

Распознавание объектов

Сергей Николенко

НИУ ВШЭ – Санкт-Петербург

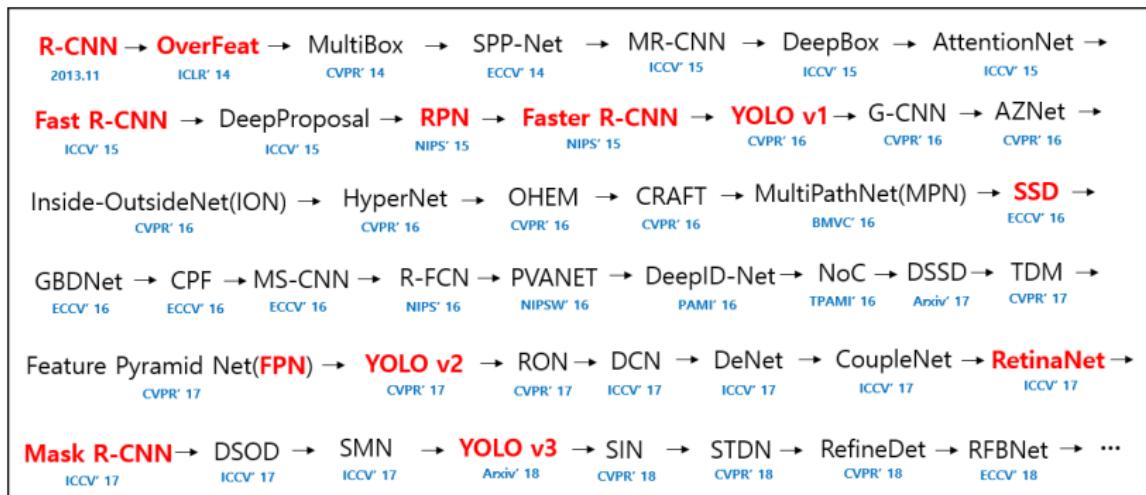
17 октября 2020 г.

Random facts:

- 17 октября в ООН – Международный день борьбы за ликвидацию нищеты; 17 октября 1987 года около 100 тысяч человек собрались в Париже, на площади Прав человека (Трокадеро), и поддержали лозунг отца Жозефа Вресински: «Там, где человек вынужден жить в нищете, нарушаются права человека. Объединиться в борьбе за уважение этих прав есть священный долг каждого»
- 17 октября 1825 г. в Париже прошла премьера оперы Ференца Листа «Дон Санчо, или Замок любви»; автору тогда ещё не было и четырнадцати
- 17 октября 1831 г. Майкл Фарадей открыл электромагнитную индукцию, 17 октября 1855 г. Генри Бессемер запатентовал свой процесс приготовления стали, 17 октября 1897 г. Константин Циолковский сообщил о постройке аэродинамической трубы, а 17 октября 1985 г. был выпущен Intel 80386, первый 32-разрядный процессор для IBM PC
- 17 октября 1869 г. по заданию газеты «New York Herald» Генри Стэнли отправился на поиски пропавшей в Африке экспедиции Дэвида Ливингстона
- 17 октября 1961 г. в нью-йоркском МОМА была выставлена картина Анри Матисса «Лодка» (Le Bateau); только 3 декабря кто-то заметил, что картина висит вверх ногами

Распознавание объектов

Plan



Постановка задачи

- Нужно выделить и распознать объекты на фотографиях.
- Мы уже обсудили базовые архитектуры, которые будут работать с самим распознаванием.
- Но как их собственно использовать?
- Надо же ещё и понять, *где* эти объекты.

Pascal VOC

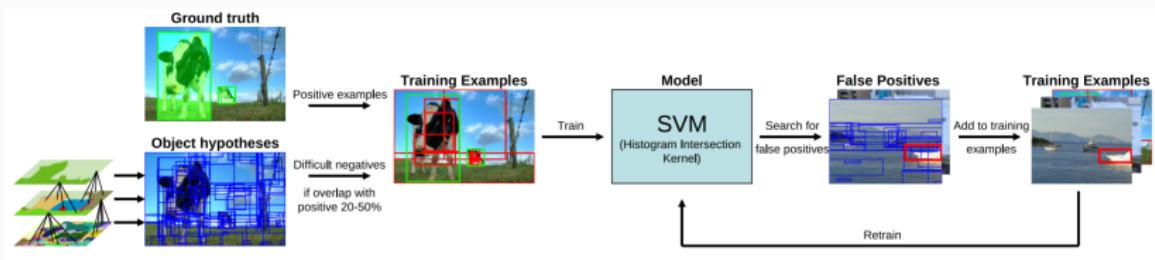
- Датасет PASCAL VOC для сегментации:



- Относительно маленький, надо сначала на ImageNet обучать.

Classical computer vision

- Object detection was done without neural networks.
- (Uijlings et al., 2012): Selective Search for Object Recognition



- I.e., we find candidates and then classify correct candidates from “hard incorrect” examples

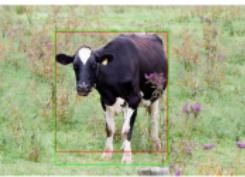
Classical computer vision

- The interesting part for us is selective search itself
- (Felzenszwalb, Huttenlocher, 2004): Efficient Graph-Based Image Segmentation
 - represent the image as a graph where edge weights show similarities between patches;
 - think of a predicate to compare different components of the graph;
 - get the basic small pieces (superpixels, in a way).



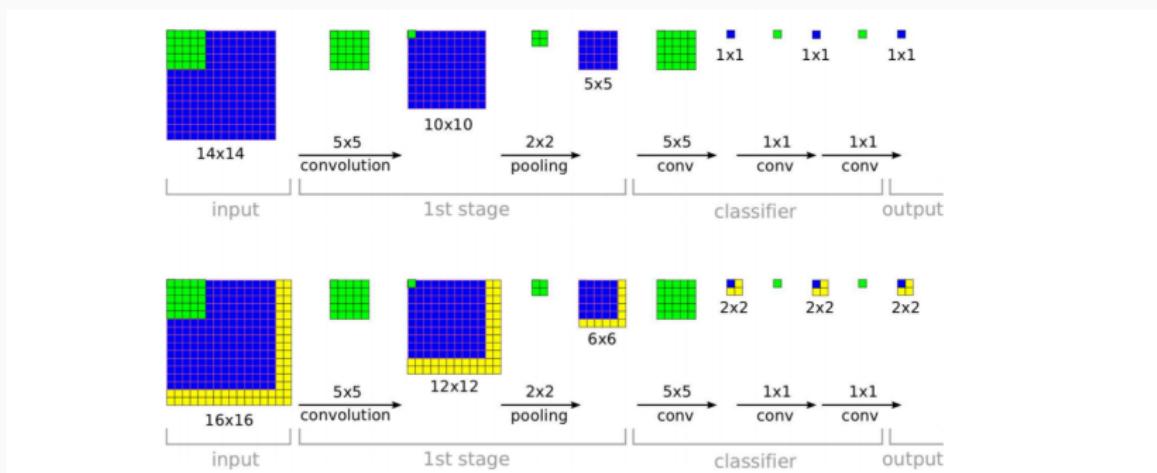
Classical computer vision

- And then Uijlings et al. (2012) begin to unite these pieces:
 - iteratively unite nearest neighbors;
 - lots of different proximity measures: color, texture, fill, size...



