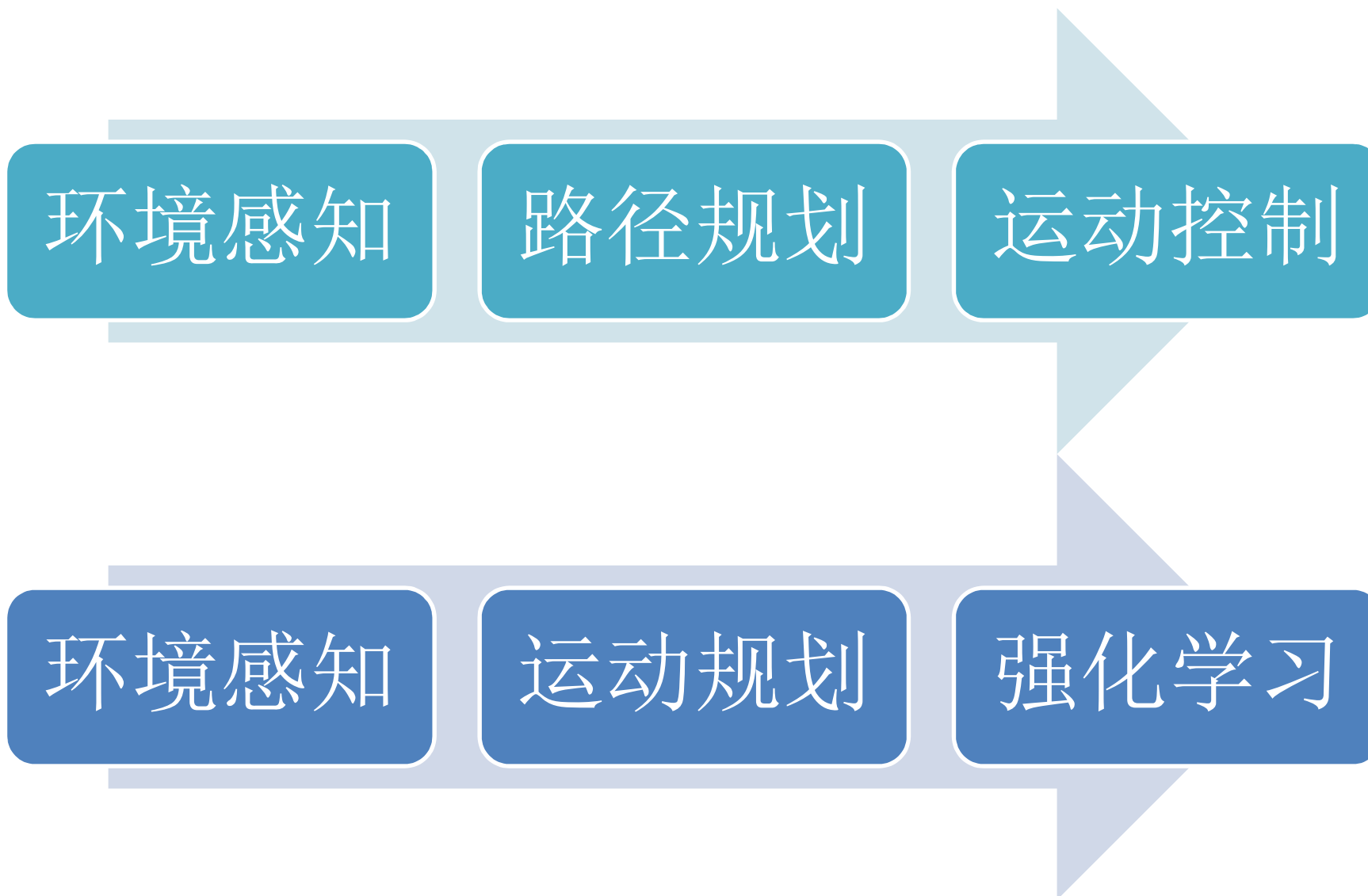


深度学习在智能车的应用

管林挺

智能驾驶的三个层级的转变



智能驾驶的三个层级

- 环境感知
- 运动规划
- 强化学习

环境感知



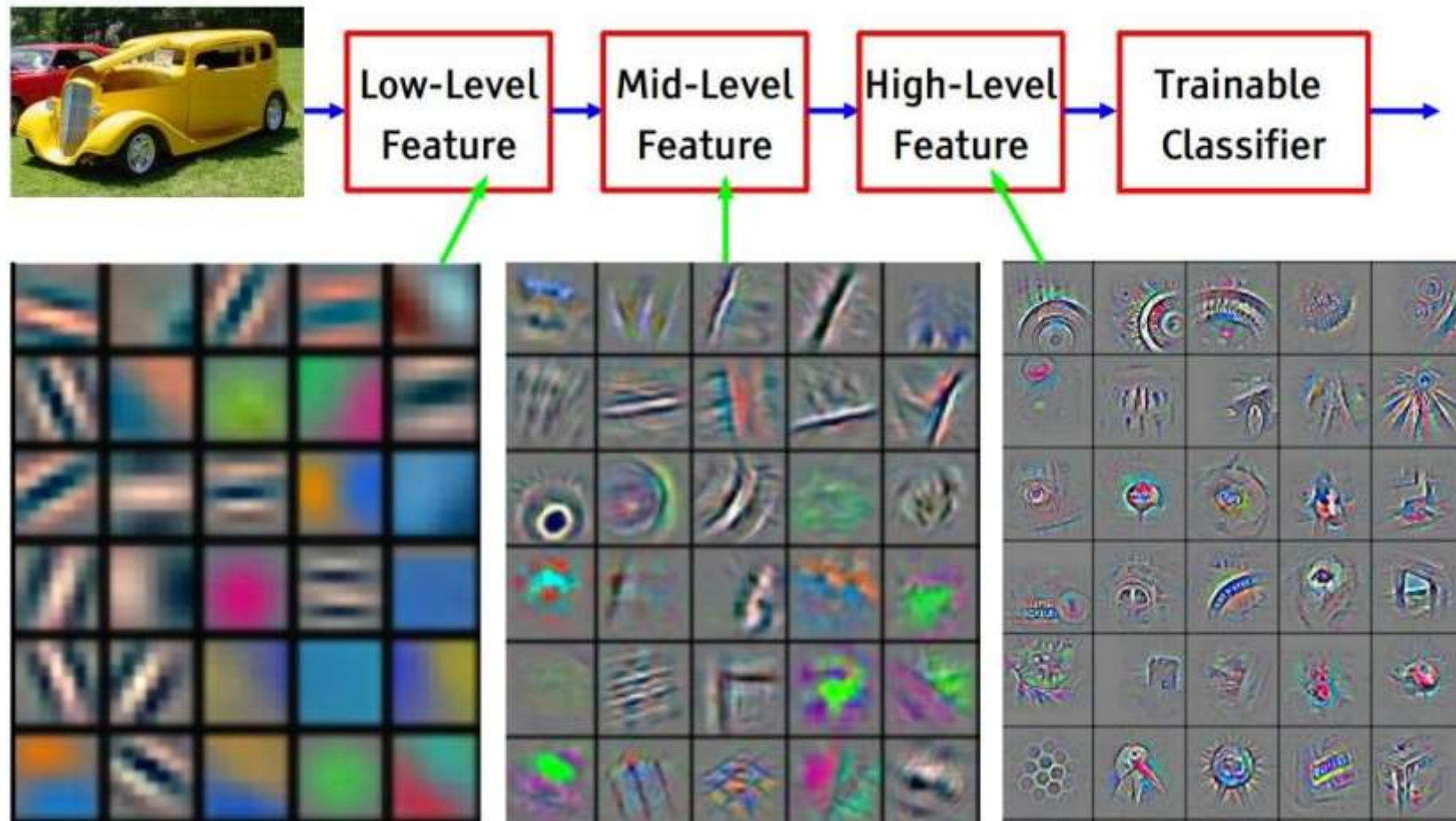
传感器主要包括摄像头、激光雷达、毫米波雷达、光电编码器、GPS和惯性测量单元等

目标检测

- 基于图像目标检测是从图像序列中检测与识别目标对象。在智能驾驶场景中，对交通目标进行自动检测和识别是一个特别重要的任务。

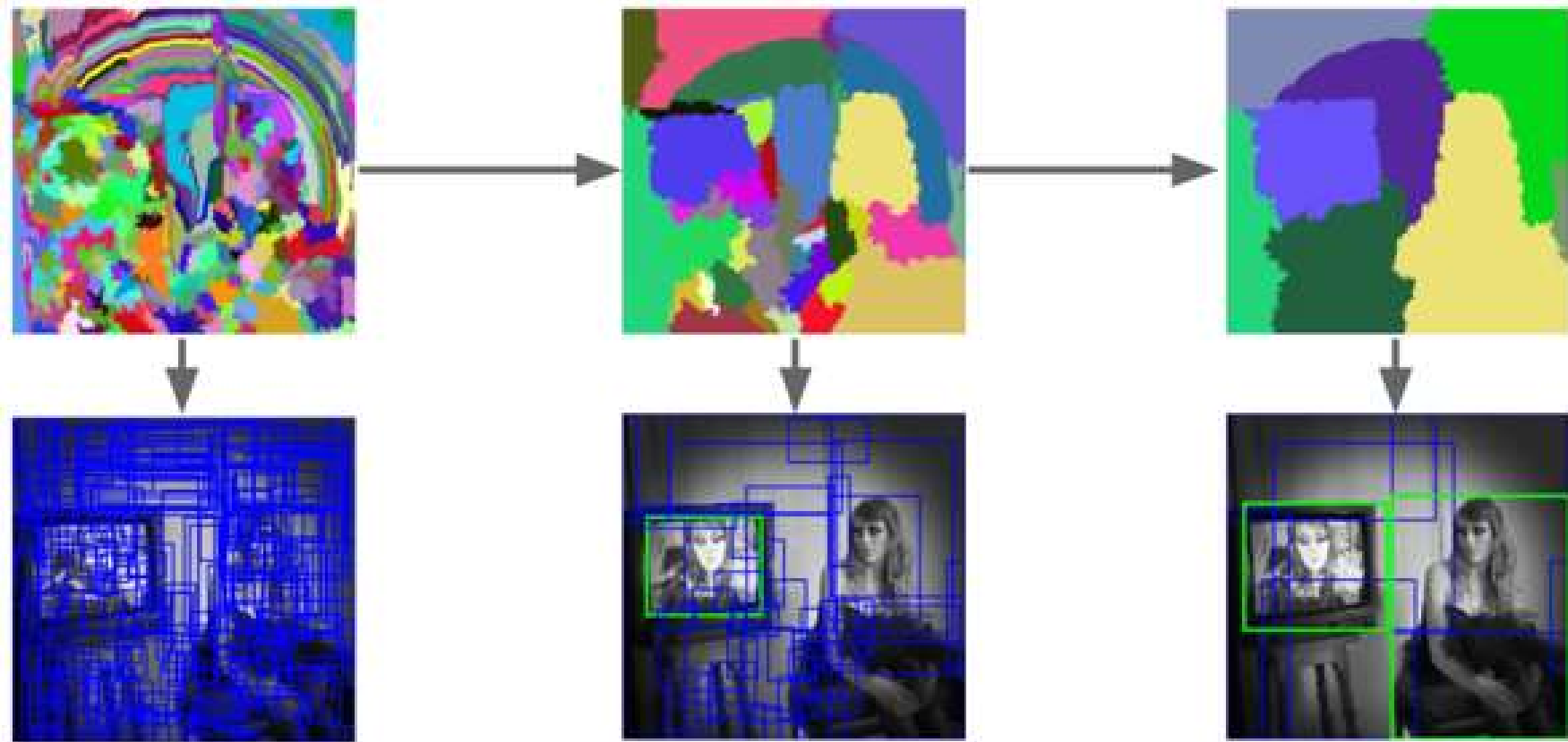


CNN features



Feature visualization of convolutional net trained on ImageNet [Zeiler & Fergus 2013]

region proposal



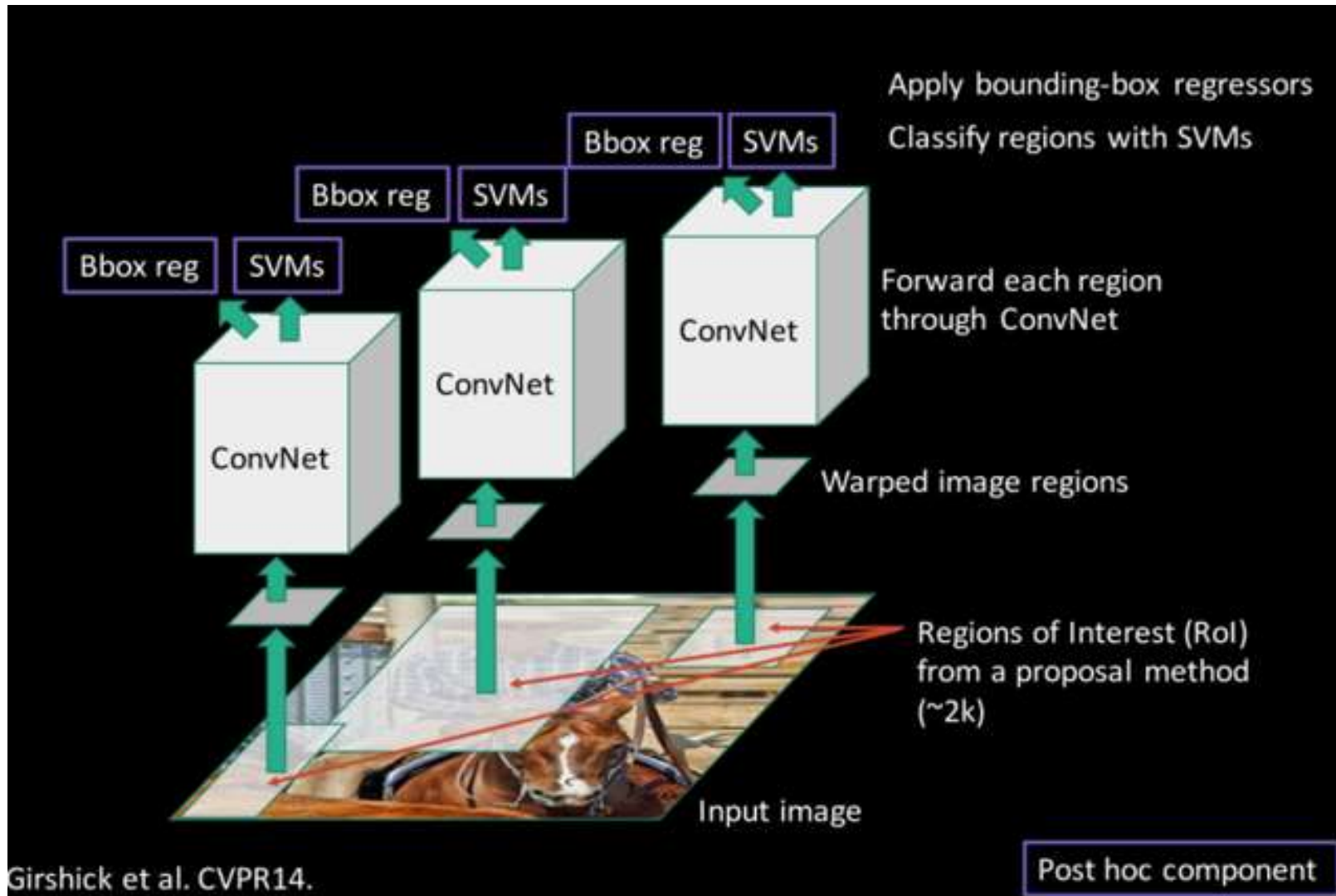
Uijlings et al, Selective Search for Object Recognition IJCV 2013

