

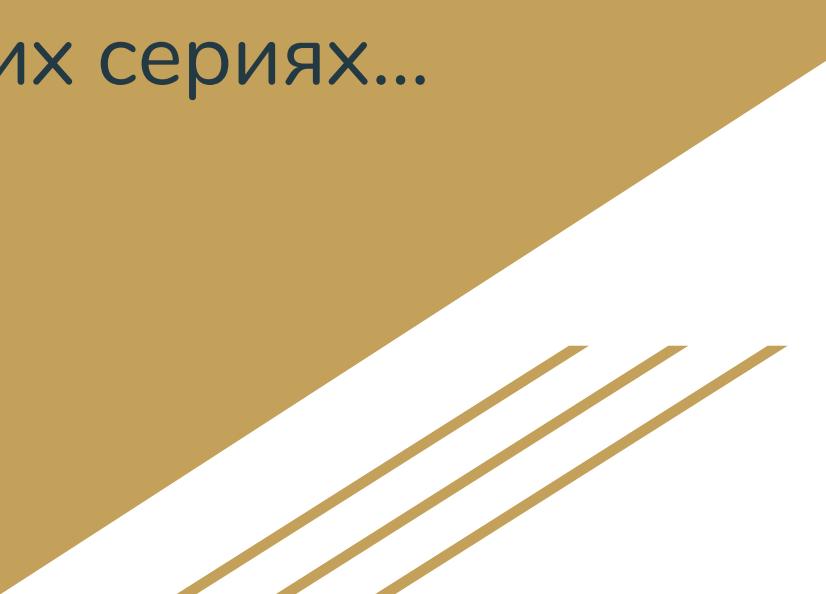
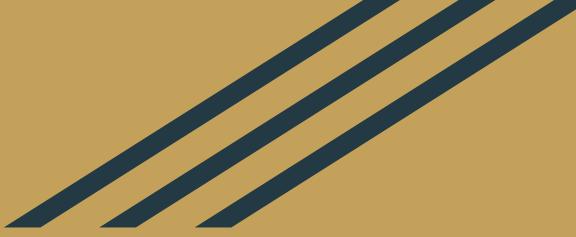
Компьютерное зрение

Лекция 9.

Segmentation, Detection, Instance Segmentation

27.06.2020

Руслан Алиев



В предыдущих сериях...

Gradient explosion / vanishing

В очень глубоких нейросетях, во время обратного распространения проходят через большое количество слоев

При сатурирующих активациях (sigmoid, tanh) или с маленькими весами градиенты затухают.

При несатурирующих активациях или с большими весами градиенты ВЗРЫВАЮТСЯ!

Обучение не масштабируется: то, что работает при 2 слоях, не будет работать при 20

Batch normalization

Один из способов борьбы с затухающими градиентами

- Нормализуем активации фильтров пространственно / по мини-батчу

Во время обучения, распределение активаций в сети меняется в течение времени, потому что меняются параметры (веса)

Обучение более стабильное если это изменение (internal covariate shift) не сильное

Если аутпут - $32 \times 32 \times 16$ изображение, батч сайз 64, нормализуем активации для каждого фильтра по всем изображениям в батче

- Т.е. Подсчитаем 16 means и variances
- Вычесть mean, поделить на variance

Batch normalization

Другие преимущества:

Аутпут нормализуется перед активацией, mean 0 var 1 означает что он в “хорошой” области для большинства функций активации

Каждое изображение обрабатывается в зависимости от других изображений в батче, своего рода регуляризация, потому что сетка “не видит” одно и то же изображение дважды

Стабилизирует обучение, можно использовать высокий learning rate

Residual connections

Обычно, атпут после двух слоев: $f(w^*f(vx))$

Residual connections: $f(w^*f(vx)) + x$

Учимся как нужно модифицировать x

Дает градиентам другой путь, гораздо меньше затухания

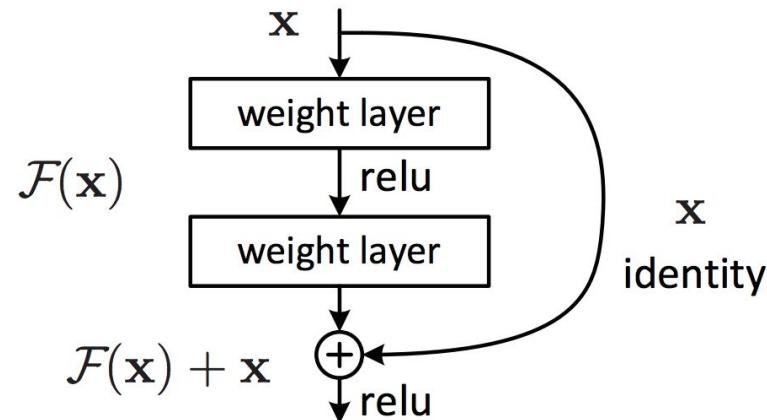
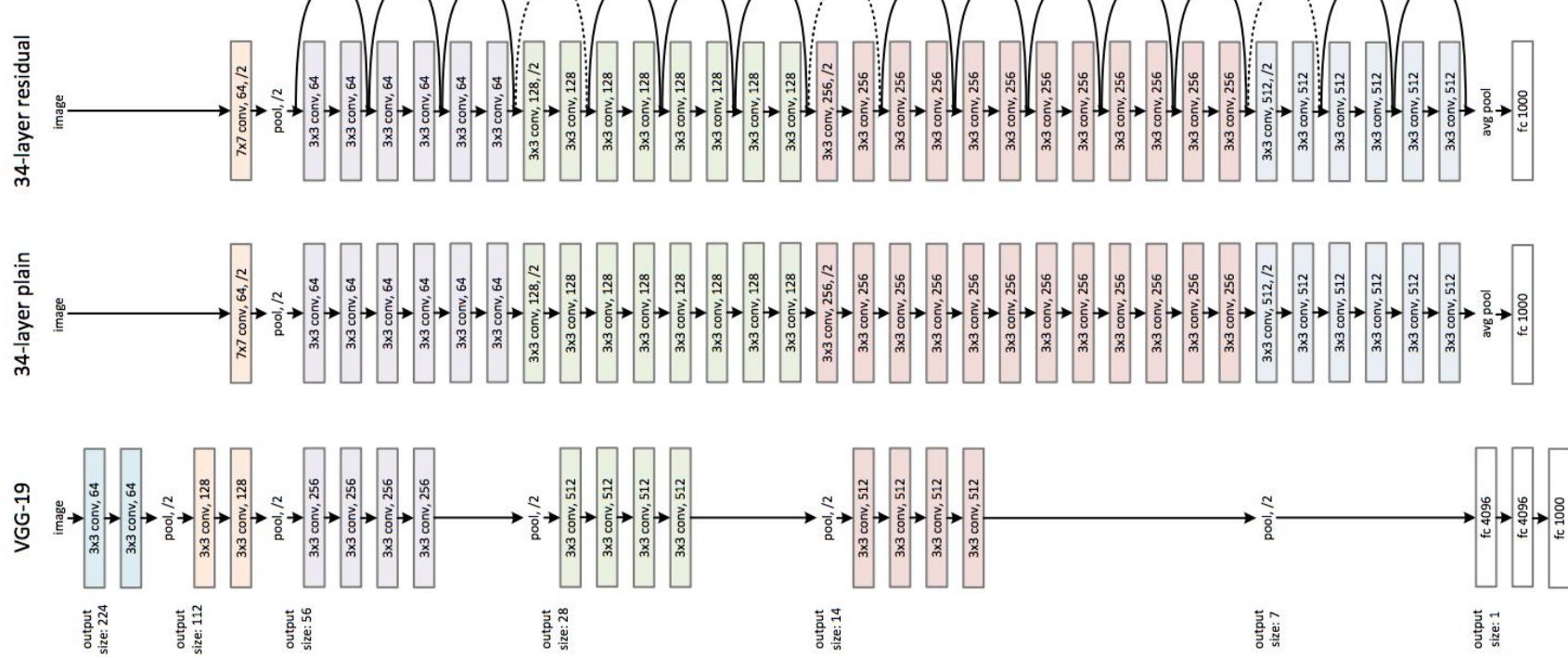


Figure 2. Residual learning: a building block.

ResNet



ResNet

3x3 conv блоки или 3x3 и 1x1 conv блоки

Residual connections

ОЧЕНЬ глубокая, 100+ слоев

Grouped convolutions

Большинство фильтров смотрят на каждый канал в инпуте

Очень дорого

Может быть это излишне? Брать информацию только из нескольких каналов

Групповые конволюции:

Делим инпут на несколько групп по каналам

Делает конволюции на каждой группе отдельно

Конкатенируем результаты конволюций по каждой из групп

Grouped convolutions

Групповые конволюции:

Делим инпут на несколько групп по каналам

Делает конволюции на каждой группе отдельно

Конкатенируем результаты конволюций по каждой из групп

Например: 3x3 conv блок, 32x32x256 инпут, 128 фильтров, 32 группы:

Делим инпут на 32 различные группы

Каждая - 32x32x8

Проходим 4 фильтрами размера, 3x3x8 по каждой из групп

Конкатенируем 4*32 канала, получаем 32x32x128 аутпут

Те же размерности инпута и аутпута, меньше вычислений

ResNeXt

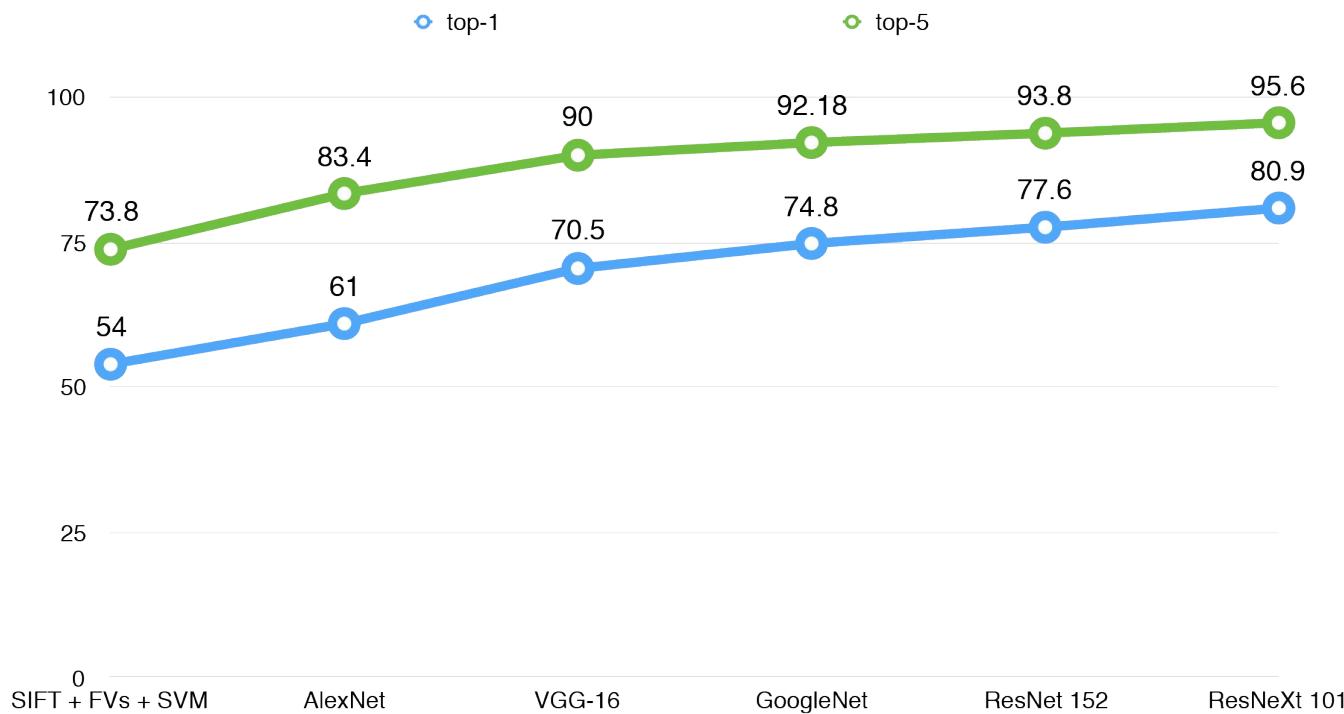
Заменяем 3×3 блоки на grouped convolutions с большим количеством каналов

Сетка “больше”, но такая же вычислительная сложность

stage	output	ResNet-50	ResNeXt-50 (32×4d)
conv1	112×112	$7 \times 7, 64$, stride 2	$7 \times 7, 64$, stride 2
		3×3 max pool, stride 2	3×3 max pool, stride 2
conv2	56×56	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128, C=32 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3	28×28	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256, C=32 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv4	14×14	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512, C=32 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
conv5	7×7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024, C=32 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
	# params.	25.5×10^6	25.0×10^6
	FLOPs	4.1×10^9	4.2×10^9

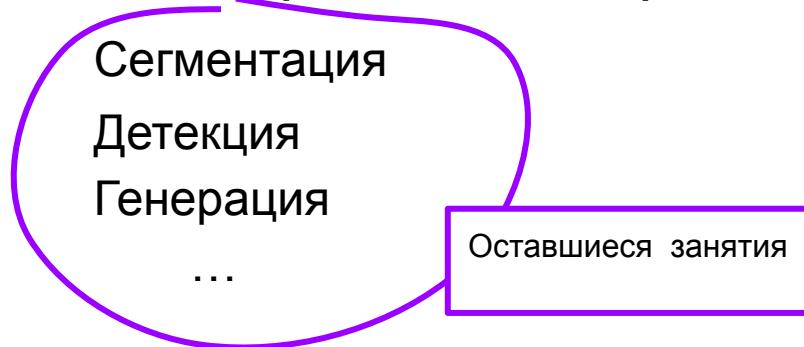
ImageNet

ImageNet становится скучным, боремся за 1-2%



ImageNet становится скучным, боремся за 1-2%

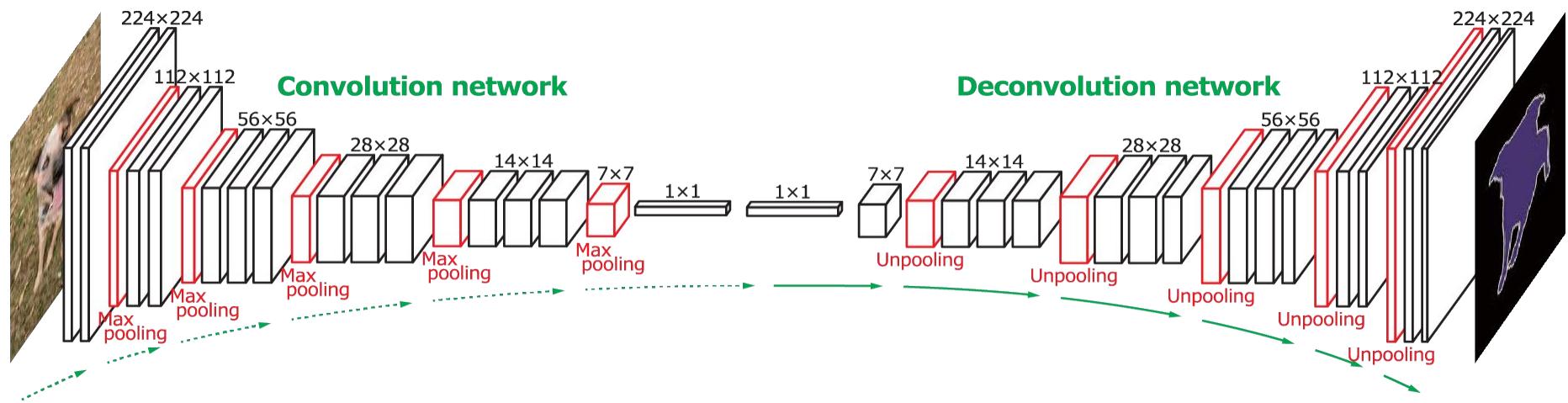
С классификацией справились, другие задания



