



孟繁亮

华为
开源软件能力中心
AI 领域首席架构师

想做团队的领跑者 需要迈过这些“槛”

成长型企业，易忽视人才体系化培养
企业转型加快，团队能力又跟不上

VS

从基础到进阶，超100+一线实战
技术专家带你系统化学习成长

团队成员技能水平不一，
难以一“敌”百人需求

VS

解决从小白到资深技术人所遇到
80%的问题

寻求外部培训，奈何价更高且
集中式学习

VS

多样、灵活的学习方式，包括
音频、图文 和视频

学习效果难以统计，产生不良循环

VS

获取员工学习报告，查看学习
进度，形成闭环



课程顾问「橘子」

回复「QCon」
免费获取
学习解决方案

极客时间企业账号 # 解决技术人成长路上的学习问题

孟繁亮



目录

1.Mask R-CNN介绍

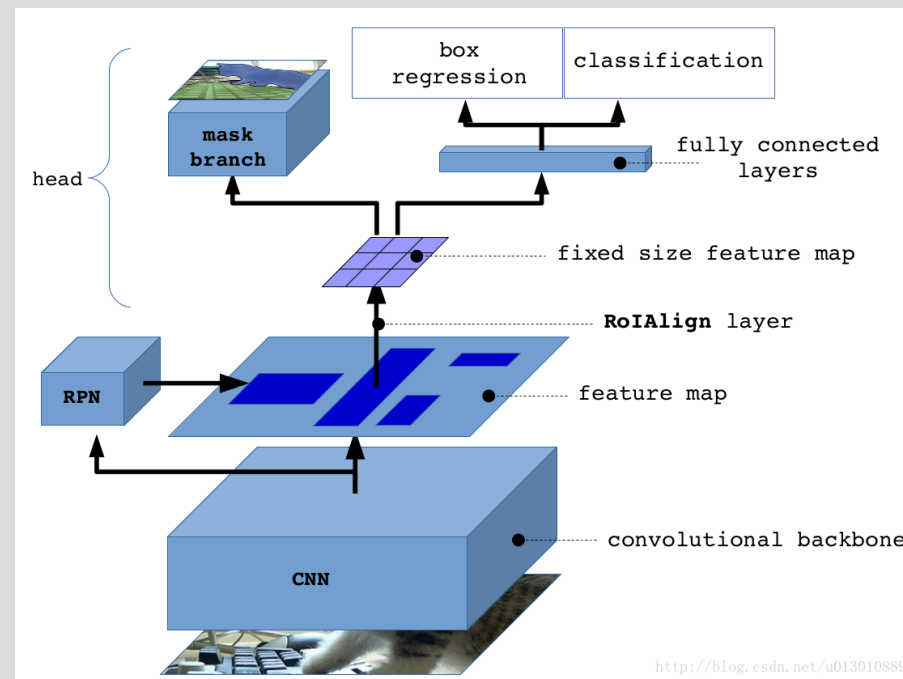
2.ResNet+FPN

3.RPN: Anchor

4.RPN

5.ROI & head

Mask R-CNN 能做什么？

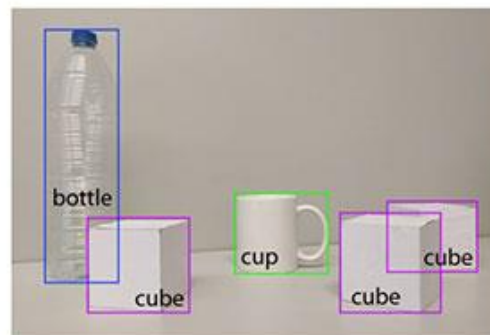


医学影像、车辆、道路、GIS、人体姿态。。。

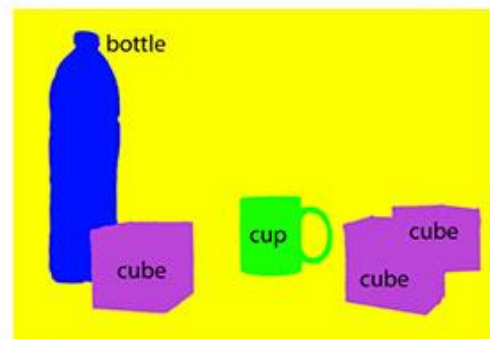
R-CNN 进化: ResNet+FPN+RPN(RoIAlign)+FCN



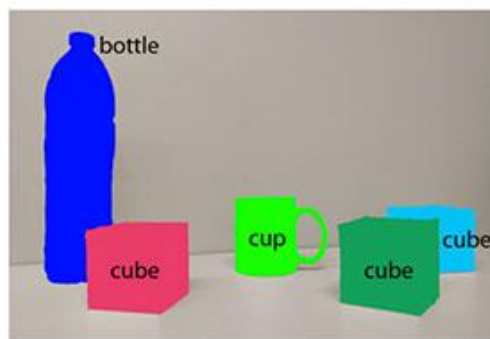
(a) Image classification



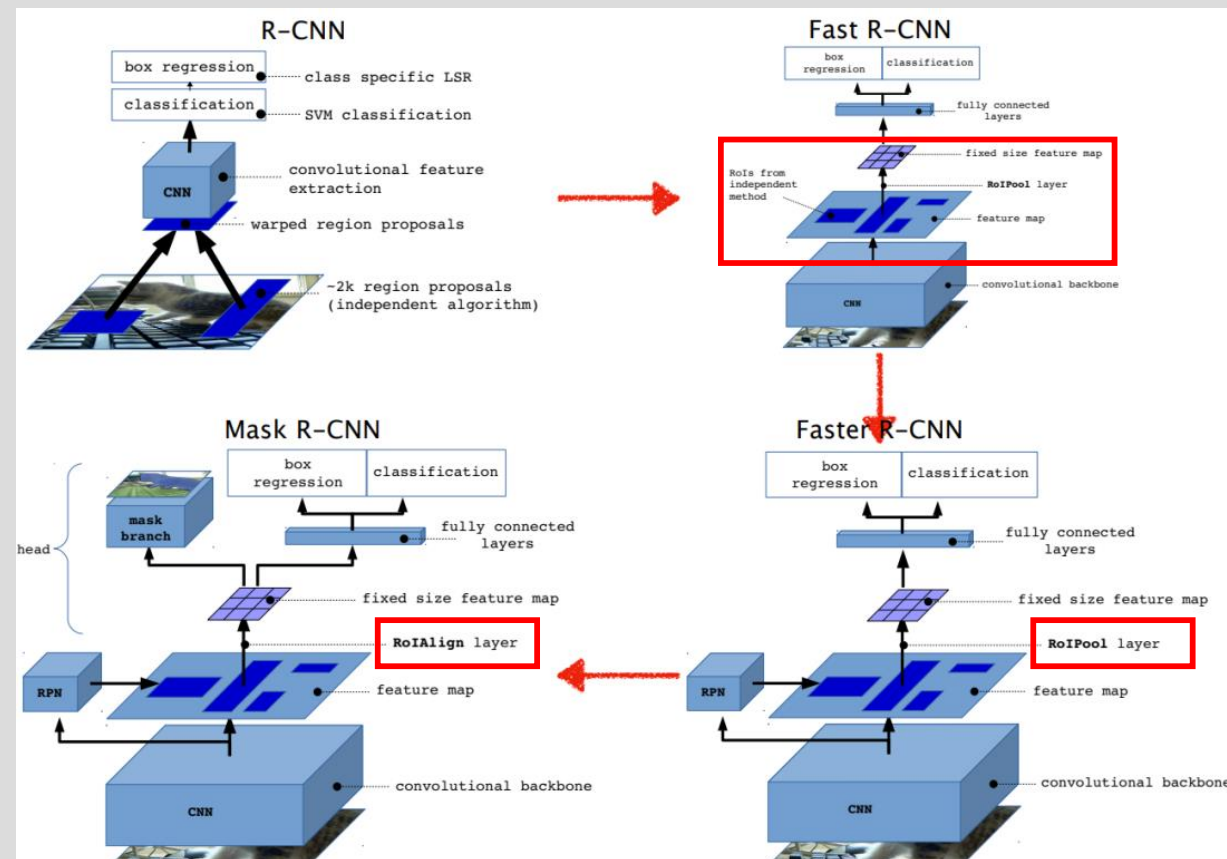
(b) Object localization



(c) Semantic segmentation

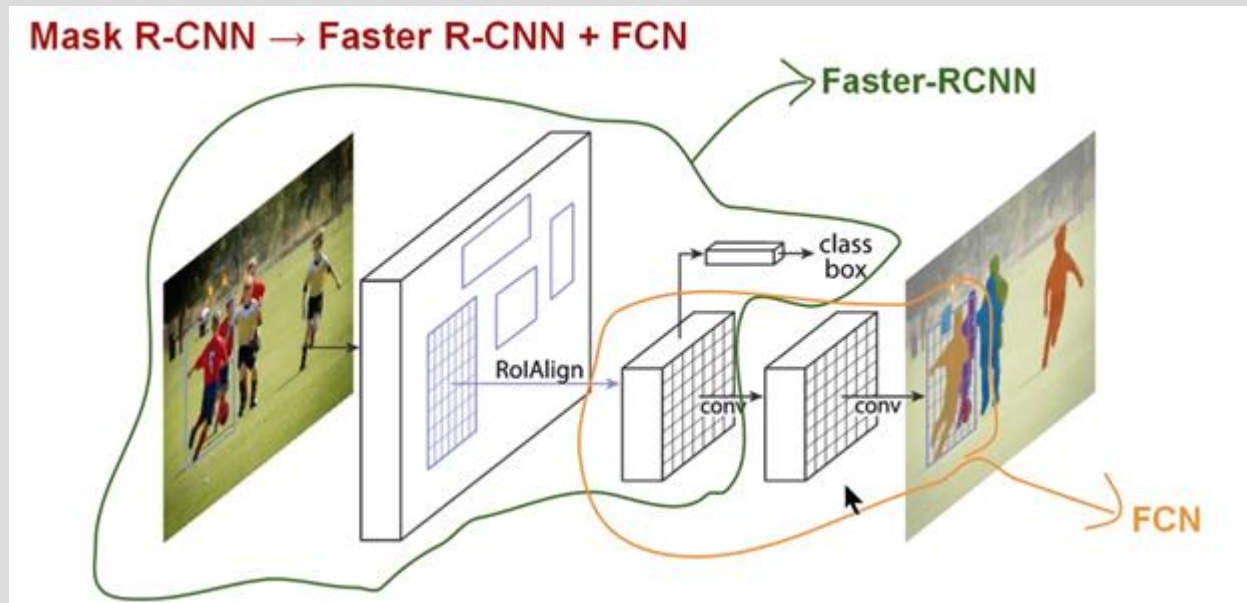


(d) Instance segmentation

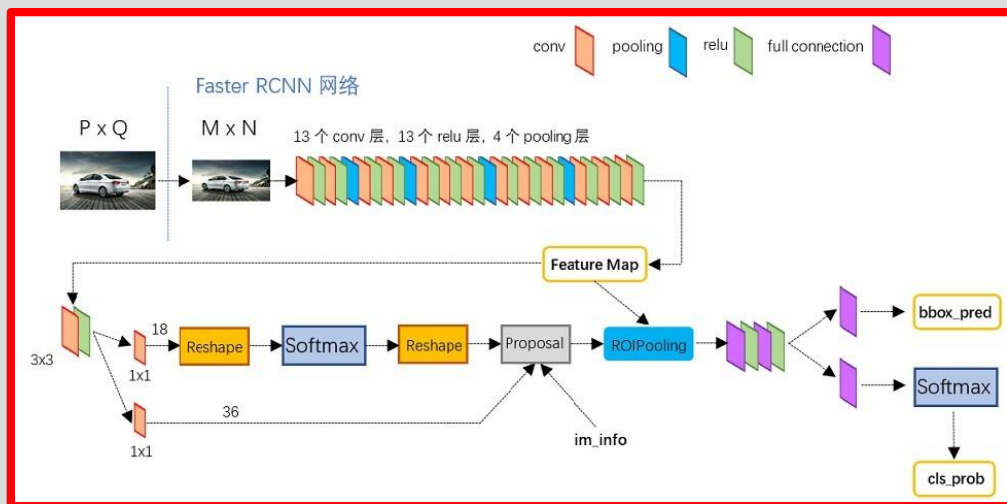
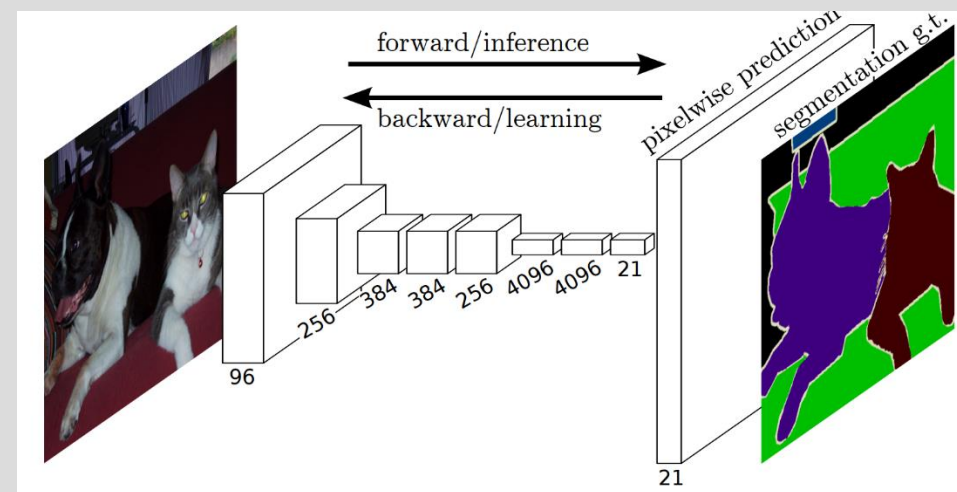


RCNN追求 更好的Feature map, 更精确的RoIAlign

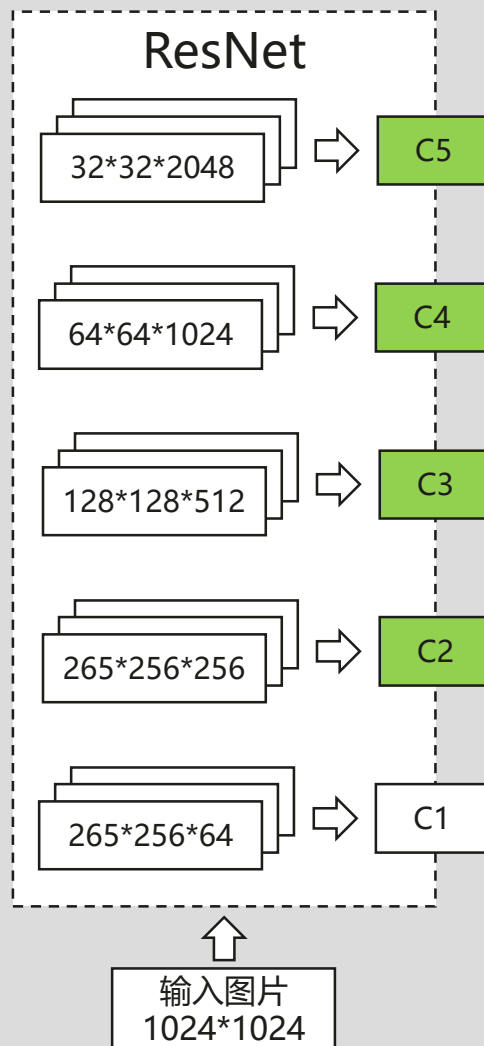
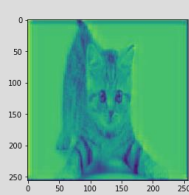
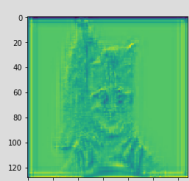
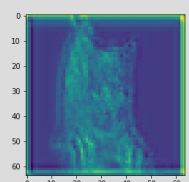
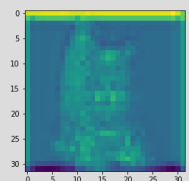
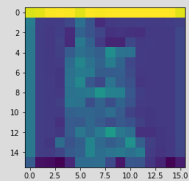
Mask R-CNN 对 Faster R-CNN+FCN 改进



- 1、backbone: ResNet+FPN
- 2、RoI Pooling 改进为RoIAlign
- 3、增加Mask 分支



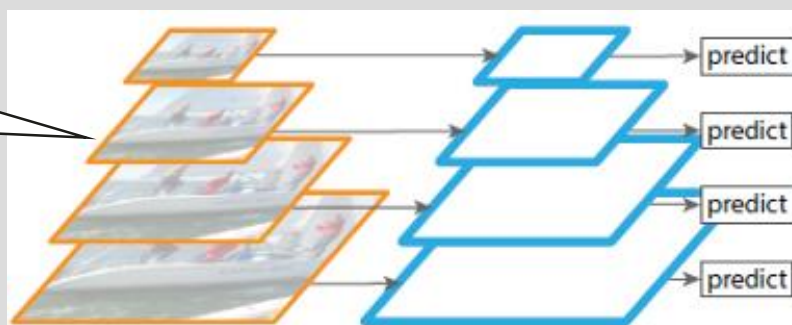
Mask R-CNN step 1 : ResNet



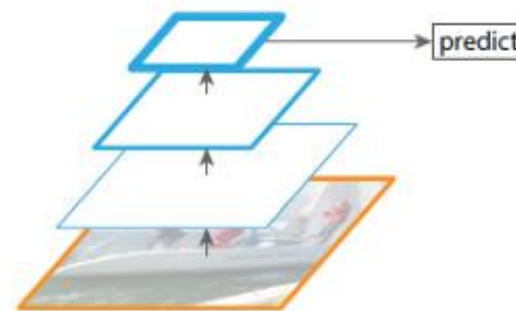
```
def resnet_graph(input_image, architecture, stage5=False, train_bn=True):
    """Build a ResNet graph.
    architecture: Can be resnet50 or resnet101
    stage5: Boolean. If False, stage5 of the network is not created
    train_bn: Boolean. Train or freeze Batch Norm layers
    """
    assert architecture in ["resnet50", "resnet101"]
    # Stage 1
    x = KL.ZeroPadding2D((3, 3))(input_image)
    x = KL.Conv2D(64, (7, 7), strides=(2, 2), name='conv1', use_bias=True)(x)
    x = BatchNorm(name='bn_conv1')(x, training=train_bn)
    x = KL.Activation('relu')(x)
    C1 = x = KL.MaxPooling2D((3, 3), strides=(2, 2), padding="same")(x)
    # Stage 2
    x = conv_block(x, 3, [64, 64, 256], stage=2, block='a', strides=(1, 1), train_bn=train_bn)
    x = identity_block(x, 3, [64, 64, 256], stage=2, block='b', train_bn=train_bn)
    C2 = x = identity_block(x, 3, [64, 64, 256], stage=2, block='c', train_bn=train_bn)
    # Stage 3
    x = conv_block(x, 3, [128, 128, 512], stage=3, block='a', train_bn=train_bn)
    x = identity_block(x, 3, [128, 128, 512], stage=3, block='b', train_bn=train_bn)
    x = identity_block(x, 3, [128, 128, 512], stage=3, block='c', train_bn=train_bn)
    C3 = x = identity_block(x, 3, [128, 128, 512], stage=3, block='d', train_bn=train_bn)
    # Stage 4
    x = conv_block(x, 3, [256, 256, 1024], stage=4, block='a', train_bn=train_bn)
    block_count = {"resnet50": 5, "resnet101": 22}[architecture]
    for i in range(block_count):
        x = identity_block(x, 3, [256, 256, 1024], stage=4, block=chr(98 + i), train_bn=train_bn)
    C4 = x
    # Stage 5
    if stage5:
        x = conv_block(x, 3, [512, 512, 2048], stage=5, block='a', train_bn=train_bn)
        x = identity_block(x, 3, [512, 512, 2048], stage=5, block='b', train_bn=train_bn)
        C5 = x = identity_block(x, 3, [512, 512, 2048], stage=5, block='c', train_bn=train_bn)
    else:
        C5 = None
    return [C1, C2, C3, C4, C5]
```