region proposals

Method	Approach	Outputs Segments	Outputs Score	Control #proposals	Time (sec.)	Repea- tability	Recall Results	Detection Results
Bing [18]	Window scoring		✓	✓	0.2	***	*	.
CPMC [19]	Grouping	✓	✓	✓	250	-	**	*
EdgeBoxes [20]	Window scoring		✓	✓	0.3	**	***	***
Endres [21]	Grouping	✓	✓	✓	100	-	***	**
Geodesic [22]	Grouping	✓		✓	1	*	***	**
MCG [23]	Grouping	✓	✓	✓	30	*	***	***
Objectness [24]	Window scoring		✓	✓	3	.	*	.
Rahtu [25]	Window scoring		✓	✓	3	.	.	*
RandomizedPrim's [26]	Grouping	✓		✓	1	*	*	**
Rantalankila [27]	Grouping	✓		✓	10	**	.	**
Rigor [28]	Grouping	✓		✓	10	*	**	**
SelectiveSearch [29]	Grouping	✓	✓	✓	10	**	***	***
Gaussian				✓	0	.	.	*
SlidingWindow				✓	0	***	.	.
Superpixels		✓			1	*	.	.
Uniform				✓	0	.	.	.

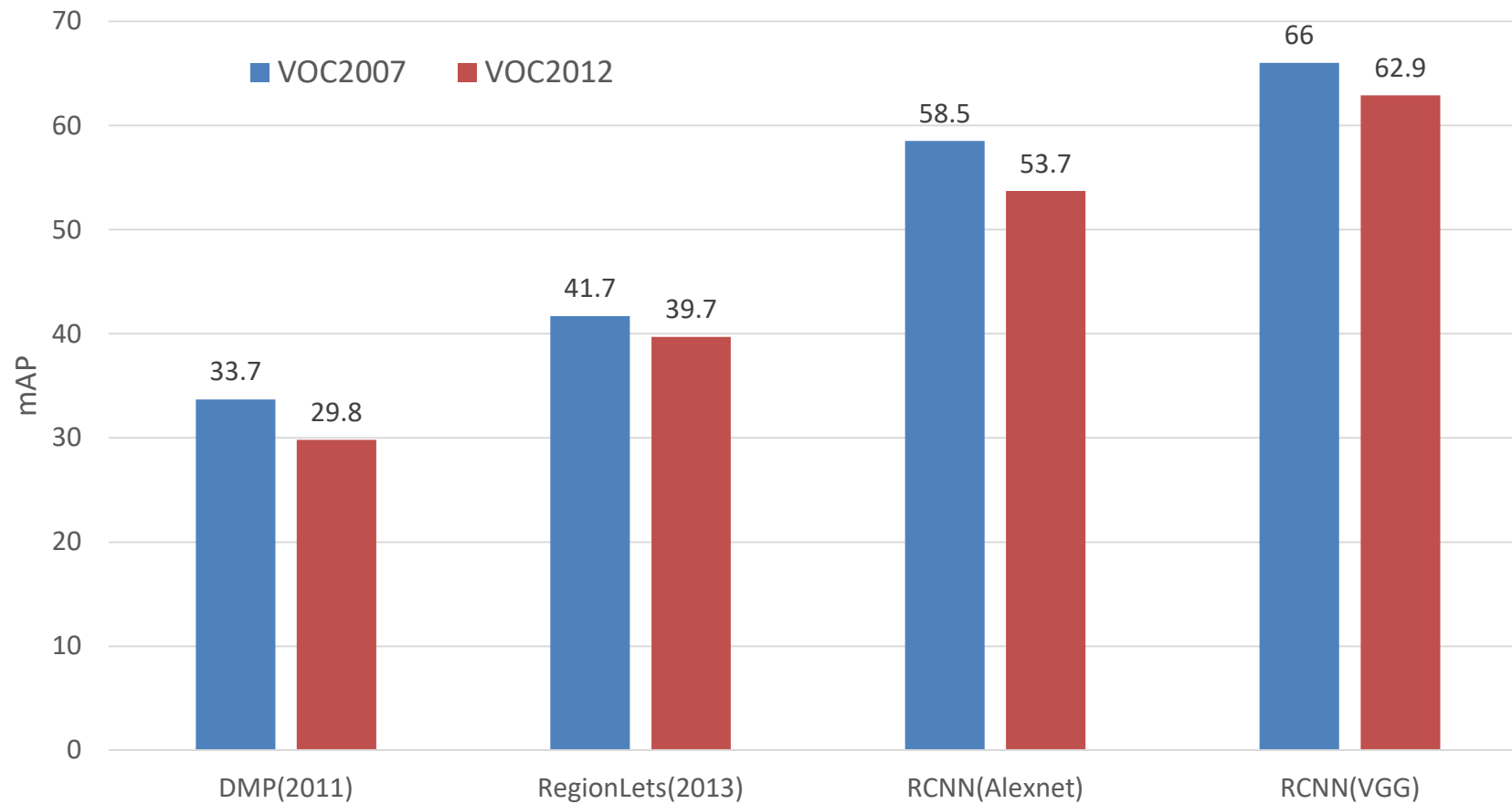
Hosang et al, What makes for effective detection proposals?, PAMI 2015

R-CNN

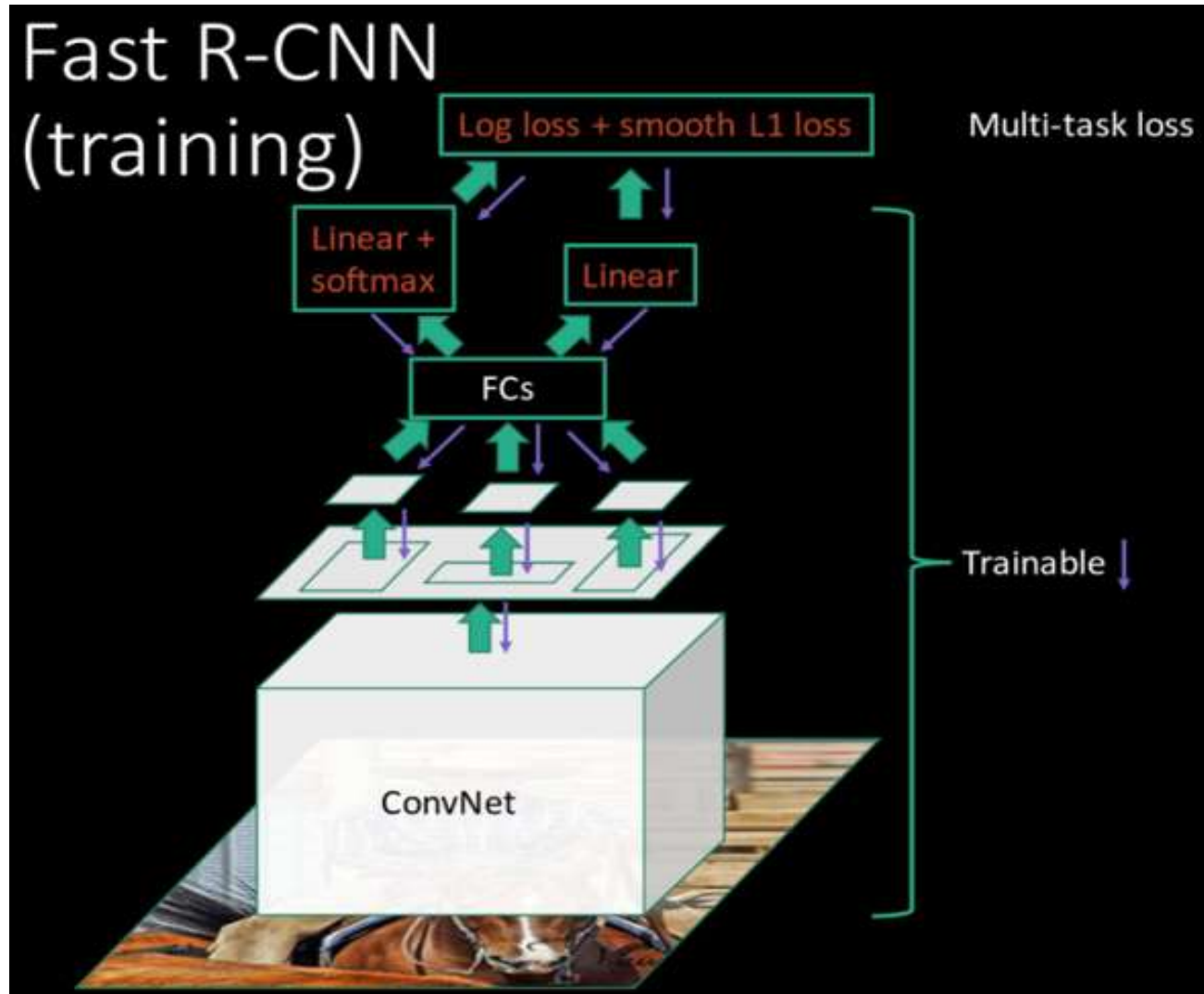


Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014

result



Fast R-CNN

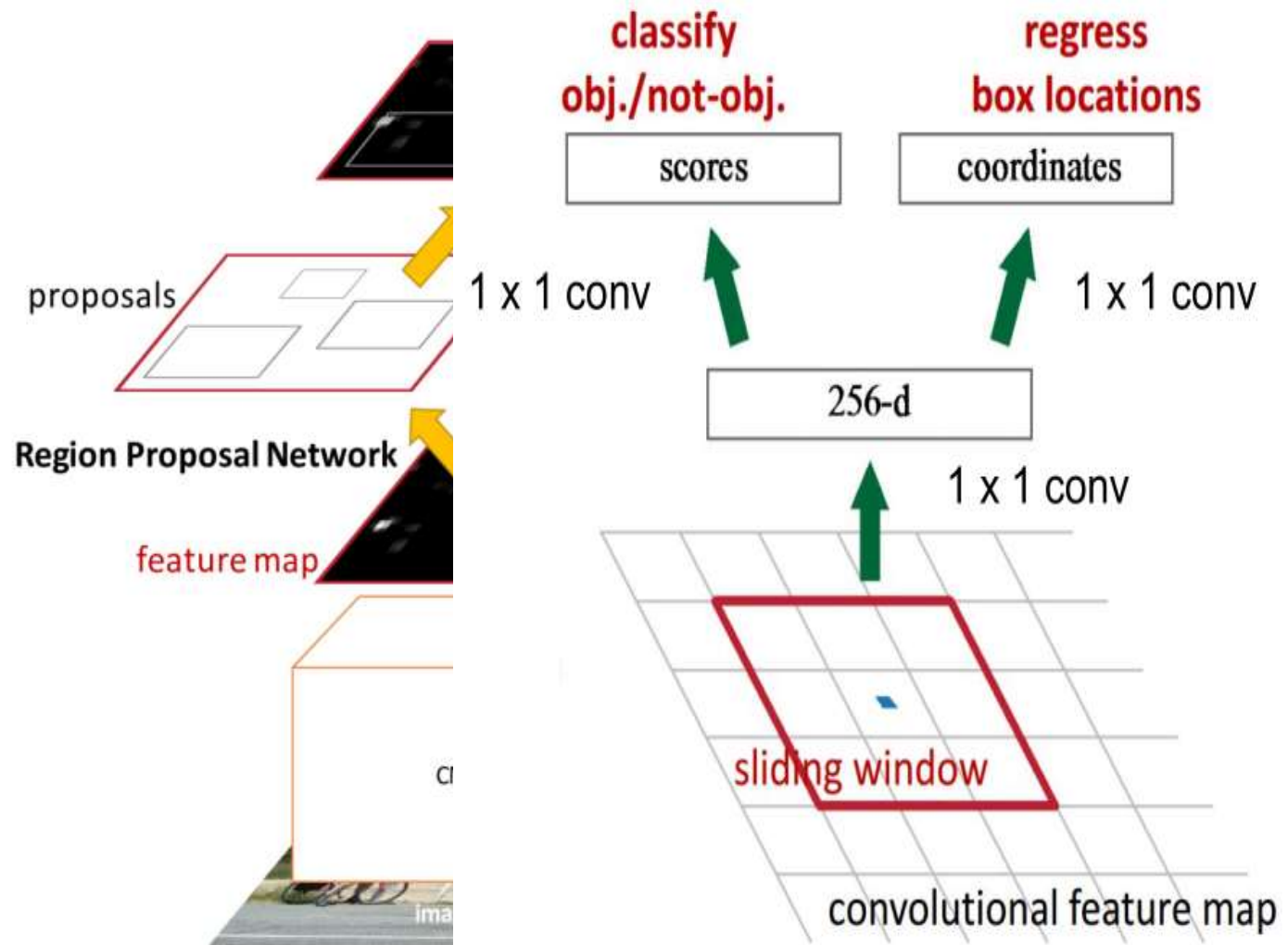


Girschick, "Fast R-CNN", ICCV 2015

Fast R-CNN

	R-CNN	Fast R-CNN
Train time(hours)	84	9.5
Test time (sec pre image)	47+2	0.32+2
mAP(VOC 2007)	66.0	66.9

