计算机视觉

第7章 目标检测

陈飞: chenfei314@fzu.edu.cn

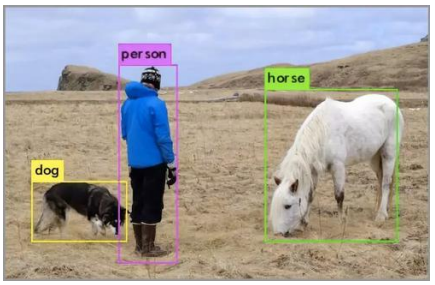


本章内容

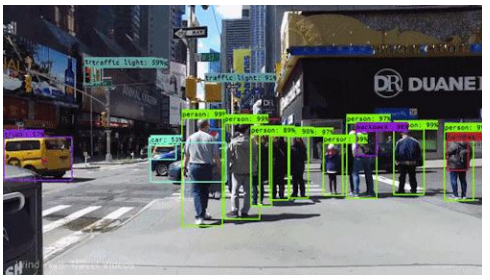
- 目标检测概述
- 经典目标检测算法
- 目标检测的发展
- 常用数据集
- 评价标准

7.1目标检测概述

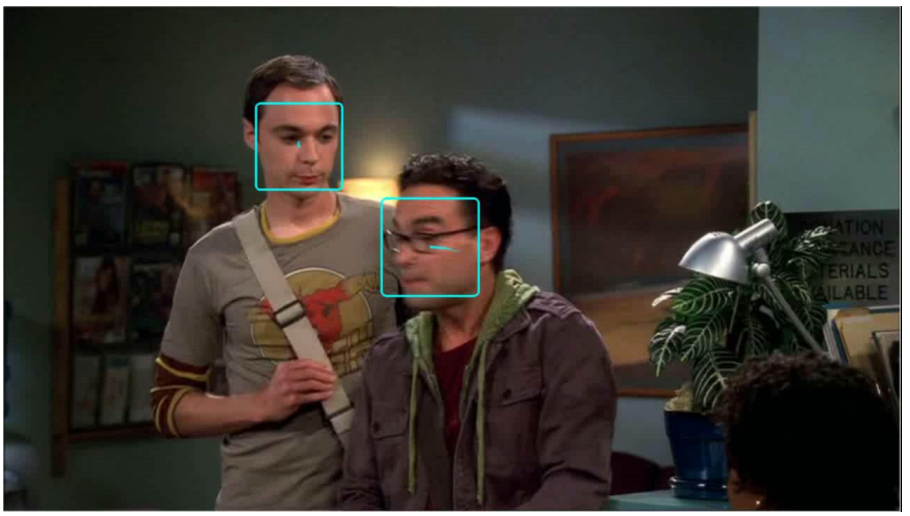
- 检测图片中物体的
  - 类别标签
  - 位置坐标(矩形框)



问题1：人类看一眼就能在图片里找到物体目标，似乎毫不费力。但如果把这个问题交给机器，它该怎么做？



人脸检测



问题2：有些人脸没有检测到，难点在哪里？

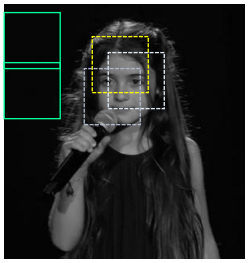
人眼检测vs.机器检测

5

7.2经典目标检测算法

- 基于“滑动窗口”方法
- 实例：HOG+SVM的面部识别
- 基于“区域提名”方法
- 区域卷积神经网络(R-CNN)
- Fast R-CNN
- Mask R-CNN

经典的滑动窗口检测法

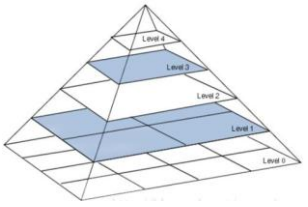


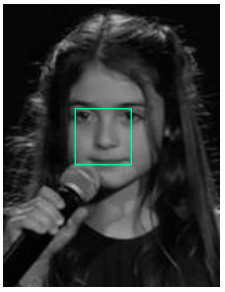
问题3:

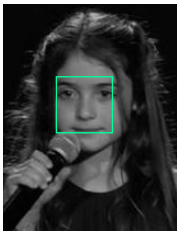
1. 窗口大小如何确定?
2. 窗口图像如何表达?
3. 多类别目标如何处理?
4. 很多重叠相似框如何筛选?

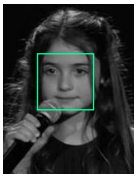
窗口大小如何确定?

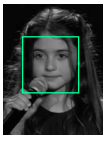
- 图像高斯金字塔
  - 多尺度







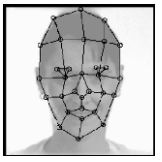


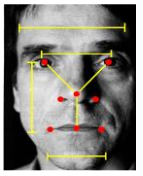


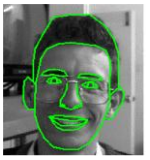
窗口图像如何表达

图像块（窗口）特征提取

- 颜色特征
- 边缘、边界特征
- 兴趣点特征
- 纹理特征
- 形状特征



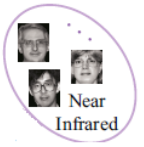




多类别目标如何处理?

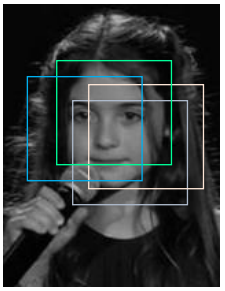
图像块（窗口）分类

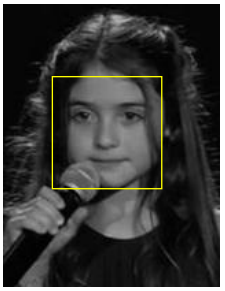
- 基于距离的分类器
- 贝叶斯分类器
- MAP分类器
- 线性判据
- 神经网络
- 支持向量机
- 决策树



