人工智能之深度学习

SSD

主讲人：Vincent ying

课程要求

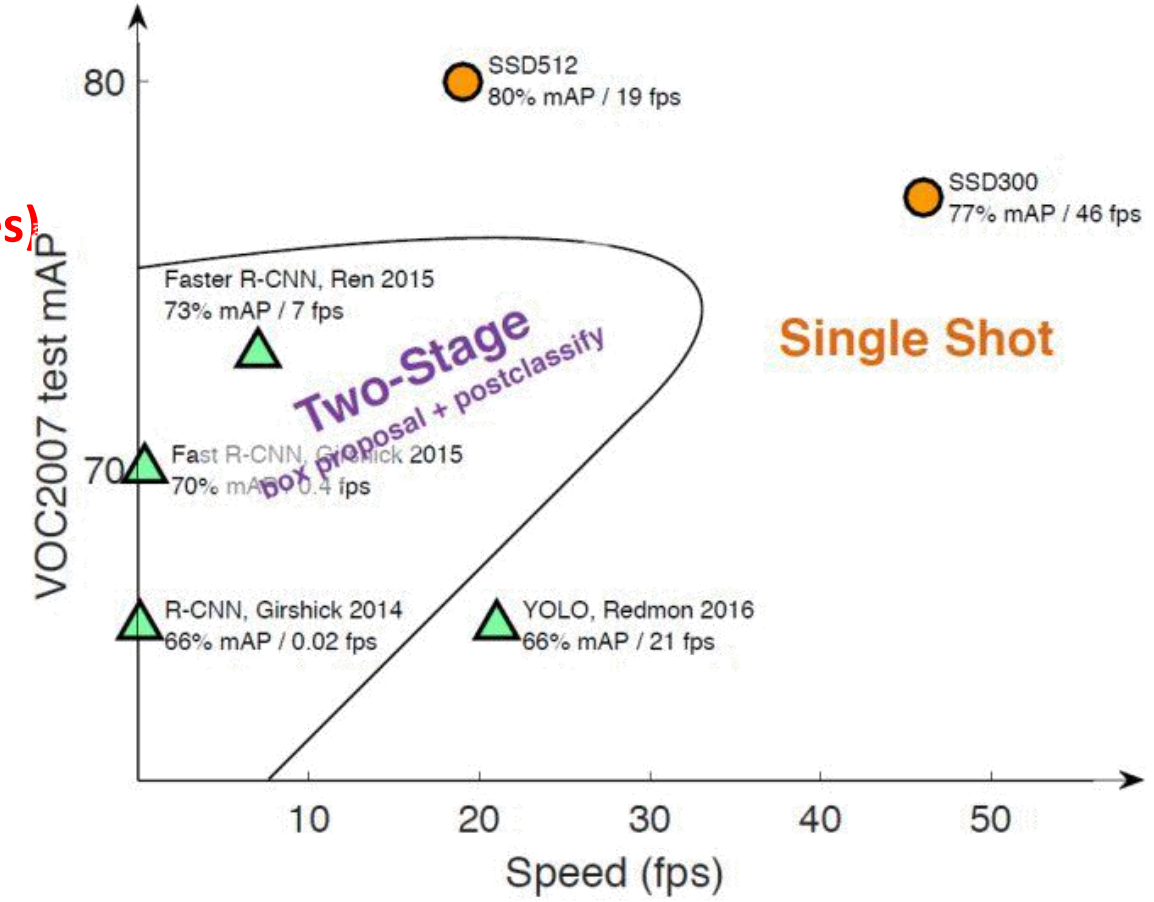
- 课上课下“九字”真言
  - 认真听，**善摘录，勤思考**
  - **多温故，乐实践**，再发散
- 四不原则
  - 不懒散惰性，不迟到早退
  - 不请假旷课，不拖延作业
- 一点注意事项
  - 违反“四不原则”，不推荐就业

课程内容

- SSD

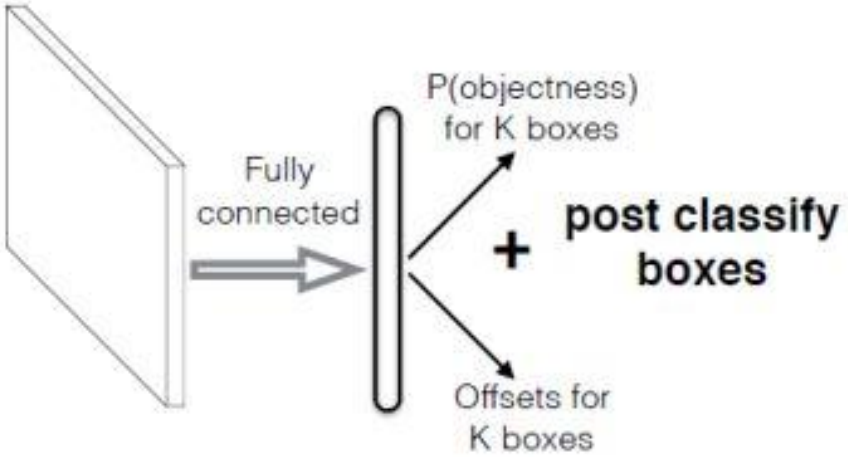
SSD

- Single Shot MultiBox Detector(SSD, 单步多框目标检测)
  - **One-Stage**
  - 均匀的密集抽样
    - **Prior boxes/Default boxes(Anchor boxes)**
    - 不同尺度抽样
  - 不同scale尺度的特征图抽样
  - 对于小目标检测效果不错
  - 预测速度快
  - 训练困难(正负样本极度不均衡)

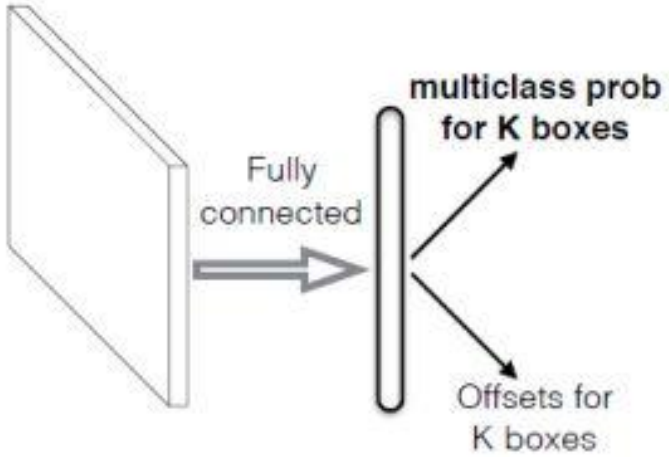


SSD

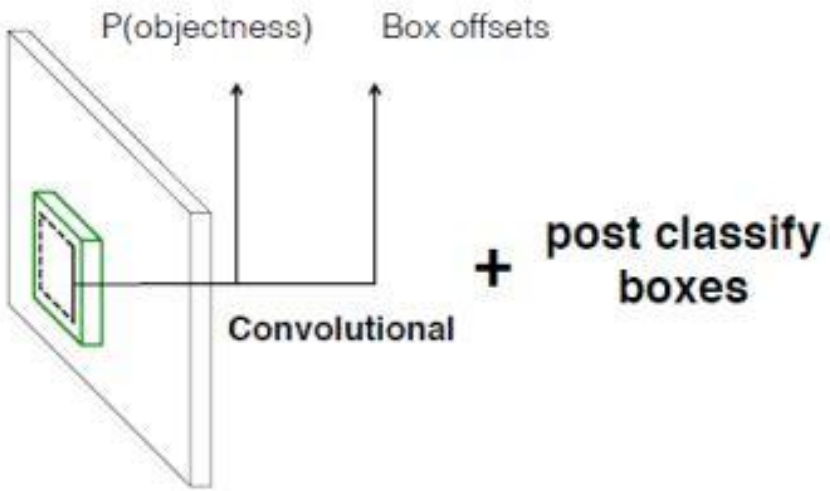
MultiBox [Erhan et al. CVPR14]



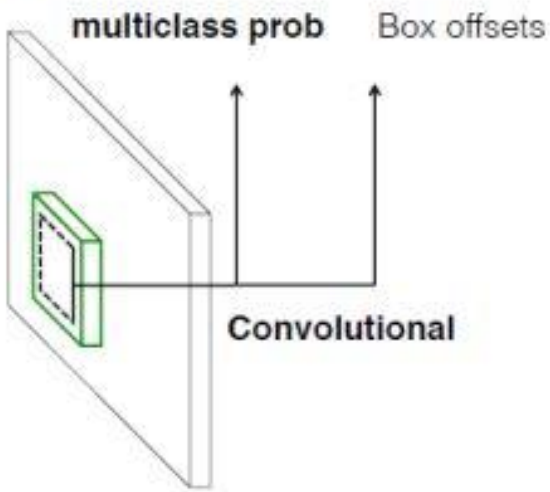
YOLO [Redmon et al. CVPR16]



Faster R-CNN [Ren et al. NIPS15]



SSD



卷积神经网络典型CNN-VGGNet

- VGG Net

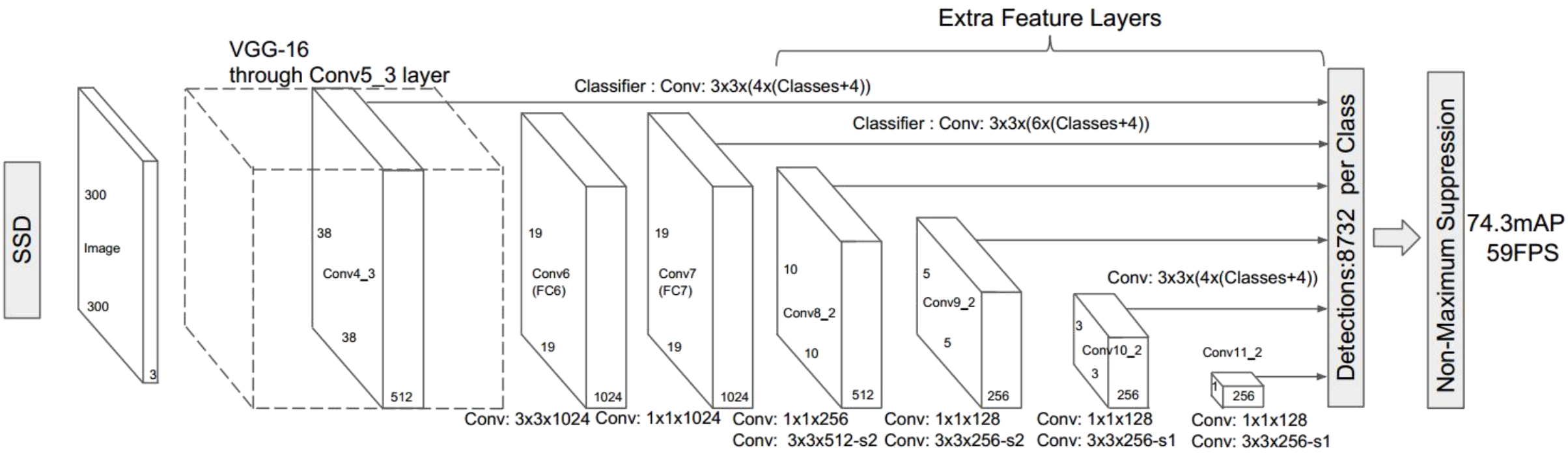
```
INPUT: [224x224x3]
CONV3-64: [224x224x64]
CONV3-64: [224x224x64]
POOL2: [112x112x64]
CONV3-128: [112x112x128]
CONV3-128: [112x112x128]
POOL2: [56x56x128]
CONV3-256: [56x56x256]
CONV3-256: [56x56x256]
CONV3-256: [56x56x256]
POOL2: [28x28x256]
CONV3-512: [28x28x512]
CONV3-512: [28x28x512]
CONV3-512: [28x28x512]
POOL2: [14x14x512]
CONV3-512: [14x14x512]
CONV3-512: [14x14x512]
CONV3-512: [14x14x512]
POOL2: [7x7x512]
FC: [1x1x4096]
FC: [1x1x4096]
FC: [1x1x1000]
```