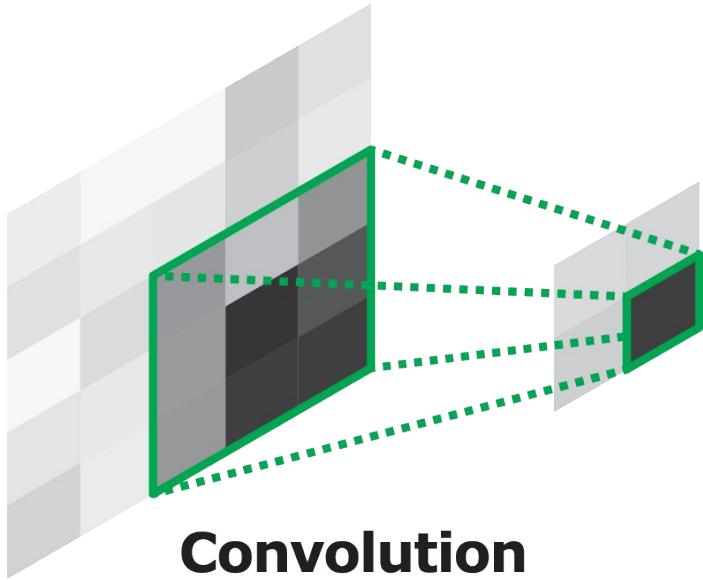
Semantic Segmentation



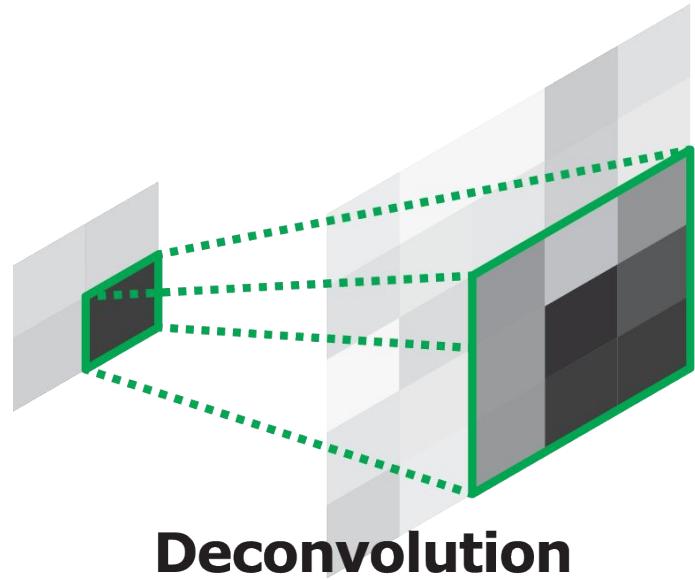
Semantic Segmentation



Semantic Segmentation

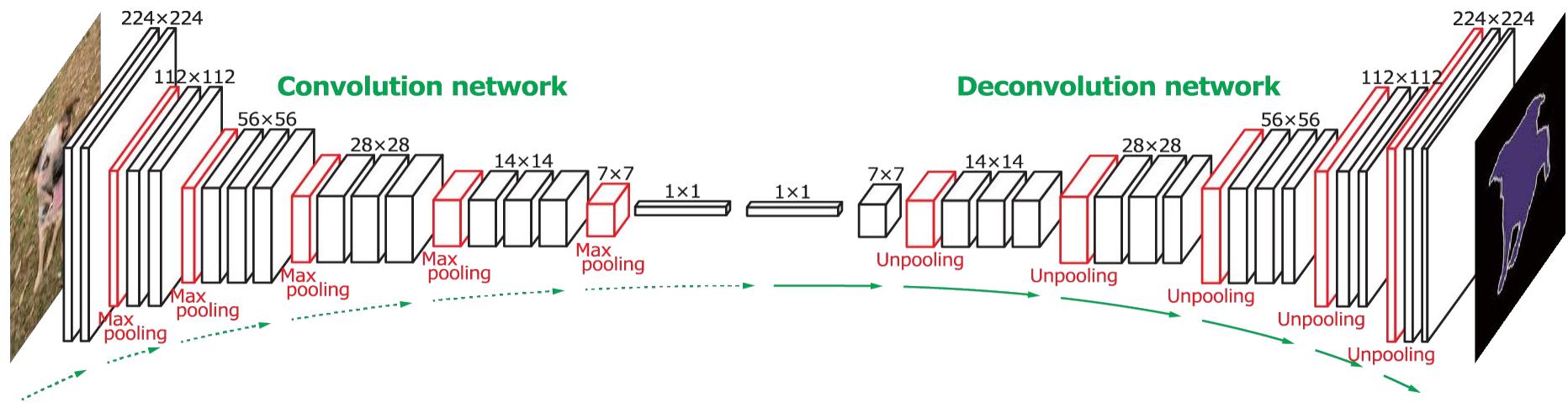


Convolution

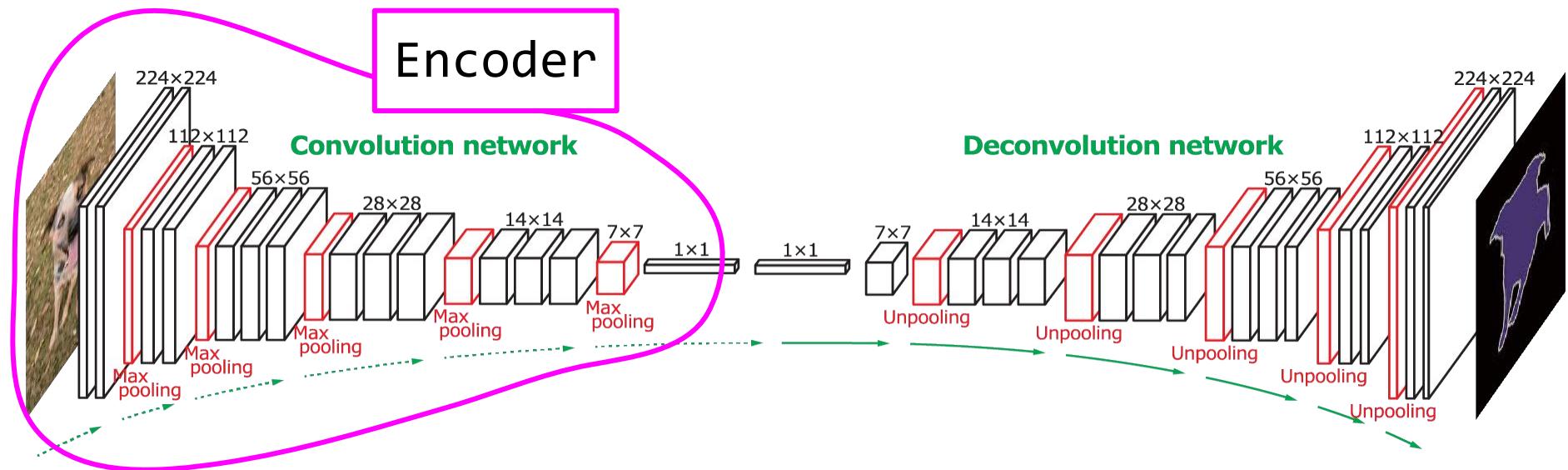


Deconvolution

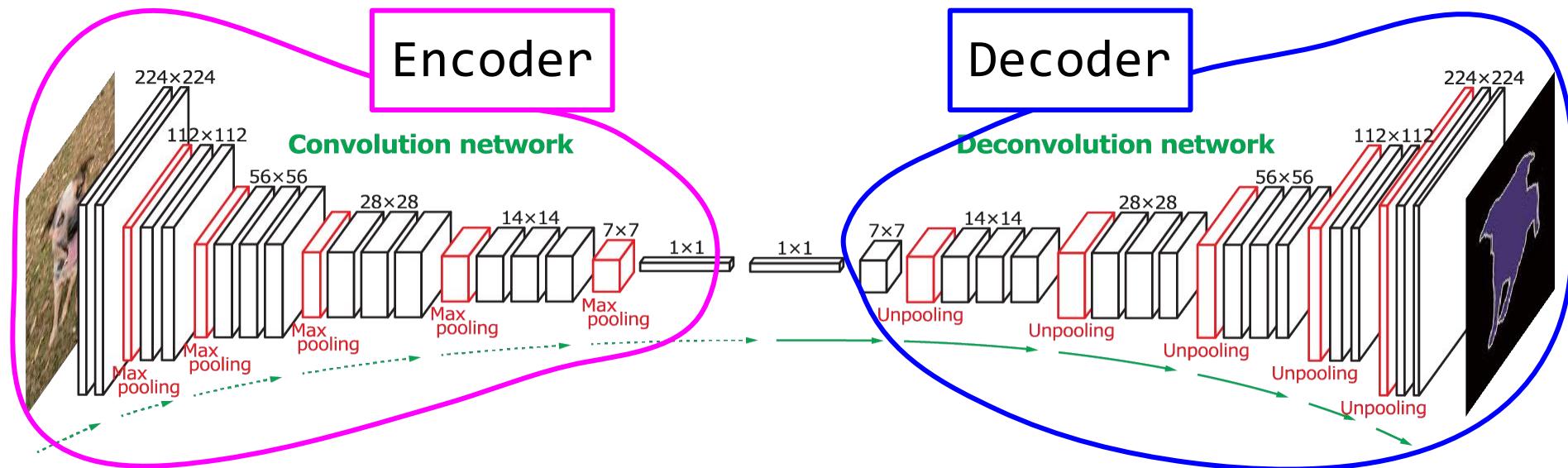
Semantic Segmentation



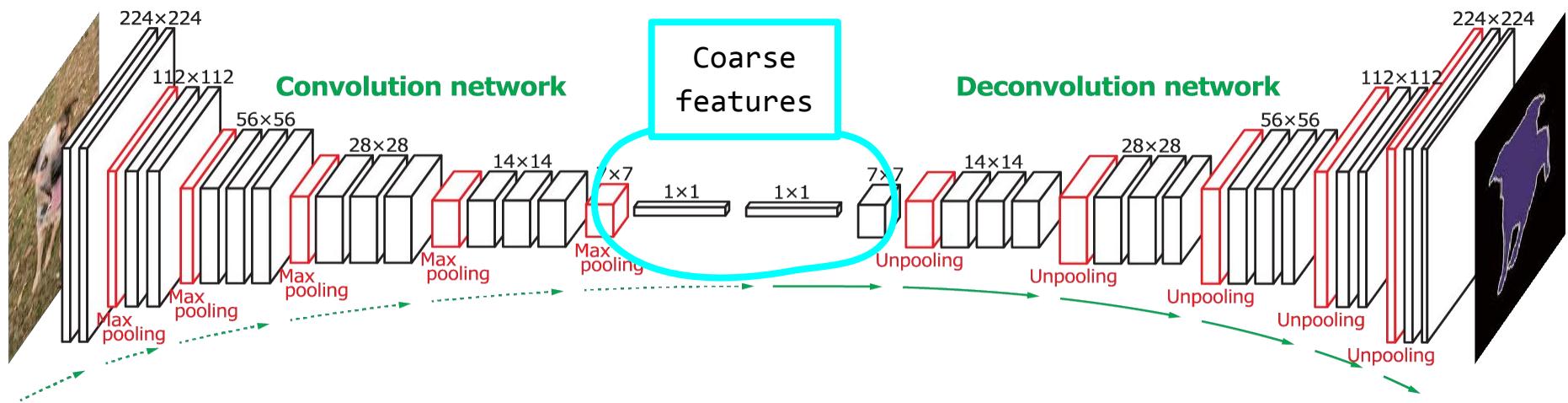
Semantic Segmentation



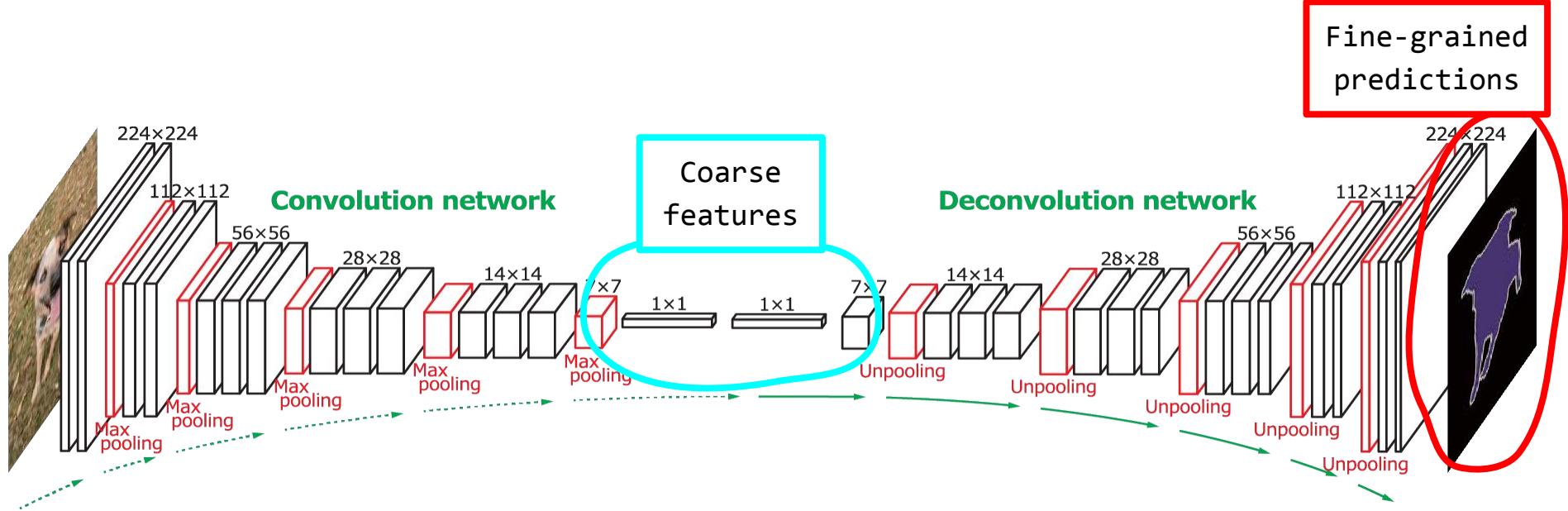
Semantic Segmentation



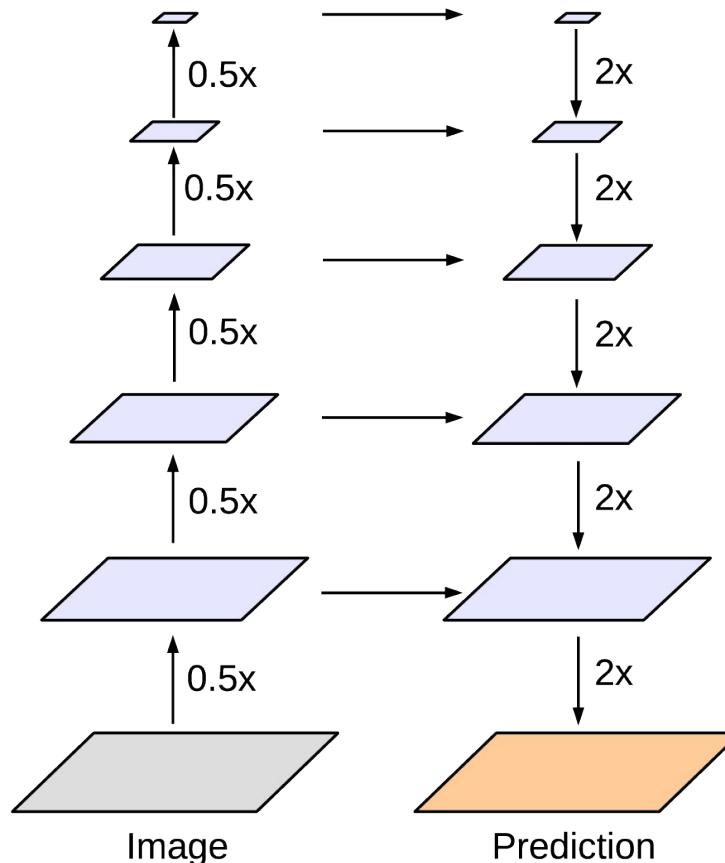
Semantic Segmentation



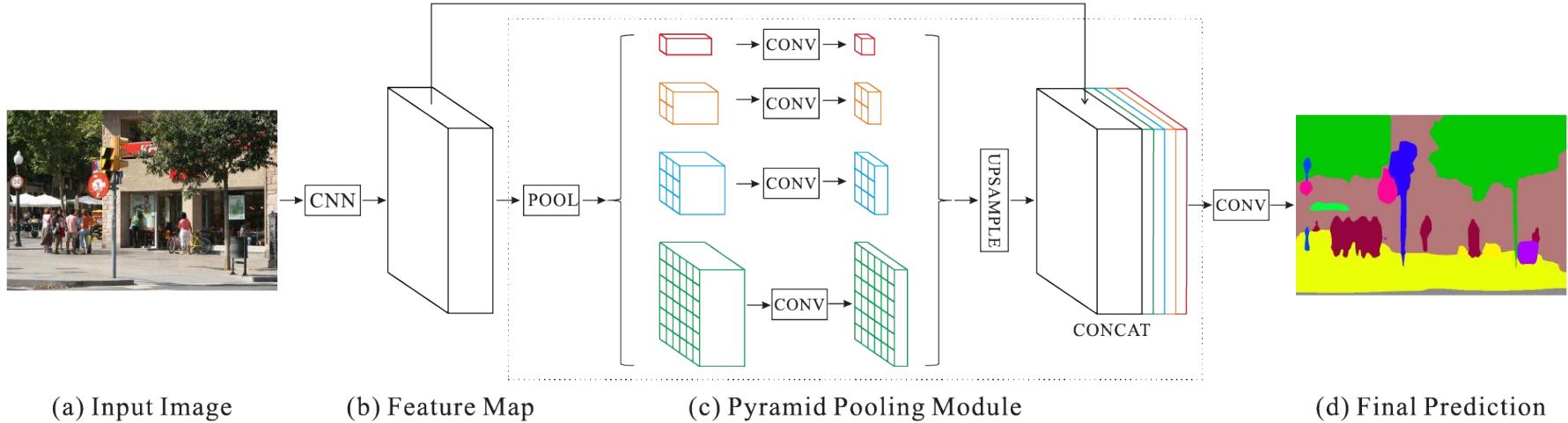
Semantic Segmentation



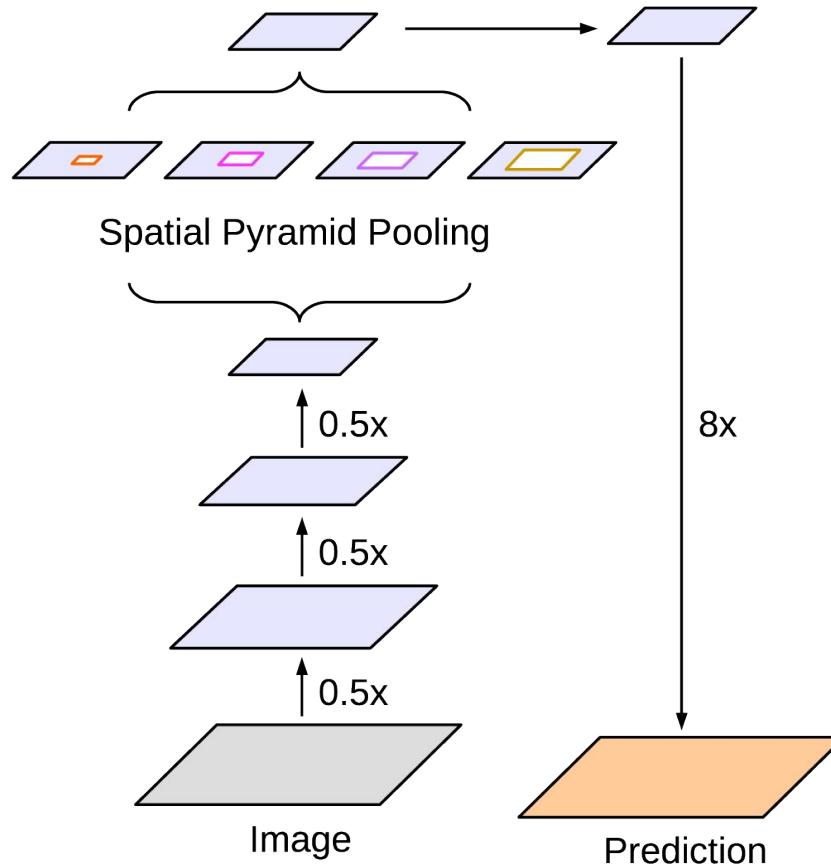
