

# 通用物体检测



主讲人 张士峰

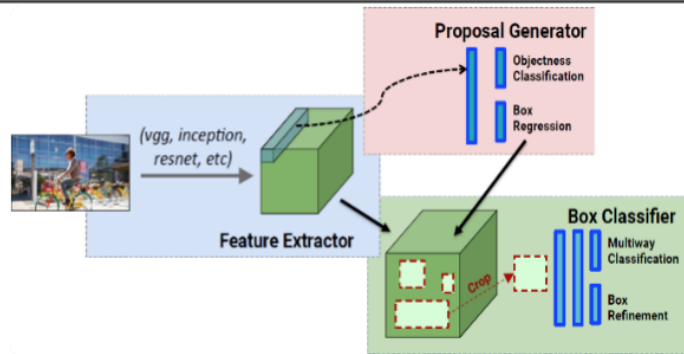
中国科学院自动化研究所  
模式识别国家重点实验室



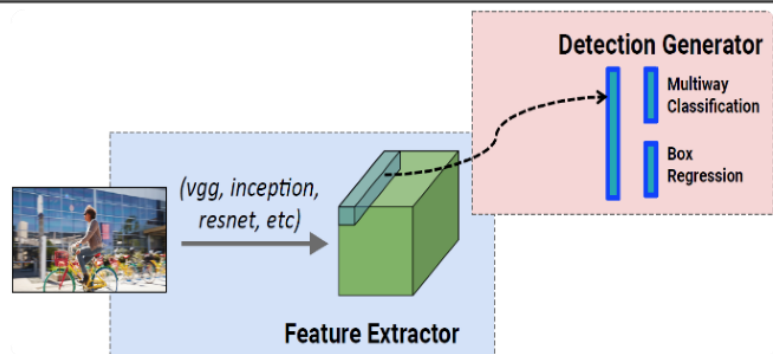


## 内容回顾：物体检测算法的总结

### 基于锚框

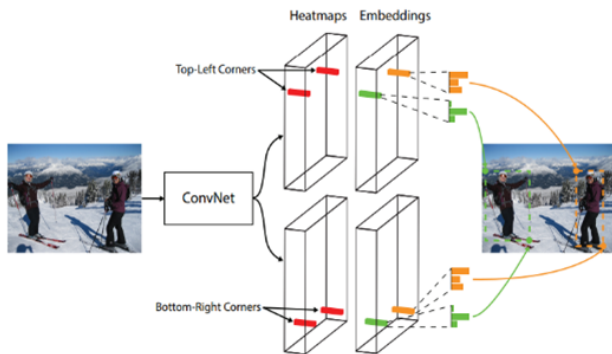


多阶段法

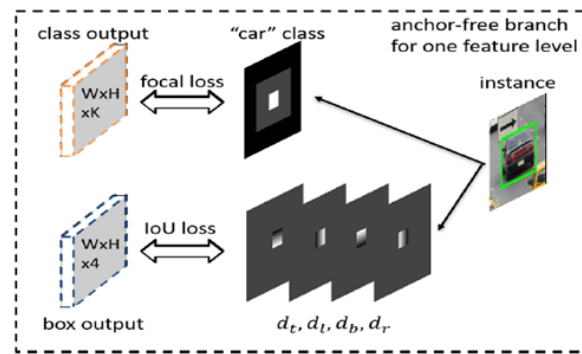


单阶段法

### 无需锚框



关键点法

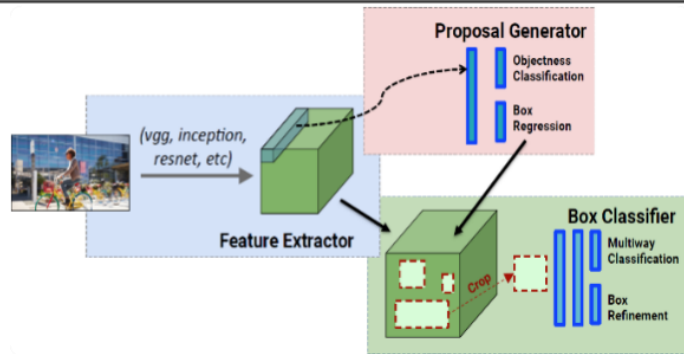


中心域法

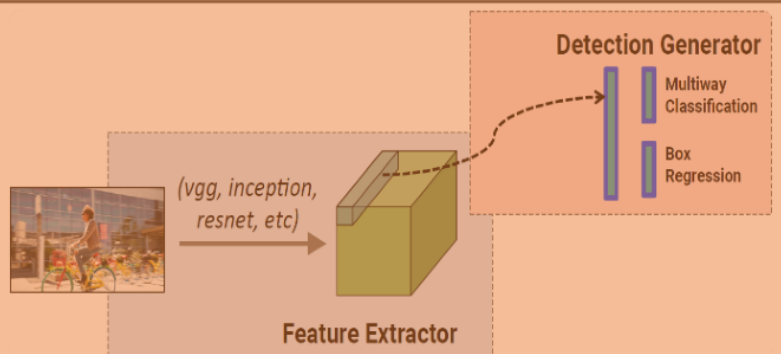


# 内容回顾：物体检测算法的总结

基于锚框

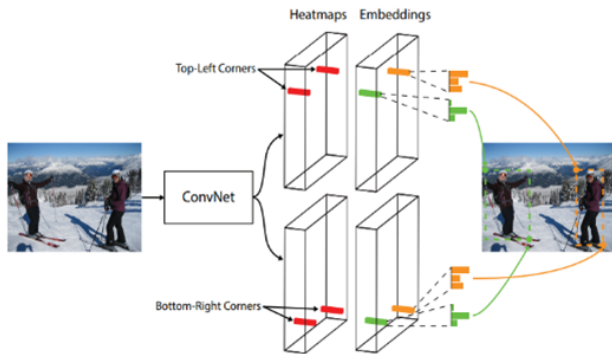


多阶段法

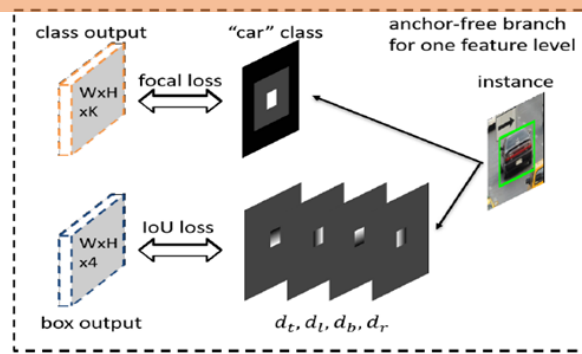


单阶段法

无需锚框



关键点法

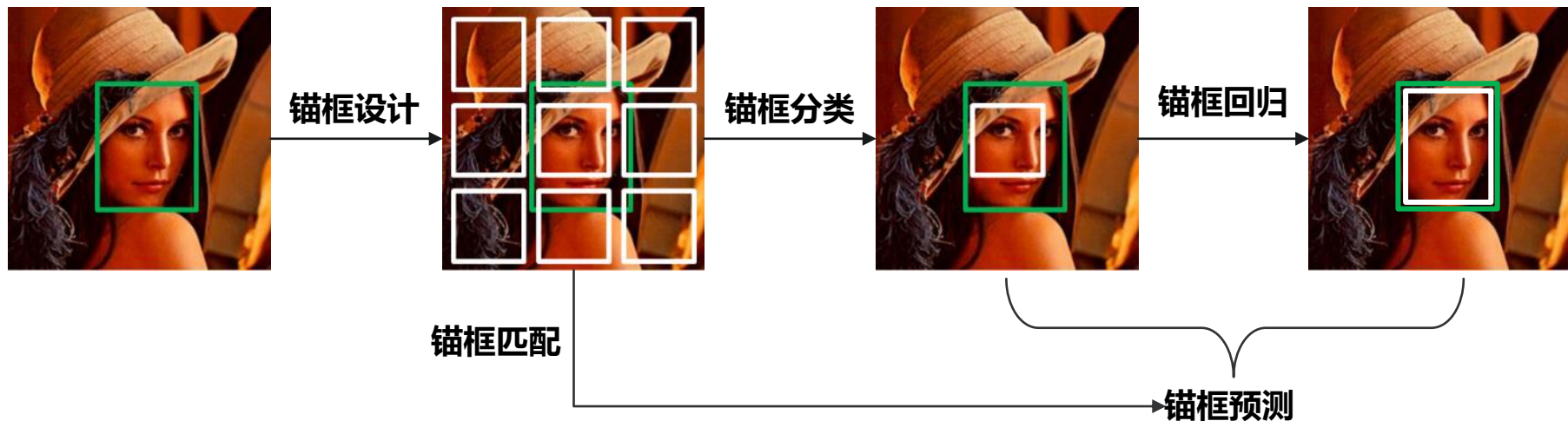


中心域法



## 内容回顾：物体检测算法的总结

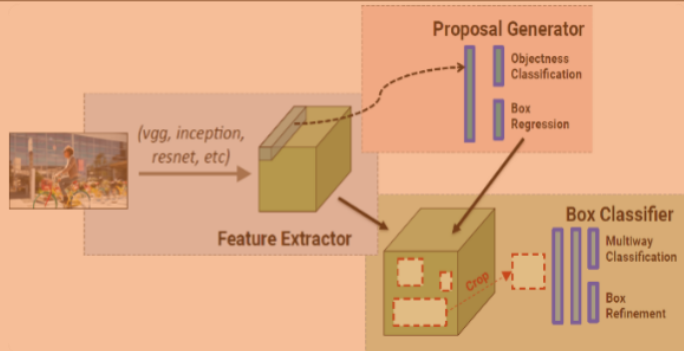
### 基于锚框的单阶段法 (SSD/RetinaNet)



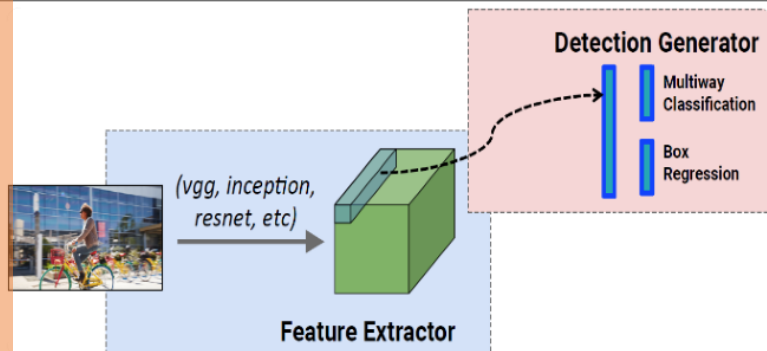


## 内容回顾：物体检测算法的总结

基于锚框

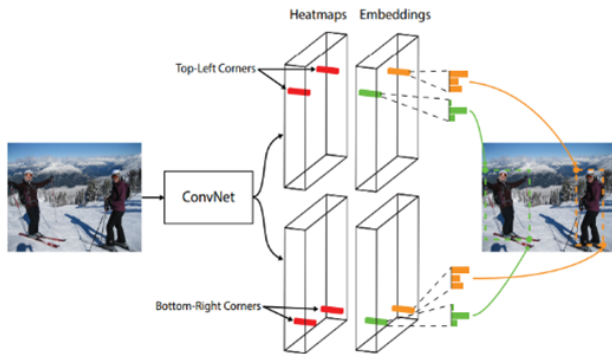


多阶段法

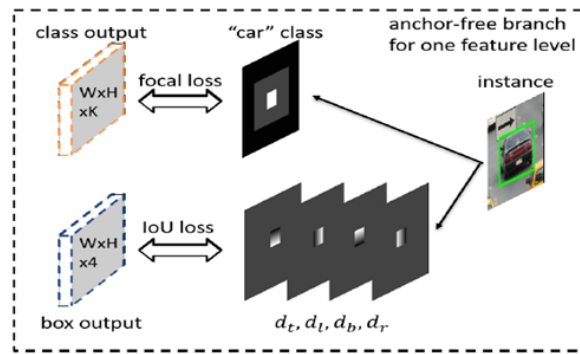


单阶段法

无需锚框



关键点法

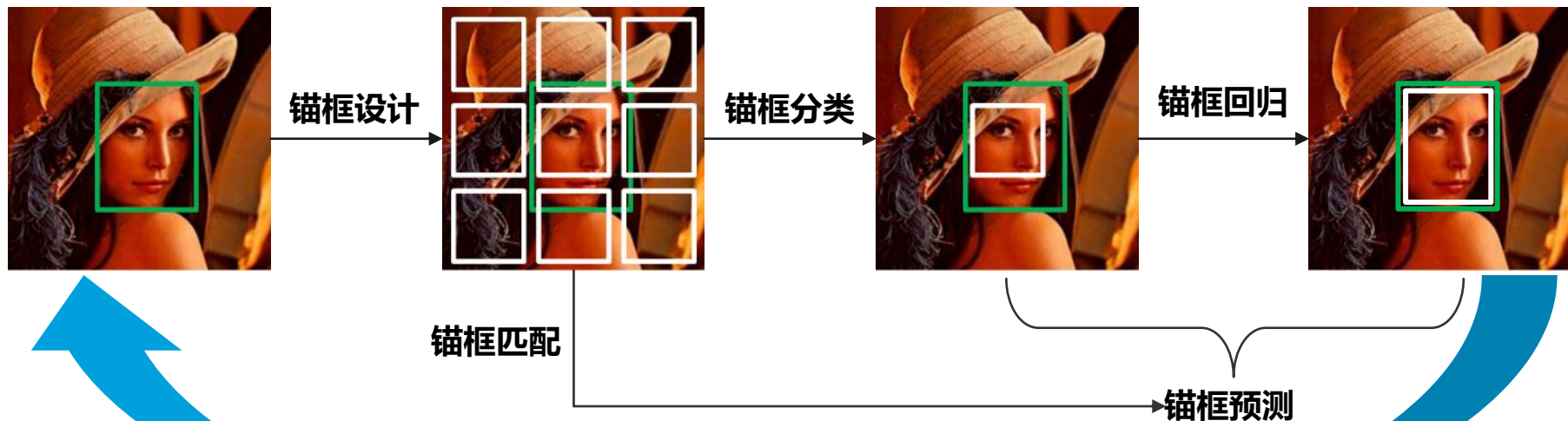


中心域法



## 内容回顾：物体检测算法的总结

### 基于锚框的多阶段法 (Faster R-CNN)



级联地重复这个过程



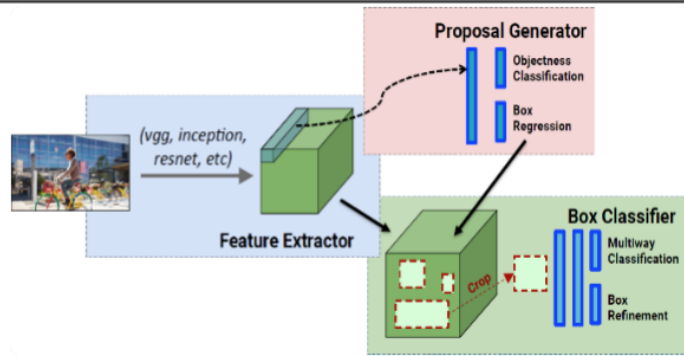
## 内容回顾：物体检测算法的总结（基于锚框）

基于锚框的检测算法		多阶段法	单阶段法
相同点	检测思想	铺设的锚框为检测起点，对锚框的类别和位置进行矫正	
	检测起点	铺设的锚框	
	检测结果	矫正的锚框	
不同点	难点问题之一	小尺度物体	正负样本的平衡
	锚框矫正次数	$\geq 2$ 次	1次
	检测精度	较高	较低
	检测速度	较慢	较快

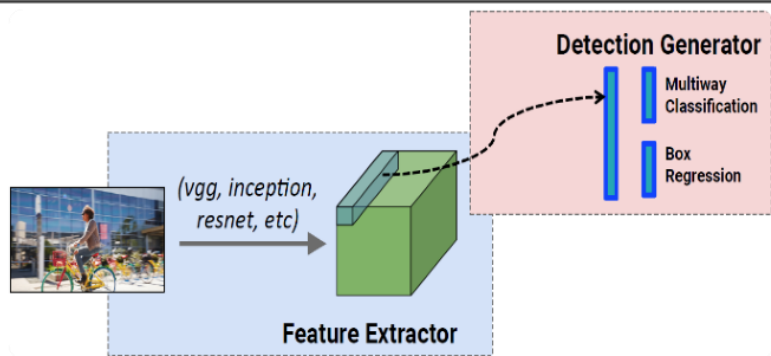


# 内容回顾：物体检测算法的总结

基于锚框

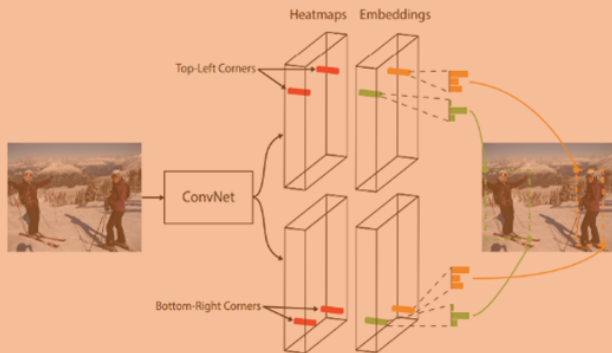


多阶段法

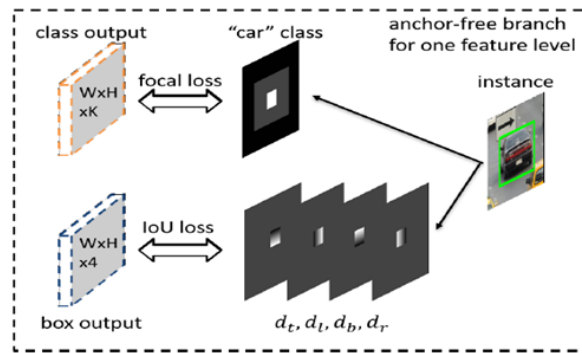


单阶段法

无需锚框



关键点法



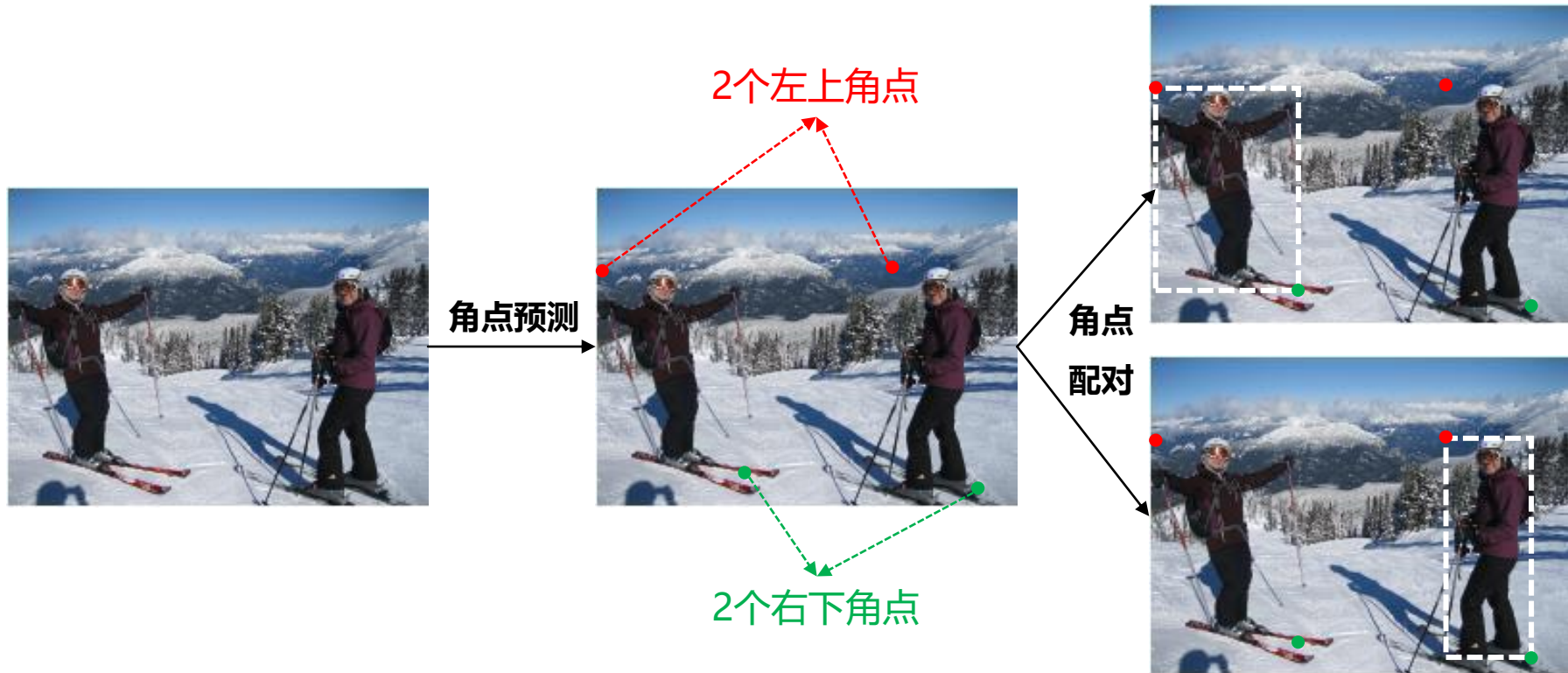
中心域法





## 内容回顾：物体检测算法的总结

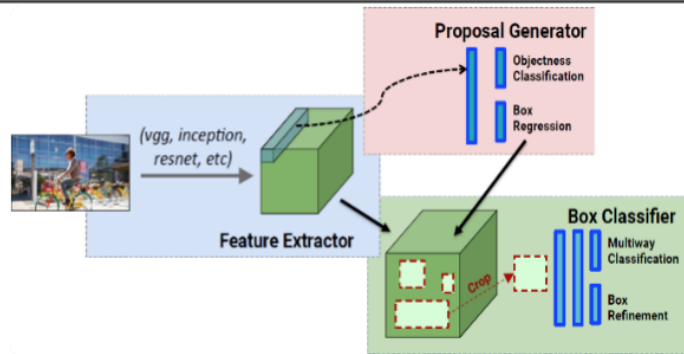
# 无需锚框的关键点法 (CornerNet)