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

最优模式

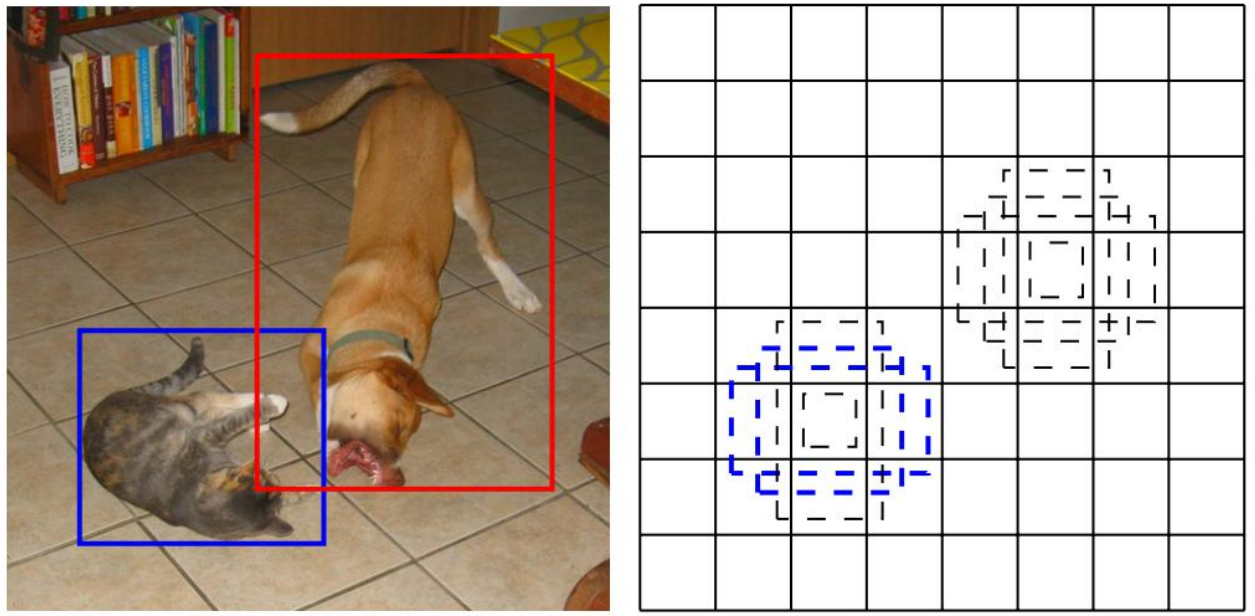
VGG-Net结构

SSD


SSD
300
Image
300
3
VGG-16
through Conv5_3 layer
38
Conv4_3
38
512
19
Conv6
(FC6)
19
1024
19
Conv7
(FC7)
19
1024
10
Conv8_2
10
512
5
Conv9_2
5
256
3
Conv10_2
3
256
Conv11_2
1
256
Extra Feature Layers
Classifier : Conv: 3x3x(4x(Classes+4))
Classifier : Conv: 3x3x(6x(Classes+4))
Conv: 3x3x(4x(Classes+4))
Conv: 3x3x1024
Conv: 1x1x1024
Conv: 1x1x256
Conv: 3x3x512-s2
Conv: 1x1x128
Conv: 3x3x256-s2
Conv: 1x1x128
Conv: 3x3x256-s1
Conv: 1x1x128
Conv: 3x3x256-s1
Detections:8732 per Class
Non-Maximum Suppression
74.3mAP
59FPS

SSD

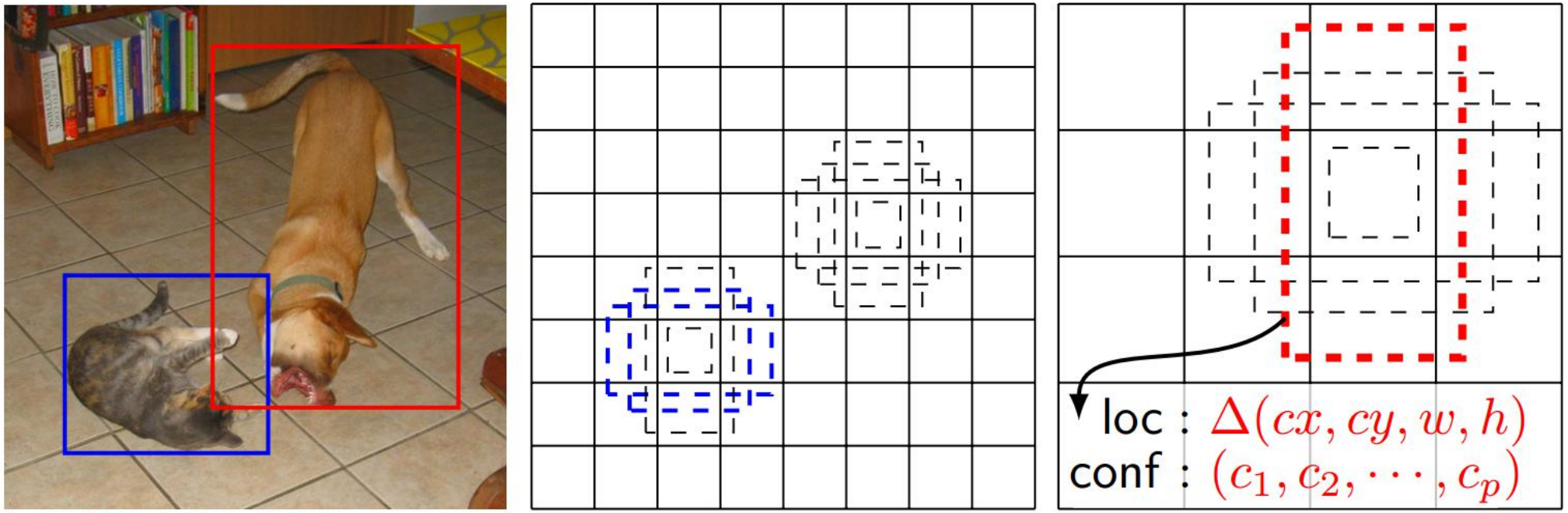
- 1. 重用Faster R-CNN的Anchors机制(**Default boxes and aspect ratios**)
  - 在feature map上提取各种不同尺度大小的default box，也就是类似Anchor的一系列大小固定的框。不同feature map上尺度是一样的。



(a) Image with GT boxes (b) 8 × 8 feature map

SSD

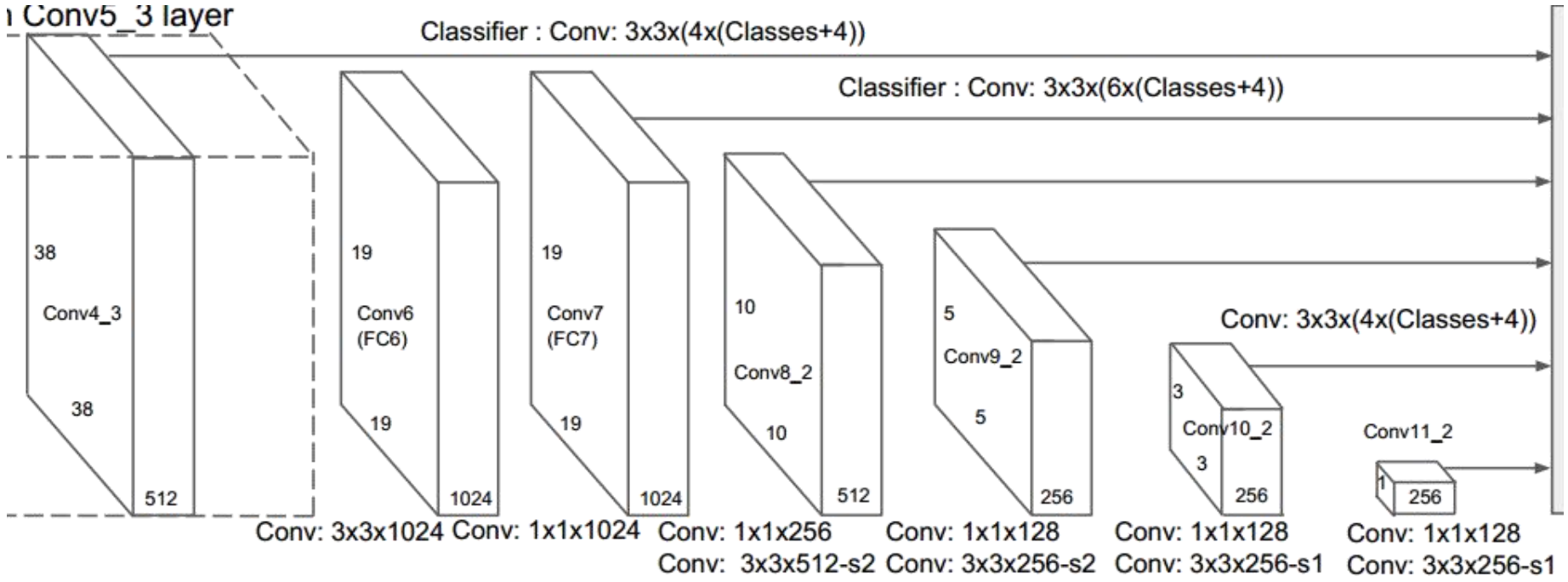
- 2. 多尺度特征图抽样(**Multi-scale feature maps for detection**)
  - 比较大的特征图负责小样本检测，比较小的特征图负责大样本检测。
  - 在不同尺度大小的feature map上提取default box。



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

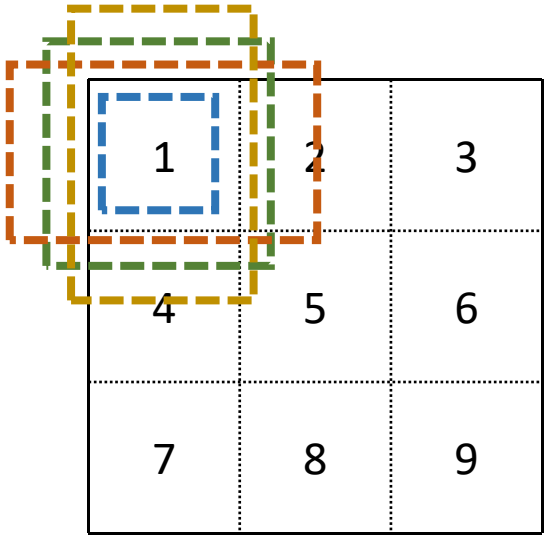
SSD

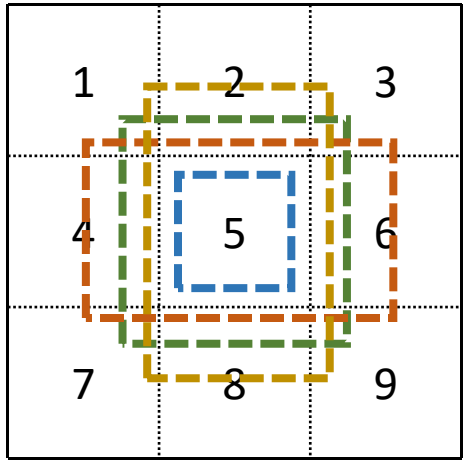
- 3. 全卷积网络结构(**Convolutional predictors for detection**)
  - 结合R FCN网络的优点，将所有的全连接网络全部更改为全卷积网络结构。
  - 使用卷积来提取候选框特征(offset box+score)。