Faster R-CNN



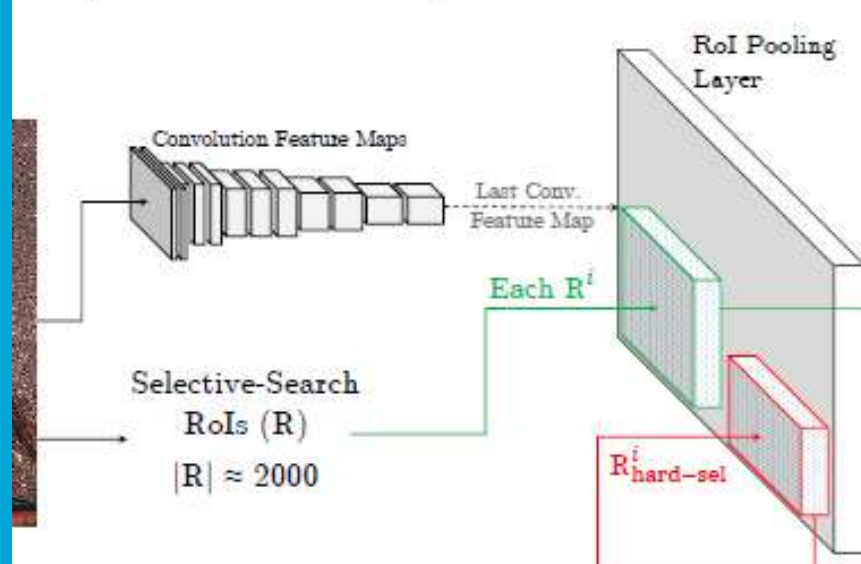
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Faster R-CNN

	R-CNN	Fast R-CNN	Faster R-CNN
Time (sec pre image)	47+2	0.32+2	0.2
mAP(VOC 2007)	66.0	66.9	66.9

OHEM

Convolutional Network

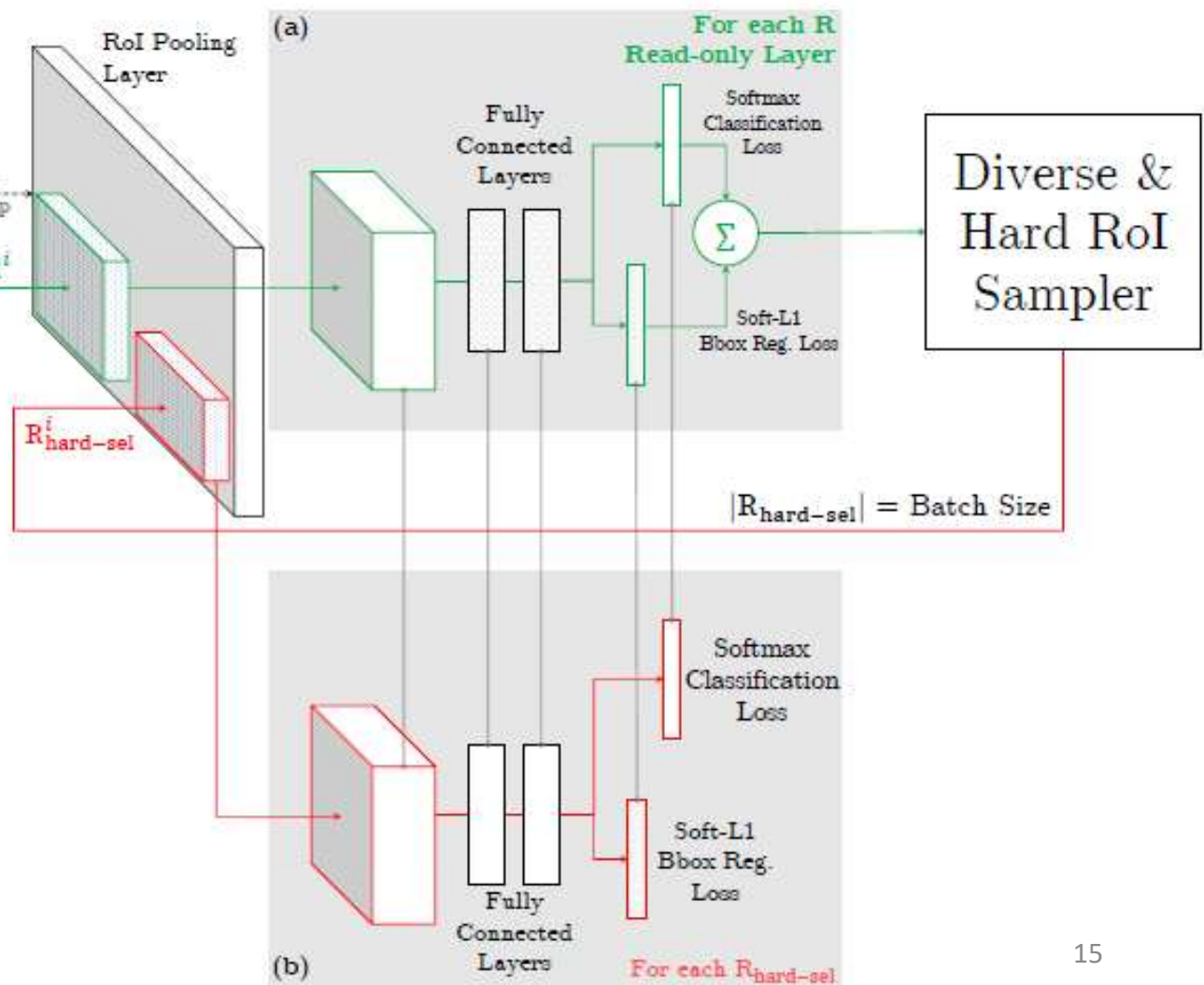


Backward Computation for:

1. Each $R_{hard-sel}^i$ (—→)
2. Gradient Accumulation by RoI Pooling Layer
3. Conv. Network (-----→)

Backward each R^i
Backward for each Image

RoI Network



Faster R-CNN

	R-CNN	Fast R-CNN	Faster R-CNN	OHEM
Time (sec pre image)	47+2	0.32+2	0.2	0.2
mAP(VOC 2007)	66.0	66.9	66.9	69.9

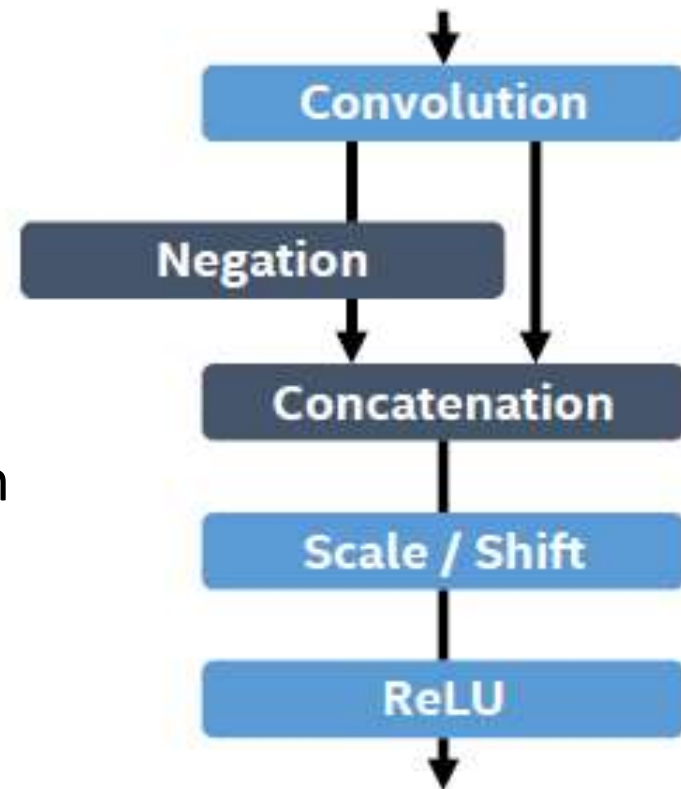
PVANET

- Less channels with more layers
- Concatenated Relu
- Inception
- HyperNet

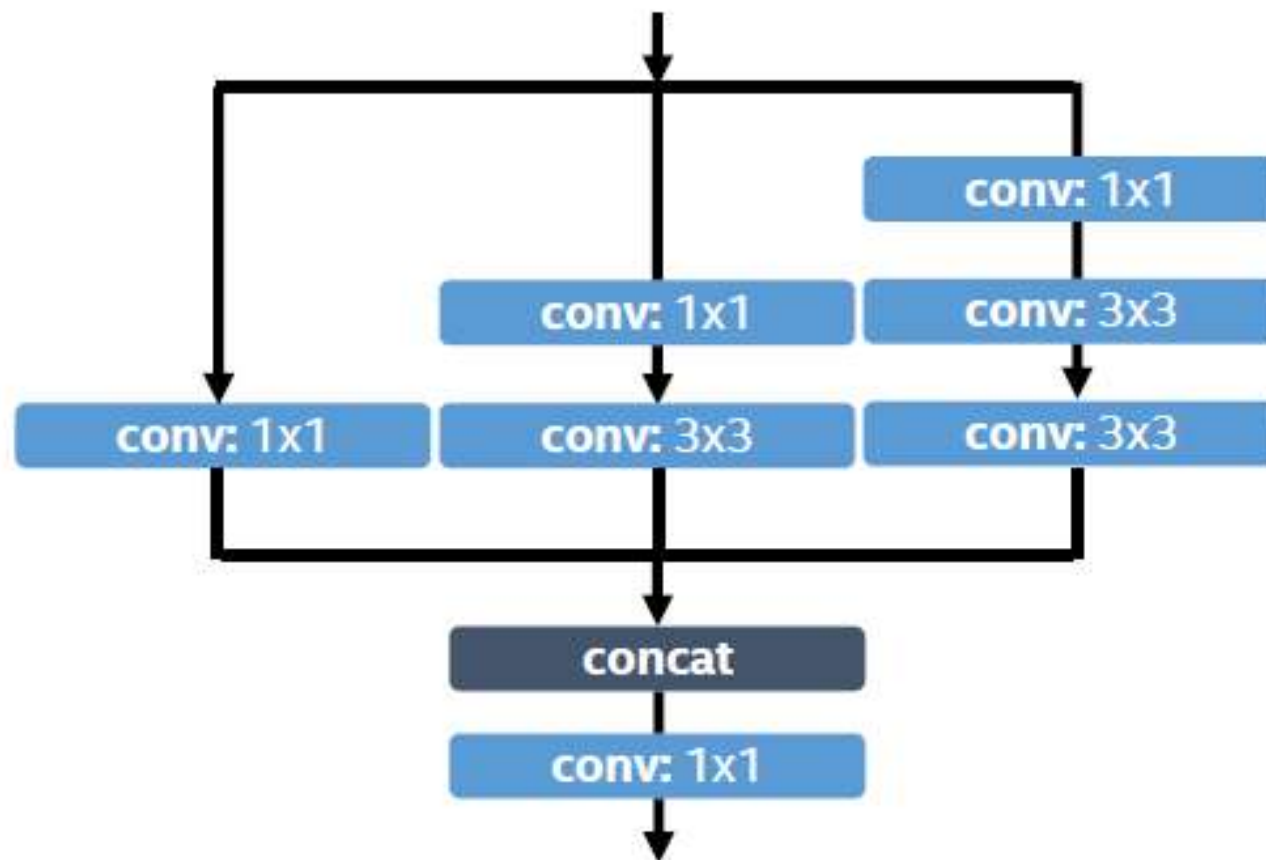
Concatenated Relu

Apply to early stage of CNNs

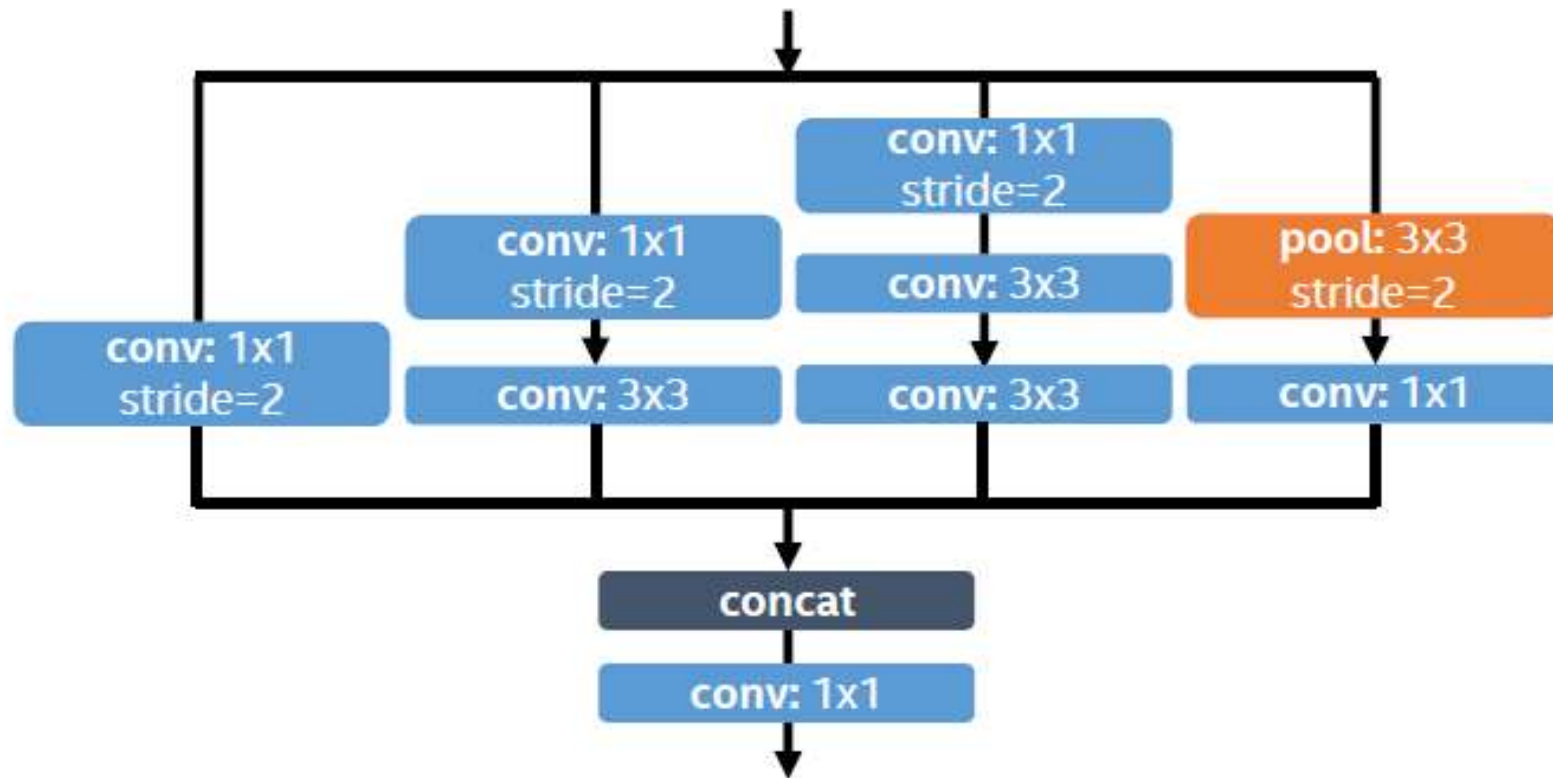
In early stage, one node's activation is the opposite side of another's