很多重叠相似框如何筛选

- 非极大值抑制
  - Non-Maximum Suppression
  - 搜素局部区域,抑制非极大值





•选取多个矩形框的交集

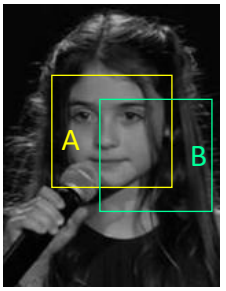
•选取多个矩形框的并集

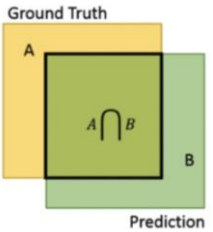
•选取置信度最高的一个

交并比

交并比(Intersection over Union, IoU);
物体检测需定位物体Bounding Box

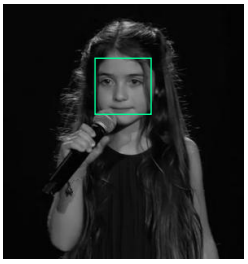
$$IoU = (A \cap B)/(A \cup B)$$

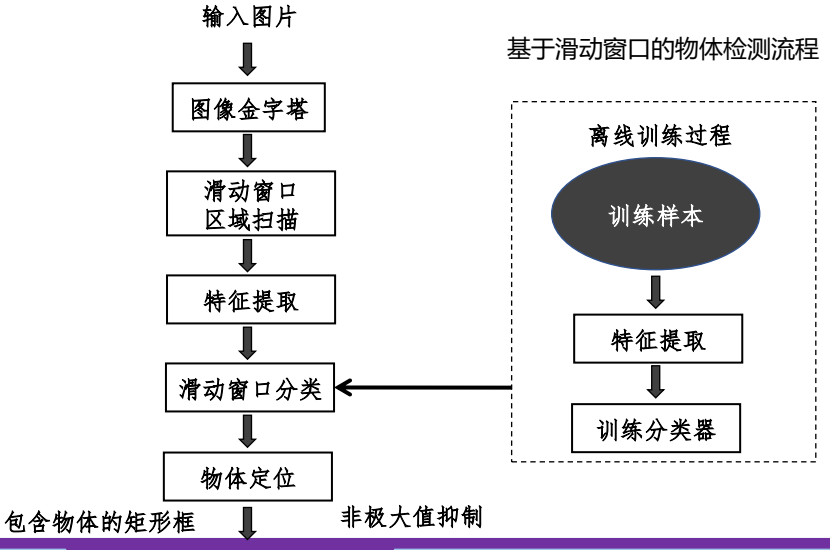




矩形框A、B的重叠面积
占A、B并集的面积比例

基于滑动窗口的物体检测方法

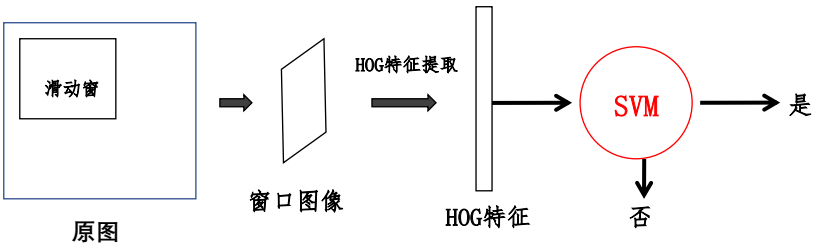




实例：“滑动窗口”面部识别

原理：

- 提取正负类训练集的HOG特征
- 训练SVM分类器
- 利用NMS移除重叠候选框。



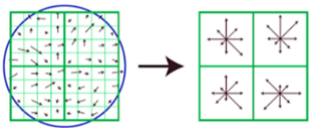
实例：“滑动窗口”面部识别

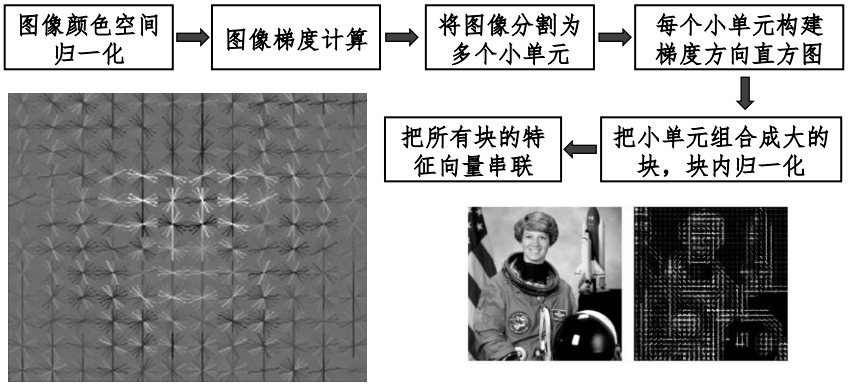
基于滑动窗口和二分类的物体检测算法

1. 选择阈值t和水平方向步长dx，垂直方向步长dy；
2. 建立图像金字塔；
3. For 金字塔中的每一层；
   对窗口使用分类器，得到分类概率c
   if c>t
   将该窗口放入候选列表L中
   end
   以步长dx和dy滑动窗口
   end
4. 将L中的候选框按分类概率c从大到小排序；
5. For 候选列表中的每一个候选窗口w；
   去除L中所有与w重叠超过一定阈值的候选窗口；
   end
6. L中剩余的窗口即为物体检测的结果。

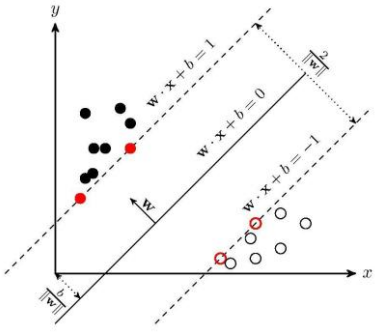
HOG特征

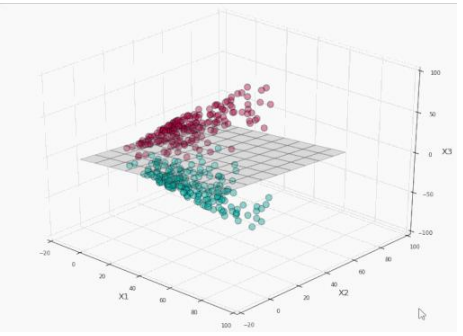
- Histogram of Oriented Gradients
- 提取正负类训练集的HOG特征





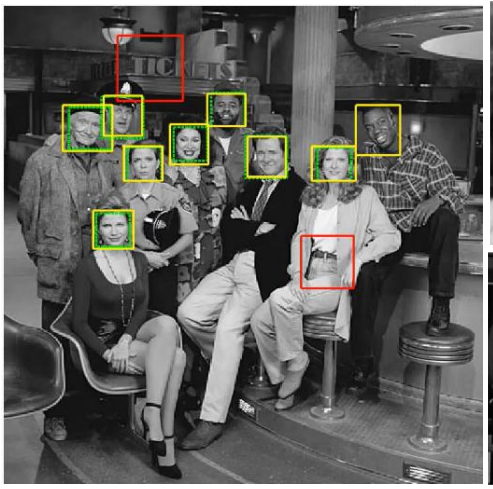
支持向量机(SVM)

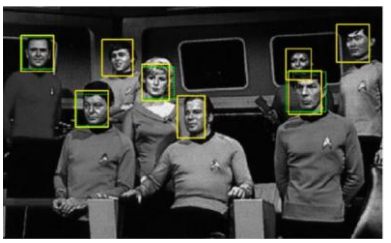




1) SVM适用小样本学习；　2) 计算的复杂性取决于支持向量的数目；
3)少数支持向量决定了最终结果；　4) 可以表示为凸优化问题

部分实验结果





“滑动窗口”的问题

计算量大

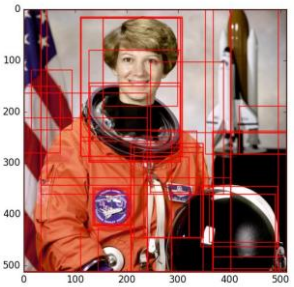
- 例如：482x348图像，所有可能的窗口数目约为70亿个。
- 需要分类的窗口数目过多，导致无法使用复杂的特征和分类器。

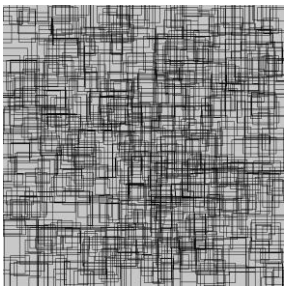
解决方法：

- 选择候选区域（缩小搜索空间）
- 使用复杂的特征和分类器

基于区域提名的物体检测方法

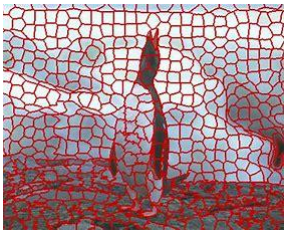
- 选择性搜索
  - 找出所有潜在可能包含目标的区域
  - 计算速度快、召回率高
  - 基于颜色、纹理、大小和形状等线索