Mask R-CNN step 2 : FPN 的演化

计算量太大



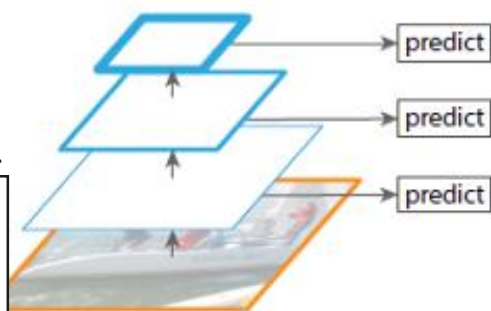
(a) Featurized image pyramid



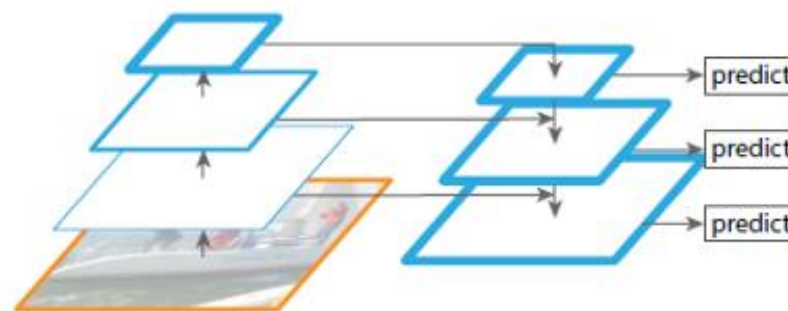
(b) Single feature map

无法识别小目标

错过重用特征层高分辨率信息(小目标的召回率还是有点低)



