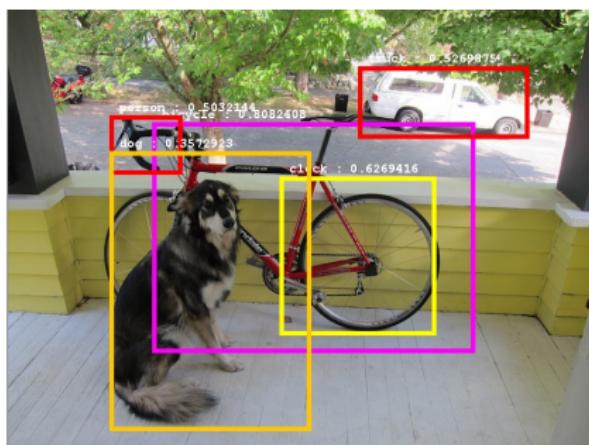
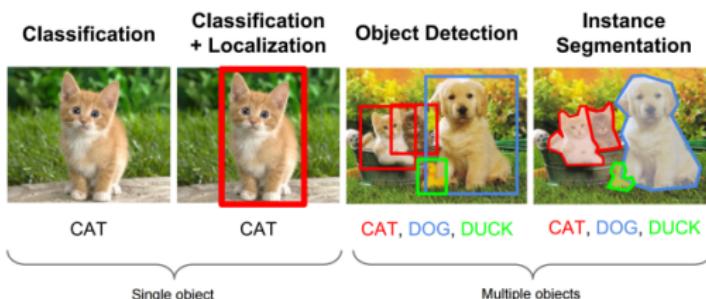


Детекция объектов, двухстадийные детекторы

Виктор Китов
victorkitov.github.io



Локализация объектов



- Локализация: не только классифицировать объект (единственный), но и выделить рамкой.
 - два выхода - вероятности классов и (x, y, w, h)
- Детекция: объектов может не быть или быть несколько разных классов.
- Instance-сегментация: выделить не рамкой, а маской.

Приложения: подсчёт числа людей

Подсчёт числа людей:

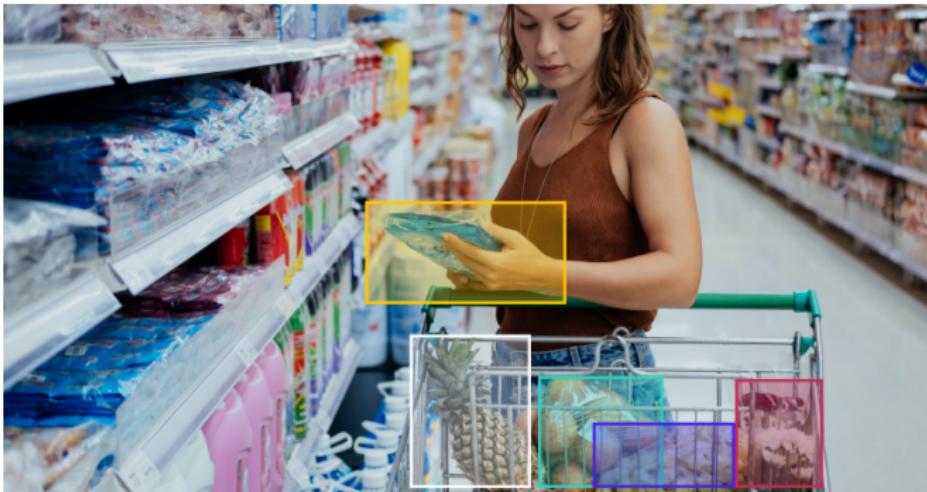
- оценка популярности мероприятий, управление потоками, чтобы не было заторов



Приложения: поведение покупателей в магазине

Поведение покупателей в магазине:

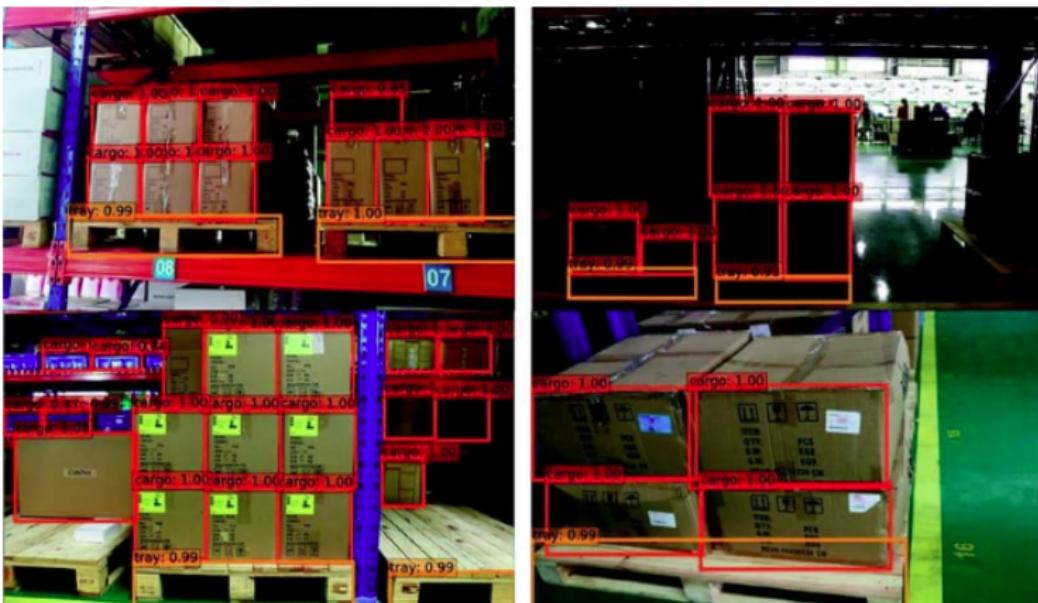
- безопасность, оптимизация расположения товаров на полках, автоматическое выписывание чека



Приложения: управление складом

Управление складом:

- распределение товаров по складу, учёт товаров.



Приложения: безопасность

Безопасность:

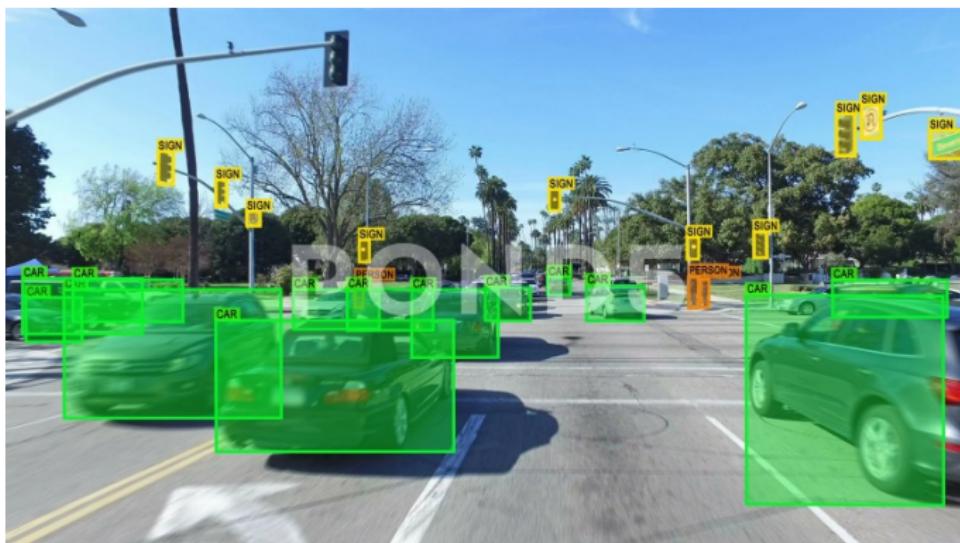
- обнаружение запрещенных предметов, незаконных действий, краж, идентификация человека



Приложения: self-driving cars

Self-driving cars:

- обнаружение светофоров, знаков, съездов, стоянок, пешеходов, др. машин



Приложения



ball tracking



object identification

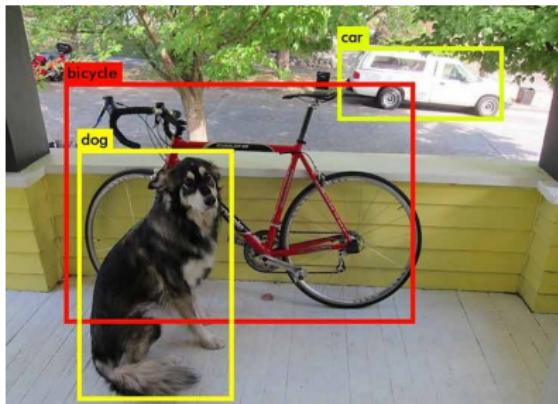


activity recognition



face identification,
emotion identification

Постановка задачи



Обучающая выборка:

- img1.jpg: X(1), Y(1), X(2), Y(2), класс, ... X(1), Y(1), X(2), Y(2), класс
- img2.jpg: X(1), Y(1), X(2), Y(2), класс, ... X(1), Y(1), X(2), Y(2), класс
-

Популярные датасеты для детекции объектов

Dataset	train		validation		trainval		test	
	images	objects	images	objects	images	objects	images	objects
VOC-2007	2,501	6,301	2,510	6,307	5,011	12,608	4,952	14,976
VOC-2012	5,717	13,609	5,823	13,841	11,540	27,450	10,991	-
ILSVRC-2014	456,567	478,807	20,121	55,502	476,688	534,309	40,152	-
ILSVRC-2017	456,567	478,807	20,121	55,502	476,688	534,309	65,500	-
MS-COCO-2015	82,783	604,907	40,504	291,875	123,287	896,782	81,434	-
MS-COCO-2018	118,287	860,001	5,000	36,781	123,287	896,782	40,670	-
OID-2018	1,743,042	14,610,229	41,620	204,621	1,784,662	14,814,850	125,436	625,282

Метод скользящего окна (sliding window)



Скользим окном и смотрим на корреляцию с образцом.

- важно сопоставлять в разных масштабах

Ограничения:

Метод скользящего окна (sliding window)



Скользим окном и смотрим на корреляцию с образцом.

- важно сопоставлять в разных масштабах

Ограничения:

- разнообразие образца (марка машины, одежда человека)
- сильно влияют цвет, ракурс, освещение.

Содержание

1 Меры качества

2 Двухстадийные детекторы

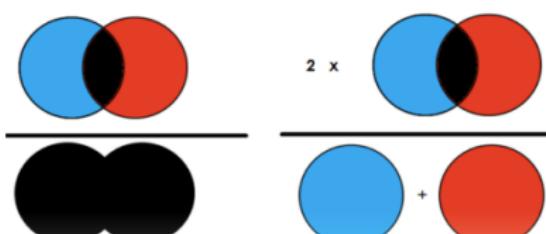
Меры качества

- Intersection over union (IoU) = близость Жаккарда (в детекции - только для прямоугольников)

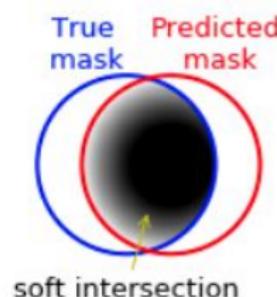
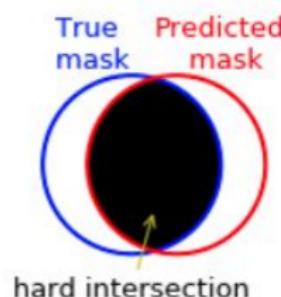
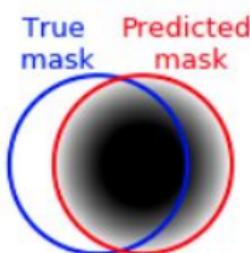
$$IoU = \frac{TP}{TP + FP + TN} = \frac{a}{b}$$

$$Dice = \frac{TP + TP}{TP + TP + FP + TN} = \frac{2a}{a + b}$$

$$Dice = \frac{\frac{2a}{b}}{\frac{a+b}{b}} = \frac{2 \cdot \frac{a}{b}}{\frac{a}{b} + 1} = \frac{2 \cdot IoU}{IoU + 1}$$



Сглаженные варианты



$$\text{IoU}_{soft}^1 = \frac{\langle Y, P \rangle}{\|Y\|_1 + \|P\|_1 - \langle Y, P \rangle}; \quad \text{Dice}_{soft}^1 = \frac{2\langle Y, P \rangle}{\|Y\|_1 + \|P\|_1}$$

$$\text{IoU}_{soft}^2 = \frac{\langle Y, P \rangle}{\|Y\|_2^2 + \|P\|_2^2 - \langle Y, P \rangle}; \quad \text{Dice}_{soft}^2 = \frac{2\langle Y, P \rangle}{\|Y\|_2^2 + \|P\|_2^2}$$