OverFeat

- OverFeat (Sermanet et al., 2014):
 - use AlexNet as the base classifier;
 - run it for all possible patches in the picture; not so hard for a convolutional network:

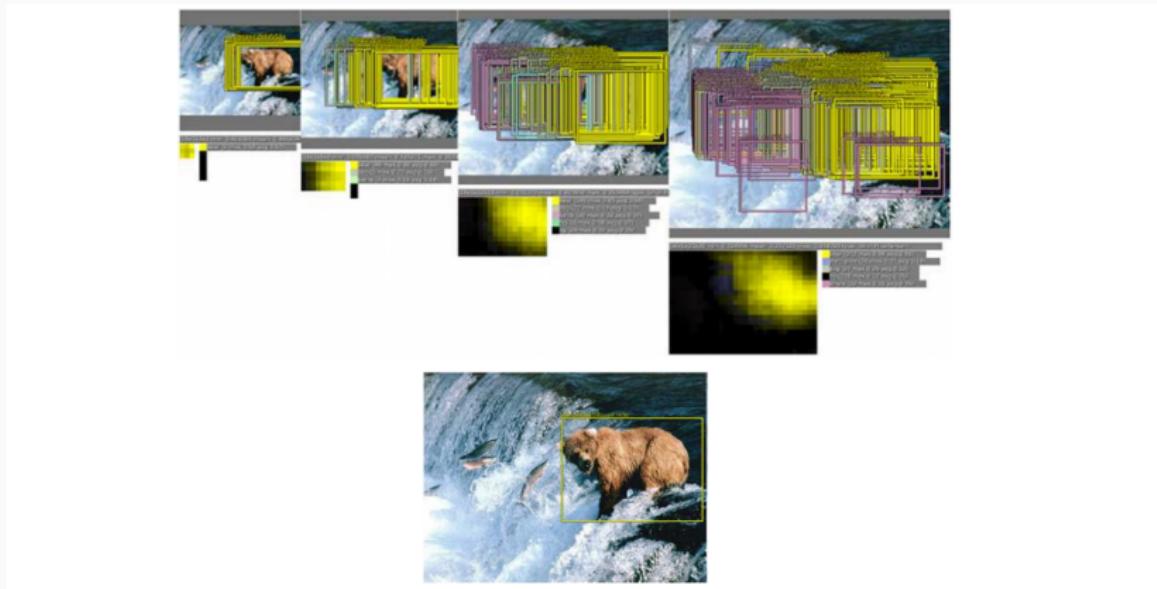


OverFeat

- Then replace the classifier with a regression that predicts bounding boxes.
 - We get a lot of intersecting bboxes:

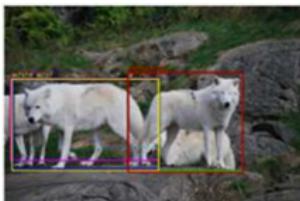


- Than can be then united with a greedy algorithm:

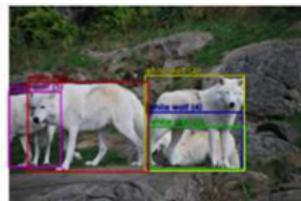


OverFeat

- Sometimes works very well, sometimes not so well, especially in difficult conditions:



Top 5:
white wolf
white wolf
timber wolf
timber wolf
Arctic fox



Groundtruth:
white wolf
white wolf (2)
white wolf (3)
white wolf (4)
white wolf (5)



Top predictions:
person (confidence 6.0)

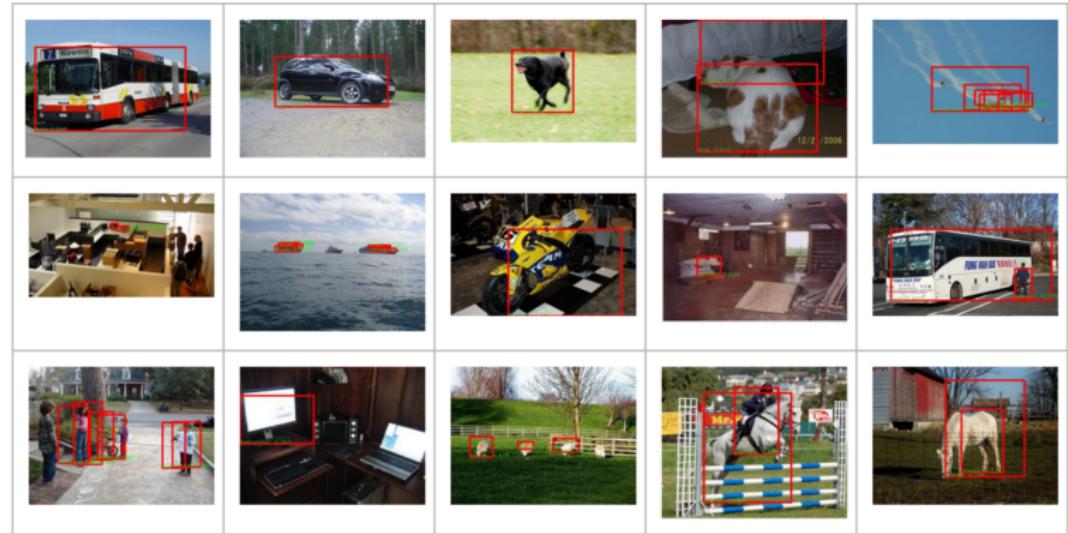


Groundtruth:
drum
lamp
lamp (2)
guitar
person
person (2)
person (3)
microphone
microphone (2)

- (Erhan et al., 2014), a work from Google.
- The idea is to use one network to find potential bboxes regardless of the classes.
- Train a network that outputs, given an image:
 - some predefined number of bboxes;
 - for each bbox, output the coordinates (two corners, four numbers) and certainty that there is an object (one number);
 - during training, assign network outputs to the correct answers with a bipartite matching algorithm (there are few objects so it's fast enough).

