



西安交通大学

XI'AN JIAOTONG UNIVERSITY

人工智能与机器人研究所

Institute of Artificial Intelligence and Robotics



计算机视觉与模式识别 (CVPR)

课程实验——目标检测

西安交通大学人工智能学院

实验地点：计算机视觉与深度学习实验室（科学馆204）

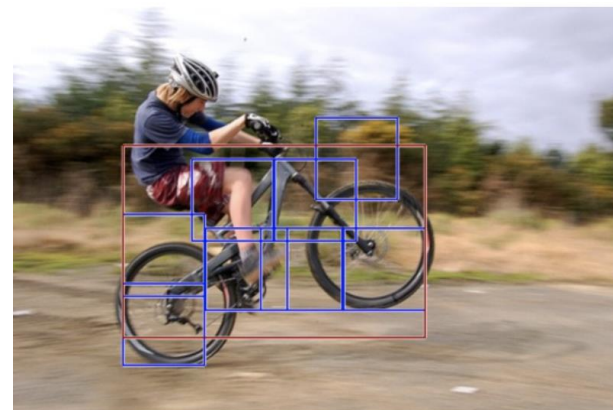
指导教师：姚慧敏

Email: hmyao267@mail.xjtu.edu.cn

Homework 5

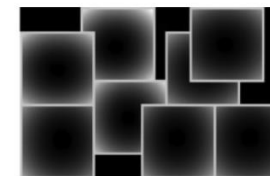
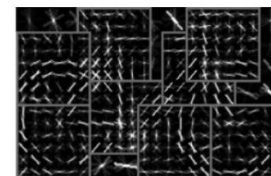
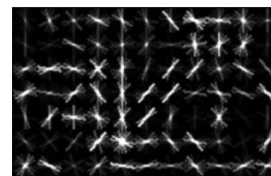
一、传统目标检测方法与基于深度学习的目标检测方法比较

- ✓ 以PASCAL VOC 2012 Dataset为实验数据集，按照原始数据集的要求进行训练/验证/测试集的划分
- ✓ 以3-5个人的小组完成任务
- ✓ 理解并复现传统目标检测方法-可变形部件模型（DPM）方法¹
- ✓ 学习分析深度学习模型用于目标检测的方法与DPM的关系²
- ✓ 运行一种基于深度学习的目标检测方法
- ✓ 编写实验报告，包括对算法的简要描述、实验设定、实验结果等，篇幅不限



¹ <https://cs.brown.edu/people/pfelzens/latent-release4/>

² Ross Girshick, Forrest Iandola, Trevor Darrell, Jitendra Malik, "Deformable Part Models are Convolutional Neural Networks"

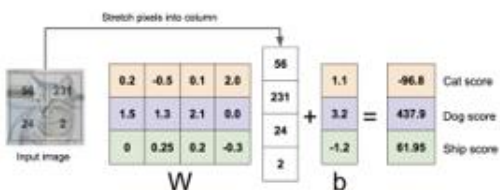


课程内容

总结：理解线性分类器的三种视角

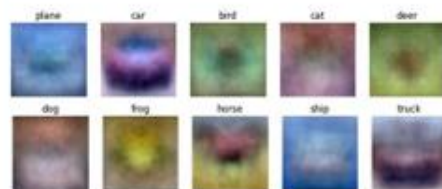
代数视角

$$f(x, W) = Wx$$



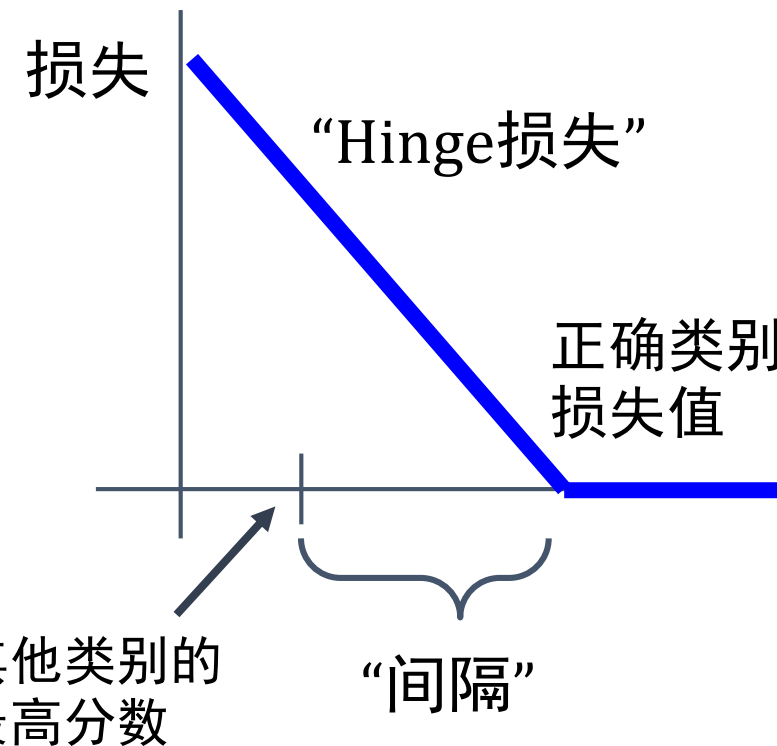
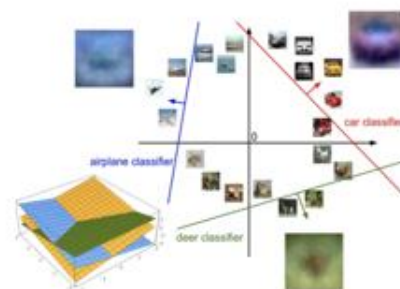
视觉视角

每类一个模板



几何视角

超平面分割空间

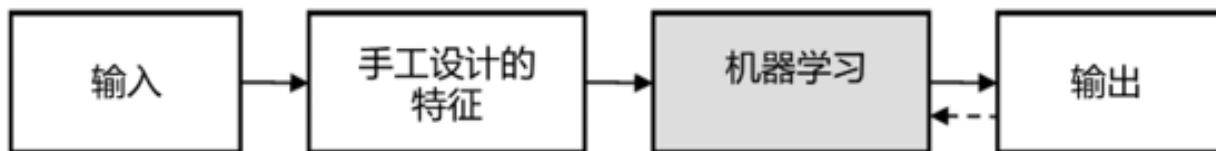
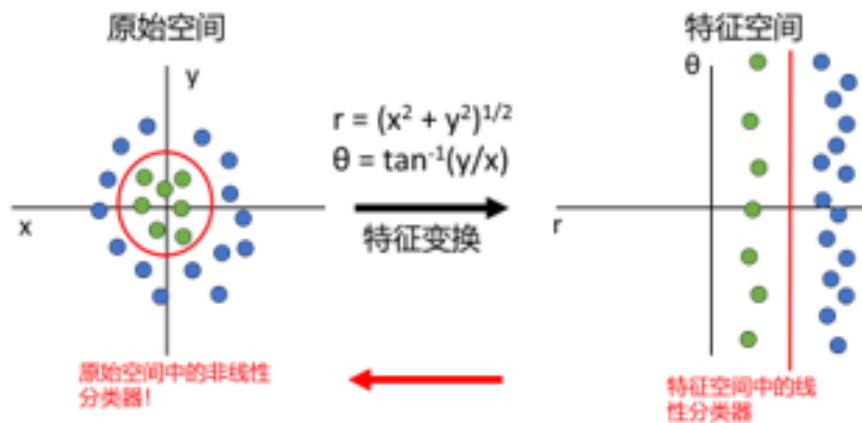