U-net/Segnet



Spatial pyramid pooling

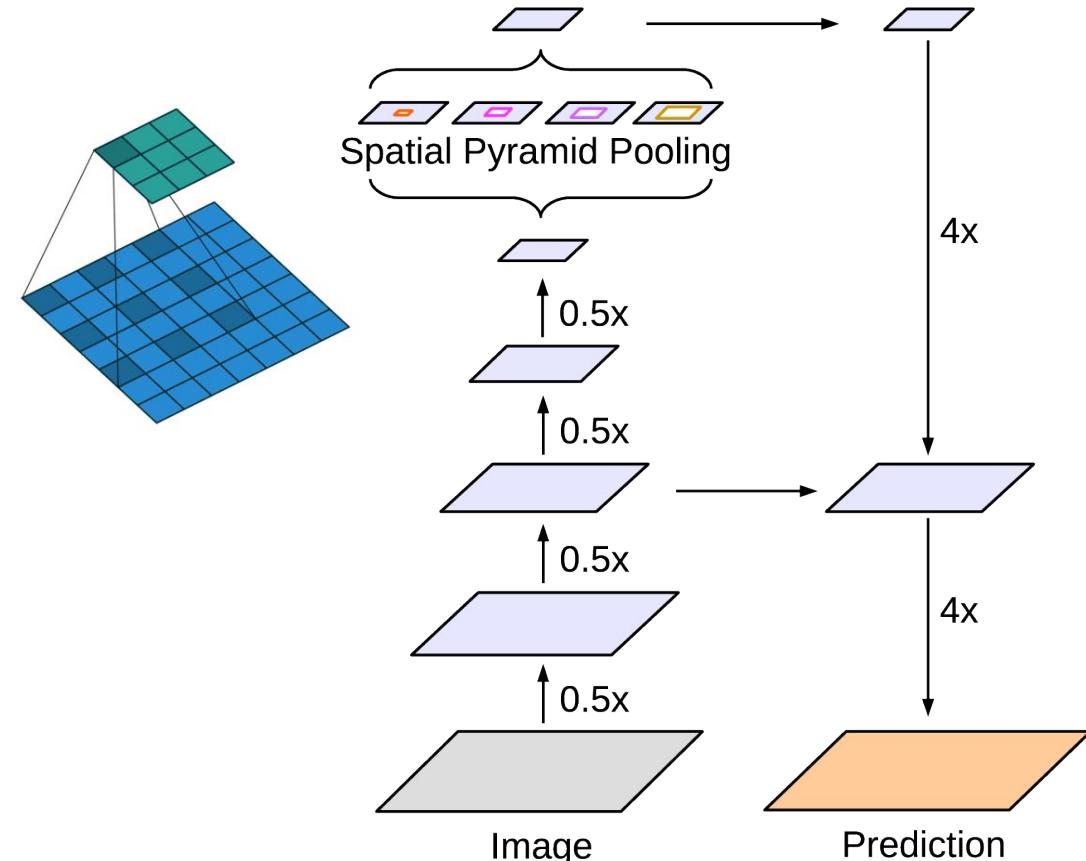


Spatial pyramid pooling



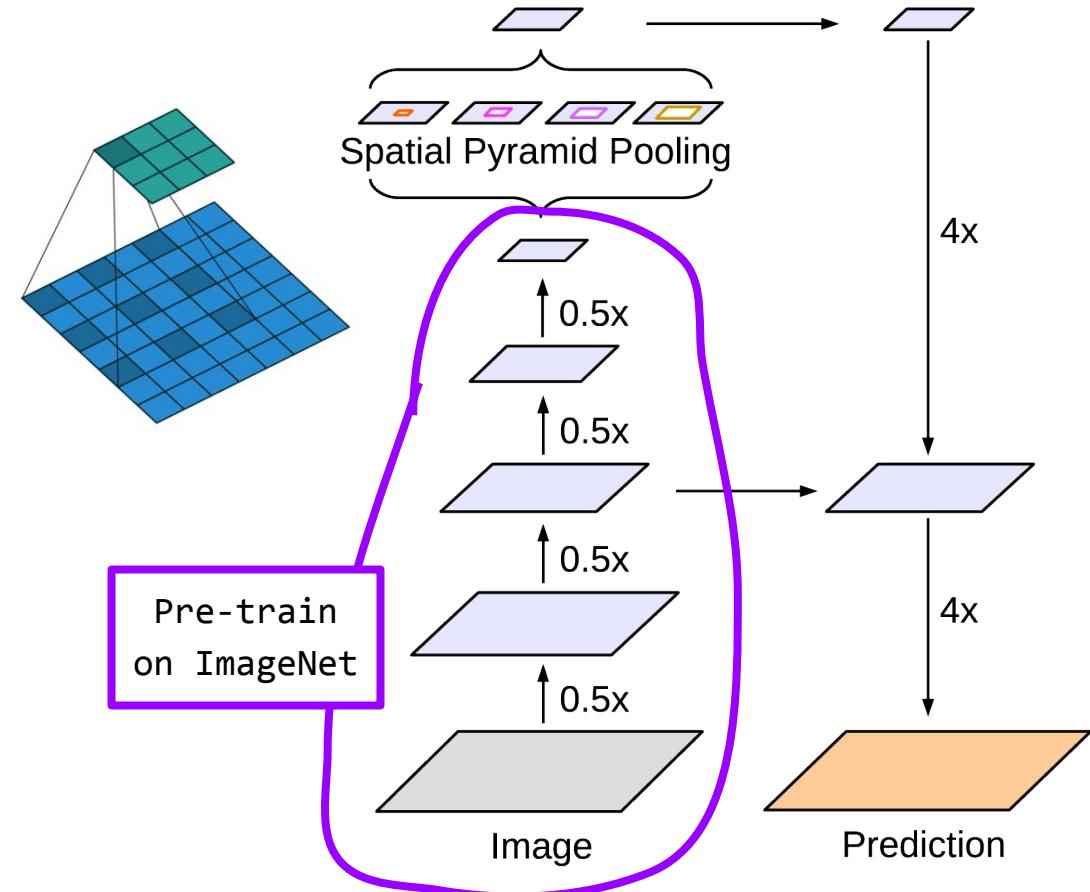
DeepLabv3+

Dilated convolutions



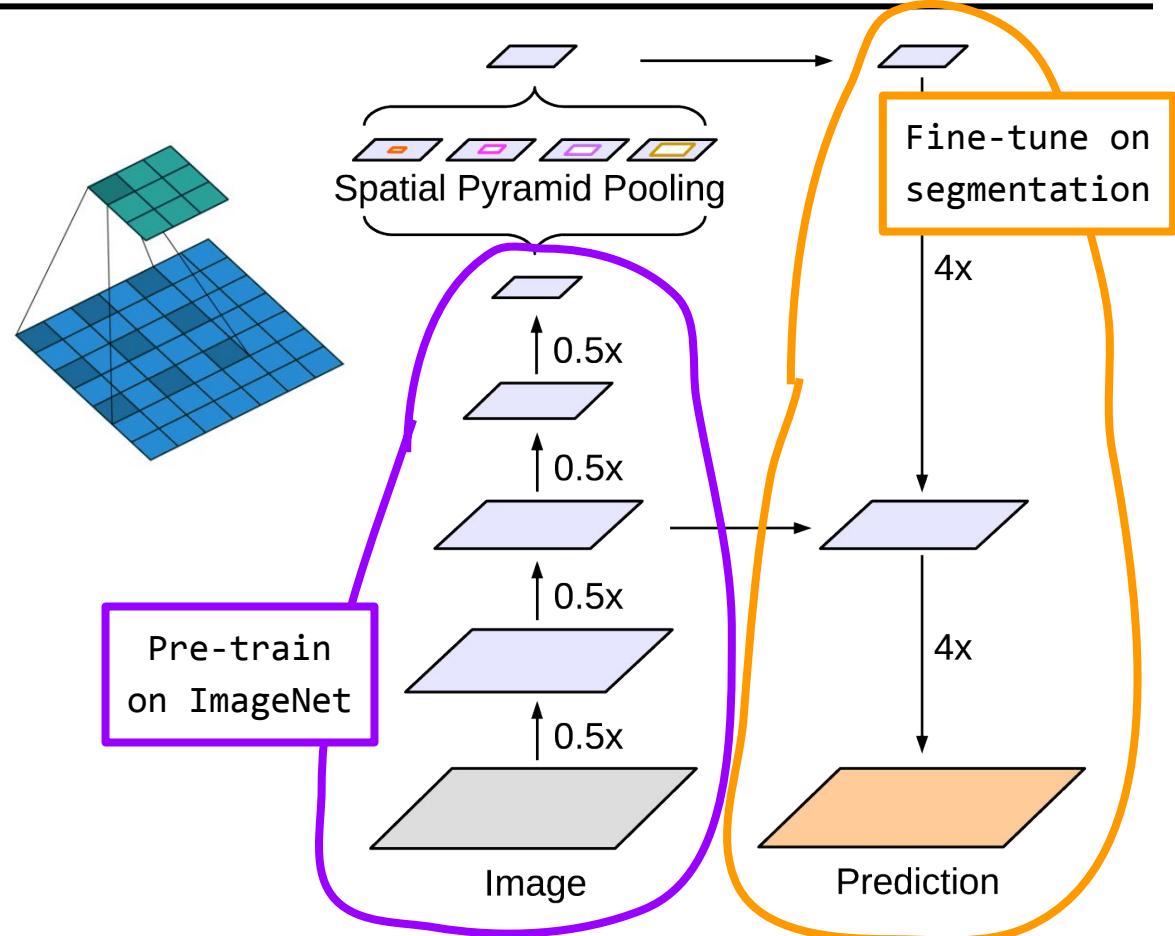
DeepLabv3+

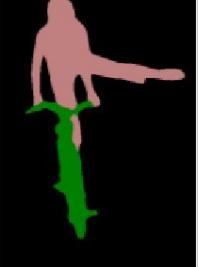
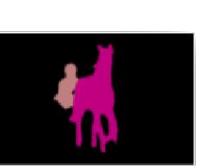
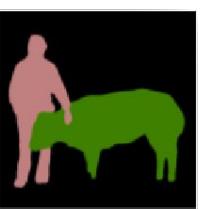
Dilated convolutions



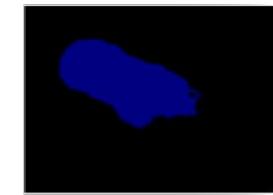
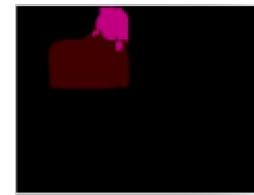
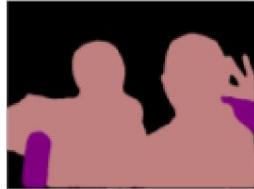
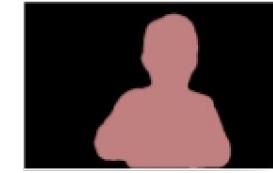
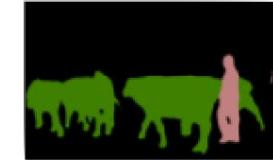
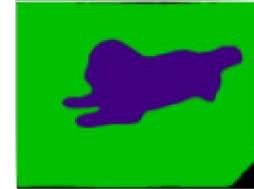
DeepLabv3+