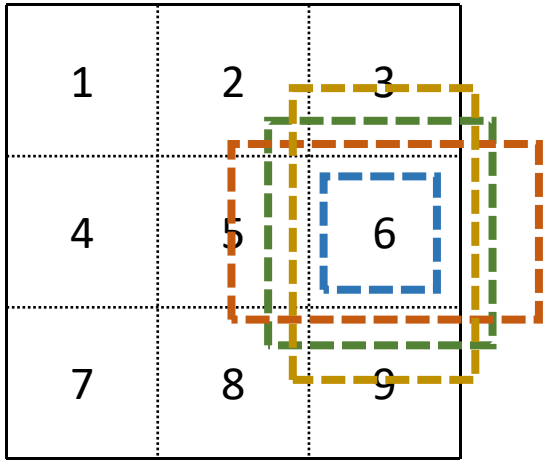


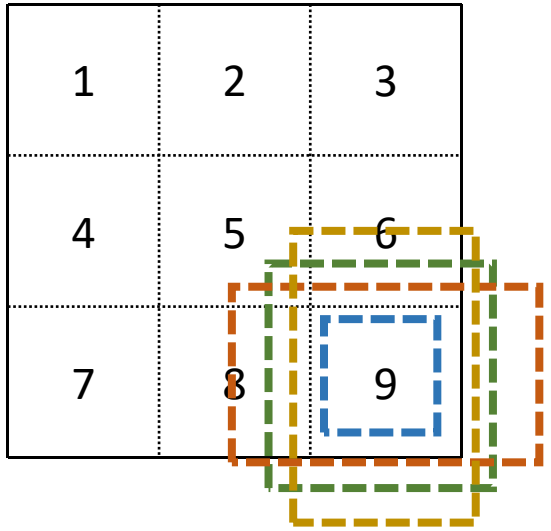
SSD

1	2	3
4	5	6
7	8	9

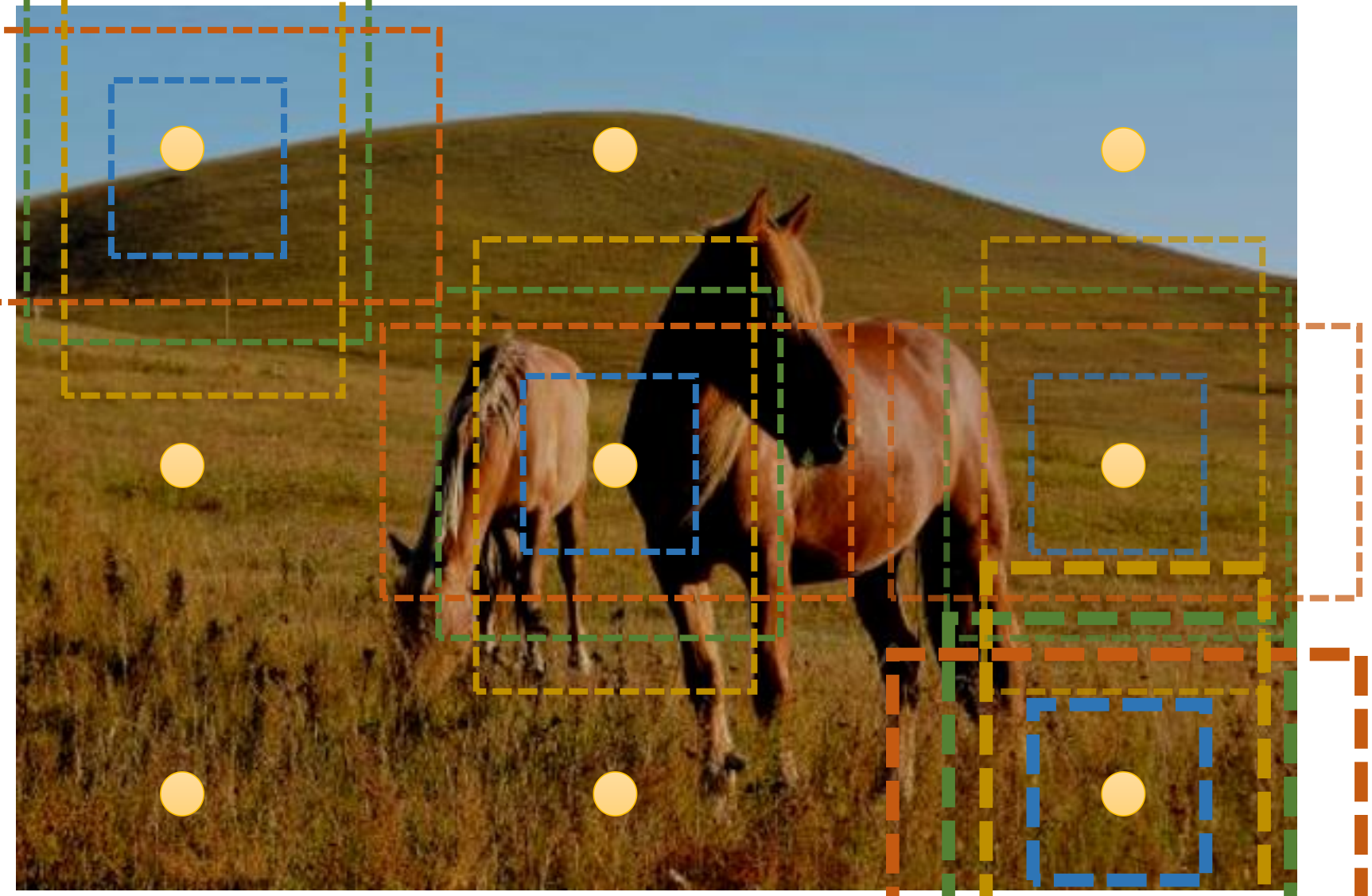




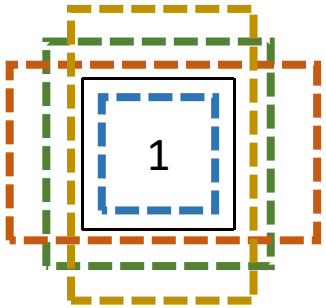


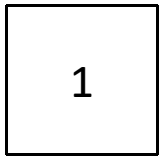


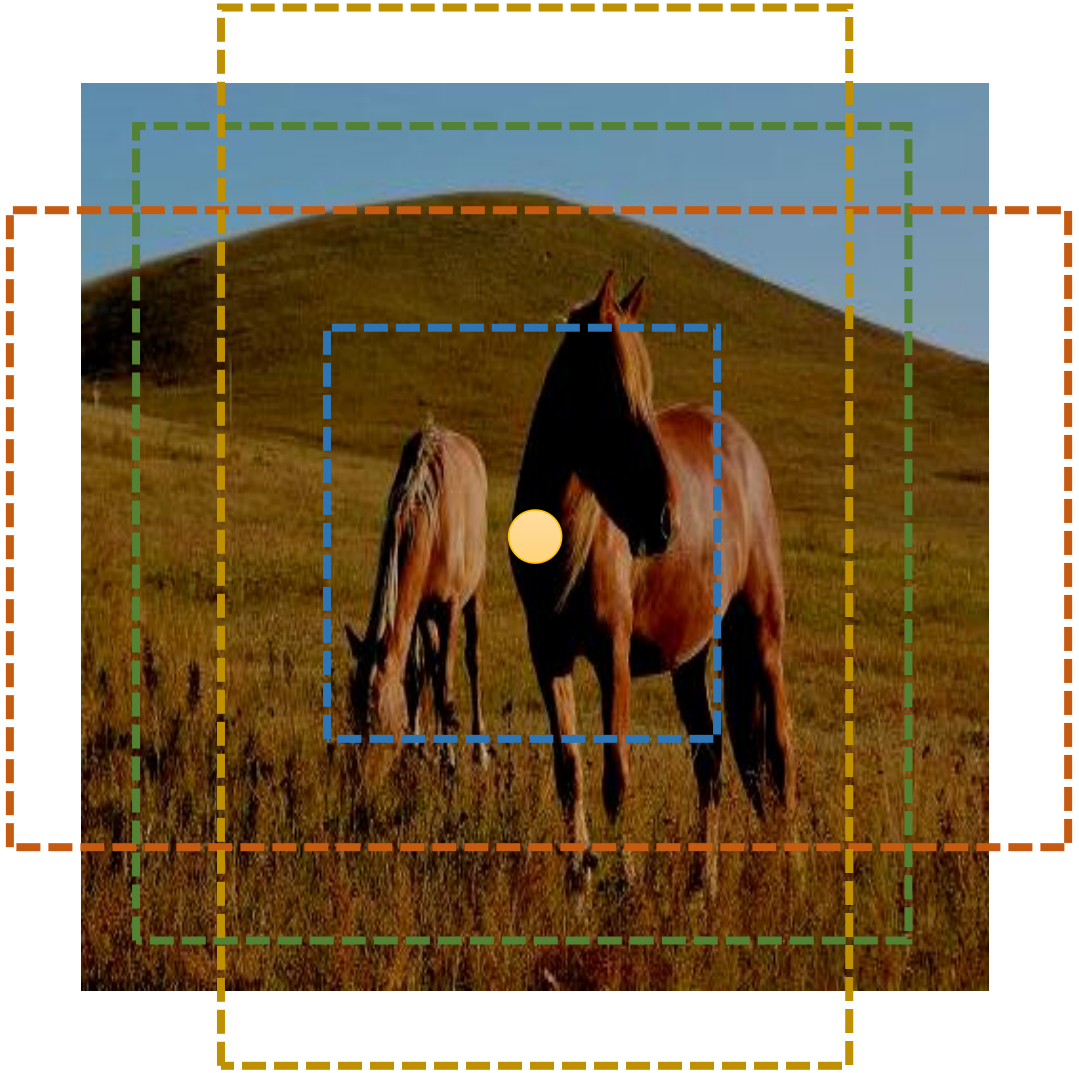
SSD



SSD



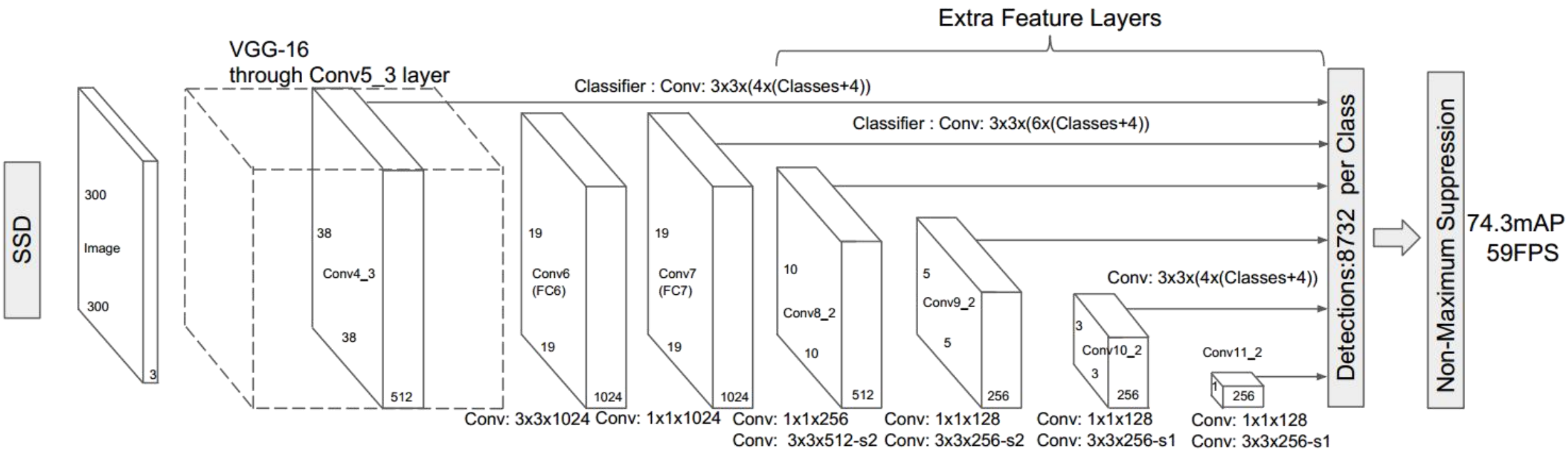




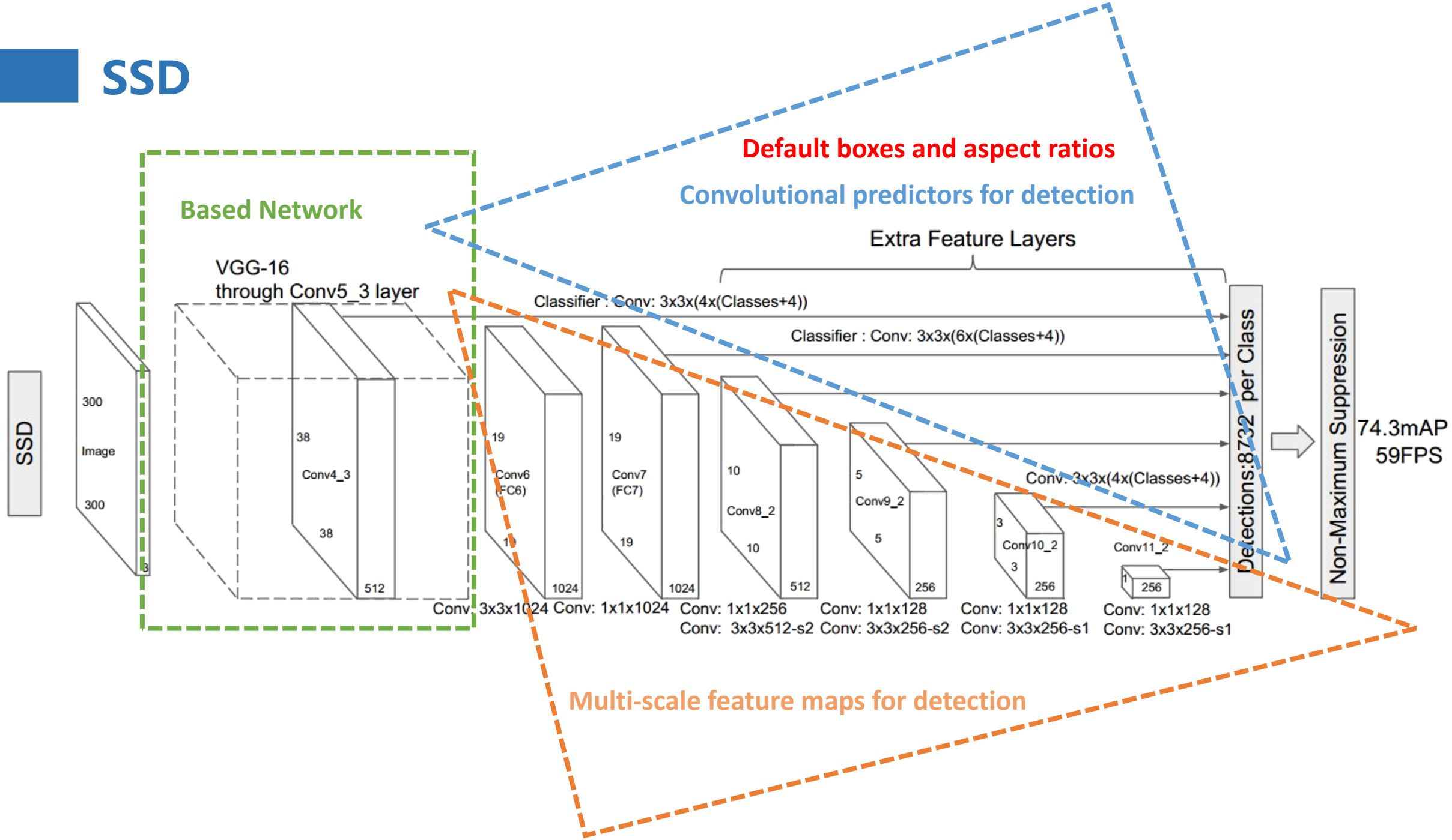
SSD

Layer Name	Output Size	Default Box Number	Total Box Number
conv4-3	38x38	4	5776
fc7	19x19	6	2166
conv8-2	10x10	6	600
conv9-2	5x5	6	150
conv10-2	3x3	4	36
conv11-2	1x1	4	4
			8732

SSD

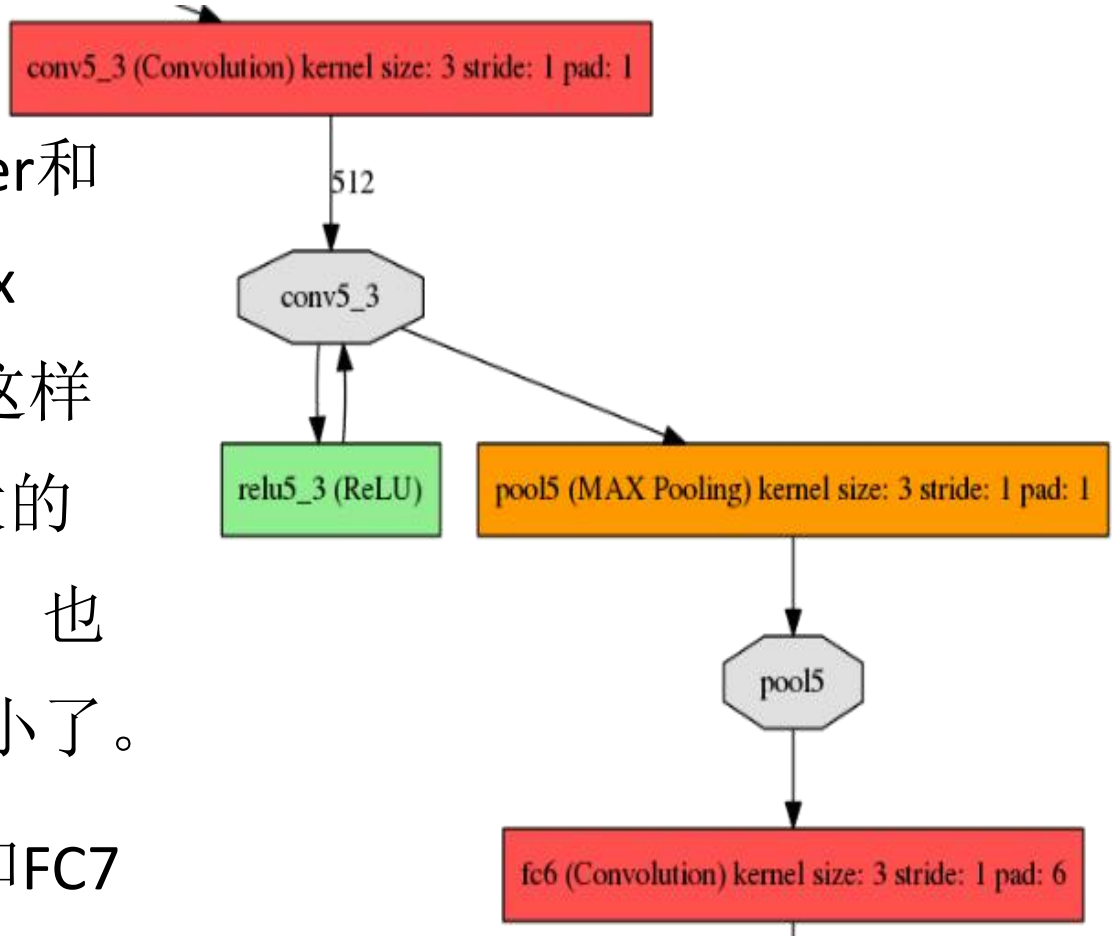

SSD
300
Image
300
3
VGG-16
through Conv5_3 layer
38
Conv4_3
38
512
19
Conv6
(FC6)
19
1024
19
Conv7
(FC7)
19
1024
10
Conv8_2
10
512
5
Conv9_2
5
256
3
Conv10_2
3
256
Conv11_2
1
256
Extra Feature Layers
Classifier : Conv: 3x3x(4x(Classes+4))
Classifier : Conv: 3x3x(6x(Classes+4))
Conv: 3x3x(4x(Classes+4))
Conv: 3x3x1024
Conv: 1x1x1024
Conv: 1x1x256
Conv: 3x3x512-s2
Conv: 1x1x128
Conv: 3x3x256-s2
Conv: 1x1x128
Conv: 3x3x256-s1
Conv: 1x1x128