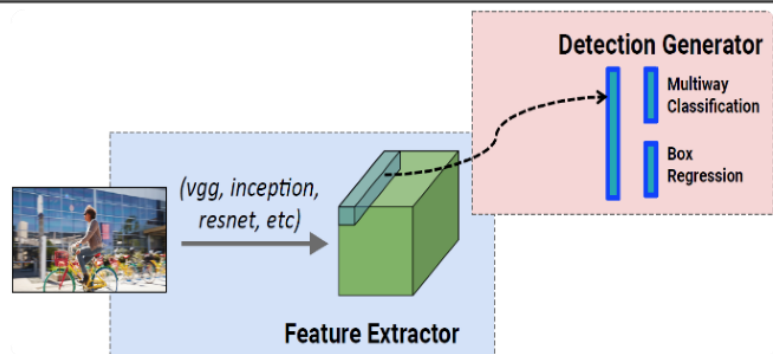


# 内容回顾：物体检测算法的总结

## 基于锚框

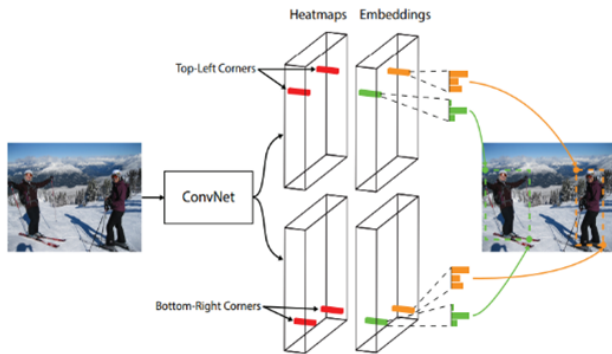


多阶段法

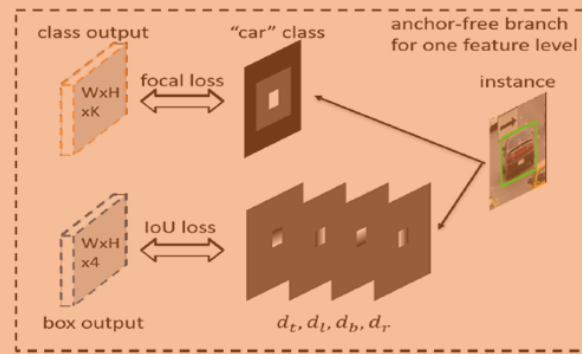


单阶段法

## 无需锚框



关键点法

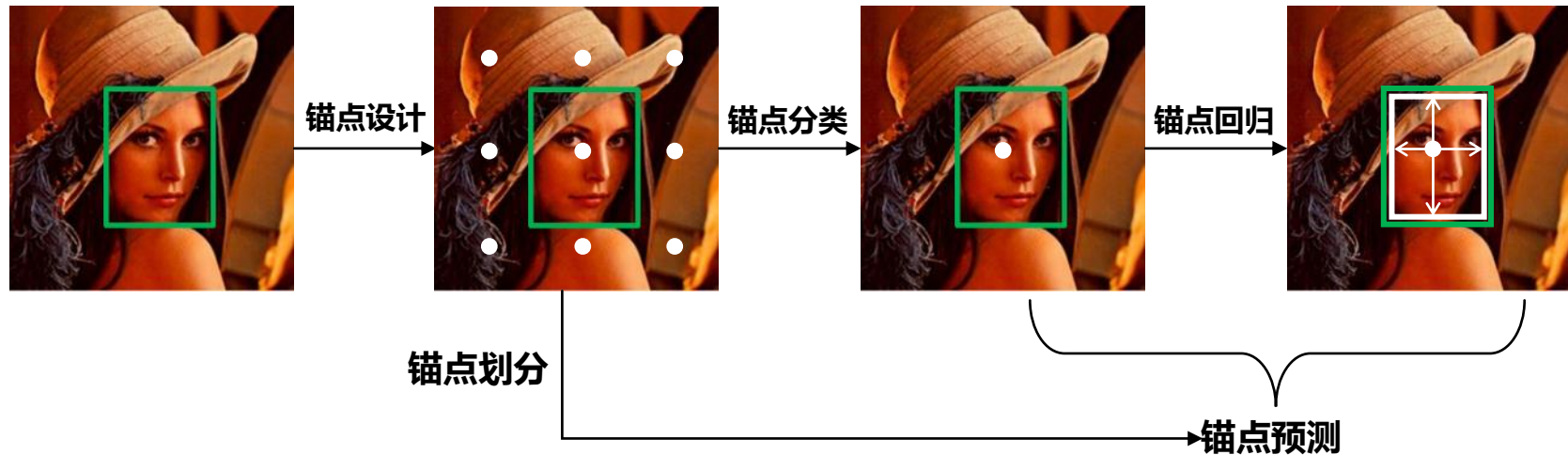


中心域法



## 内容回顾：物体检测算法的总结

# 无需锚框的中心域法（FCOS）





## 内容回顾：物体检测算法的总结（无需锚框）

无需锚框算法	关键点法	中心域法
算法动机	移除掉锚框，减少超参数，增加灵活性	
算法思想	先检测关键点，再进行配对来框定物体	铺设锚点替代锚框来检测物体
算法优点	全新的检测流程，为检测带来了新的思路	减少超参数，简化计算
算法难点	不同关键点之间的配对问题	正负样本的划分问题
计算速度	流程比较复杂，速度相对较慢	流程比较简单，速度相对较快
检测精度	精度能达到甚至超过基于锚框的单阶段法	



## 目录



物体检测环境配置



通用物体检测概述



基于锚框的检测算法



无需锚框的检测算法



物体检测算法的对比总结

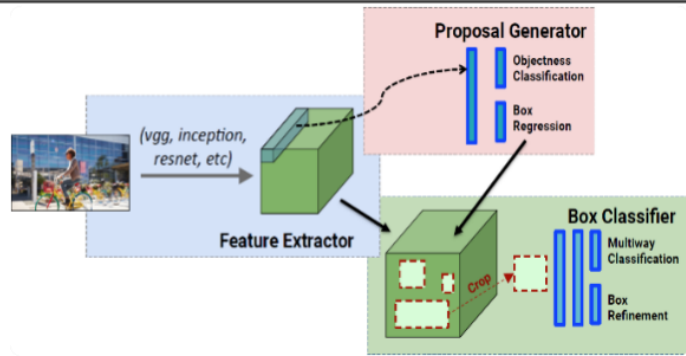


实用检测算法的研究思路

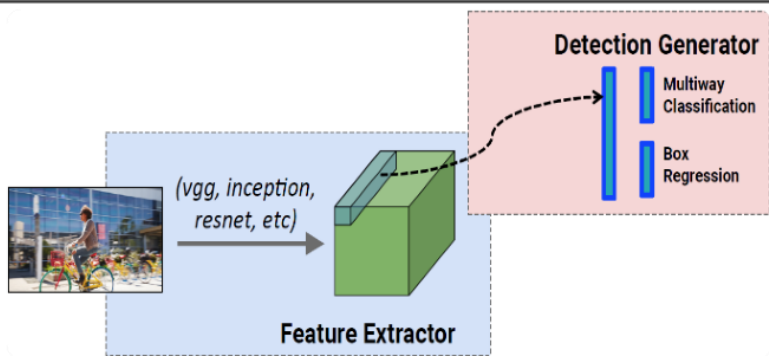


# 实用检测算法的研究思路：对比探索

基于锚框

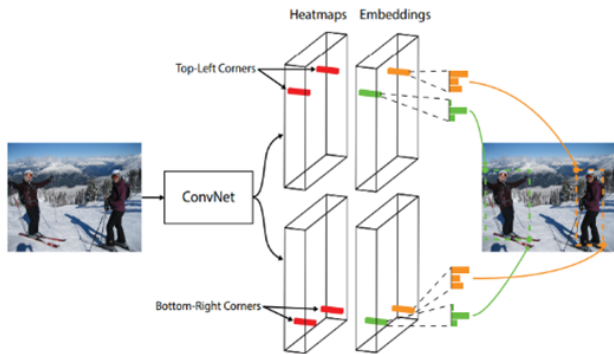


多阶段法

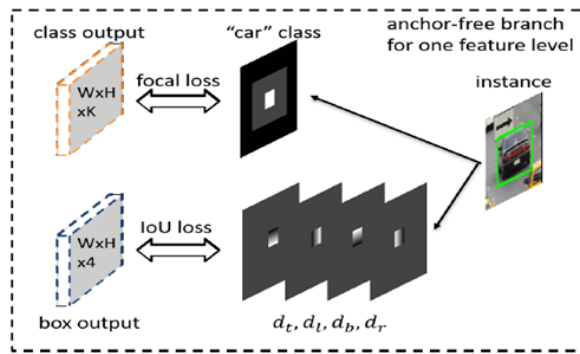


单阶段法

无需锚框



关键点法



中心域法

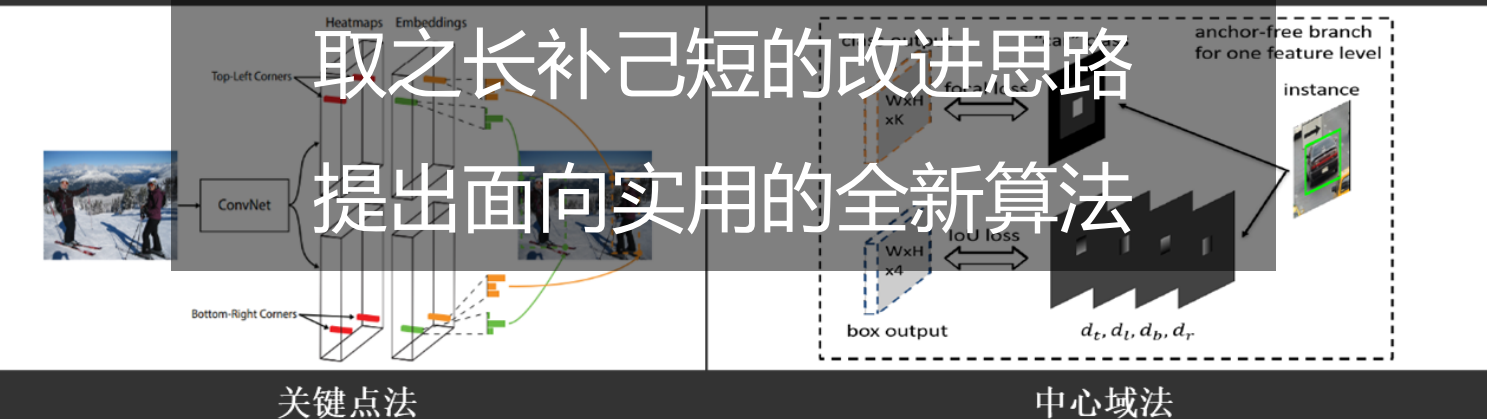


## 实用检测算法的研究思路：对比探索

基于锚框



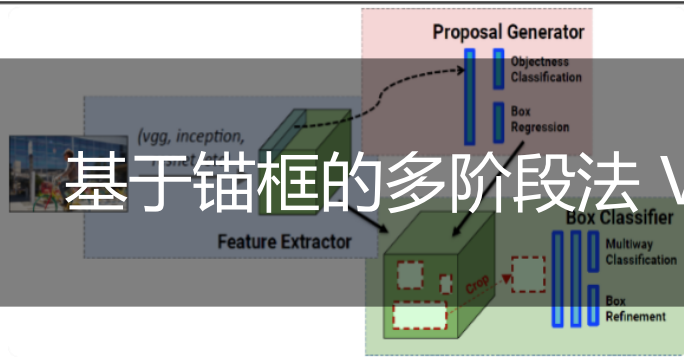
无需锚框



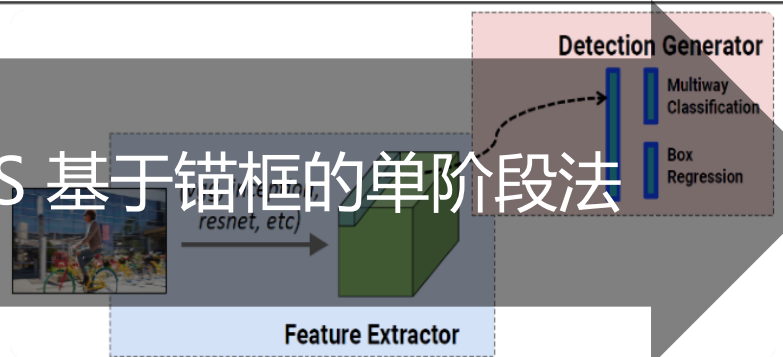


# 基于锚框的多阶段法 VS 单阶段法

基于锚框

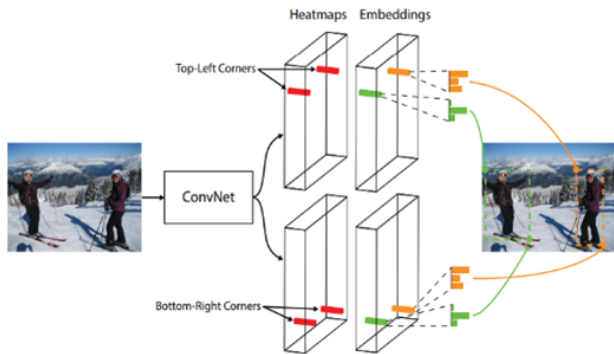


多阶段法

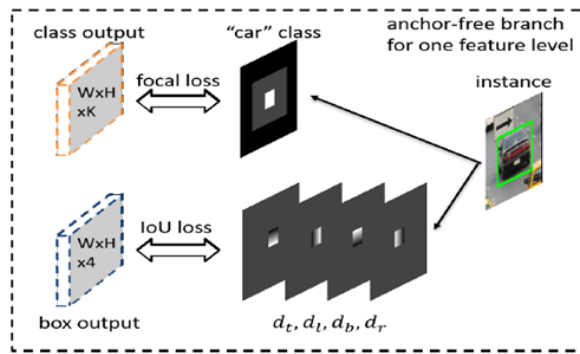


单阶段法

无需锚框



关键点法

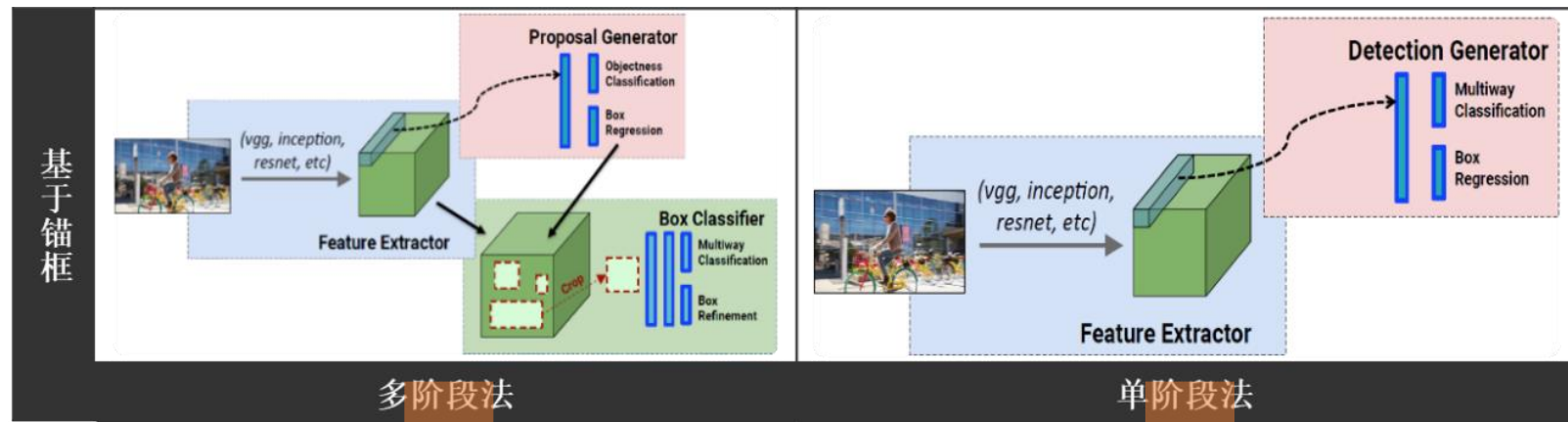


中心域法





## 基于锚框的多阶段法 VS 单阶段法：本质区别



精度较高，但速度较慢

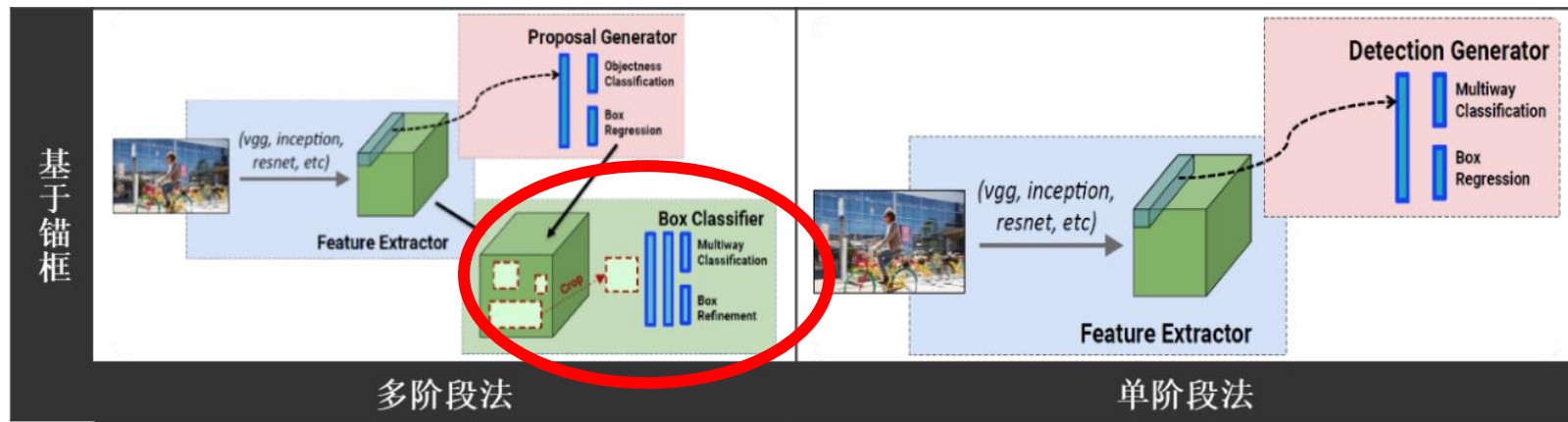
VS

速度较快，但精度较低

哪些区别带来的差异？



## 基于锚框的多阶段法 VS 单阶段法：本质区别

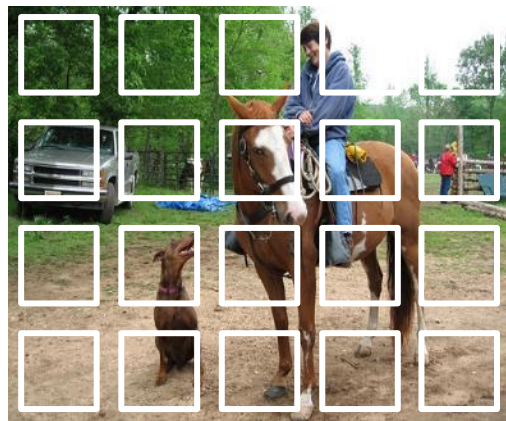


- 多阶段法与单阶段法的主要区别：多了一个额外的检测步骤
- 该步骤让多阶段法具备以下特点：① 二阶段的分类；② 二阶段的回归
- ③ 二阶段的特征；④ 特征的校准