多类别SVM损失函数：

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

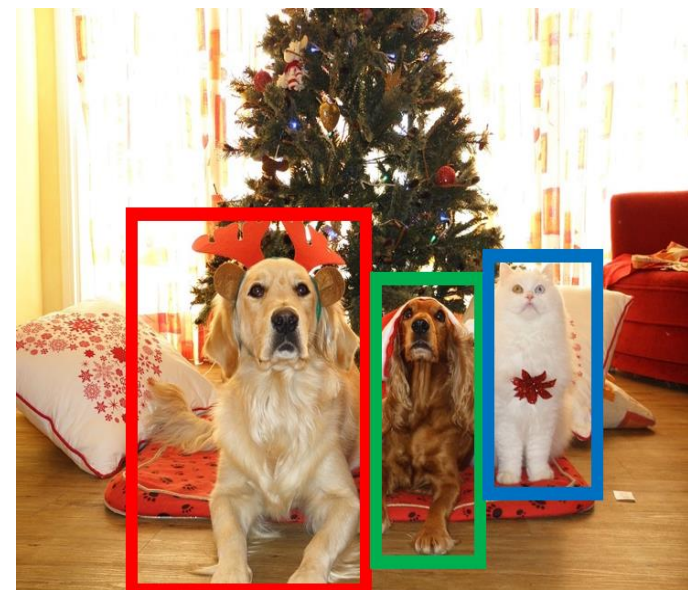
课程内容

特征变换 + 线性分类器允许出现非线性的决策边界



(b) 经典的机器学习处理流程 (pipeline)

目标检测



目录

- 一. 实验内容
- 二. 实验目的
- 三. 实验原理
- 四. 数据集与编程环境
- 五. 实验步骤及要求

一、实验内容

1. 目标检测——任务定义

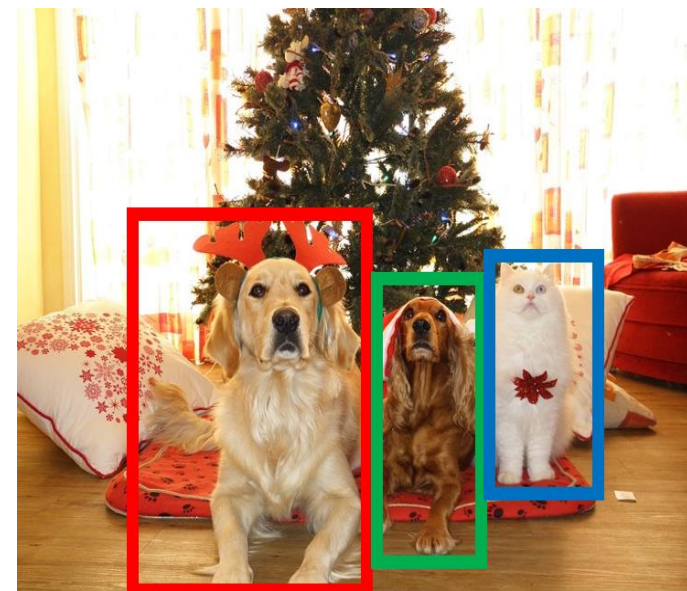
输入: 单张图像

输出: 一系列检测到的目标物;

对于每个目标预测:

1) 类别标签(来自已知不变的类别集)

2) 边界框 (位置: x, y ; 大小: w, h)



2. 方法: **特征+深度学习**、R-CNN系列、Yolo系列、Transformer等

实验内容: 用HOG特征+SVM设计一个单目标检测器。

二、实验目的

1. 掌握HOG特征描述子和线性SVM分类算法
2. 理解滑动窗扫描进行目标检测方法
3. 掌握交并比(IoU)的原理与作用
4. 理解召回率、检测精度的概念，掌握用均值平均精度(mAP)评估目标检测器的方法

三、实验原理（1/12）

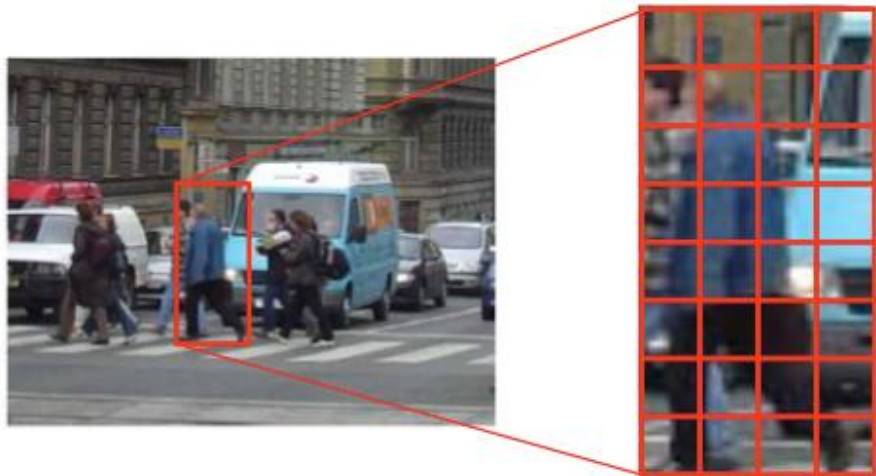
文献：方向梯度直方图（Histogram of Oriented Gradients, HOG）是法国人Dalal和Triggs在2005年CVPR会议上提出的特征提取算法，并将其与 SVM 配合，用于行人检测。

1. Dalal & Triggs. **Histograms of Oriented Gradients for Human Detection**, CVPR 2005
2. Navneet Dalal. **Finding People in Images and Videos**. Human-Computer Interaction [cs.HC]. Institut National Polytechnique de Grenoble - INPG, 2006. English.