Conv: 3x3x256-s1
Detections:8732 per Class
Non-Maximum Suppression
74.3mAP
59FPS

SSD

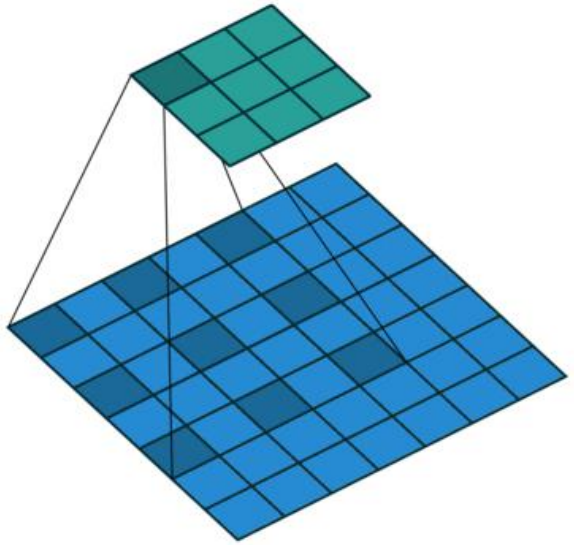

Default boxes and aspect ratios
Convolutional predictors for detection
Based Network
Extra Feature Layers
VGG-16
through Conv5_3 layer
Classifier : Conv: 3x3x(4x(Classes+4))
Classifier : Conv: 3x3x(6x(Classes+4))
Conv: 3x3x(4x(Classes+4))
SSD
300
Image
300
3
38
Conv4_3
38
512
19
Conv6
(FC6)
19
1024
19
Conv7
(FC7)
19
1024
10
Conv8_2
10
512
5
Conv9_2
5
256
3
Conv10_2
3
256
Conv11_2
1
256
Conv: 3x3x1024
Conv: 1x1x1024
Conv: 1x1x256
Conv: 3x3x512-s2
Conv: 1x1x128
Conv: 3x3x256-s2
Conv: 1x1x128
Conv: 3x3x256-s1
Conv: 1x1x128
Conv: 3x3x256-s1
Detections:8732 per Class
Non-Maximum Suppression
74.3mAP
59FPS
Multi-scale feature maps for detection

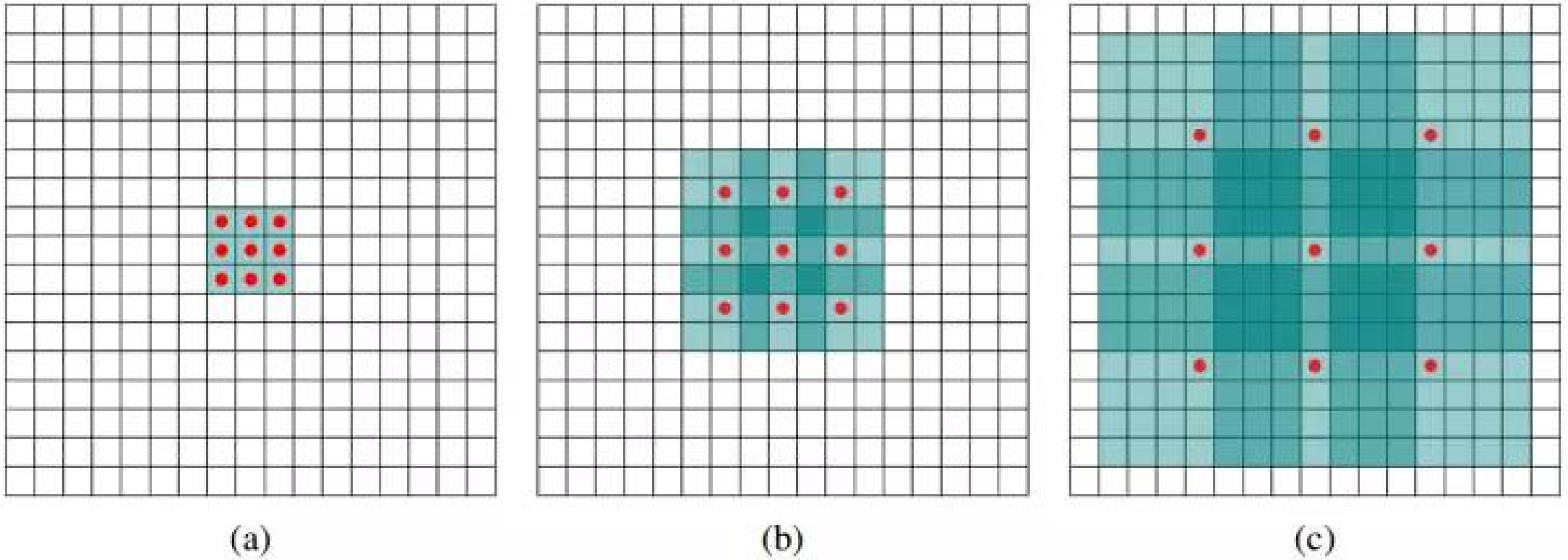
SSD

- 基础网络结构使用VGG，并且将FC6 Layer和FC7 Layer转换为卷积层，并将原来的Max Pooling5的大小从2x2-s2变化为3x3-s1，这样pooling5操作后feature map还是保持较大的尺寸，这样就会导致之后的感受野变小，也就是一个点对应到原始图形中的区域变小了。
- 为了保障感受野以及利用到原来的FC6和FC7的模型参数，使用atrous algorithm的方式来增大感受野，也就是**膨胀卷积/空洞卷积**。