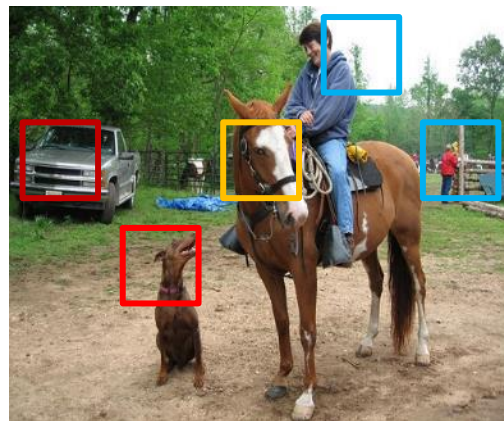
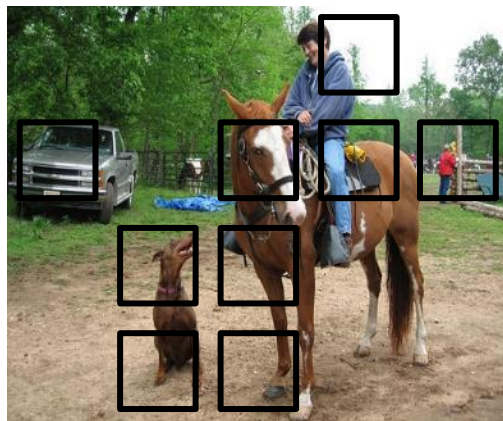


## 基于锚框的多阶段法 VS 单阶段法：二阶段分类

### 多阶段法的二阶段分类过程



第一阶段的二分类  
(背景 or 前景)

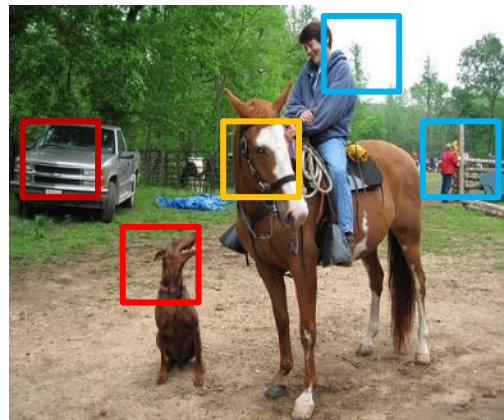
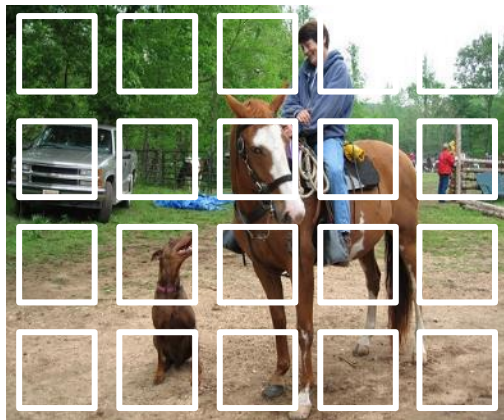


第二阶段的多分类  
(具体类别)



## 基于锚框的多阶段法 VS 单阶段法：二阶段分类

### 单阶段法的一阶段分类过程

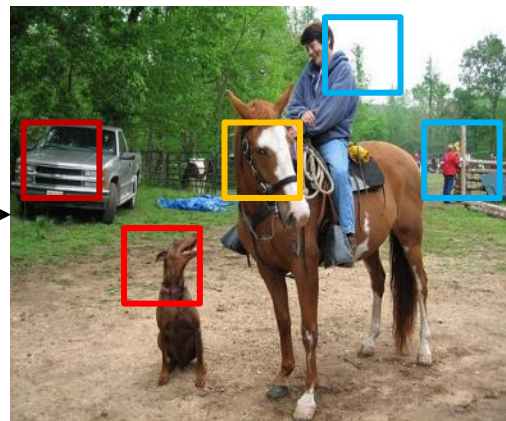
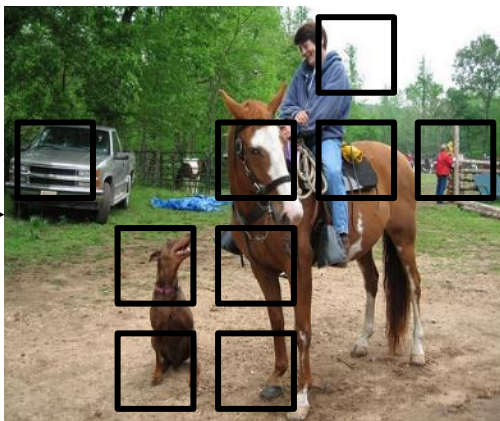
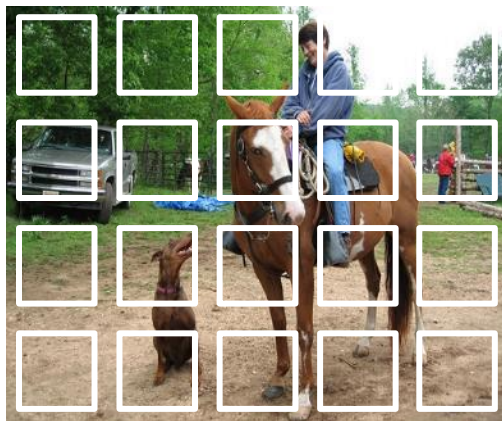


多分类  
(具体类别)

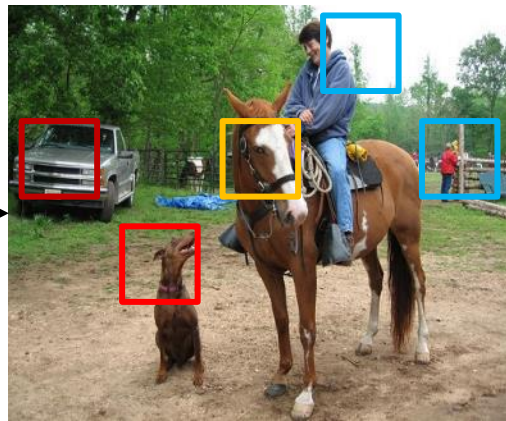


## 基于锚框的多阶段法 VS 单阶段法：二阶段分类

二  
阶  
段  
分  
类



一  
阶  
段  
分  
类

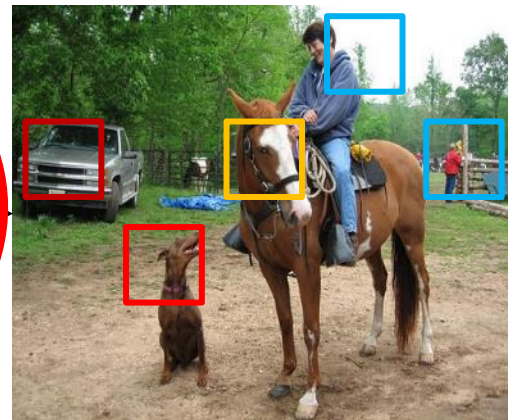
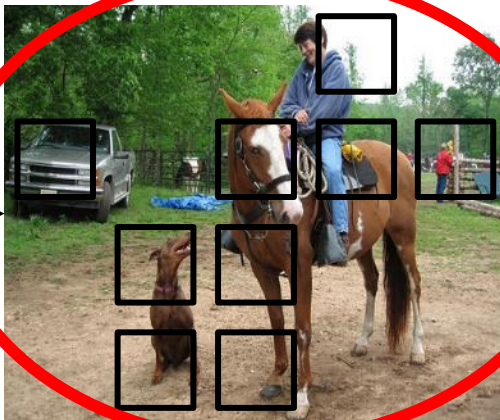
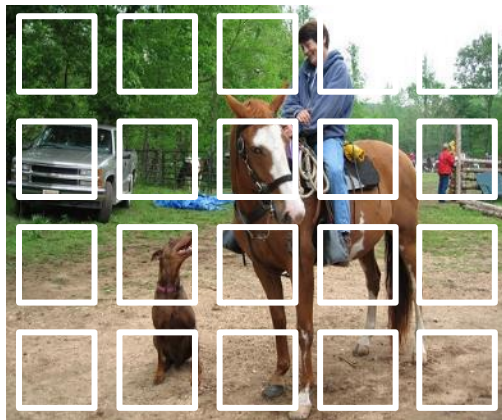




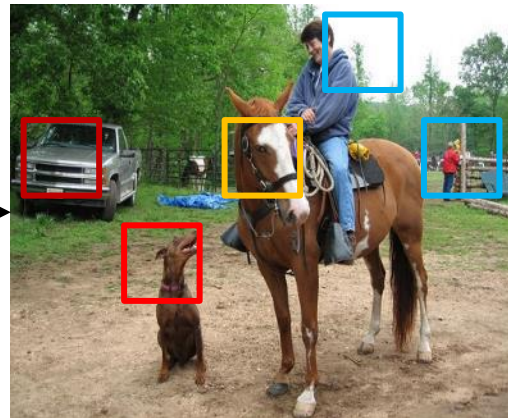
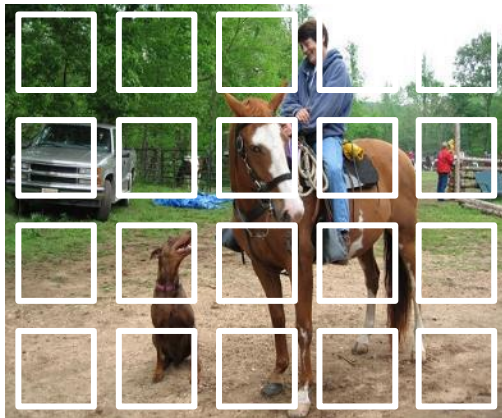


## 基于锚框的多阶段法 VS 单阶段法：二阶段分类

二  
阶  
段  
分  
类



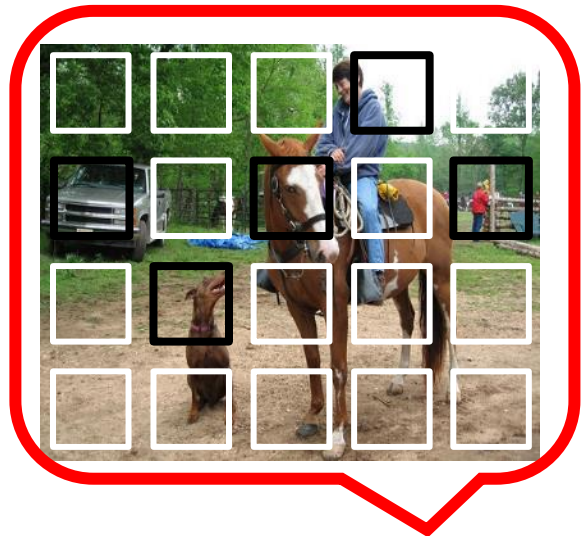
一  
阶  
段  
分  
类



作用？



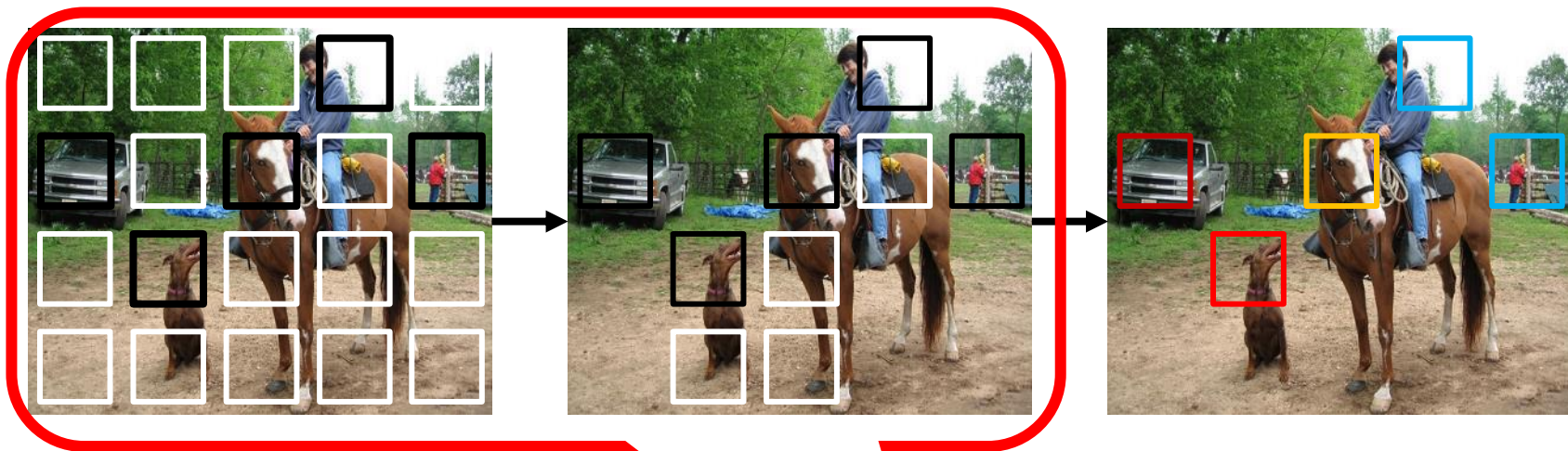
## 基于锚框的多阶段法 VS 单阶段法：二阶段分类



- 问题：正负样本极度失衡（图中黑色为正，白色为负，比例为5:15）
- 影响：负样本极多，难以训练，导致分类效果不佳
- 方案：二阶段分类可以缓解该问题



## 基于锚框的多阶段法 VS 单阶段法：二阶段分类



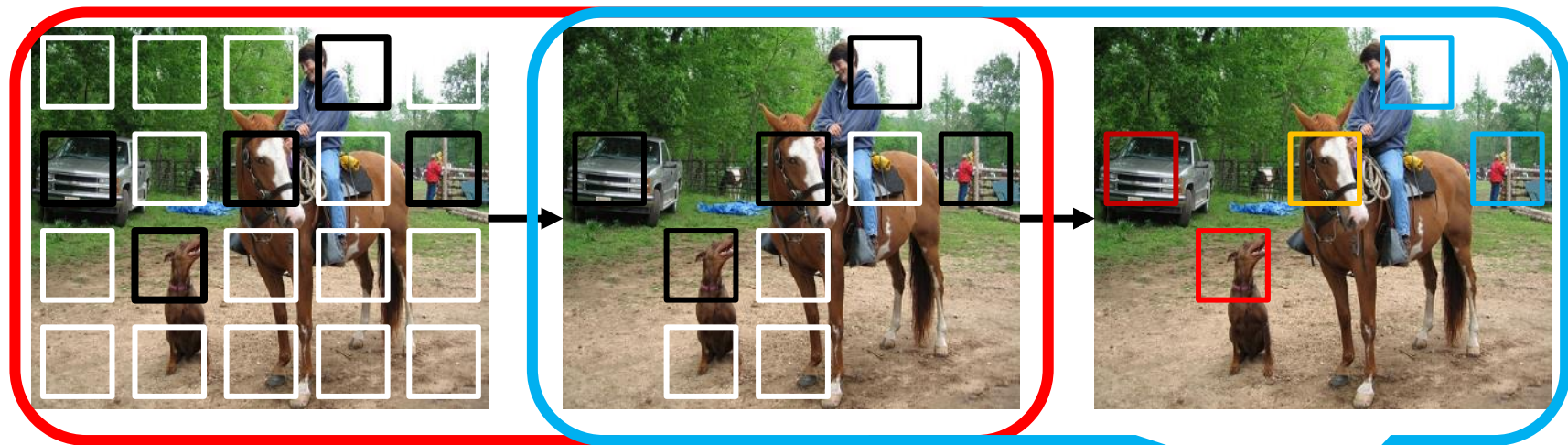
- 问题：正负样本极度失衡（图中黑色为正，白色为负，比例为5:15）
- 影响：负样本极多，难以训练，导致分类效果不佳
- 方案：二阶段分类可以缓解该问题

虽然正负样本比例为5:15  
但只做简单的二分类任务





## 基于锚框的多阶段法 VS 单阶段法：二阶段分类



- 问题：正负样本极度失衡（图中黑色为正，白色为负，比例为5:15）

- 影响：负样本极多，难以训练，导致分类效果不佳

- 方案：二阶段分类可以缓解该问题

经过第一阶段二分类的过滤之后

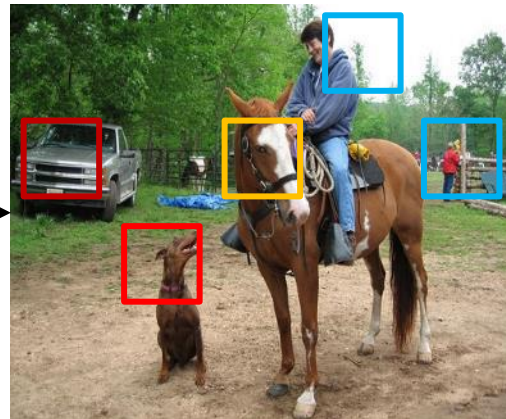
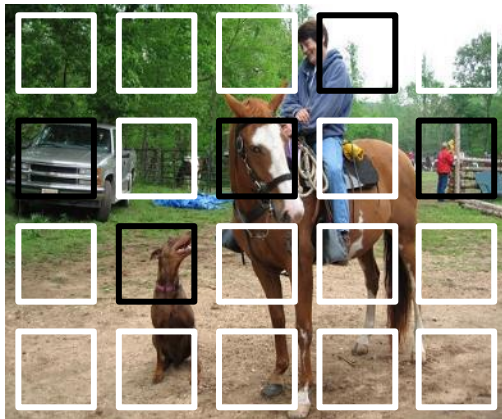
第二阶段多分类正负样本比例约1:1