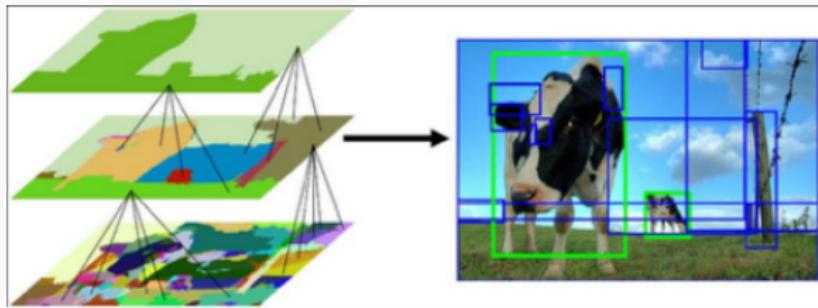
MultiBox

- Results:

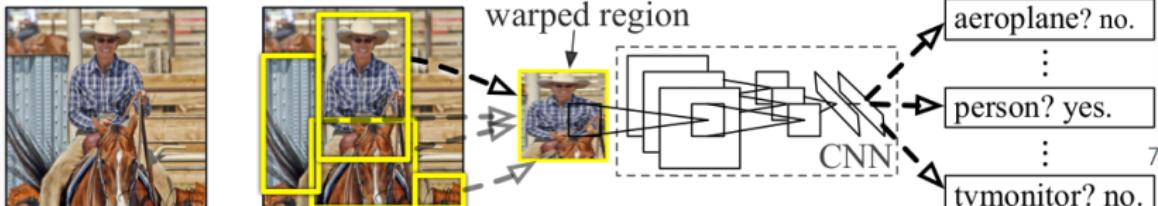


R-CNN

- R-CNN: Region-based Convolutional Network method (Girshick et al., 2014).
- Идея: будем выделять участки каким-то внешним алгоритмом (например, selective search).



- Затем на них выделять признаки CNN (предобученной на ImageNet и fine-tuned на нужном) и классифицировать.



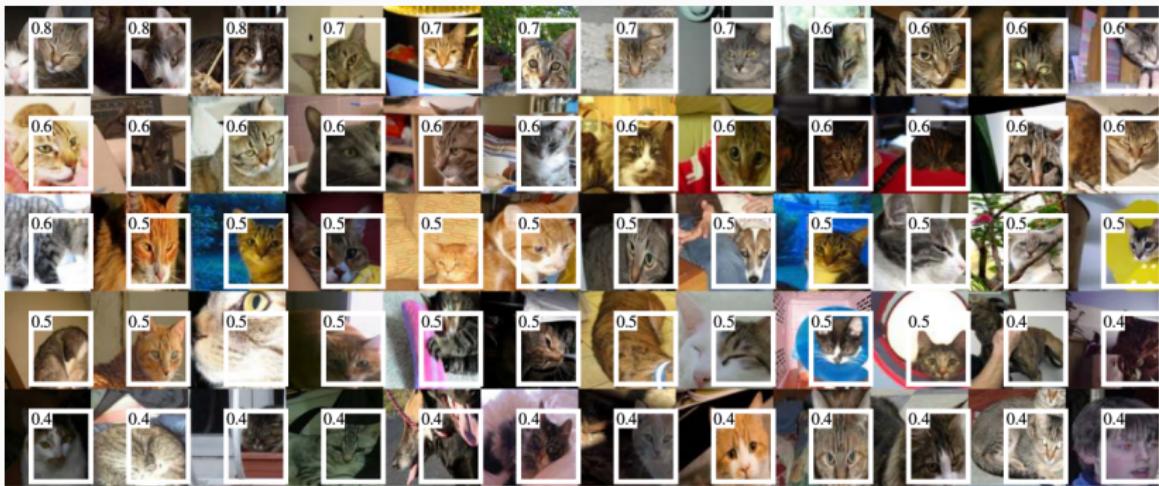
R-CNN

- Можно визуализировать: берём нейрон высокого уровня, показываем, на каких участках он активируется сильнее.



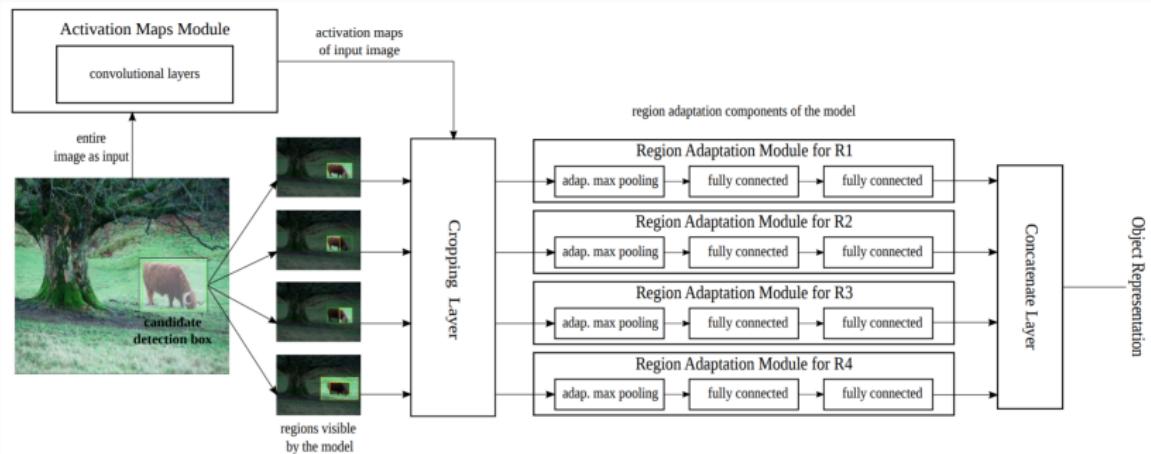
R-CNN

- Можно визуализировать: берём нейрон высокого уровня, показываем, на каких участках он активируется сильнее.

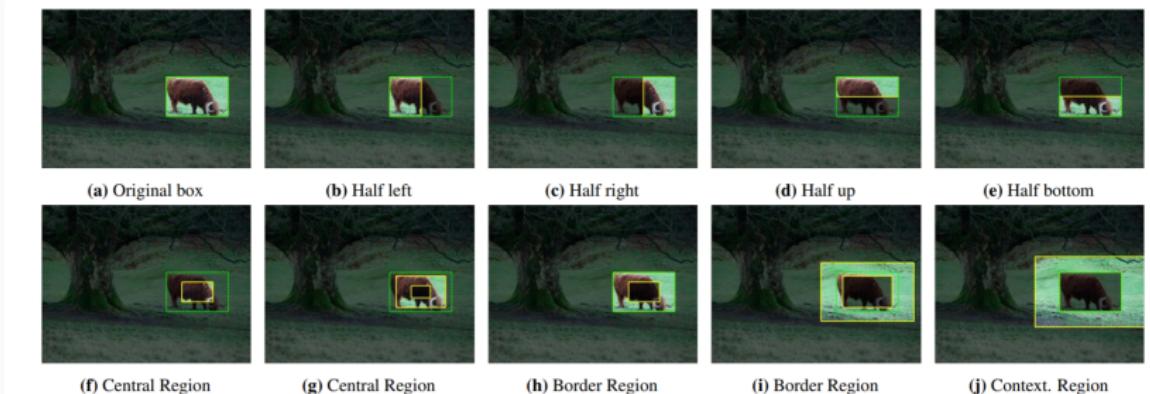


MR-CNN

- (Gidaris, Komodakis, 2015): one part of the network chooses where to focus, the other parts do the processing:

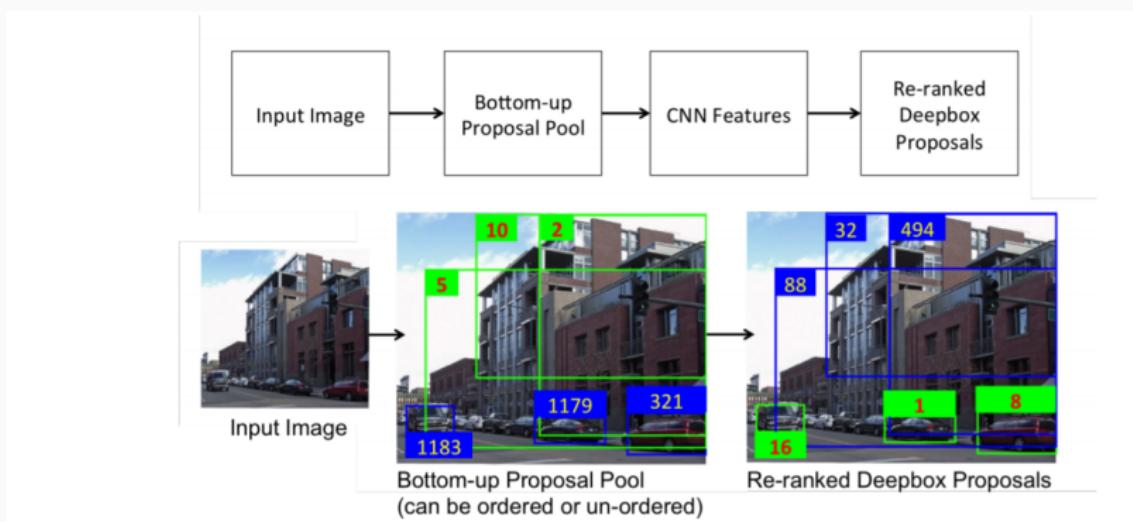


- Adaptive max pooling – kind of like a pyramid with one layer.
- It can train different types of regions:



DeepBox

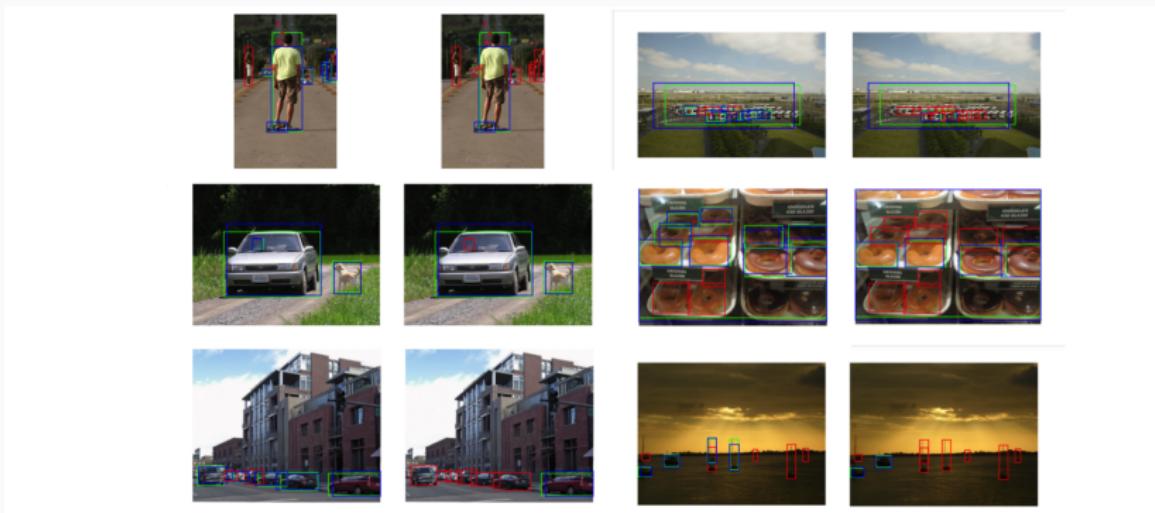
- (Kuo et al., 2015), DeepBox: first generate proposals, then rerank them.
- No classes, just looking for objects:



- DeepBox reranks by the data from CNNs, similar to R-CNN:
 - R-CNN is very slow;
 - would be faster to compute features only once, but then the scale might be wrong;
 - DeepBox runs several times at different scales and chooses the scale suitable for a given proposal.

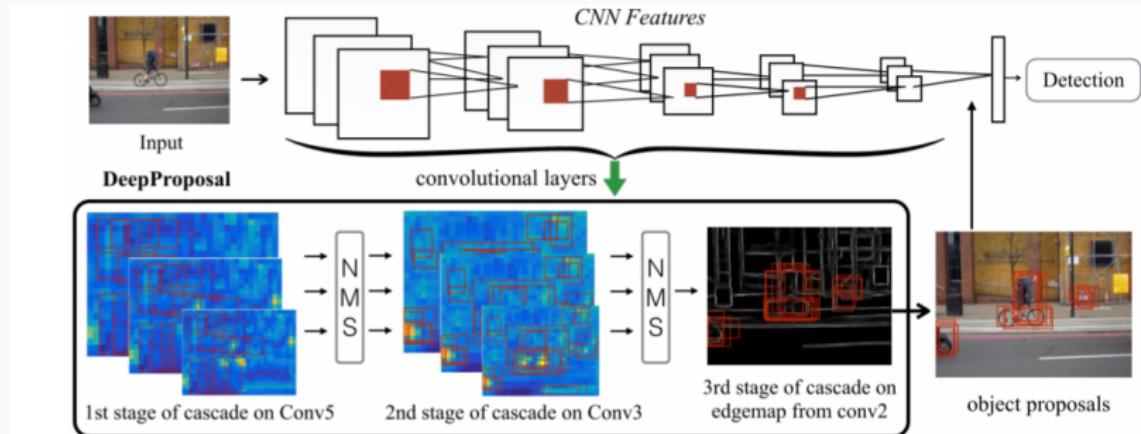
DeepBox

- It generalizes well even to previously unseen objects:



DeepProposal

- (Ghodrati et al., 2015), DeepProposal: how to produce proposals from CNN features.
- Start looking at the last layer and filter as we move back:



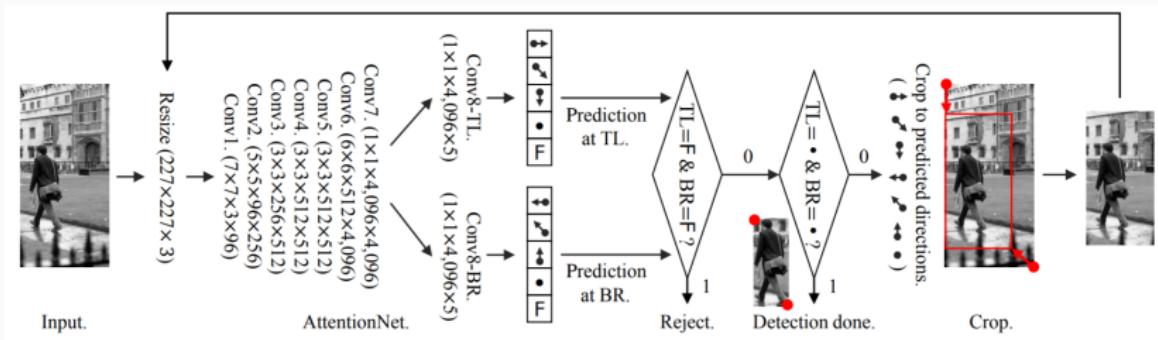
AttentionNet

- (Yoo et al., 2015), AttentionNet: iteratively cut the bbox until we get to the final result.