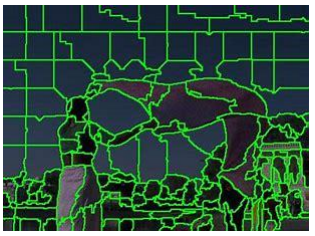




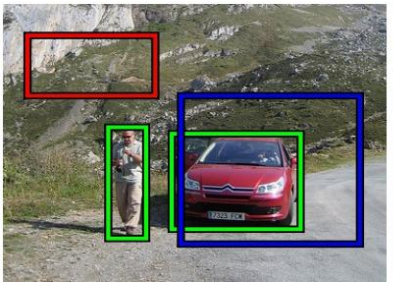
选择性搜索算法

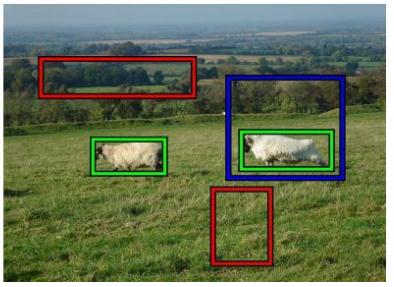
- 层次化的分组算法
  - 过分割，超像素
  - 相似度比较，合并区域
- 多样性的合并策略
  - 相似性度量（基于不同特征）

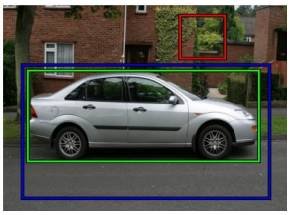




目标可能性（Objectness）



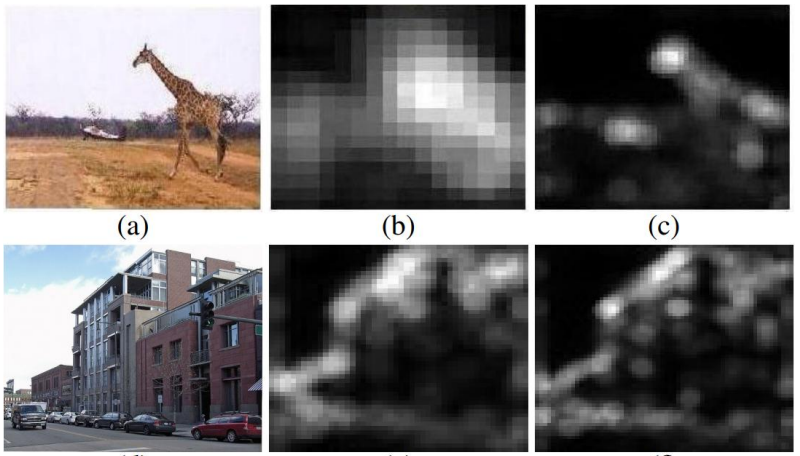




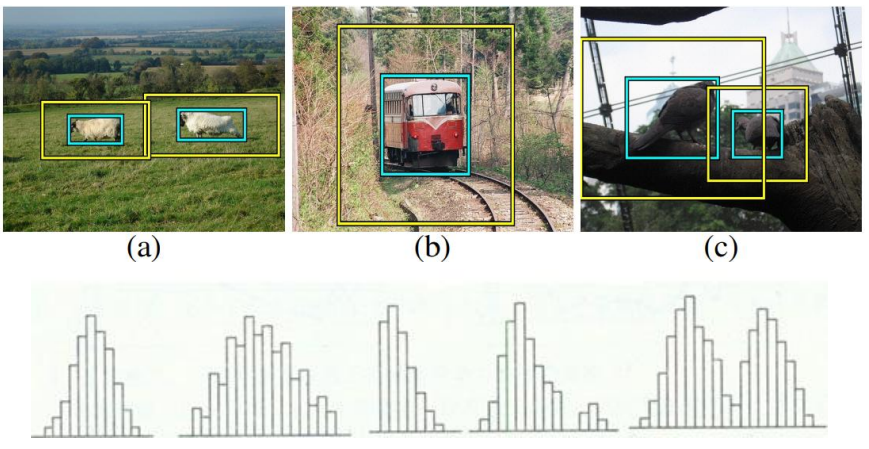
- 多尺度显著性
- 颜色对比度
- 边缘密度
- 超像素线索
- 边缘框

Alexe, et al. 2010

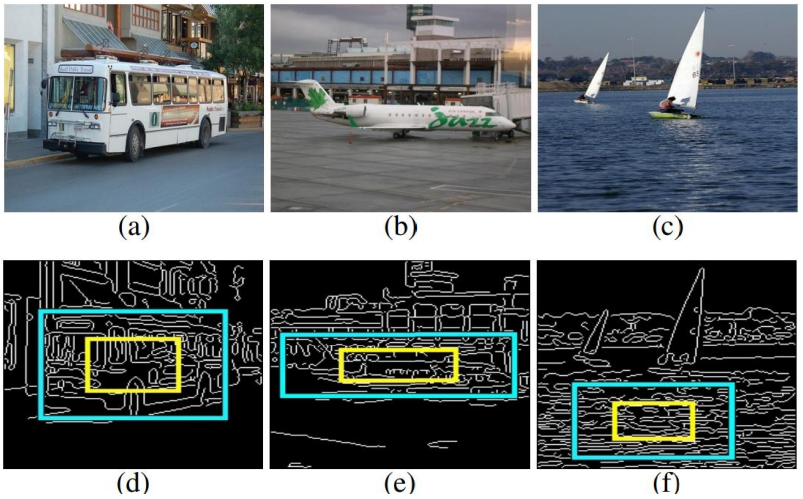
多尺度显著性



颜色对比度

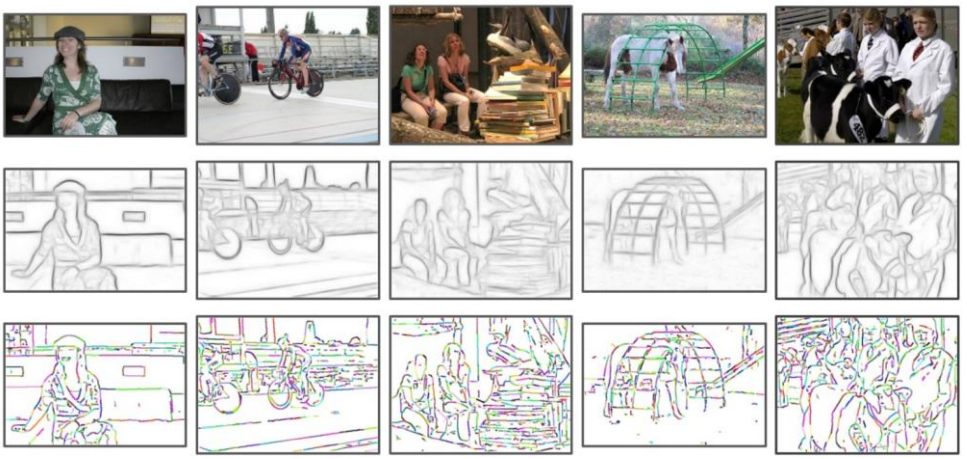


边缘密度



边缘密度

边缘框(EdgeBox)