Dilated convolutions



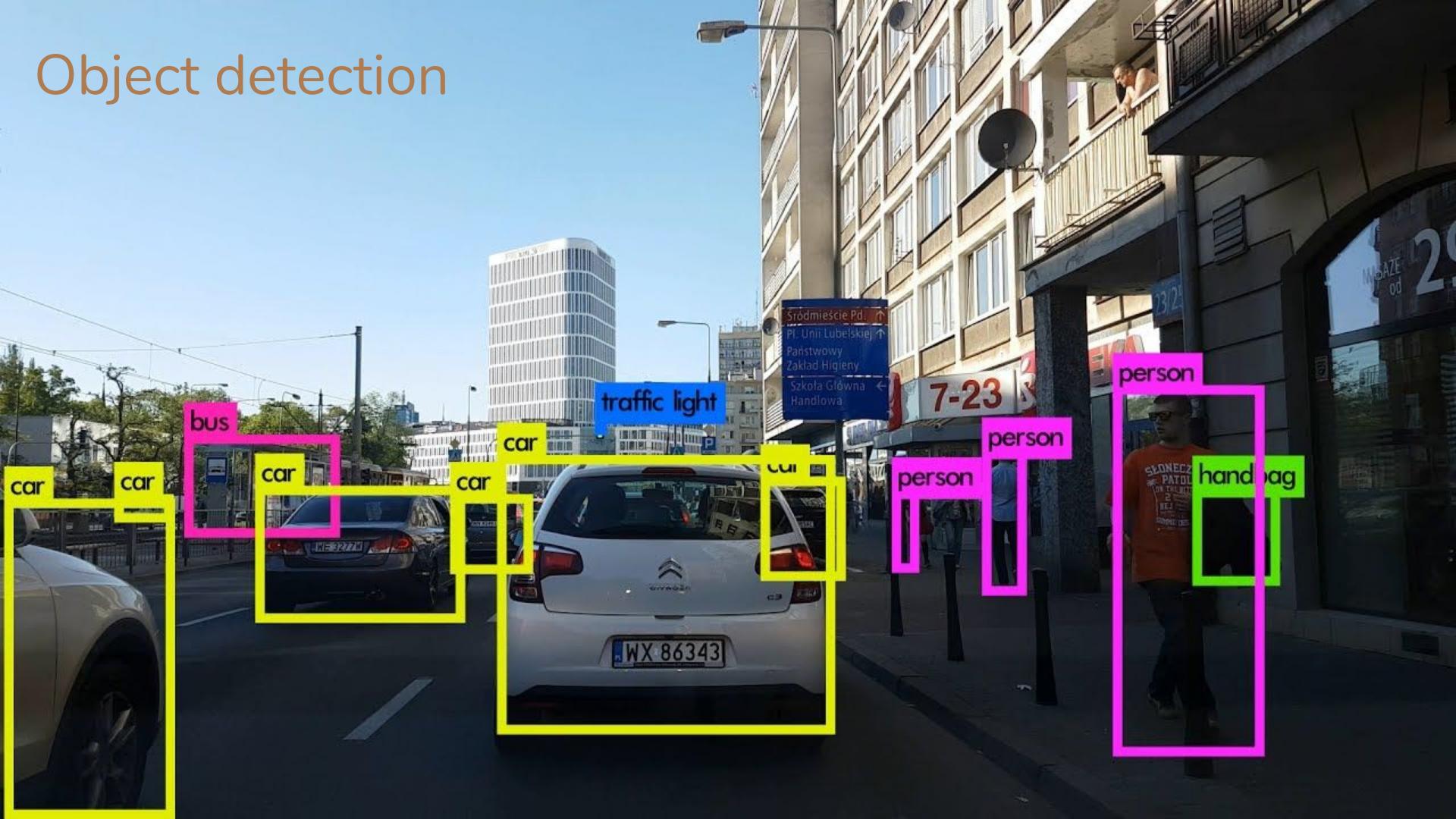


C. Lothar Lenz
www.pferdefotoarchiv.de

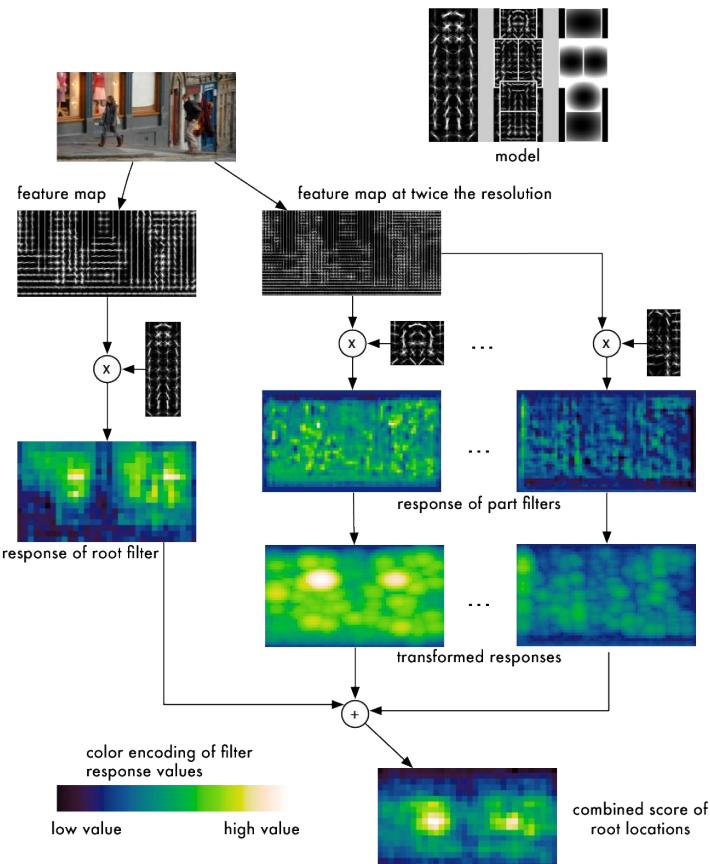




Object detection



Deformable parts models



Метрики детекции

Множество классов, множество объектов на изображении

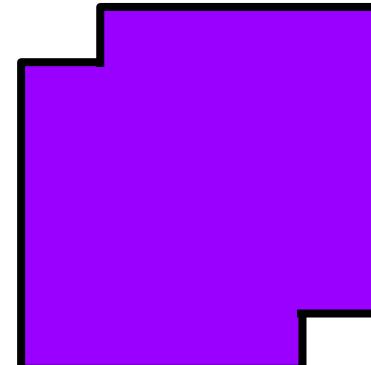
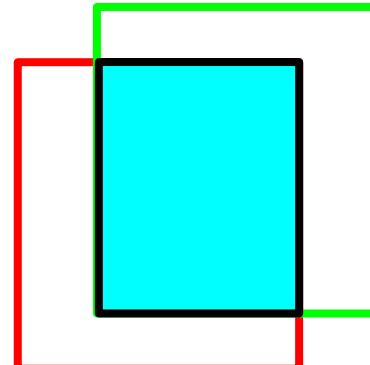
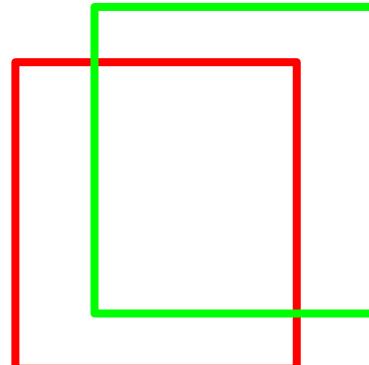
Accuracy не подойдет

“Correct” bounding box:

Intersection / Union > 0.5

Intersection: $\text{Ground truth} \cap \text{prediction}$

Union: $\text{Ground truth} \cup \text{prediction}$



Метрики детекции

“Correct” bounding box:

$$\text{Intersection} / \text{Union} > 0.5$$

Recall:

$$\text{Correct bounding boxes} / \text{total ground-truth boxes}$$

Precision:

$$\text{Correct bounding boxes} / \text{total predicted boxes}$$

Наиболее уверенные предсказания: High precision, low recall

Все предсказания:

Low precision, high recall

Метрики детекции

Precision-Recall curve: изменяем threshold, строим график precision и recall

Average precision:

Площадь под PR-кривой

Только для одного класса

Берем среднее (mean) AP

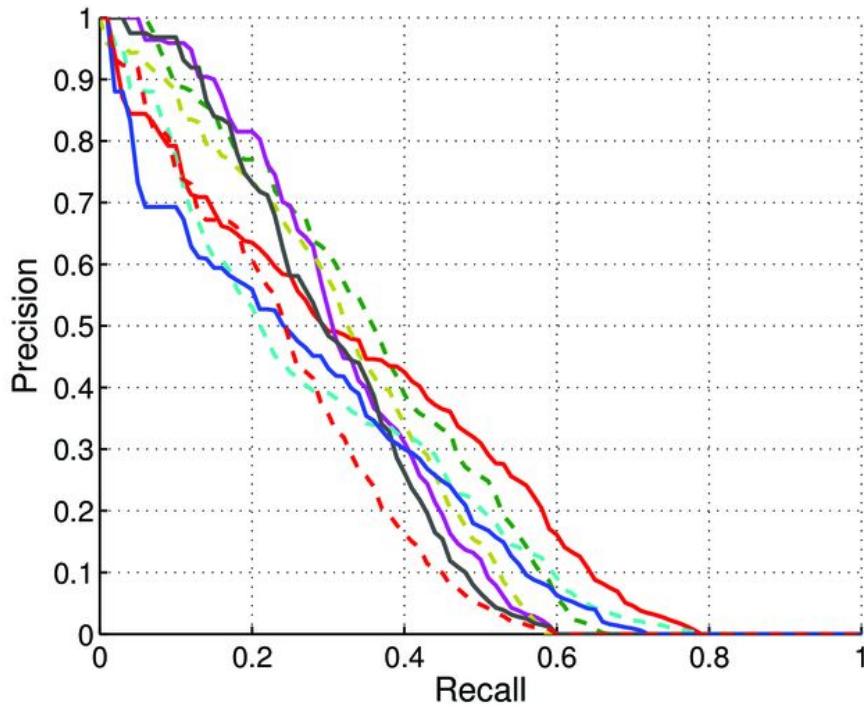
по всем классам:

Mean AP (mAP)

Стандартная метрика детекции

Иногда для конкретного IoU

mAP@.5 or mAP@.75



PASCAL VOC

ОДИН ИЗ ПЕРВЫХ КРУПНЫХ ДАТАСЕТОВ ДЛЯ ДЕТЕКЦИИ

20 classes

11,530 training images

27,450 annotated objects

DPM: 33.6% mAP

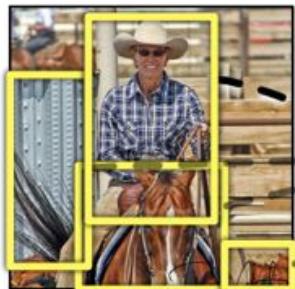
DPM - донейросетовой алгоритм

Как мы можем использовать CNN для детекции?

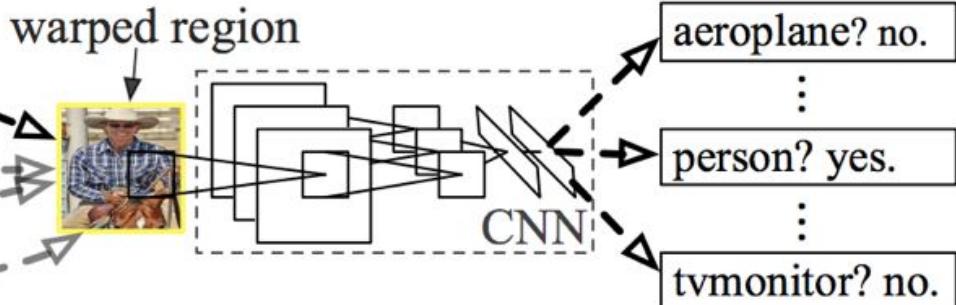
R-CNN: Regions with CNN features



1. Input image



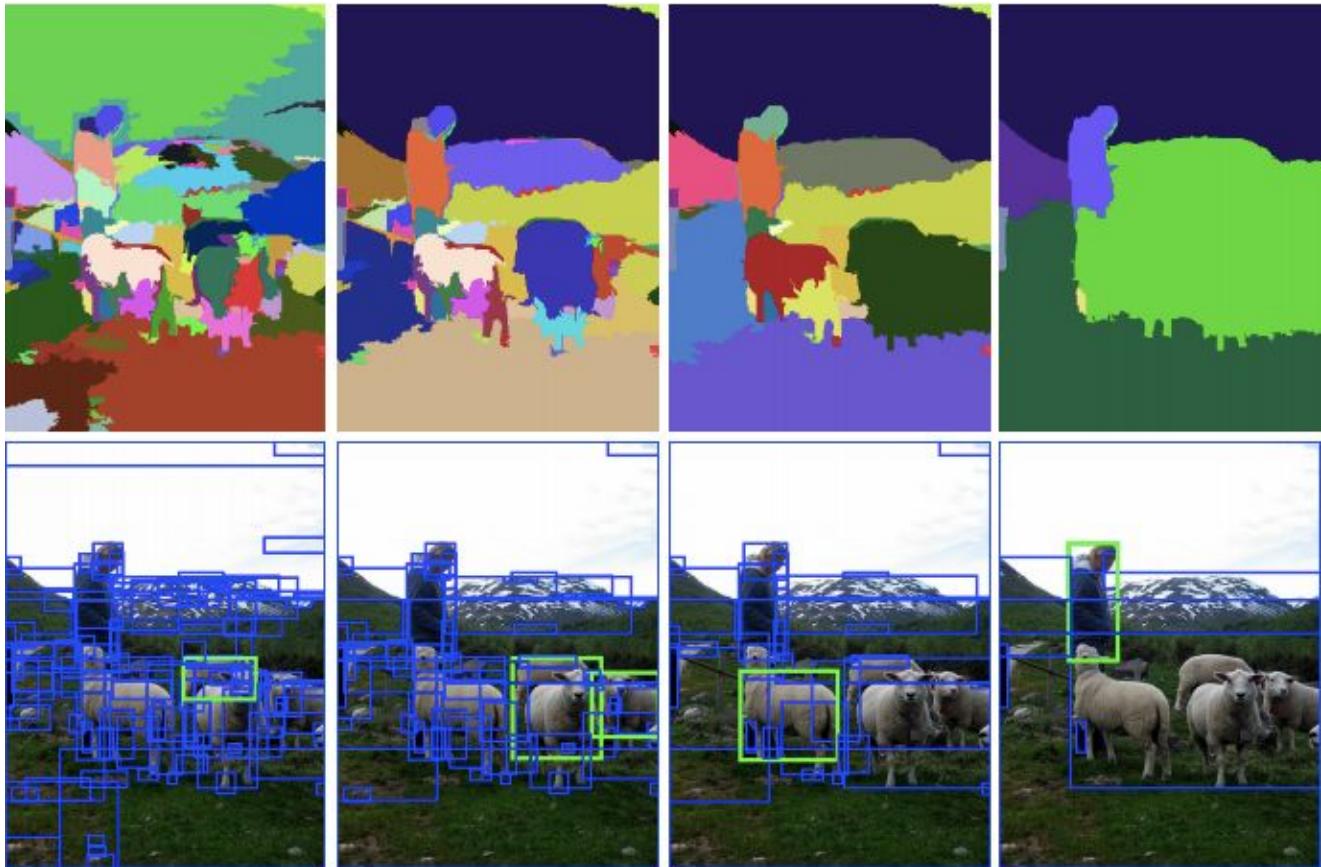
2. Extract region proposals (~2k)



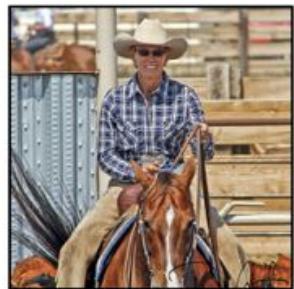
3. Compute CNN features

4. Classify regions

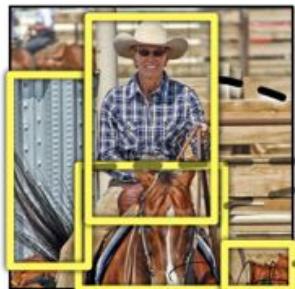
Selective search: fewer proposals



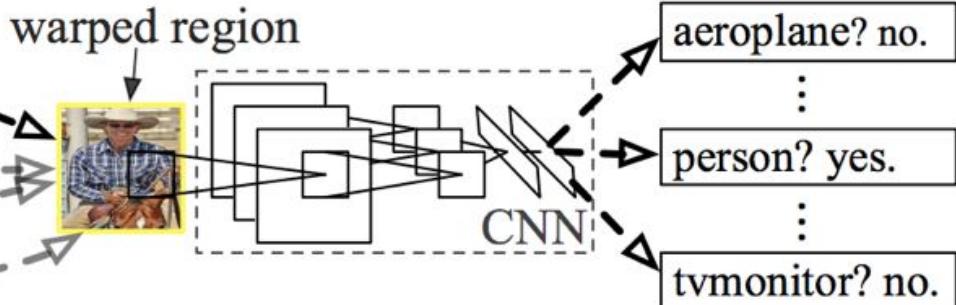
R-CNN: Regions with CNN features



1. Input image



2. Extract region proposals (~2k)



3. Compute CNN features

4. Classify regions

Много постпроцессинга, 20 секунд/на изображение

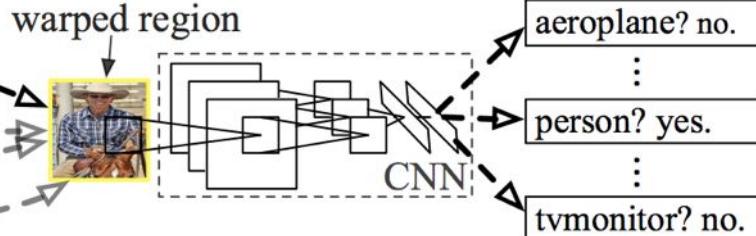
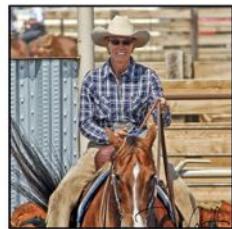
Pascal VOC:

AlexNet

53.3% mAP

VGG-16

62.4% mAP

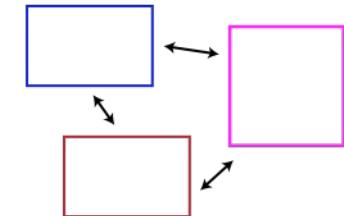
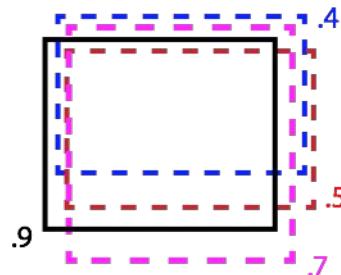
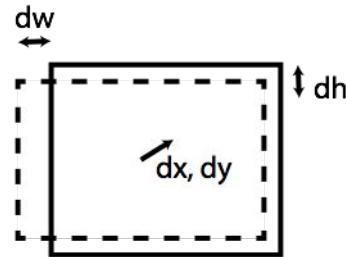


1. Input image

2. Extract region proposals (~2k)

3. Compute CNN features

4. Classify regions



5. Bounding box regression

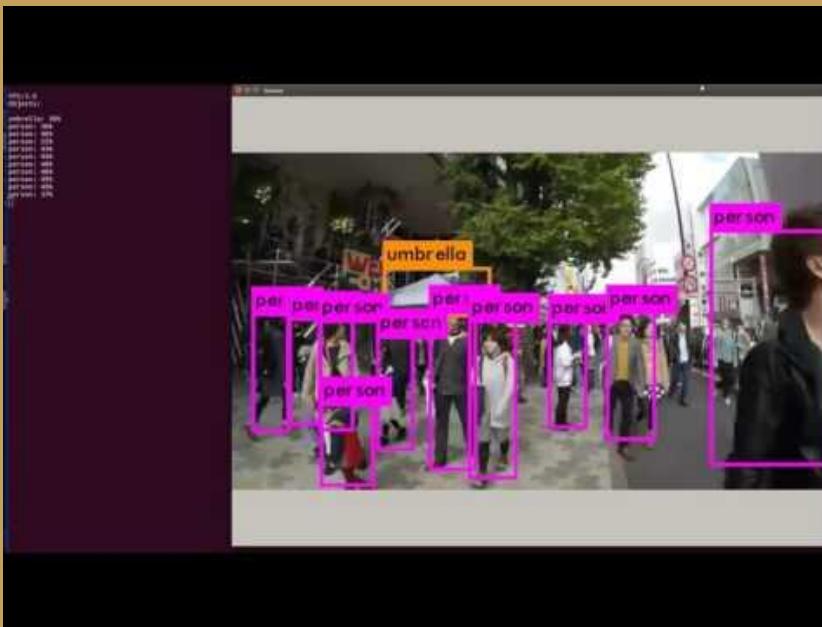
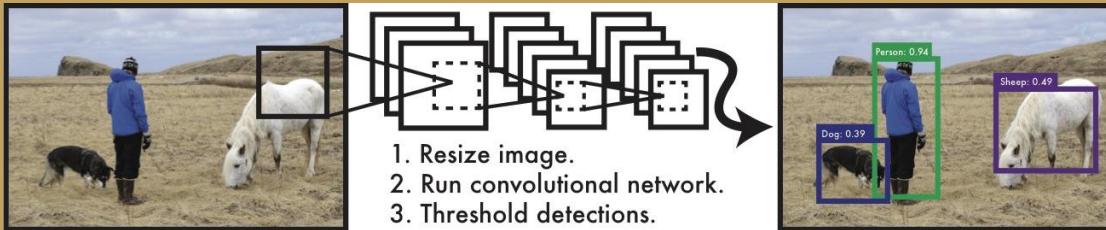
6. Non-max suppression