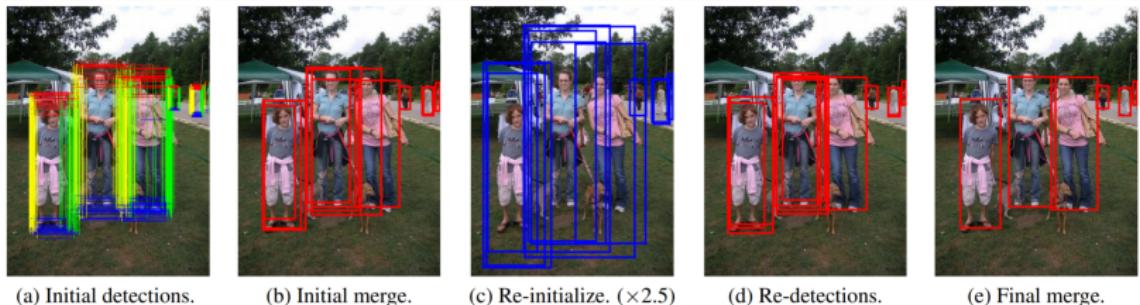
AttentionNet

- Formally it's just classification with respect to what we need to do with bbox angles:



AttentionNet

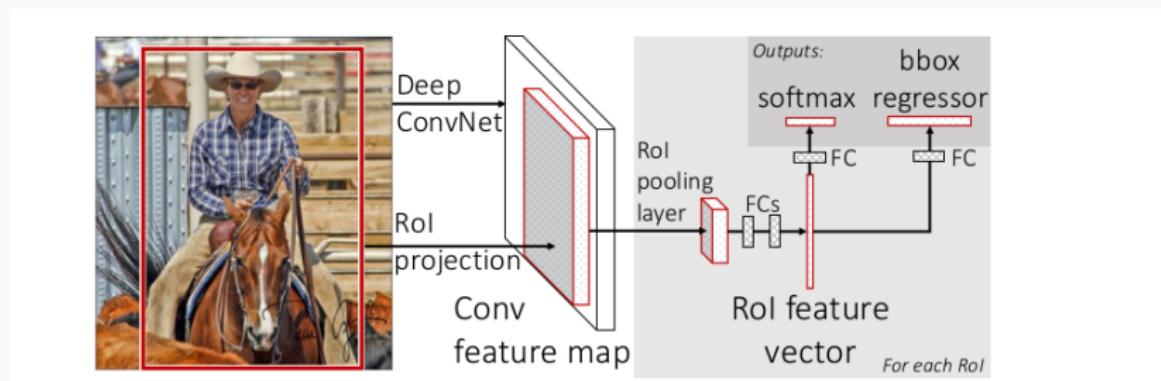
- It can even be generalized to several objects with selective search:



- Но у R-CNN есть проблемы:
 - надо обучать в несколько приёмов (сначала CNN, потом SVM на её признаках, потом bounding box регрессоры)
 - обучать долго (надо выделить признаки из каждого участка в тренировочном датасете и записать на диск для обучения SVM и регрессоров);
 - распознавать очень медленно (47с на картинку на GPU).
- Причина в том, что для каждого участка надо делать проход по CNN.

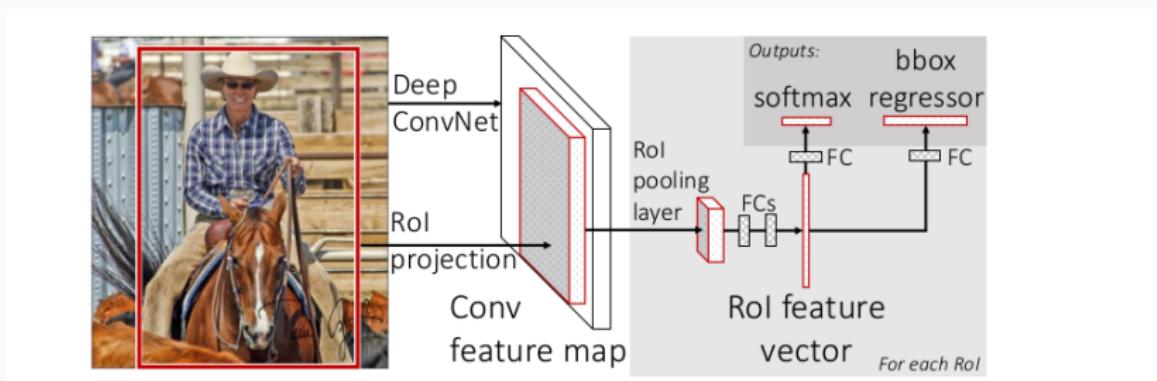
Fast R-CNN

- Поэтому – *Fast R-CNN*.
- ROI (region of interest) projection собирает признаки из заданного участка.
- Теперь нужно считать признаки для всей картинки только один раз.



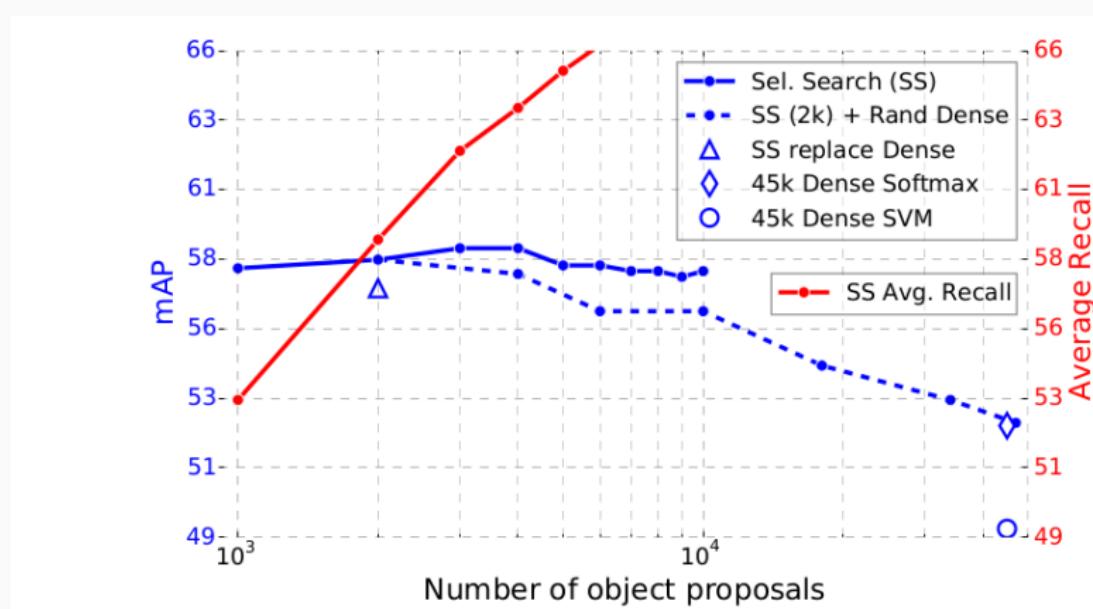
Fast R-CNN

- Функция потери складывается из двух: ошибка классификации и ошибка bounding box регрессии.
- Получается на два порядка быстрее и ничем не хуже (иногда лучше) по качеству.



