

Object Detection

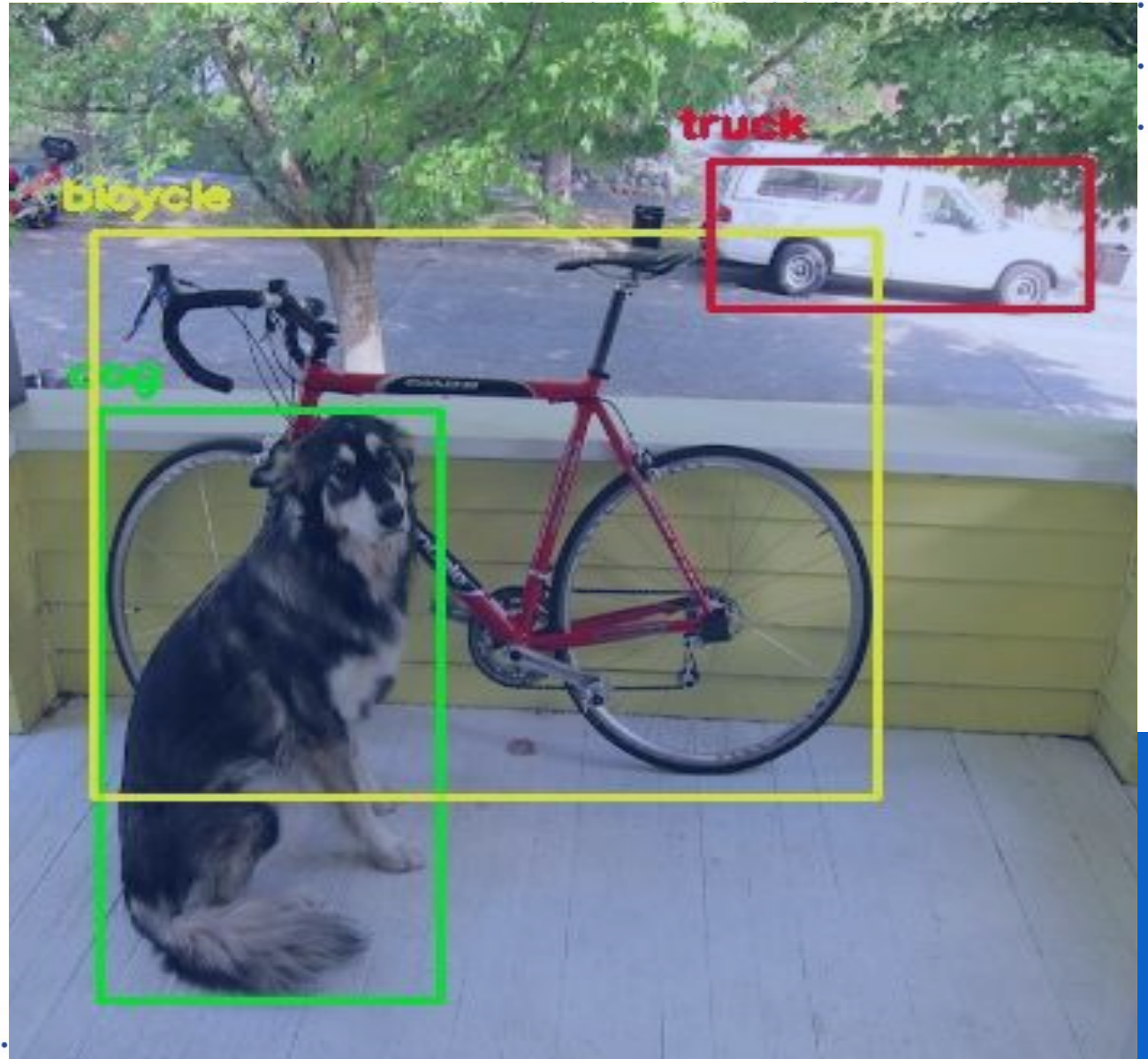
ПЛАН

- Введение в проблему Object Detection
- Первые алгоритмы (Selective Search)
- Старые Алгоритмы
- Fast-RCNN
- Faster-RCNN
- YOLOv3/v4
- RetinaNET