7. RNN rescoring??



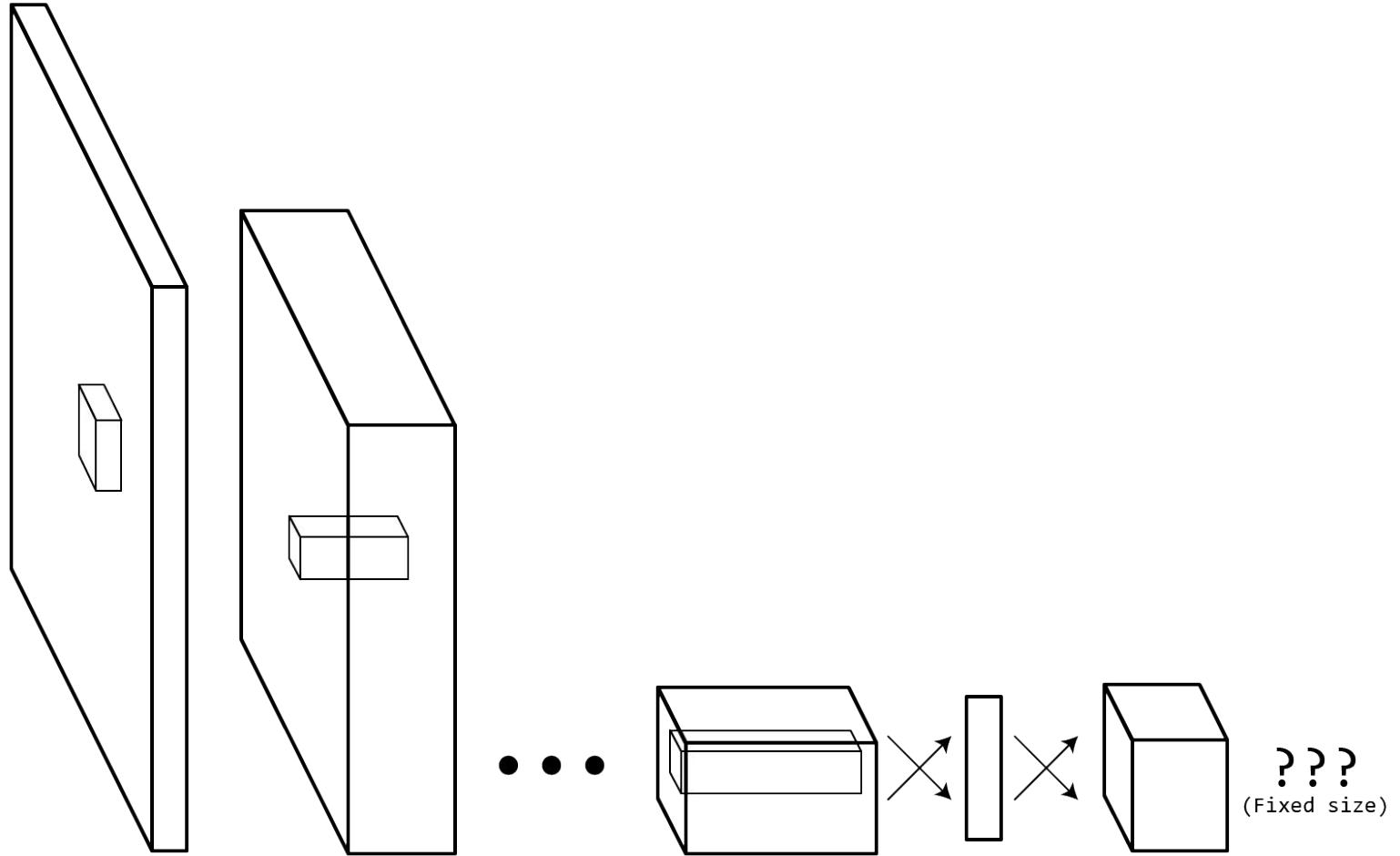
YOLO

YOLO



	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img

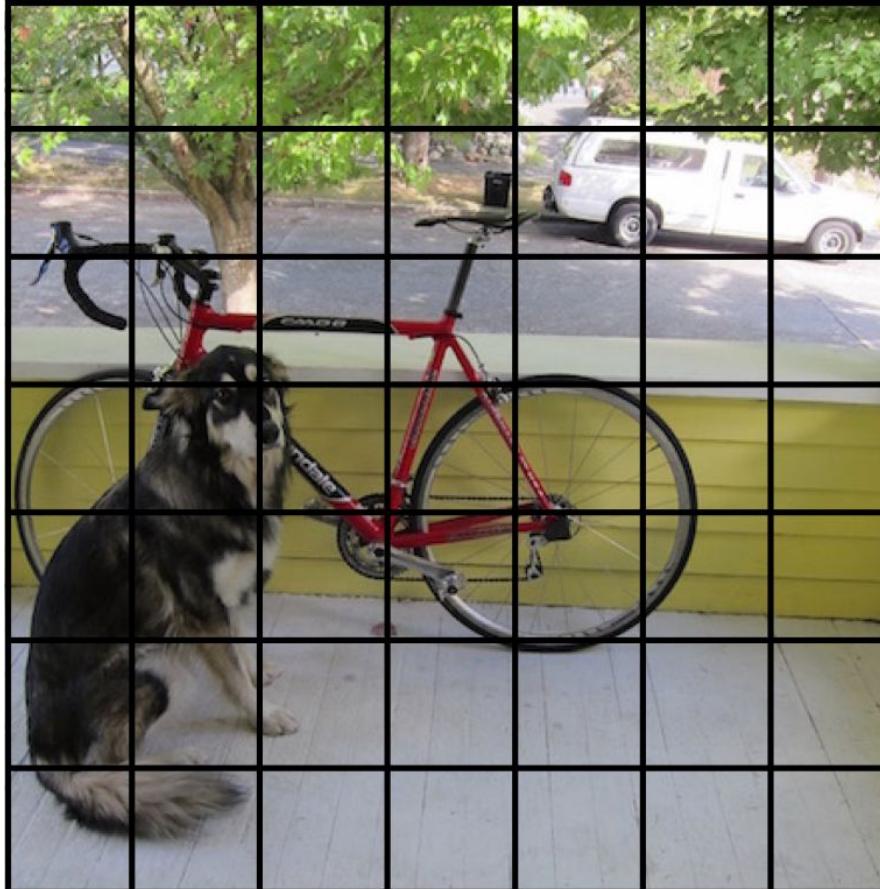
	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
YOLO	63.4	45 FPS	22 ms/img



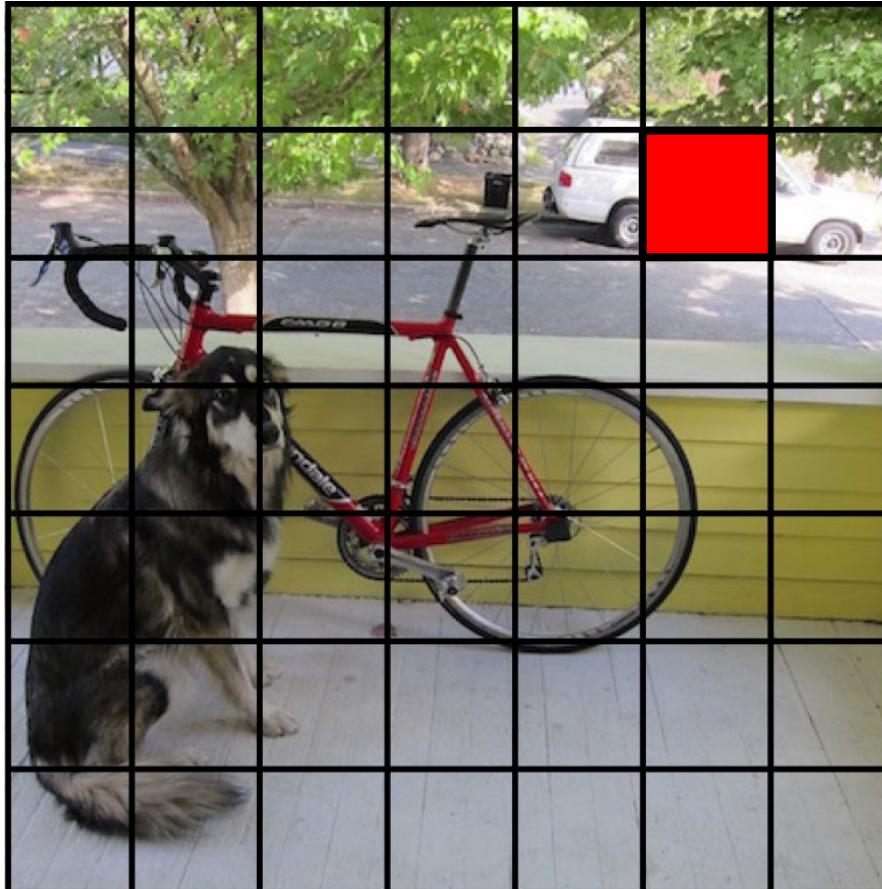
Исходное изображение



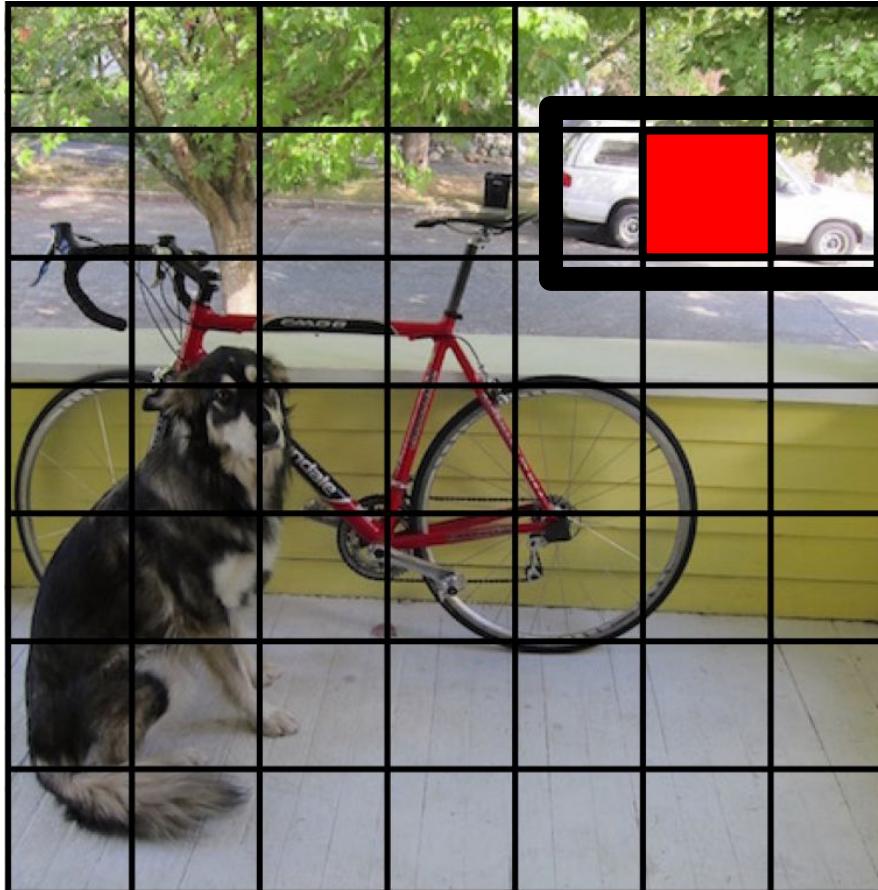
Делим его на блоки



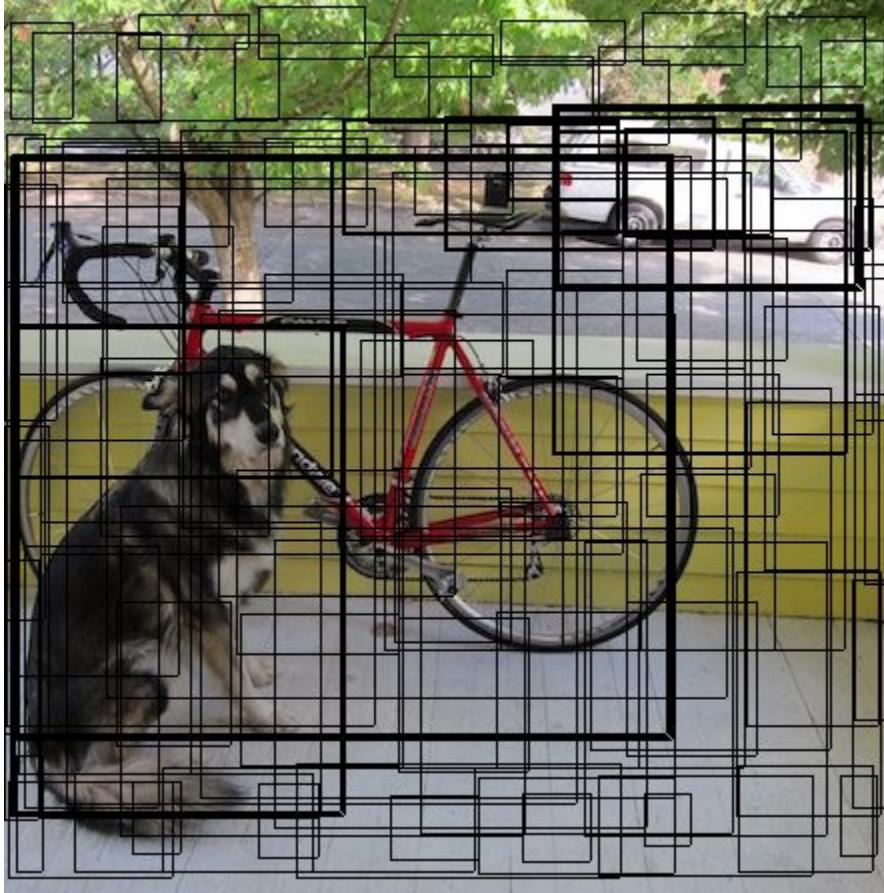
Для каждого блока предсказываем $P(\text{obj})$