虽然正负样本比例为5:15  
但只做简单的二分类任务



## 基于锚框的多阶段法 VS 单阶段法：二阶段分类

一  
阶  
段  
分  
类



不仅正负样本比例为5:15  
而且是复杂的多分类任务



## 基于锚框的多阶段法 VS 单阶段法：二阶段回归

### 多阶段法的二阶段回归过程



第一阶段  
的初步回归矫正



第二阶段  
的精准回归矫正



## 基于锚框的多阶段法 VS 单阶段法：二阶段回归

### 单阶段法的一阶段回归过程

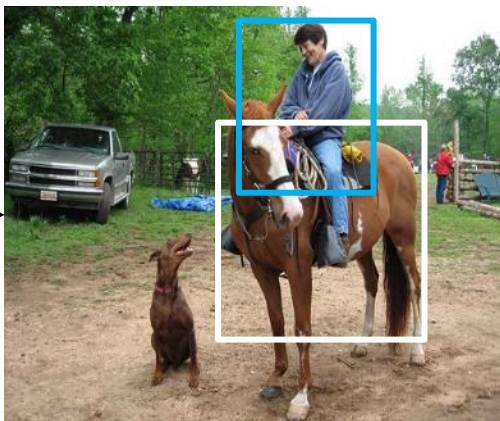


单阶段的回归矫正



## 基于锚框的多阶段法 VS 单阶段法：二阶段回归

二  
阶  
段  
回  
归



一  
阶  
段  
分  
类





## 基于锚框的多阶段法 VS 单阶段法：二阶段回归

### 二阶段回归

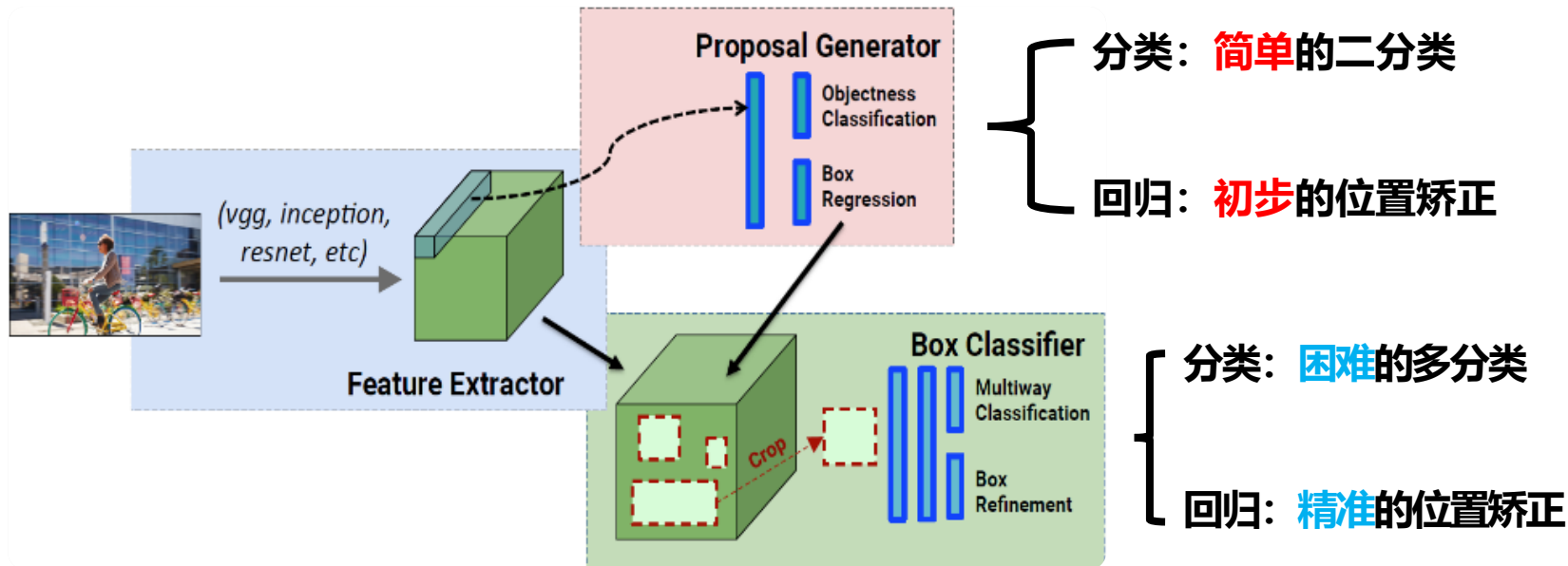


- ①经过两个阶段的回归矫正，使得检测结果的位置更加精准
- ②经过第一阶段的回归矫正，可以为第二阶段提供更多的正样本





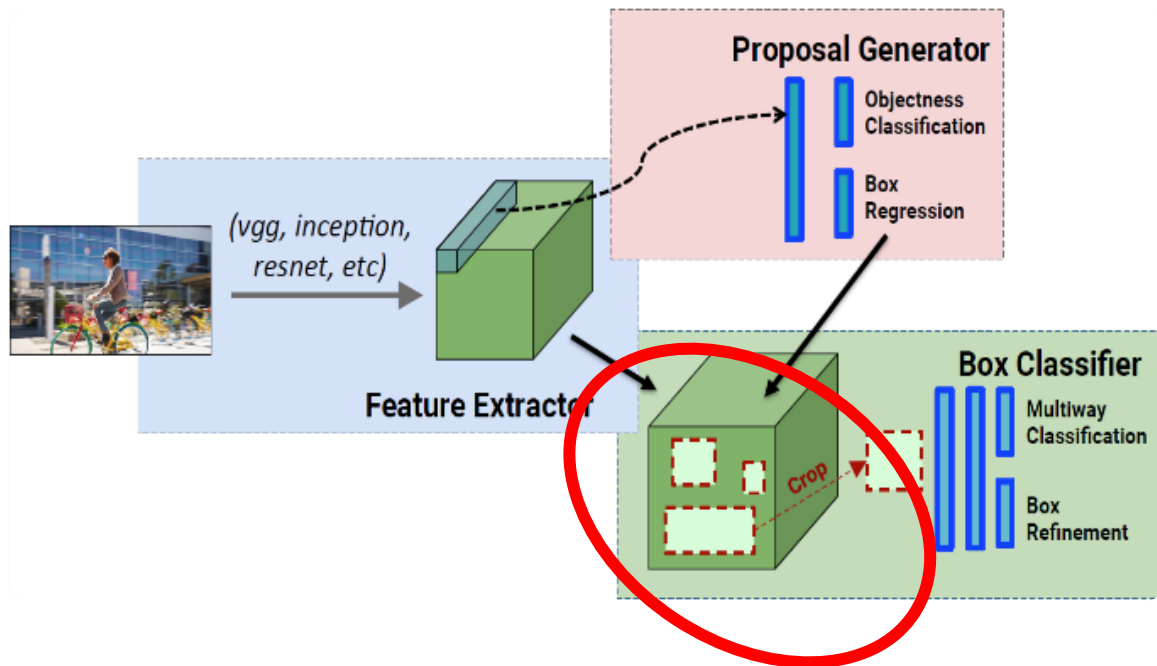
## 基于锚框的多阶段法 VS 单阶段法：二阶段特征



- 基础网络的特征是共享的
- 但两个阶段有着自己独有的特征来负责难度不同的任务



## 基于锚框的多阶段法 VS 单阶段法：特征的校准



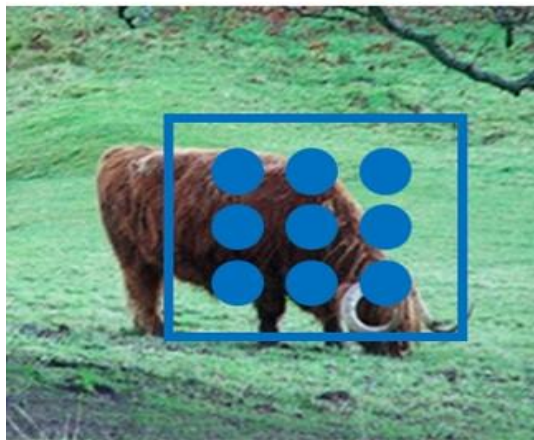
为什么  
要做特征校准？

- 多阶段法会使用一个RoI池化的操作，根据矩形框把特征扣出来进行校准

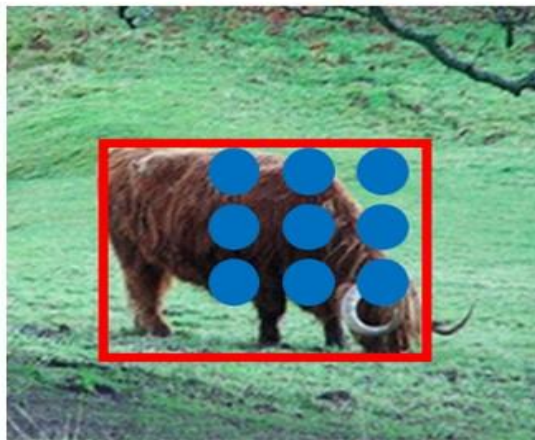




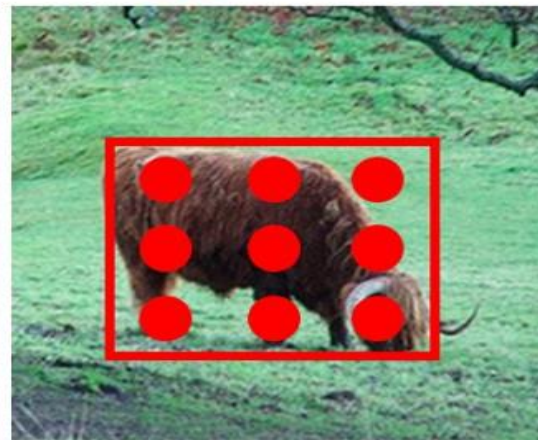
## 基于锚框的多阶段法 VS 单阶段法：特征的校准



(a)



(b)



(c)

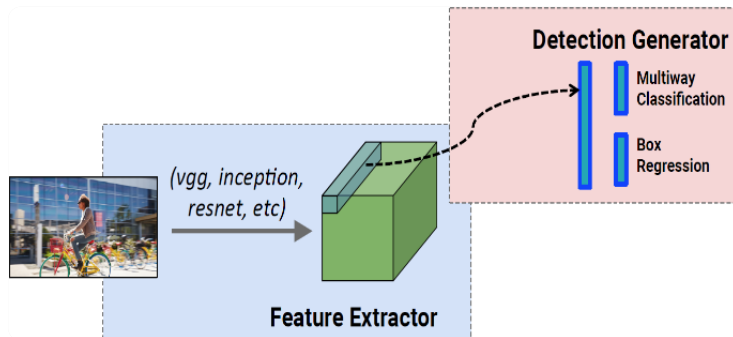
(a): 第一阶段中, 初始的锚框, 初始的特征; 图(b): 经过第一阶段校正后的锚框, 初始的特征; (c): 经过第一阶段校正后的锚框, 经过RoI池化校准后的特征

- 锚框回归后位置发生改变, 因此特征应根据锚框进行校准, 提高准确度
- 调整特征大小, 使其一致易于批处理



## 基于锚框的多阶段法 VS 单阶段法：总结

单阶段法



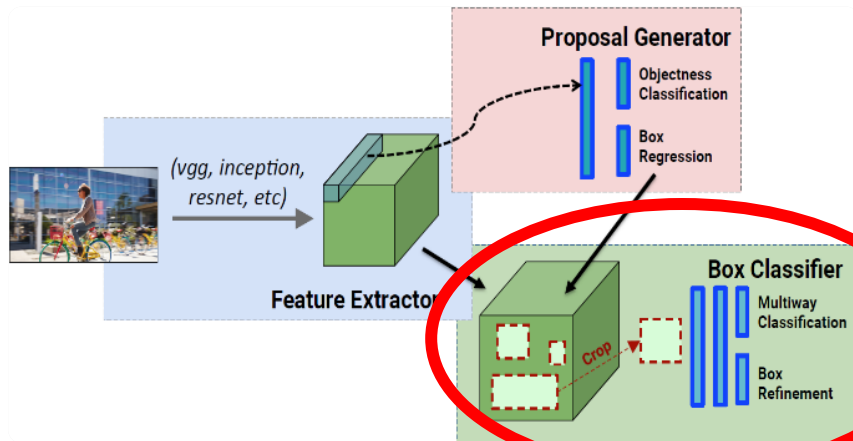
精度提高，但速度也变慢



① 二阶段的分类；② 二阶段的回归

③ 二阶段的特征；④ 特征的校准

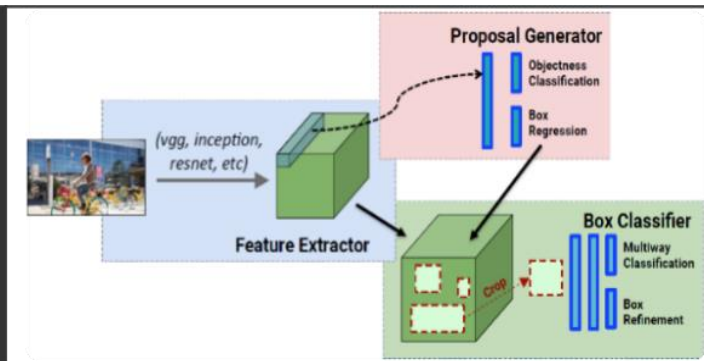
多阶段法



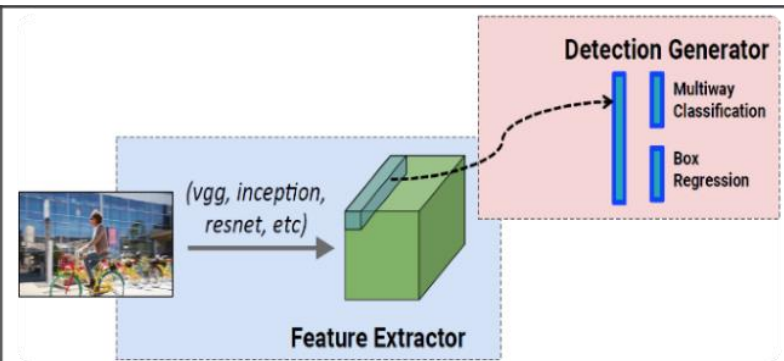


# 实用物体检测算法RefineDet

基于锚框



多阶段法

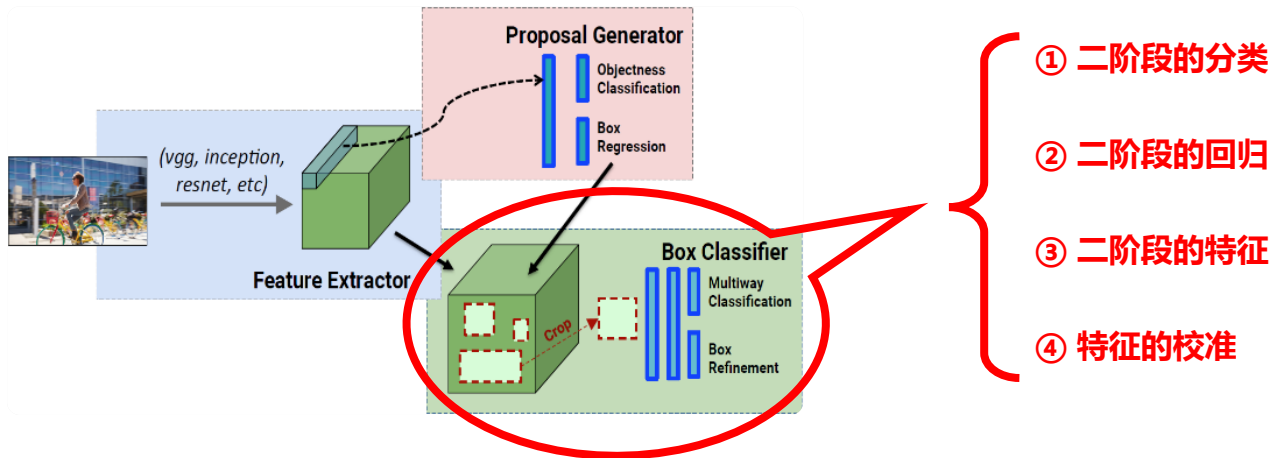


单阶段法

- 多阶段法：精度较高，速度较慢
- 单阶段法：精度较低，速度较快
- 找到两者的本质区别，取之长补己短的改进思路
- 提出RefineDet算法：单阶段法的速度+多阶段法的精度



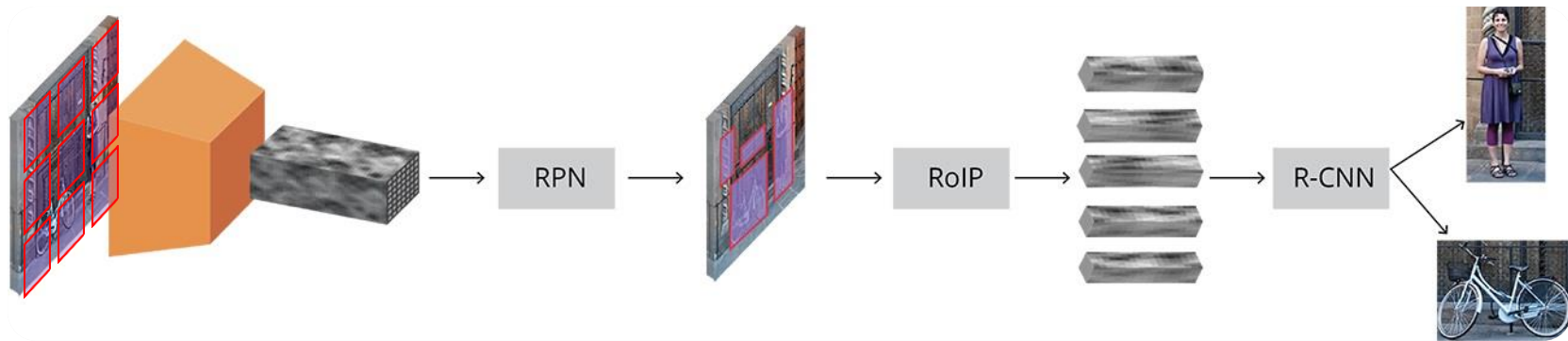
## 实用物体检测算法RefineDet: 改进思路



- 多阶段法精度较高的原因：第二阶段带来的4个方面改进，让精度得以提升
- 多阶段法速度较慢的原因：4个改进中有些改进比较耗时，导致速度变慢
- 前3个改进可以在单阶段法中高效实现：二阶段分类、回归、特征（高性价比）
- 最后1个改进，需要逐区域进行，导致速度变慢，且提升有限（低性价比）



## 实用物体检测算法RefineDet: 特征校准RoI池化



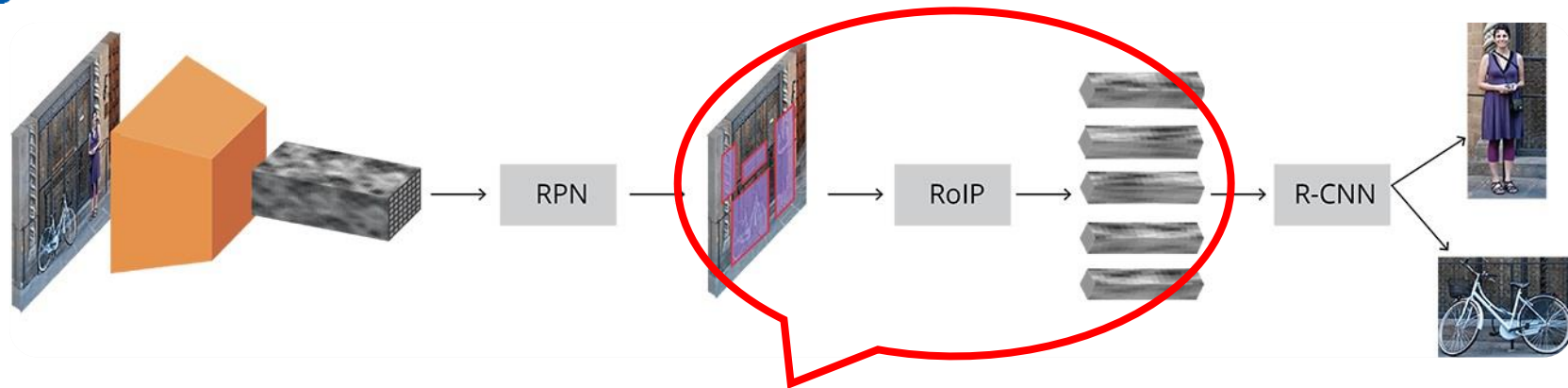
Faster R-CNN中RPN步骤:

- ① 整张图传入VGG16或ResNet提取特征
- ② 选择下采样倍数为16的特征层作为检测层
- ③ 根据检测层预设一系列大小和比例的锚框 (9个)
- ④ 对锚框进行二分类和回归得到若干候选区域

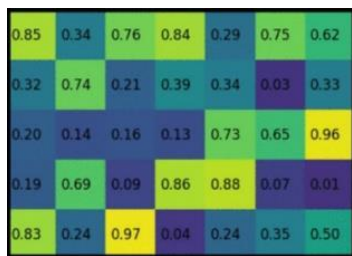
Faster R-CNN中Fast R-CNN步骤:

- ① **利用RoIPooling在检测层的特征上提取每个候选区域对应的特征**
- ② 输入CNN/FC子网络来增强候选区域的特征
- ③ 对候选区域进行多分类和回归得到检测结果

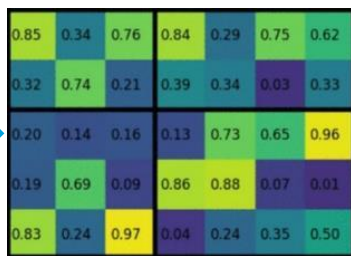
# 实用物体检测算法RefineDet：特征校准RoI池化



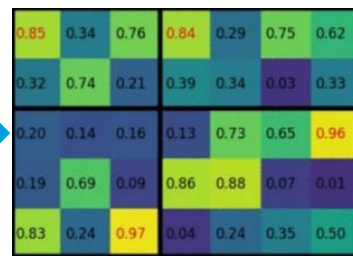
利用RoIPooling在检测层的特征上提取每个候选区域对应的特征，并转换为大小一致的特征



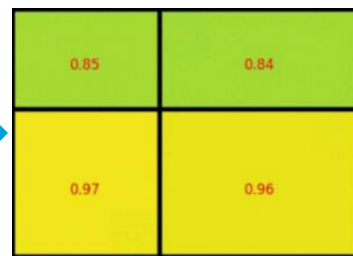
映射并取整



分块并取整



选取最大值

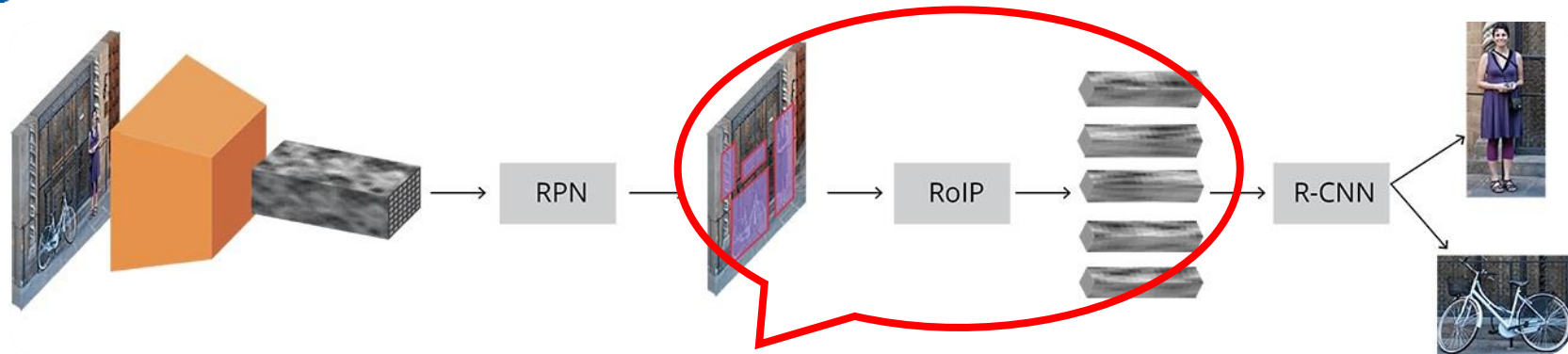


转换为2x2特征

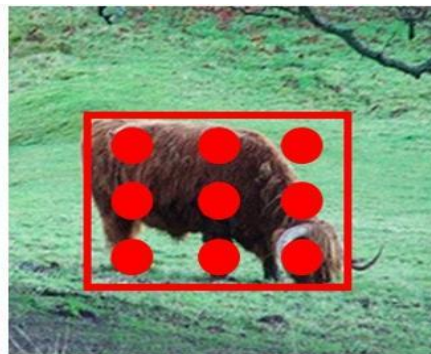
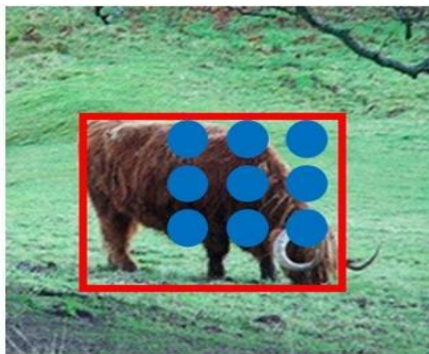
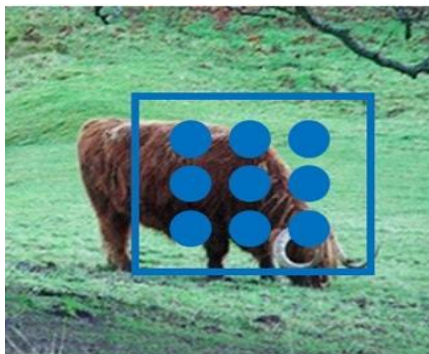