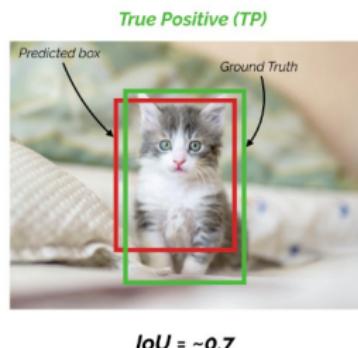
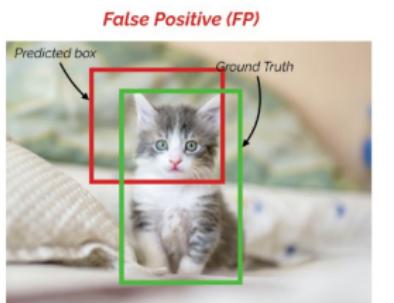
Меры качества

- Точность= TP/\hat{P} , Полнота= TP/P

$$\text{Precision} = \frac{\text{Green}}{\text{Red} + \text{Green}}$$


$$\text{Recall} = \frac{\text{Green}}{\text{Green} + \text{Yellow}}$$


- TP: вероятность класса \geq порога и IoU \geq порога
- FP: вероятность класса $<$ порога или IoU $<$ порога



Кривая точности-полноты (precision-recall curve)

- Задаём порог на минимальное пересечение IoU_{min}
 - верная детекция - верный класс и $IoU \geq IoU_{min}$

Кривая точности-полноты (precision-recall curve)

- Задаём порог на минимальное пересечение IoU_{min}
 - верная детекция - верный класс и $IoU \geq IoU_{min}$
- Для порога уверенности $t = 1, \dots, 0$ (по \downarrow уникальных значений уверенности):
 - выделяем объекты
 - считаем $Pr(t), Rec(t)$; достраиваем $Prec(Recall)$
- Выход: $Prec(Recall)$ - зависимость точности от полноты

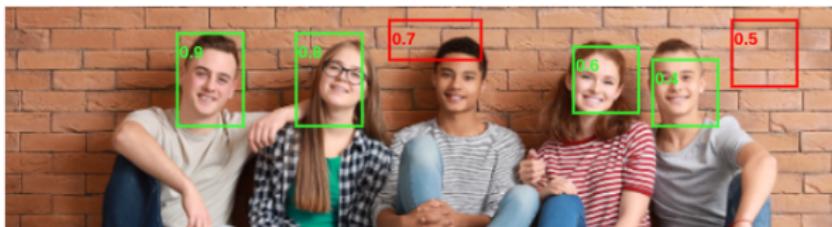
Кривая точности-полноты (precision-recall curve)

- Задаём порог на минимальное пересечение IoU_{min}
 - верная детекция - верный класс и $IoU \geq IoU_{min}$
- Для порога уверенности $t = 1, \dots, 0$ (по \downarrow уникальных значений уверенности):
 - выделяем объекты
 - считаем $Pr(t), Rec(t)$; достраиваем $Prec(Recall)$
- Выход: $Prec(Recall)$ - зависимость точности от полноты
- Пересчитываем интерпolatedированную точность (*interpolated precision*)

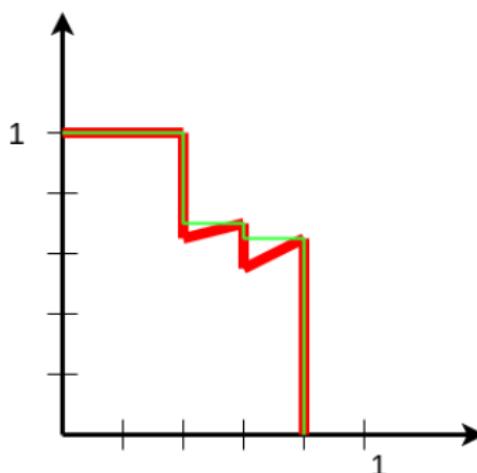
$$\overline{Pr}(i) = \max_{j:Rec(j) \geq Rec(i)} \{Pr(j)\}$$

- Получим сглаженный график $\overline{Prec}(Recall)$.