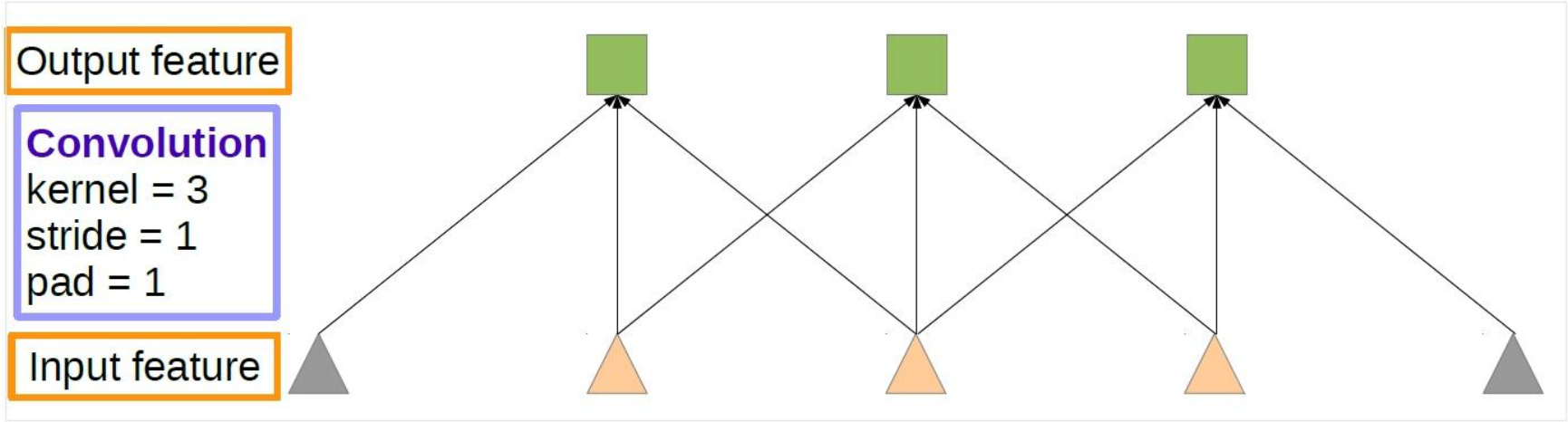


SSD

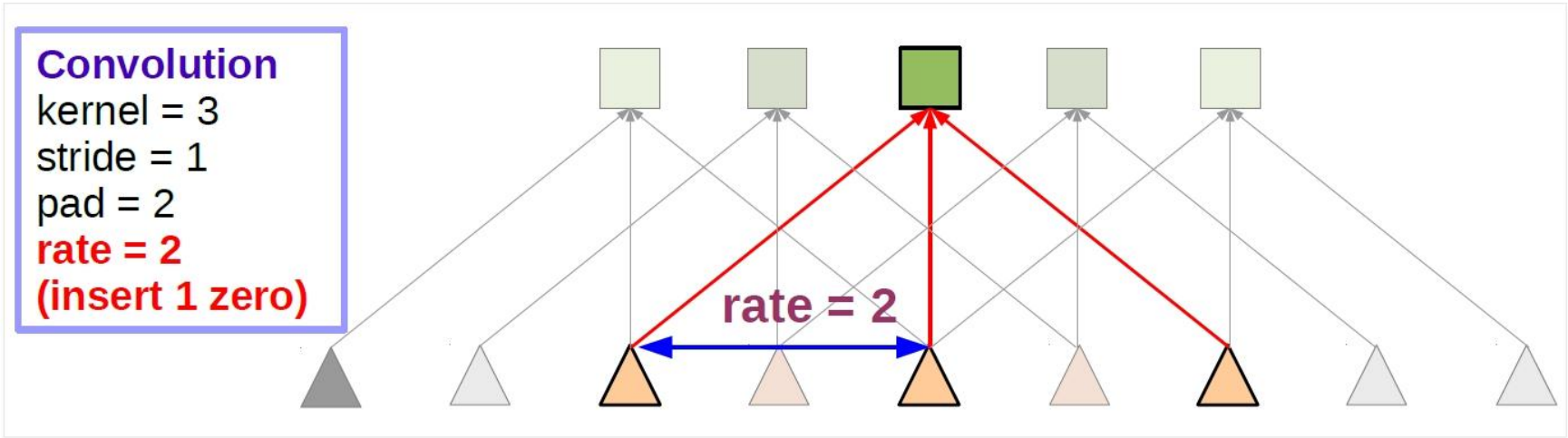




SSD

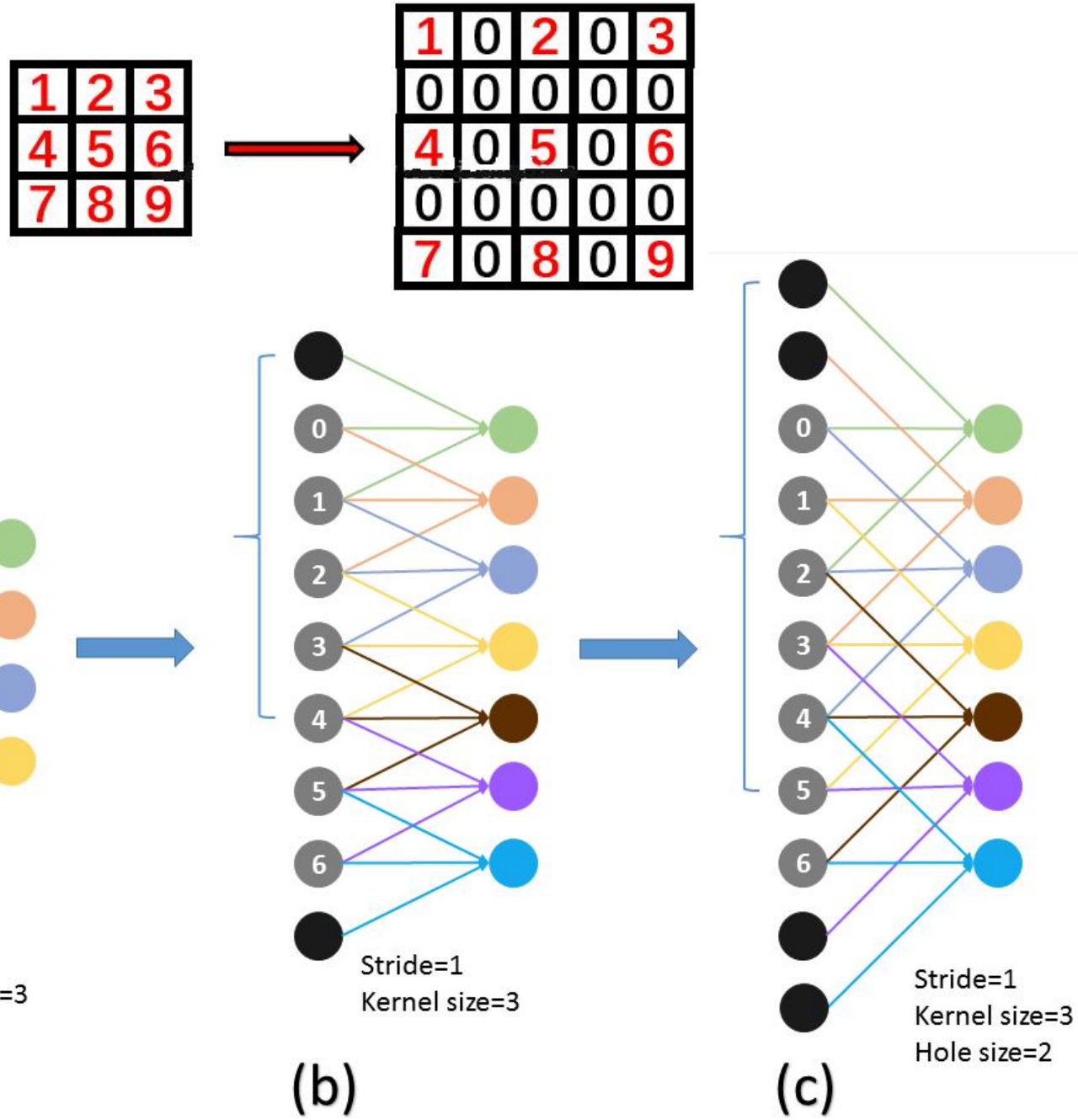


(a) Sparse feature extraction



(b) Dense feature extraction

SSD

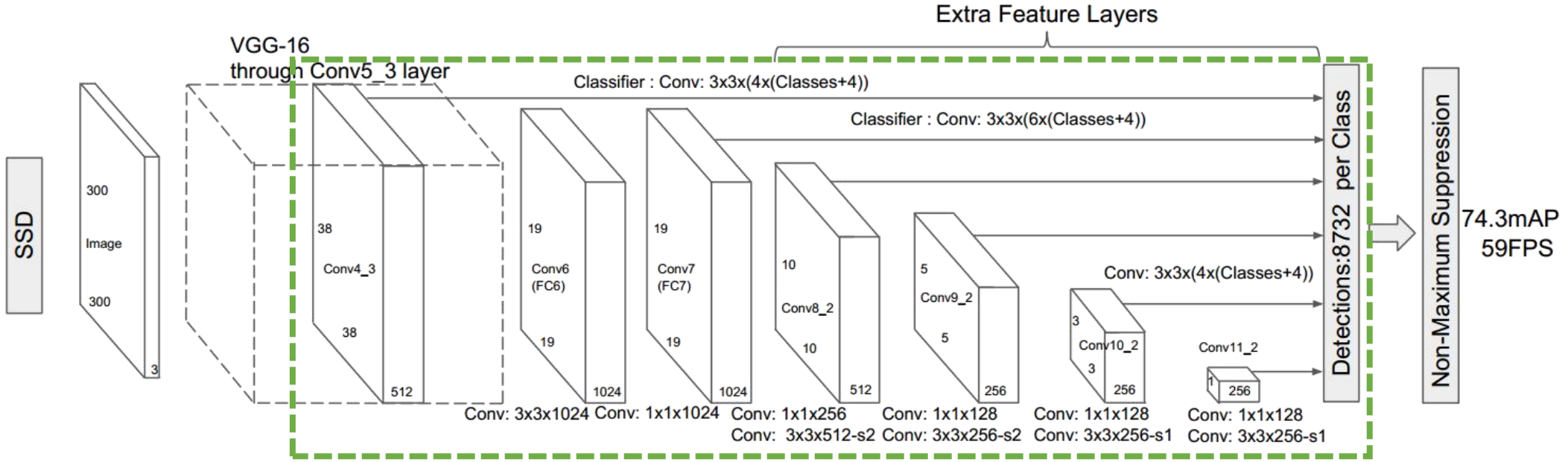

1 2 3
4 5 6
7 8 9
1 0 2 0 3
0 0 0 0 0
4 0 5 0 6
0 0 0 0 0
7 0 8 0 9
0
1
2
3
4
5
6
Stride=2
Kernel size=3
(a)
Stride=1
Kernel size=3
(b)
Stride=1
Kernel size=3
Hole size=2
(c)

