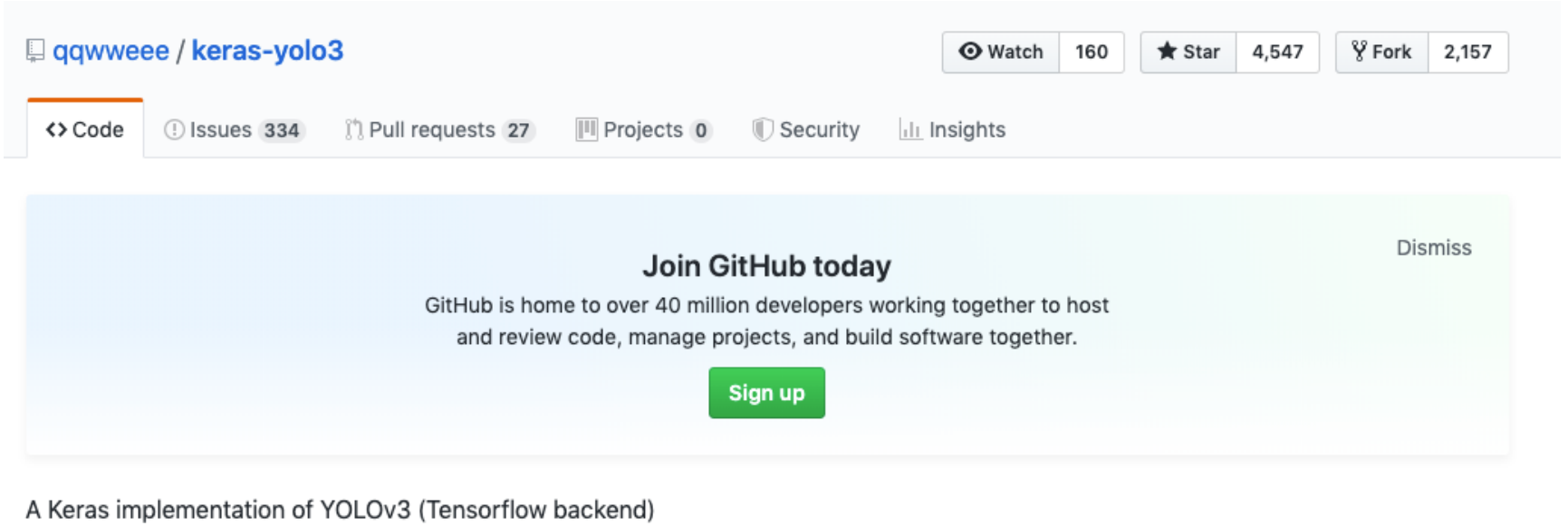
CSDN学院　IT实战派

Python 深度学习

深度学习实战：实战 Keras YOLOv3 目标检测

讲师：彭靖田

YOLOv3 Keras 版本



模型转换：将 Darknet 训练的模型权重转换为 Keras h5 模型文件

```
git clone https://github.com/qqwweee/keras-yolo3.git

wget https://pjreddie.com/media/files/yolov3.weights
python convert.py yolov3.cfg yolov3.weights model_data/yolo.h5
python yolo_video.py [OPTIONS...] --image, for image detection mode, OR
python yolo_video.py [video_path] [output_path (optional)]
```