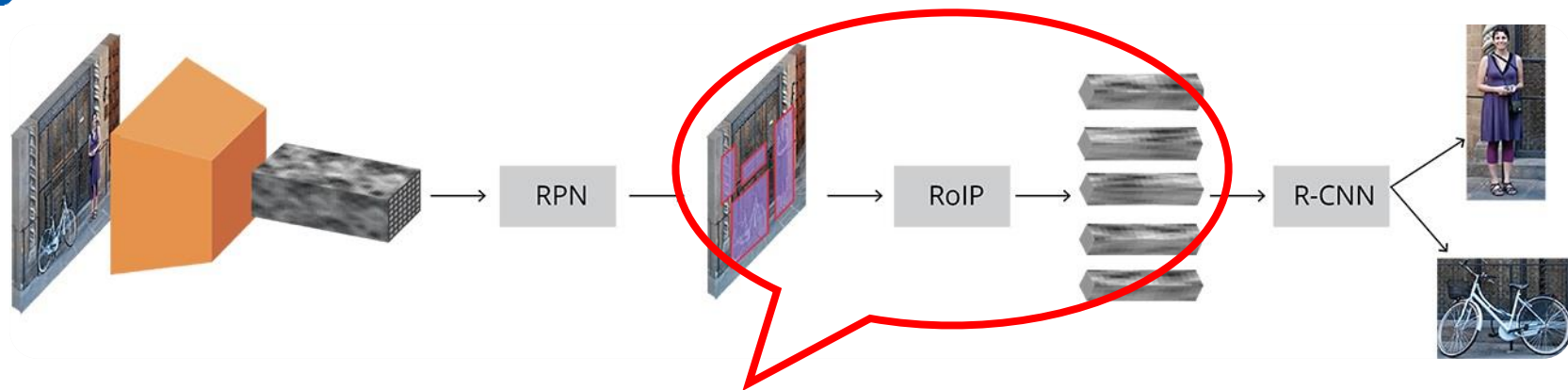
## 实用物体检测算法RefineDet: 特征校准RoI池化



- 目的①：特征大小一致后，不同候选区域可以组成一个批次，从而进行批处理
- 目的②：进过第一阶段的校准，锚框的位置发生了变化，因此需要进行特征校准



## 实用物体检测算法RefineDet: 特征校准RoI池化



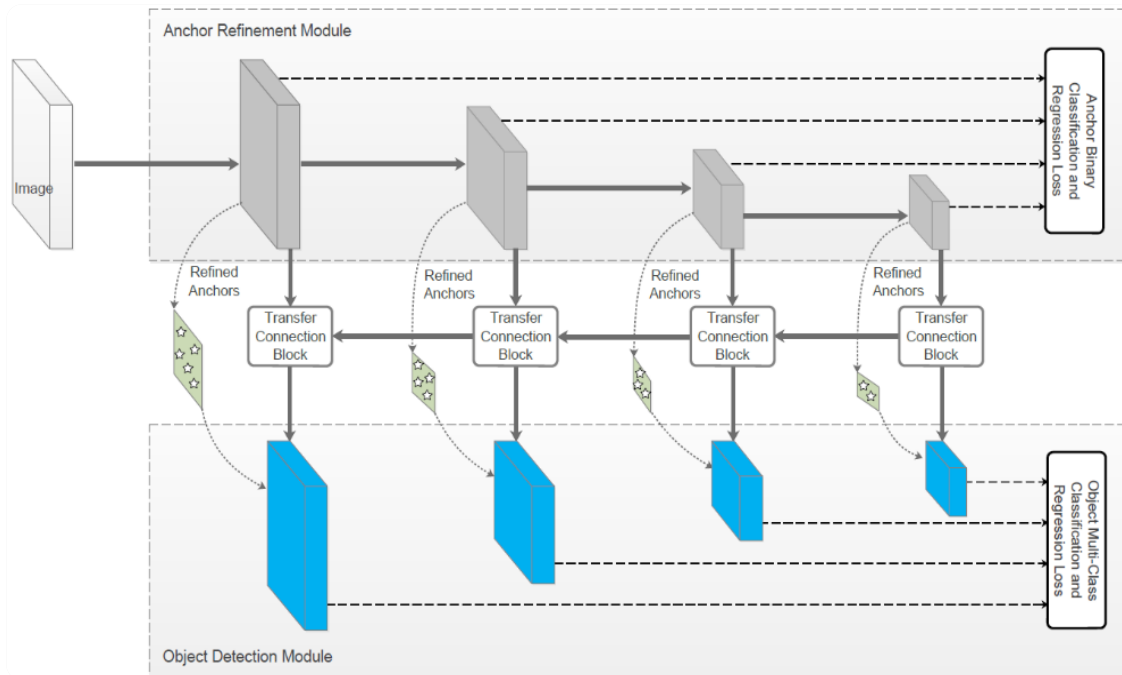
- 每个候选区域的大小不一样，位置不一样
- 每个候选区域需要单独处理，不能并行加速
- RoI池化以及其改进版，涉及到取整或插值等操作
- 因此，特征校准的RoI池化比较慢，尤其是候选区域较多时





## 实用物体检测算法RefineDet：整体框架

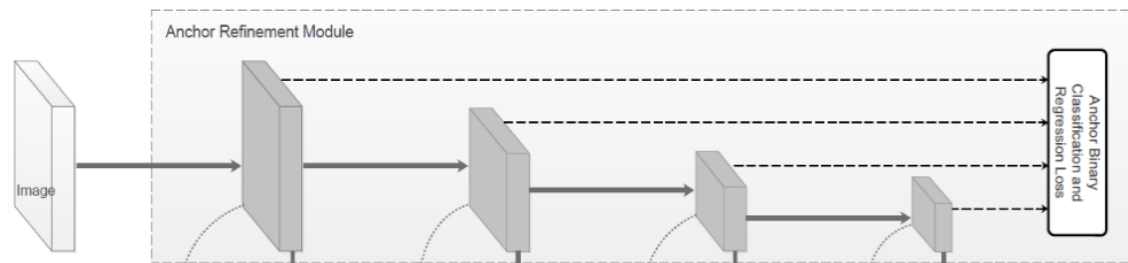
- 在单阶段法中，集成多阶段法中高性价比的改进，抛弃低性价比的改进
- 从而在保持单阶段法的速度的同时，获得多阶段法的精度



- 锚框校准模块 (ARM)
  - 过滤负样本
  - 初步的边框校正
- 传输连接模块 (TCB)
  - 转换ARM特征
  - 融合高层特征
- 物体检测模块 (ODM)
  - 更好的特征
  - 精细的分类和回归

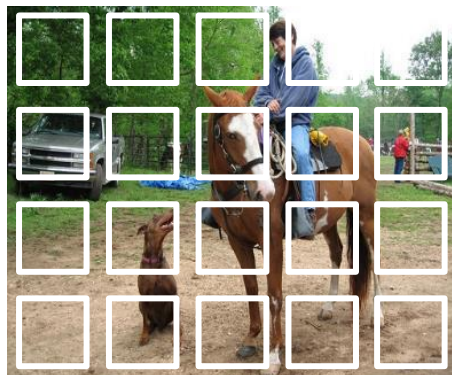


## 实用物体检测算法RefineDet: 锚框校准模块

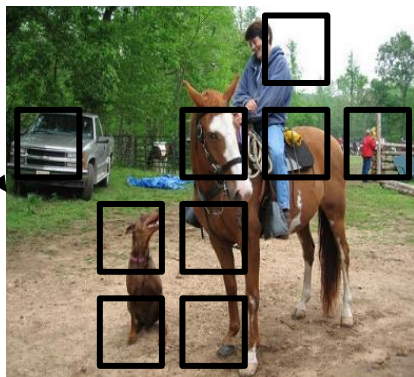


### ➤ 锚框校准模块 (ARM)

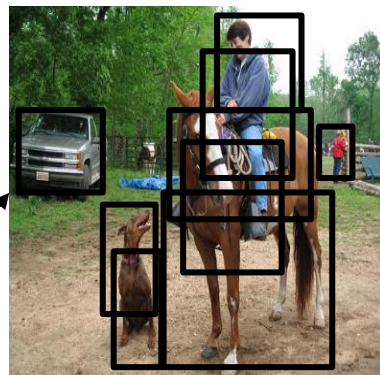
- 过滤负样本
- 初步的边框校正



二分类

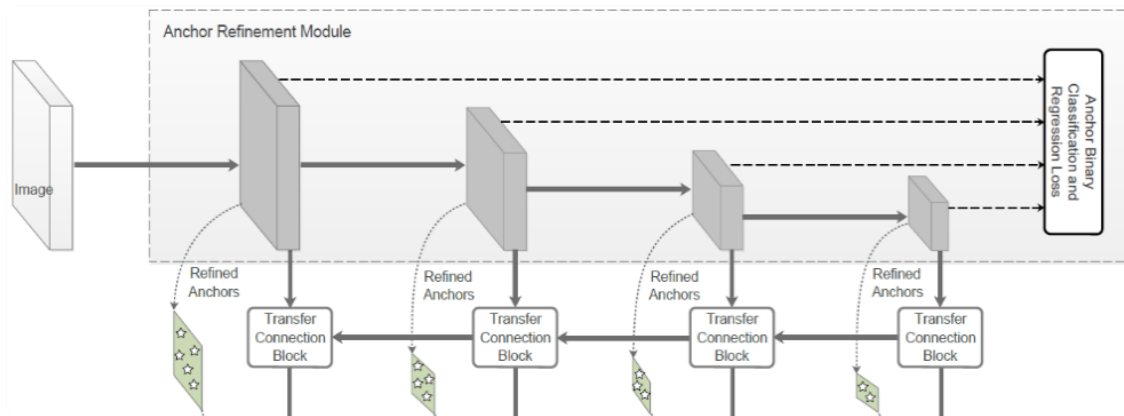


回归





## 实用物体检测算法RefineDet：传输连接模块

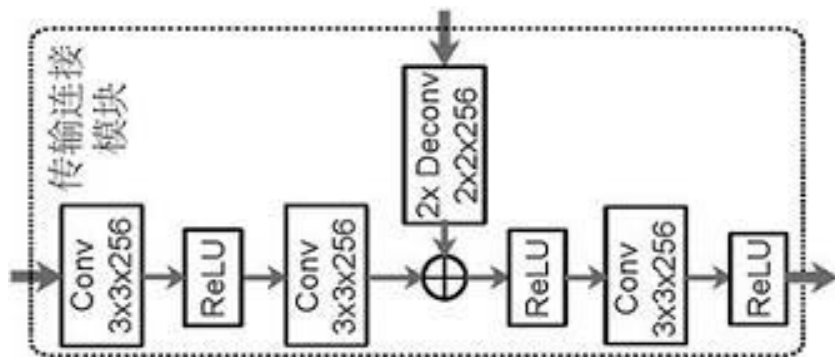


### ➤ 锚框校准模块 (ARM)

- 过滤负样本
- 初步的边框校正

### ➤ 传输连接模块 (TCB)

- 转换ARM特征
- 融合高层特征



■ 传输连接模块≈FPN

■ 当时FPN没有公布代码

■ 实现的TCB有些细节不一样

■ 使用ReLU, DeConv等



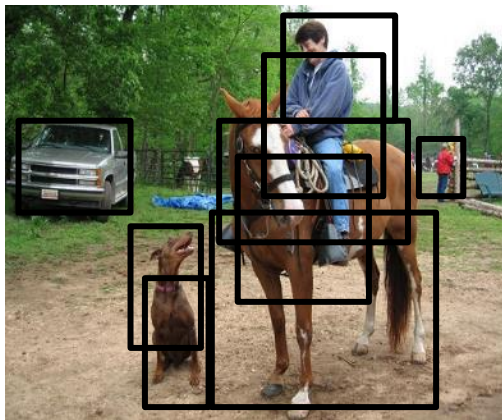
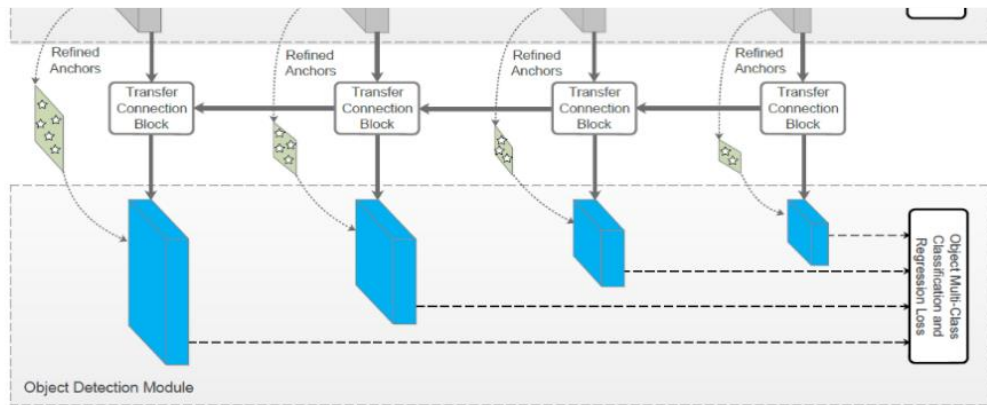
# 实用物体检测算法RefineDet: 物体检测模块

## ➤ 传输连接模块 (TCB)

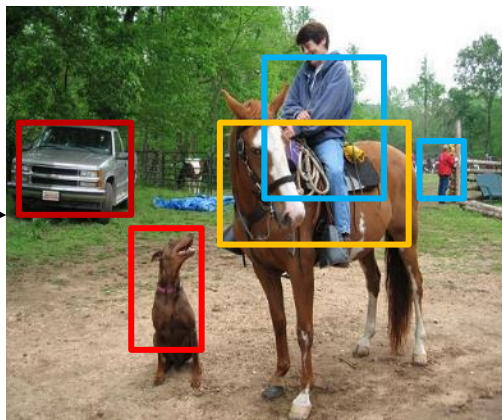
- 转换ARM特征
- 融合高层特征

## ➤ 物体检测模块 (ODM)

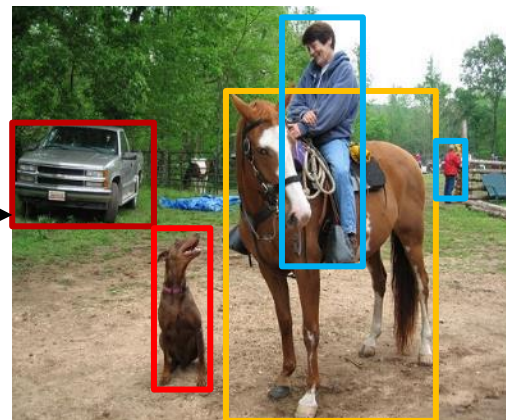
- 更好的特征
- 精细的分类和回归



分类

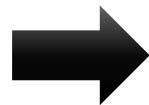
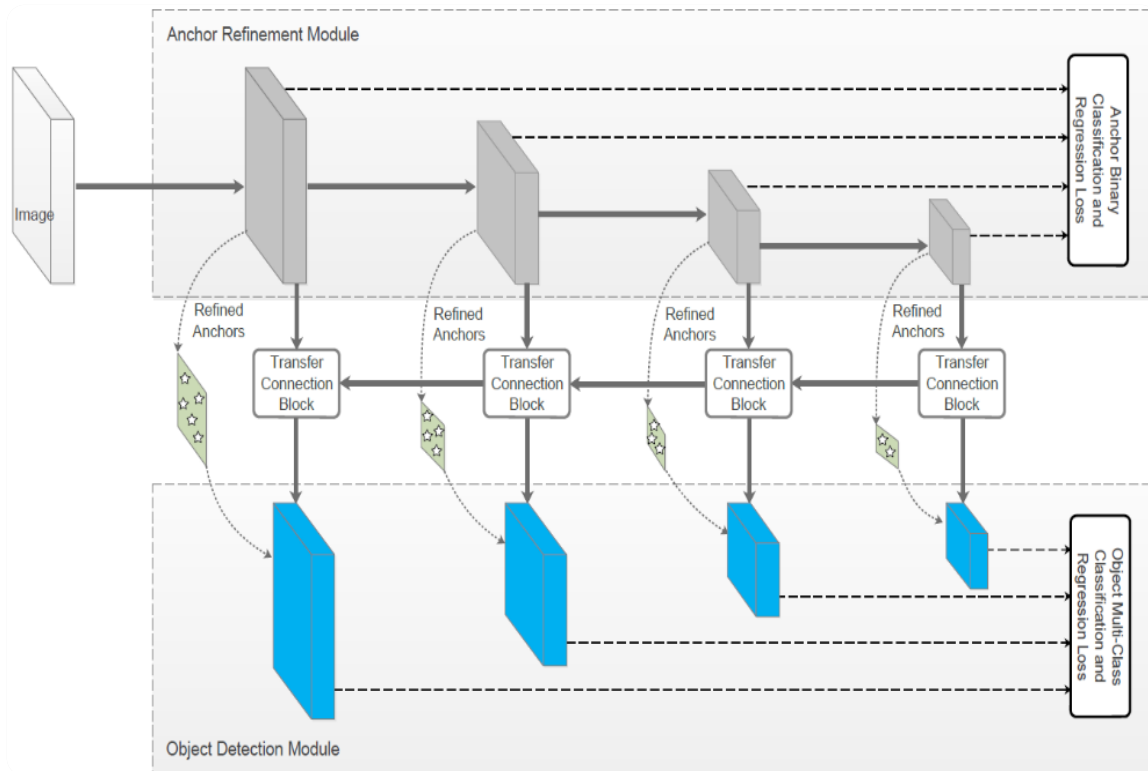


回归

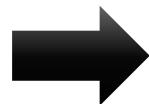




## 实用物体检测算法RefineDet: 二阶段分类



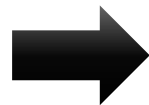
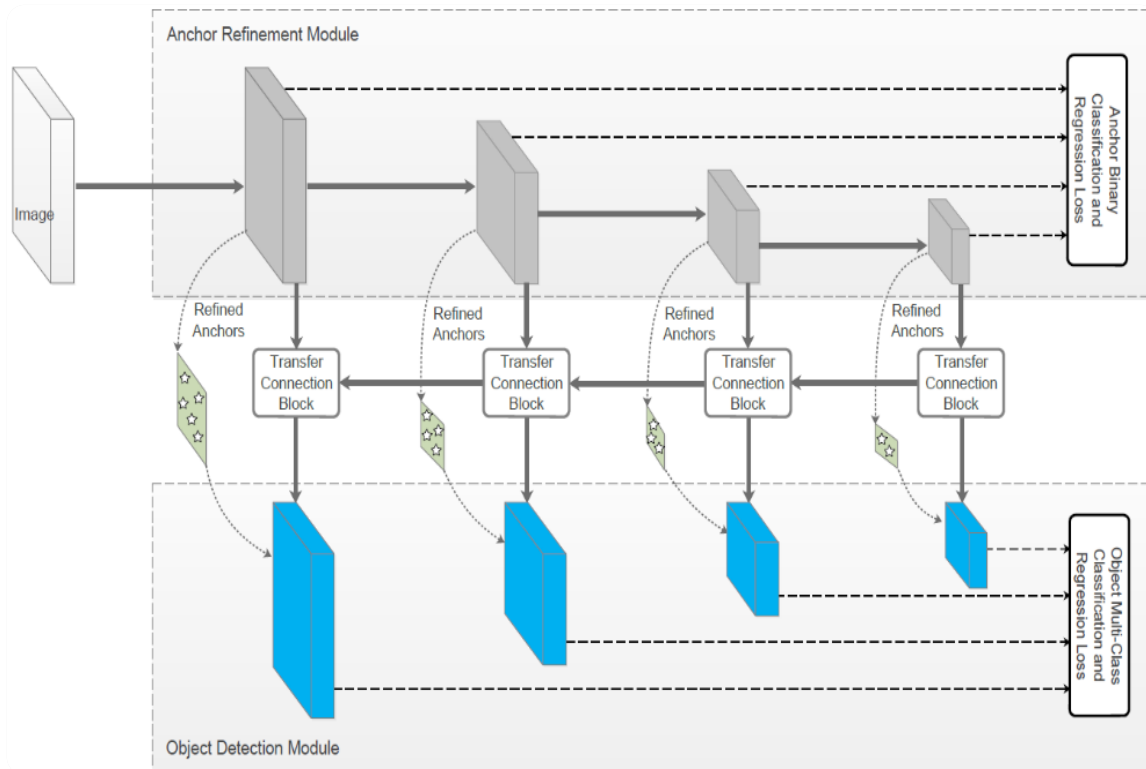
第一阶段分类  
(二分类)