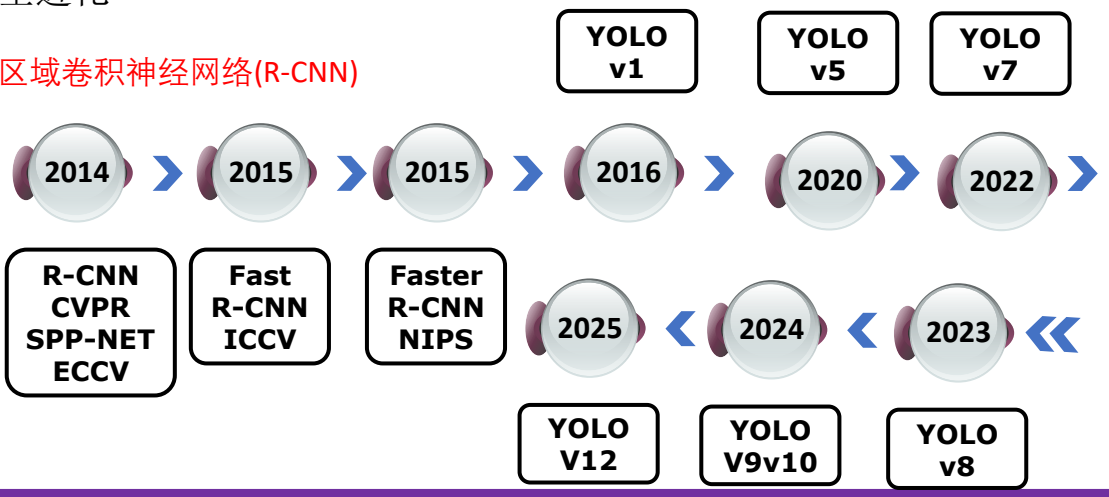


Zitnick, et al. 2014

7.3目标检测的发展

- 模型进化

区域卷积神经网络(R-CNN)

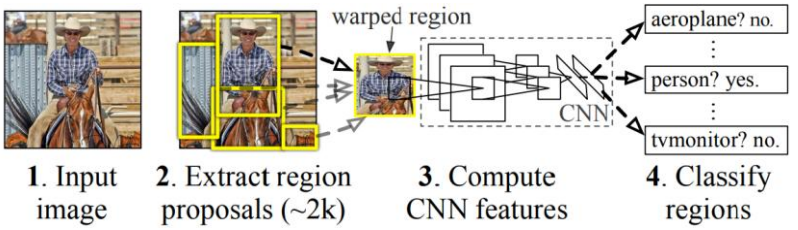


区域卷积神经网络(R-CNN)

- 传统方法→R-CNN

区域提名 + 手工特征 + 分类器

选择性搜索 (Selective Search) + CNN特征 + 支持向量机(SVM) + 边界框回归 (Bounding box regression)



选择性搜索

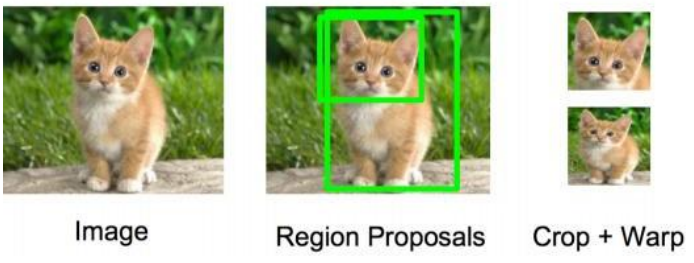
- 采样过分割，将图像分割成小区域（1K~2K个）
- 按照合并规则合并可能性最高的相邻两个区域。

合并策略：

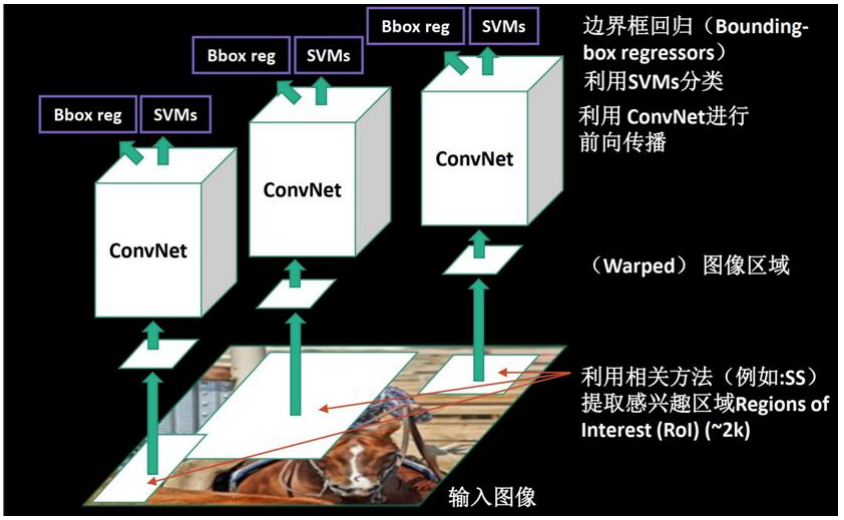
1. 颜色（颜色直方图）相近的
2. 纹理（梯度直方图）相近的
3. 合并后总面积小的
4. 合并后，总面积在其BBOX中所占比例大的

变形(Warp)

- 候选框都需要缩放到固定的大小（CNN）
  - 在原始区域目标周围取一块区域进行等比缩放
  - 对目标区域进行填充，再等比缩放
  - 直接将原始目标区域非等比缩放

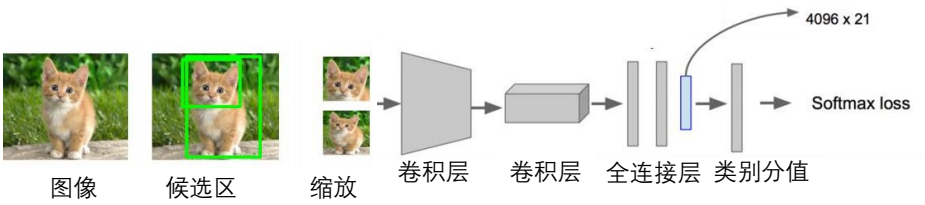


R-CNN架构



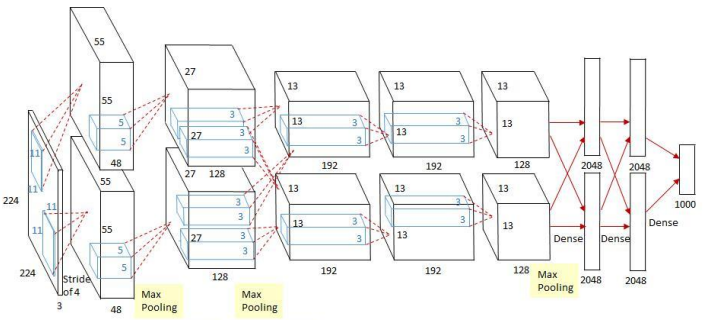
微调(Fine-tuning)

- 在ImageNet上对CNN模型进行预训练；(1000类)
- 在所有候选区域上对CNN进行微调；(PASCAL VOC:20类)
  - N类→N+1类（ 采用参数随机初始化 ）
  - N类(正样本)： 跟Ground-truth重合IOU>=0.5
  - 1类(负样本)： 背景类别 IOU<0.5