https://github.com/qqwweee/keras-yolo3

Keras YOLOv3目标检测（图像）

```
(base) django:~/projects/DjangoPeng/dl-in-python/notebook_samples/chapter-6/keras-yolo3 (master) $ python yolo_video.py --image
Using TensorFlow backend.
Image detection mode
 Ignoring remaining command line arguments: ./path2your_video,
2019-10-20 17:53:50.899847: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow b
inary was not compiled to use: AVX2 FMA
WARNING:tensorflow:From /Users/django/anaconda3/lib/python3.7/site-packages/tensorflow/python/framework/op_def_library.py:263: colocat
e_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
model_data/yolo.h5 model, anchors, and classes loaded.
Input image filename:test/0.png
(416, 416, 3)
Found 11 boxes for img
truck 0.57 (626, 333) (715, 402)
car 0.32 (376, 196) (395, 210)
car 0.34 (409, 210) (428, 225)
car 0.44 (624, 334) (716, 398)
car 0.45 (398, 210) (417, 226)
car 0.47 (282, 190) (301, 206)
car 0.47 (379, 207) (398, 225)
car 0.85 (429, 235) (452, 256)
car 0.96 (321, 279) (360, 321)
car 0.98 (343, 242) (375, 268)
car 0.99 (500, 304) (558, 355)
2.5877315460000005
Input image filename:
```