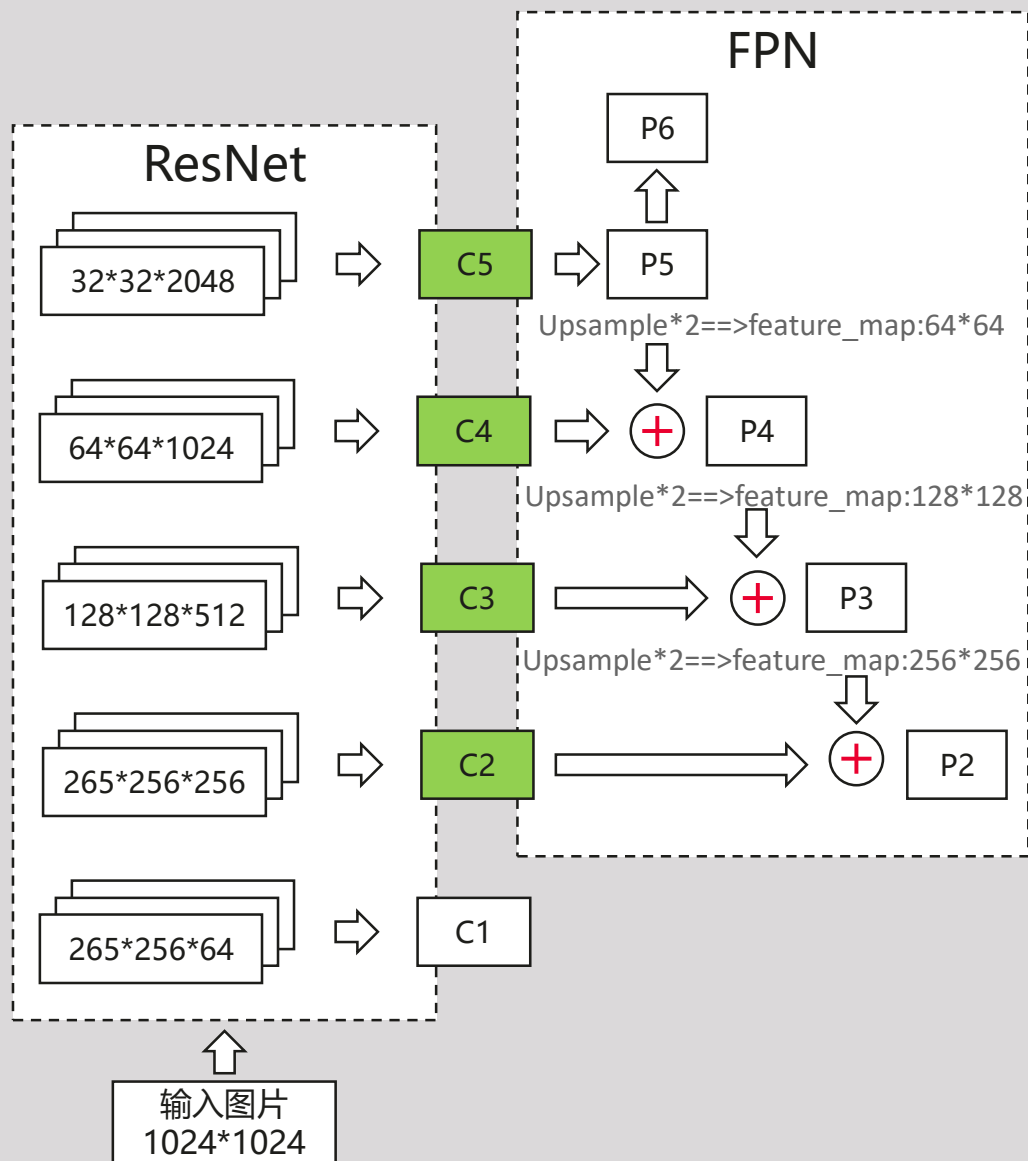
(c) Pyramidal feature hierarchy



(d) Feature Pyramid Network

Bottom-up pathway
Top-down pathway
lateral connections
重用特征层高分辨率信息

Mask R-CNN step 3 : ResNet → FPN



```
# Build the shared convolutional layers.
# Bottom-up Layers
# Returns a list of the last layers of each stage, 5 in total.
# Don't create the thread (stage 5), so we pick the 4th item in the list.
if callable(config.BACKBONE):
    _, C2, C3, C4, C5 = config.BACKBONE(input_image, stage5=True,
                                         train_bn=config.TRAIN_BN)
else:
    _, C2, C3, C4, C5 = resnet_graph(input_image, config.BACKBONE,
                                     stage5=True, train_bn=config.TRAIN_BN)

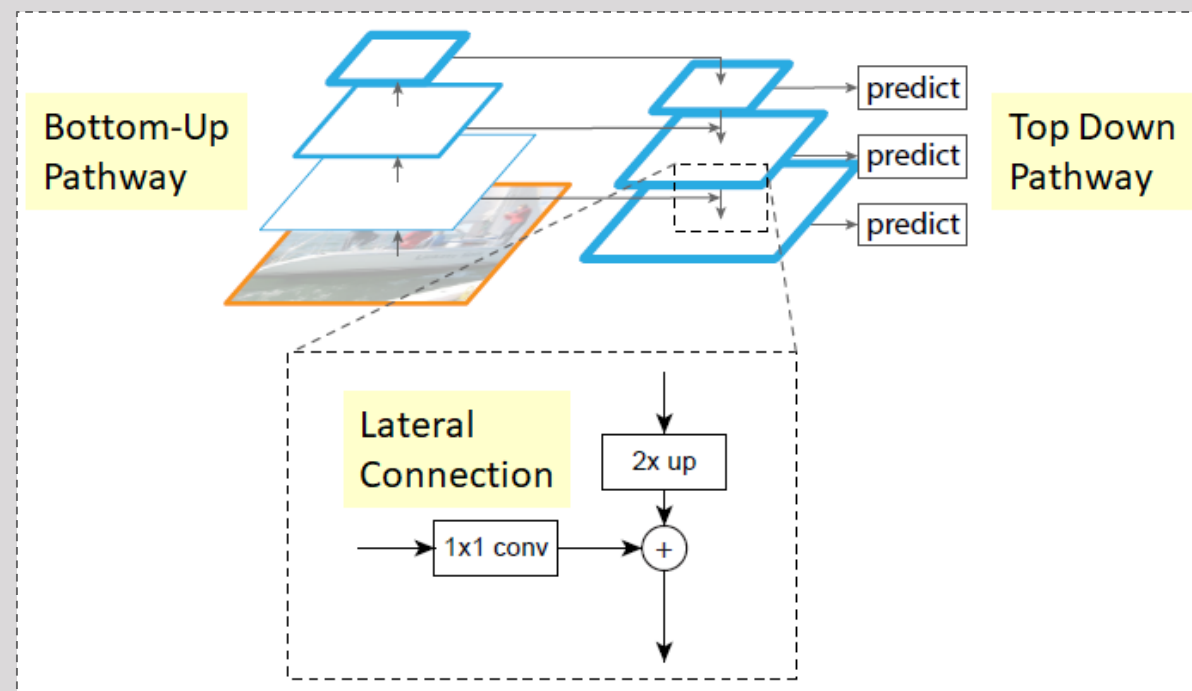
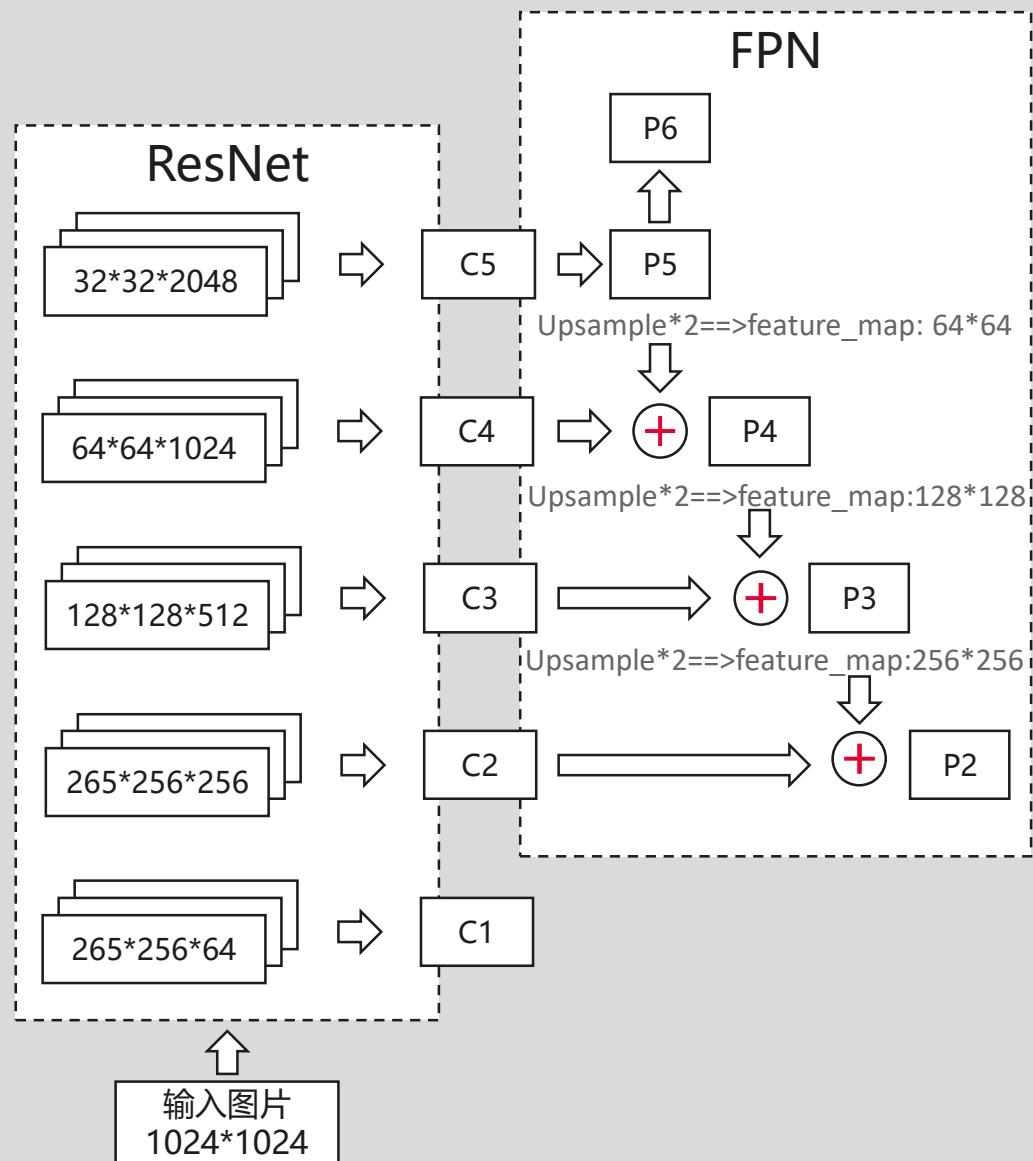
# Top-down Layers
# TODO: add assert to verify feature map sizes match what's in config
P5 = KL.Conv2D(config.TOP_DOWN_PYRAMID_SIZE, (1, 1), name='fpn_c5p5')(C5)
P4 = KL.Add(name="fpn_p4add")([
    KL.UpSampling2D(size=(2, 2), name="fpn_p5upsampled")(P5),
    KL.Conv2D(config.TOP_DOWN_PYRAMID_SIZE, (1, 1), name='fpn_c4p4')(C4)])
P3 = KL.Add(name="fpn_p3add")([
