



# Pytorch(下)II



**主讲人 王博士**

中国科学院自动化研究所博士。主要研究领域包括目标识别、语义分割、深度学习。曾经参加全国视频图像分析技术挑战赛，获得目标检测识别第二名。



- ✓ 目标识别介绍
- ✓ 传统方法和深度学习对比
- ✓ Faster RCNN介绍
- ✓ **Faster RCNN代码详解**

□运行环境: Linux, python 2.7/3.6, pytorch 0.4.0/0.4.1, CUDA 8.0+

□安装必要的python包: `pip install -r requirements.txt`

□下载并解压VOC数据集:

```
cd faster-rcnn.pytorch && mkdir data  
ln -s /your/dataset VOCdevkit2007
```

□下载vgg16 pretrain model:

```
cd data && mkdir pretrained_model  
ln -s /your/model vgg16_caffe.pth
```

## □编译一些辅助代码/层:

```
cd lib && sh make.sh
```

## □目的:

- `python setup.py build_ext --inplace`
  - `model/utils/bbox.pyx` #计算box之间IoU
  - `pycocotools/maskApi.c` #读取coco数据集中二值掩膜
- `model/nms/src/*` #NMS gpu代码
- `model/roi_pooling/src` #RoI pooling 层 (SPP空间金字塔池化的简化版)
- `model/roi_align/src` # RoI align 层
- `model/roi_crop/src` # RoI crop 层

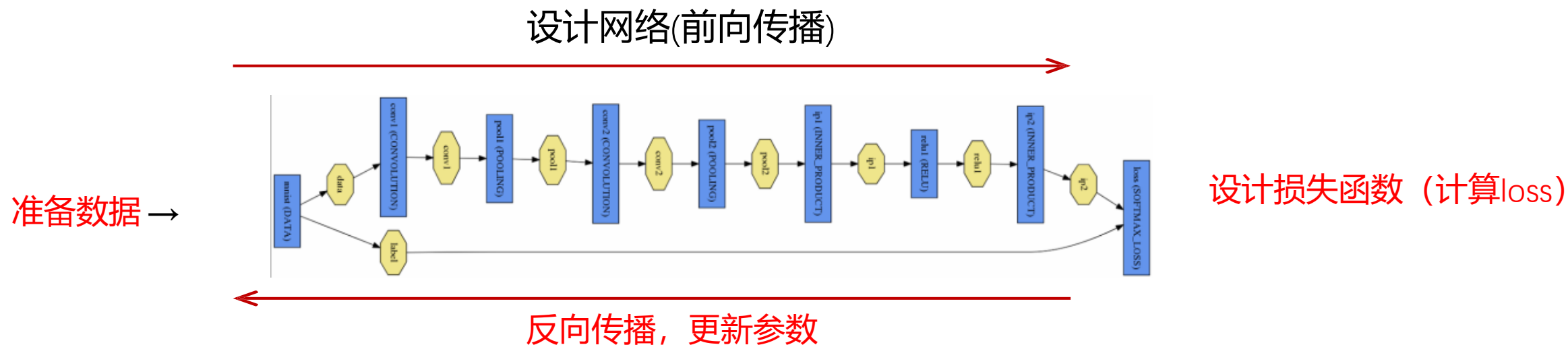
## □用途：

- 实现新的功能，但是pytorch不支持
- 代码运行速度有要求

## □途径—C拓展：

- 准备.c文件实现
- 利用工具编译为python可以调用的模块
- 嵌入到pytorch网络中

## □实例展示



主函数: `trainval_net.py`

- 准备数据: Dataset+DataLoader
- 网络设计: vgg16+ faster rcnn
- 损失计算: label准备, 调用loss计算
- 参数更新: for循环迭代

- ❑ 读取数据集中全部图像和gt boxes 信息到roidb, `lib/roi_data_layer/roidb.py::combined_roidb(args.imdb_name)`
- ❑ 利用roidb初始化dataset类, `lib/roi_data_layer/roidb.py:: class roibatchLoader(data.Dataset)`
- ❑ 用dataset初始化dataloader

dataset中getitem函数返回内容: data, im\_info, gt\_boxes, num\_boxes,

其中利用`lib/roi_data_layer/minibatch.py:: get_minibatch`获取原始图像和gt boxes

注:

数据集图像的短边resize到600

同一个batch中图像的长宽比保持一致

## □lib/model/faster\_rcnn

- vgg16.py

- resnet.py

全部继承\_fasterRCNN(lib/model/faster\_rcnn/faster\_rcnn.py)