方法概述：将检测窗口中的由cell构成的所有块的HOG描述子组合成最终的特征向量。然后使用SVM分类器进行目标和非目标的**二分类**（检测）。

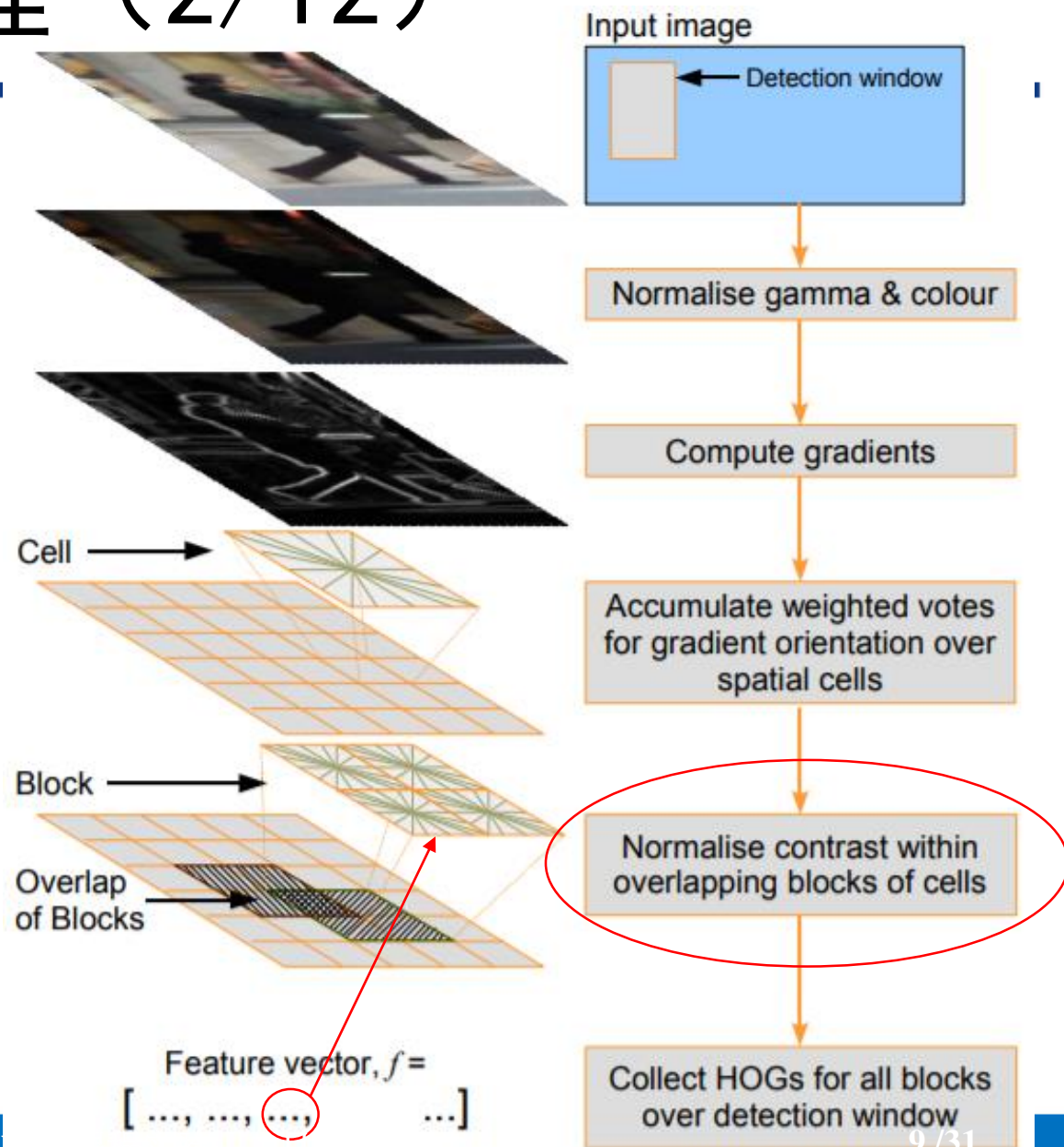
三、实验原理 (2/12)

方向梯度直方图HOG：描述子



例：检测窗口尺寸128×64；
cell大小为8×8；
统计直方图有9个bin；
block大小为2×2

检测窗口HOG特征向量的维度为：
 $(16-2+1) * (8-2+1) * (2 * 2 * 9) = 3780$

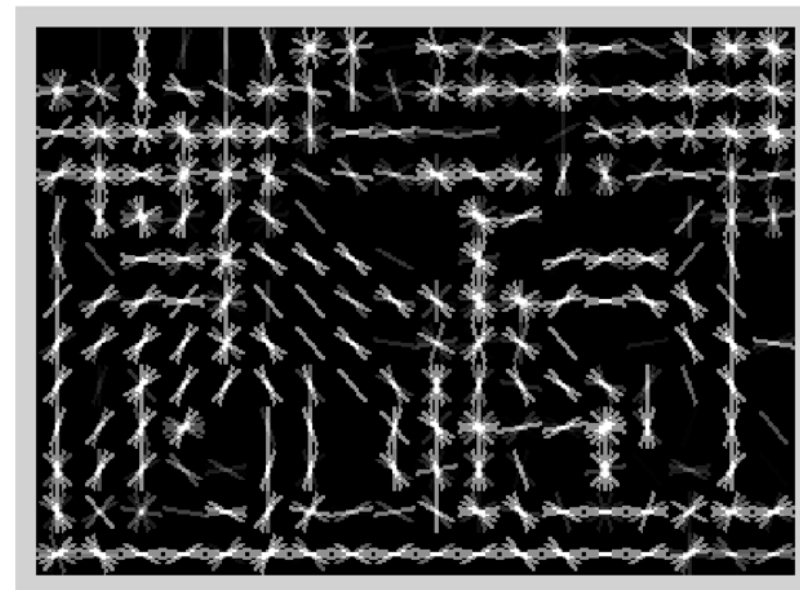


三、实验原理 (3/12)

方向梯度直方图HOG：可视化



图像 I



HOG特征图 H

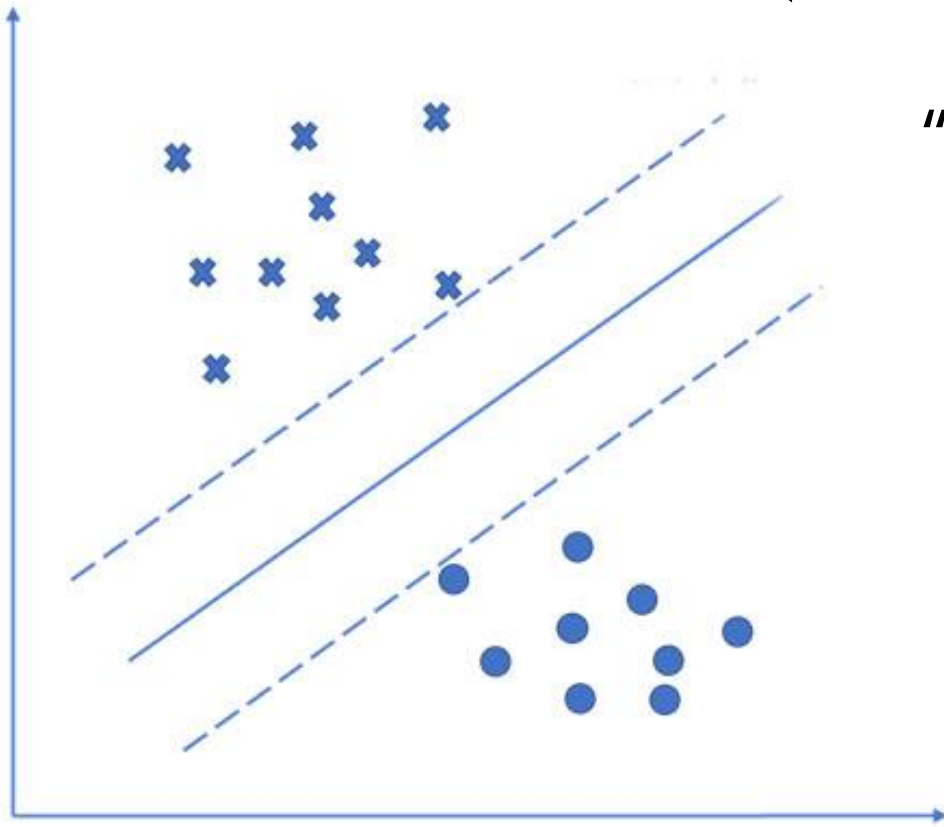
skimage.feature.hog

HOGgles: Visualizing Object Detection Features
<http://www.cs.columbia.edu/~vondrick/ihog/>