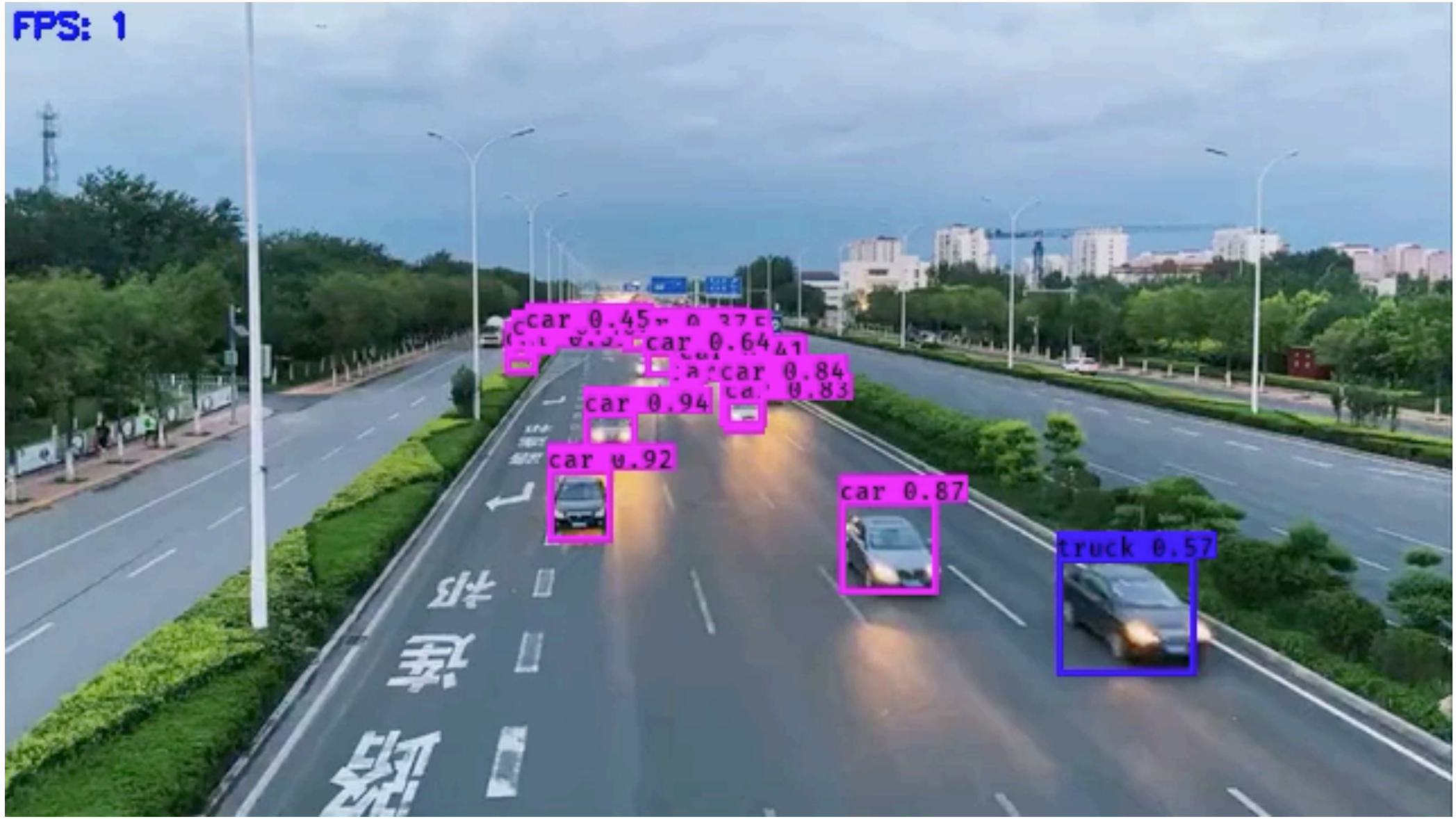
Keras YOLOv3目标检测（图像）

Keras YOLOv3目标检测（视频）

```
(base) django:~/projects/DjangoPeng/dl-in-python/notebook_samples/chapter-6/keras-yolo3 (master) $ python yolo_video.py
 --input test/1.mov --output test/1_out.mov
Using TensorFlow backend.
2019-10-20 18:01:04.825226: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that th
is TensorFlow binary was not compiled to use: AVX2 FMA
WARNING:tensorflow:From /Users/django/anaconda3/lib/python3.7/site-packages/tensorflow/python/framework/op_def_library.
py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
model_data/yolo.h5 model, anchors, and classes loaded.
!!! TYPE: <class 'str'> <class 'int'> <class 'float'> <class 'tuple'>
(416, 416, 3)
Found 14 boxes for img
truck 0.57 (465, 245) (526, 297)
car 0.31 (221, 152) (235, 164)
car 0.35 (282, 147) (294, 156)
car 0.37 (273, 146) (285, 154)
car 0.40 (224, 147) (236, 158)
car 0.41 (294, 167) (308, 180)
car 0.41 (298, 158) (310, 168)
car 0.45 (230, 144) (243, 155)
car 0.64 (283, 154) (296, 165)
car 0.83 (318, 175) (335, 190)
car 0.84 (316, 167) (336, 187)
car 0.87 (369, 220) (412, 261)
car 0.92 (240, 206) (268, 238)
car 0.94 (256, 181) (279, 199)
2.625074355999999
```

```
(416, 416, 3)
Found 16 boxes for img
truck 0.51 (218, 312) (263, 360)
truck 0.59 (14, 309) (90, 360)
truck 0.62 (425, 280) (477, 317)
bus 0.35 (20, 311) (88, 360)
car 0.32 (259, 197) (267, 205)
car 0.44 (224, 203) (235, 215)
car 0.48 (364, 207) (380, 221)
car 0.49 (218, 312) (263, 360)
car 0.51 (622, 243) (639, 257)
car 0.56 (436, 284) (478, 318)
car 0.56 (297, 196) (309, 204)
car 0.70 (242, 206) (257, 218)
car 0.87 (269, 219) (288, 234)
car 0.89 (217, 239) (245, 263)
car 0.90 (369, 214) (386, 226)
car 0.96 (126, 248) (162, 280)
0.8398972850000064
(416, 416, 3)
Found 13 boxes for img
truck 0.49 (371, 242) (399, 262)
car 0.31 (344, 226) (365, 244)
car 0.39 (371, 242) (399, 262)
car 0.41 (211, 206) (227, 223)
car 0.43 (622, 243) (640, 257)
car 0.57 (257, 202) (270, 212)
car 0.65 (210, 214) (225, 229)
car 0.78 (0, 326) (67, 360)
car 0.79 (266, 214) (287, 228)
car 0.92 (237, 215) (258, 230)
car 0.95 (275, 237) (302, 263)
car 0.95 (341, 202) (358, 213)
car 0.95 (191, 304) (240, 357)
0.8728487460000025
(416, 416, 3)
Found 8 boxes for img
car 0.32 (245, 195) (259, 204)
car 0.55 (350, 228) (375, 250)
car 0.70 (333, 198) (349, 208)
car 0.75 (202, 213) (222, 235)
```