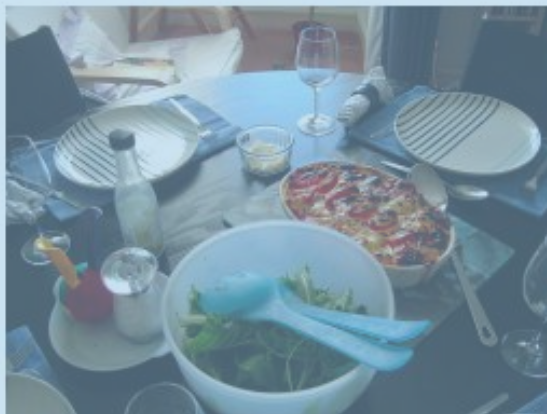


Введение в проблему Object Detection

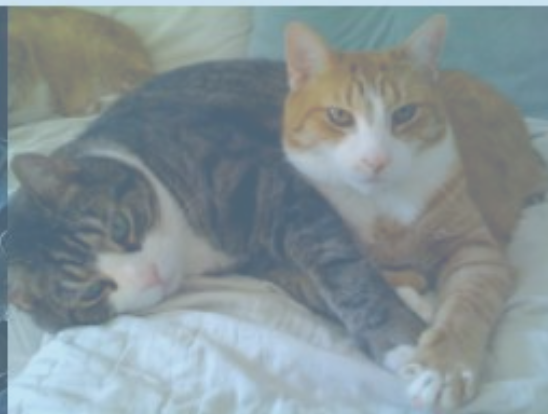
- Классификация объектов
- Нахождение границы объекта на изображении.
- Нахождение всех копий объекта.



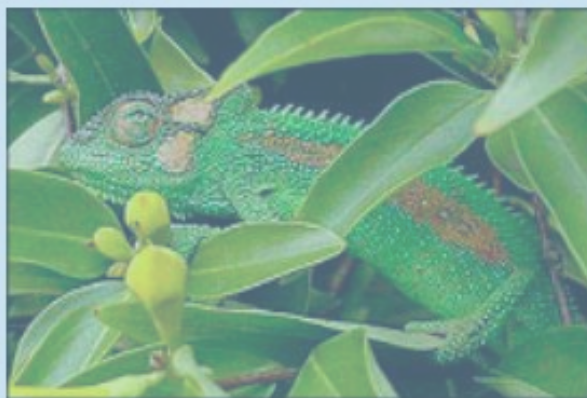
ВВЕДЕНИЕ В ПРОБЛЕМУ **OBJECT DETECTION**



(a)



(b)



(c)



(d)

Как Выделить объекты??

- a) Нет единого критерия.
- b) Можно отличить по цвету (но не текстуре).
- c) Можно отличить по текстуре (но не цвету).
- d) Колеса рядом с корпусом (не по цвету/ не по текстуре).

Первые алгоритмы Selective Search

Selective Search by
Hierarchical Grouping

Селективный поиск
иерархической
группировкой

Поиск исходных регионов



ПЕРВЫЕ АЛГОРИТМЫ

SELECTIVE SEARCH

ОБЪЕДИНЯЕМ РЕГИОНЫ НА ОСНОВЕ 3-Х СТРАТЕГИЙ

- 1) Схожесть цветов** – насколько цветовая гистограмма одной области похожа на цветовую гистограмму другой области
*учитывается размер областей
- 2) Схожесть текстур** – насколько текстуры одной группы пикселей похожи на текстуры другой группы пикселей
*не учитывается размер изображения
*метод извлечения признаков SIFT (в OpenCV)
- 3) Схожесть размеров + IntersectionOverUnion** (отношение площади пересечения к площади объединения) – насколько один объект перекрывает другой

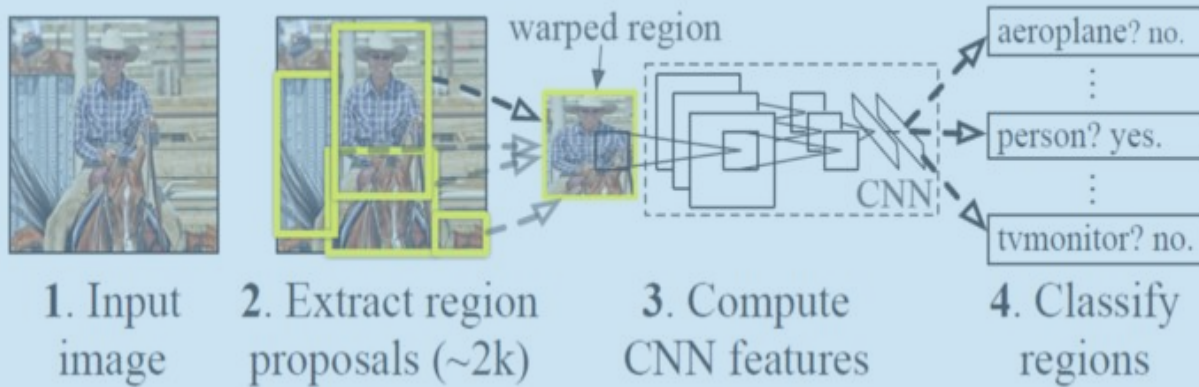
Итоговая формула:

$$s(r_i, r_j) = a_1 s_{\text{colour}}(r_i, r_j) + a_2 s_{\text{texture}}(r_i, r_j) + a_3 s_{\text{size}}(r_i, r_j) + a_4 s_{\text{fill}}(r_i, r_j),$$