三、实验原理（4/12）

支持向量机(SVM):

$$f(\mathbf{x}, \mathbf{W}) = \mathbf{W}\mathbf{x} + \mathbf{b}$$



“Hinge损失” 函数形式:

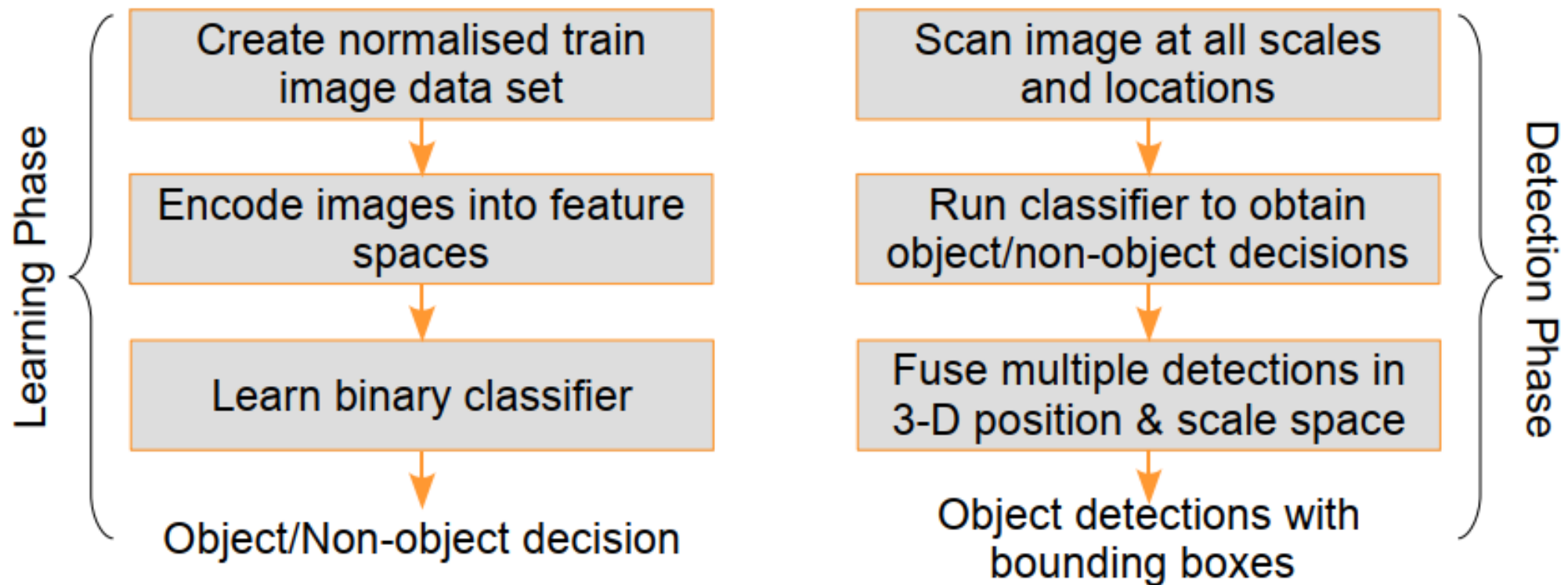
$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

或:
$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)^2$$

$$L(W) = \frac{1}{N} \sum_{i=1}^N L_i(f(x_i, W), y_i) + \lambda R(W)$$

三、实验原理 (5/12)

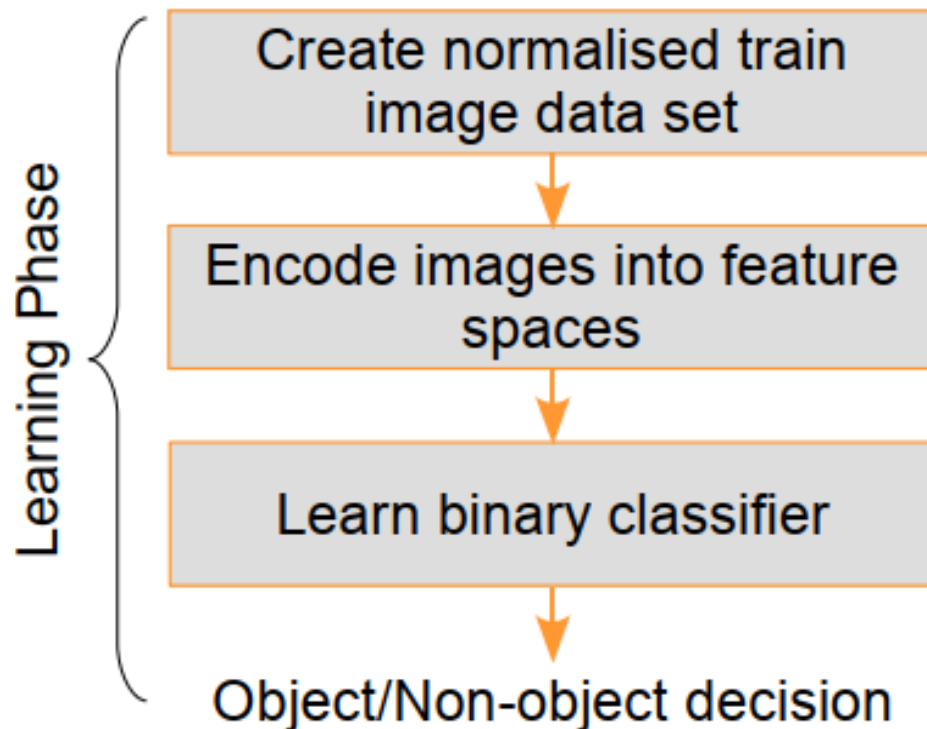
目标检测整体架构:



Navneet Dalal. Finding People in Images and Videos

三、实验原理（6/12）

模型学习：

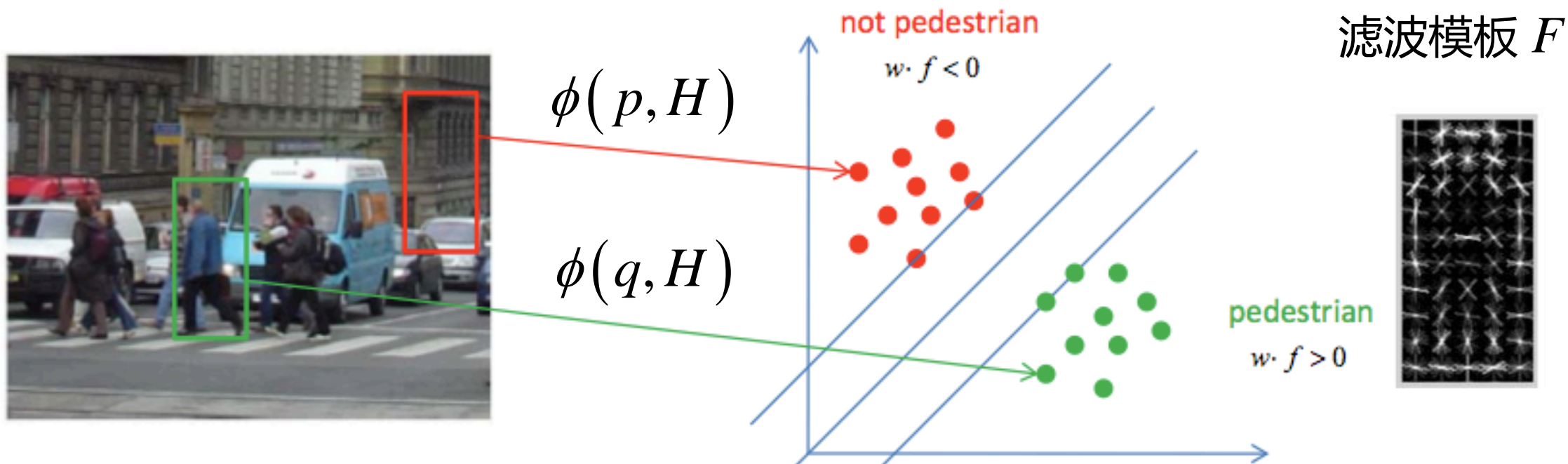


学习阶段：构造固定大小的正负样本训练集，从训练窗口中提取HOG视觉特征，并在它们之上训练一个二元（目标/目标）分类器。

Navneet Dalal. Finding People in Images and Videos

三、实验原理（7/12）

模型学习：

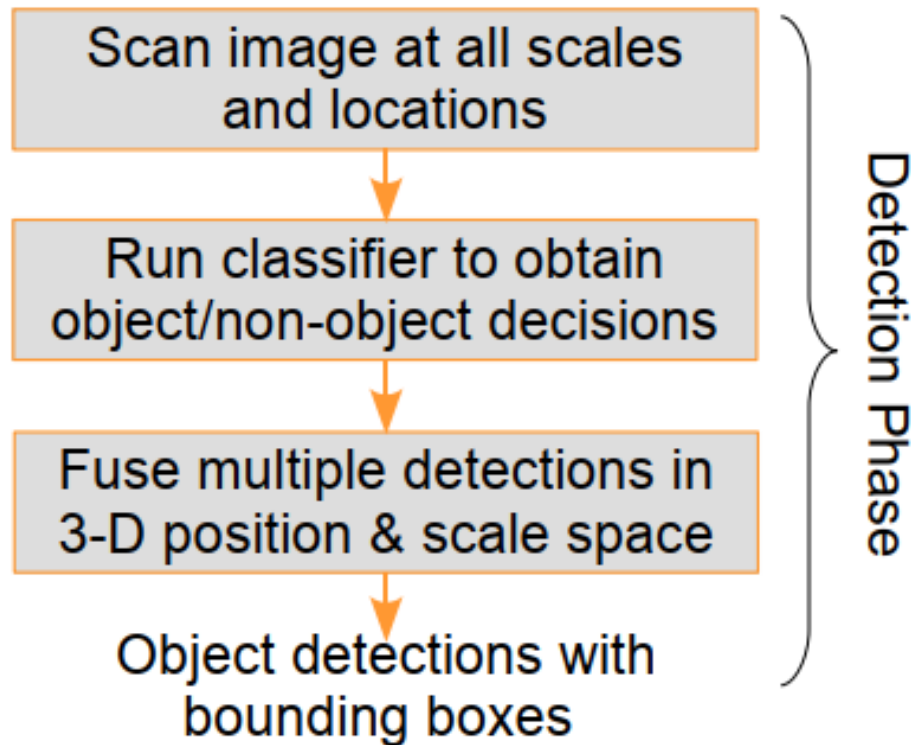


$\phi(p, H)$: 位于 p 点处检测窗口的HOG特征描述子

学习阶段：负样本数量远大于正样本数量

三、实验原理（8/12）

目标检测：

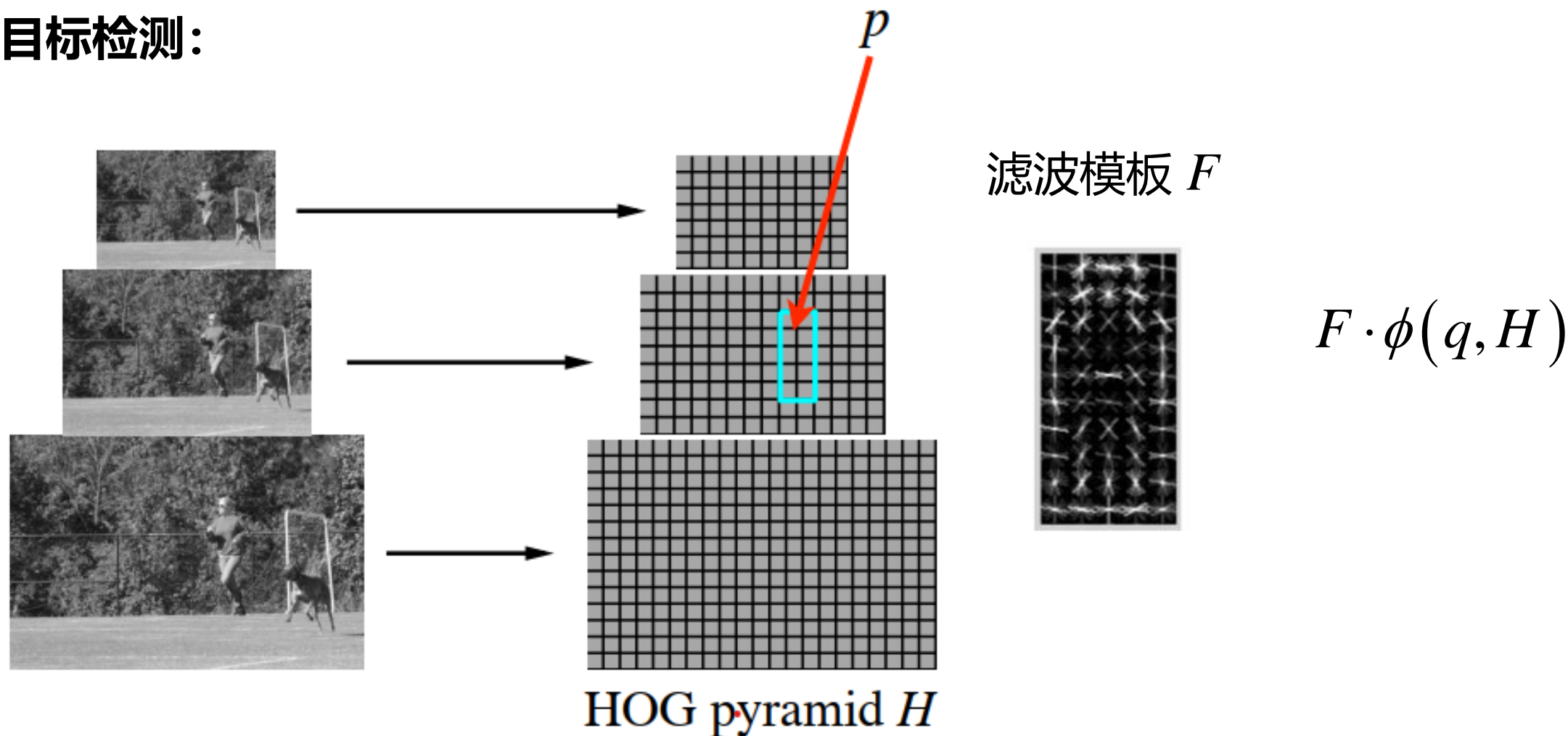


检测阶段：使用学习到的二元分类器在所有位置和尺度上，用与模板相同大小的窗口扫描测试图像，以进行目标/非目标决策。这些初步决定后来被融合以产生最终的物体检测。