Keras YOLOv3目标检测（视频）



Try it

EDU

CSDN学院　IT实战派

Python 深度学习

深度学习实战：实战 YOLOv3 迁移学习模型训练

讲师：彭靖田

数据准备：VOC 2007

The PASCAL Visual Object Classes Challenge 2007

[click on an image to see the annotation]

Development Kit

The development kit provided for the VOC challenge 2007 is available. You can:

- Download the training/validation data (450MB tar file)
- Download the development kit code and documentation (250KB tar file)
- Download the PDF documentation (120KB PDF)
- Browse the HTML documentation
- View the guidelines used for annotating the database

The updated development kit made available 11-Jun-2007 contains two changes:

- Bug fix: There was an error in the `VOCevaldet` function affecting the accuracy of precision/recall curve calculation. Pleas
- Changes to the location of results files to support running on both VOC2007 and VOC2006 test sets, see below.

It should be possible to untar the updated development kit over the previous version with no adverse effects.

http://host.robots.ox.ac.uk/pascal/VOC/voc2007/

模型训练：YOLOv3 Keras版本

1. Generate your own annotation file and class names file.
   One row for one image;
   Row format: `image_file_path box1 box2 ... boxN`;
   Box format: `x_min,y_min,x_max,y_max,class_id` (no space).
   For VOC dataset, try `python voc_annotation.py`
   Here is an example:

```
path/to/img1.jpg 50,100,150,200,0 30,50,200,120,3
path/to/img2.jpg 120,300,250,600,2
...
```

2. Make sure you have run `python convert.py -w yolov3.cfg yolov3.weights model_data/yolo_weights.h5`
   The file model_data/yolo_weights.h5 is used to load pretrained weights.

3. Modify train.py and start training.
   `python train.py`
   Use your trained weights or checkpoint weights with command line option `--model model_file` when using yolo_video.py Remember to modify class path or anchor path, with `--classes class_file` and `--anchors anchor_file`.