Fast R-CNN

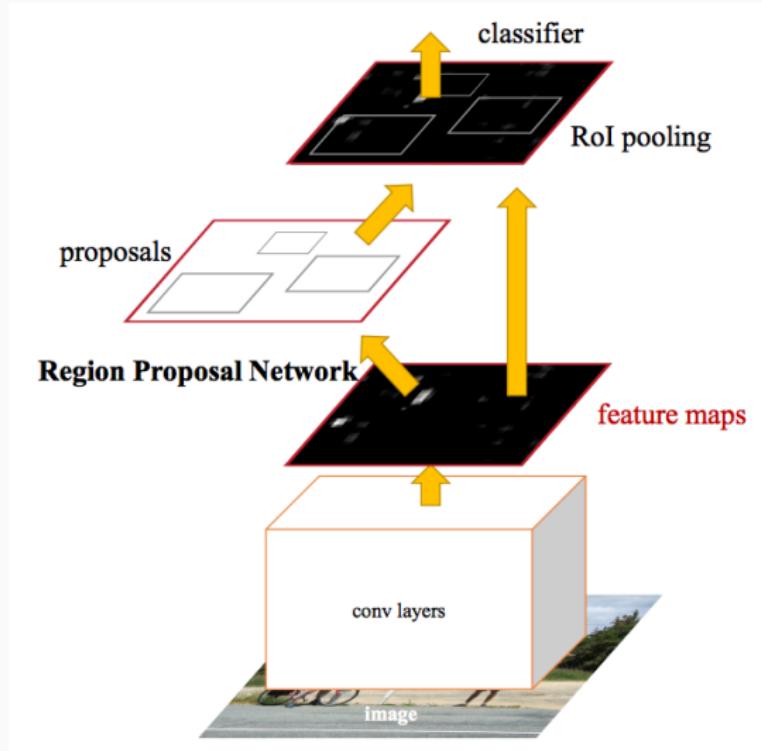
- Сами кандидаты в участки по-прежнему из selective search.
- Выяснилось, что добавлять больше кандидатов не очень помогает.



- Осталось одно узкое место: selective search для выбора участков.
- Оказывается, это тоже можно вставить в сеть!
- Получается *Faster R-CNN* (Ren et al., 2015).

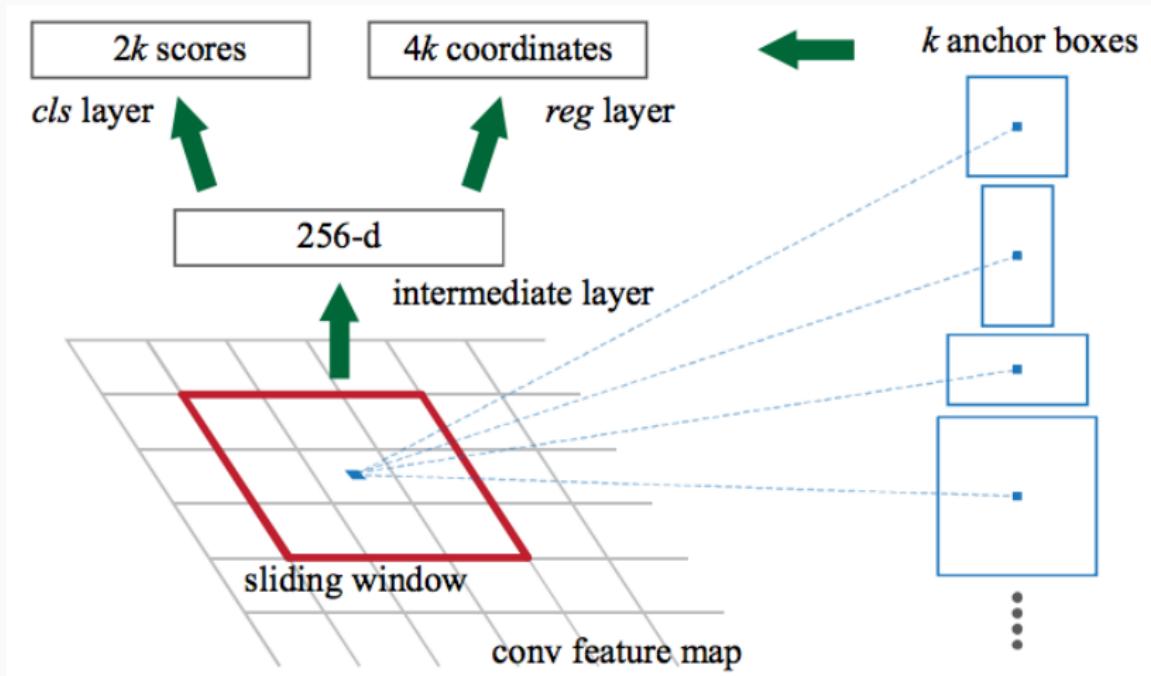
Faster R-CNN

- Отдельная Region Proposal Network:



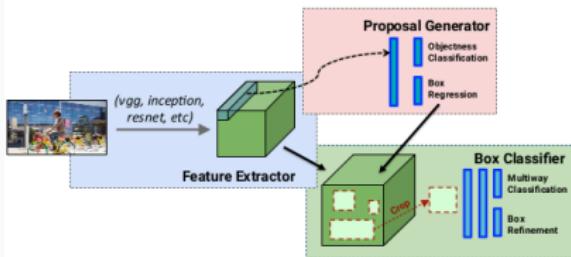
Faster R-CNN

- Оценивает каждое отдельное окно из существующих anchor boxes:

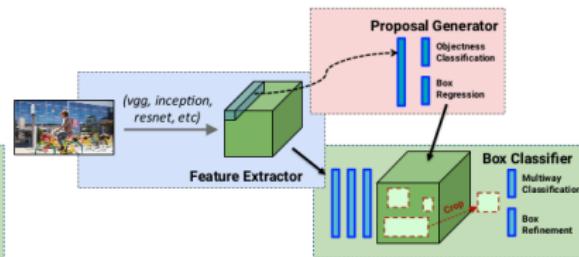


Faster R-CNN

- Можно и ещё больше сэкономить, не вычисляя на каждом участке вообще никаких сложных слоёв.
- R-FCN (Region-based Fully Convolutional Network) вырезает признаки из самого последнего перед классификацией слоя (Dai et al., 2016).
- Вот в чём разница:



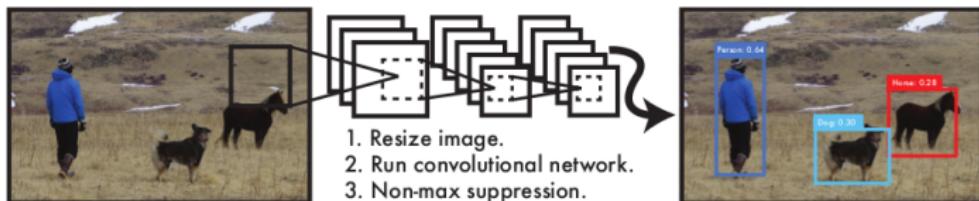
(b) Faster RCNN.