Кривая точности-полноты (precision-recall curve)



порог	precision	recall
0.9	1	1/5
0.8	1	2/5
0.7	2/3	2/5
0.6	3/4	3/5
0.5	3/5	3/5
0.4	4/6	4/5
0.3	4/6	4/5
...
0	4/6	4/5



Average precision, mean average precision

- Average precision (AP) -площадь под $\overline{Prec}(Recall)$:

$$AP = \int_0^1 \overline{Prec}(Recall) d(Recall)$$

- Mean average precision - макроусреднённая AP по классам:

$$mAP = \frac{1}{C} \sum_{c=1}^C AP(c)$$

Содержание

- 1 Меры качества
- 2 Двухстадийные детекторы

Детекция объектов: простейший подход



Простейший подход:

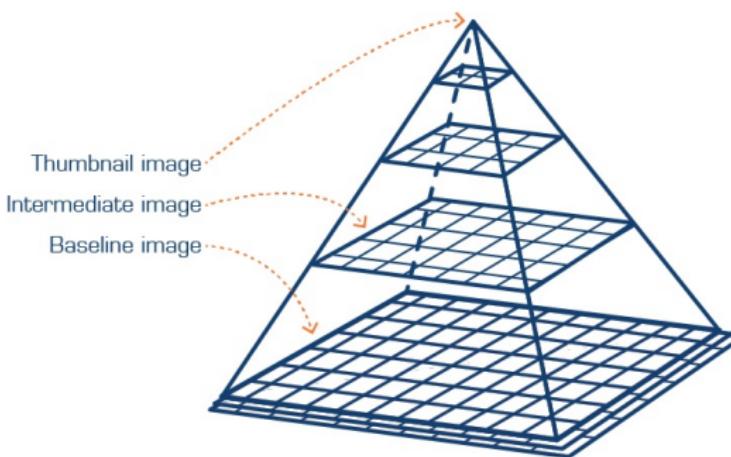
- ① извлечь участки на всех позициях и разных размеров
- ② применить классификатор к каждому участку.
- неэффективно, но отражает идеологию др. методов.

Извлечение участков разного размера

Проблема: детектор натренирован под фикс. разрешение.

Решение: извлекаем участки одного размера с разных масштабов исх. изображения.

Гауссова пирамида



Также нужно извлекать участки разной формы.

Генерация участков-кандидатов

- Алгоритм selective search¹ генерирует участки-кандидаты.



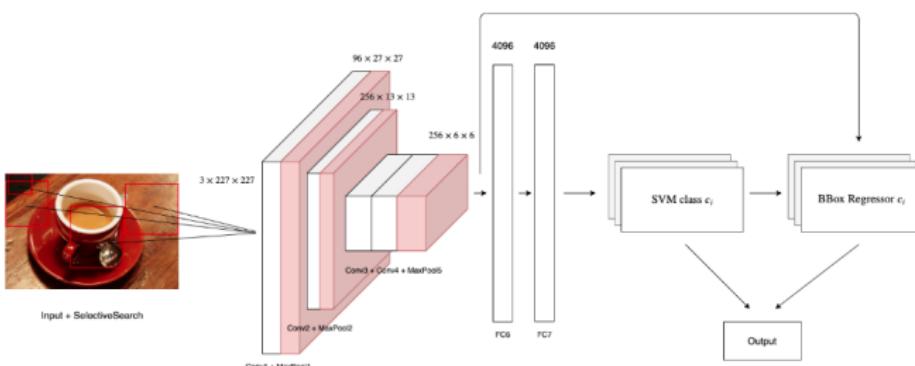
Алгоритм:

- объединяем RGB+XY пиксели в суперпиксели (например, используя SLIC).
- пока $\#\text{суперпикселей} > 1$:
 - обводим каждый суперпиксель рамкой, запоминаем.
 - объединяем суперпиксели
 - малые по размеру, похожие по цвету, текстуре (HoG), рядом друг с другом
- Выход алгоритма: множество рамок, в которых потенциально м. быть объект.