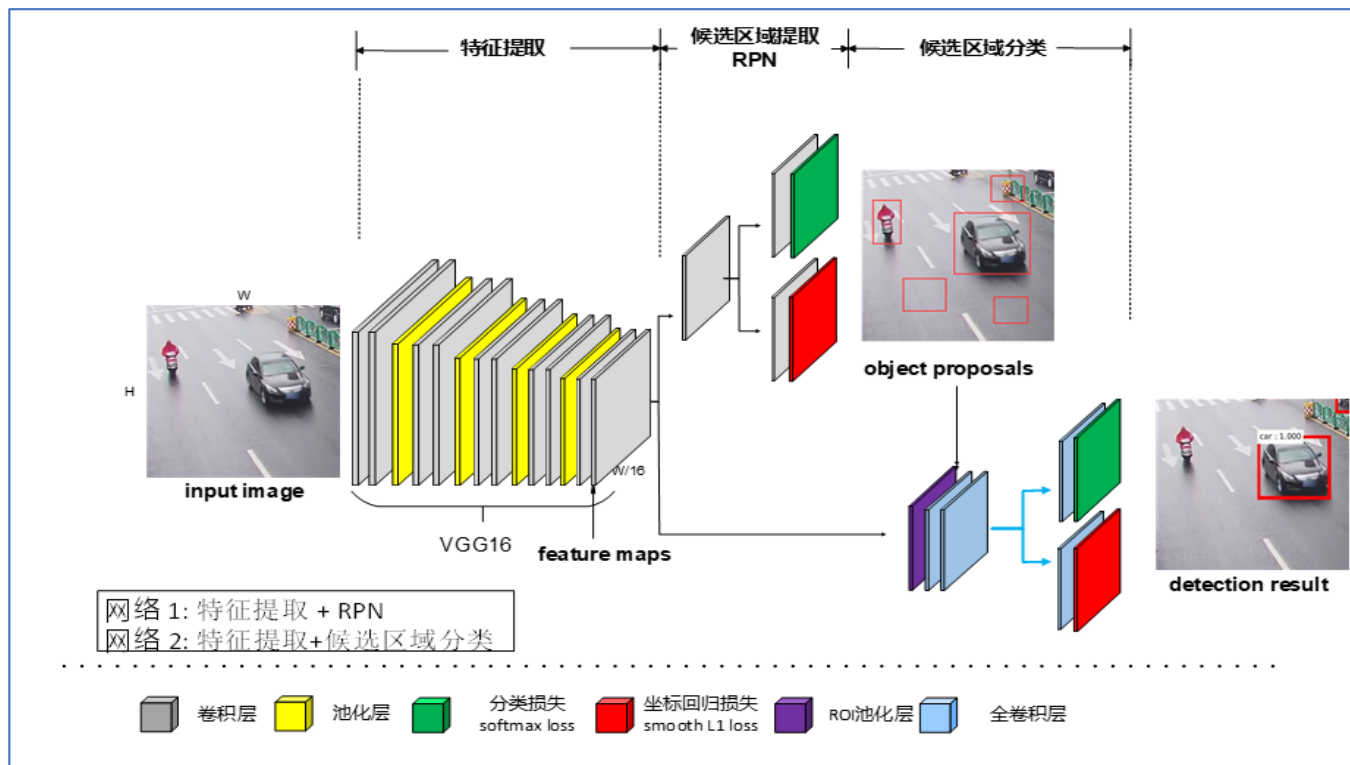
## □主要的类成员:

self.RCNN\_base #vgg网络部分

self.RCNN\_rpn #RPN部分

self.RCNN\_cls\_score # fast rcnn 分类

self.RCNN\_bbox\_pred #fast rcnn 坐标回归



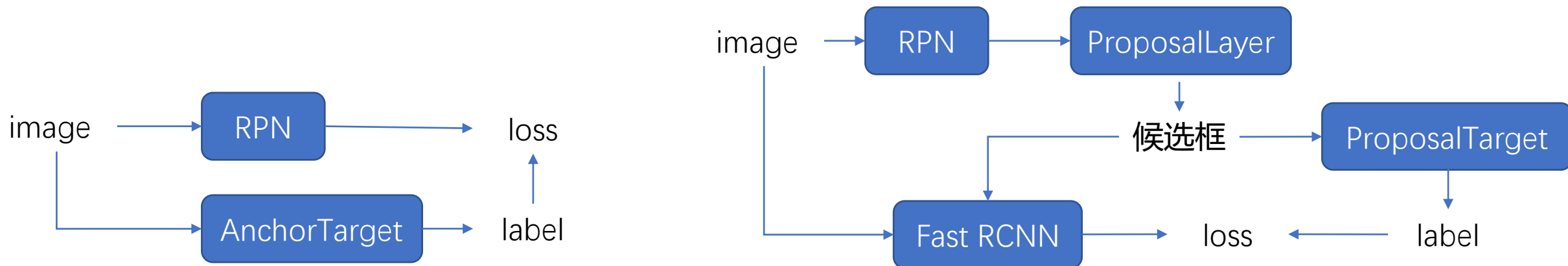


# 损失计算—生成各种label

## □三个layer

- AnchorTargetLayer: 负责在训练RPN的时候, 从上万个anchor中选择一些(比如256)进行训练, 以使得正负样本比例大概是1:1. 同时给出训练的位置参数目标。即返回 `gt_rpn_loc`和`gt_rpn_label`。
- ProposalLayer: 在RPN输出的候选区域, 选取得分较高的N个候选框, 然后进行NMS, 然后选择一定数目 (2000或者300), 用以候选区域分类训练或者测试。
- ProposalTargetLayer: 负责在训练候选区域分类的时候, 从RoIs选择一部分(比如128个)用以训练。同时给定训练目标, 返回 (`sample_Roi`, `gt_Roi_loc`, `gt_Roi_label`)

AnchorTargetCreator和ProposalTargetCreator是为了生成训练的目标, 只在训练阶段用到, ProposalCreator是RPN为第二个分类网络生成RoIs, 在训练和测试阶段都会用到。三个共同点在于他们都不需要考虑反向传播

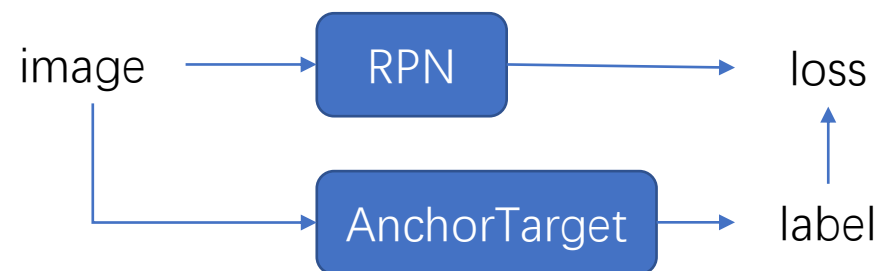
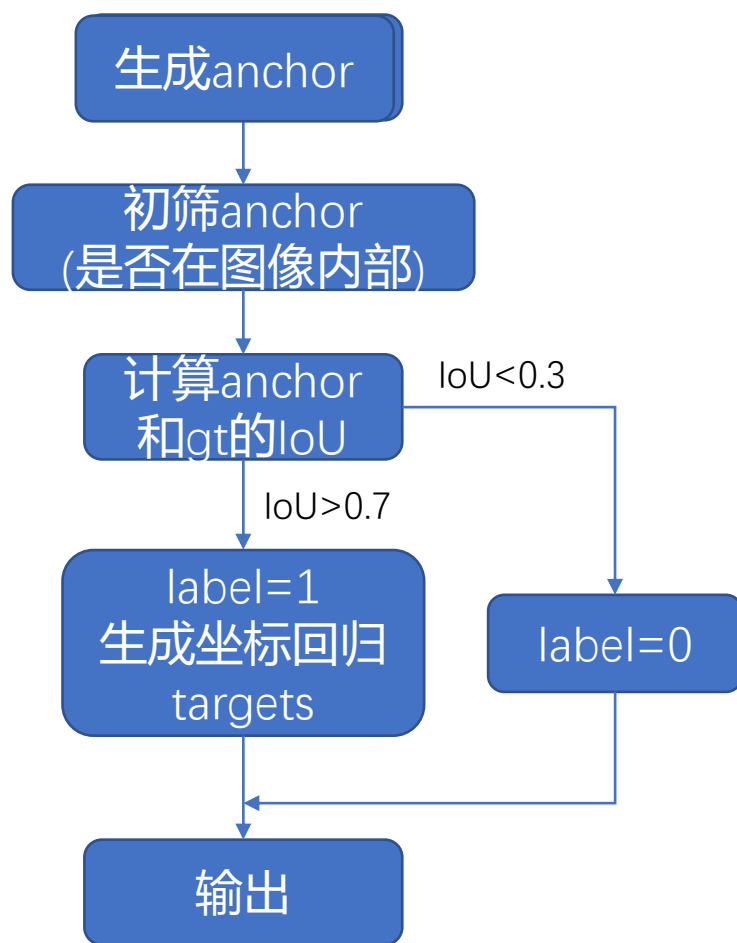