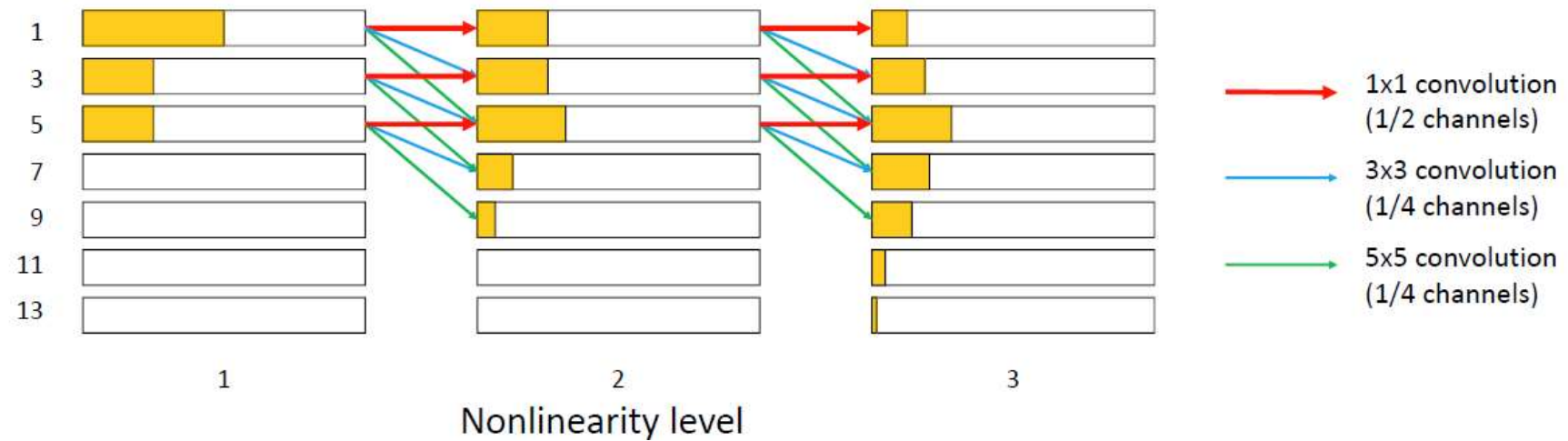
Inception



Inception for reducing feature map size by half



Receptive field size



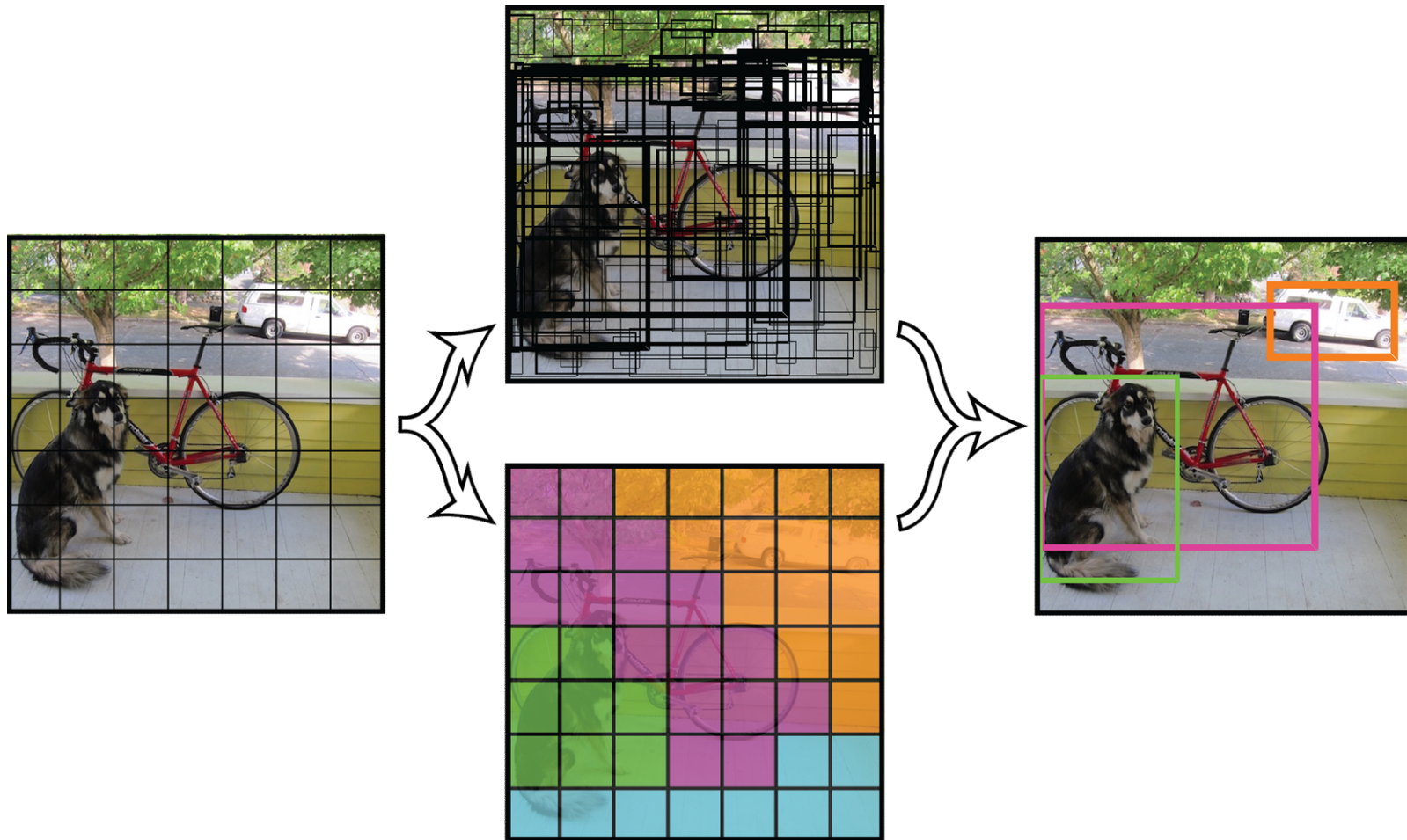
detailed structure of PVANET

Name	Type	Stride	Output size	Residual	C.ReLU #1x1-KxK-1x1	#1x1	#3x3	Inception #5x5	#pool	#out
conv1_1	7x7 C.ReLU	2	528x320x32		X-16-X					
pool1_1	3x3 max-pool	2	264x160x32							
conv2_1	3x3 C.ReLU		264x160x64	O	24-24-64					
conv2_2	3x3 C.ReLU		264x160x64	O	24-24-64					
conv2_3	3x3 C.ReLU		264x160x64	O	24-24-64					
conv3_1	3x3 C.ReLU	2	132x80x128	O	48-48-128					
conv3_2	3x3 C.ReLU		132x80x128	O	48-48-128					
conv3_3	3x3 C.ReLU		132x80x128	O	48-48-128					
conv3_4	3x3 C.ReLU		132x80x128	O	48-48-128					
conv4_1	Inception	2	66x40x256	O		64	48-128	24-48-48	128	256
conv4_2	Inception		66x40x256	O		64	64-128	24-48-48		256
conv4_3	Inception		66x40x256	O		64	64-128	24-48-48		256
conv4_4	Inception		66x40x256	O		64	64-128	24-48-48		256
conv5_1	Inception	2	33x20x384	O		64	96-192	32-64-64	128	384
conv5_2	Inception		33x20x384	O		64	96-192	32-64-64		384
conv5_3	Inception		33x20x384	O		64	96-192	32-64-64		384
conv5_4	Inception		33x20x384	O		64	96-192	32-64-64		384
downscale	3x3 max-pool	2	66x40x128							
upscale	4x4 deconv	2	66x40x384							
concat	concat		66x40x768							
convf	1x1 conv		66x40x512							
Total										

Experimental results

Model	Computation cost (MAC)				Running time		mAP (%)
	Shared CNN	RPN	Classifier	Total	ms	x(PVANET)	
PVANET+	7.9	1.3	27.7	37.0	46	1.0	82.5
Faster R-CNN + ResNet-101	80.5	N/A	219.6	300.1	2240	48.6	83.8
Faster R-CNN + VGG-16	183.2	5.5	27.7	216.4	110	2.4	75.9
R-FCN + ResNet-101	122.9	0	0	122.9	133	2.9	82.0

YOLO

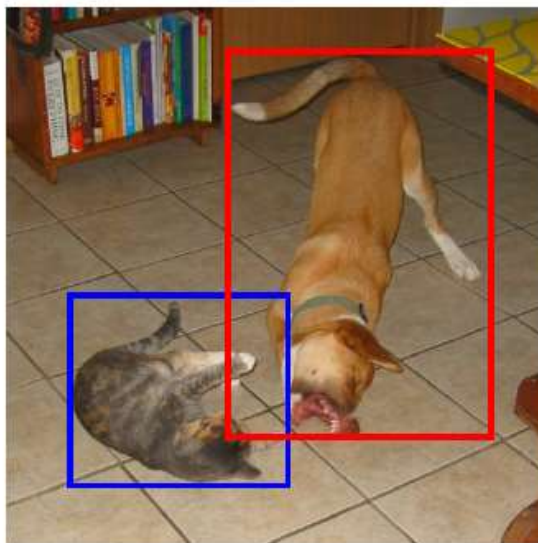


Redmon et al, You Only Look Once: Unified, Real-Time Object Detection, CVPR 2016

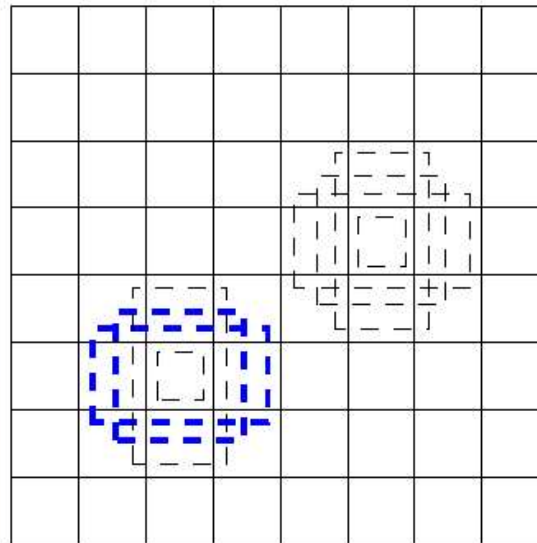
YOLO

	YOLO	Fast R-CNN	Faster R-CNN
Time (sec pre image)	0.022	0.32+2	0.2
mAP(VOC 2007+2012)	63.4	70	73.2

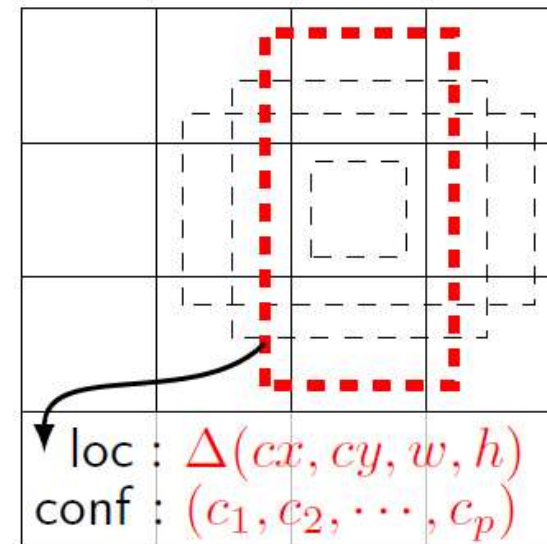
SSD: Single Shot MultiBox Detector



(a) Image with GT boxes

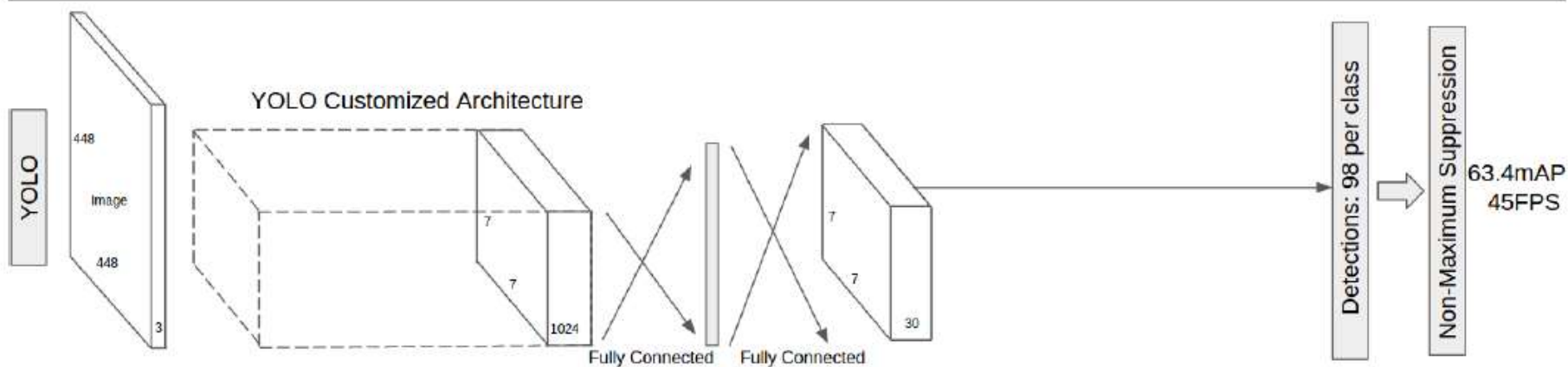
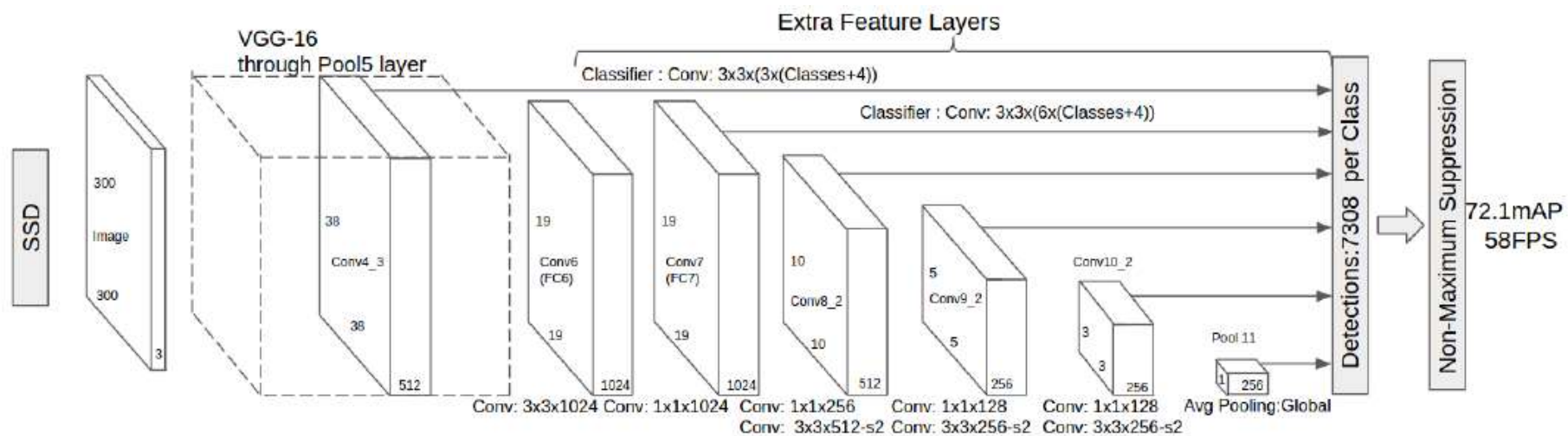