¹<http://www.huppenen.nl/publications/selectiveSearchDraft.pdf>

R-CNN²

R-CNN scheme:



²<https://arxiv.org/pdf/1311.2524.pdf>

Алгоритм R-CNN

- ❶ SelectiveSearch генерирует ~ 2000 регионов-кандидатов
- ❷ Регионы-кандидаты масштабируются к 224x224 (AlexNet)
- ❸ Кодировщик (AlexNet) сверточные слои (но не полно связные) извлекают 4096 признаков для каждого региона.
- ❹ SVM классификатор обучается на $C + 1$ класс (+1 для фона)
- ❺ Регрессия обучается уточнять координаты регионов, предложенных SelectiveSearch:

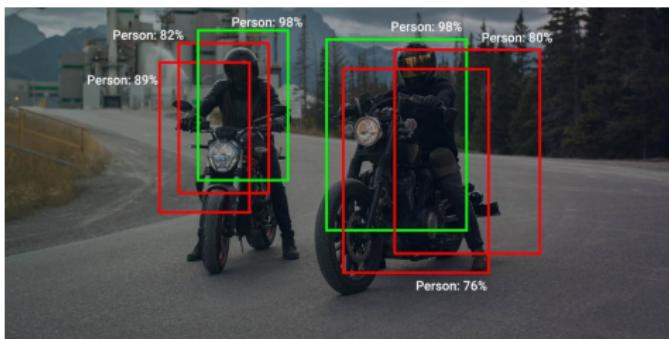
$(\hat{x}, \hat{y}, \hat{h}, \hat{w})$ -предсказанный регион, (x, y, h, w) -реальный регион

регрессия предсказывает: $\left(\frac{x - \hat{x}}{w}, \frac{y - \hat{y}}{h}, \ln \left(\frac{\hat{w}}{w} \right), \ln \left(\frac{\hat{h}}{h} \right) \right)$

регрессия обучается на регионах с $\text{IoU} > 0.3$ с истинным регионом

- ❻ Убираем лишние выделения (non-maximum suppression)

Подавление не-максимумов (non-maximum suppression)



Алгоритм подавления не-максимумов:

- отбрасываем регионы с низкой уверенностью
- упорядочиваем регионы по убыванию уверенности
- последовательно, начиная с региона максимальной уверенности, к менее уверенным:
 - если регион имеет высокий IoU с др. регионом более низкой уверенности, отбрасываем 2й регион.

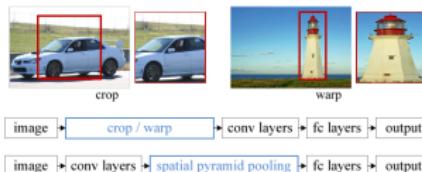
Недостатки R-CNN

Недостатки R-CNN:

- требуется отдельная процедура обучения для
 - кодировщика (донастройка по log-loss)
 - классификатора (SVM)
 - регрессии
- для каждого региона кандидата нужно применять полносверточный энкодер
 - регионов-кандидатов много и они сильно пересекаются

SPP-net³

- Проблемы R-CNN:
 - CNN переприменяется к каждому региону
 - необходимо приводить разрешение к 224x224 (деформация либо обрезка краёв с потерей информации)

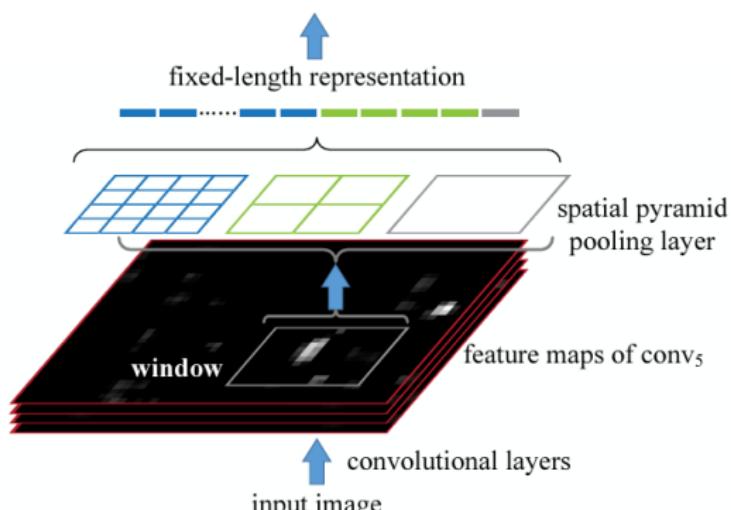


Идеи SPP-net (spatial pyramid pooling net):