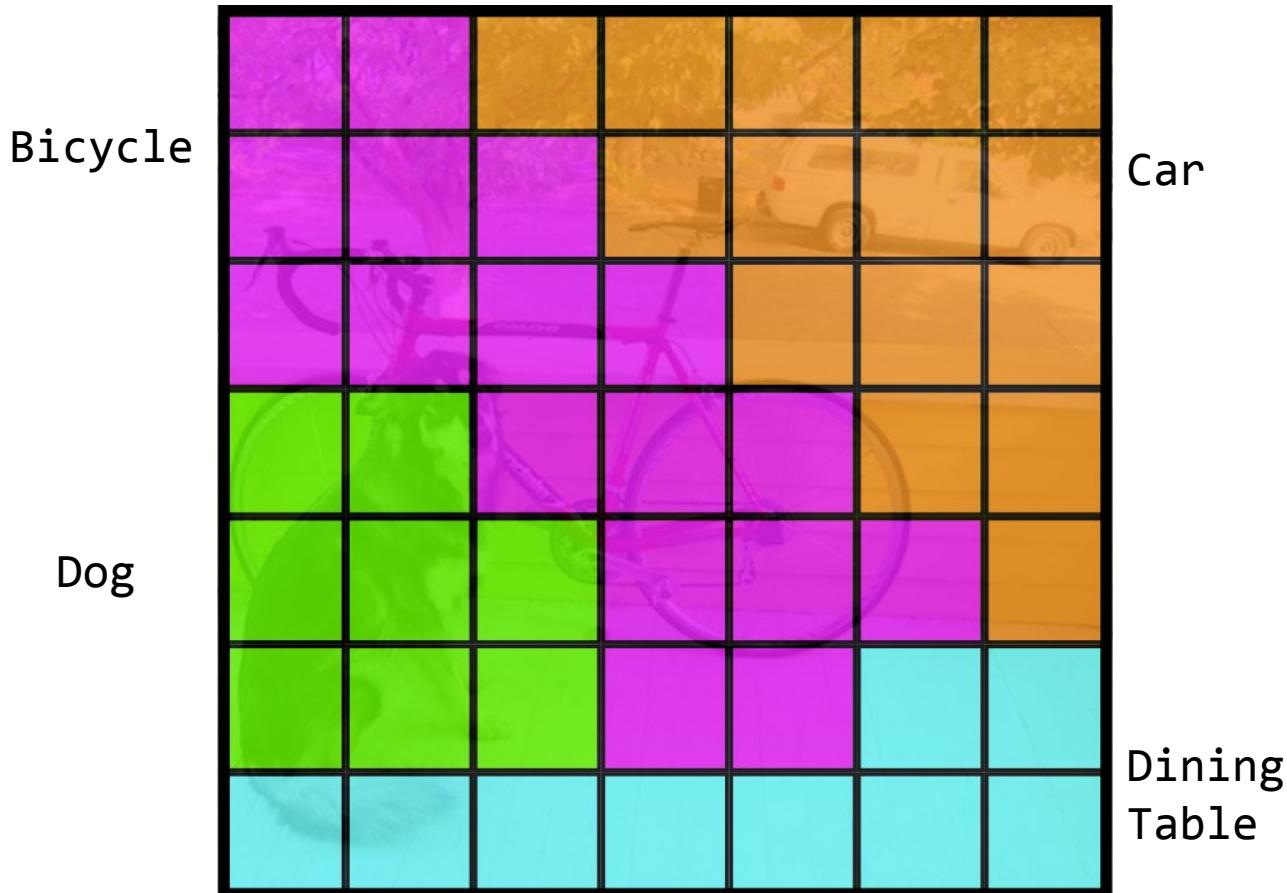
Также предсказываем bounding box



Также предсказываем bounding box



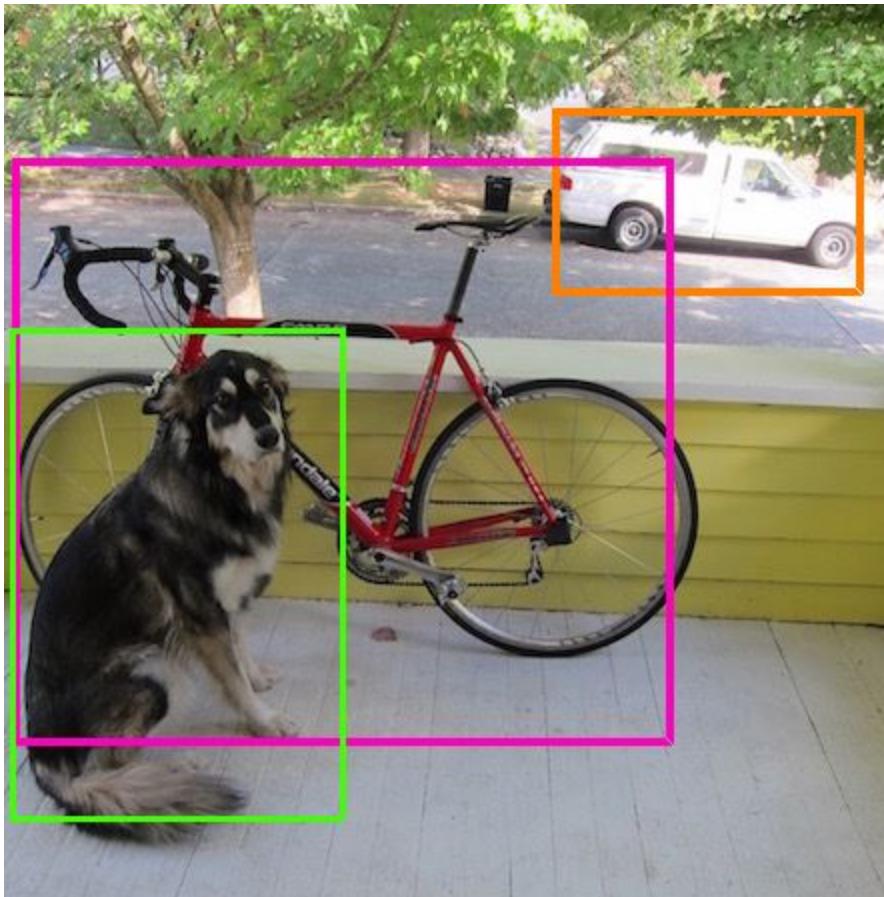
Также предсказываем вероятности классов



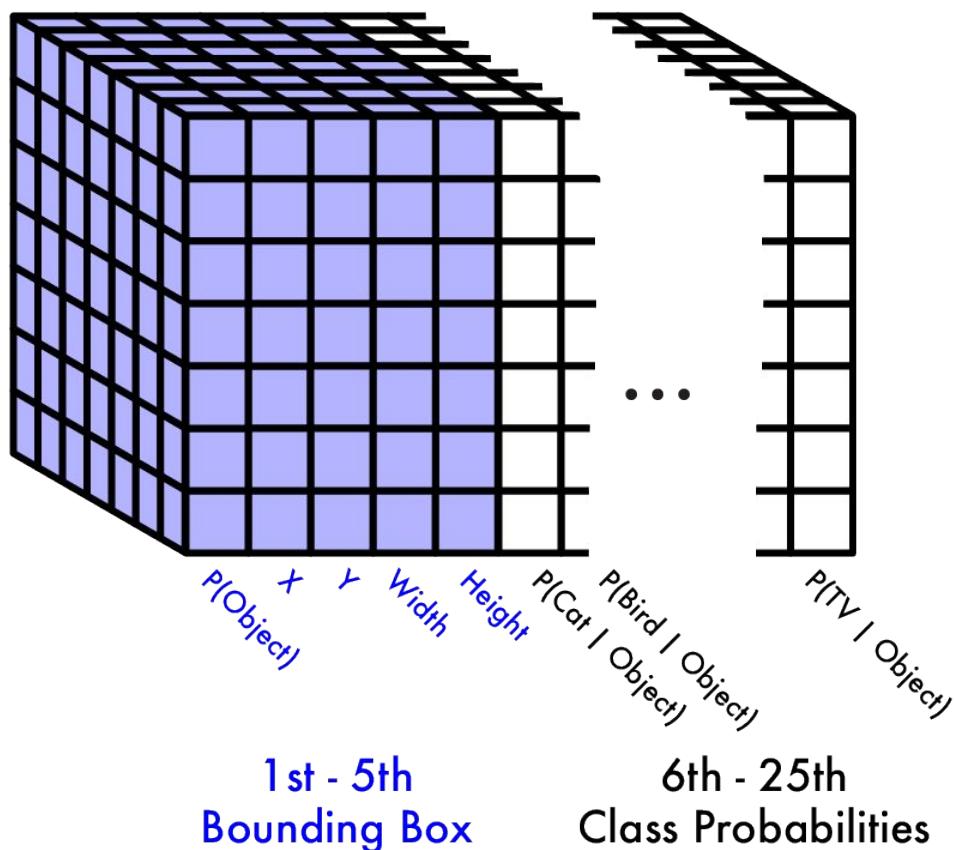
Также предсказываем вероятности классов



Thresholding и Non-Maximum-Suppression

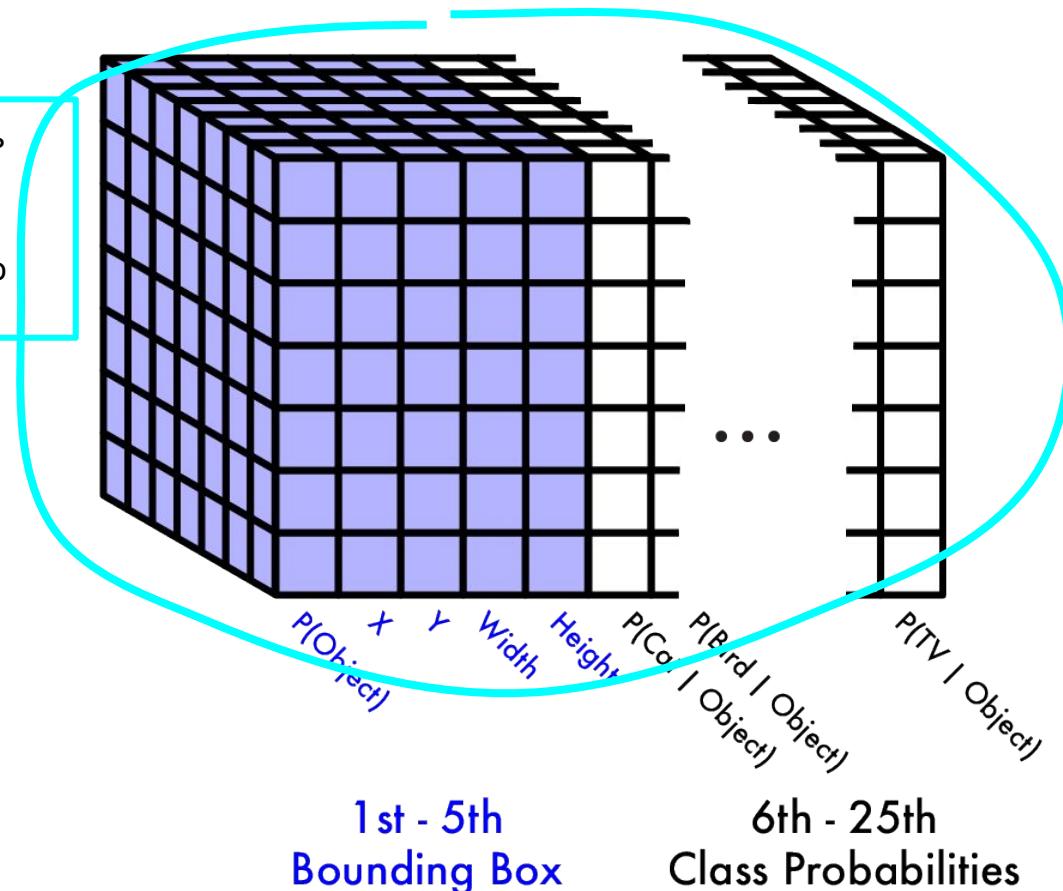


Тензор, который мы предсказываем



Тензор, который мы предсказываем

Можем предсказывать
несколько боксов,
достаточно застэкать
этот тензор несколько
раз



Много боксов на каждую клетку

Будет предсказывать 5 боксов для каждой клетки

Хотим пространственное разнообразие
между боксами

Большие, средние, маленькие

Не предсказываем боксы напрямую

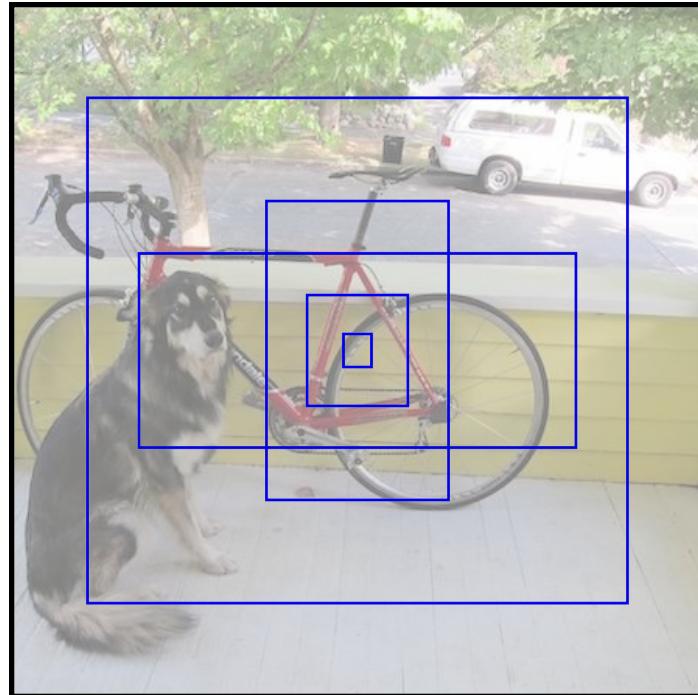
Предсказываем отклонение
от приорных значений

Приорные значения должны быть разумными

В YOLO используется k-means clustering

баундинг боксов из тренировочной

выборки (w,h)



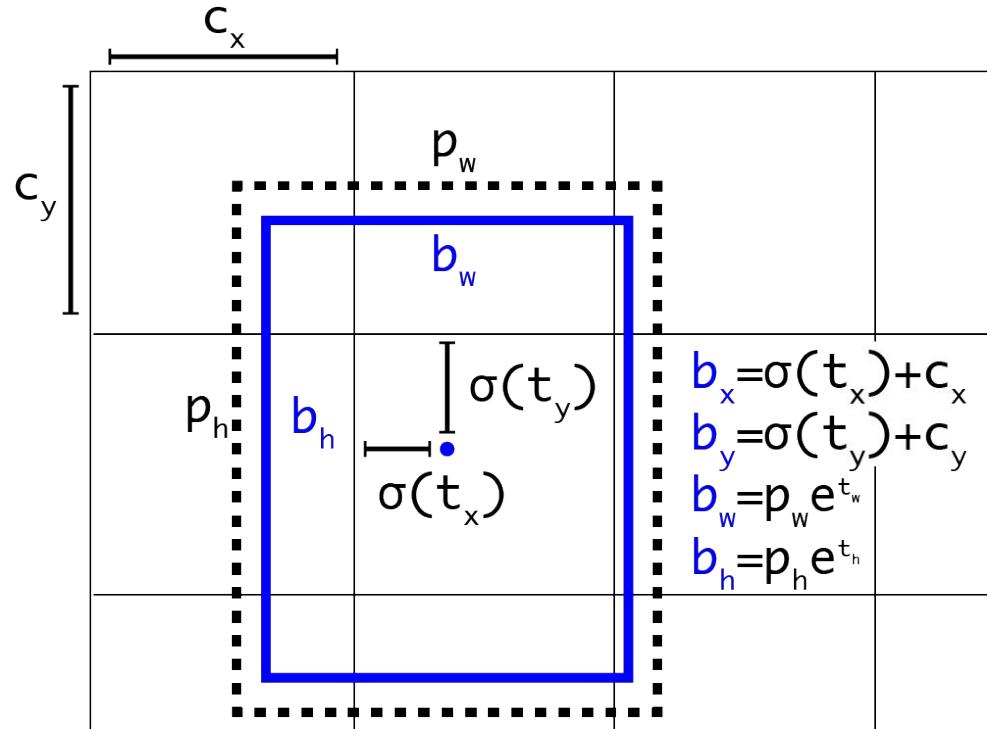
Bounding box encoding (YOLOv2 и v3)

Зная координаты клетки c_x и c_y

Приорные значения

ширины и высоты (p_w и p_h)

Предсказываем баундинг бокс
как:



YOLO loss function

$$L(\text{YOLO}) = \alpha_1 L(\text{confidence}) + \alpha_2 L(\text{localization}) + \alpha_3 L(\text{classification})$$

Можно подобрать значения всех alphas

$L(\text{confidence})$: binary cross-entropy

$L(\text{localization})$: RMSE

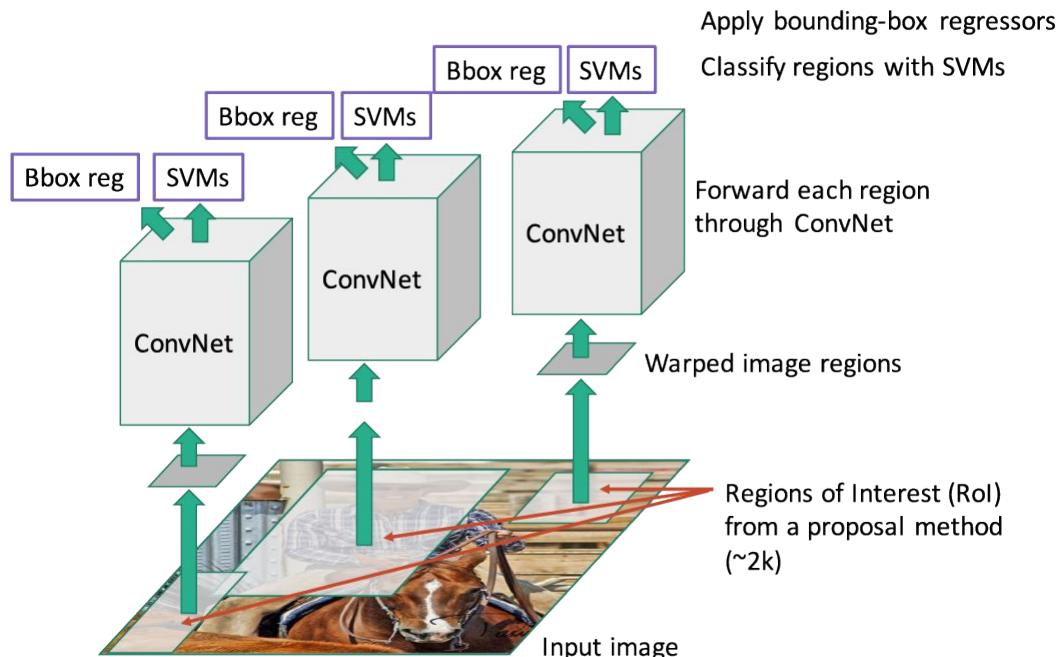
$L(\text{classification})$: multi-class cross-entropy или
1 v all binary cross-entropy

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
YOLO	63.4	45 FPS	22 ms/img
YOLOv2	78.6	40 FPS	25 ms/img
YOLOv3	83.1	30 FPS	33 ms/img

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
YOLO	63.4	45 FPS	22 ms/img
YOLOv2	78.6	40 FPS	25 ms/img
YOLOv3	83.1	30 FPS	33 ms/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img
Resnet FRCNN	76.4	??	??
ResNet + COCO data	83.8	??	??
R-FCN	83.6	6 FPS	170 ms/img