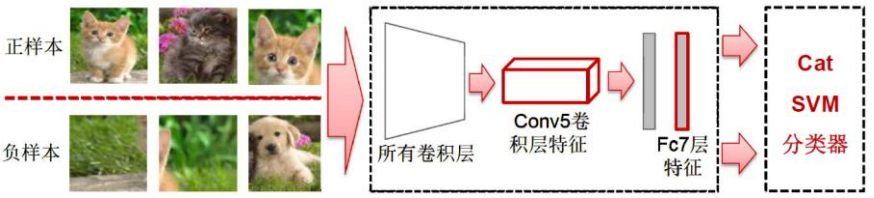
微调(Fine-tuning)

- 样本数据量相对较小，运用深度卷积网络容易出现严重的过拟合现象；
- 用Alexnet,VGG等网络在ImageNet等上预训练模型，然后对网络最后面的几层进行重新训练。
- 大数据集上的基础特征的提取，对于小数据同样适用，降低计算量



R-CNN:分类部分

- 在全连接最后一层特征上训练线性SVMs分类器；
- 每个类别(N类)对应一个SVM分类器；
- 正样本：所有Ground-truth区域；
- 负样本：跟重合IOU<0.3的候选区域



R-CNN:回归部分

- 在全连接最后层特征上训练Bounding box回归模型；
- 提升定位性能:每个类别（N类）训练一个回归模型
- 将候选区的BBox做重新映射P→G
- 训练输入 $\{(P^i, G^i)\}_{i=1,\ldots,N}$ $P^i = (P^i_x, P^i_y, P^i_w, P^i_h)$
- P的IOU>0.6 中心(x,y),宽高(w,h) $G = (G_x, G_y, G_w, G_h)$
- Squared Loss:

$$\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\operatorname{argmin}} \sum_i^N (t^i_\star - \hat{\mathbf{w}}_\star^{\mathrm{T}} \boldsymbol{\phi}_5(P^i))^2 + \lambda \|\hat{\mathbf{w}}_\star\|^2 .$$

CNN的Conv5特征

$$t_x = (G_x - P_x)/P_w \quad t_w = \log(G_w/P_w)$$
$$t_y = (G_y - P_y)/P_h \quad t_h = \log(G_h/P_h).$$

R-CNN:测试

- 选择性搜索提取2000候选区域；
- 将所有区域缩放到227x227；
- 使用fine-tune过的AlexNet计算2套特征
- 为每个类别执行：
- Fc7特征→SVM分类器→类别分值；
- 使用非极大值抑制（IoU>=0.5）获取无冗余的区域子集；
- Conv5特征→BBox 回归模型→BBox偏差
- 使用Bbox偏差修正区域子集

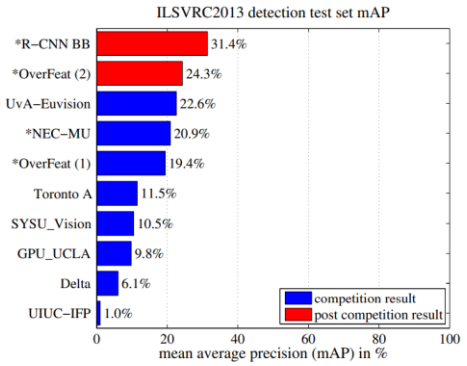
R-CNN:性能

- mAP:数据集中所有类的平均精度的平均值。
- mAP大幅提升；

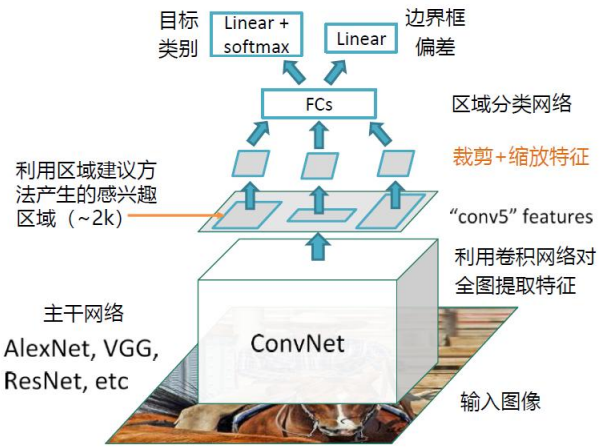
问题：

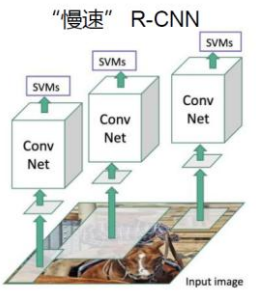
- 训练时间很长（84小时）
- 测试阶段很慢；
- 复杂的多阶段训练

	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image (with proposals)	50 seconds	2 seconds	**0.2 seconds**
(Speedup)	1x	25x	**250x**
mAP (VOC 2007)	66.0	**66.9**	**66.9**



Fast R-CNN



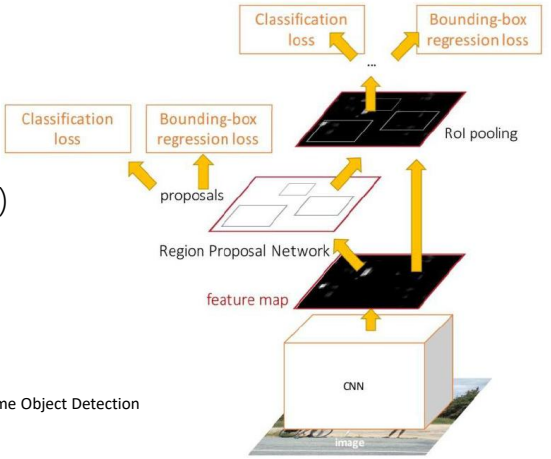


[1] Girshick, "Fast R CNN", ICCV 2015.

Faster R-CNN

利用卷积网络产生候选区域!

四种首损失联合训练:
RPN分类损失（目标/非目标）
RPN边界框坐标回归损失
候选区域分类损失
最终边界框坐标回归损失



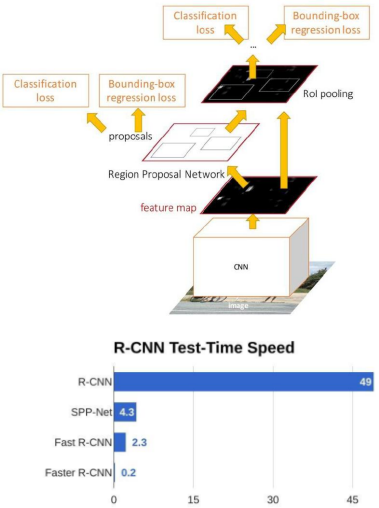
[1]Ren et al, “Faster R CNN: Towards Real Time Object Detection with Region Proposal Networks”, NIPS 2015.