(c) R-FCN.

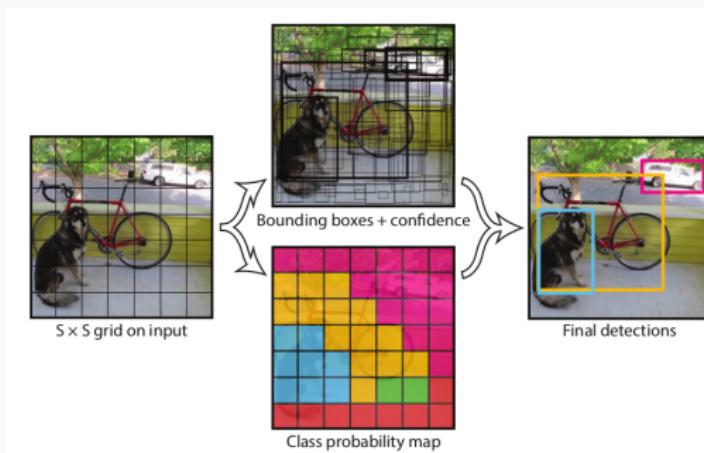
YoLo

- You only look once (YOLO; Redmon et al., 2016).
- За один проход ищет и прямоугольники с объектами (bounding boxes), и сами объекты.

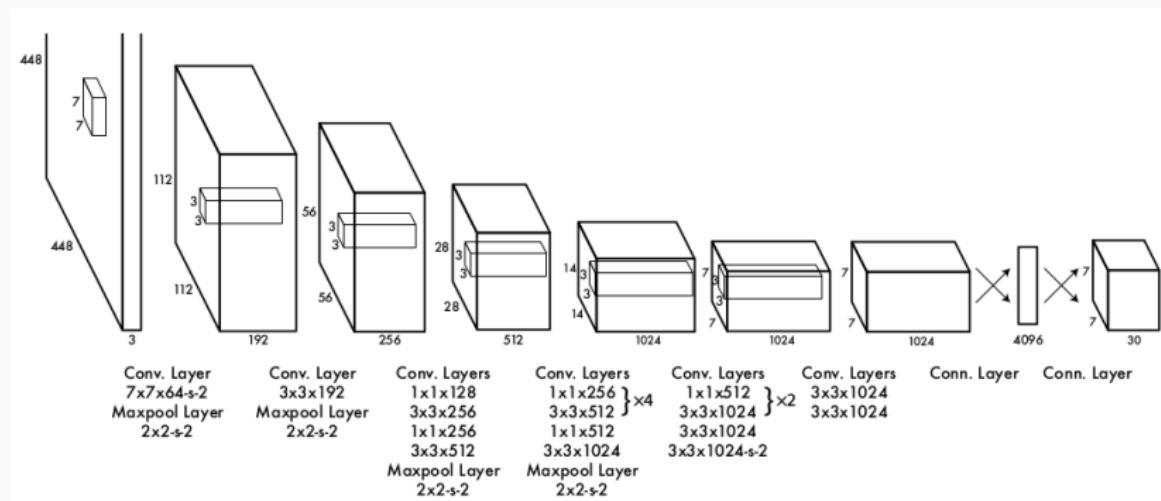


- Разбивает картинку на решётку $S \times S$.
- В каждой ячейке предсказывает и прямоугольники, и вероятности; и потом просто

$$p(\text{class}_i \mid \text{obj})p(\text{obj})p(\text{bbox}).$$

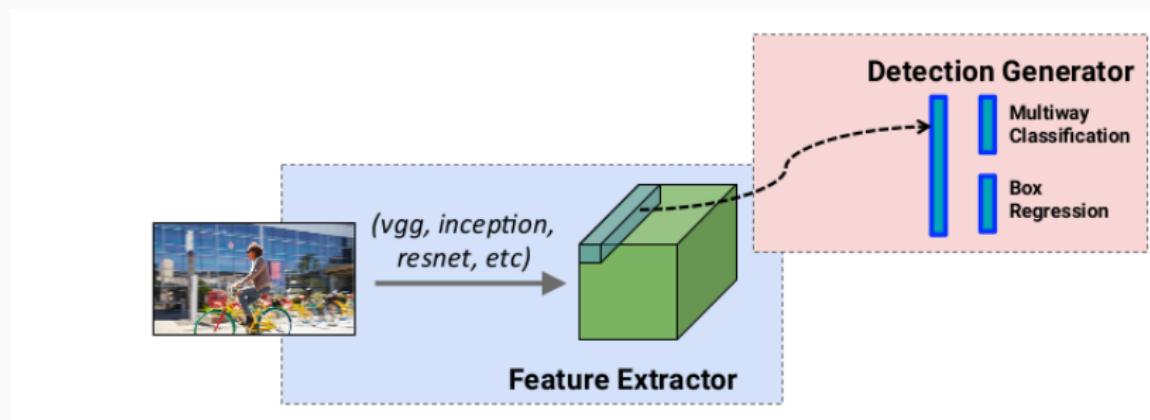


- Архитектура YOLO.

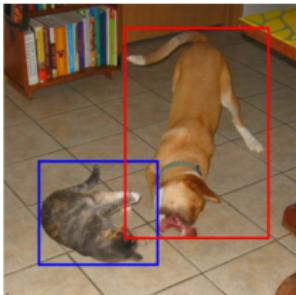


- Первый YOLO уступал по точности, но работал быстро. Сейчас появились YOLOv2 и YOLOv3, которые и по точности не хуже.

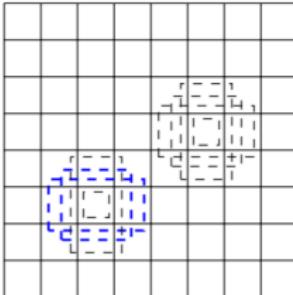
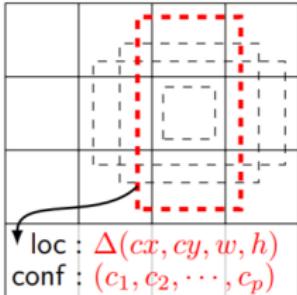
- Further development of the idea to predict everything at once – *single-shot detectors* (SSD; Liu et al., 2016).
- A single network that predicts:
 - several class labels;
 - several corresponding positions for anchor boxes (bounding boxes of several predefined sizes).