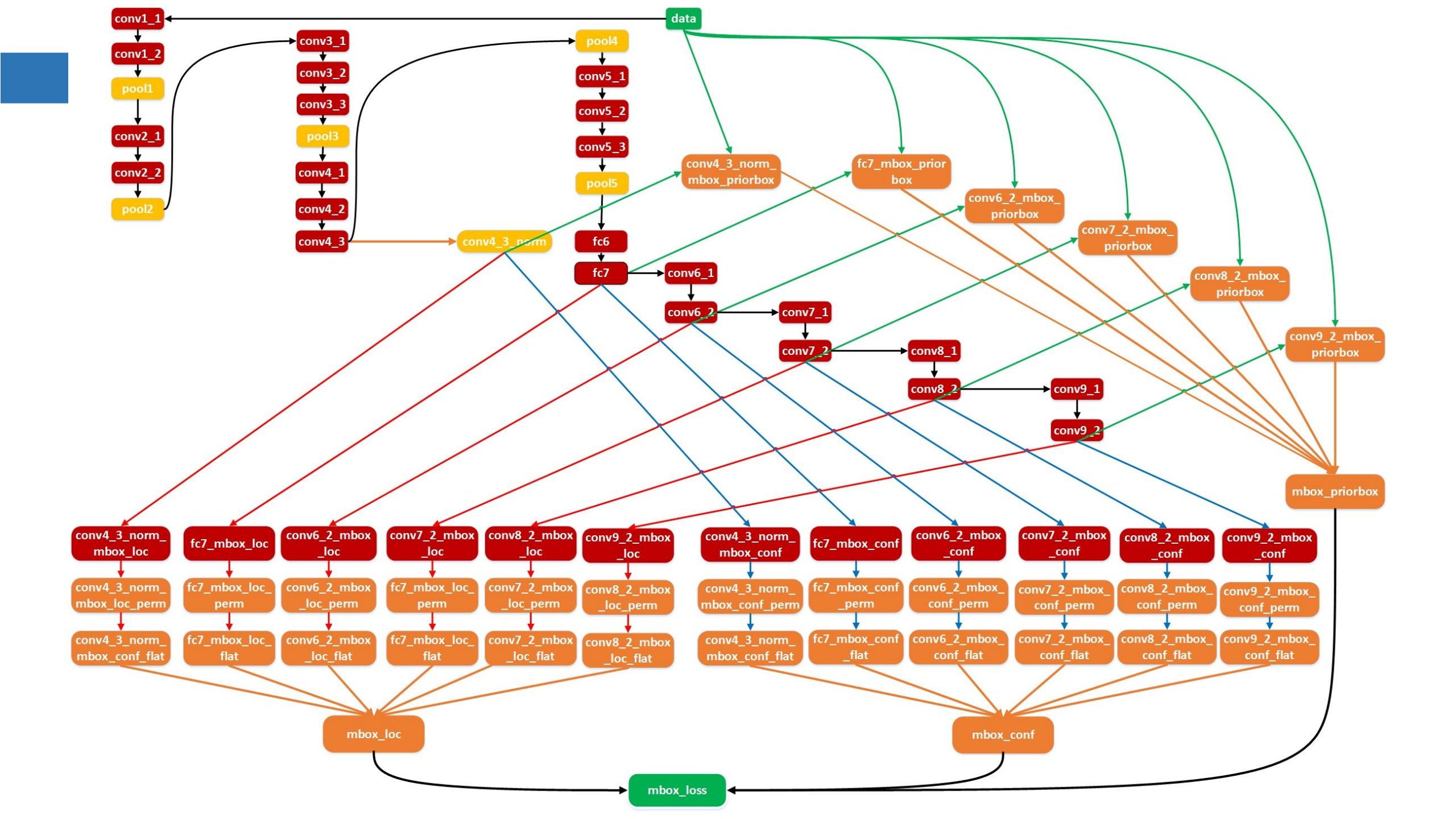
SSD

- **膨胀卷积核尺寸=膨胀系数*(原始卷积核尺寸-1)+1**
- Conv6(fc6)中卷积核kernel为3，pad为6， dilation为6，所以相当于真实的卷积核大小为13，pad为6是为了保障输出feature map大小尺度不变，仍为19x19。
- 膨胀卷积(Dilated Convolution)的存在是为了解决一下几个问题的：
  - 普通的数据上采样层参数不可学习；
  - 内部数据结构丢失，空间层级化信息丢失；
  - 小物体信息无法重建。
- TensorFlow中膨胀卷积/空洞卷积API：
  - tf.nn.atrous_conv2d(value, filters, rate, padding, name=None)

SSD

- 在基础网络之后，使用不同层次卷积的feature map来分别提取default box，对于每个layer的feature map使用两个并行的3x3卷积分别来提取位置信息(offset box)和置信度信息；结合Default box和Ground Truth box构建损失函数。
- 对于Con4_3的数据提取的时候，会先对feature map做一个L2 norm的操作，因为层次比较靠前，防止出现数据值过大的情况。




conv1_1
conv1_2
pool1
conv2_1
conv2_2
pool2
conv3_1
conv3_2
conv3_3
pool3
conv4_1
conv4_2
conv4_3
conv4_3_norm
pool4
conv5_1
conv5_2
conv5_3
pool5
fc6
fc7
data
conv4_3_norm_mbox_priorbox
fc7_mbox_priorbox
conv6_2_mbox_priorbox
conv7_2_mbox_priorbox
conv8_2_mbox_priorbox
conv9_2_mbox_priorbox
conv6_1
conv6_2
conv7_1
conv7_2
conv8_1
conv8_2
conv9_1
conv9_2
mbox_priorbox
conv4_3_norm_mbox_loc
fc7_mbox_loc
conv6_2_mbox_loc
conv7_2_mbox_loc
conv8_2_mbox_loc
conv9_2_mbox_loc
conv4_3_norm_mbox_conf
fc7_mbox_conf
conv6_2_mbox_conf
conv7_2_mbox_conf
conv8_2_mbox_conf
conv9_2_mbox_conf
conv4_3_norm_mbox_loc_perm
fc7_mbox_loc_perm
conv6_2_mbox_loc_perm
conv7_2_mbox_loc_perm
conv8_2_mbox_loc_perm
conv4_3_norm_mbox_conf_perm
fc7_mbox_conf_perm
conv6_2_mbox_conf_perm
conv7_2_mbox_conf_perm
conv8_2_mbox_conf_perm
conv9_2_mbox_conf_perm
conv4_3_norm_mbox_conf_flat
fc7_mbox_loc_flat
conv6_2_mbox_loc_flat
conv7_2_mbox_loc_flat
conv8_2_mbox_loc_flat
conv4_3_norm_mbox_conf_flat
fc7_mbox_conf_flat
conv6_2_mbox_conf_flat
conv7_2_mbox_conf_flat
conv8_2_mbox_conf_flat
conv9_2_mbox_conf_flat
mbox_loc
mbox_conf
mbox_loss

SSD

- 在CNN网络中，层次越深，feature map的尺寸(size)会越来越小，这样设计主要是为了以下两个目的：
  - 减少计算与内存的需求；
  - 最终提取的feature map在某种程度上具备平移和尺度不变性，契合分类的业务场景要求。
- 在目标检测场景中，经常需要处理不同尺度的物体，在某些网络中，会通过将图像转换为不同尺度大小的图像独立的通过网络处理，然后将这些不同尺度的图像结果合并，但是实际上，**在同一个网络中，对不同层次上的feature maps进行特征的处理实际上效果是一样的，并且所有尺度的物体处理参数是共享的，计算会更快。**

SSD

- CNN网络中不同layers有着不同尺寸的感受野(receptive fields)，也就是说不同layers上的一个点，在原输入图像中对应的尺寸大小是不一样的。
- SSD结构中，default boxes不需要和每一层layer的receptive fields对应，通过产生不同scale大小的boxes来负责图像中特定区域以及物体的特定尺寸。

SSD

- 在提取先验框的时候，主要通过**尺度(大小)**和**长宽比**两个方面来进行设置，其中在先验框尺度上遵守一个线性递增规则：**随着特征图大小降低，先验框尺度线性增加。**

论文中：S_min和S_max值分别为0.2和0.9

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m - 1}(k - 1), \quad k \in [1, m]$$

s_k为相比于原输入图像，先验框尺度的比例值，conv4-3对应的feature map上的s_k值特例为0.1。

m为5，即: fc7、conv8-2、conv9-2、conv10-2、conv11-2；对于conv4-3的尺度特定给为0.1

SSD

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m - 1}(k - 1), \quad k \in [1, m]$$

- Input Image Size: 300x300, S_min: 0.2, S_max: 0.9
-

Layer Name	s_k	Default Box Scale(s_k)
conv4-3	0.1	30
fc7	0.2	60
conv8-2	0.375	112
conv9-2	0.55	165
conv10-2	0.725	217
conv11-2	0.9	270
		image_size*s_k

SSD

- 在提取先验框的时候，主要通过**尺度(大小)**和**长宽比**两个方面来进行设置，在长宽比上，论文中建议比率值选择范围为: [1, 2, 3, 1/2, 1/3]。对于Conv4-3、Conv10-2以及Conv11-2这三层，由于仅使用4个先验框，不使用1:3和3:1的这个比例值。

$$a = \left\{1, 2, 3, \frac{1}{2}, \frac{1}{3}\right\}$$

$$w_k^r = s_k \cdot \sqrt{a_r} \qquad h_k^r = s_k \Big/ \sqrt{a_r}$$

SSD

- 除了使用上述5个长宽比外，还引入一个特殊尺度并且长宽比为1的先验框。引入这个框的主要目的是为了体现最终的候选框中出现两个长宽比为1但是大小不同的正方形先验框。

$$s_k' = \sqrt{s_k \cdot s_{k+1}}$$

$$s_{m+1} = 300 * (0.9 + 0.175) = 322$$

虚拟值

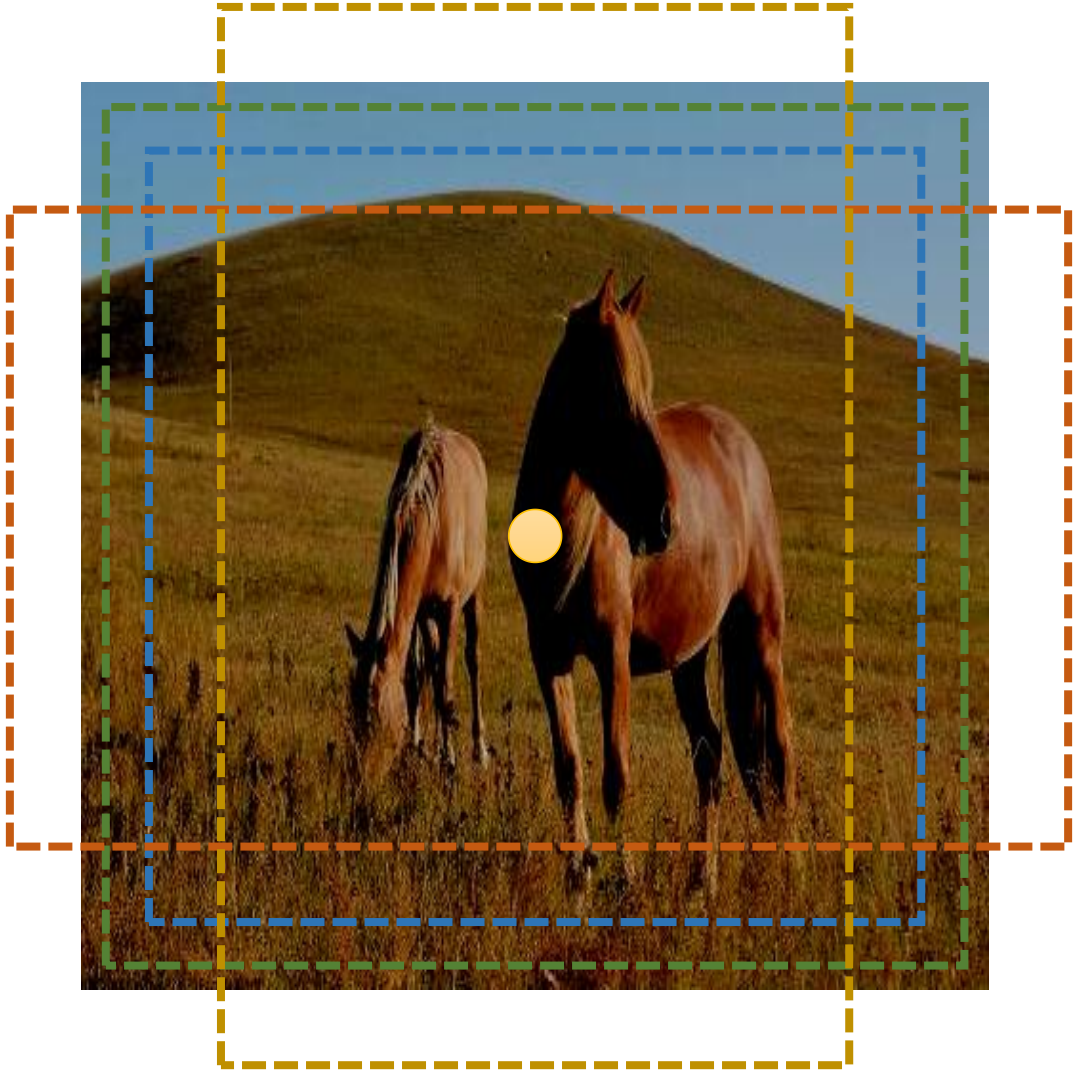
layer	s_k	s_k2
conv4-3	30	42
fc7	60	82
conv8-2	112	136
conv9-2	165	189
conv10-2	217	242
conv11-2	270	295