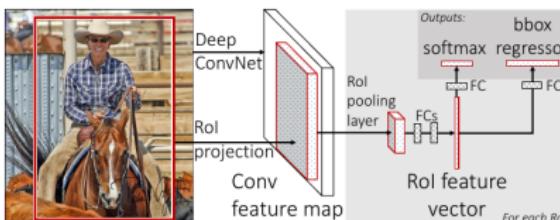
- применить CNN ко всему изображению 1 раз
 - ускорение 10x - 100x
- spatial pyramid pooling приводит к фикс. размеру изображение произвольного размера

³<https://arxiv.org/pdf/1406.4729v4.pdf>

Spatial pyramid pooling



Fast R-CNN⁴

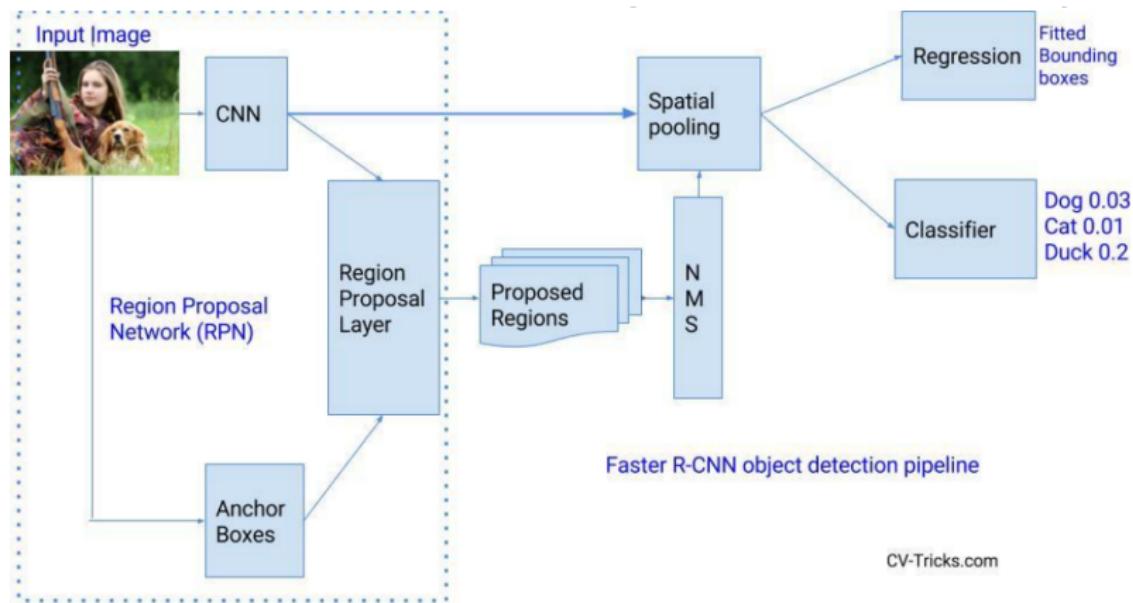


Fast R-CNN: SPP-net со следующими отличиями:

- Однослойный spatial pyramid max pooling на фикс. сетке 7x7 (названный ROI pooling).
 - произвольный тензор (выделяющий область) -> 49C признаков
- Классификатор и bbox-регрессия реализованы доп. слоями сети.
 - потери=классификационные+регрессионные (от локализации)