Navneet Dalal. Finding People in Images and Videos

三、实验原理 (9/12)

目标检测:



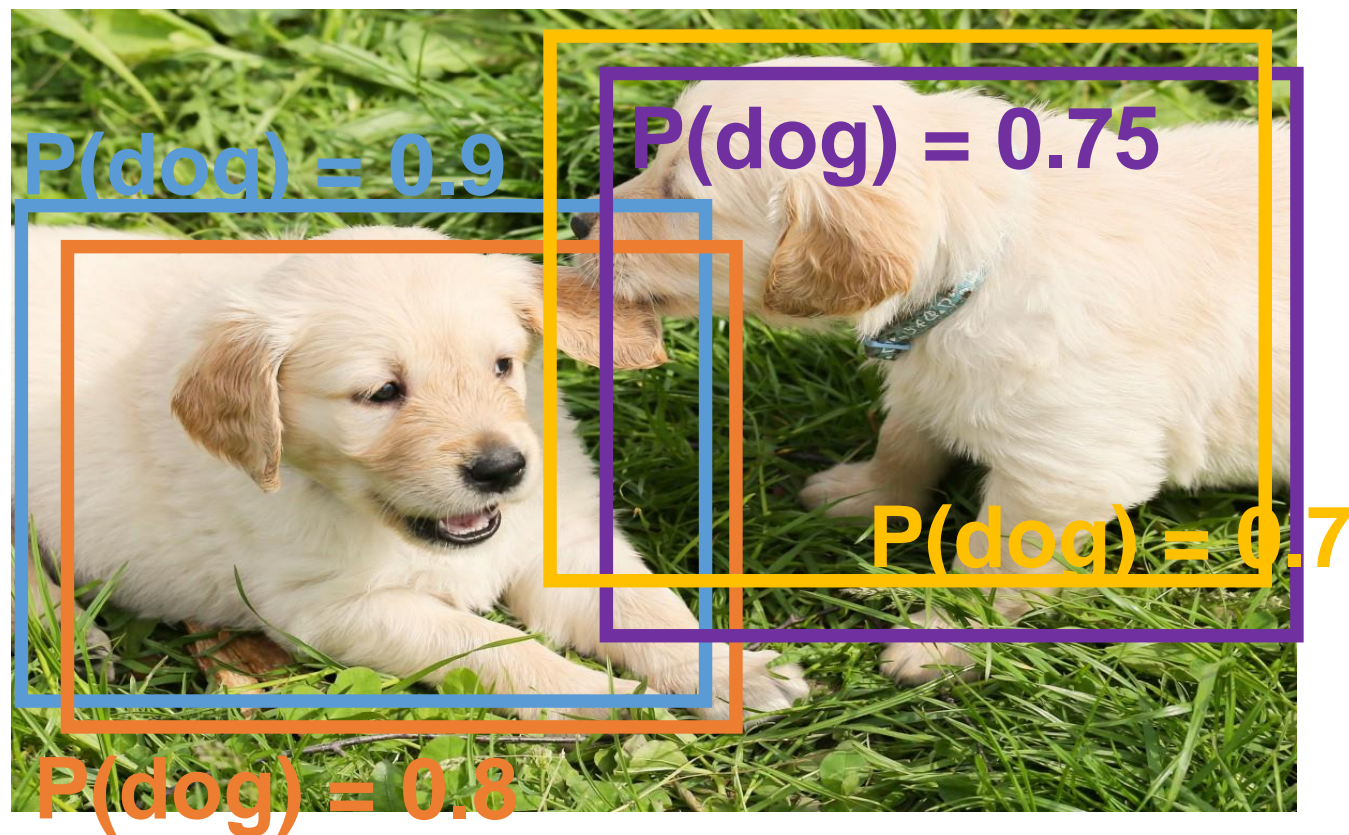
三、实验原理 (10/12)

问题：目标检测器经常输出许多重叠的检测框(交叠框)

$$IoU = \frac{\text{预测值与真实值重叠部分}}{\text{预测值与真实值集合部分}}$$

解决方法：利用**非最大抑制**
(NMS)对原始检测进行后处理

- 1.选择下一个得分更高的框
- 2.让**交并比IoU**大于阈值(如0.7)，消除低分的框
- 3.若还有框，转到步骤1



Puppy image is CC0 Public Domain

三、实验原理 (11/12)

评估目标检测器：均值平均精度(mAP)

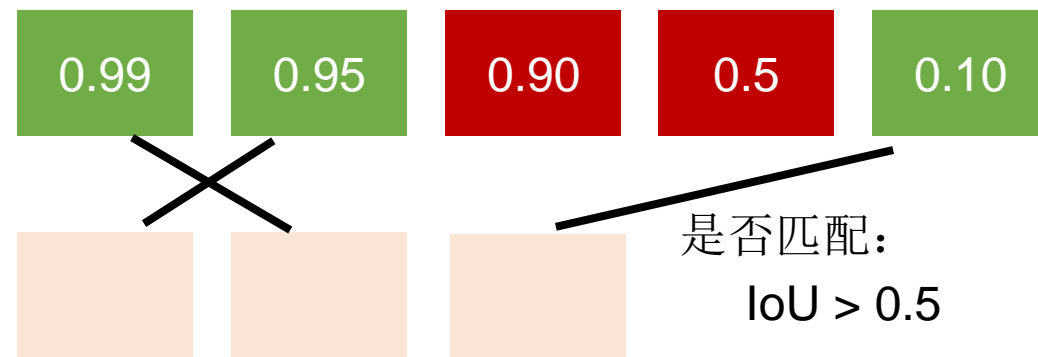
- 一、利用NMS在**所有测试图像**上运行目标检测器
- 二、对**每个类别**，计算平均精度(AP)