第二阶段分类  
(多分类)



## 实用物体检测算法RefineDet: 二阶段回归



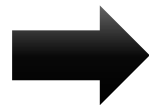
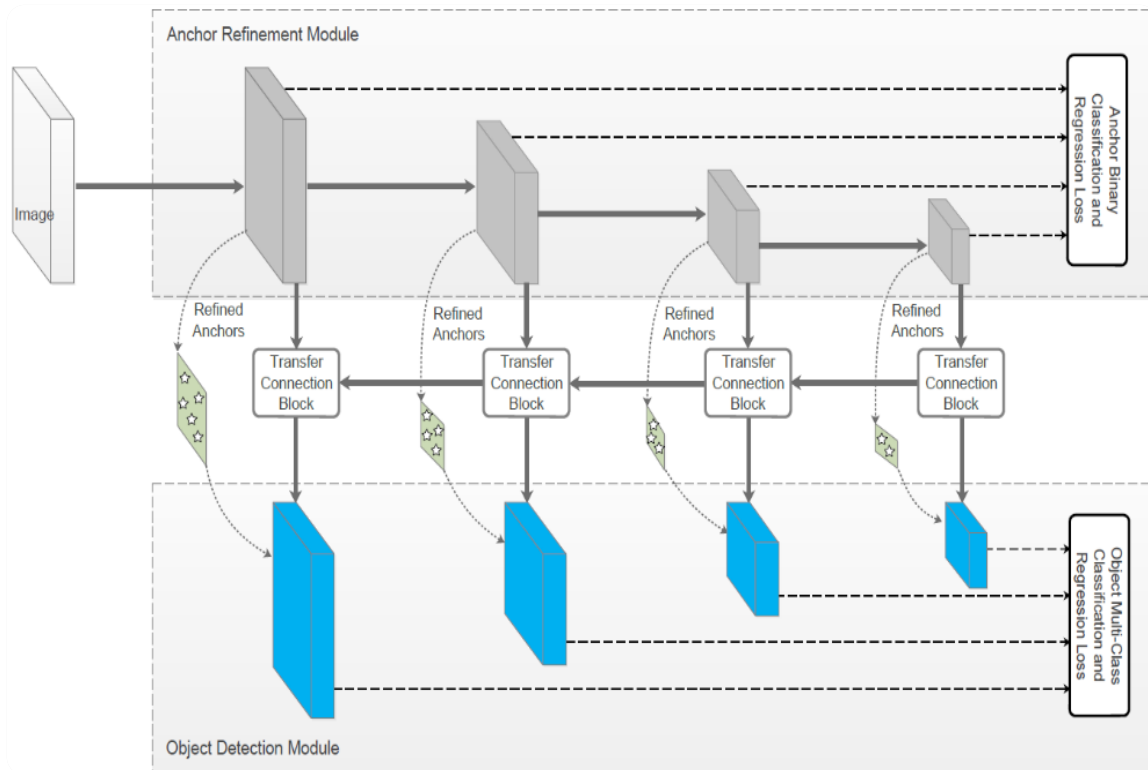
**第一阶段回归**  
(初始矫正)



**第二阶段回归**  
(精准矫正)



## 实用物体检测算法RefineDet: 二阶段特征



第一阶段特征  
(基础网络)

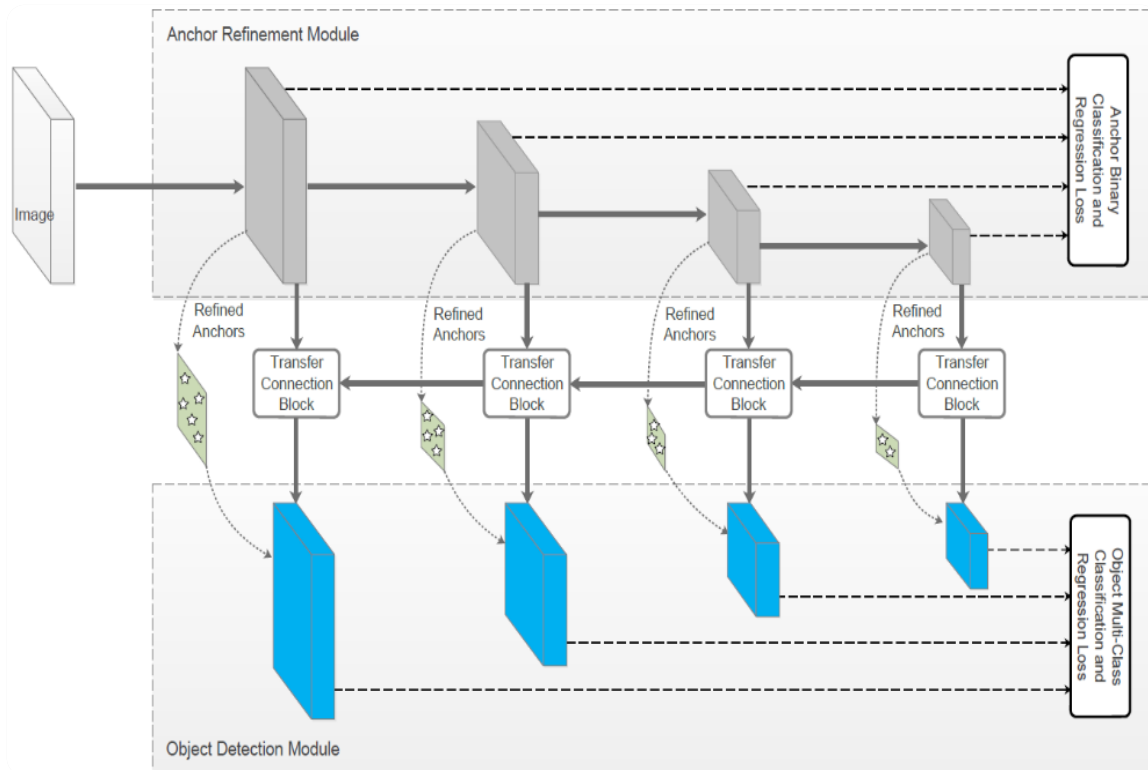


第二阶段特征  
(FPN强化)





## 实用物体检测算法RefineDet：二阶段特征



全卷积网络实现

少量额外计算



## 实用物体检测算法RefineDet：速度精度

System	VOC2007 test <i>mAP</i>	FPS (Titan X)	Number of Boxes	Input resolution
<a href="#">Faster R-CNN (VGG16)</a>	73.2	7	~6000	~1000 x 600
<a href="#">YOLO (GoogleNe)</a>	63.4	45	98	448 x 448
<a href="#">YOLOv2 (Darknet-19)</a>	78.6	40	845	544 x 544
<a href="#">SSD300* (VGG16)</a>	77.2	46	8732	300 x 300
<a href="#">SSD512* (VGG16)</a>	79.8	19	24564	512 x 512
<a href="#">RefineDet320 (VGG16)</a>	80.0	40	6375	320 x 320
<a href="#">RefineDet512 (VGG16)</a>	<b>81.8</b>	24	16320	512 x 512

- 单阶段法的速度
- 多阶段法的精度



## 实用物体检测算法RefineDet：速度精度

System	VOC2007 test <i>mAP</i>	FPS (Titan X)	Number of Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	~6000	~1000 x 600
YOLO (GoogleNe)	63.4	45	98	448 x 448
YOLOv2 (Darknet-19)	78.6	40	845	544 x 544
SSD300* (VGG16)	77.2	46	8732	300 x 300
SSD512* (VGG16)	79.8	19	24564	512 x 512
RefineDet320 (VGG16)	80.0	40	6375	320 x 320
RefineDet512 (VGG16)	81.8	24	16320	512 x 512

- 相对于baseline的SSD，不仅精度提高2个点，速度变快5 FPS
- 精度提高得益于二阶段分类、二阶段回归、二阶段特征
- 速度变快是因为①用了更少的检测层 (6->4)；②用了更少锚框(25K->16K)



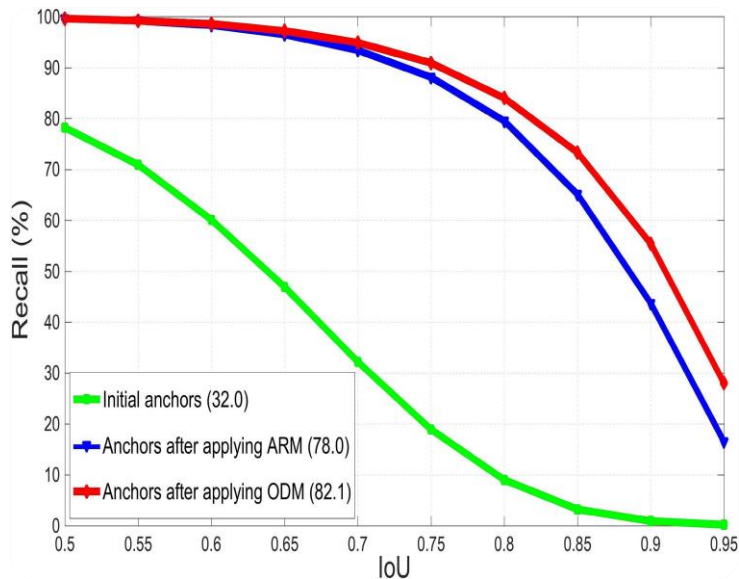
## 实用物体检测算法RefineDet: 有效性验证

Component	RefineDet320			
negative anchor filtering?	✓			
two-step cascaded regression?	✓	✓		
transfer connection block?	✓	✓	✓	
mAP (%)	80.0	79.5	77.3	76.2

■ 二阶段分类 **+0.5**

■ 二阶段回归 **+2.2**

■ 二阶段特征 **+1.1**





# 实用物体检测算法RefineDet: 代码开源

sfzhang15 / RefineDet

Watch 65

★ Unstar 1.4k

🔗 Fork 397

<> Code

🔔 Issues 1

🔗 Pull requests 0

🎮 Actions

📁 Projects 0

📖 Wiki

🛡 Security

📊 Insights

⚙ Settings

Single-Shot Refinement Neural Network for Object Detection, CVPR, 2018

Edit

object-detection Manage topics

🔗 119 commits

🌿 1 branch

📦 0 packages

🔖 0 releases

👤 3 contributors

📄 View license

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾



sfzhang15 Merge pull request #166 from lengly/patch-1 ...