(b) 8×8 feature map



(c) 4×4 feature map

SSD and YOLO



result

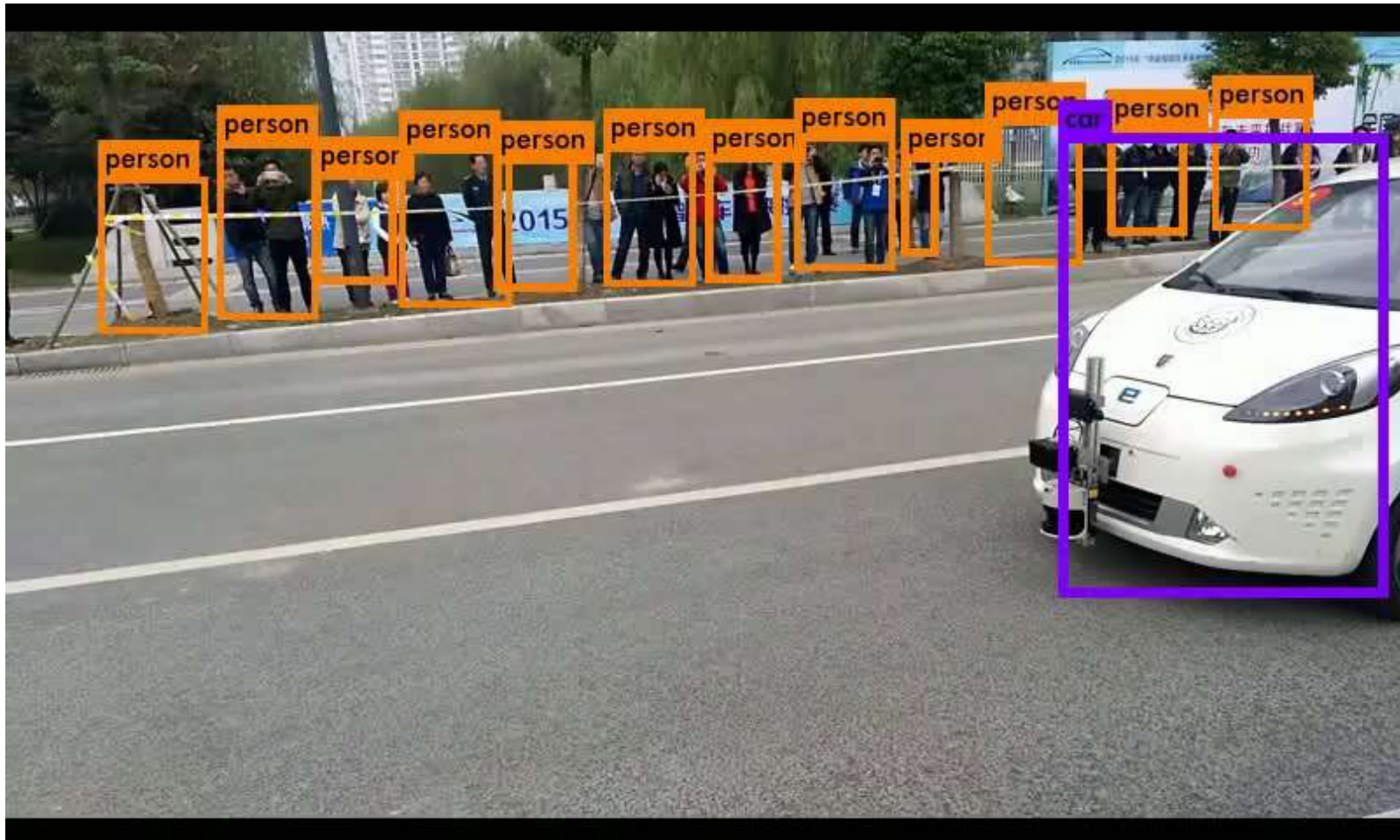
Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
Faster [2]	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
SSD300	72.1	75.2	79.8	70.5	62.5	41.3	81.1	80.8	86.4	51.5	74.3	72.3	83.5	84.6	80.6	74.5	46.0	71.4	73.8	83.0	69.1
SSD500	75.1	79.8	79.5	74.5	63.4	51.9	84.9	85.6	87.2	56.6	80.1	70.0	85.4	84.9	80.9	78.2	49.0	78.4	72.4	84.6	75.5

Table 1: **PASCAL VOC2007 test detection results.** Both Fast and Faster R-CNN use input images whose minimum dimension is 600. The two SSD models have exactly the same settings except that they have different input sizes (300×300 vs. 500×500). It is obvious that larger input size leads to better results.

Run time

Method	<i>mAP</i>	FPS	# Boxes
Faster R-CNN [2](VGG16)	73.2	7	300
Faster R-CNN [2](ZF)	62.1	17	300
YOLO [5]	63.4	45	98
Fast YOLO [5]	52.7	155	98
SSD300	72.1	58	7308
SSD500	75.1	23	20097

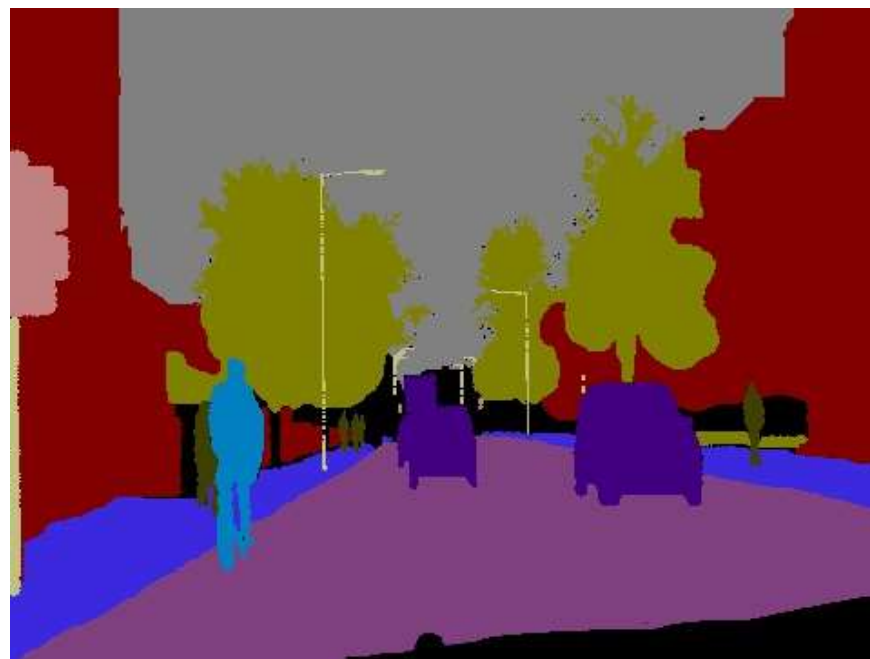
demo(yolo)



从目标检测到语义分割

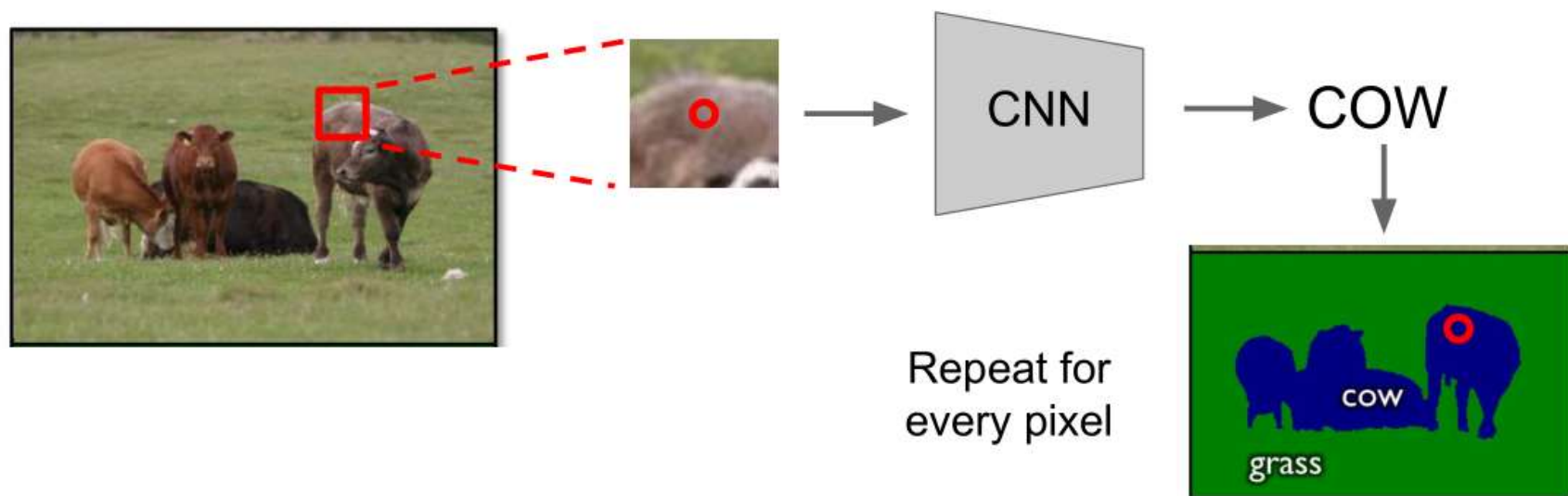
- 目标检测的是粗略对象信息，如行人、车辆等的矩形位置，但是并不能提供详细的细致路面信息，如哪里是车道、哪里是人行道及哪里是绿化带等。
- 语义分割任务对图像做非常细致（像素级）的语义分割信息，为智能驾驶提供充分的道路信息。

语义分割



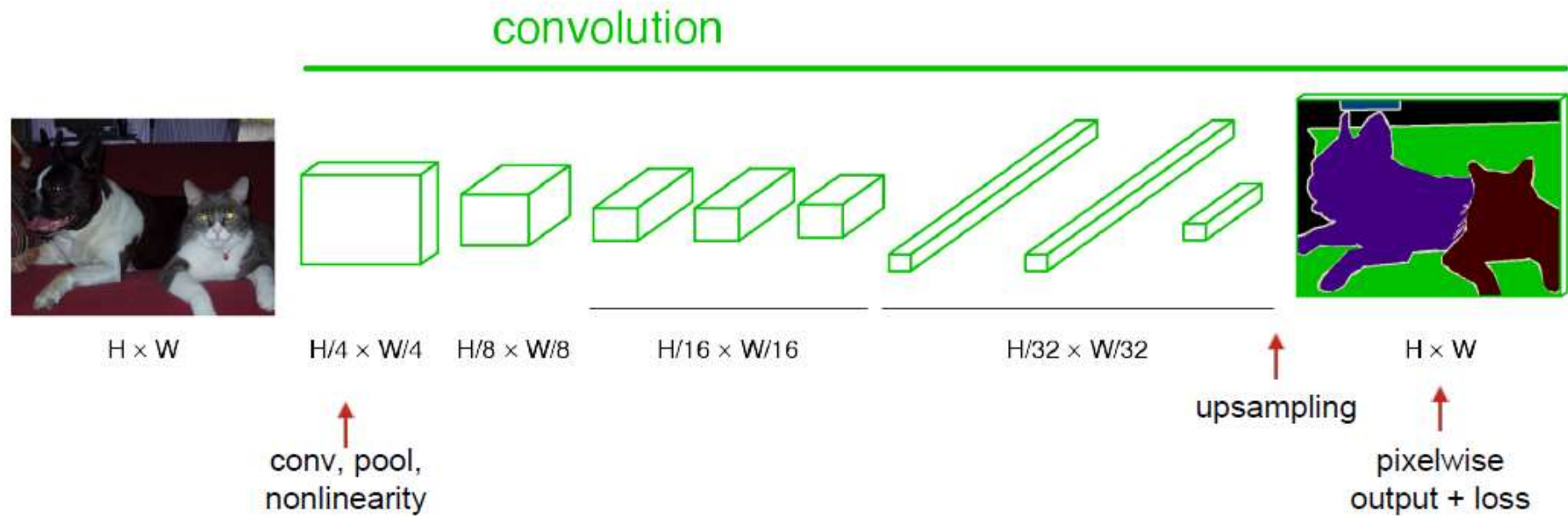
图像的语义分割是要得到图像中对应位置每个像素的分类结果

经典的语义分割方法



缺点：耗时，无法建模大的上下文信息

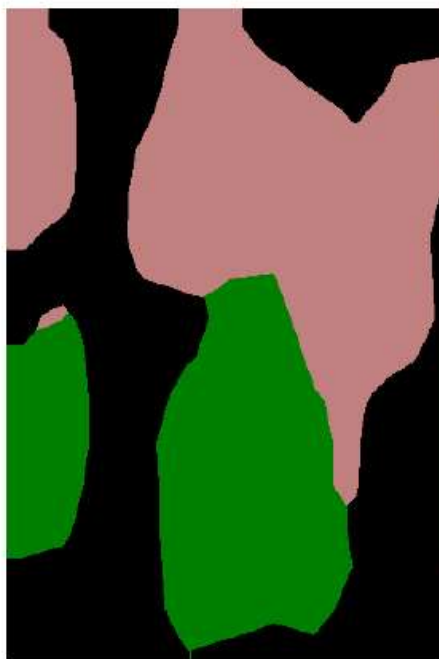
全卷积网络



Long, Shelhamer, and Darrell,
Fully Convolutional Networks for Semantic Segmentation, CVPR 2015