AP = 精度和召回率曲线下的面积

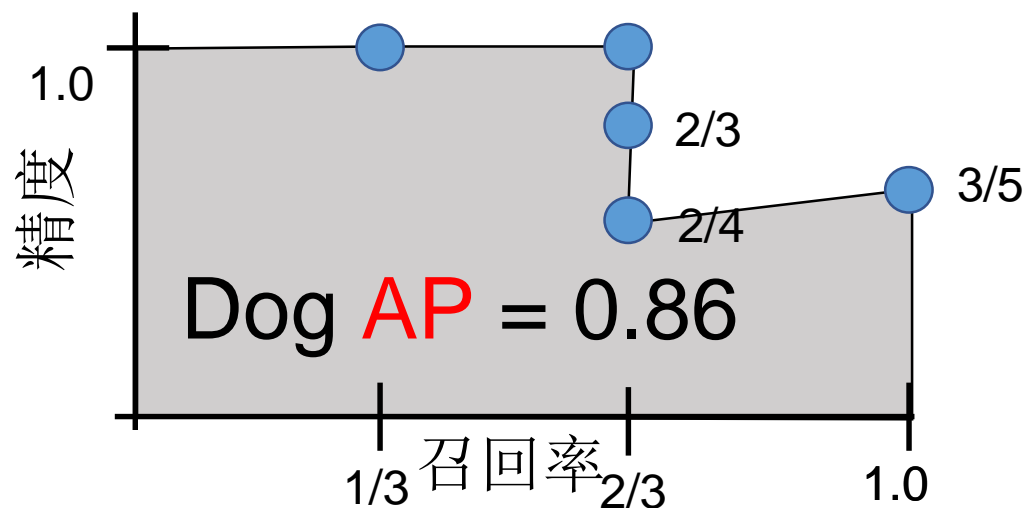
1. 对于每个检测(从最高分到最低分)
 - 1) 如果它能匹配一些真实值框(满足 $IoU > 0.5$)，则标记它为**正**然后消除这个真实值。
 - 2) 否则标记它为**负**
 - 3) 在精度召回率(PR)曲线上描点
2. 平均精度(AP) = PR曲线下的面积

如何获得 $AP = 1.0$: 命中所有 $IoU > 0.5$ 的真实值框，并且没有排名高于任何“真阳性”(TP)的“假阳性”(FP)检测

按分数排列所有狗的检测



所有真实的狗的边界框



三、实验原理 (12/12)

评估目标检测器：均值平均精度(mAP)

三、均值平均精度(mAP)=每个类的平均精度(AP)的均值

车 $AP = 0.65$

猫 $AP = 0.80$

狗 $AP = 0.86$

$mAP@0.5 = 0.77$

四、对于“COCO mAP”：计算每个 IoU 阈值 (0.5, 0.55, 0.6, ..., 0.95) 的 $mAP@$ 阈值 并取平均值

$mAP@0.5 = 0.77$

$mAP@0.55 = 0.71$

$mAP@0.60 = 0.65$

...

$mAP@0.95 = 0.2$

$COCO\ mAP = 0.4$

目录

一. 实验内容

二. 实验目的

三. 实验原理

四. 数据集与编程环境

五. 实验步骤及要求

四、数据集与编程环境（1/3）

数据集：UIUC车辆检测、INRIA行人检测

UIUC数据集：数据集中包含用于评估目标检测算法的**汽车侧视图**，由Shivani Agarwal, Aatif Awan和Dan Roth在UIUC收集整理的。



数据集中图像均为灰度图像，png格式，总共**1328张图片**，在**PNGImages**文件夹中，**Annotations**中是标记的汽车边界框位置的txt文件：

- 1050张**训练**图像（550张汽车图像和500张非汽车图像，图像大小40*100）
- 170张**单尺度测试**图像，其中包含200张与训练集汽车比例大致相同的图像
- 108张**多尺度测试**图像，包含139辆不同尺度的汽车

四、数据集与编程环境 (2/3)

数据集：UIUC车辆检测、INRIA行人检测

INRIA数据集：是 Dalal 在图像和视频中检测直立行人的研究工作中收集的,是一组有标记站立或行走的人(直立行人，身高>100)的图像。

- 70X134H96
- 96X160H96
- Test
- test_64x128_H96
- Train
- train_64x128_H96



数据集内有6个文件夹：

- Train 和Test文件夹分别对应于原始训练图像和测试图像。
- /96X160H96 和 /70X134H96 文件夹下的图片，是标准化正训练或测试图
- train_64x128_H96 和 test_64x128_H96无图片，需从上述四个文件夹中生成对应标准化的数据集，包括正样本和负样本。

四、数据集与编程环境（3/3）

实验环境： 编程语言不限

- Python: **scikit-learn**、**joblib**、**Opencv**、**matplotlib**等

1) 用HOG 特征+线性 SVM 分类器设计的目标检测器

<https://github.com/bikz05/object-detector>

2) 均值平均精度(mAP, mean Average Precision)

<https://github.com/Cartucho/mAP>

- C++
- Matlab

五、实验步骤及要求（1/6）

具体步骤：

- 1) 获取训练集的HOG特征
- 2) 训练模型
- 3) 用测试集测试训练的模型
- 4) 评估目标检测器

五、实验步骤及要求 (2/6)

1. 获取训练集中正负样本的HOG特征——extract-features.py

```
print("Calculating the descriptors for the negative samples and saving them")
for im_path in glob.glob(os.path.join(neg_im_path, "*")):
    im = cv2.imread(im_path, cv2.IMREAD_GRAYSCALE)
    if des_type == "HOG":
        fd = feature.hog(im, orientations=orientations, pixels_per_cell=pixels_per_cell,
                        cells_per_block=cells_per_block, visualize=visualize)
        fd_name = os.path.split(im_path)[1].split(".")[0] + ".feat"
        fd_path = os.path.join(neg_feat_ph, fd_name)
        joblib.dump(fd, fd_path)
print("Negative features saved in {}".format(neg_feat_ph))
```

读入图片

计算HOG特征

保存HOG特征

五、实验步骤及要求 (3/6)

2. 训练模型并保存——train-classifier.py

1) 读入正负样本的特征并给出标签

```
for feat_path in glob.glob(os.path.join(pos_feat_path, "*.feat")):  
    fd = joblib.load(feat_path)  
    fds.append(fd)  
    labels.append(1)
```

2) 通过正负样本训练SVM模型

clf = LinearSVC() 或 SVC(kernel='linear')

clf.fit(fds, labels)

joblib.dump(clf, model_path)

五、实验步骤及要求（4/6）

3.用测试集测试训练的模型——test-classifier.py (单幅图)

- ✓ 对每幅不同尺度下的图像进行滑动窗扫描，获取HOG特征并用分类器做分类。
- ✓ 如果检测为目标则用边界框框出。
- ✓ 图像扫描完成后应用非极大抑制(NMS)来消除重叠多余的目标。

```
in sliding_window(ir
```

五、实验步骤及要求 (5/6)