Latest commit 52b6fe2 on 18 Mar 2019

📁 cmake

Adding the code of RefineDet

2 years ago

📁 data

Update README.md

2 years ago

📁 docker

fix dockerfile

15 months ago

📁 docs

Adding the code of RefineDet

2 years ago

<https://github.com/sfzhang15/RefineDet>



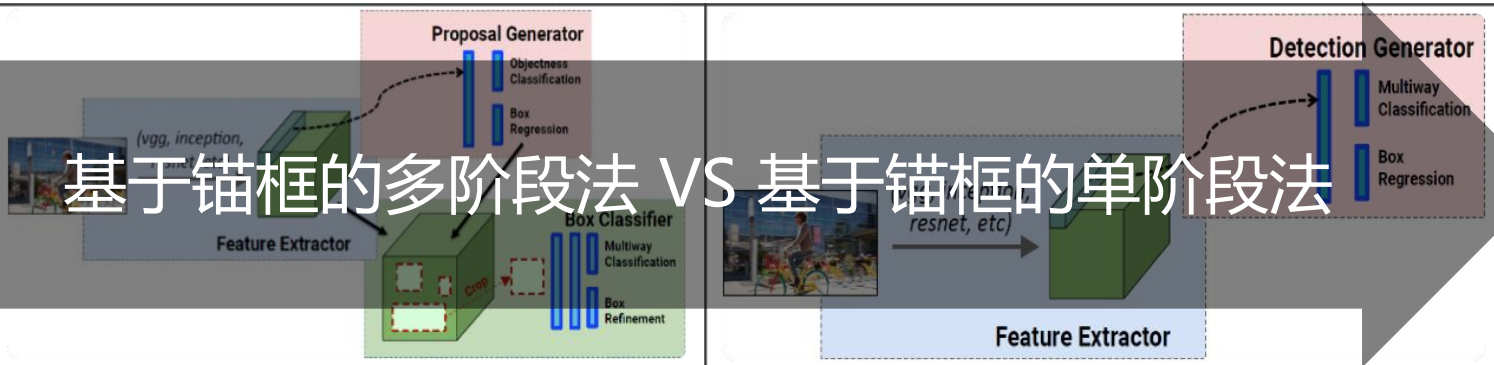
## 实用物体检测算法RefineDet

基于锚框

基于锚框的多阶段法 VS 基于锚框的单阶段法

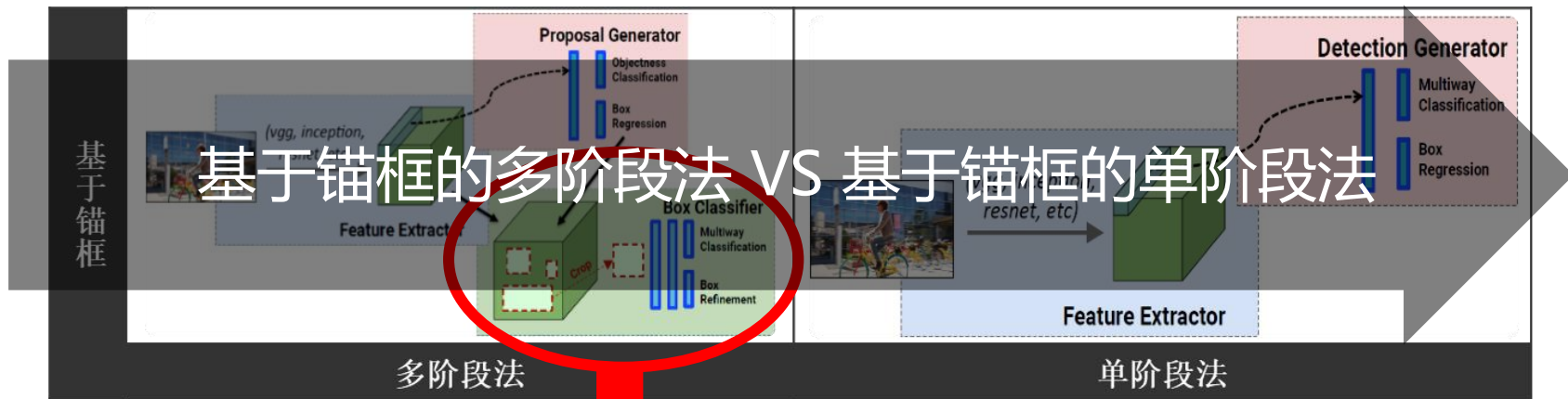
多阶段法

单阶段法





## 实用物体检测算法RefineDet



① 二阶段的分类

② 二阶段的回归

③ 二阶段的特征

④ 特征的校准

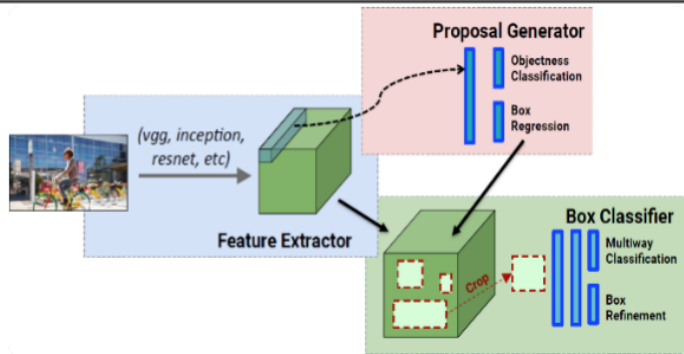
实用物体检测算法  
RefineDet





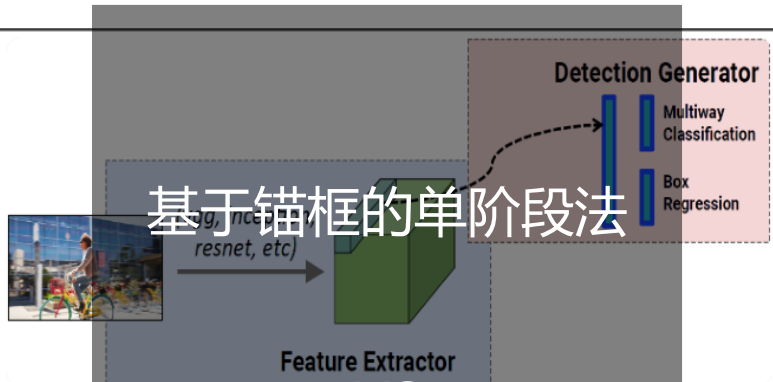
# 实用检测算法的研究思路：对比探索

基于锚框



多阶段法

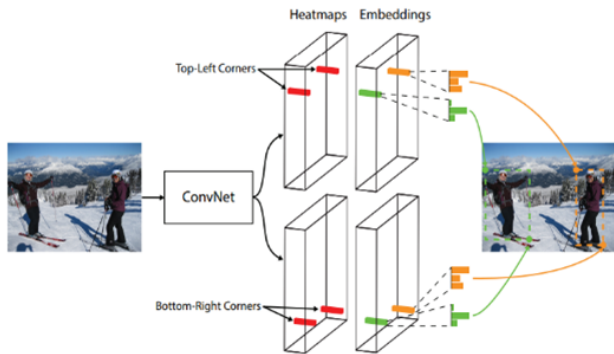
基于锚框的单阶段法



单阶段法

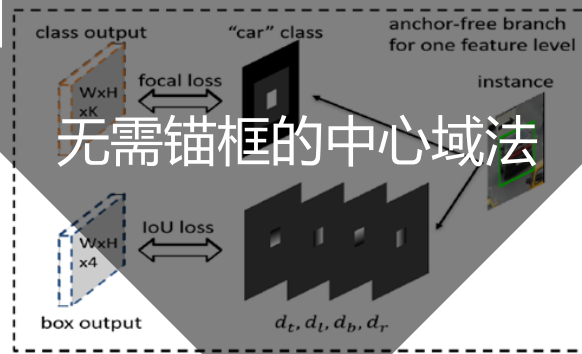
VS

无需锚框



关键点法

无需锚框的中心域法



中心域法



## 基于锚框的单阶段法 VS 无需锚框的中心域法

检测流程非常相似

基于锚框的单阶段法



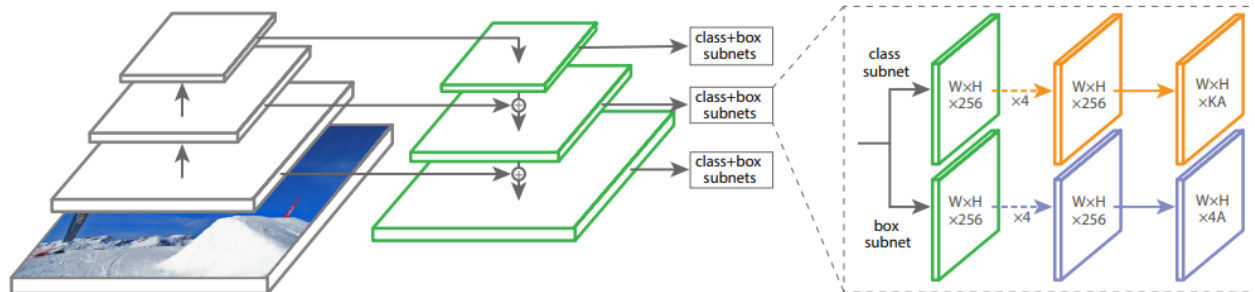
无需锚框的中心域法



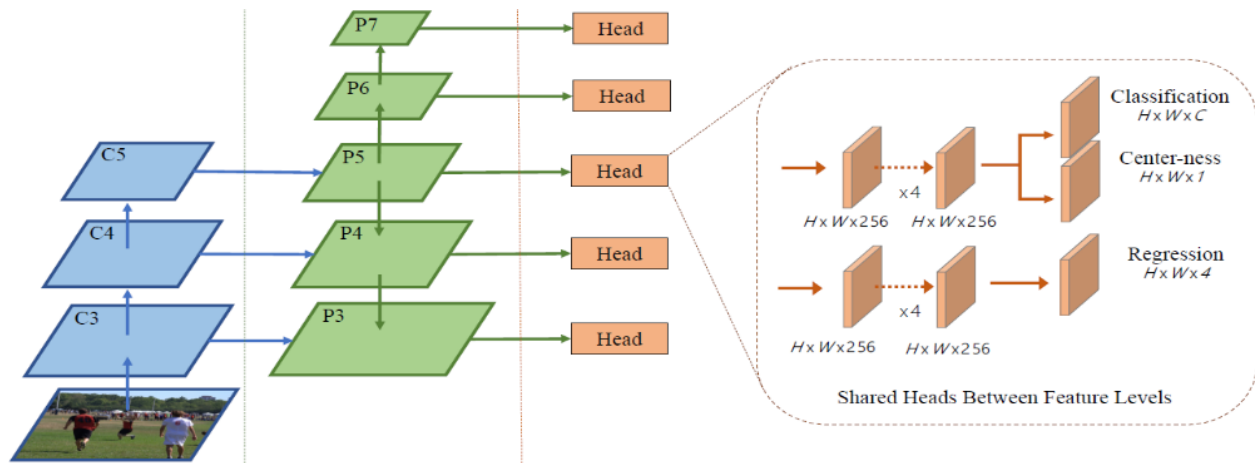


# RetinaNet 与 FCOS 的相同点

基于锚框的单阶段法  
RetinaNet



无需锚框的中心域法  
FCOS



■ 相同的基础网络

■ 特征金字塔

■ 分类子网络

■ 回归子网络

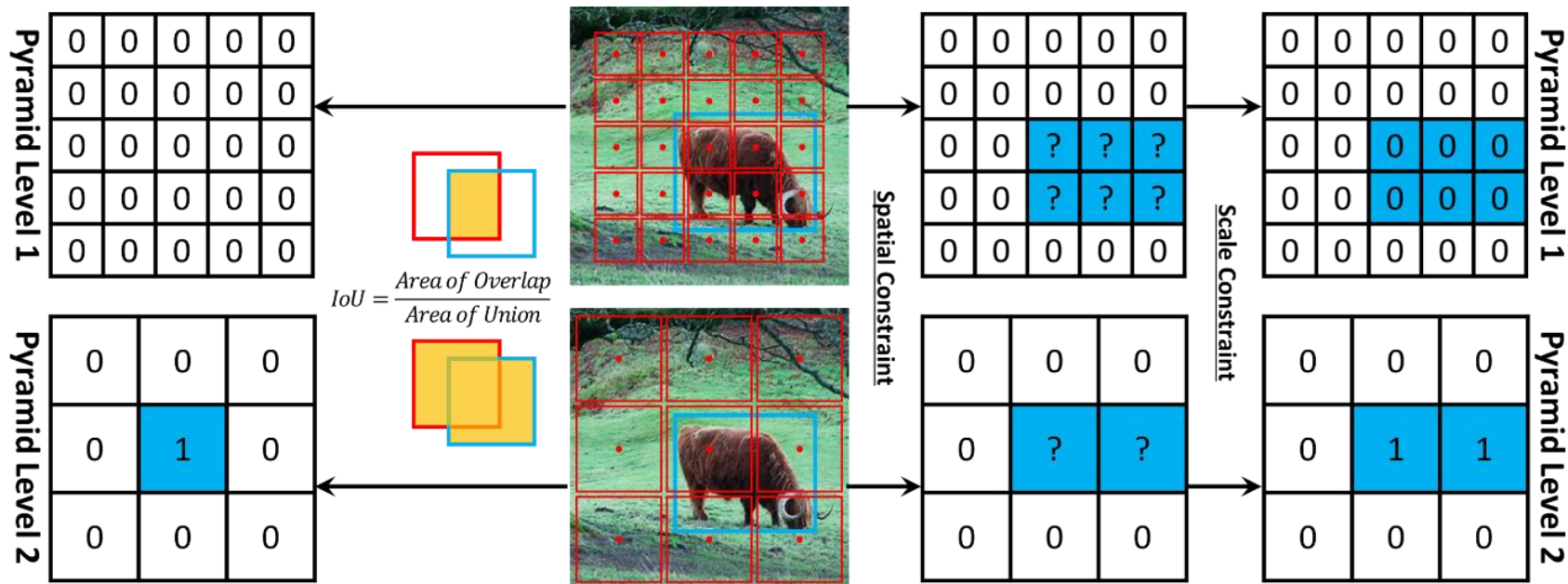


## RetinaNet和FCOS的不同点



## RetinaNet和FCOS的不同点：①正负样本定义

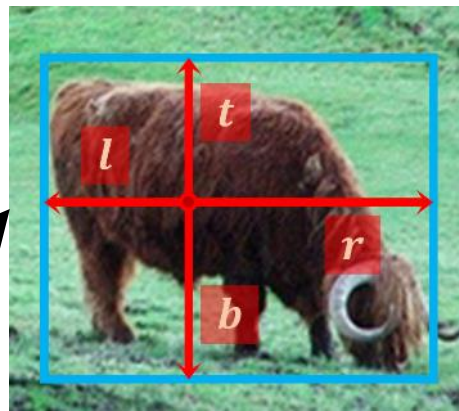
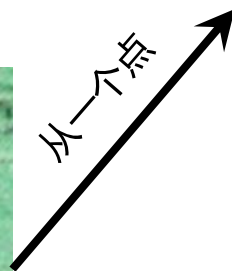
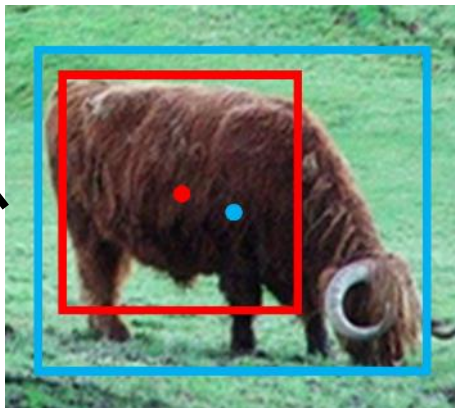
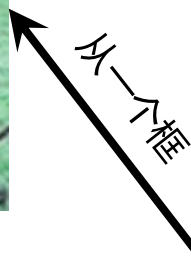
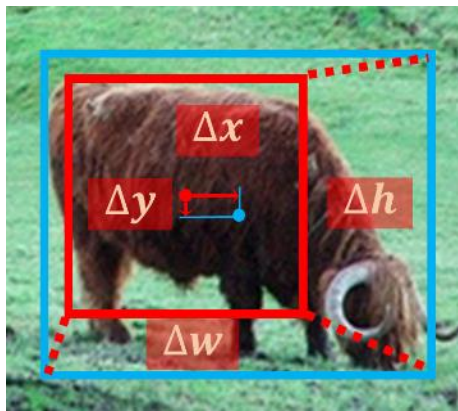
- RetinaNet: 利用IoU选取正负训练样本
- FCOS: 利用空间和尺度上的限制来选取正负训练样本





## RetinaNet和FCOS的不同点：②回归起点

- RetinaNet: 从一个矩形框（锚框）开始回归物体
- FCOS: 从一个点（锚点）开始回归物体

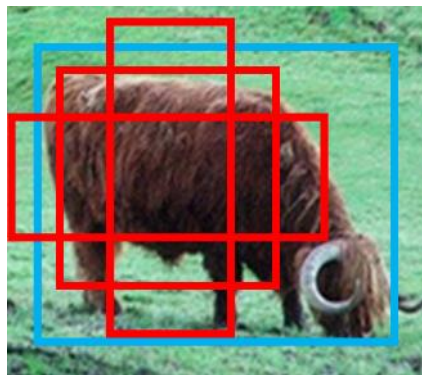




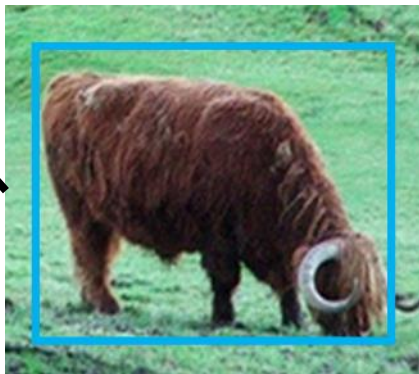


## RetinaNet和FCOS的不同点：③每个位置铺设的样本个数

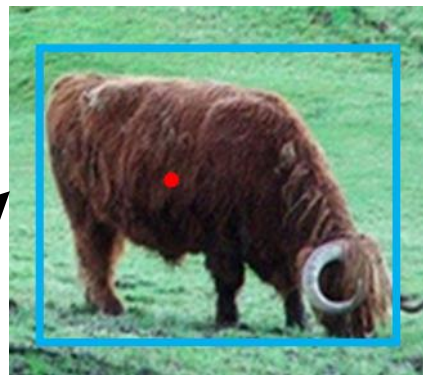
- RetinaNet: 在每个位置铺设了9个锚框 (3个比例和3个尺度)
- FCOS: 在每个位置只铺设了一个锚点



每个位置9个锚框



每个位置1个锚点







## RetinaNet和FCOS的不同点

- ① 正负训练样本定义的不同
- ② 回归的起点不同
- ③ 每个位置铺设的样本个数不同

**哪一个本质区别？**

**让FCOS在速度稍微变快的同时，精度得到提高**

**找到这个高效的本质区别，并加以改进提出ATSS**



## 实用物体检测算法ATSS：排除实现不一致

- ① 正负训练样本定义的不同
- ② 回归的起点不同

Inconsistency	FCOS	RetinaNet (#A=1)					
GroupNorm	✓	✓	✓	✓	✓	✓	✓
GIoU Loss	✓		✓	✓	✓	✓	✓
In GT Box	✓			✓	✓	✓	✓
Centerness	✓				✓	✓	✓
Scalar	✓						✓
AP (%)	37.8	32.5	33.4	34.9	35.3	36.8	37.0



## 实用物体检测算法ATSS：排除实现不一致

- ① 正负训练样本定义的不同
- ② 回归的起点不同

Inconsistency	FCOS	RetinaNet (#A=1)					
GroupNorm	✓	✓	✓	✓	✓	✓	
GIoU Loss	✓		✓	✓	✓	✓	
In GT Box	✓			✓	✓	✓	
Centerness	✓				✓	✓	
Scalar	✓						✓
AP (%)	37.8	32.5	33.4	34.9	35.3	36.8	37.0



## 实用物体检测算法ATSS：排除实现不一致

- ① 正负训练样本定义的不同
- ② 回归的起点不同

Inconsistency	FCOS	RetinaNet (#A=1)					
GroupNorm	✓	✓	✓	✓	✓	✓	
GIoU Loss	✓		✓	✓	✓	✓	✓
In GT Box	✓			✓	✓	✓	✓
Centerness	✓				✓	✓	✓
Scalar	✓						✓
AP (%)	37.8	32.5	33.4	34.9	35.3	36.8	37.0



## 实用物体检测算法ATSS：排除实现不一致

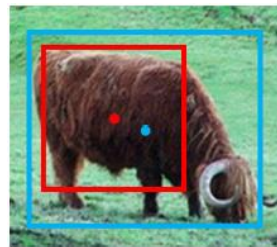
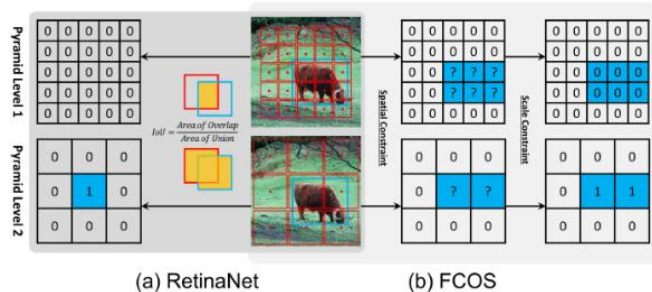
- ① 正负训练样本定义的不同
- ② 回归的起点不同

Inconsistency	FCOS	RetinaNet (#A=1)					
GroupNorm	✓	✓	✓	✓	✓	✓	
GIoU Loss	✓		✓	✓	✓	✓	✓
In GT Box	✓			✓	✓	✓	✓
Centerness	✓				✓	✓	✓
Scalar	✓						✓
AP (%)	37.8	32.5	33.4	34.9	35.3	36.8	37.0

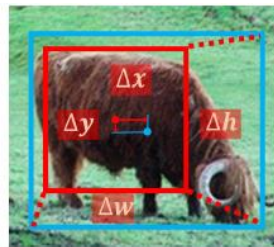


## 实用物体检测算法ATSS：探索本质区别

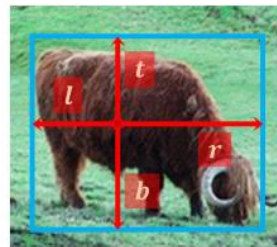
- ① 正负训练样本定义的不同
- ② 回归的起点不同



(a) Positive sample



(b) RetinaNet



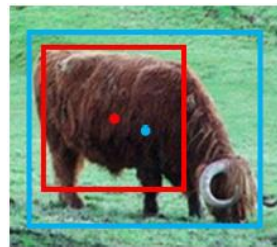
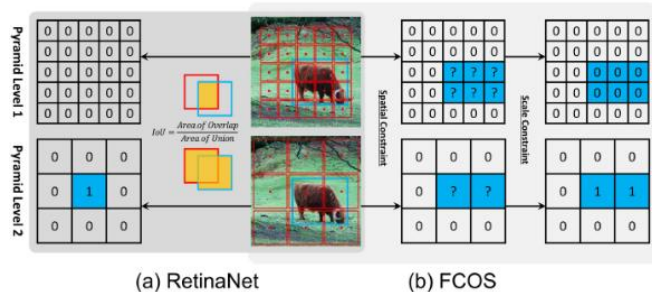
(c) FCOS

Classification \ Regression	Box	Point
	Intersection over Union	36.9
Spatial and Scale Constraint	37.8	37.8

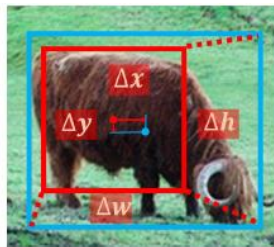


## 实用物体检测算法ATSS：探索本质区别

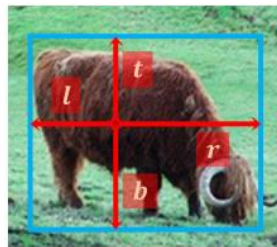
- ① 正负训练样本定义的不同
- ② 回归的起点不同



(a) Positive sample



(b) RetinaNet



(c) FCOS

Classification \ Regression	Box	Point
	Intersection over Union	36.9
Spatial and Scale Constraint	37.8	37.8

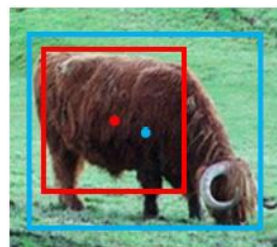
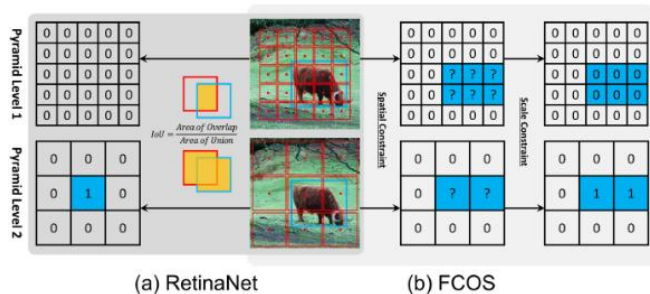




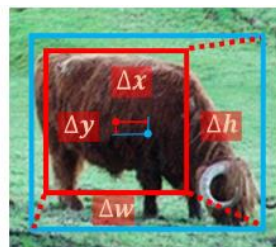
# 实用物体检测算法ATSS：探索本质区别

① 正负训练样本定义的不同

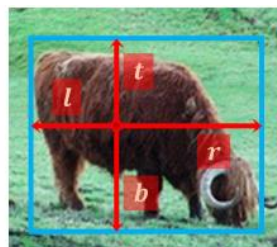
② 回归的起点不同



(a) Positive sample



(b) RetinaNet



(c) FCOS

Classification \ Regression	Box	Point
	Intersection over Union	36.9
Regression	37.0	37.8
	Spatial and Scale Constraint	37.8



## 实用物体检测算法ATSS：自适应训练样本选取

① 正负训练样本定义的不同

② ~~回归的起点不同~~



## 实用物体检测算法ATSS：自适应训练样本选取

### ① 正负训练样本定义的不同

### ② 回归的起点不同

```
for each level  $i \in [1, \mathcal{L}]$  do  
     $\mathcal{S}_i \leftarrow$  select  $k$  anchors from  $A_i$  whose center are closest  
    to the center of ground-truth  $g$  based on L2 distance;  
     $\mathcal{C}_g = \mathcal{C}_g \cup \mathcal{S}_i$ ;  
end for
```

```
compute IoU between  $\mathcal{C}_g$  and  $g$ :  $\mathcal{D}_g = IoU(\mathcal{C}_g, g)$ ;  
compute mean of  $\mathcal{D}_g$ :  $m_g = Mean(\mathcal{D}_g)$ ;  
compute standard deviation of  $\mathcal{D}_g$ :  $v_g = Std(\mathcal{D}_g)$ ;  
compute IoU threshold for ground-truth  $g$ :  $t_g = m_g + v_g$ ;
```

- $L$ 是检测层的个数
- $A_i$ 是第 $i$ 个检测层所关联的锚框
- $g$ 是真实标注
- $k$ 是超参数，默认值为9
- $\mathcal{S}_i$ 是 $g$ 在第 $i$ 个检测层上选取的 $k$ 个候选正样本
- $\mathcal{C}_g$ 是 $g$ 所有的候选正样本



## 实用物体检测算法ATSS：自适应训练样本选取

① 正负训练样本定义的不同

② 回归的起点不同

**for** each level  $i \in [1, \mathcal{L}]$  **do**

$\mathcal{S}_i \leftarrow$  select  $k$  anchors from  $A_i$  whose center are closest  
to the center of ground-truth  $g$  based on L2 distance;

$\mathcal{C}_g = \mathcal{C}_g \cup \mathcal{S}_i$ ;

**end for**

compute IoU between  $\mathcal{C}_g$  and  $g$ :  $\mathcal{D}_g = IoU(\mathcal{C}_g, g)$ ;

compute mean of  $\mathcal{D}_g$ :  $m_g = Mean(\mathcal{D}_g)$ ;

compute standard deviation of  $\mathcal{D}_g$ :  $v_g = Std(\mathcal{D}_g)$ ;

compute IoU threshold for ground-truth  $g$ :  $t_g = m_g + v_g$ ;

■  $\mathcal{D}_g$ 是所有候选正样本 $\mathcal{C}_g$ 跟 $g$ 的IoU



## 实用物体检测算法ATSS：自适应训练样本选取

① 正负训练样本定义的不同

② 回归的起点不同

**for** each level  $i \in [1, \mathcal{L}]$  **do**

$\mathcal{S}_i \leftarrow$  select  $k$  anchors from  $A_i$  whose center are closest  
to the center of ground-truth  $g$  based on L2 distance;

$\mathcal{C}_g = \mathcal{C}_g \cup \mathcal{S}_i$ ;

**end for**

compute IoU between  $\mathcal{C}_g$  and  $g$ :  $\mathcal{D}_g = \text{IoU}(\mathcal{C}_g, g)$ ;

compute mean of  $\mathcal{D}_g$ :  $m_g = \text{Mean}(\mathcal{D}_g)$ ;

compute standard deviation of  $\mathcal{D}_g$ :  $v_g = \text{Std}(\mathcal{D}_g)$ ;

compute IoU threshold for ground-truth  $g$ :  $t_g = m_g + v_g$ ;

- $m_g$ 是 $g$ 跟所有候选正样本 $\mathcal{C}_g$ 的IoU的均值
- $v_g$ 是 $g$ 跟所有候选正样本 $\mathcal{C}_g$ 的IoU的标准差



## 实用物体检测算法ATSS：自适应训练样本选取

① 正负训练样本定义的不同

② 回归的起点不同

**for** each level  $i \in [1, \mathcal{L}]$  **do**

$\mathcal{S}_i \leftarrow$  select  $k$  anchors from  $A_i$  whose center are closest to the center of ground-truth  $g$  based on L2 distance;

$\mathcal{C}_g = \mathcal{C}_g \cup \mathcal{S}_i$ ;

**end for**

compute IoU between  $\mathcal{C}_g$  and  $g$ :  $\mathcal{D}_g = IoU(\mathcal{C}_g, g)$ ;

compute mean of  $\mathcal{D}_g$ :  $m_g = Mean(\mathcal{D}_g)$ ;

compute standard deviation of  $\mathcal{D}_g$ :  $v_g = Std(\mathcal{D}_g)$ ;

compute IoU threshold for ground-truth  $g$ :  $t_g = m_g + v_g$ ;