R-CNN медленная

Сетка проходит по каждому ROI независимо

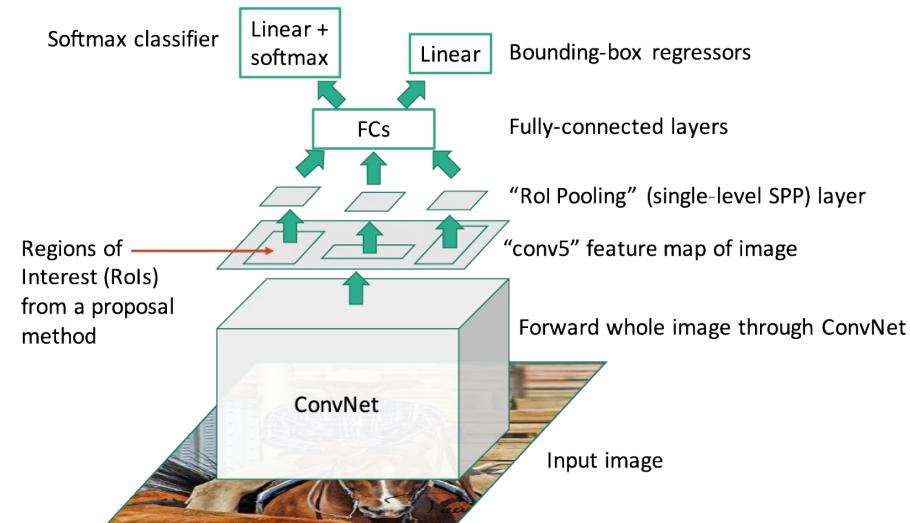


Fast R-CNN

Прогоняем через сетку все изображение один раз, извлекаем ROI из feature map, а не из исходного изображения

ROI Pool:

Конвертируем ROI различного размера
к тензору фиксированного размера



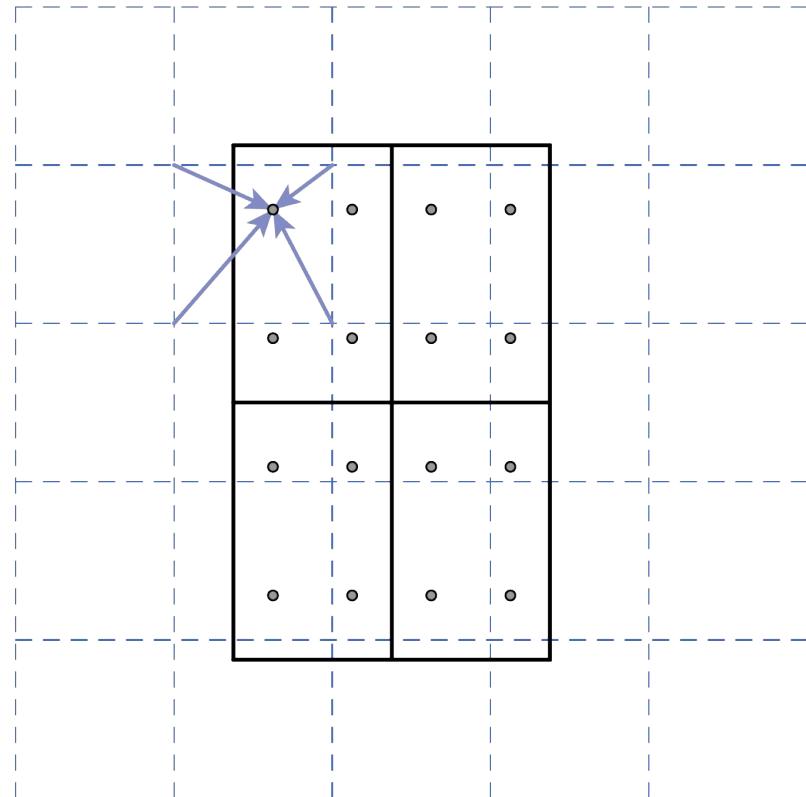
ROI Align

Лучше чем ROI Pool,
поэтому рассмотрим его поподробнее

Делим ROI сеткой
фиксированного размера

Делаем билинейную интерполяцию

Делаем над этим пулинг (max, avg, ...)



Fast R-CNN

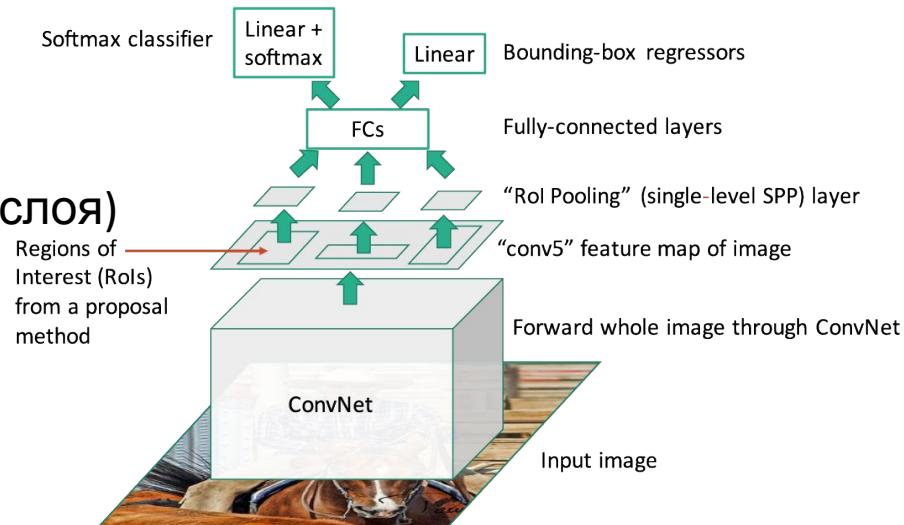
Прогоняем через сетку все изображение один раз, извлекаем ROI из feature map, а не из исходного изображения

ROI Pool:

Конвертируем ROI различного размера
к аутпуту фиксированного размера

Намного быстрее, только один прогон
через сетку (кроме последнего линейного слоя)

Все еще медленный region proposer,
Selective search занимает 2 сек

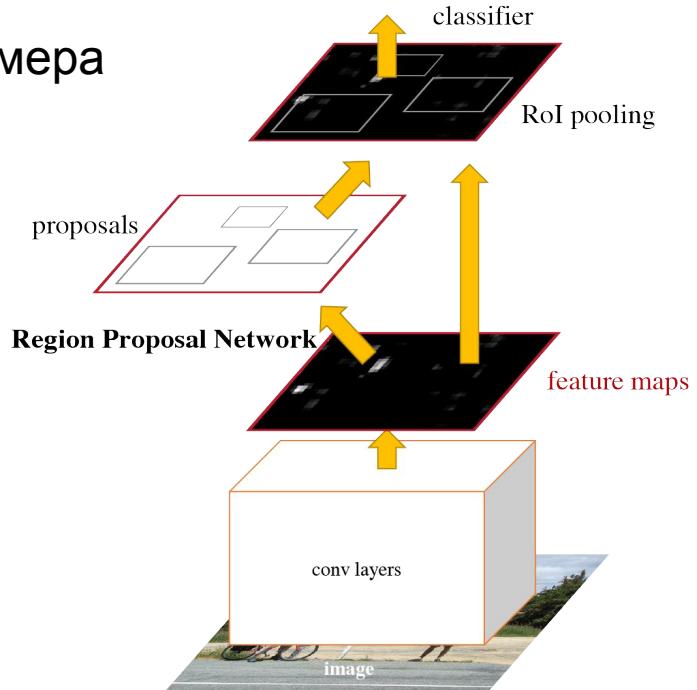


Faster R-CNN

Используем сетку, для нахождения ROI и генерации фичей

ROI Pool для того чтобы добиться одинакового размера у всех ROI.

Дополнительные слои для классификации и предсказания боксов для ROI



Победили PASCAL VOC, нужно больше данных

Average Precision (AP %)

	mean	aero plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table	dog	horse	motor bike	person	potted plant	sheep	sofa	train	tv/ monitor	submission date
	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	▼	
▶ FOCAL_DRFCN(VOC+COCO, single model) [?]	88.8	95.0	93.3	91.8	82.9	81.9	91.6	93.0	97.1	76.7	92.5	71.7	96.2	94.9	94.2	93.7	75.3	93.3	80.0	94.7	85.4	01-Mar-2018
▶ R4D_faster_rcnn [?]	88.6	94.6	92.3	91.3	82.3	79.4	91.8	91.8	97.4	76.6	93.6	75.3	97.0	94.6	93.5	92.6	75.1	92.0	80.9	94.4	86.5	20-Nov-2016
▶ R-FCN, ResNet Ensemble(VOC+COCO) [?]	88.4	94.8	92.9	90.6	82.4	81.8	89.9	91.7	97.1	76.0	93.4	71.9	96.6	94.3	93.9	92.8	75.7	91.9	80.8	93.6	86.4	09-Oct-2016
▶ HIK_FRCN [?]	87.9	95.0	93.2	91.3	80.3	77.7	90.6	89.9	97.8	72.8	93.7	70.7	97.2	95.4	94.0	91.8	72.7	92.8	81.1	94.1	86.2	19-Sep-2016
▶ ** VIM_SSD ** [?]	87.6	95.3	92.0	88.7	81.6	78.5	91.4	93.2	95.7	74.9	91.6	73.5	94.2	93.0	93.2	93.0	70.5	93.0	79.1	94.3	85.0	11-May-2018
▶ ** Deformable R-FCN, ResNet-101 (VOC+COCO) ** [?]	87.1	94.0	91.7	88.5	79.4	78.0	89.7	90.8	96.9	74.2	93.1	71.3	95.9	94.8	93.2	92.5	71.7	91.8	78.3	93.2	83.3	23-Mar-2017
▶ RefineDet (VOC+COCO,single model,VGG16,one-stage) [?]	86.8	94.7	91.5	88.8	80.4	77.6	90.4	92.3	95.6	72.5	91.6	69.9	93.9	93.5	92.4	92.6	68.8	92.4	78.5	93.6	85.2	16-Mar-2018
▶ FasterRcnn-ResNeXt101(COCO+07++12, single model) [?]	86.8	93.9	93.4	88.3	80.2	72.6	89.4	89.3	96.8	73.0	91.5	72.3	95.4	94.5	93.8	91.7	70.7	90.6	81.2	92.6	83.9	04-May-2017
▶ ** PSSNet(VOC+COCO) ** [?]	85.5	92.4	91.4	85.9	78.6	75.8	88.0	89.8	95.2	72.4	87.8	72.2	94.0	92.7	93.2	92.3	70.7	88.8	76.1	92.1	81.2	30-Mar-2018
▶ R-FCN, ResNet (VOC+COCO) [?]	85.0	92.3	89.9	86.7	74.7	75.2	86.7	89.0	95.8	70.2	90.4	66.5	95.0	93.2	92.1	91.1	71.0	89.7	76.0	92.0	83.4	09-Oct-2016
▶ ** MONet(VOC+COCO) ** [?]	84.3	92.4	90.5	84.7	75.4	71.6	87.2	88.9	94.6	70.5	86.9	71.0	92.3	91.8	90.8	91.7	69.8	89.1	75.1	91.3	79.6	01-Apr-2018
▶ PVANet+ [?]	84.2	93.5	89.8	84.1	75.6	69.7	88.2	87.9	93.4	70.0	87.7	75.3	92.9	90.5	90.9	90.2	67.3	86.4	80.3	92.0	78.8	26-Oct-2016
▶ FSSD512 [?]	84.2	92.8	90.0	86.2	75.9	67.7	88.9	89.0	95.0	68.8	90.9	68.7	92.8	92.1	91.4	90.2	63.1	90.1	76.9	91.5	82.7	07-Nov-2017
▶ BlitzNet512 [?]	83.8	93.1	89.4	84.7	75.5	65.0	86.6	87.4	94.5	69.9	88.8	71.7	92.5	91.6	91.1	88.9	61.2	90.4	79.2	91.8	83.0	19-Jul-2017
▶ PFPNet512 VGG16 07++12+COCO [?]	83.8	93.0	89.9	85.1	75.8	66.4	88.4	88.3	94.0	67.9	89.5	69.7	92.0	91.8	91.6	88.7	61.1	89.1	78.4	90.5	84.3	18-Oct-2017
▶ Faster RCNN, ResNet (VOC+COCO) [?]	83.8	92.1	88.4	84.8	75.9	71.4	86.3	87.8	94.2	66.8	89.4	69.2	93.9	91.9	90.9	89.6	67.9	88.2	76.8	90.3	80.0	10-Dec-2015

Common Objects in COntext (COCO)

80 объектов

117,261 train/val изображений

902,435 инстансов

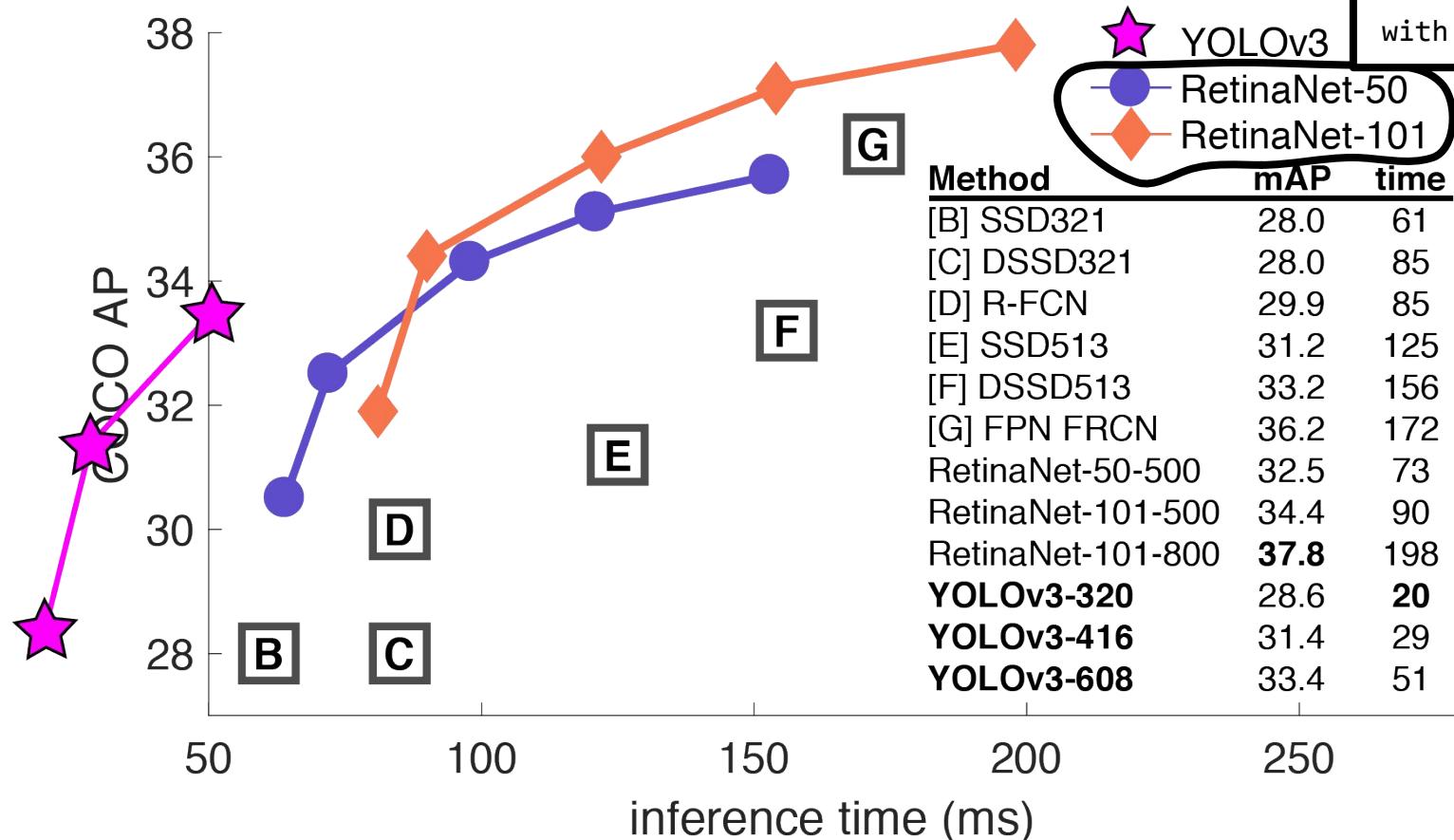
Новая метрика детекции, тAP усредненное по IOU [.5 - .95]

Маска сегментации для каждого инстанса

В оригинале создано Майкрософтом, но они испугались копирайта и в итоге теперь open source



Performance (COCO)