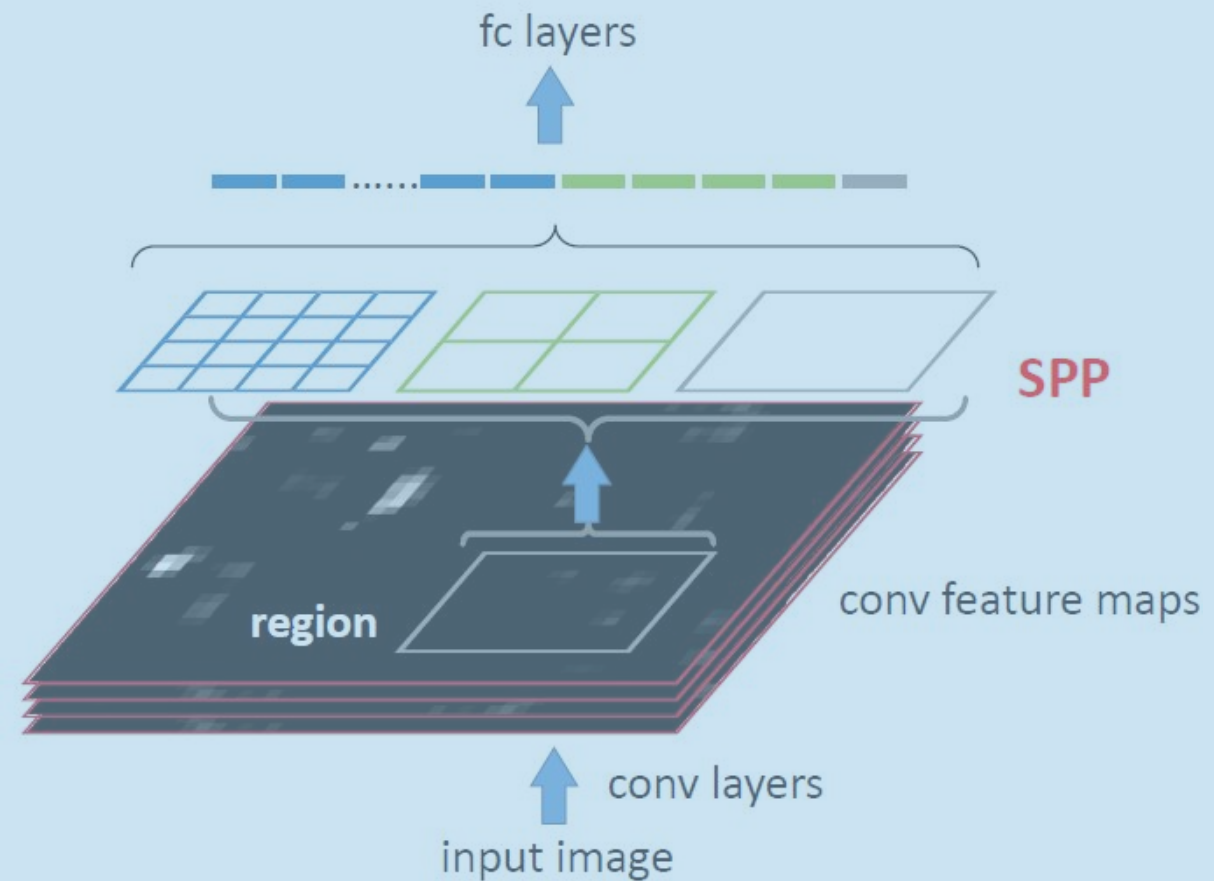
СТАРЫЕ АЛГОРИТМЫ 1

RCNN (Region Based Proposal)

R-CNN: *Regions with CNN features*

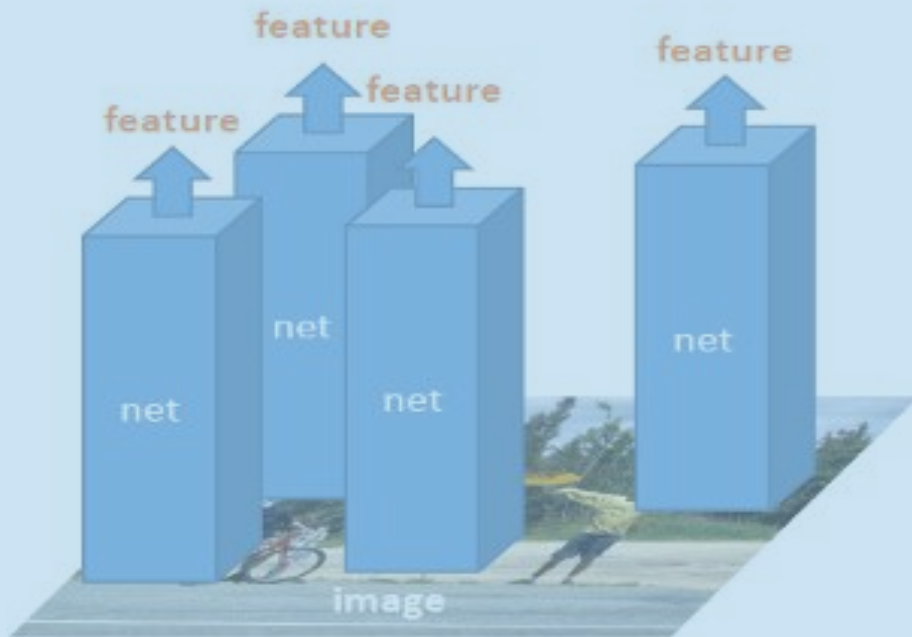


SPP (Spatial Pyramid Pooling)