    KL.UpSampling2D(size=(2, 2), name="fpn_p4upsampled")(P4),
    KL.Conv2D(config.TOP_DOWN_PYRAMID_SIZE, (1, 1), name='fpn_c3p3')(C3)])
P2 = KL.Add(name="fpn_p2add")([
    KL.UpSampling2D(size=(2, 2), name="fpn_p3upsampled")(P3),
    KL.Conv2D(config.TOP_DOWN_PYRAMID_SIZE, (1, 1), name='fpn_c2p2')(C2)])

# Attach 3x3 conv to all P layers to get the final feature maps.
P2 = KL.Conv2D(config.TOP_DOWN_PYRAMID_SIZE, (3, 3), padding="SAME", name="fpn_p2")(P2)
P3 = KL.Conv2D(config.TOP_DOWN_PYRAMID_SIZE, (3, 3), padding="SAME", name="fpn_p3")(P3)
P4 = KL.Conv2D(config.TOP_DOWN_PYRAMID_SIZE, (3, 3), padding="SAME", name="fpn_p4")(P4)
P5 = KL.Conv2D(config.TOP_DOWN_PYRAMID_SIZE, (3, 3), padding="SAME", name="fpn_p5")(P5)

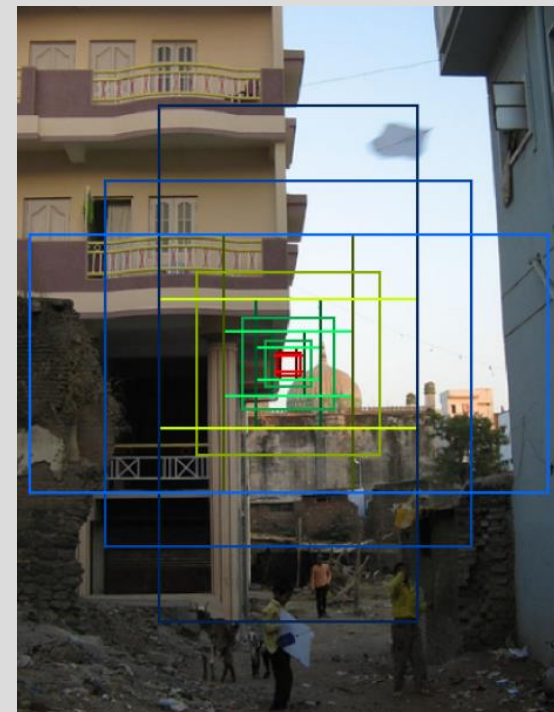
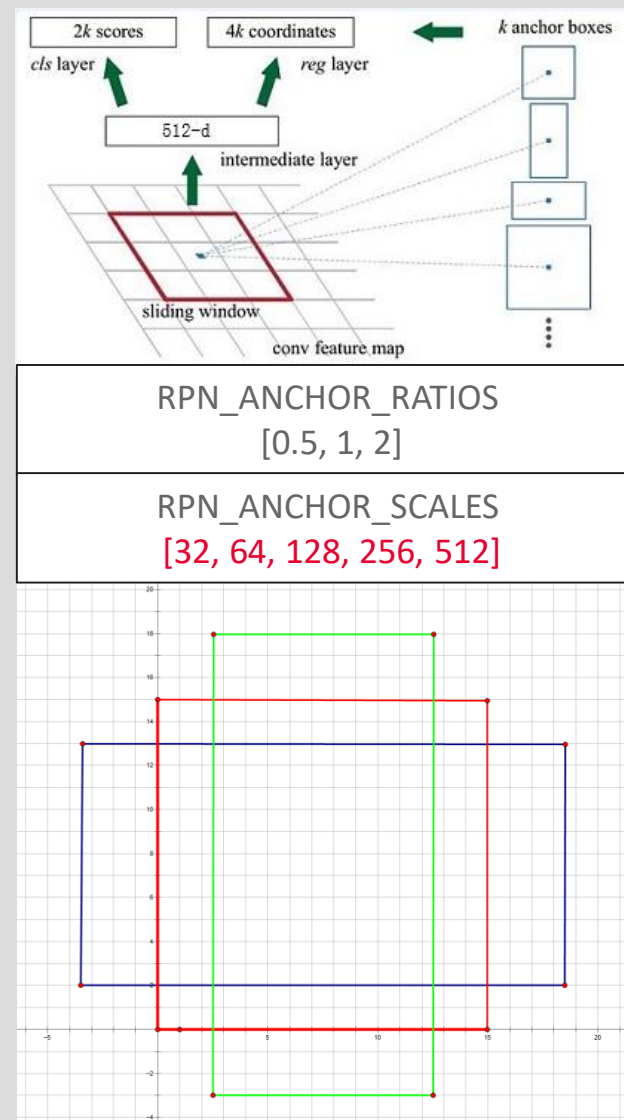
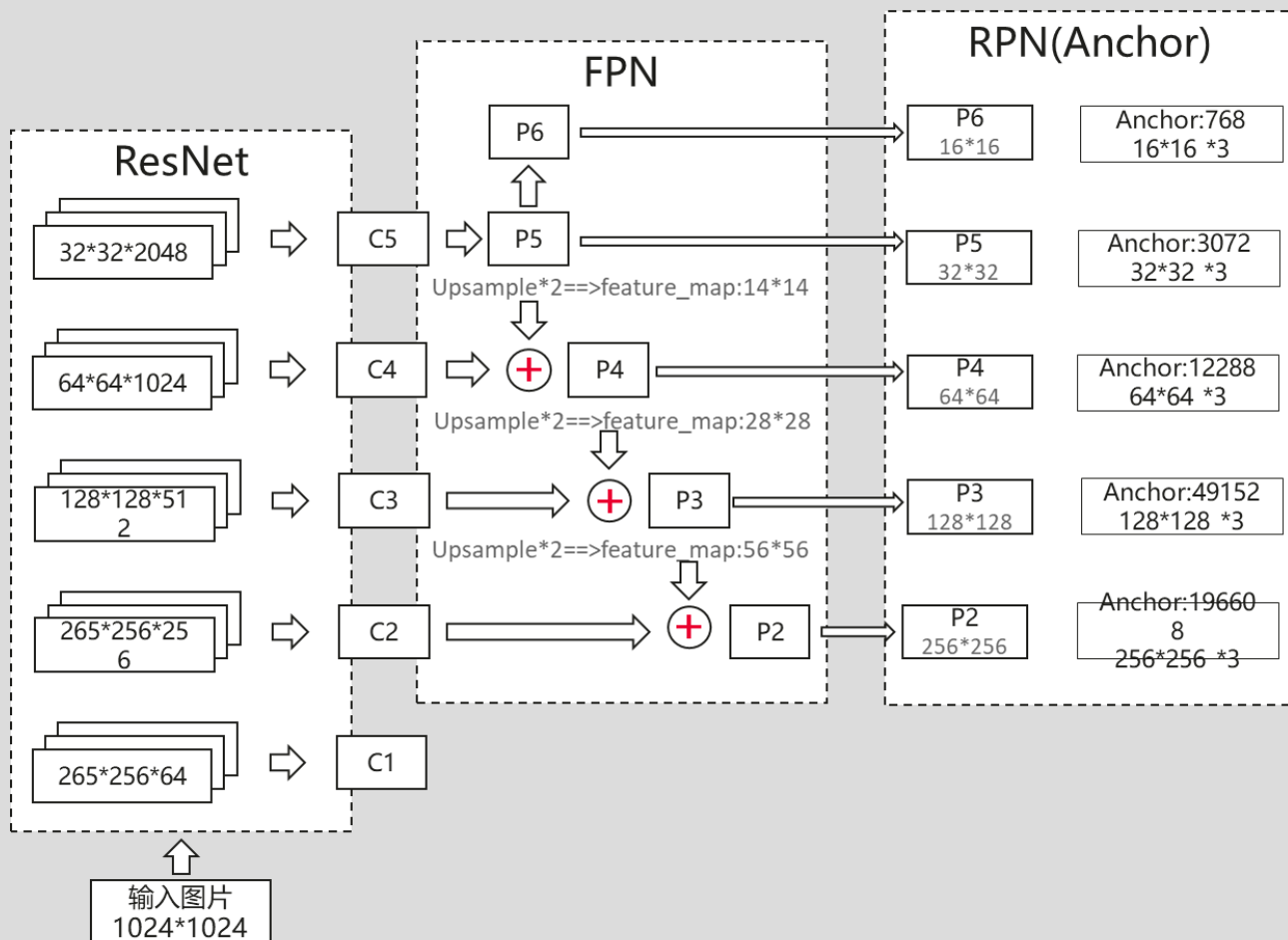
# P6 is used for the 5th anchor scale in RPN. Generated by
# subsampling from P5 with stride of 2.
P6 = KL.MaxPooling2D(pool_size=(1, 1), strides=2, name="fpn_p6")(P5)

# Note that P6 is used in RPN, but not in the classifier heads.
rpn_feature_maps = [P2, P3, P4, P5, P6]
mrcnn_feature_maps = [P2, P3, P4, P5]
```