多层特征融合

FCN-32s



FCN-16s



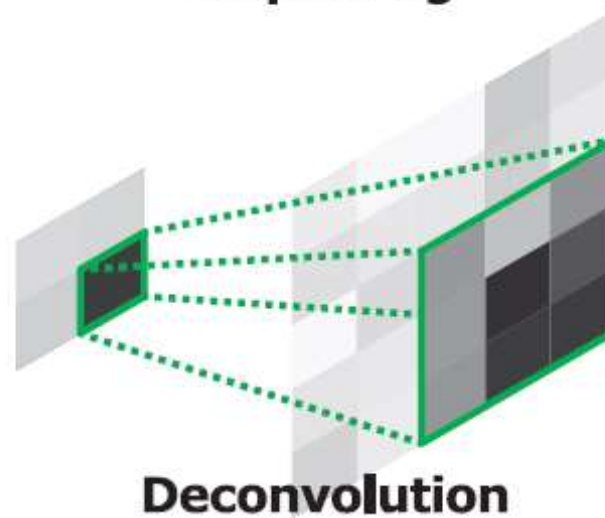
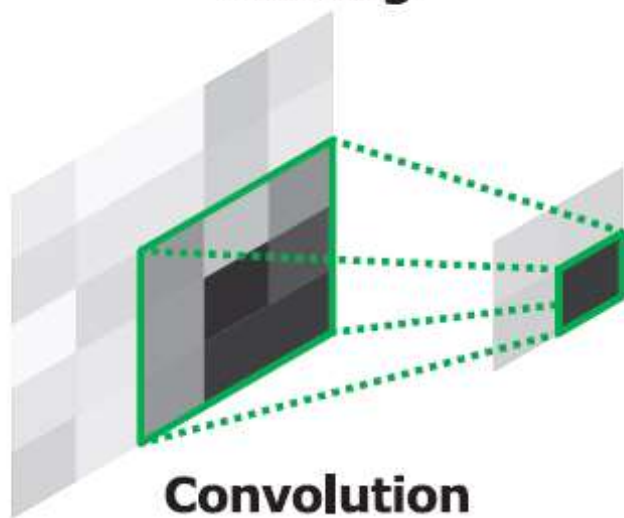
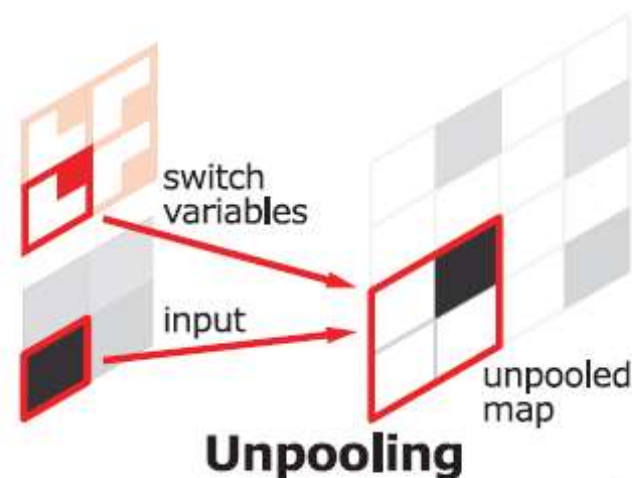
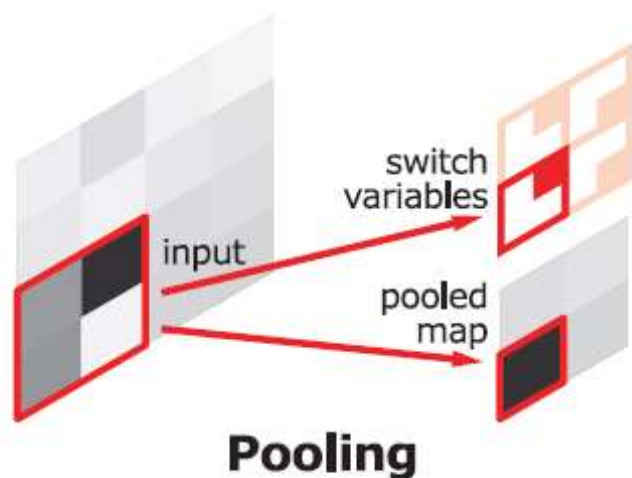
FCN-8s



Ground truth

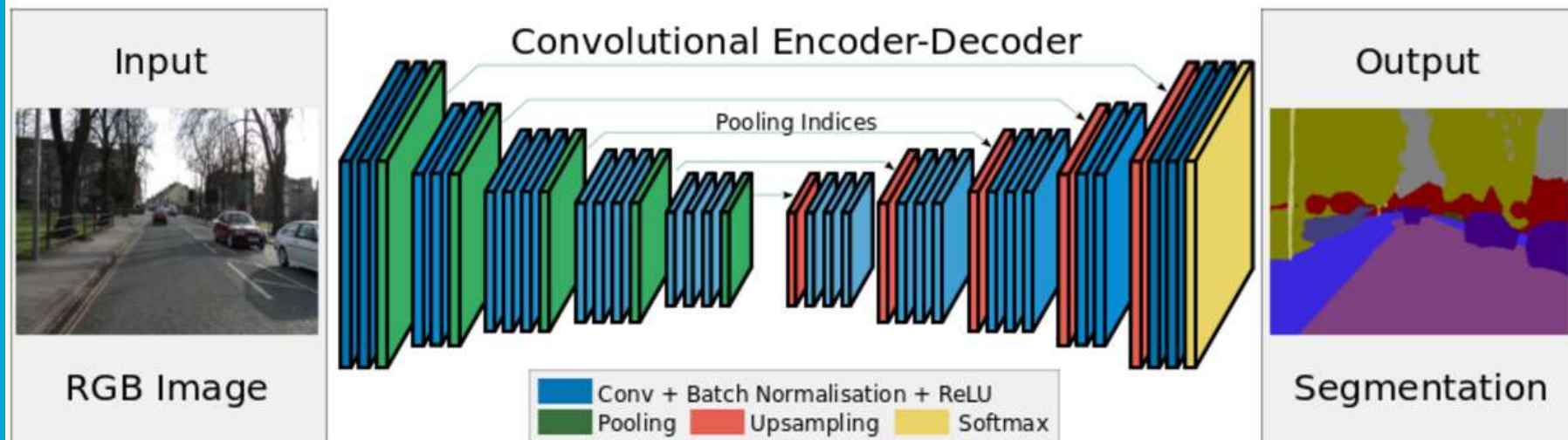


图片放大的不同策略



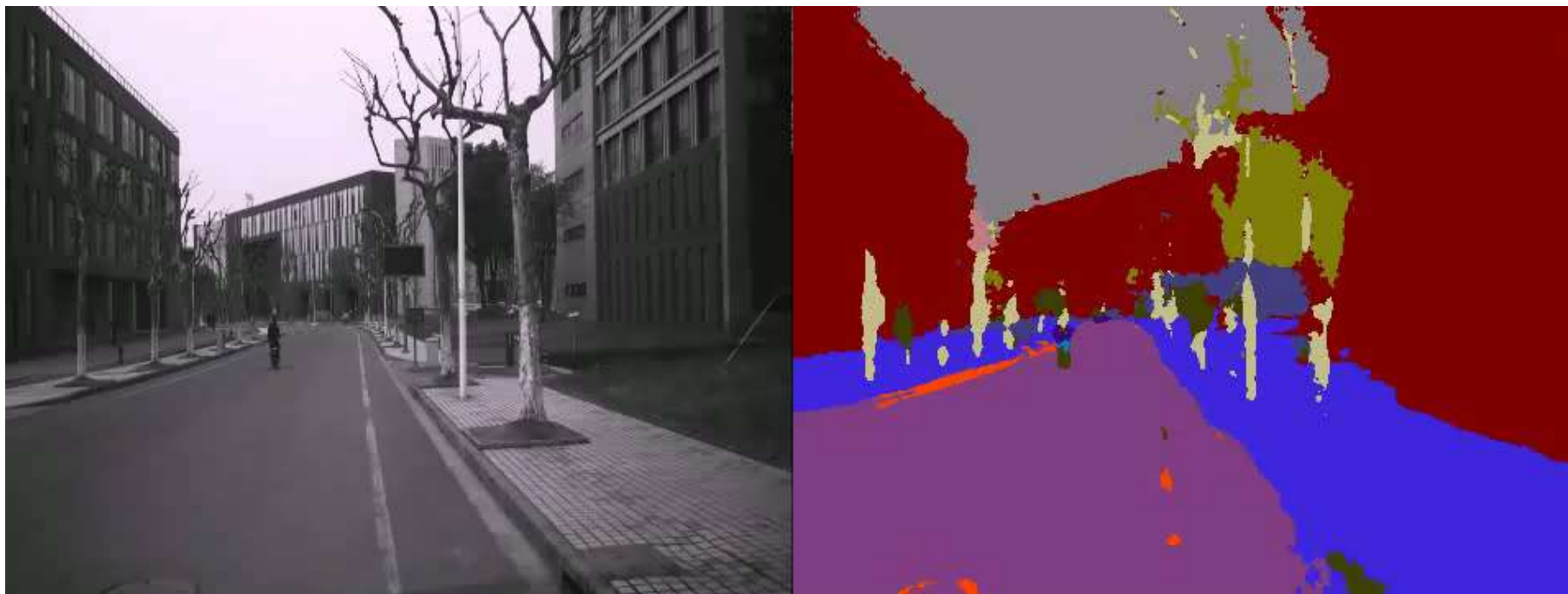
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

SegNet



Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

实例



智能驾驶的三个层级

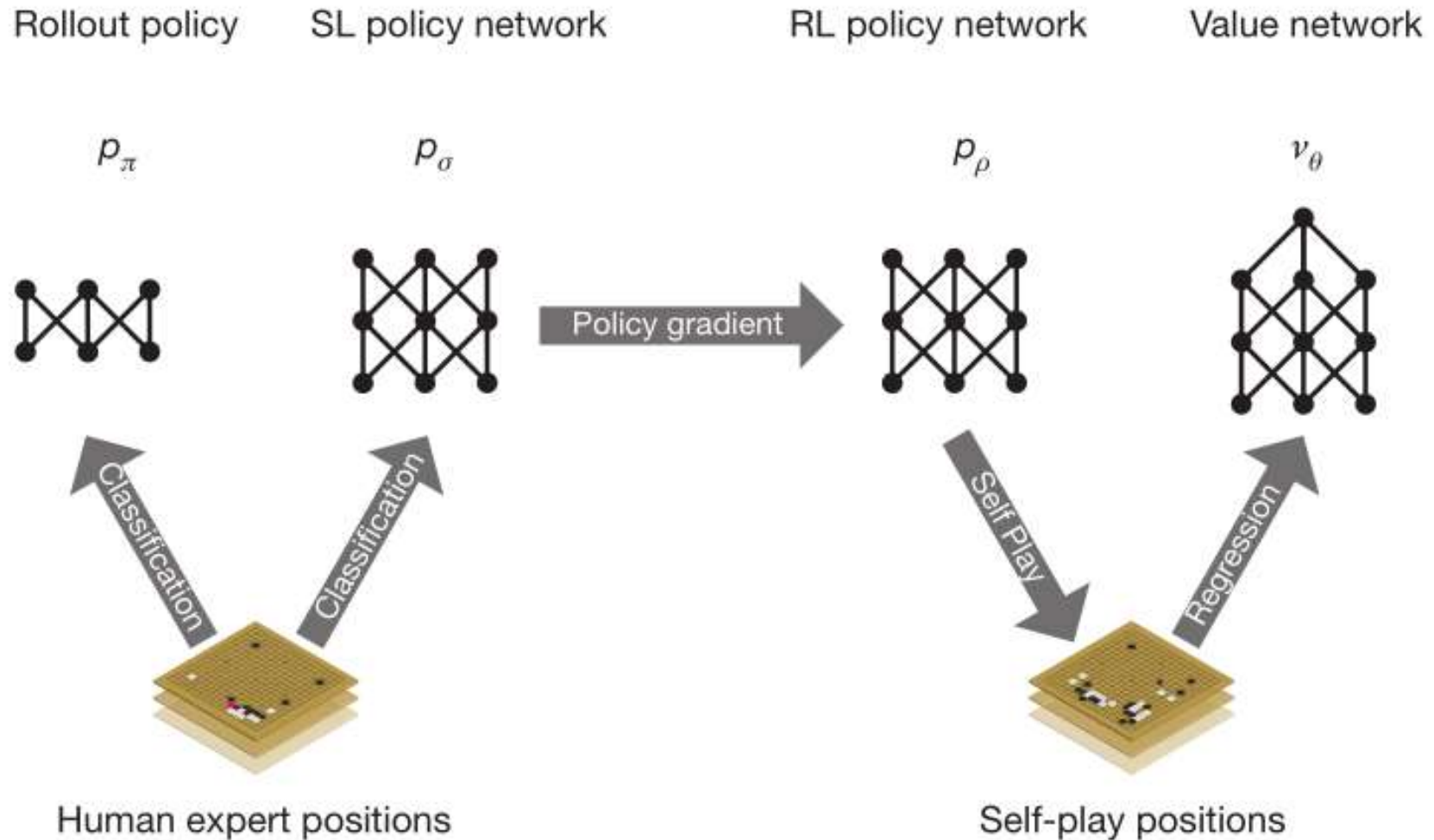
- 环境感知
- 运动规划
- 强化学习

运动规划



复杂交通场景中，路面情况非常多，难以通过穷举规则解决

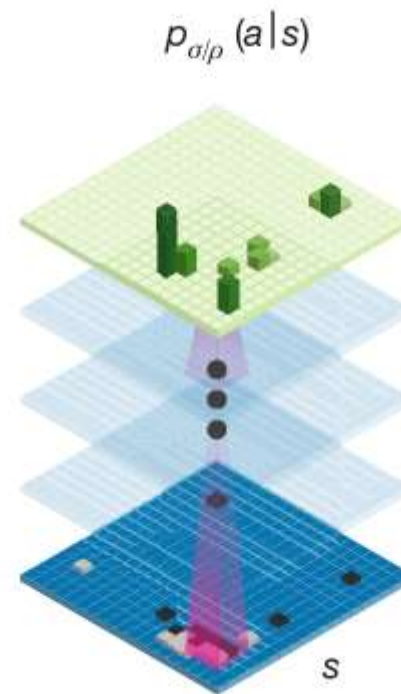
无法穷举的围棋：AlphaGo



Silver et al, Mastering the game of Go with deep neural networks and tree search, Nature 2016