Faster R-CNN

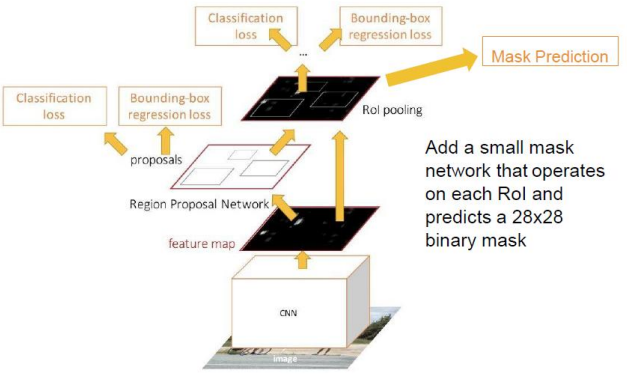
Faster R-CNN是一个两阶段目标检测器

第一阶段: 每张图运行一次
- 主干网络(Backbone)
- 区域建议网络(RPN)

第二阶段: 每个区域运行一次
- 扣取区域特征: RoI pool / align
- 预测目标类别
- 预测边界框偏移量

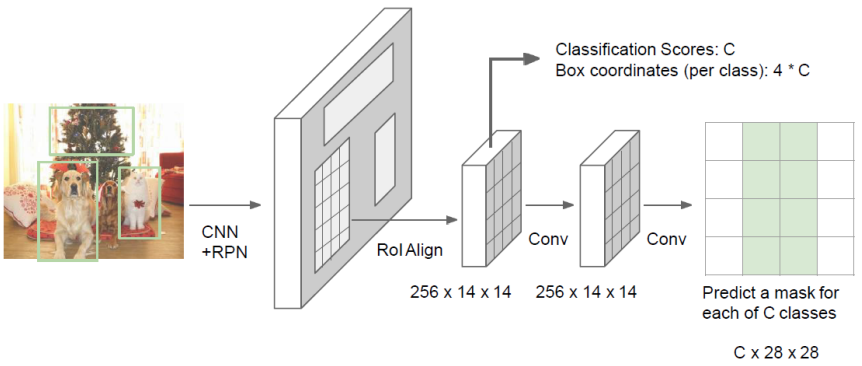


实例分割：Mask R-CNN



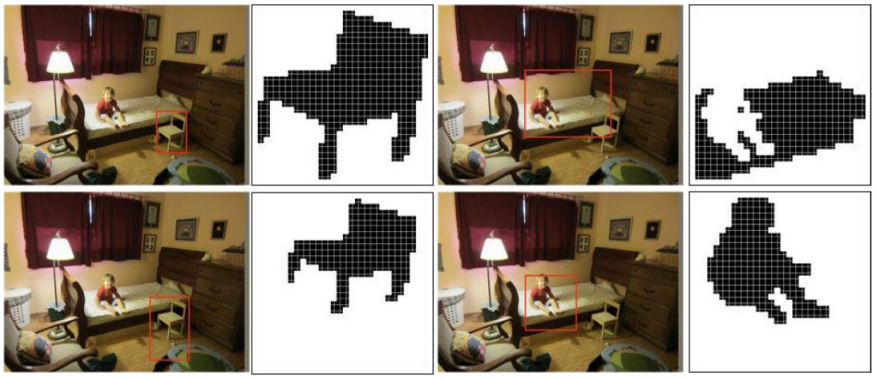
He et al, “Mask R-CNN”, ICCV, 2017.

MASK R-CNN



MASK R-CNN

- **Mask R-CNN训练阶段使用的Mask样例**

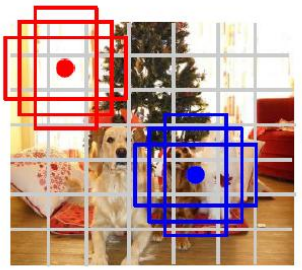


MASK R-CNN

- **Mask R-CNN实例分割结果**

一阶段目标检测 YOLO / SSD /RetinaNet

Input image
3 x H x W



Divide image into grid
7 x 7

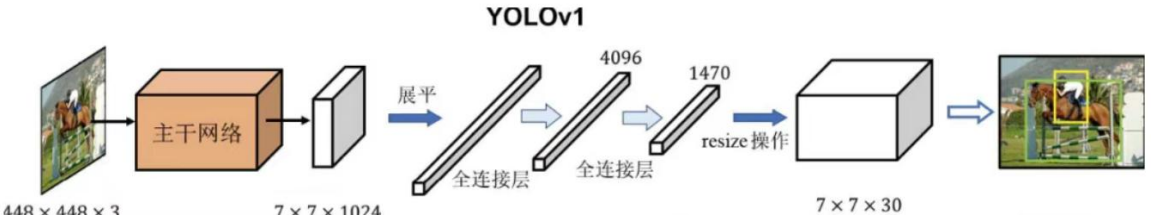
Image a set of **base boxes** centered at each grid cell
Here B = 3

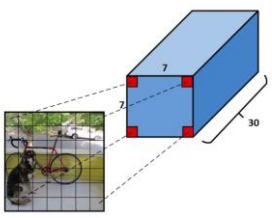
Within each grid cell:
- Regress from each of the B base boxes to a final box with 5 numbers: (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)
- Looks a lot like RPN, but category-specific!

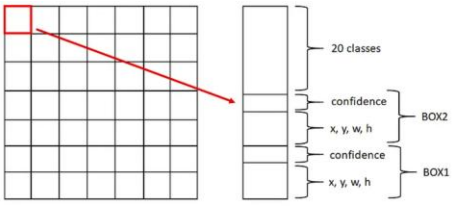
Output:
7 x 7 x (5 * B + C)

Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016
Lin et al, "Focal Loss for Dense Object Detection", ICCV 2017

YOLO v1 (2016)

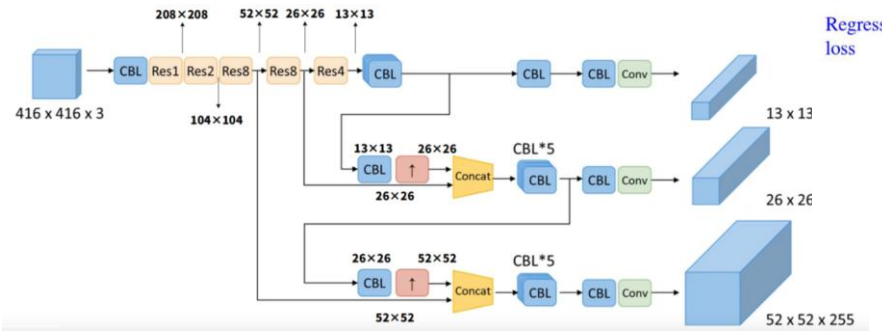


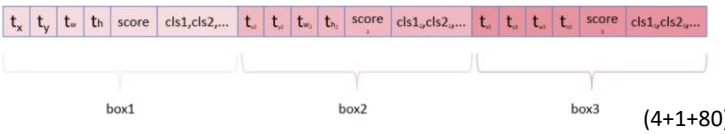




- **7×7网格划分过粗，小目标检测能力薄弱；**
- **无锚框设计依赖原始回归，定位精度偏低；**
- **每个网格仅预测2个边界框，难以应对密集目标场景**

YOLO v3 (2018)





(4+1+80)×3=255

Regression loss

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2\right]+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right]$$

Confidence loss

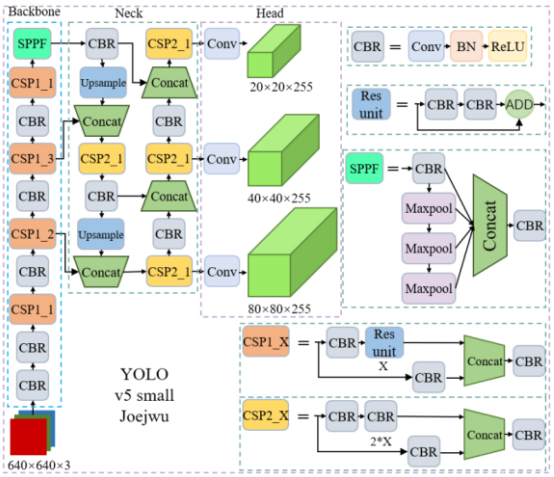
$$+\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{noobj}\left(C_i-\hat{C}_i\right)^2$$