СТАРЫЕ АЛГОРИТМЫ 2

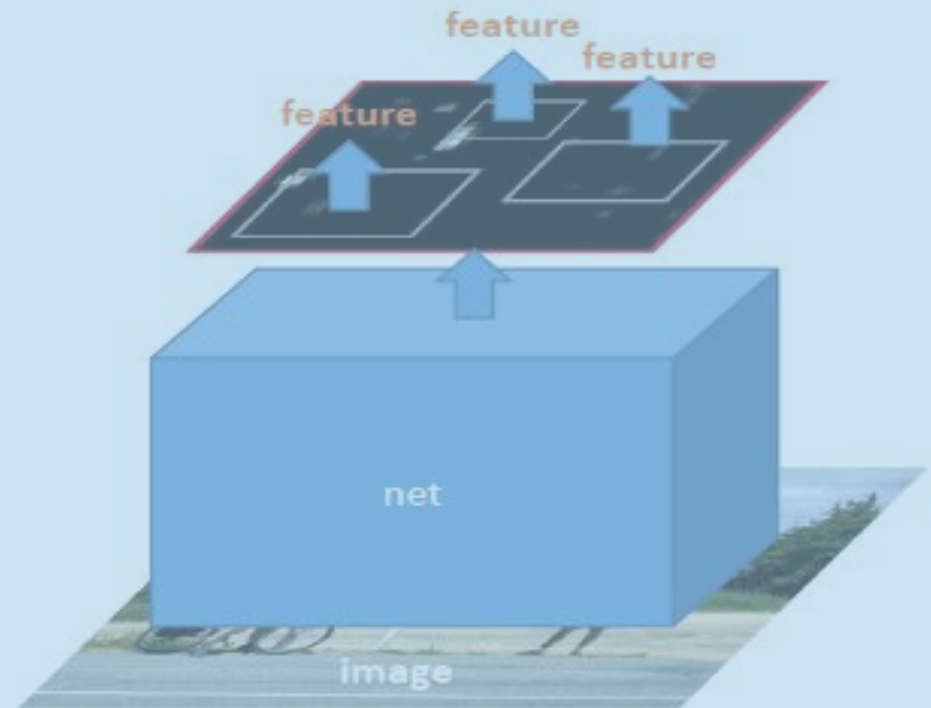
RCNN (Region Based Proposal)



R-CNN

2000 nets on image regions

SPP (Spatial Pyramid Pooling)