SSD

$$a=\left\{1,2,3,\frac{1}{2},\frac{1}{3}\right\} \qquad w_k^r = s_k \cdot \sqrt{a_r} \qquad h_k^r = s_k \Big/ \sqrt{a_r}$$

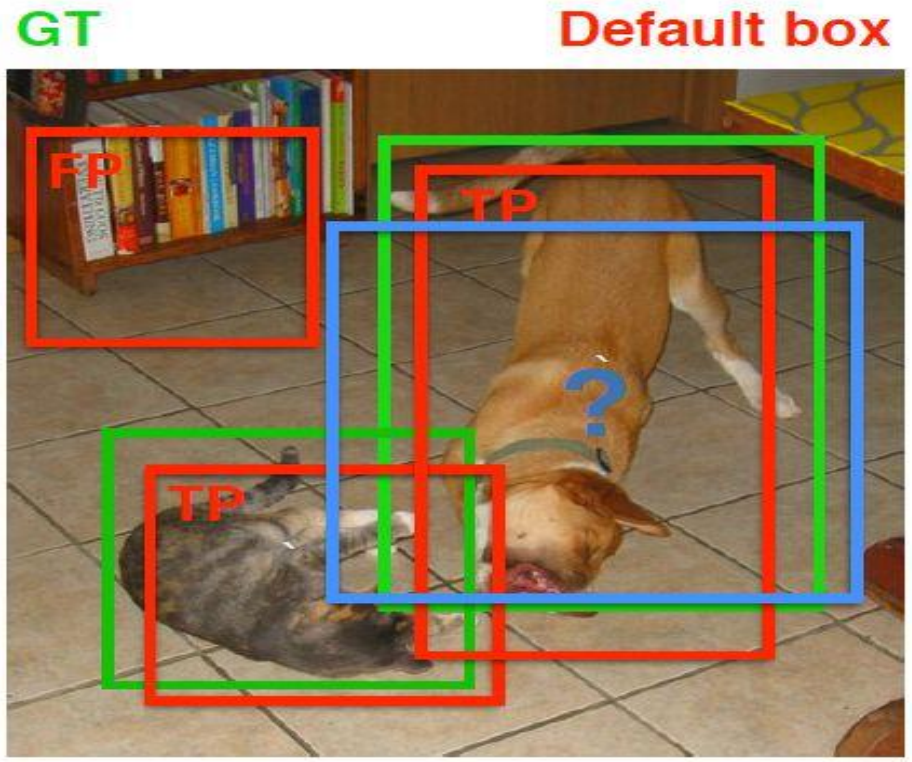
Layer Name	s_k,s_k2	Default Box
conv4-3	30,42	(30,30)、(42,42)、(42,21)、(21,42)
fc7	60,82	(60,60)、(82,82)、(84,42)、(105,35)、(42,82)、(35,105)
conv8-2	112,136	(112,112)、(136,136)、(158,79)、(195,65)、(79,158)、(65,195)
conv9-2	165,189	(165,165)、(189,189)、(232,116)、(285,95)、(116,232)、(95,285)
conv10-2	217,242	(217,217)、(242,242)、(306,153)、(153,306)
conv11-2	270,295	(270,270)、(295,295)、(380,190)、(190,380)

SSD



SSD

- Positive Boxes:
  - Ground Truth Boxes
  - **IoU > 0.5** default boxes(简单理解)
- Negative Boxes:
  - IoU <= 0.5 default boxes
  - **Hard Negative Mining**



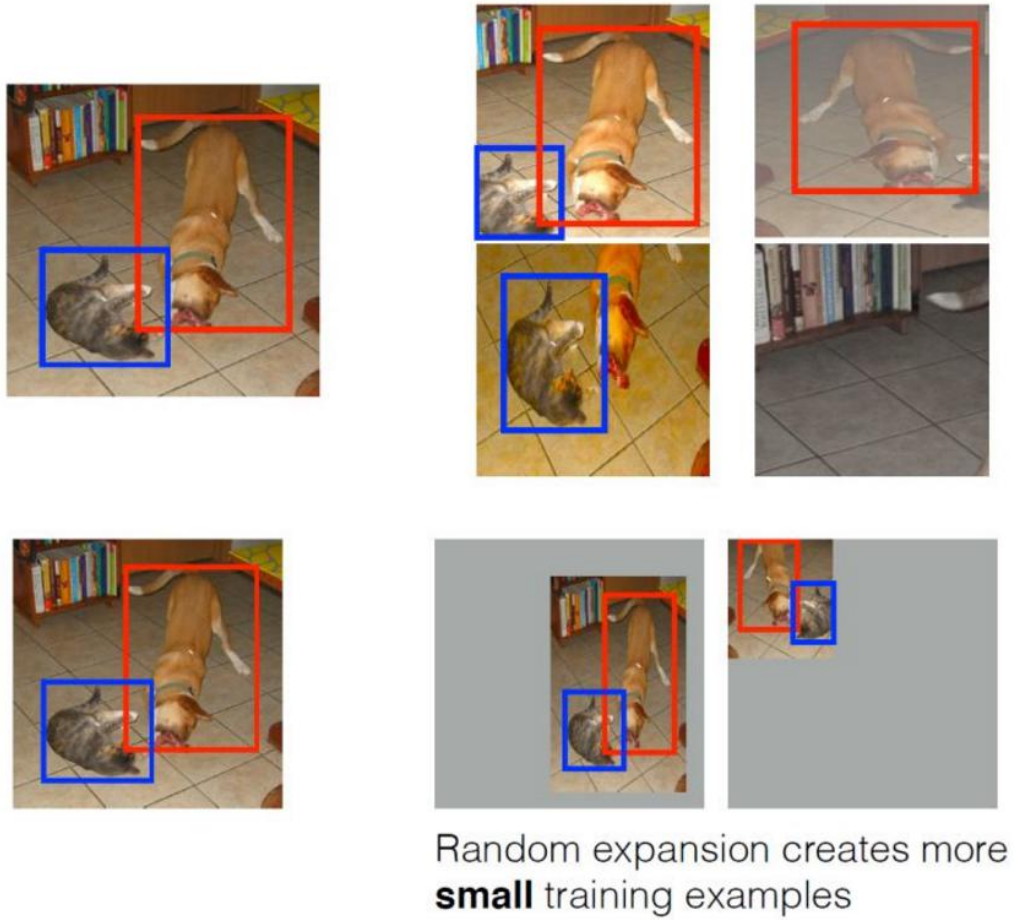
- NOTE: Positive Boxes:Negative Boxes = **1:3**

SSD

- Hard Negative Mining
  - 在生成先验框后，会产生很多符合Ground Truth Box的先验框，但是不符合的边框会更多，也就是negative boxes的数目远多于positive boxes的数目，也就会导致数据之间极度不均衡的情况出现，训练的时候比较难收敛。故在SSD中，采用R-CNN中介绍的**难负样本挖掘算法**对数据进行处理。将每个物体位置上对应的default boxes是negative的boxes按照前向loss的大小进行排序，获取loss比较大的N个negative boxes参与模型训练，最终保证正负样本比例在1:3左右。

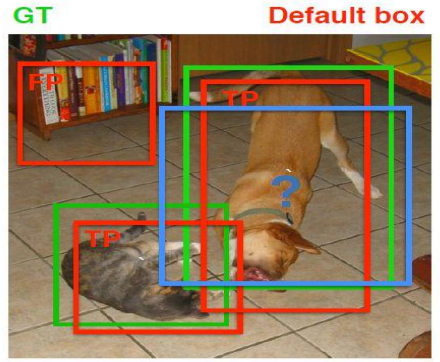
SSD

- Data augmentation(数据增强)
  - 水平翻转(Horizontal Flip)
  - 随机剪裁加颜色扭曲(Random Crop & Color Distortion)
  - 随机采集块域(Randomly sample a path)



Random expansion creates more **small** training examples

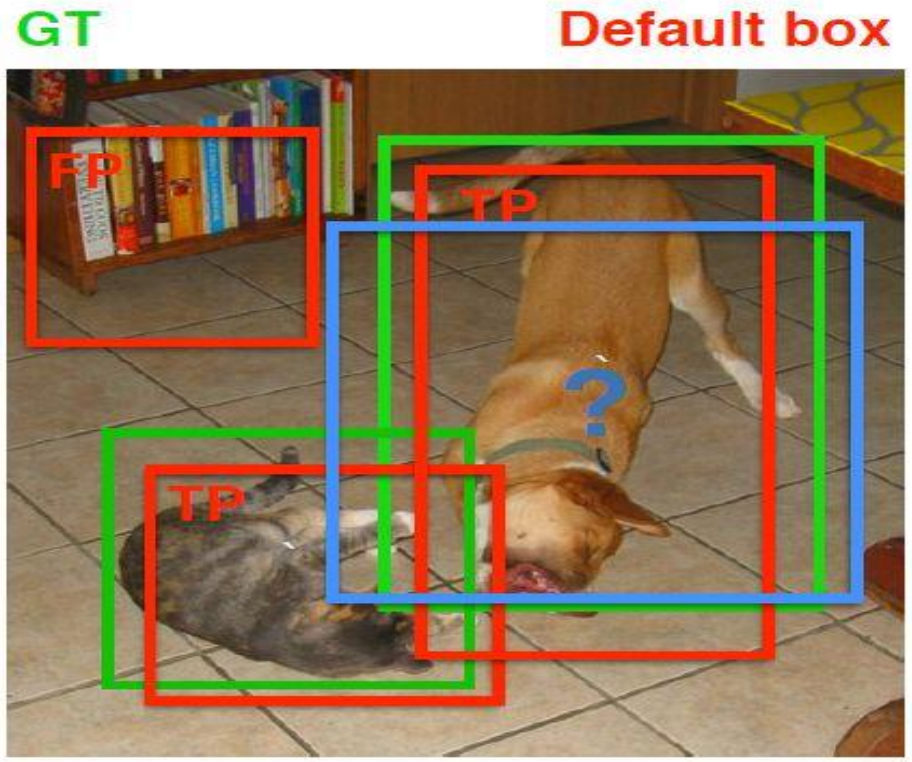
SSD



- 训练数据类别给定标准：
  - 正样本：若先验框和Ground Truth框匹配， 那么认为当前先验框为正样本；
  - 负样本：若先验框和所有Ground Truth框都不匹配，那么认为当前先验框为负样本。
  - NOTE: 采用hard negative mining(难负样本挖掘算法)选择loss大的样本作为负样本，正负样本比例1:3；
- SSD的先验框与Ground Truth的匹配原则主要有几点点。
  - 1. 对于图片中每个Ground Truth，找到与其IoU最大的先验框，该先验框与其匹配，
  - 2. 对于剩余的未匹配先验框，若其和某个Ground Truth的 IoU 大于某个阈值（一般是0.5），那么该先验框也与这个Ground Truth进行匹配。这意味着某个Ground Truth可能与多个先验框匹配，这是可以的。
  - 3. 如果某个先验框和多个Ground Truth的IoU值大于阈值或者是最大IoU的先验框，那么这个先验框仅和IoU最大的那个Ground Truth匹配。