Mask R-CNN step 3 : RestNet → FPN



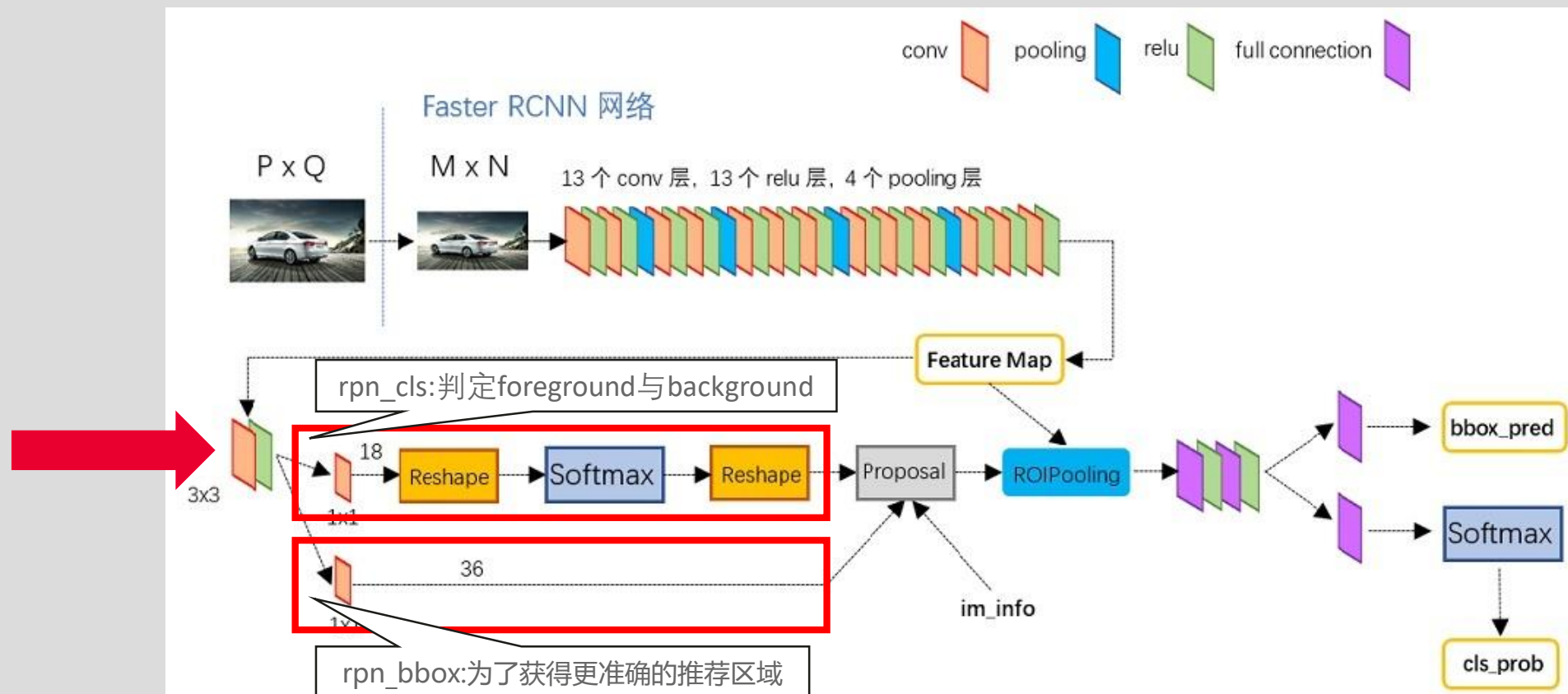
Mask R-CNN step 4 : RPN 的 Anchor



Mask R-CNN step 4 : RPN

RPN(Anchor)

P6 16*16	Anchor:768 16*16 *3
P5 32*32	Anchor:3072 32*32 *3
P4 64*64	Anchor:12288 64*64 *3
P3 128*128	Anchor:49152 128*128 *3
P2 256*256	Anchor:196608 256*256 *3



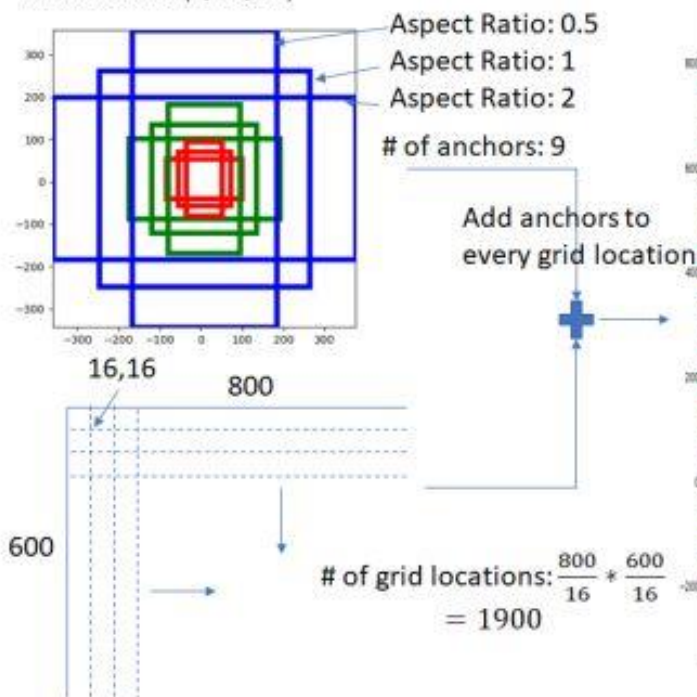
Faster RCNN vs Mask RCNN: RPN – Anchors

一个Cell 9个Anchors vs 15个Anchors

Generate Anchors

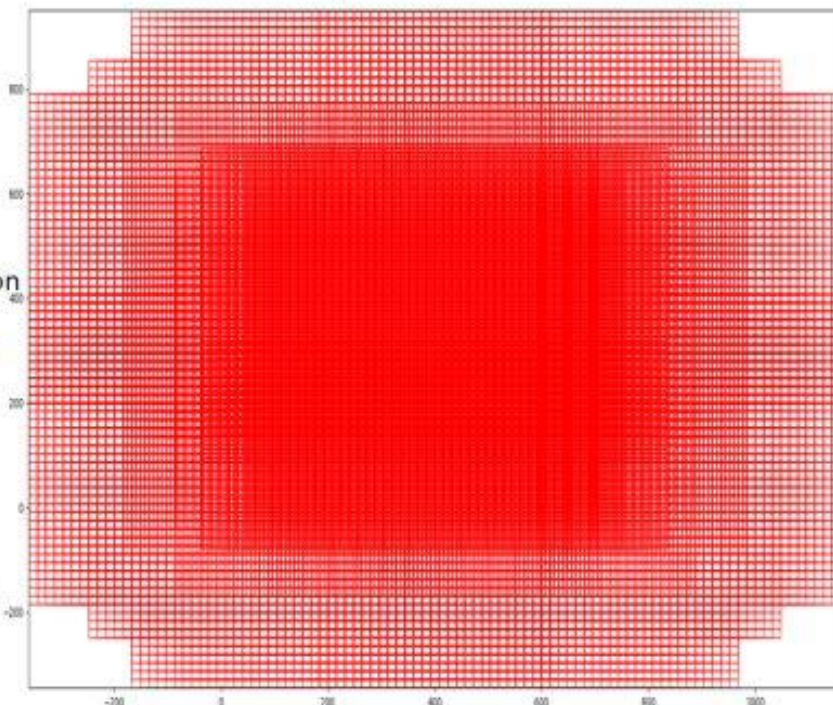
Given:

- Set of aspect ratios (0.5, 1, 2)
- Stride length (downscaling performed by resnet head: 16)
- Anchor Scales (8, 16, 32)



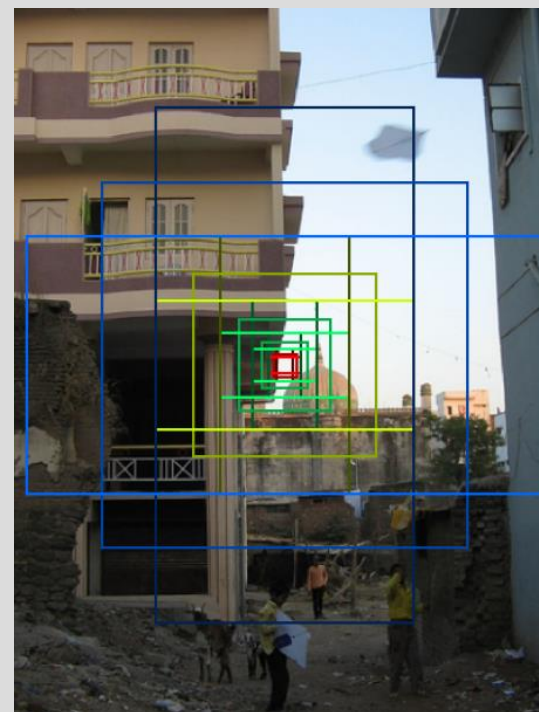
Create uniformly spaced grid with spacing = stride length

Total number of anchors: $1900 * 9 = 17100$
Some boxes lie outside the image boundary