EDU

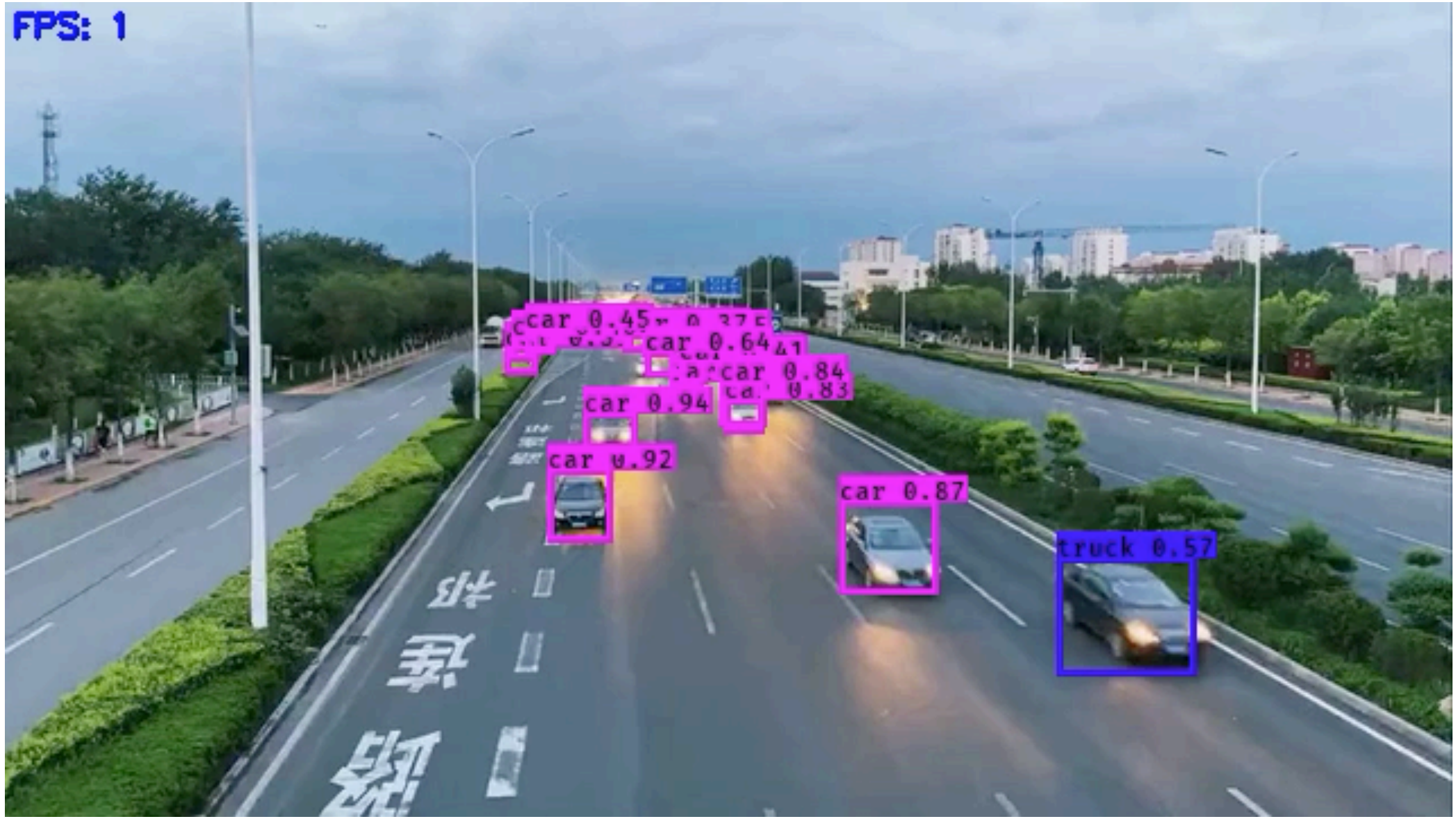
CSDN学院　IT实战派

Python 深度学习

深度学习实战：实战短小精干的 Tiny-YOLOv3 目标检测

讲师：彭靖田

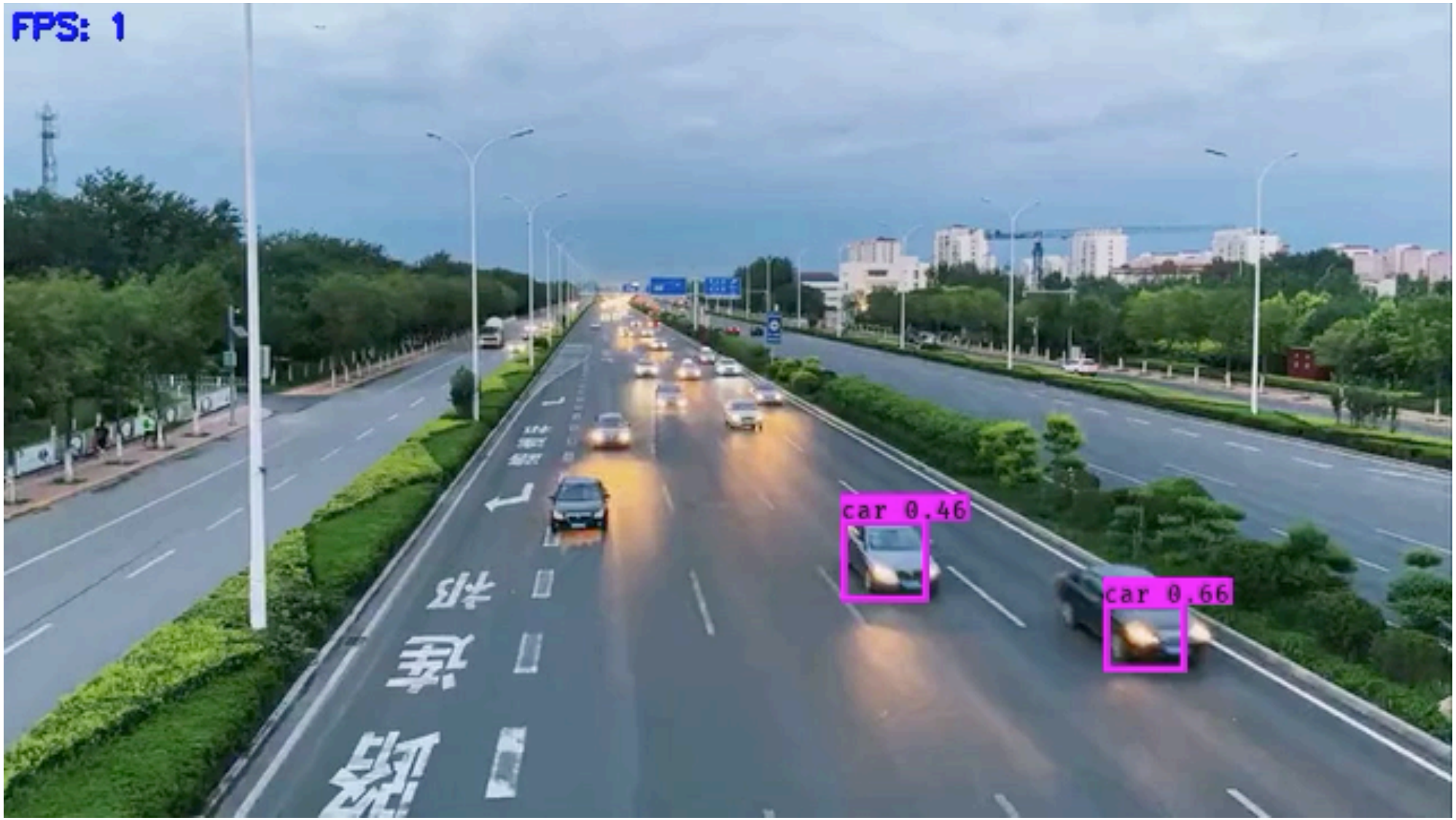
Keras YOLOv3目标检测（视频）



Tiny YOLOv3 在COCO 数据集上的性能

SSD321	COCO trainval	test-dev	45.4	-	16		link
DSSD321	COCO trainval	test-dev	46.1	-	12		link
R-FCN	COCO trainval	test-dev	51.9	-	12		link
SSD513	COCO trainval	test-dev	50.4	-	8		link
DSSD513	COCO trainval	test-dev	53.3	-	6		link
FPN FRCN	COCO trainval	test-dev	59.1	-	6		link
Retinanet-50-500	COCO trainval	test-dev	50.9	-	14		link
Retinanet-101-500	COCO trainval	test-dev	53.1	-	11		link
Retinanet-101-800	COCO trainval	test-dev	57.5	-	5		link
YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45	cfg	weights
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35	cfg	weights
YOLOv3-608	COCO trainval	test-dev	57.9	140.69 Bn	20	cfg	weights
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg	weights
YOLOv3-spp	COCO trainval	test-dev	60.6	141.45 Bn	20	cfg	weights

Keras Tiny YOLOv3目标检测（视频）



8x

Tiny YOLOv3 Keras 版本

```
$ wget https://pjreddie.com/media/files/yolov3-tiny.weights
$ python convert.py yolov3-tiny.cfg yolov3-tiny.weights model_data/yolov3-tiny.h5
Using TensorFlow backend.
Loading weights.
Weights Header:  0 2 0 [32013312]
Parsing Darknet config.
Creating Keras model.
...
Parsing section yolo_1
Model: "model_1"
_________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=========================================================================================
input_1 (InputLayer)         (None, None, None, 3 0
_________________________________________________________________________________________
conv2d_1 (Conv2D)            (None, None, None, 1 432         input_1[0][0]
_________________________________________________________________________________________
...
=========================================================================================
Total params: 8,858,734
Trainable params: 8,852,366
Non-trainable params: 6,368
_________________________________________________________________________________________
None
Saved Keras model to model_data/yolov3-tiny.h5
Read 8858734 of 8858734.0 from Darknet weights.
```