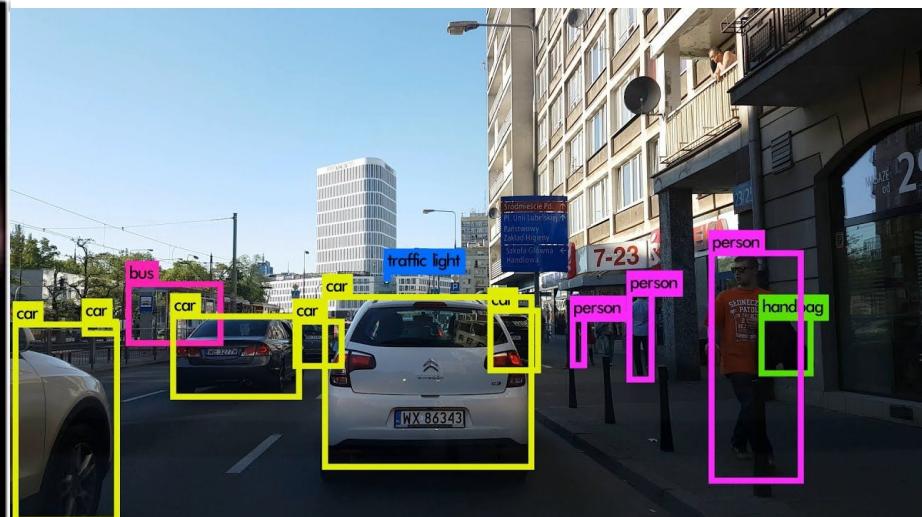
Segmentation

Detection

Лейблы пикселей
Только категории



Лейблы боксов
Категории + инстансы

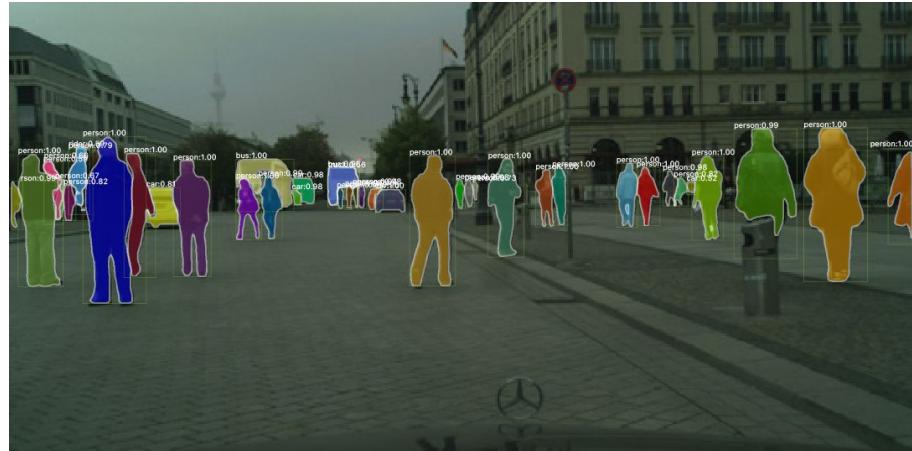
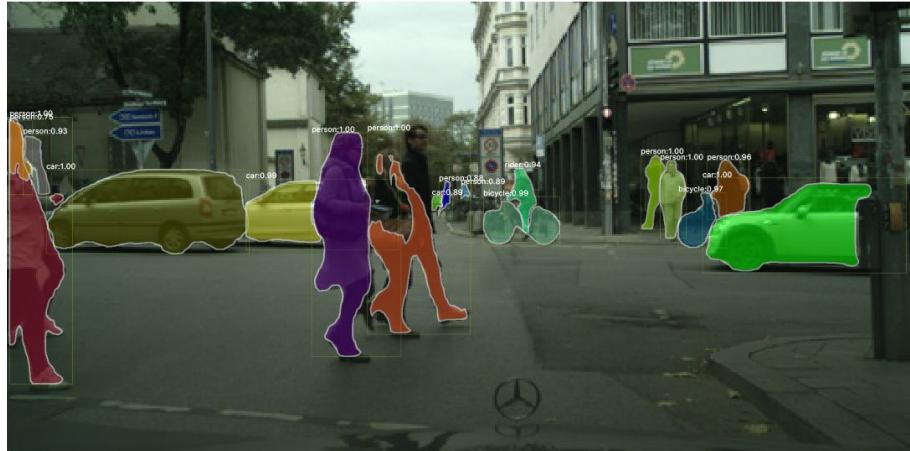


Instance Segmentation

На входе изображение, на выходе instance-level segmentation

К какому классу относится каждый пиксель

А также к какому инстансу

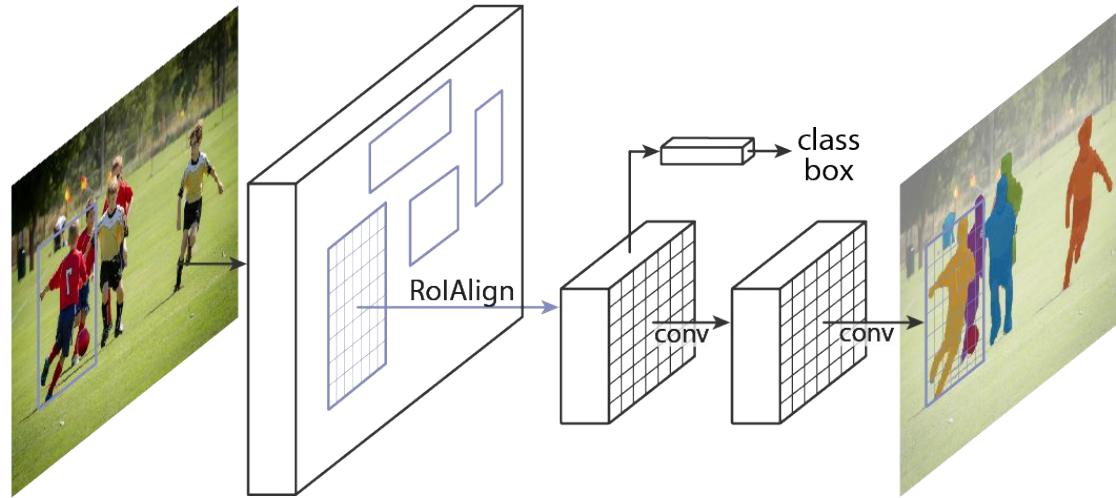


Mask R-CNN Demo



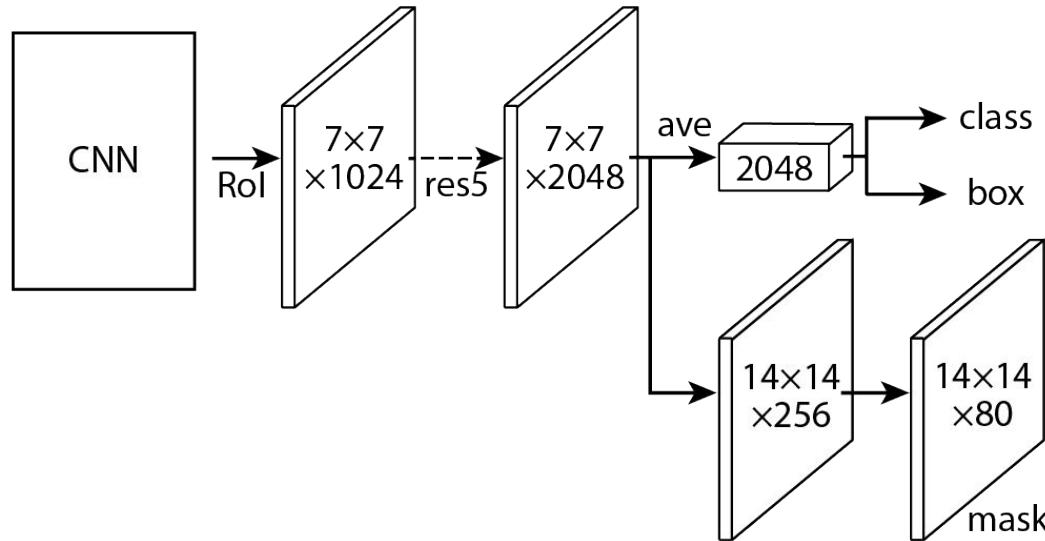
Mask R-CNN

Похожа на R-CNN но предсказывает маску и бокс



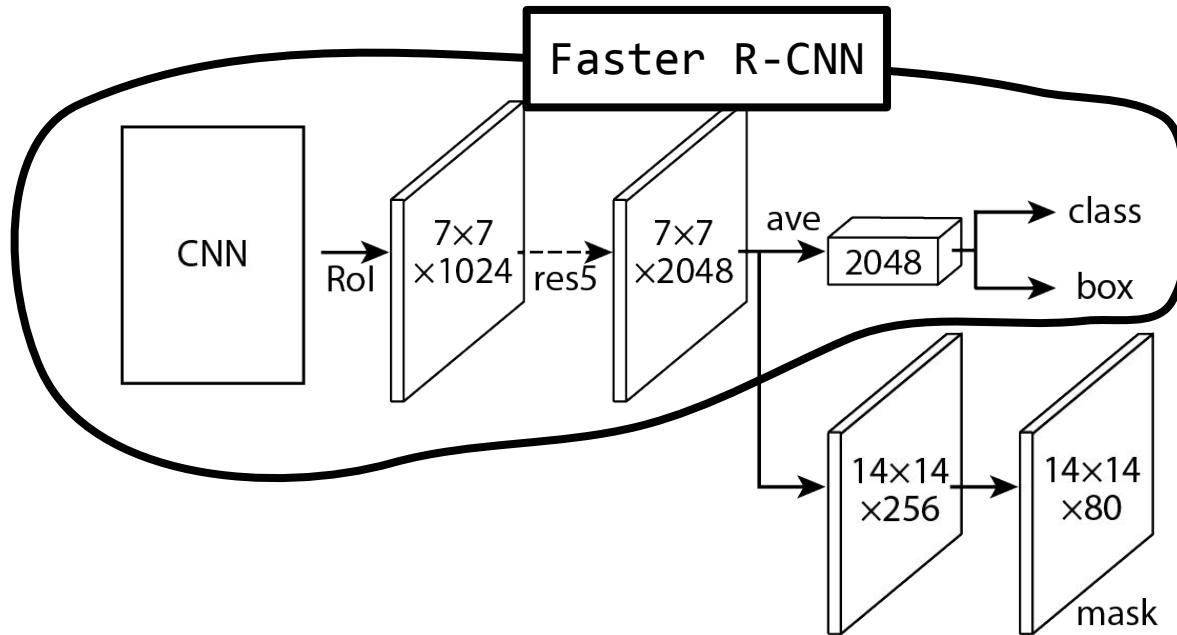
Mask R-CNN

Похожа на R-CNN но предсказывает маску и бокс



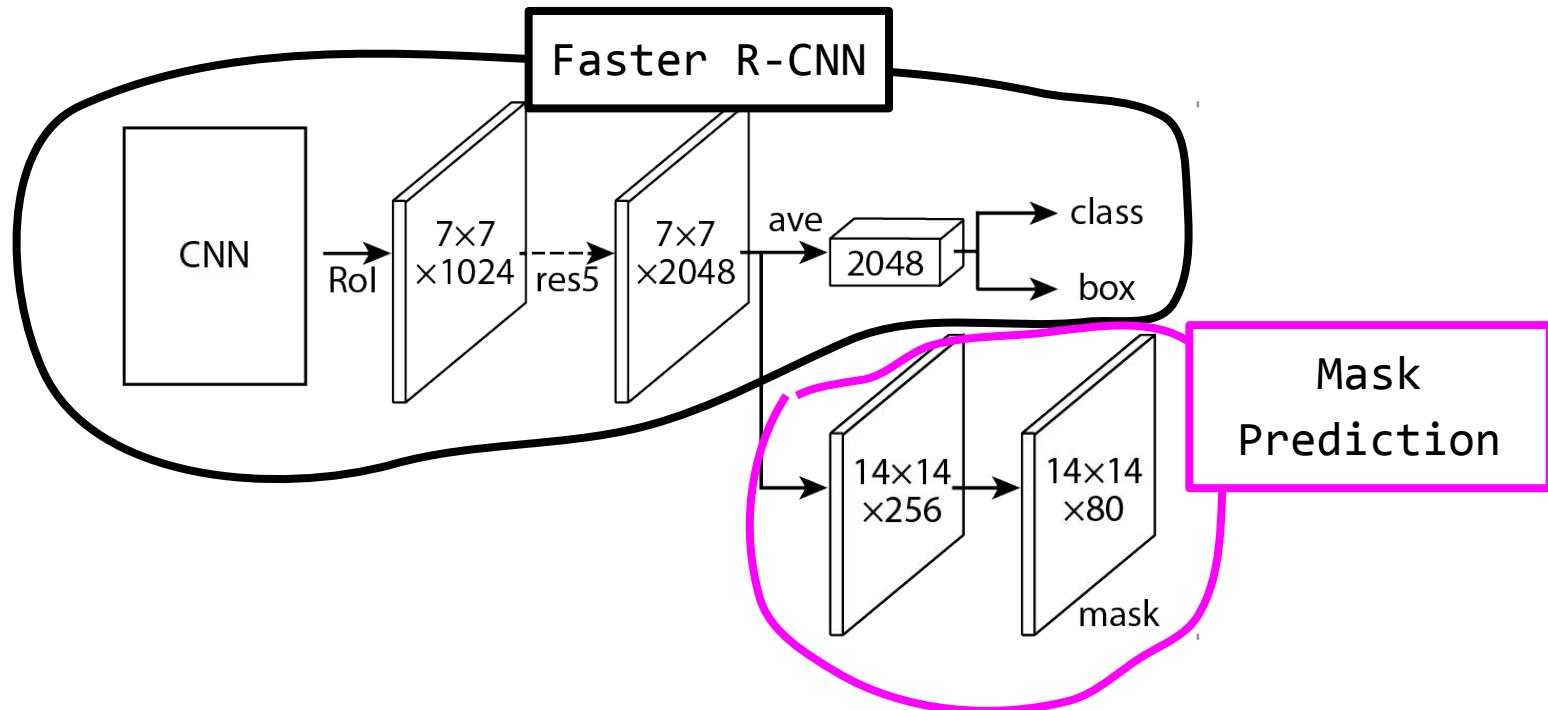
Mask R-CNN

Похожа на R-CNN но предсказывает маску и бокс



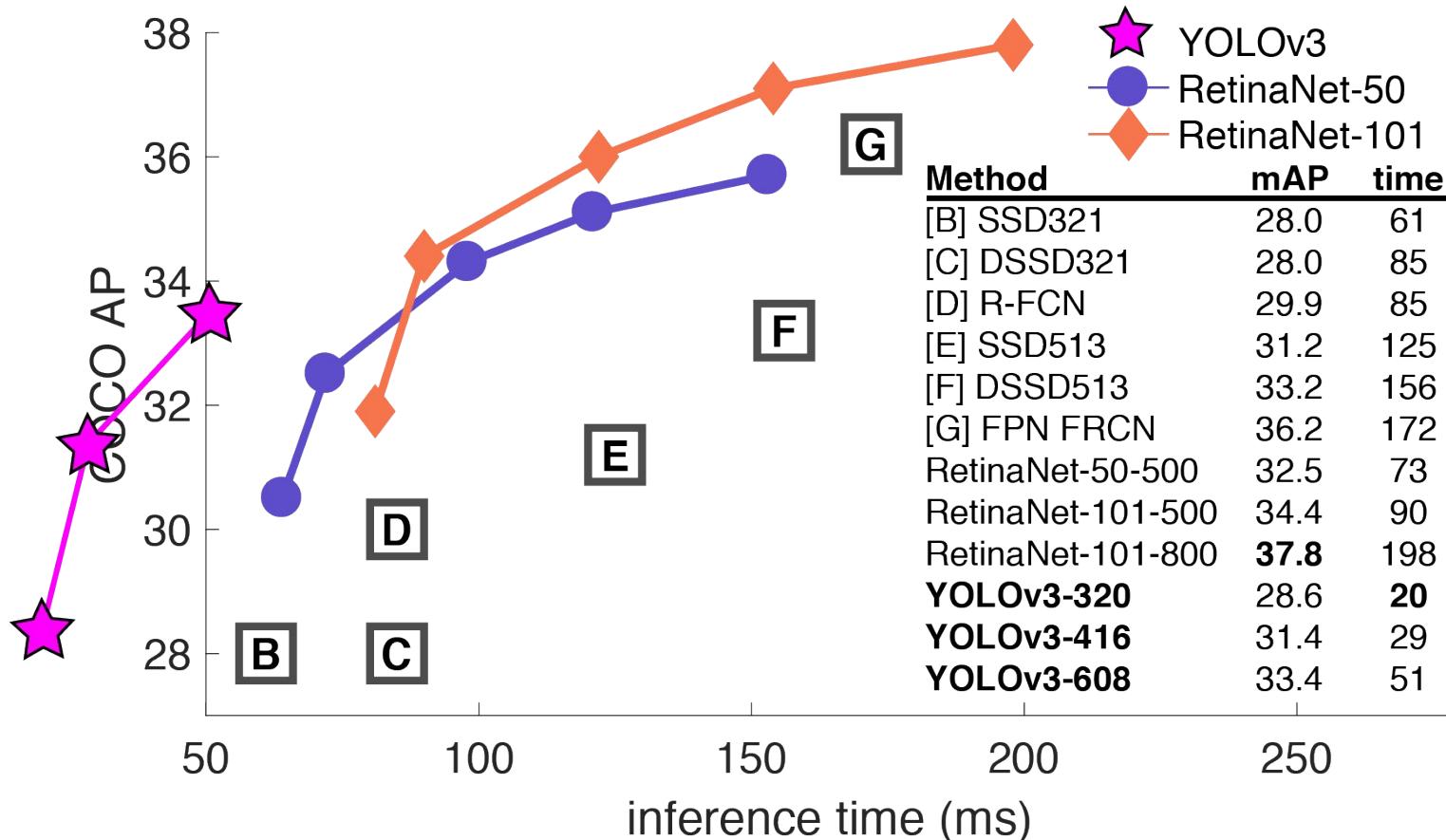
Mask R-CNN

Похожа на R-CNN но предсказывает маску и бокс



Mask R-CNN on COCO: 41 mAP

Mask R-CNN
где-то там



Segmentation, Detection, Instance Segmentation



В COCO датасете еще есть подписи!

5 подписей на изображений

Детекция/сегментация это
(скорей всего) просто
сопоставление паттернов

Чтобы подписать (caption)
изображение, сетке нужно
во-настоящему понять его

Необходимо моделировать и
визуальную информацию, и язык



The man at bat readies to swing at the pitch while the umpire looks on.



A large bus sitting next to a very tall building.



A horse carrying a large load of hay and two people sitting on it.



Bunk bed with a narrow shelf sitting underneath it.



It's coding time