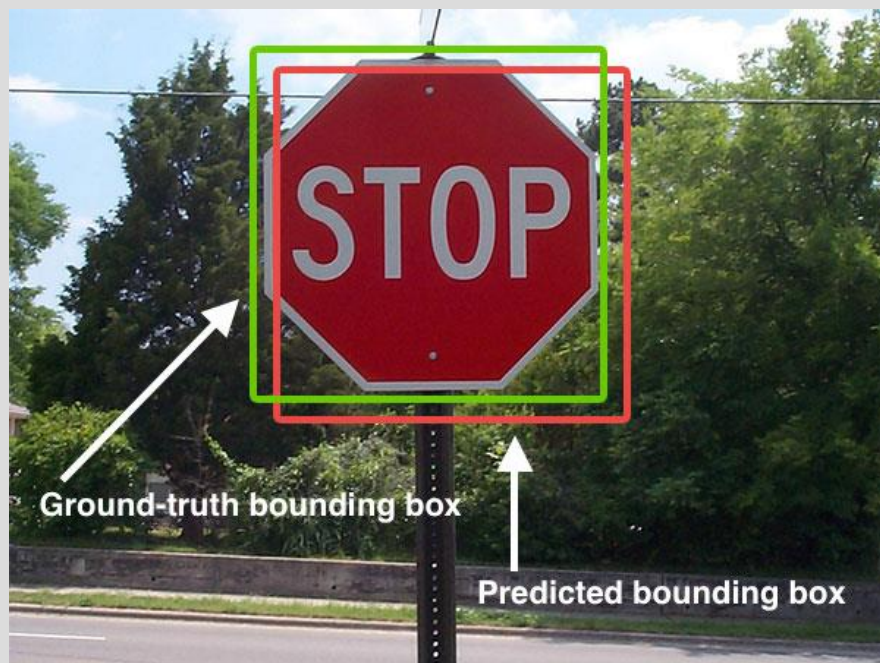
RPN_ANCHOR RATIOS
[0.5, 1, 2]

RPN_ANCHOR_SCALES
[32, 64, 128, 256, 512]



Region Proposal: anchor box与Ground Truth(GT)

--通过IoU 确定正负样本



$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



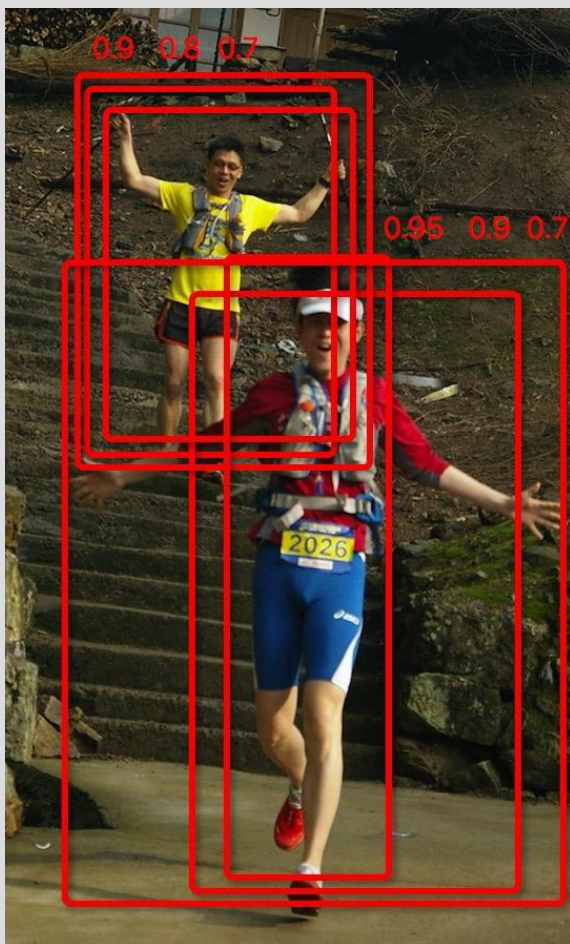
计算每个Anchors与该图片上标注的真实框ground truth之间的IoU

如果anchor box与ground truth的IoU值最大，标记为正样本，label=1

如果anchor box与ground truth的IoU>0.7，标记为正样本，label=1

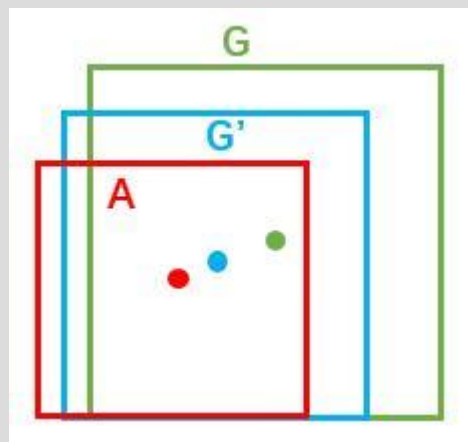
如果anchor box与ground truth的IoU<0.3，标记为负样本，label=-1

region proposal:使用 NMS 找到最合适的anchor box



进一步对预测框进行越界剔除和使用nms非最大值抑制，剔除掉重叠的框；
比如，设定IoU为0.7的阈值，即仅保留覆盖率不超过0.7的局部最大分数的
box（粗筛）。

最后留下大约2000个anchor，然后再取前N个box（比如300个）



按置信率排序: 0.95, 0.9, 0.9, 0.8, 0.7, 0.7

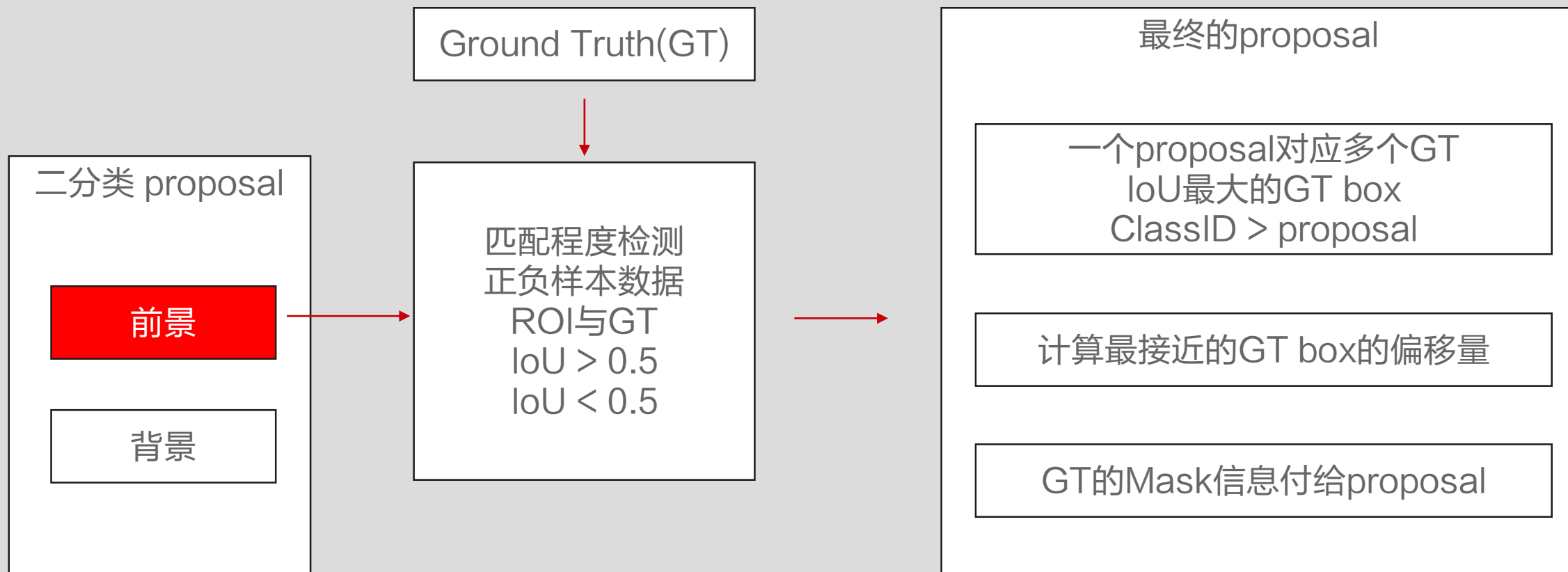
取最大0.95的框为一个物体框

剩余5个框中，去掉与0.95框重叠率IoU大于0.6(可以另行
设置)，则保留0.9, 0.8, 0.7三个框

重复上面的步骤，直到没有框了，0.9为一个框

选出来的为: 0.95, 0.9

生成 head (RCNN)网络数据集



想做团队的领跑者 需要迈过这些“槛”