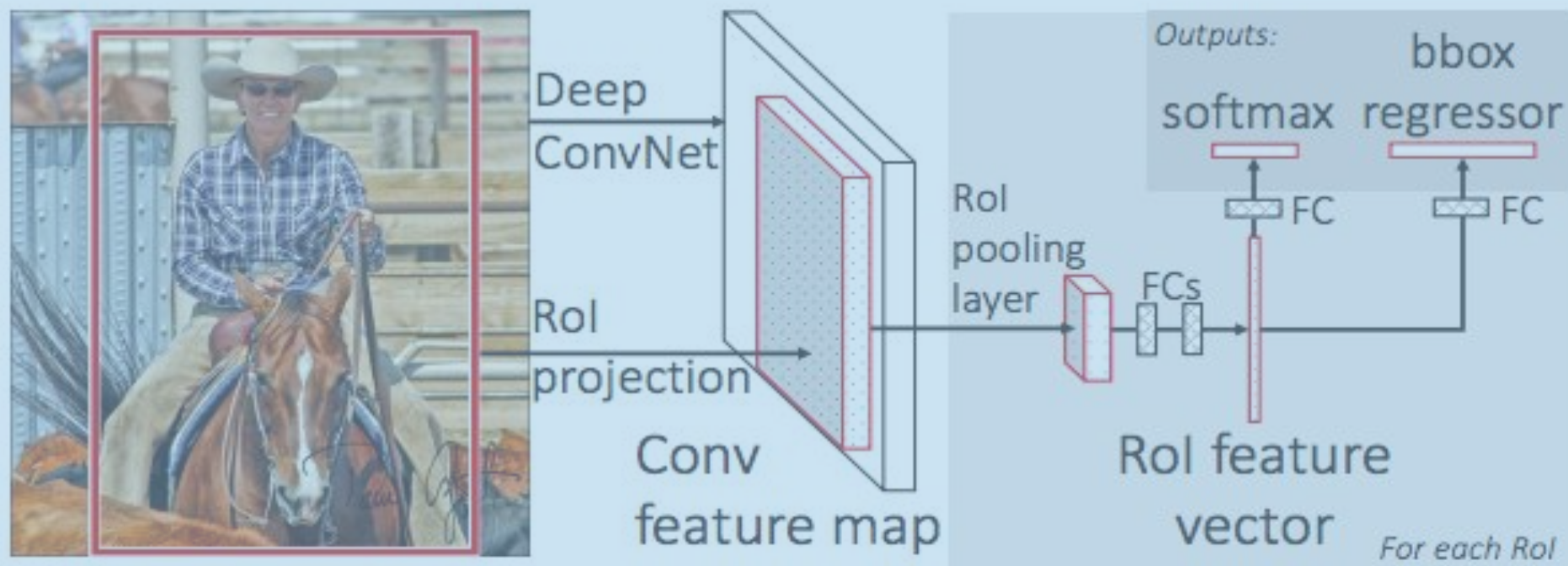


SPP-net

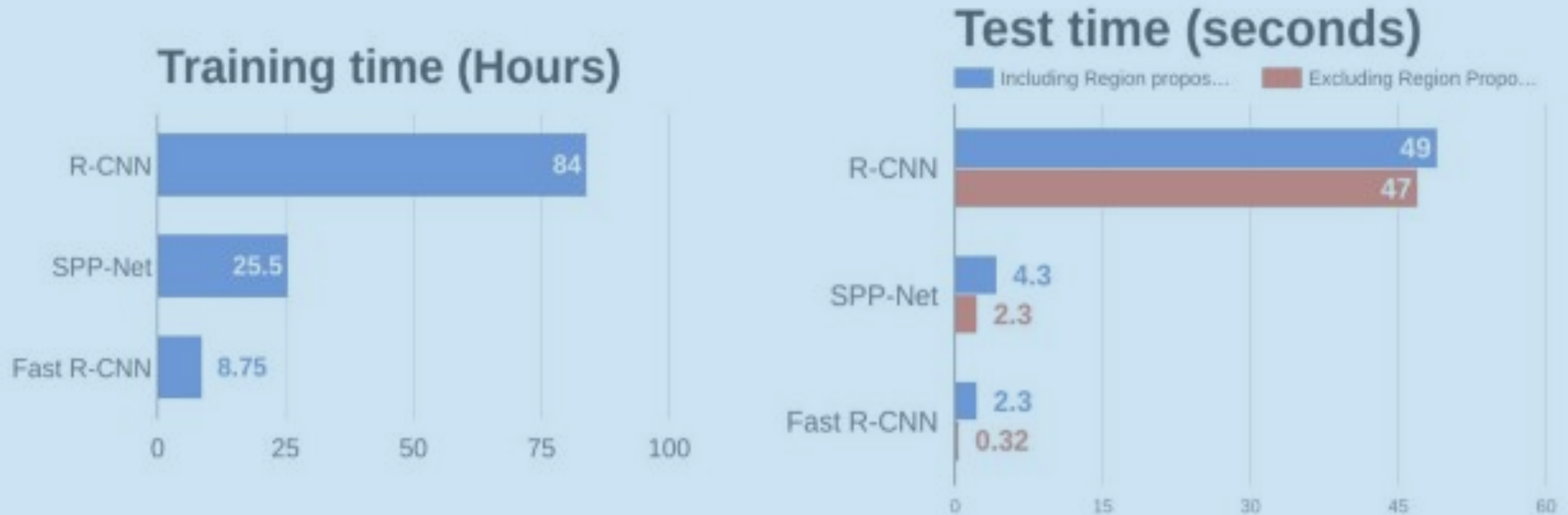
1 net on full image

Fast-RCNN - 1

Не требует дополнительных внешних классификаторов и обучается в один проход

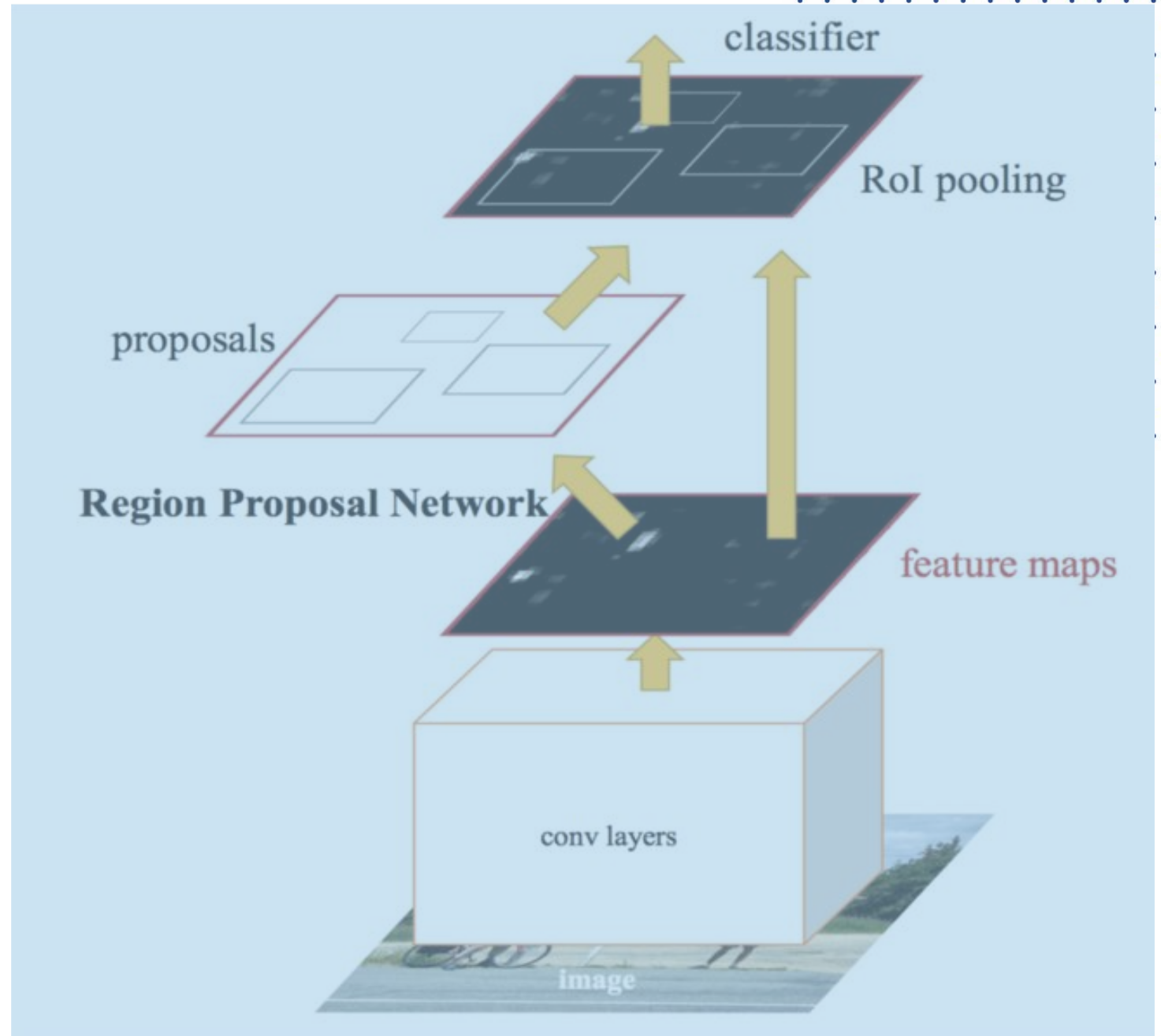


Fast-RCNN - 3

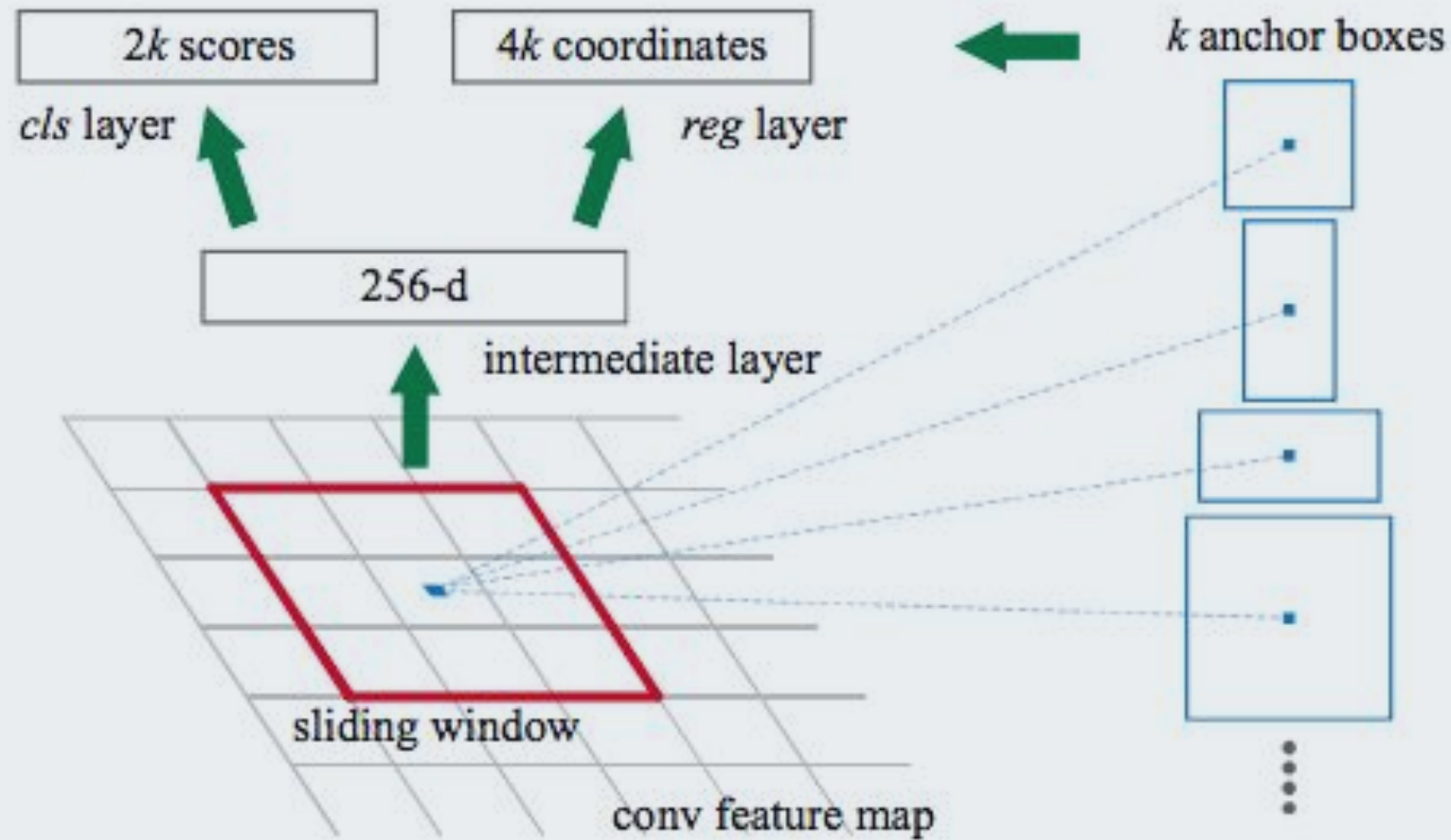