SL policy network-走棋网络

- 使用卷积神经网络
- 三千万个盘面数据（KGS Go Server）
- 预测专业棋手的下一步棋
- 57%的准确率
- 建模了“棋感”
- 大局观非常强
- 完全不搜索



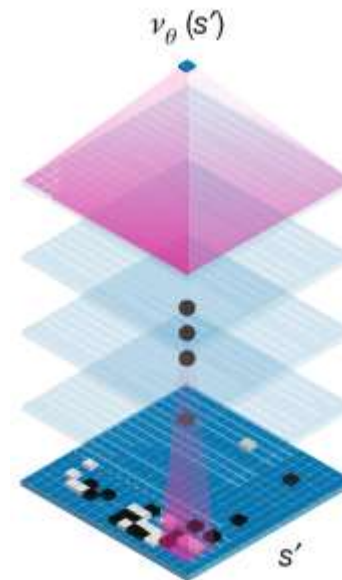
Rollout policy –快速走子

- 使用围棋领域知识来选择局部特征
- 使用线性回归+规则
- 速度非常快（2微秒每步）
- 准确度低（24.2%）
- 能准确判断局部的死活
- 大局观不强
- 用于评估盘面

Value network-估值网络

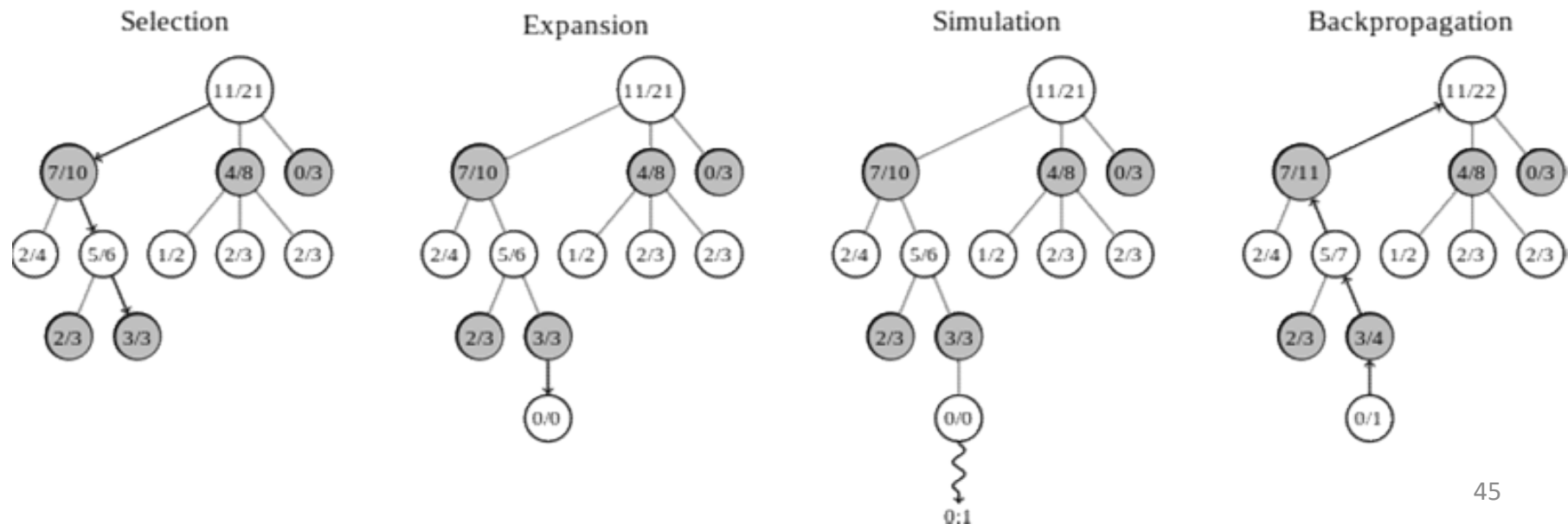
- 看一眼就可以给出当前盘面输赢的概率
- 使用三千万局棋训练（由自我对局产生）
- 每局只取一个盘面（防止过拟合）
- 与快速走子互补（各取一半效果最好）

$$V(s_L) = (1 - \lambda)v_{\theta}(s_L) + \lambda z_L$$

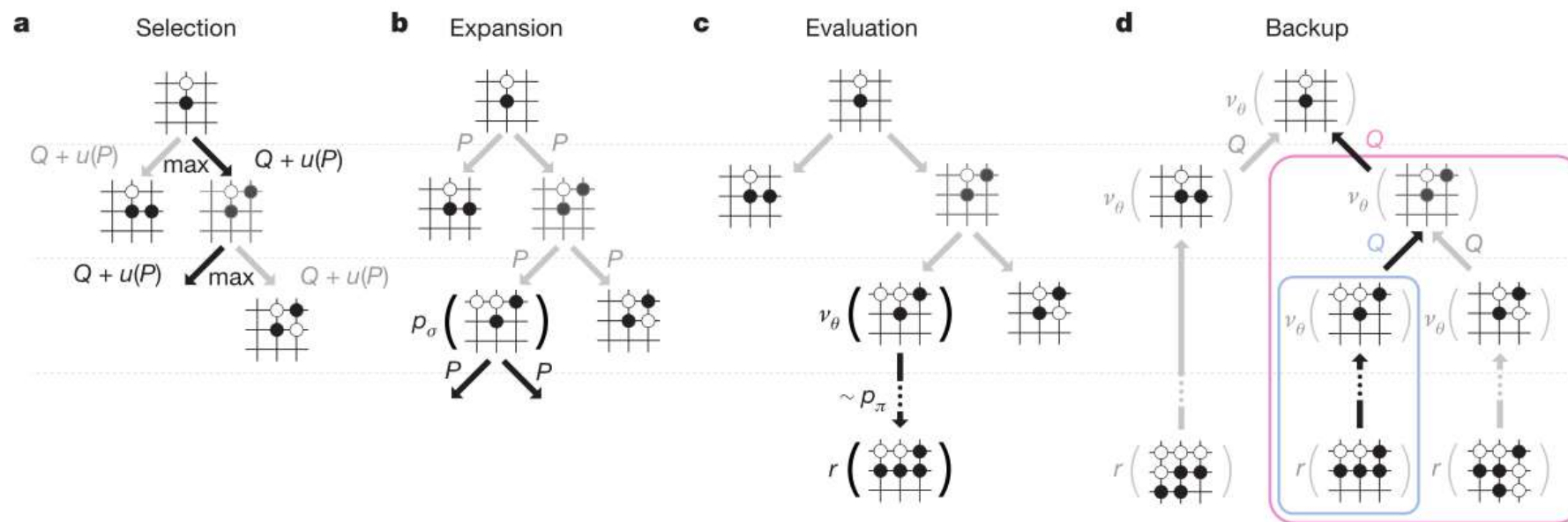


蒙特卡罗搜索树

- 没有任何人工的特征
- 通过不断自对弈来提高能力
- 问题：初始策略太简单（均匀分布）



Alpha Go 算法



AlphaGo的启发

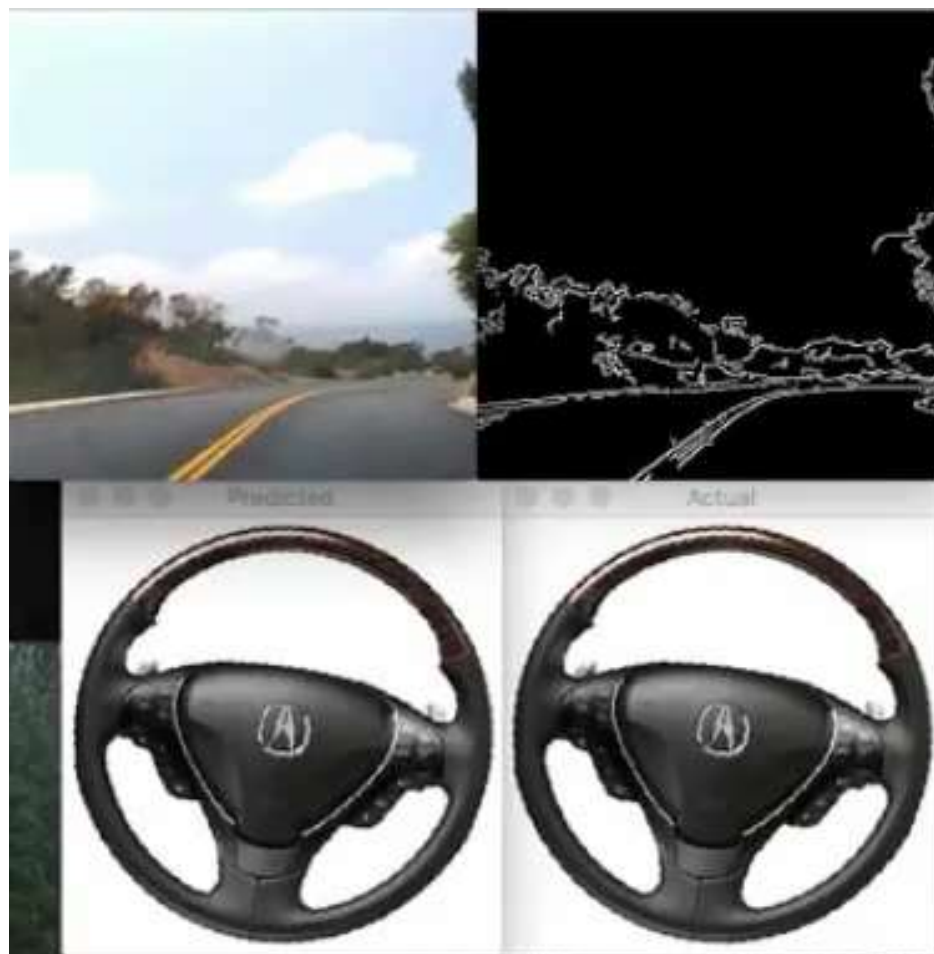
- 用神经网络及大量的数据对我们难以手工定义的函数分布进行模拟学习
- 使用梯度下降及权值共享使得神经网络模型逼近函数的真实分布
- 多种互补策略的融合可以达到非常好的效果

建模 “车感”

- 采集大量熟练驾驶员的驾驶数据
 - 图像、雷达、油门、车速、方向盘转角等
- 使用卷积神经网络训练模型
- 直接预测车辆的车速和方向盘转角



用图像预测方向盘转角



<https://github.com/SullyChen/Caffe-Autopilot>

轨迹的快速模拟与评估

- 对交通场景内的车辆、行人的运动轨迹分布进行采样及简单运动模拟
- 根据是否符合交通规则及对周围车辆、行人的友好性、安全性，对车辆的运动轨迹打分
- 辅助判断车辆运动轨迹的合理性