## □关于矩形框的描述：

- Bbox: bounding box, 边界框。其中Ground Truth Bounding Box是每一张图中人工标注的框的位置。一张图中有几个目标, 就有几个框(一般小于10个框)。Faster R-CNN的预测结果也可以叫 bounding box, 不过一般叫 Predict Bounding Box.
- Anchor: 人为在图像中假想的具有一定尺度、比例的框。一个feature map的锚的数目有上万个 (比如 20000) 。
- RoI: region of interest, 候选矩形框。Faster R-CNN之前传统的做法是利用selective search从一张图上大概2000个候选框框。现在利用RPN进行候选框提取。
- loc: bbox, anchor和RoI, 本质上都是一个框, 可以用四个数 ( $y_{min}$ ,  $x_{min}$ ,  $y_{max}$ ,  $x_{max}$ ) 表示框的位置, 还可以用 ( $y$ ,  $x$ ,  $h$ ,  $w$ ) 表示, 即框的中心坐标和长宽。Ground truth中记录的是前一种方式, 在训练中进行位置回归的时候, 用的是后一种的表示。

# AnchorTargetLayer

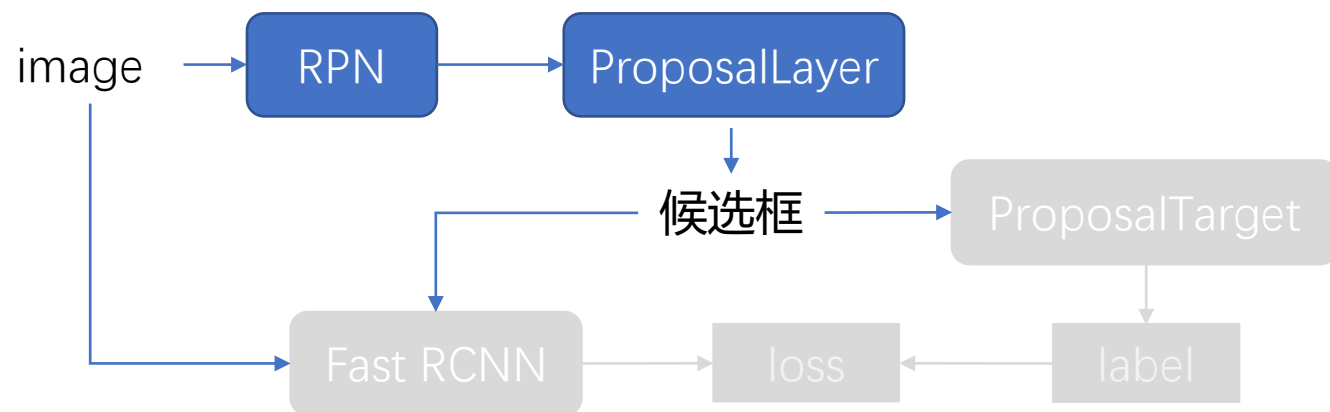
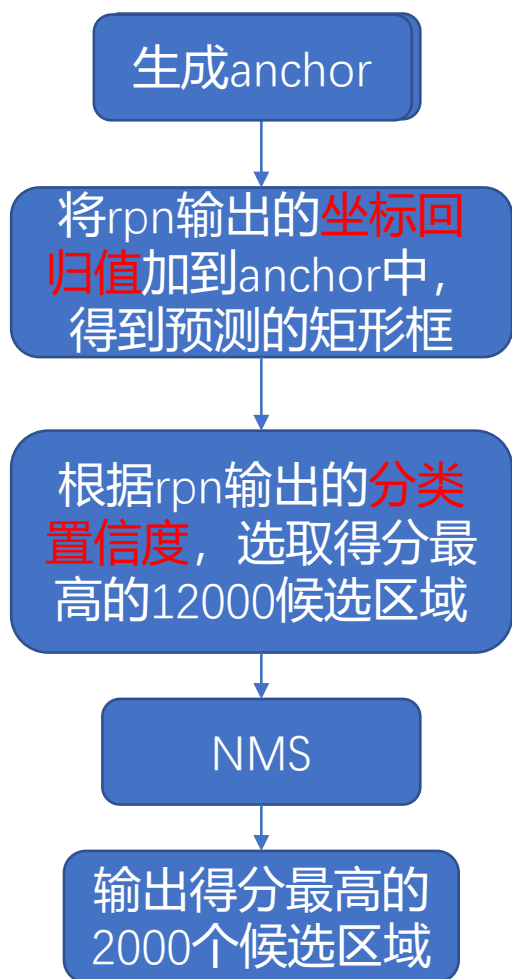
- 负责在训练RPN的时候，从上万个anchor中选择一些(比如256)进行训练，以使得正负样本比例大概是1:1. 同时给出训练的位置参数目标。即返回gt\_rpn\_loc和gt\_rpn\_label。