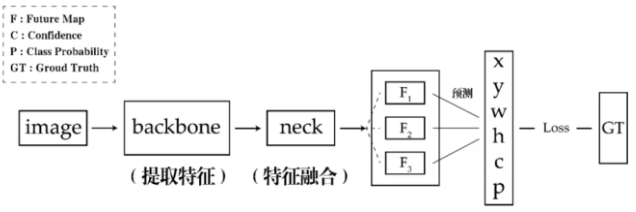
Classification loss

$$+\sum_{i=0}^{S^2}\mathbb{1}_{i}^{obj}\sum_{c\in classes}\left(p_i(c)-\hat{p}_i(c)\right)^2$$

- 围绕“特征金字塔+残差网络+类别预测优化”展开
- 彻底解决v1、v2的小目标漏检问题

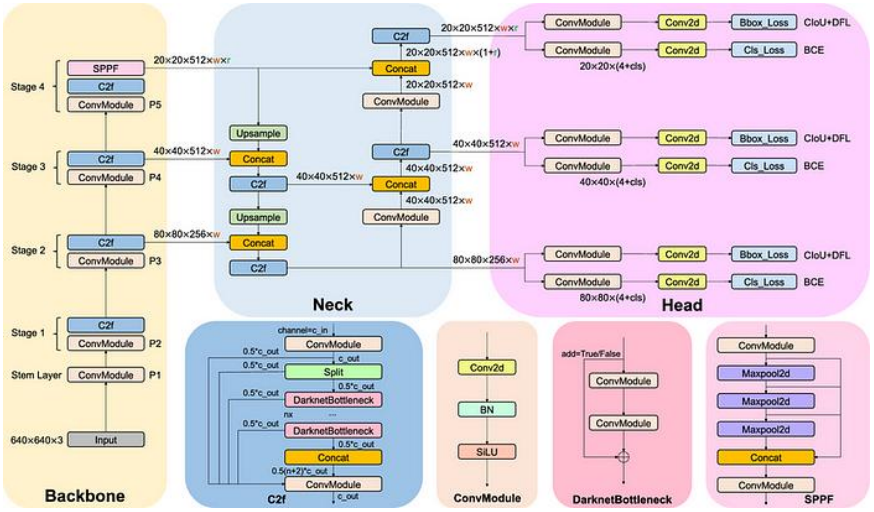
YOLO v5 (2020)



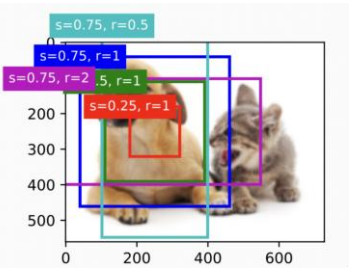


- YOLOv5提供Nano（n）、Small（s）、Medium（m）、Large（l）、Xtra-Large（x）五种型号；
- 参数规模从1.9M到89M；
- 算力需求覆盖手机端（CPU）、边缘设备（Jetson Nano）、云端GPU.

YOLO v8 (2023)

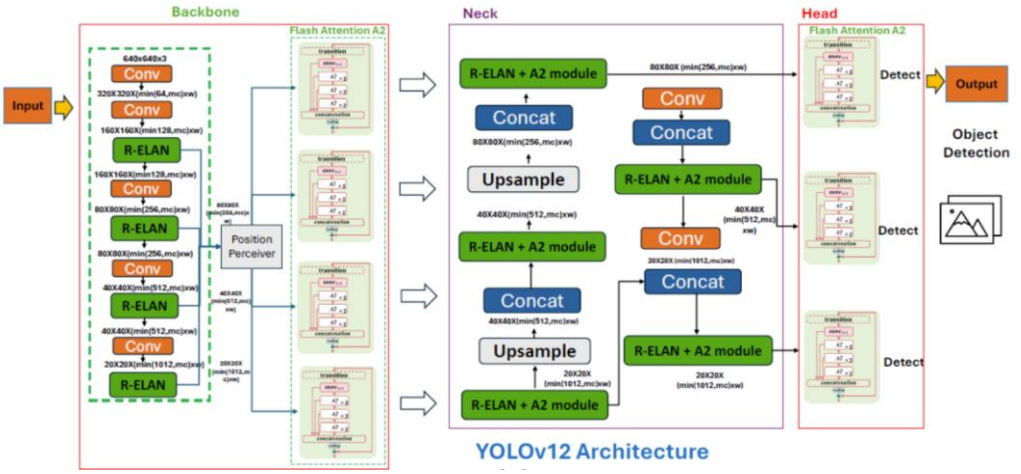


- 彻底摒弃锚框，采用“中心坐标+宽高直接回归”。
- 主干网络升级为改进型CSPDarknet8，优化残差连接与通道配比。



50

YOLO v12 (2025)



- **构建注意力中心框架**突破传统 YOLO 依赖 CNN 架构的局限

51

如何设计网络结构?

- **通用三段式骨架（所有检测器的底层逻辑）**
- **Backbone（骨干网络）**：负责特征提取。**提取高、中、低不同层次的语义特征。**
  常见部件：ResNet, DarkNet, EfficientNet, CSPNet。
- **Neck（颈部网络）**：负责特征融合。
  常见部件：FPN（特征金字塔）、PANet（双向融合）、BiFPN（加权融合）、NAS-FPN（搜索出来的）。
- **Head（检测头）**：负责最终预测。输入融合后的特征，输出框、类别、置信度。
  类型：**Anchor-Based**：预设锚点框；**Anchor-Free**：直接预测关键点或中心点。

52

核心设计原则

- **感受野 vs. 分辨率**
  - 大物体需要大感受野,小物体需要高分辨率.
- **特征融合方式**
  - 简单加/拼接
  - 加权融合
  - 双向路径增强
- **计算效率瓶颈**
  - 深度可分离卷积
  - CSP（跨阶段局部）：把输入特征图分成两路，一路直接跳过去，一路做卷积，最后再合并。
  - 通道数裁剪