<https://arxiv.org/pdf/1504.08083.pdf>

FastER-RCNN

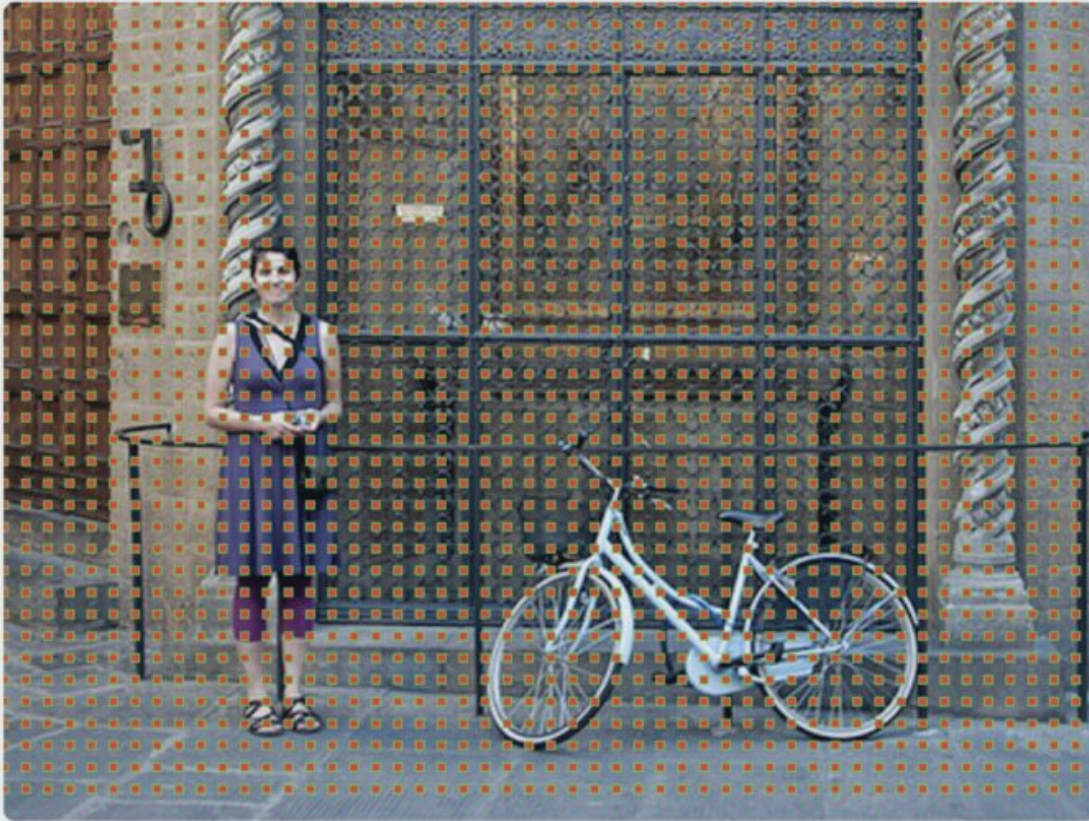


Faster-RCNN – RPN 1

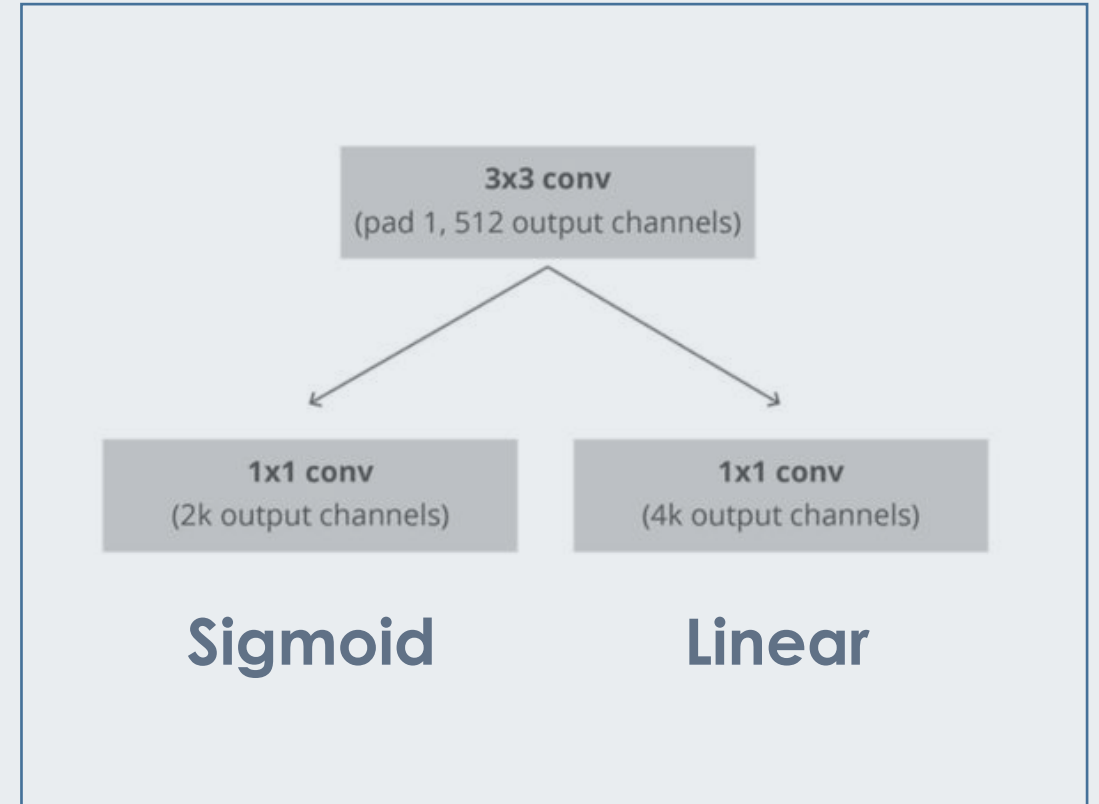


Faster-RCNN – RPN 2

Anchors:

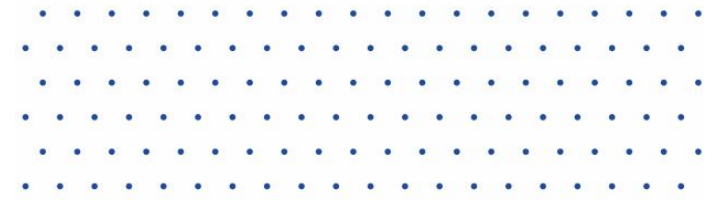


RPN LAYER:



<https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/>

Faster-RCNN – RPN 3