成长型企业，易忽视人才体系化培养
企业转型加快，团队能力又跟不上

VS

从基础到进阶，超100+一线实战
技术专家带你系统化学习成长

团队成员技能水平不一，
难以一“敌”百人需求

VS

解决从小白到资深技术人所遇到
80%的问题

寻求外部培训，奈何价更高且
集中式学习

VS

多样、灵活的学习方式，包括
音频、图文 和视频

学习效果难以统计，产生不良循环

VS

获取员工学习报告，查看学习
进度，形成闭环

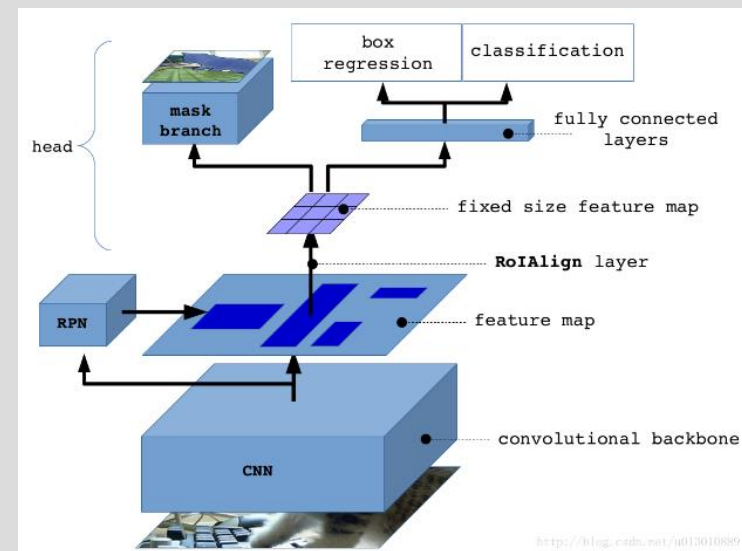
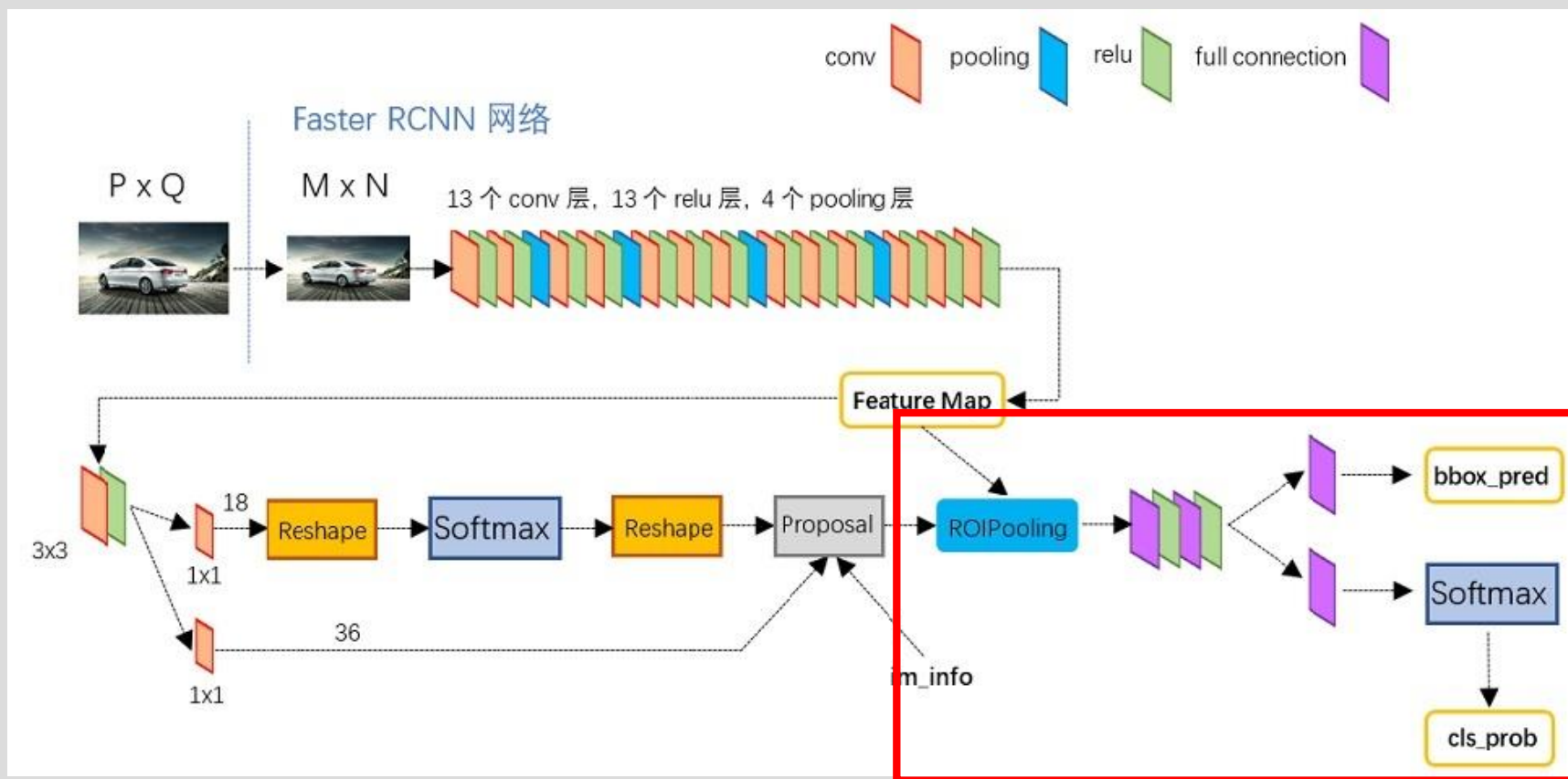


课程顾问「橘子」

回复「QCon」
免费获取
学习解决方案

极客时间企业账号 # 解决技术人成长路上的学习问题

分类和目标检测



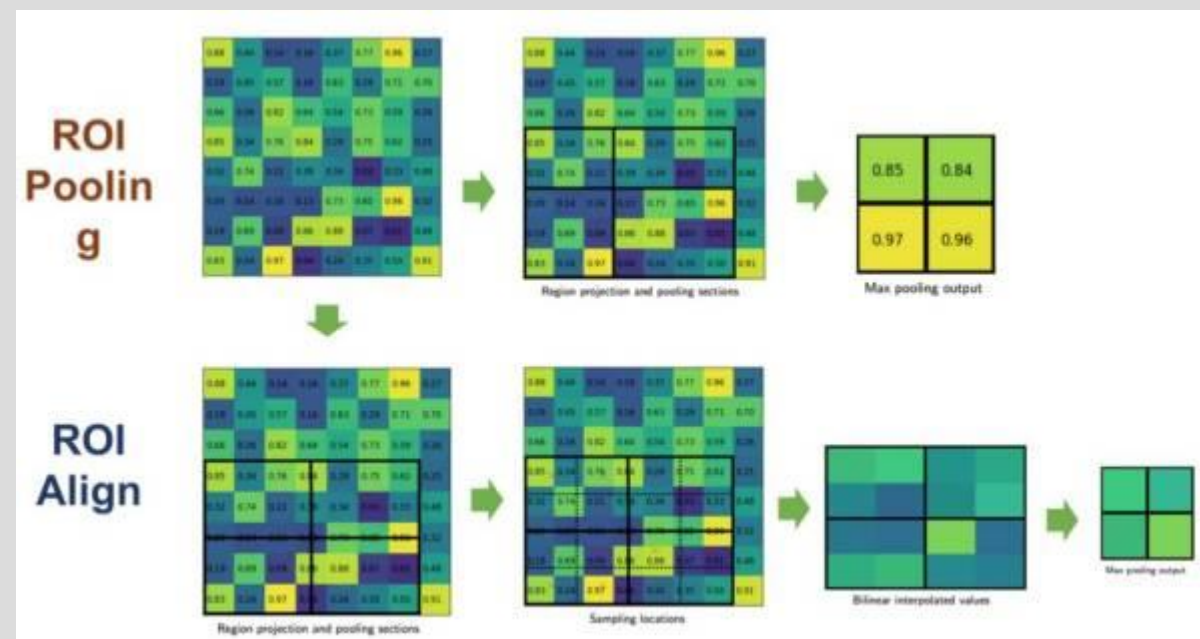
通过full connect层与softmax计算每个region proposal具体属于哪个类别（如人，马，车等），输出cls_prob概率向量

同时再次利用bounding box regression获得每个region proposal的位置偏移量bbox_pred，用于回归获得更加精确的目标检测框

RoiPooling & RoiAlign

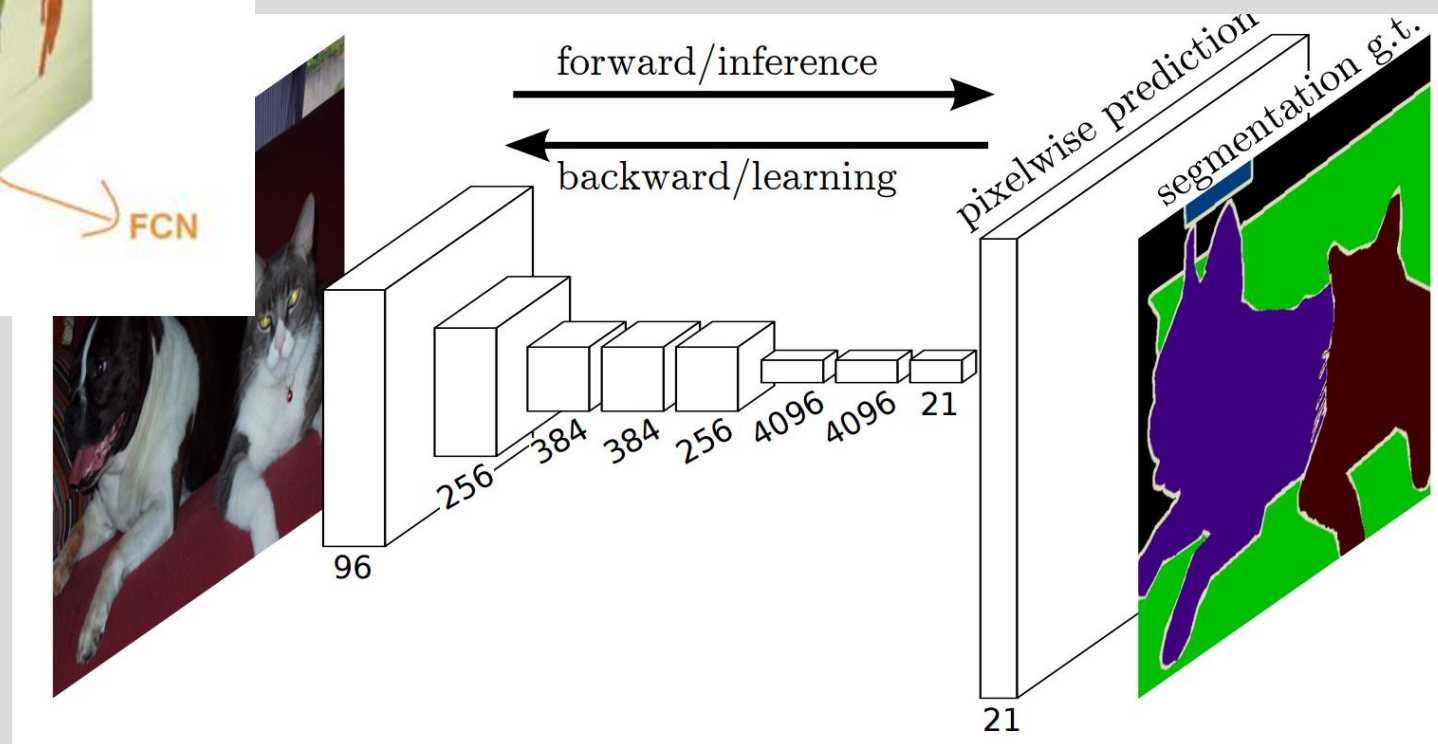
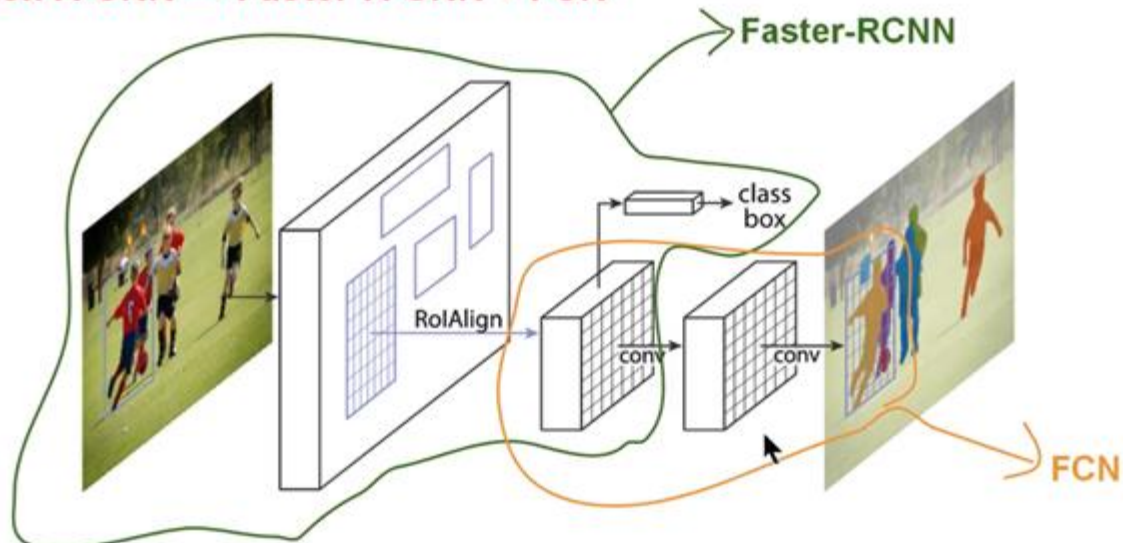
最近邻插值 换成了 双线性插值。换完插值法的 RoiPooling 就有了一个更加高大上的: RoiAlign