53

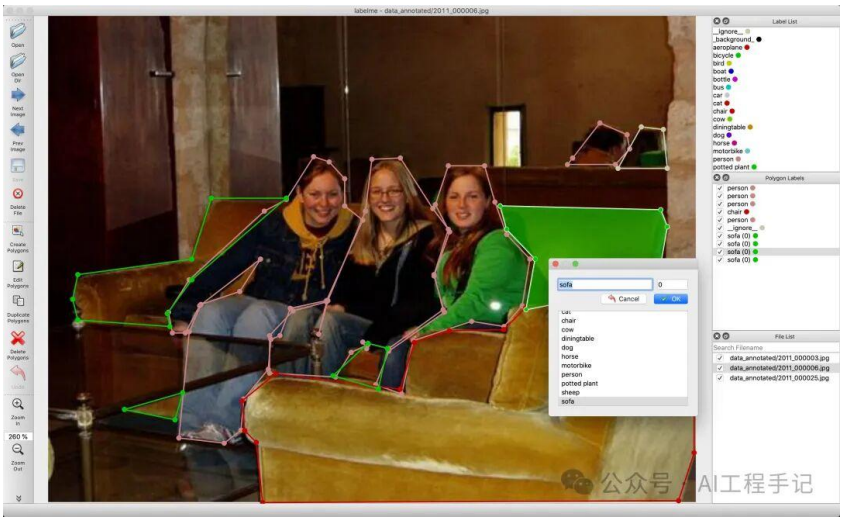
7.4常用数据集

- PASCAL VOC 07/12：VOC2007 有5k个训练图像，超过12k的标注目标；VOC2012有11k个训练图像，超过27k个标注目标
- ILSVRC：ImageNet Large Scale Visual Recognition Challenge包含1000个类别、超过100万个图像。
- MS-COCO：包含了超过200万个实例，且平均每张图像中有3.5个类别、7.7个实例，也包括了多种视角的图像。
- Open Image：谷歌提供的数据集，包含190万张图像上的600个类别，每张图像有8.3个对象类别。

Labelme

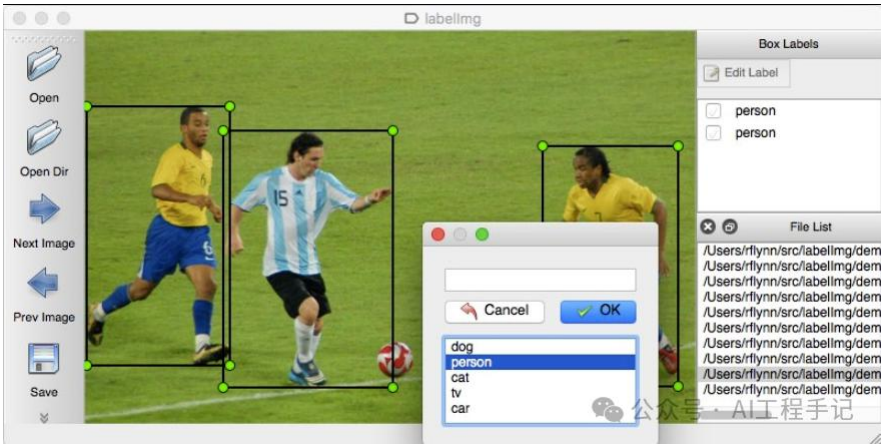
labelme 是一款开源的图像/视频标注工具，标签可用于目标检测、分割和分类。

支持图像的标注的组件有：矩形框，多边形，圆，线，点。



LabelImg

LabelImg是一个图形化的图像注释工具。它是用Python编写的，使用Qt作为其图形界面。注释被保存为PASCAL VOC格式的XML文件，该格式被ImageNet使用。此外，它还支持YOLO和Create ML格式。

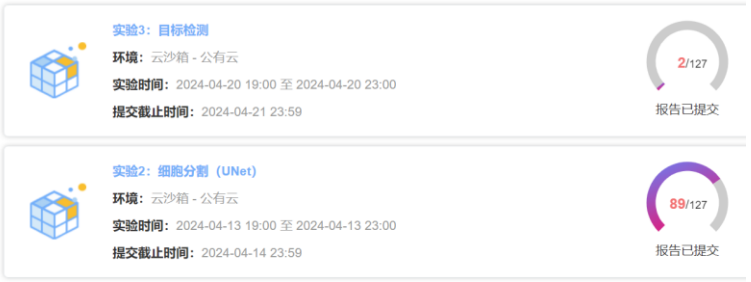


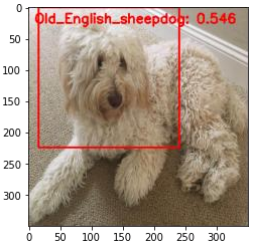
7.5 评价指标

- IoU交并比、精确度、召回率、准确率

		Prediction（预测）		
		Positive（正样本）	Negative（负样本）	T/F：表示预测的对错 P/N：表示预测的结果 目标检测中**正负样本**指的是模型自己预测出来的框与GT的IoU大于你设定的阈值即为正样本。
Actual（实际）	True	TP 识别对了(T)，识别结果为P，实际为P。	TN 识别对了(T)，识别结果为N，实际为N。	$F1Score = \frac{2 \times Precison \times Recall}{Precison + Recall}$
	False	FP 识别错了(F)，识别结果为P，实际为N。	FN 识别错了(F)，识别结果为N，实际为P	召回率：$Recall = \frac{TP}{TP + FN}$
		精确度：$Precision = \frac{TP}{TP + FP}$		准确率：$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$

案例





1. 实验目的

- 理解模型的保存与加载；
- 掌握传统目标检测算法流程；
- 掌握滑动窗口、图像金字塔、非极大值抑制等概念；
- 掌握 resnet50 分类模型的使用；

大作业

问题 1. 建立基于先验信息的手目标分割。

手先验信息：

测试图像：

问题2. 建立手目标协同分割模型。

测试图像组 1：

瑕疵检测任务

- 一维条码中可能存在的断码、白点、黑点等影响条码外观的瑕疵，检测这些瑕疵，并用红色矩形框将其标出。

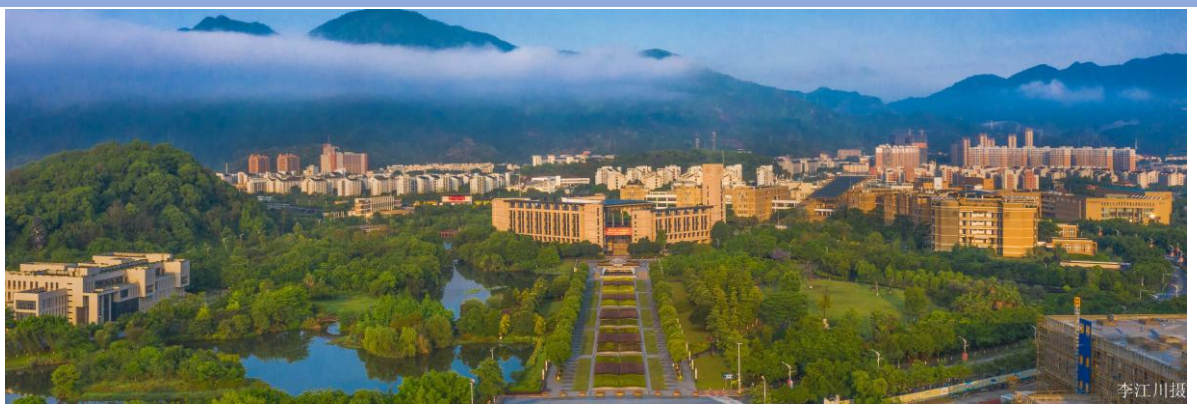
瑕疵检测任务

思路提示：

1. 将原图二值化后，提取连通体，根据联通体的特征（如长度、宽度等），筛选出条码线条；
2. 再根据条码线条的角度，对图像进行旋转矫正；
3. 然后对条码线条进行膨胀，提取出条码区域中所有的线条；
4. 进行单方向的膨胀，连接条码线条间断点；
5. 比较间断点连接前后的条码线条，从而确定瑕疵区域。

期末笔试考试时间

第12周	18	19	20	21	22	23	24
第13周	25	26	27	28	29	30	31
2026年6月份							
第14周	1	2	3	4	5	6	7
第15周	8	9	10	11	12	13	14



Thank you!