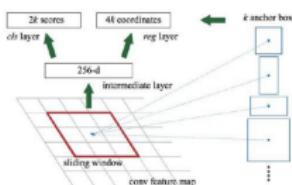
⁴<https://arxiv.org/pdf/1504.08083.pdf>

Faster R-CNN



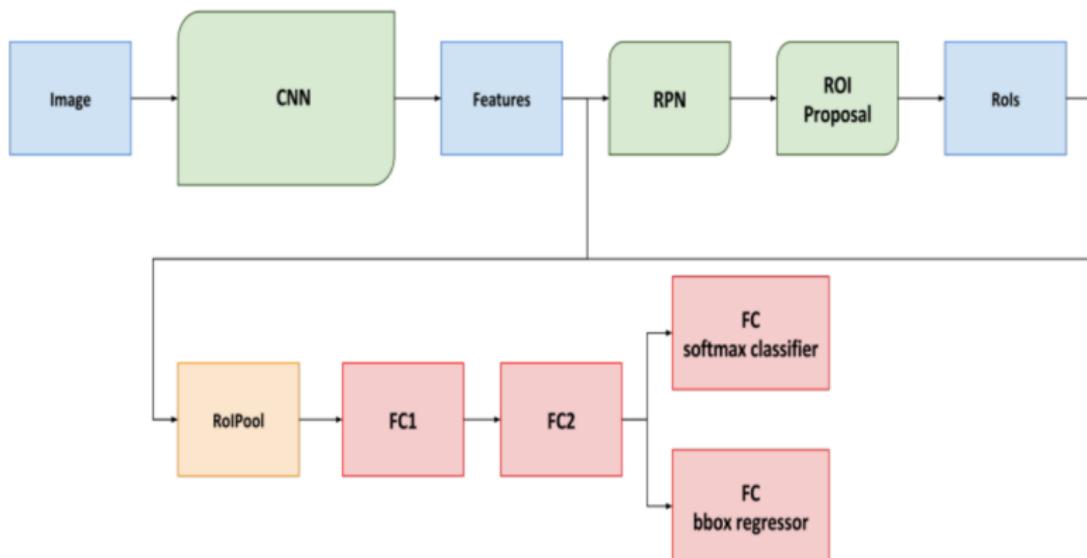
- Полностью нейросетевое решение:
 - SelectiveSearch->генерация регионов-кандидатов через Region Proposal Network (RPN).

Faster R-CNN



- Region proposal network: 1 этап
- Проходим скользящим окном 3×3 по карте признаков, генерируем $k=9$ регионов-кандидатов.
 - регионы-кандидаты - шаблоны размера 128, 256, 512 и соотношением сторон 1:1, 1:2, 2:1
- Выход RPN:
 - вероятность присутствия/отсутствия объекта: $2k$
 - положение объекта: $4k$ координаты (уточненных регрессией)
- Постпроцессинг регионов-кандидатов: удаляем слишком узкие или выходящие за пределы изображения регионы, NMS с $\text{IoU}=0.9$

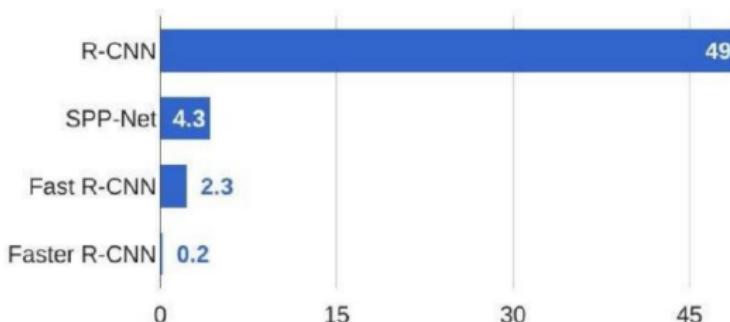
Faster R-CNN



Faster R-CNN

- Обучение: учим поочередно итеративно RPN и оставшуюся часть сети.
- Применение:
 - изображение->CNN+RPN
 - оставляем 6000 перспективных регионов
 - удаляем слишком тонкие регионы или попавшие за пределы изображения
 - подавление не-максимумов с высоким $\text{IoU}=0.9$
 - оставшиеся регионы->RoI
 pooling ->классификация+регрессия
 - итоговое подавление не-максимумов

Сравнение скорости работы



- По скорости работы Faster-RCNN уступает одностадийным детекторам, но превосходит их по точности.

Заключение

- Детекция: выделение объектов рамками+классификация.
- Баланс 2-х потерь:
 - потери классификации (что изображено)
 - потери локализации (координаты рамки)
- Каждый объект идентифицируется многими рамками.
 - используем подавление не-максимумов.
- Двухстадийные методы:
 - 1 извлекают регионы-кандидаты
 - 2 классифицируют и уточняют расположение каждого региона
- Методы:
 - R-CNN -> SPP-net -> Fast R-CNN -> Faster-RCNN.