SSD

- Positive Boxes:
  - Ground Truth Boxes
  - **Max IoU default boxes**
  - **IoU > 0.5** default boxes(简单理解)
- Negative Boxes:
  - IoU <= 0.5 default boxes
  - **Hard Negative Mining**
- NOTE: Positive Boxes:Negative Boxes = **1:3**



SSD

- 在SSD中，损失函数被定义为**位置误差**（locatization loss， loc）与**置信度误差**（confidence loss, conf）的加权和。

$$L(x,c,l,g)=\frac{1}{N}\left(L_{conf}(x,c)+\alpha L_{loc}(x,l,g)\right)$$

N为匹配的default boxes的数目(正样本)

α为位置误差的权重，用于控制loc和conf误差两者的比重

x为Default box和Ground Truth box的匹配情况
c为类别置信度预测值向量
l为Predicted offset box
g为Ground Truth offset box

L_conf为置信度误差

L_loc为位置误差

SSD

- x_i,j_p=1表示第i个default box和第j个ground truth box是匹配的，并且类别为p。

$$x_{i,j}^{p} = \{0,1\} \qquad \sum_{i} x_{i,j}^{p} \geq 1$$

- c_i_p表示第i个default box预测为类别p的置信度的值。

$$c_i^p = \left(c_i^0, c_i^1, \ldots . c_i^k\right) \qquad \hat{c}_i^p = \frac{\exp\left(c_i^p\right)}{\sum_p \exp\left(c_i^p\right)}$$

SSD

$$L_{conf}(x,c) = -\sum_{i \in Pos}^{N} x_{i,j}^{p} \log\left(\hat{c}_i^{p}\right) - \sum_{i \in Neg} \log\left(\hat{c}_i^{0}\right)$$

$$\hat{c}_i^{p} = \frac{\exp\left(c_i^{p}\right)}{\sum_{p} \exp\left(c_i^{p}\right)}$$

SSD

$$L_{loc}(x,l,g)=\sum_{i\in Pos}^{N}\left(\sum_{m\in\{cx,cy,w,h\}}x_{i,j}^{p}\cdot smooth_{L1}\left(l_i^m-\hat{g}_j^m\right)\right)$$

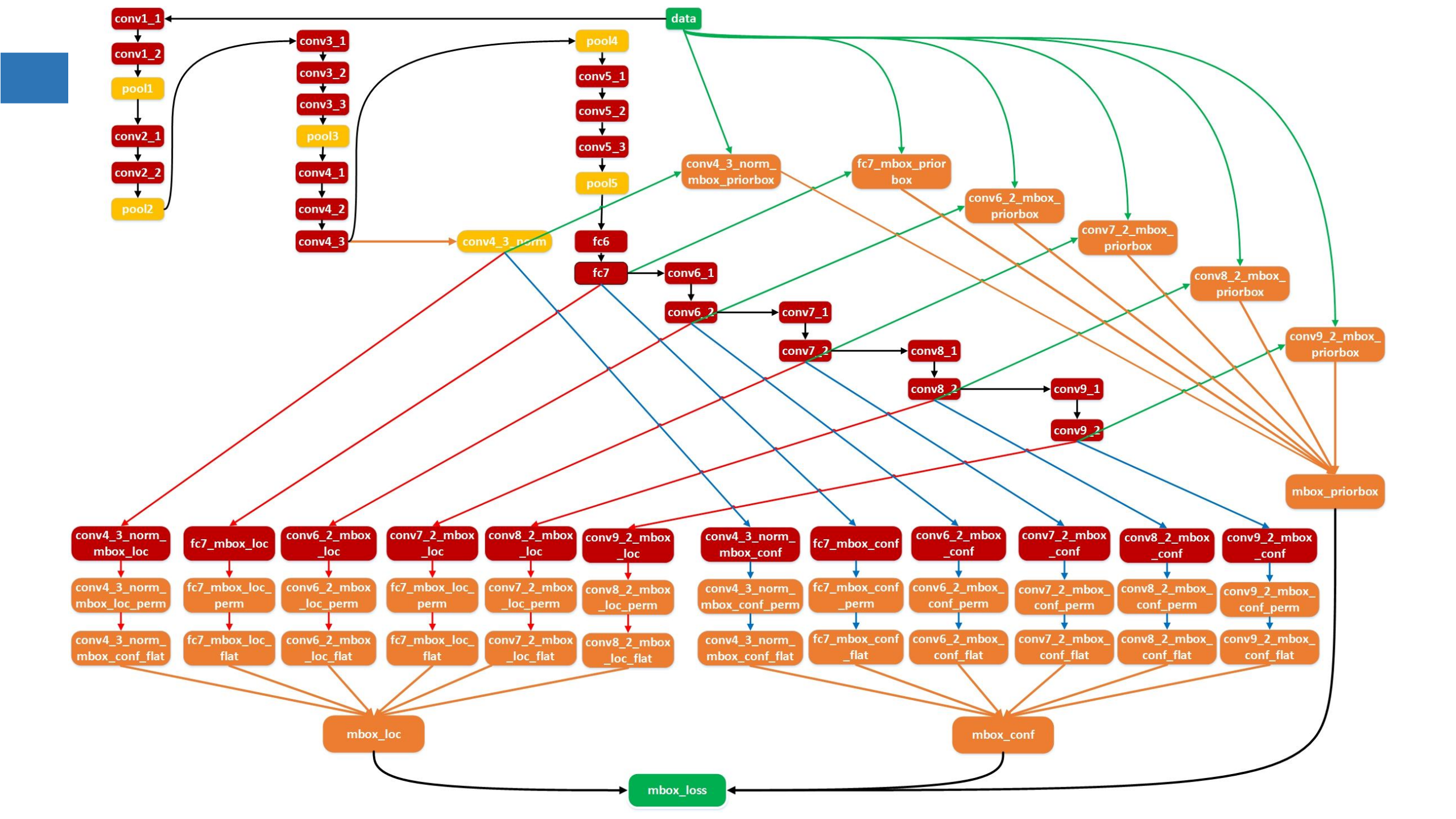
$$\hat{g}_j^{cx}=\left(g_j^{cx}-d_i^{cx}\right)/d_i^{w} \qquad \hat{g}_j^{cy}=\left(g_j^{cy}-d_i^{cy}\right)/d_i^{h}$$

$$\hat{g}_j^{w}=\log\left(\frac{g_j^{w}}{d_i^{w}}\right) \qquad \hat{g}_j^{h}=\log\left(\frac{g_j^{h}}{d_i^{h}}\right)$$

d为Default Box

SSD

- 预测过程比较简单，对于每个预测框，首先根据类别置信度确定其类别（置信度最大者）与置信度值，并过滤掉属于背景的预测框。然后根据置信度阈值（如0.5）过滤掉阈值较低的预测框。对于留下的预测框进行解码，根据Default Box先验框+offset box偏移量预测值做线性转换得到其真实的位置参数（解码后一般还需要做clip，防止预测框位置超出图片）。解码之后，一般需要根据置信度进行降序排列，然后仅保留top-k（如400）个预测框。最后就是进行NMS算法，过滤掉那些重叠度较大的预测框。最后剩余的预测框就是检测结果了。


conv1_1
conv1_2
pool1
conv2_1
conv2_2
pool2
conv3_1
conv3_2
conv3_3
pool3
conv4_1
conv4_2
conv4_3
conv4_3_norm
pool4
conv5_1
conv5_2
conv5_3
pool5
fc6
fc7
data
conv4_3_norm_mbox_priorbox
fc7_mbox_priorbox
conv6_2_mbox_priorbox
conv7_2_mbox_priorbox
conv8_2_mbox_priorbox
conv9_2_mbox_priorbox
conv6_1
conv6_2
conv7_1
conv7_2
conv8_1
conv8_2
conv9_1
conv9_2
mbox_priorbox
conv4_3_norm_mbox_loc
fc7_mbox_loc
conv6_2_mbox_loc
conv7_2_mbox_loc
conv8_2_mbox_loc
conv9_2_mbox_loc
conv4_3_norm_mbox_conf
fc7_mbox_conf
conv6_2_mbox_conf
conv7_2_mbox_conf
conv8_2_mbox_conf
conv9_2_mbox_conf
conv4_3_norm_mbox_loc_perm
fc7_mbox_loc_perm
conv6_2_mbox_loc_perm
conv7_2_mbox_loc_perm
conv8_2_mbox_loc_perm
conv4_3_norm_mbox_conf_perm
fc7_mbox_conf_perm
conv6_2_mbox_conf_perm
conv7_2_mbox_conf_perm
conv8_2_mbox_conf_perm
conv9_2_mbox_conf_perm
conv4_3_norm_mbox_conf_flat
fc7_mbox_loc_flat
conv6_2_mbox_loc_flat
conv7_2_mbox_loc_flat
conv8_2_mbox_loc_flat
conv4_3_norm_mbox_conf_flat
fc7_mbox_conf_flat
conv6_2_mbox_conf_flat
conv7_2_mbox_conf_flat
conv8_2_mbox_conf_flat
conv9_2_mbox_conf_flat
mbox_loc
mbox_conf
mbox_loss

SSD

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
Fast [6]	07+12	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
Faster [2]	07	69.9	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6
Faster [2]	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
Faster [2]	07+12+COCO	78.8	84.3	82.0	77.7	68.9	65.7	88.1	88.4	88.9	63.6	86.3	70.8	85.9	87.6	80.1	82.3	53.6	80.4	75.8	86.6	78.9
SSD300	07	68.0	73.4	77.5	64.1	59.0	38.9	75.2	80.8	78.5	46.0	67.8	69.2	76.6	82.1	77.0	72.5	41.2	64.2	69.1	78.0	68.5
SSD300	07+12	74.3	75.5	80.2	72.3	66.3	47.6	83.0	84.2	86.1	54.7	78.3	73.9	84.5	85.3	82.6	76.2	48.6	73.9	76.0	83.4	74.0
SSD300	07+12+COCO	79.6	80.9	86.3	79.0	**76.2**	57.6	87.3	88.2	88.6	60.5	85.4	**76.7**	**87.5**	**89.2**	84.5	81.4	55.0	81.9	**81.5**	85.9	78.9
SSD512	07	71.6	75.1	81.4	69.8	60.8	46.3	82.6	84.7	84.1	48.5	75.0	67.4	82.3	83.9	79.4	76.6	44.9	69.9	69.1	78.1	71.8
SSD512	07+12	76.8	82.4	84.7	78.4	73.8	53.2	86.2	87.5	86.0	57.8	83.1	70.2	84.9	85.2	83.9	79.7	50.3	77.9	73.9	82.5	75.3
SSD512	07+12+COCO	**81.6**	**86.6**	**88.3**	**82.4**	76.0	**66.3**	**88.6**	**88.9**	**89.1**	**65.1**	**88.4**	73.6	86.5	88.9	**85.3**	**84.6**	**59.1**	**85.0**	80.4	**87.4**	**81.2**

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	$\sim 1000 \times 600$
Fast YOLO	52.7	155	1	98	448×448
YOLO (VGG16)	66.4	21	1	98	448×448
SSD300	74.3	46	1	8732	300×300
SSD512	76.8	19	1	24564	512×512
SSD300	74.3	59	8	8732	300×300
SSD512	76.8	22	8	24564	512×512

	SSD300				
more data augmentation?		✔	✔	✔	✔
include $\{\frac{1}{2}, 2\}$ box?	✔		✔	✔	✔
include $\{\frac{1}{3}, 3\}$ box?	✔			✔	✔
use atrous?	✔	✔	✔		✔
VOC2007 `test` mAP	65.5	71.6	73.7	74.2	**74.3**