4.评估目标检测器——四个py文件

test-clf-full.py: 用模型对测试集数据进行检测，若检测出目标，则用边界框框出并输出坐标位置，输出格式为：

<class_name> <confidence> <left> <top> <right> <bottom>

convert_gt_txt.py: 从Annotations文件夹的标记文件中读取边界框的真实值，输出格式为：

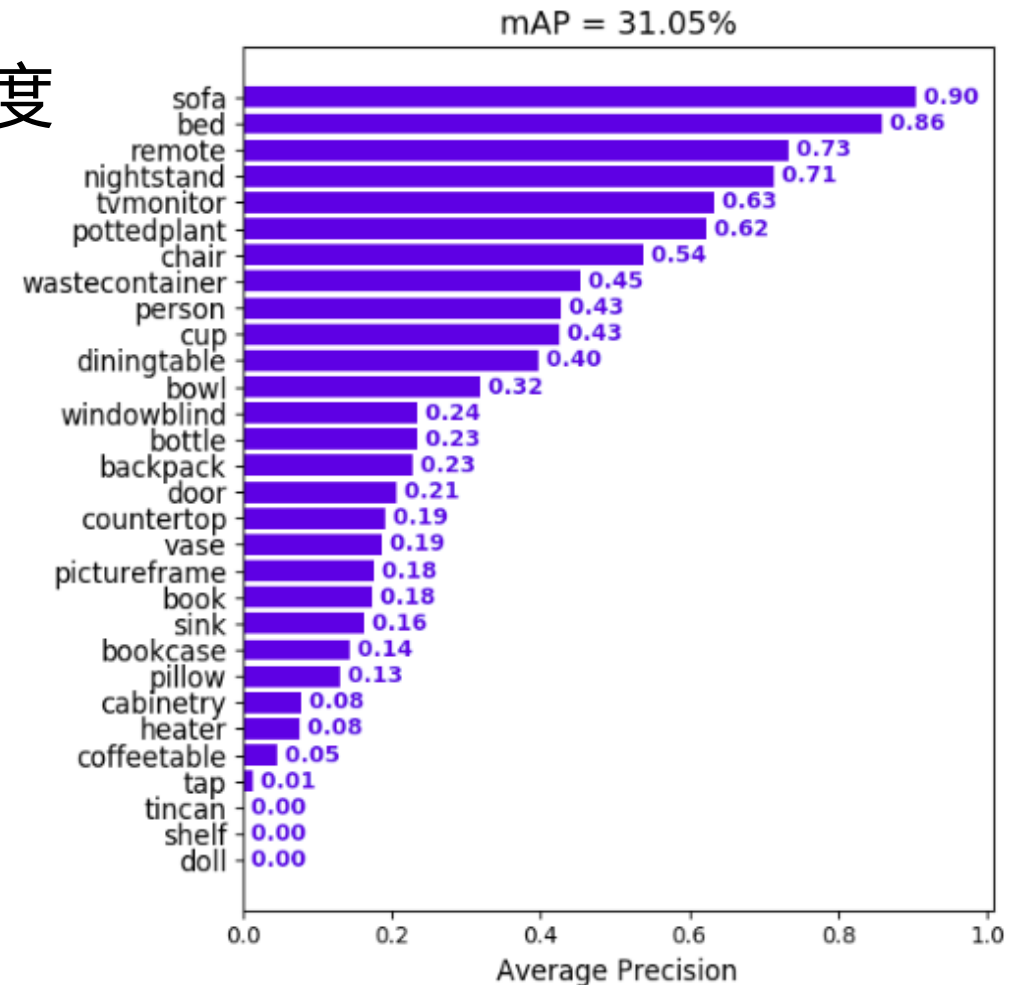
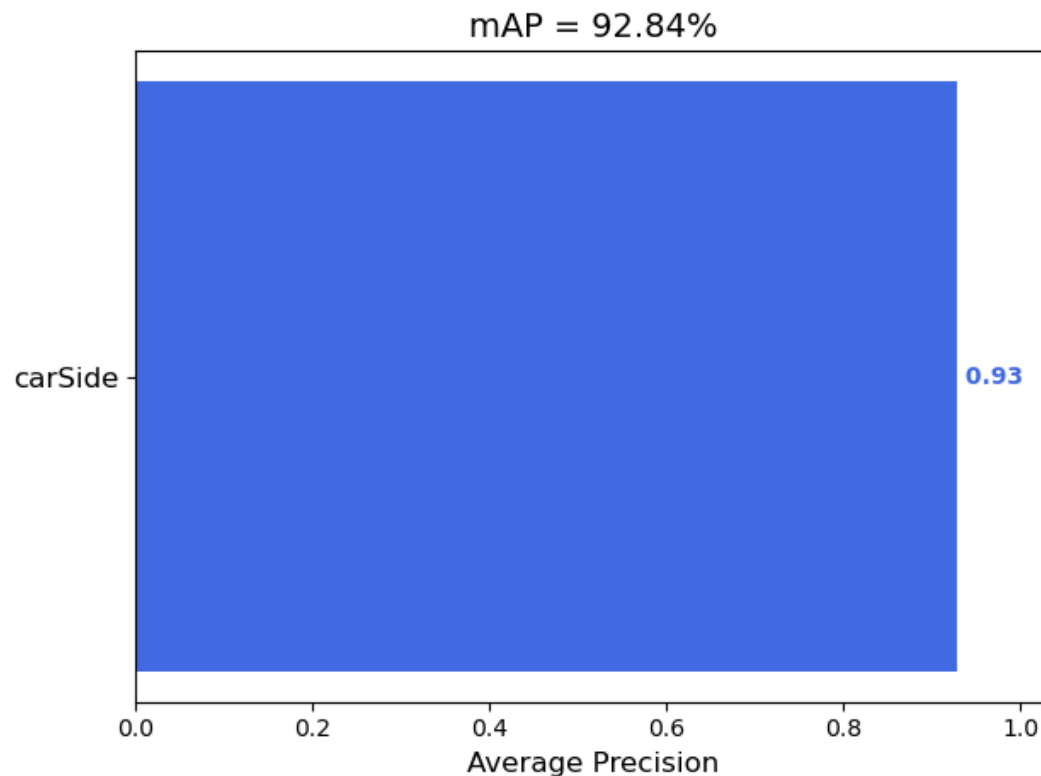
<class_name> <left> <top> <right> <bottom> [<difficult>]

intersect-gt-and-dr.py: 确保真值和检测结果文件夹中的文件数量相同，只保留两个文件夹中具有相同名称的文件。

五、实验步骤及要求 (6/6)

4.评估目标检测器——四个py文件

evaluate-mAP.py: 计算均值平均精度



小结

- **实验内容：**

用HOG特征+SVM分类器设计单目标检测器

- **实验原理：** 滑动窗扫描目标检测原理、mAP

- **编程工具：** Python、scikit-learn、joblib、Opencv等

- **实验步骤：** 获取训练集的HOG特征、训练模型
测试训练的模型、评估目标检测器

实验报告

1. 两周之内（12月16日之前）提交

2. 实验报告内容：

1) 简要描述

4) 实验结果

2) 实验内容

5) 结论与讨论

3) 实验过程

6) 附主要代码

3. 篇幅不限



西安交通大学

XI'AN JIAOTONG UNIVERSITY

人工智能与机器人研究所

Institute of Artificial Intelligence and Robotics



谢谢！