并非所有符合条件的anchor都参与计算  
每张图像，选取随机选取256个anchor参与计算，其中正负样本比例为1:1

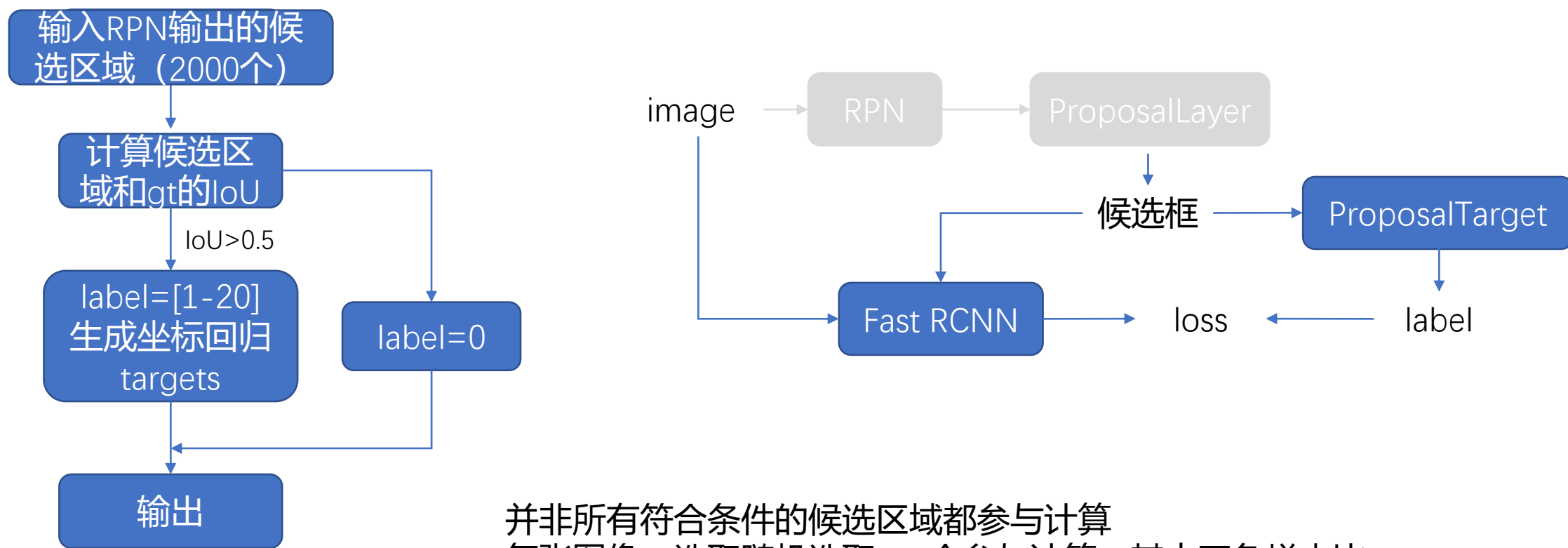
# ProposalLayer

- 在RPN输出的候选区域，选取得分较高的N个候选框，然后进行NMS，然后选择一定数目(2000或者300)，用以候选区域分类训练或者测试



# ProposalTargetLayer

- 负责在训练候选区域分类的时候，从RoIs选择一部分(比如128个)用以训练。同时给定训练目标，返回 (sample\_RoI, gt\_RoI\_loc, gt\_RoI\_label)



并非所有符合条件的候选区域都参与计算  
每张图像，选取随机选取128个参与计算，其中正负样本比例为1:3

## □四个损失

- RPN 分类损失: anchor是否为前景 (二分类)
- RPN位置回归损失: anchor位置微调
- RoI 分类损失: RoI所属类别 (21分类, 多了一个类作为背景)
- RoI位置回归损失: 继续对RoI位置微调

四个损失相加作为最后的损失, 反向传播, 更新参数。

将faster rcnn特征提取网络换做alexnet，其他参数不变，训练voc2007数据集，计算mAP，并可视化在voc2007测试集效果





**感谢各位聆听**  
**Thanks for Listening**