Keras Tiny YOLOv3目标检测（视频）

```
(base) django:~/projects/DjangoPeng/dl-in-python/notebook_samples/chapter-6/keras-yol
o3 (master) $ python yolo_video.py --input test/1.mov --output test/1_tiny_out.mov
Using TensorFlow backend.
2019-10-22 01:39:28.736790: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your
 CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
FMA
WARNING:tensorflow:From /Users/django/anaconda3/lib/python3.7/site-packages/tensorflo
w/python/framework/op_def_library.py:263: colocate_with (from tensorflow.python.frame
work.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
model_data/yolov3-tiny.h5 model, anchors, and classes loaded.
!!! TYPE: <class 'str'> <class 'int'> <class 'float'> <class 'tuple'>
(416, 416, 3)
Found 2 boxes for img
car 0.46 (370, 228) (408, 264)
car 0.66 (486, 265) (522, 295)
1.3317983030000002
(416, 416, 3)
Found 0 boxes for img
0.1023467759999992
(416, 416, 3)
Found 1 boxes for img
car 0.32 (220, 265) (262, 303)
0.09065024700000102
(416, 416, 3)
```

```
(416, 416, 3)
Found 0 boxes for img
0.10038235800000095
(416, 416, 3)
Found 1 boxes for img
car 0.54 (504, 316) (564, 354)
0.11232899699999876
(416, 416, 3)
Found 2 boxes for img
car 0.48 (121, 282) (163, 315)
car 0.81 (433, 274) (469, 304)
0.10716210500000045
(416, 416, 3)
Found 2 boxes for img
car 0.30 (91, 312) (140, 355)
car 0.45 (392, 254) (427, 283)
0.1117531600000028
(416, 416, 3)
Found 0 boxes for img
0.1070845889999994
(416, 416, 3)
Found 0 boxes for img
0.108504516
(416, 416, 3)
Found 1 boxes for img
car 0.33 (434, 260) (472, 286)
0.10148048299999957
(416, 416, 3)
Found 1 boxes for img
car 0.60 (208, 297) (240, 336)
0.10273577799999956
(416, 416, 3)
Found 0 boxes for img
0.09961098900000209
```

Try it

EDU

CSDN学院　IT实战派

Python 深度学习

深度学习实战：实战 Tiny-YOLOv3 迁移学习模型训练

讲师：彭靖田

转换 Tiny YOLOv3 预训练权重文件

```
(base) django:~/projects/DjangoPeng/dl-in-python/notebook_samples/chapter-6/keras-yolo3 (master) $
python convert.py -w yolov3-tiny.cfg yolov3-tiny.weights model_data/yolov3-tiny_weights.h5
Using TensorFlow backend.
Loading weights.
Weights Header:  0 2 0 [32013312]
Parsing Darknet config.
Creating Keras model.
Parsing section net_0
Parsing section convolutional_0
conv2d bn leaky (3, 3, 3, 16)
WARNING:tensorflow:From /Users/django/anaconda3/lib/python3.7/site-packages/tensorflow/python/frame
work/op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and
will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
2020-03-03 23:39:56.673248: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports
instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
Parsing section maxpool_0
Parsing section convolutional_1
conv2d bn leaky (3, 3, 16, 32)
Parsing section maxpool_1
Parsing section convolutional_2
conv2d bn leaky (3, 3, 32, 64)
```

模型训练：Tiny-YOLOv3 Keras版本

Try it


EDU
CSDN学院 IT实战派