险情直觉

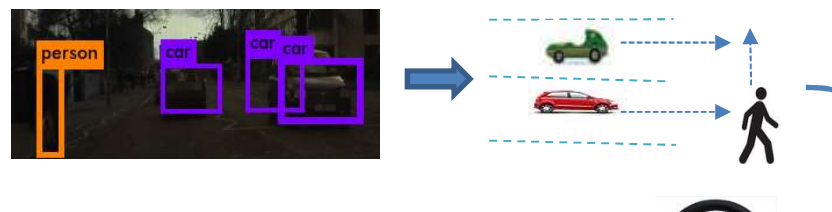
- 经验
- 能迅速
- 自动
- 但是
- 卷积
- 配合



意识
态

智能车局部路径规划算法

- 先布
- 然
- 月数
- 综及



不同的场景需要大量的训练数据，
实车测试非常不安全、不经济。

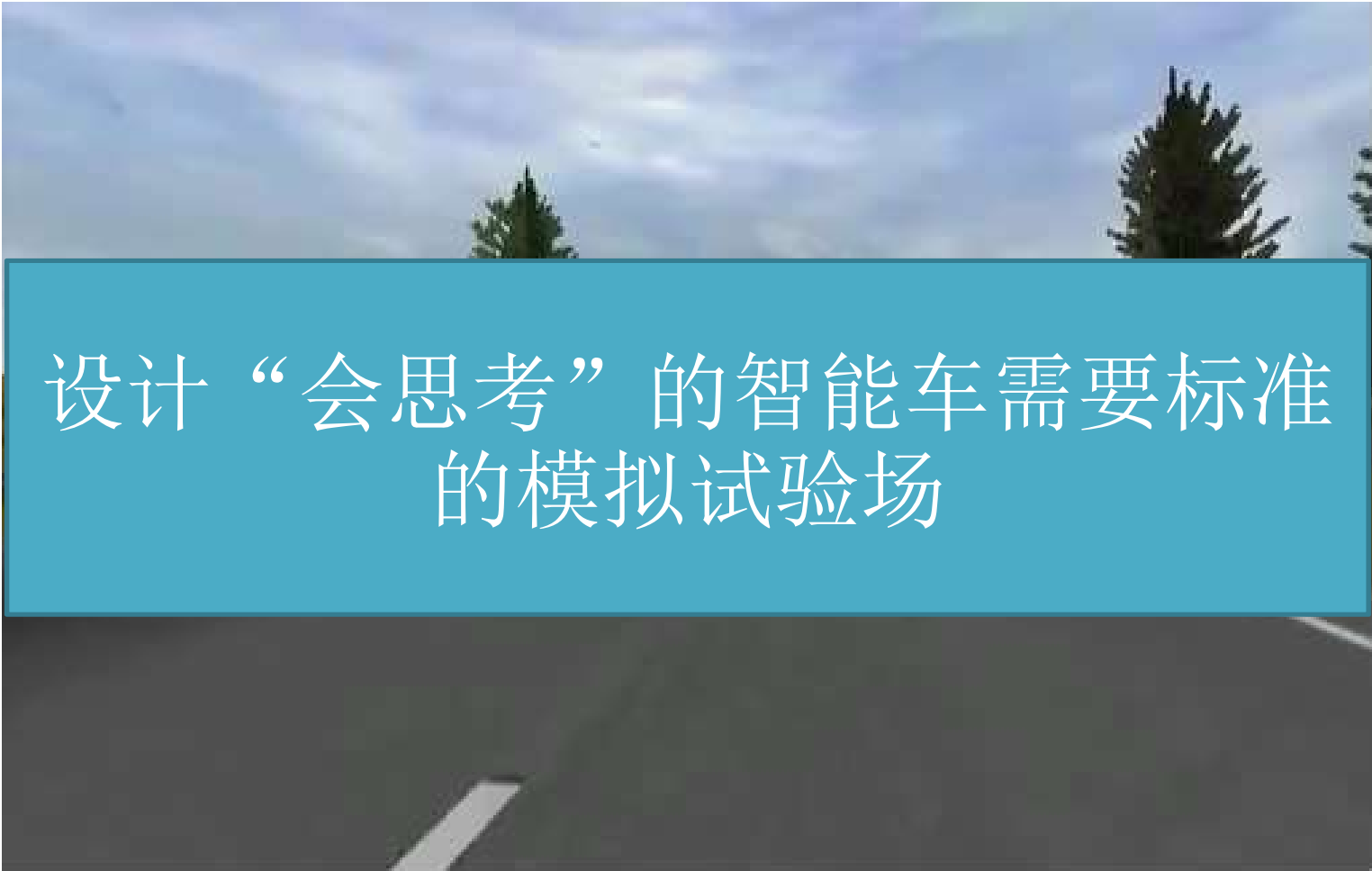


区分

还

路径
车速

3D赛车模游戏：TORCS



设计“会思考”的智能车需要标准的模拟试验场

智能驾驶的三个层级

- 环境感知
- 运动规划
- 强化学习

强化学习

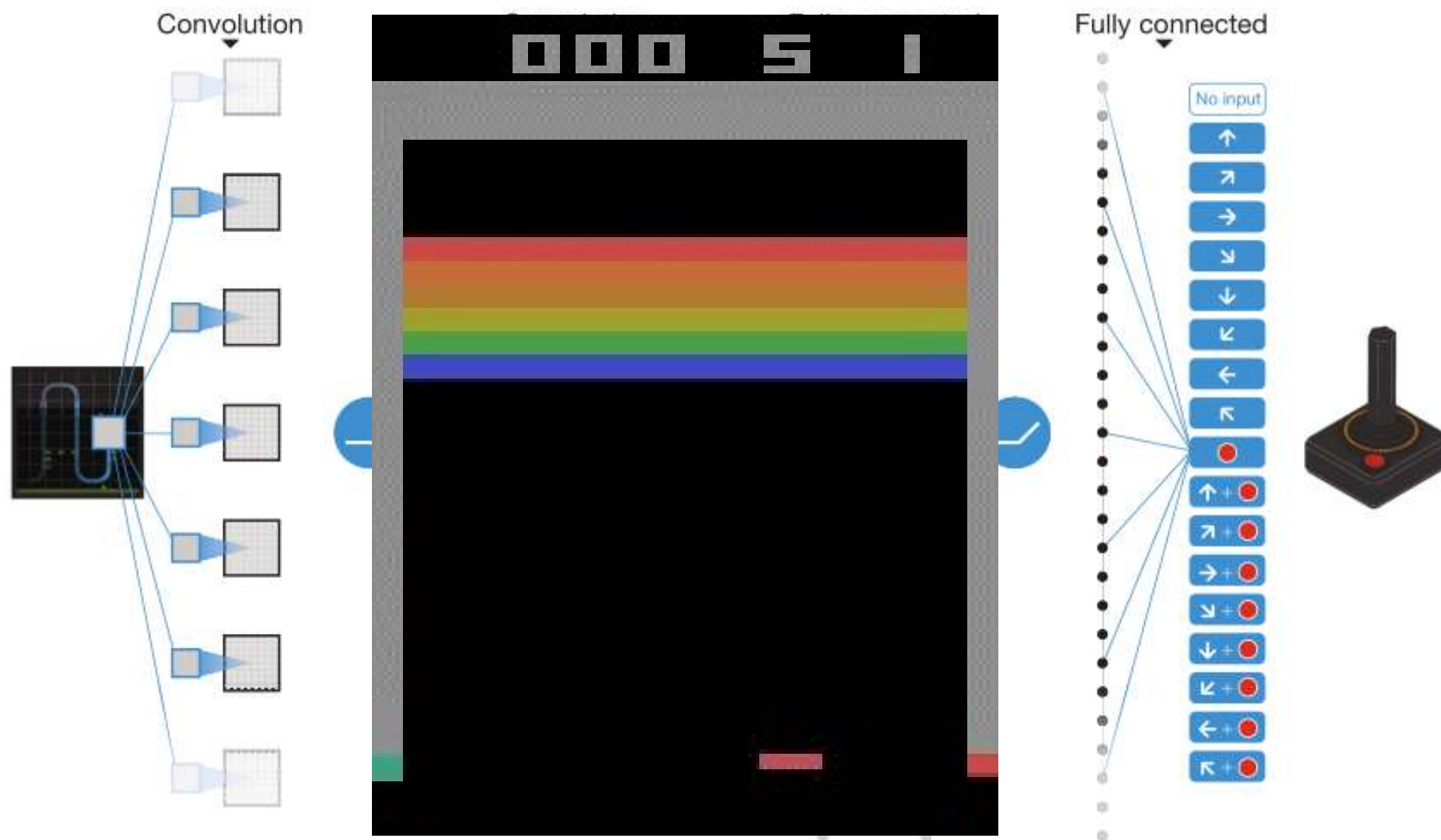
价值函数

深度强化学习

使用深度神经网络表示策略函数，价值函数和模型函数，并用随机梯度下降优化它们。



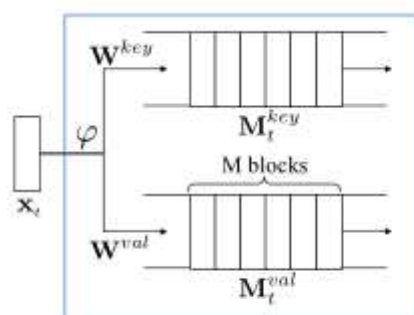
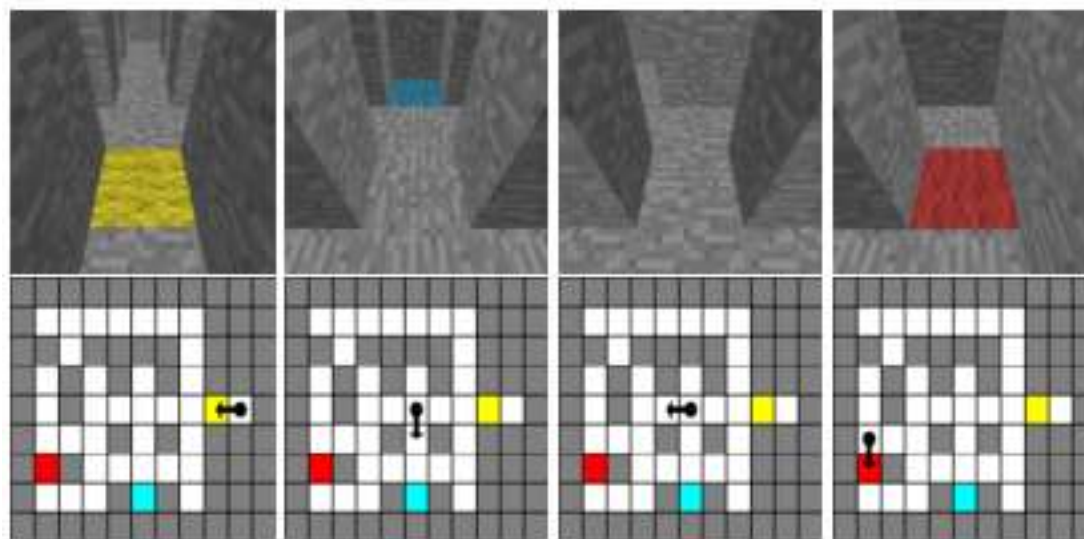
端到端的深度强化学习



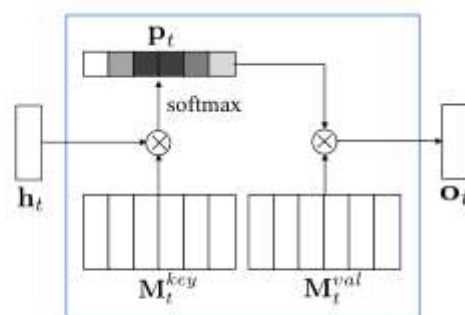
Mnih et al, Human-level control through deep reinforcement learning, Nature 2015

走三维迷宫

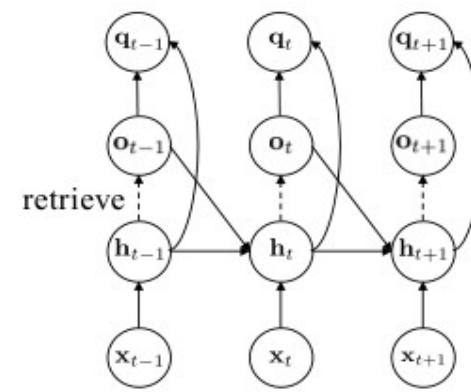
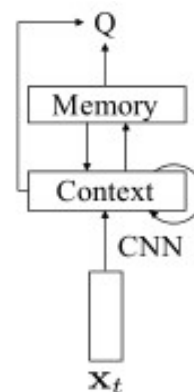
- 感知
- 记忆
- 反馈
- 动作决策



(a) Write

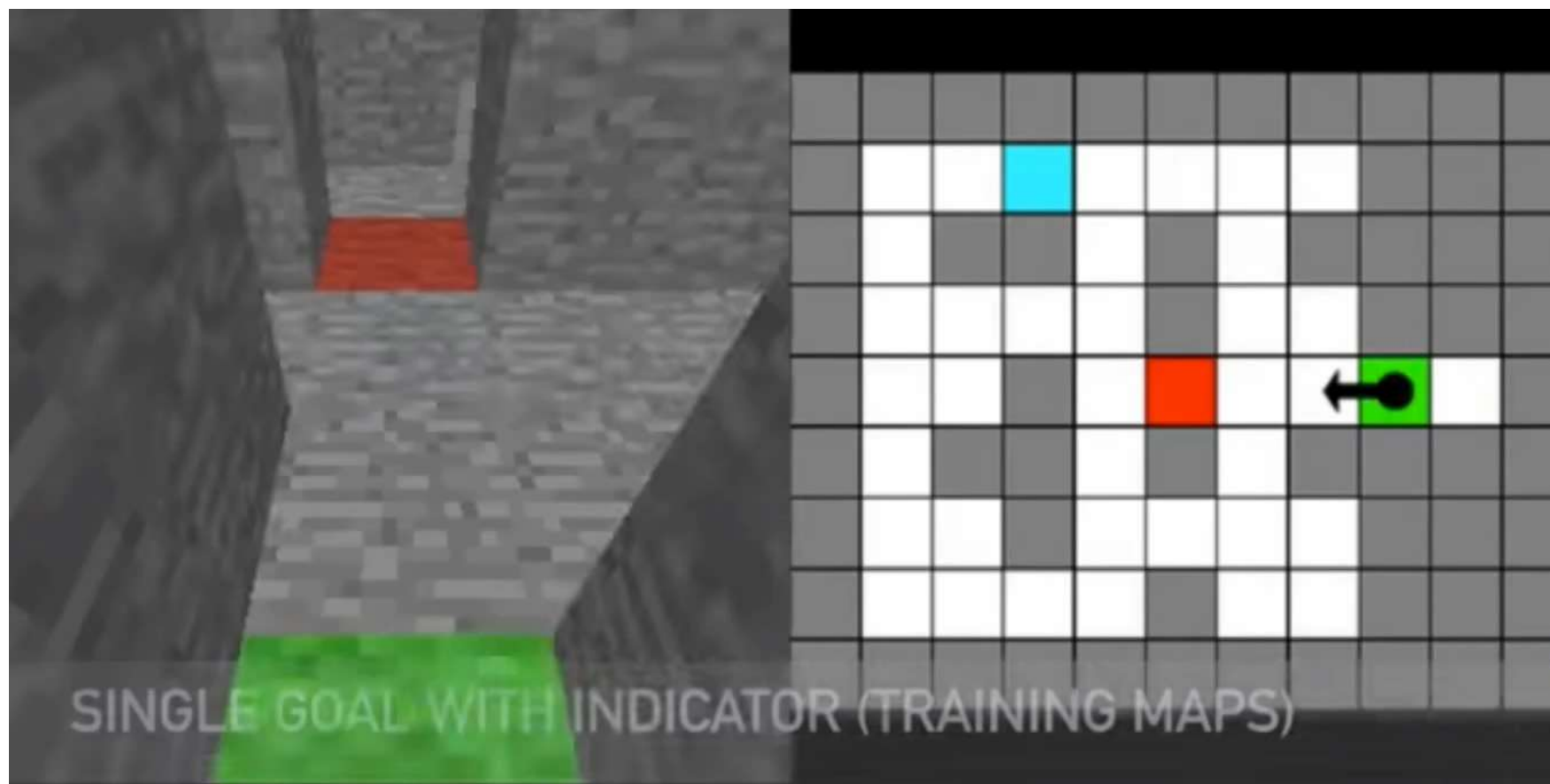


(b) Read



Oh et al, Control of Memory, Active Perception, and Action in Minecraft, arXiv 2016

例子：走三维迷宫



Oh et al, Control of Memory, Active Perception, and Action in Minecraft, arXiv 2016

构想：智能驾驶汽车模拟环境

- 用计算系统构建模拟实际道路的环境
- 对道路上的各个要素及不同场景进行建模
- 对环境设定奖惩机制，如完成指定任务加分、撞车、违反交通规则减分
- 利用深度增强学习，可以让其不断的自我训练，并产生大量的驾驶数据。
- 这个模拟场景可以检验智能驾驶原型系统的各种缺陷



谢谢！

Q&A

报告人： 管林挺

邮箱： glinting@tongji.edu.cn