RoiPooling&RoiAlign:
对于检测图片中大目标物体时, 两种方案的差别不大, 而如果是图片中有较多小目标物体需要检测, 则优先选择RoiAlign更精准



所谓的Mask: FCN

Mask R-CNN → Faster R-CNN + FCN



ModelArts快速开发： Mask RCNN

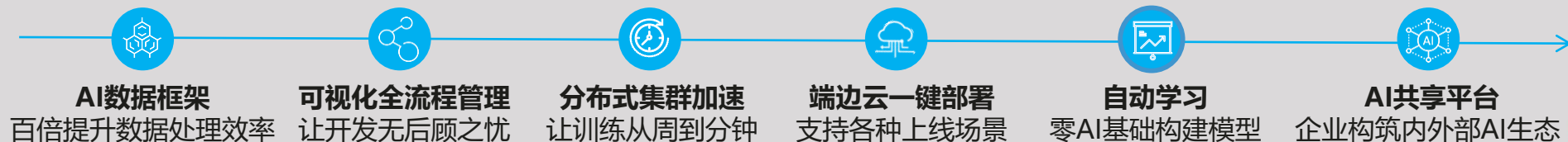
ModelArts 一站式AI开发平台



全流程可视化管理

全流程可视、过程可追溯、数据版本化管理、让开发无后顾之忧

ModelArts




DevRun/19
选择不凡

ModelArts，极快，致简

斯坦福 DAWNBenchmark

图像识别**总训练/推理时间**，ModelArts排名**世界第一**

创新的ExeML技术，**0编码，0 AI基础** 完成训练，
生成匹配目标环境的模型

DAWNBench						
Stanford DAWN						
 DAWNBench An End-to-End Deep Learning Benchmark and Competition						
ImageNet Training						
Submission Date	Model	Time to 93% Accuracy	Cost (USD)	Max Accuracy	Hardware	Framework
Mar 2019	ResNet-50 ModelArts Service of Huawei Cloud source	0:04:08	N/A	93.06%	16 nodes with InfiniBand (8*V100 with NVLink for each node)	TensorFlow v1.8.0
Dec 2018	ResNet-50 ModelArts Service of Huawei Cloud source	0:09:22	N/A	93.23%	16 * 8 * Tesla-V100(ModelArts Service)	Huawei Optimized MXNet
Sep 2018	ResNet-50 fast.ai/DIUX (Yaroslav Bulatov, Andrew Shaw, Jeremy Howard) source	0:18:06	\$118.07	93.11%	16 p3.16xlarge (AWS)	PyTorch 0.4.1
ImageNet Inference						
Submission Date	Model	1-example Latency (milliseconds)	10,000 batch classification cost (USD)	Max Accuracy	Hardware	Framework
Mar 2019	ResNet50 ModelArts Service of Huawei Cloud source	2.4500	N/A	93.40%	Huawei Cloud [pi1.2xlarge.4]	TensorFlow 1.8.0
Dec 2018	ResNet50 Perseus AI Cloud Acceleration team in Alibaba Cloud source	4.2180	\$0.02	93.16%	Alibaba Cloud [ecs.gn5i-c8g1.2xlarge]	TensorFlow 1.10.0

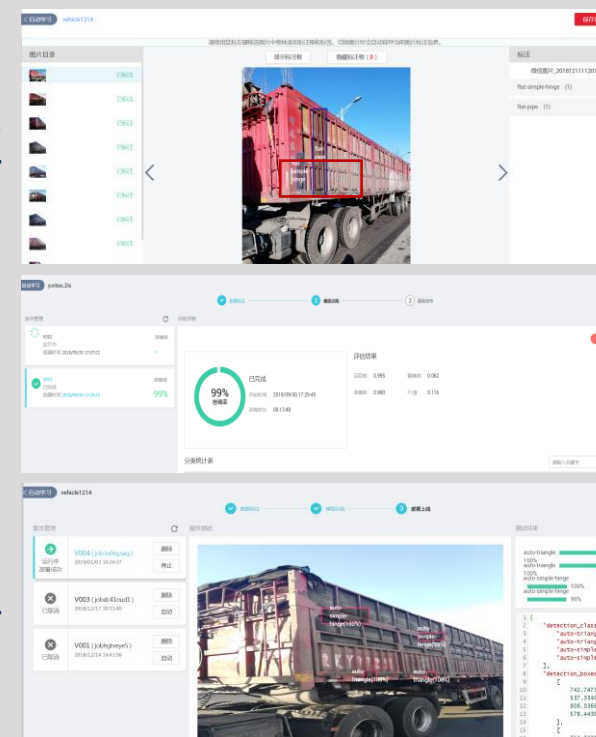
训练：
4分08秒

推理：
2.45ms

1 数据上传&标注

2 模型训练

3 模型验证&发布



DevRun/19
选择不凡

AI Market, 打造全新AI生态圈

- 经典预训练模型
- 行业使能模型
- 开源预训练模型
- 智能终端Skill

获取

上传

- 企业AI开发团队
- 高校/科研院所
- 行业ISV开发者
- 个人开发者

AI 市场

搜索模型

模型 数据集 API

已选 0 个条件 > 展开

全部 公共 发布给我的

Model Hub

Yolo_v3 Object Detector

这个模型使用Yolo_v3的目标检测框架, 使用darknet53作为backbone, 该模型在PASC3A...

HuaweiEI

1166 2019-03-20

HUAWEI CLOUD AI Developer Contest 2019 Hong Kong Gold Medal

香港爱AI美食挑战赛金牌冠军模型

来自香港科技大学的BDT团队获得了2019香港AI开发者大赛爱AI美食挑战赛香港赛区的第一...

华为云人工智能大赛

1023 2019-03-20

安全帽佩戴识别

安全帽是指对头部起防护作用及其他特定因素引起的伤害起防护作用的帽子。这个模型可以...

广联达

993 2019-03-20

市政街道井盖缺陷分类

这个模型用于预测市政街道上井盖的缺陷类型, 井盖的缺陷类别为正常, 凹陷/凸出, 形...

四海万联

844 2019-03-20

Model Hub

ResNet50_v2 Inference

This resnet_v2_50 model is trained on Image Net. It is able to provide top-5 category pre...

HuaweiEI

463 2019-02-24

基于英雄联盟小地图的英雄位置检测

掌握英雄联盟中每个英雄的位置对于教练分析比赛信息尤为重要, 这个模型可以根据英雄...

HuaweiEI

823 2019-03-20

人脸关键点定位技术

人脸关键点定位技术是计算机视觉中重要的研究课题, 该技术是人脸识别、人脸美颜、以...

智慧智能

618 2019-03-20

驾驶路面人车对象检测

自动驾驶是一件非常酷的事情, 这个模型可以检测车辆前进方向上的视觉目标, 包含行人/...

HuaweiEI

535 2019-02-24

HUAWEI CLOUD AI Developer Contest 2019 Hong Kong Silver Medal

香港爱AI美食挑战赛银牌亚军模型

来自香港混合大学的A.I. Phoenix团队获得了2019香港AI开发者大赛爱AI美食挑战赛香港...

华为云人工智能大赛

319 2019-03-20

基于AI深度学习的农作物病虫害检测

这个模型可以辅助农业生产者, 通过拍摄或者上传相关图像, 模型反馈Top1或者Top3的...

睿坤科技

312 2019-03-20

钢筋数量AI识别

钢筋是指钢筋混凝土用和预应力钢筋混凝土用钢材, 其横截面为圆形, 这个模型可以帮助...

广联达

298 2019-03-20

安全帽

工业环境进行安全帽的识别

HuaweiEI

277 2019-04-16

DevRun/2019
选择不凡

极客邦科技 会议推荐2019



ModelArts Console界面

新手入门

[关闭新手入门](#)

自动学习

自动学习功能，可根据标注数据自动设计模式、自动调参、自动训练、自动压缩和部署模型，不需要代码编写和模型开发经验，即可实现零基础构建AI模型。



图像分类



物体检测



预测分析



声音分类

AI全流程开发

如果您是一位AI开发者，ModelArts能够帮助您管理整个AI开发流程，让您专心做开发，加倍提高生产力！



数据管理



开发环境



训练作业

开始体验



模型管理



部署上线

总览

自动学习

图像分类

0 计费中 2 项目

exeML-0422

AI全流程开发

开发环境

0 计费中 1 实例

notebook-190410

训练作业

0 计费中 1 / 1 版本 / 作业

trainjob-190410

模型管理

0 计费中 5 模型

exeML-20190404_ExeML_1...

部署上线

0 计费中 4 服务

exeML-20190404_ExeML_1...

ModelArts

三步轻松生成自己的模型

费用

购买套餐包

产品定价

购买模型调优

最新特性

版本特性说明

常用文档

用户指南

常见问题

ModelArts论坛

使用MXNet实现手写数字图像...

华为云EI ModelArts实战全纪录

常用链接

软件开发服务 DevCloud

前往GitHub问题区

云存储OBS

IoT平台

云市场

[查看更多](#)

咨询·反馈