■  $t_g$ 是 $g$ 从所有候选正样本 $\mathcal{C}_g$ 中选取最终正样本的阈值

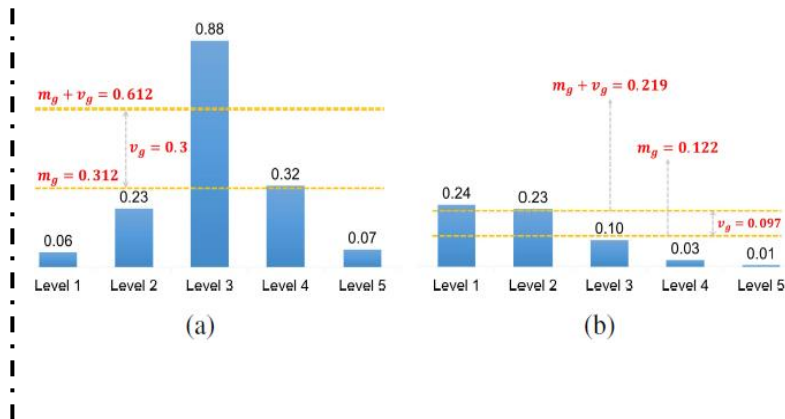


# 实用物体检测算法ATSS：自适应训练样本选取

## ① 正负训练样本定义的不同

## ② 回归的起点不同

```
for each level  $i \in [1, \mathcal{L}]$  do
     $\mathcal{S}_i \leftarrow$  select  $k$  anchors from  $A_i$  whose center are closest
    to the center of ground-truth  $g$  based on L2 distance;
     $\mathcal{C}_g = \mathcal{C}_g \cup \mathcal{S}_i$ ;
end for
compute IoU between  $\mathcal{C}_g$  and  $g$ :  $\mathcal{D}_g = IoU(\mathcal{C}_g, g)$ ;
compute mean of  $\mathcal{D}_g$ :  $m_g = Mean(\mathcal{D}_g)$ ;
compute standard deviation of  $\mathcal{D}_g$ :  $v_g = Std(\mathcal{D}_g)$ ;
compute IoU threshold for ground-truth  $g$ :  $t_g = m_g + v_g$ ;
```







# 实用物体检测算法ATSS：自适应训练样本选取

## ① 正负训练样本定义的不同

## ② 回归的起点不同

for each level  $i \in [1, \mathcal{L}]$  do

$\mathcal{S}_i \leftarrow$  select  $k$  anchors from  $A_i$  whose center are closest to the center of ground-truth  $g$  based on L2 distance;

$\mathcal{C}_g = \mathcal{C}_g \cup \mathcal{S}_i$ ;

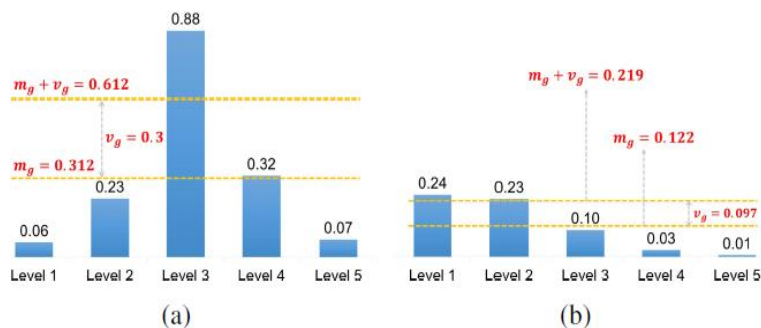
end for

compute IoU between  $\mathcal{C}_g$  and  $g$ :  $\mathcal{D}_g = IoU(\mathcal{C}_g, g)$ ;

compute mean of  $\mathcal{D}_g$ :  $m_g = Mean(\mathcal{D}_g)$ ;

compute standard deviation of  $\mathcal{D}_g$ :  $v_g = Std(\mathcal{D}_g)$ ;

compute IoU threshold for ground-truth  $g$ :  $t_g = m_g + v_g$ ;



Method	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
RetinaNet (#A=1)	37.0	55.1	39.9	21.4	41.2	48.6
RetinaNet (#A=1) + ATSS	39.3	57.5	42.8	24.3	43.3	51.3
FCOS	37.8	55.6	40.7	22.1	41.8	48.8
FCOS + Center sampling	38.6	57.4	41.4	22.3	42.5	49.8
FCOS + ATSS	39.2	57.3	42.4	22.7	43.1	51.5

ResNet-101	43.6	62.1	47.4	26.1	47.0	53.6
ResNeXt-32x8d-101	45.1	63.9	49.1	27.9	48.2	54.6
ResNeXt-64x4d-101	45.6	64.6	49.7	28.5	48.9	55.6
ResNet-101-DCN	46.3	64.7	50.4	27.7	49.8	58.4
ResNeXt-32x8d-101-DCN	47.7	66.6	52.1	29.3	50.8	59.7
ResNeXt-64x4d-101-DCN	47.7	66.5	51.9	29.7	50.8	59.4
ResNeXt-32x8d-101-DCN	50.6	68.6	56.1	<b>33.6</b>	<b>52.9</b>	62.2
ResNeXt-64x4d-101-DCN	<b>50.7</b>	<b>68.9</b>	<b>56.3</b>	33.2	<b>52.9</b>	<b>62.4</b>



## 实用物体检测算法ATSS：超参数分析

$k$	3	5	7	9	11	13	15	17	19
AP (%)	38.0	38.8	39.1	39.3	39.1	39.0	39.1	39.2	38.9

■ 超参k很鲁棒



## 实用物体检测算法ATSS：超参数分析

$k$	3	5	7	9	11	13	15	17	19
AP (%)	38.0	38.8	39.1	39.3	39.1	39.0	39.1	39.2	38.9

Scale	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
5	39.0	57.9	41.9	23.2	42.8	50.5
6	39.2	57.6	42.5	23.5	42.8	51.1
7	39.3	57.6	42.4	22.9	43.2	51.3
8	39.3	57.5	42.8	24.3	43.3	51.3
9	38.9	56.5	42.0	22.9	42.4	50.3

Aspect Ratio	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
4:1	39.1	57.2	42.3	23.1	43.1	51.4
2:1	39.0	56.9	42.5	23.3	43.5	50.6
1:1	39.3	57.5	42.8	24.3	43.3	51.3
2:1	39.3	57.4	42.3	22.8	43.4	51.0
4:1	39.1	56.9	42.6	22.9	42.9	50.7

■ 超参k很鲁棒

■ ATSS只在每个位置  
铺设一个锚框

■ ATSS使用不同尺度  
和比例的锚框，精度  
变化不大

■ ATSS对不同尺度和  
比例的锚框非常鲁棒



## 实用物体检测算法ATSS：最后一个不同点

- ① 正负训练样本定义的不同
- ② 回归的起点不同
- ③ 每个位置铺设的样本个数不同？

Inconsistency	FCOS	RetinaNet (#A=1)					
GroupNorm	✓	✓	✓	✓	✓	✓	✓
GIoU Loss	✓		✓	✓	✓	✓	✓
In GT Box	✓			✓	✓	✓	✓
Centerness	✓				✓	✓	✓
Scalar	✓					✓	✓
AP (%)	37.8	32.5	33.4	34.9	35.3	36.8	37.0

Method	#sc	#ar	AP	AP <sub>50</sub>	AP <sub>75</sub>
RetinaNet (#A=9)	3	3	36.3	55.2	38.8
+Imprs.	3	3	38.4	56.2	41.6
+Imprs.+ATSS	3	3	39.2	57.6	42.7
+Imprs.+ATSS	3	1	39.3	57.7	42.6
+Imprs.+ATSS	1	3	39.2	57.1	42.5
+Imprs.+ATSS	1	1	39.3	57.5	42.8

- 在基于IoU选取正负样本的情况下，每个位置铺设更多的锚框可以提升精度



## 实用物体检测算法ATSS：最后一个不同点

- ① 正负训练样本定义的不同
- ② 回归的起点不同
- ③ 每个位置铺设的样本个数不同？

Inconsistency	FCOS	RetinaNet (#A=1)					
GroupNorm	✓	✓	✓	✓	✓	✓	✓
GIoU Loss	✓		✓	✓	✓	✓	✓
In GT Box	✓			✓	✓	✓	✓
Centerness	✓				✓	✓	✓
Scalar	✓						✓
AP (%)	37.8	32.5	33.4	34.9	35.3	36.8	37.0

Method	#sc	#ar	AP	AP <sub>50</sub>	AP <sub>75</sub>
RetinaNet (#A=9)	3	3	36.3	55.2	38.8
+Imprs.	3	3	38.4	56.2	41.6
+Imprs.+ATSS	3	3	39.2	57.6	42.7
+Imprs.+ATSS	3	1	39.3	57.7	42.6
+Imprs.+ATSS	1	3	39.2	57.1	42.5
+Imprs.+ATSS	1	1	39.3	57.5	42.8

- 在基于IoU选取正负样本的情况下，每个位置铺设更多的锚框可以提升精度
- 在所提出的ATSS算法下，每个位置铺设更多的锚框没有带来性能的提升



## 实用物体检测算法ATSS：最后一个不同点

- ① 正负训练样本定义的不同
- ② 回归的起点不同
- ③ 每个位置铺设的样本个数不同？

Inconsistency	FCOS	RetinaNet (#A=1)					
GroupNorm	✓	✓	✓	✓	✓	✓	✓
GIoU Loss	✓		✓	✓	✓	✓	✓
In GT Box	✓			✓	✓	✓	✓
Centerness	✓				✓	✓	✓
Scalar	✓						✓
AP (%)	37.8	32.5	33.4	34.9	35.3	36.8	37.0

Method	#sc	#ar	AP	AP <sub>50</sub>	AP <sub>75</sub>
RetinaNet (#A=9)	3	3	36.3	55.2	38.8
+Imprs.	3	3	38.4	56.2	41.6
+Imprs.+ATSS	3	3	39.2	57.6	42.7
+Imprs.+ATSS	3	1	39.3	57.7	42.6
+Imprs.+ATSS	1	3	39.2	57.1	42.5
+Imprs.+ATSS	1	1	39.3	57.5	42.8

- 在选取正负样本的情况下，每个位置铺设更多的锚框可以提升精度
- 在所提出的ATSS算法下，每个位置铺设更多的锚框没有带来性能的提升
- 因此，只要合理地选取训练正负样本，每个位置铺设多少个锚框，最终结果都相似
- 结论：如果发挥多层锚框的作用，还需要进一步探索



# 实用物体检测算法ATSS：代码开源

sfzhang15 / ATSS

Watch

22

Star

646

Fork

94

Code

Issues 6

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

Settings

Bridging the Gap Between Anchor-based and Anchor-free Detection via Adaptive Training Sample Selection, CVPR, Oral, 2020

Edit

[Manage topics](#)

14 commits

1 branch

0 packages

0 releases

1 contributor

[View license](#)

Branch: master

[New pull request](#)

[Create new file](#)

[Upload files](#)

[Find file](#)

[Clone or download](#)



sfzhang15 Update README.md

Latest commit 3a6f8d8 17 days ago

atss\_core

Update loss.py

2 months ago

configs

Initial commit

4 months ago

demo

Initial commit

4 months ago

docker

Initial commit

4 months ago

<https://github.com/sfzhang15/ATSS>



## Paper award nominees

Weakly-supervised Domain Adaptation via GAN and Mesh Model for Estimating 3D Hand Poses Interacting Objects

Seungryul Baek; Kwang In Kim; Tae-Kyun Kim

Unsupervised Learning of Probably Symmetric Deformable 3D Objects from Images in the Wild

Shangzhe Wu; Christian Rupprecht; Andrea Vedaldi

Bridging the Gap Between Anchor-based and Anchor-free Detection via Adaptive Training Sample Selection

Shifeng Zhang; Cheng Chi; Yongqiang Yao; Zhen Lei; Stan Li

Momentum Contrast for Unsupervised Visual Representation Learning

Kaiming He; Haoqi Fan; Yuxin Wu; Saining Xie; Ross Girshick

BSP-Net: Generating Compact Meshes via Binary Space Partitioning

Zhiqin Chen; Andrea Tagliasacchi; Hao Zhang

Disentangled image generation through structured noise injection

Yazeed Alharbi; Peter Wonka

UC-Net: Uncertainty Inspired RGB-D Saliency Detection via Conditional Variational Autoencoders

Jing Zhang; Deng-Ping Fan; Yuchao Dai; Saeed Anwar; Fatemeh Sadat Saleh; Tong Zhang; Nick Barnes

TextureFusion: High-Quality Texture Acquisition for Real-Time RGB-D Scanning

Joo Ho Lee; Hyunho Ha; Yue Dong; Xin Tong; Min H. Kim

Controllable Orthogonalization in Training DNNs

Lei Huang; Li Liu; Fan Zhu; Diwen Wan; Zehuan Yuan; Bo Li; Ling Shao

DeepCap: Monocular Human Performance Capture Using Weak Supervision

Marc Habermann; Weipeng Xu; Michael Zollhöfer; Gerard Pons-Moll; Christian Theobalt

Total3DUnderstanding: Joint Layout, Object Pose and Mesh Reconstruction for Indoor Scenes from a Single Image

Yinyu Nie; Xiaoguang Han; Shihui Guo; Yujian Zheng; Jian Chang; Jian.J Zhang

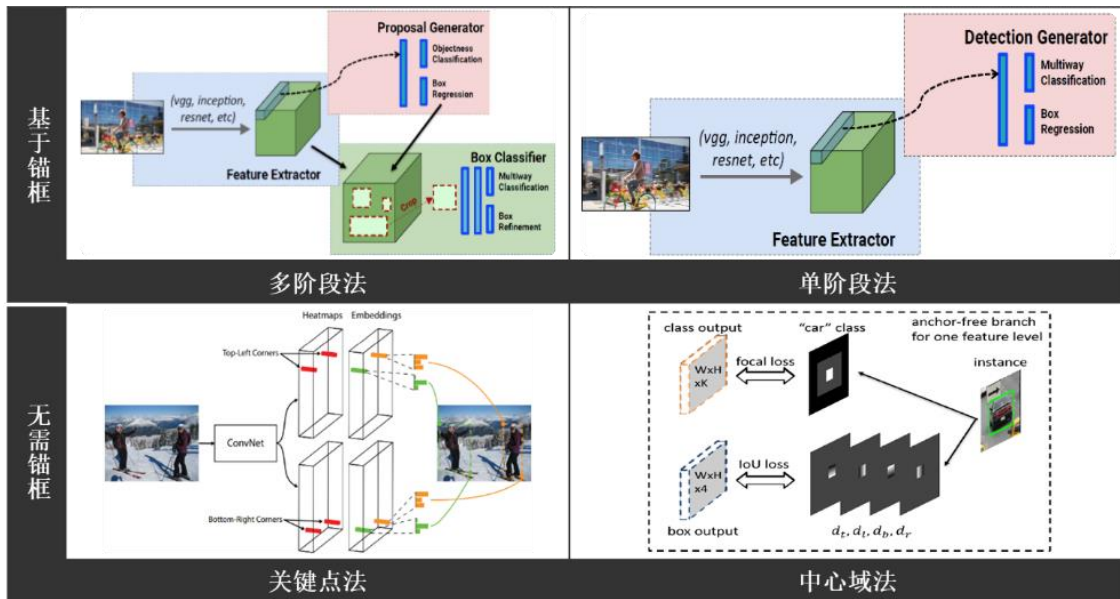
Transferring Cross-domain Knowledge for Video Sign Language Recognition

Dongxu Li; Xin Yu; Chenchen Xu; Lars Petersson; Hongdong Li



## 实用检测算法的研究思路：对比探索

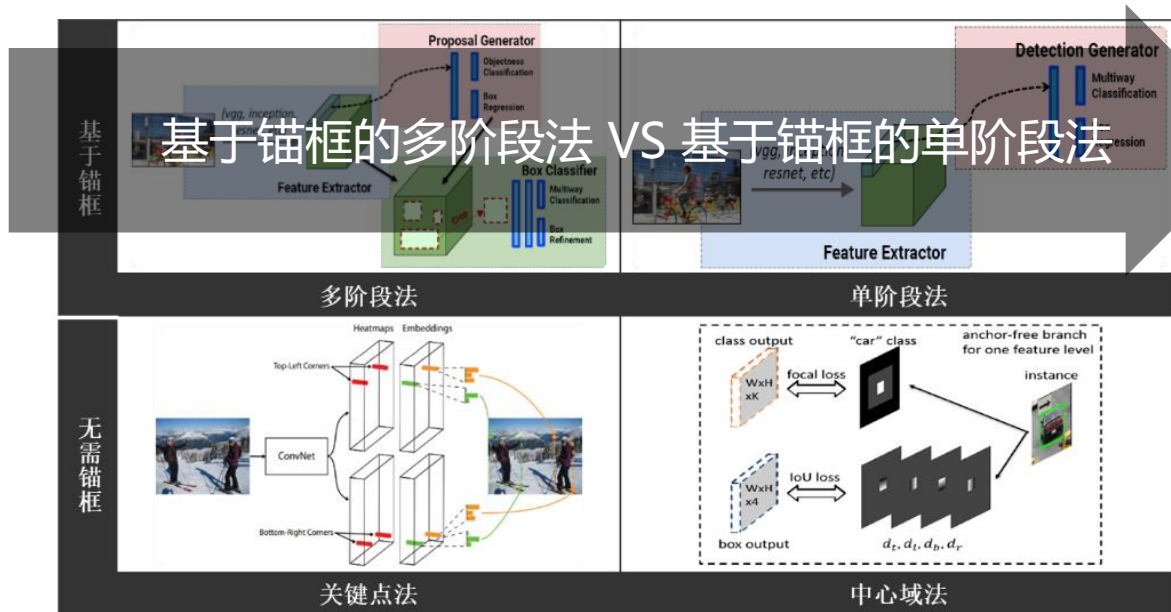
- 对比不同类型的检测算法，探索两者之间的本质区别
- 取之长补己短的改进思路，提出面向实用的全新算法





## 实用检测算法的研究思路：对比探索

- 对比不同类型的检测算法，探索两者之间的本质区别
- 取之长补己短的改进思路，提出面向实用的全新算法



① 二阶段的分类

② 二阶段的回归

③ 二阶段的特征

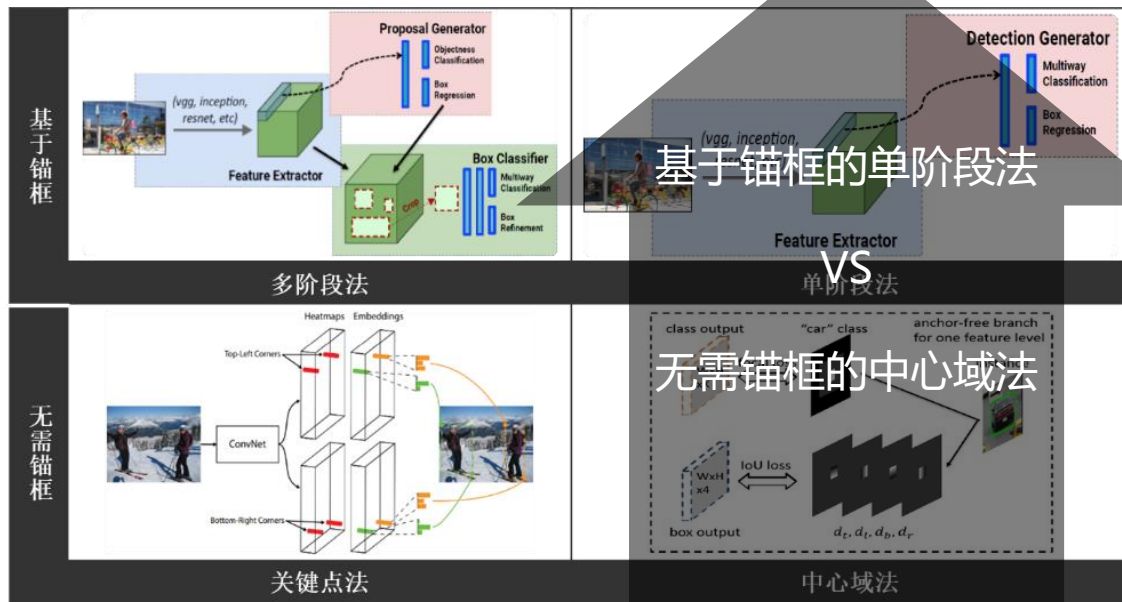
④ 特征校准

RefineDet算法



## 实用检测算法的研究思路：对比探索

- 对比不同类型的检测算法，探索两者之间的本质区别
- 取之长补己短的改进思路，提出面向实用的全新算法



① 正负样本定义

② 回归起始状态

③ 每个位置样本数量

ATSS算法



## 课程作业

### ■ 单步调试ATSS代码，总结RetinaNet和FCOS的区别

1. 代码链接 (<https://github.com/sfzhang15/ATSS>)
2. 按照安装教程，利用Anaconda配置ATSS的环境
3. 利用PyCharm单步调试ATSS的代码
4. 总结RetinaNet和FCOS的区别，熟悉ATSS的改进



结语

感谢聆听！

Thanks for Listening