- To get correct answers, we compare standard anchor boxes with real ones; this yields the correct values of offsets and everything else:

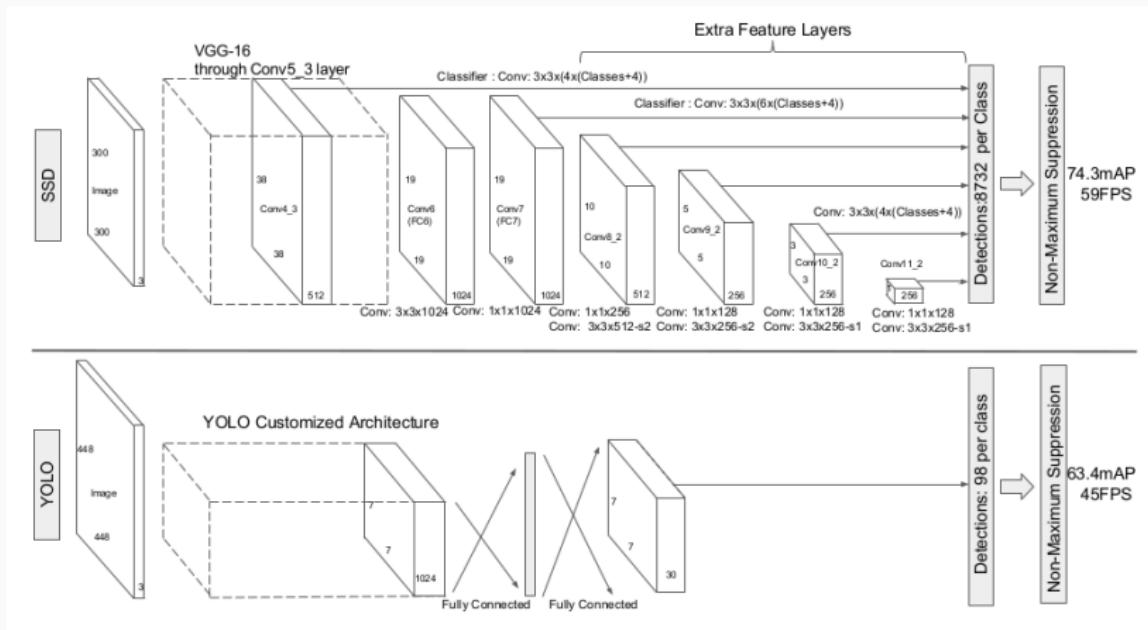


(a) Image with GT boxes

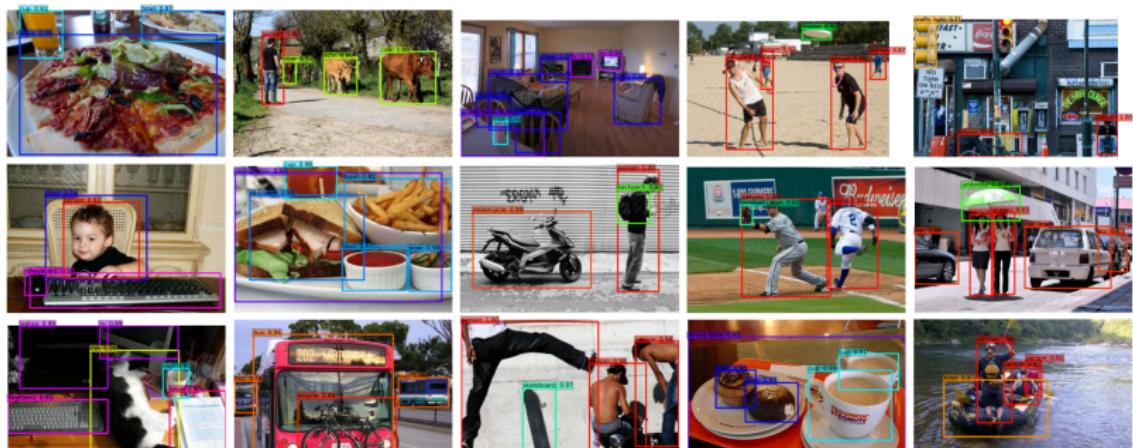
(b) 8×8 feature map(c) 4×4 feature map

- The objective function is similar to MultiBox but with classes.

- The architecture is more complex than YoLo, a *lot* of outputs:



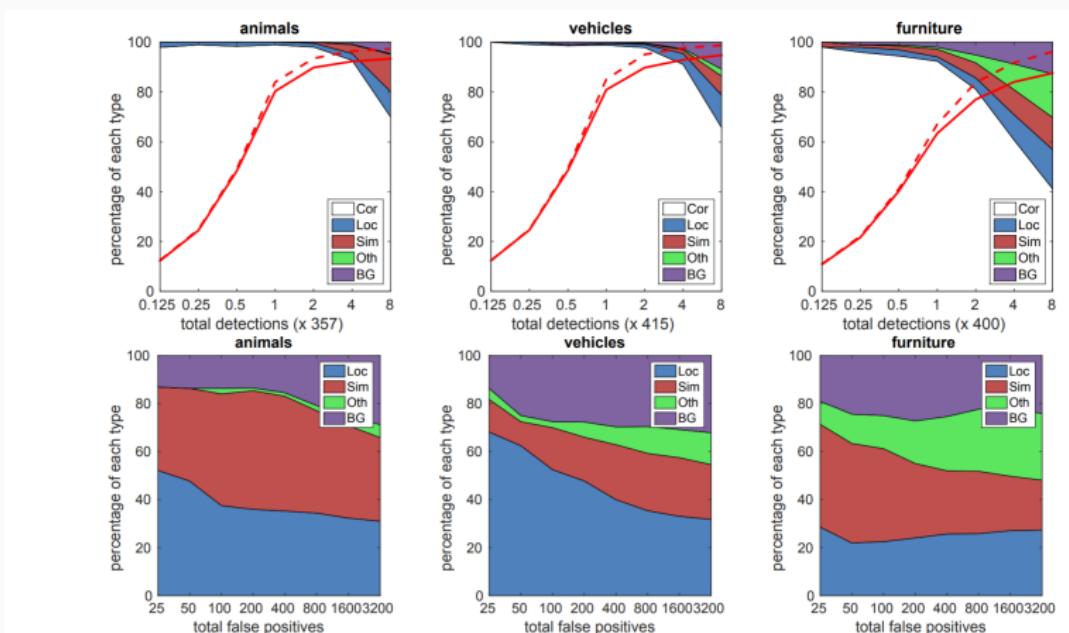
- Again, works pretty well (everything will work pretty well from now on):



- At the moment, YoLo was the only real-time detector that yielded > 70% mAP on Pascal VOC2007:

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	~ 1000×600
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

- We can analyze where the errors arise:



Спасибо!

Спасибо за внимание!