Стратегия обучения:

1. Изображения пропускается через conv слои базовой сети (например VGG) и возвращается карта признаков с последнего слоя сети.
2. Затем с помощью **sliding window** RPN слой проходит по карте признаков.
3. Для каждой локации, **k (k=9) anchor** создаются RPN слоем (**3 scales: 128, 256 и 512, а так же 3 aspect ratios: 1:1, 1:2, 2:1**).
4. Делим anchors на «положительные» (>0.7 IoU с bbox) и отрицательные (<0.3 IoU с bbox).
5. Для одного изображения берем 256 anchor 128 «положительных» и 128 «отрицательных» методом случайного семплирования.
6. Обучаем RPN
7. Обучаем Fast RCNN

YOLOv3

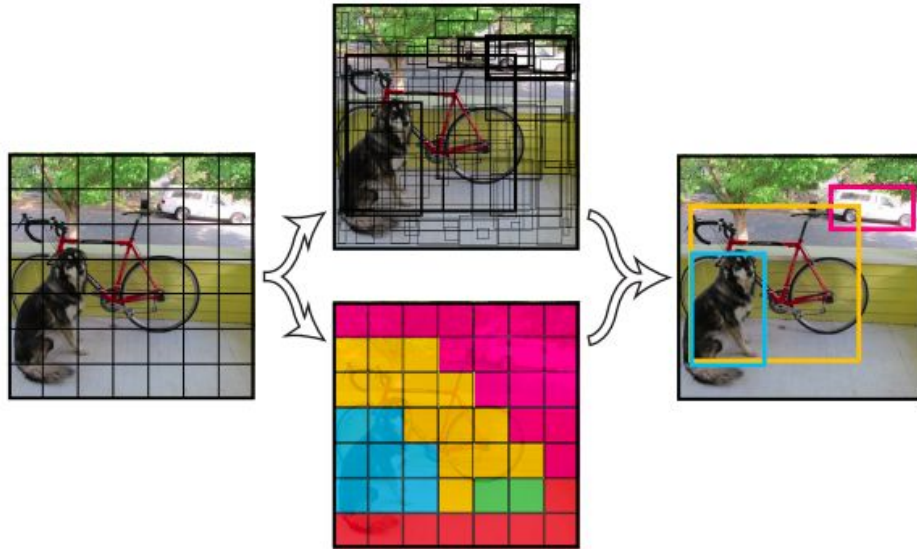
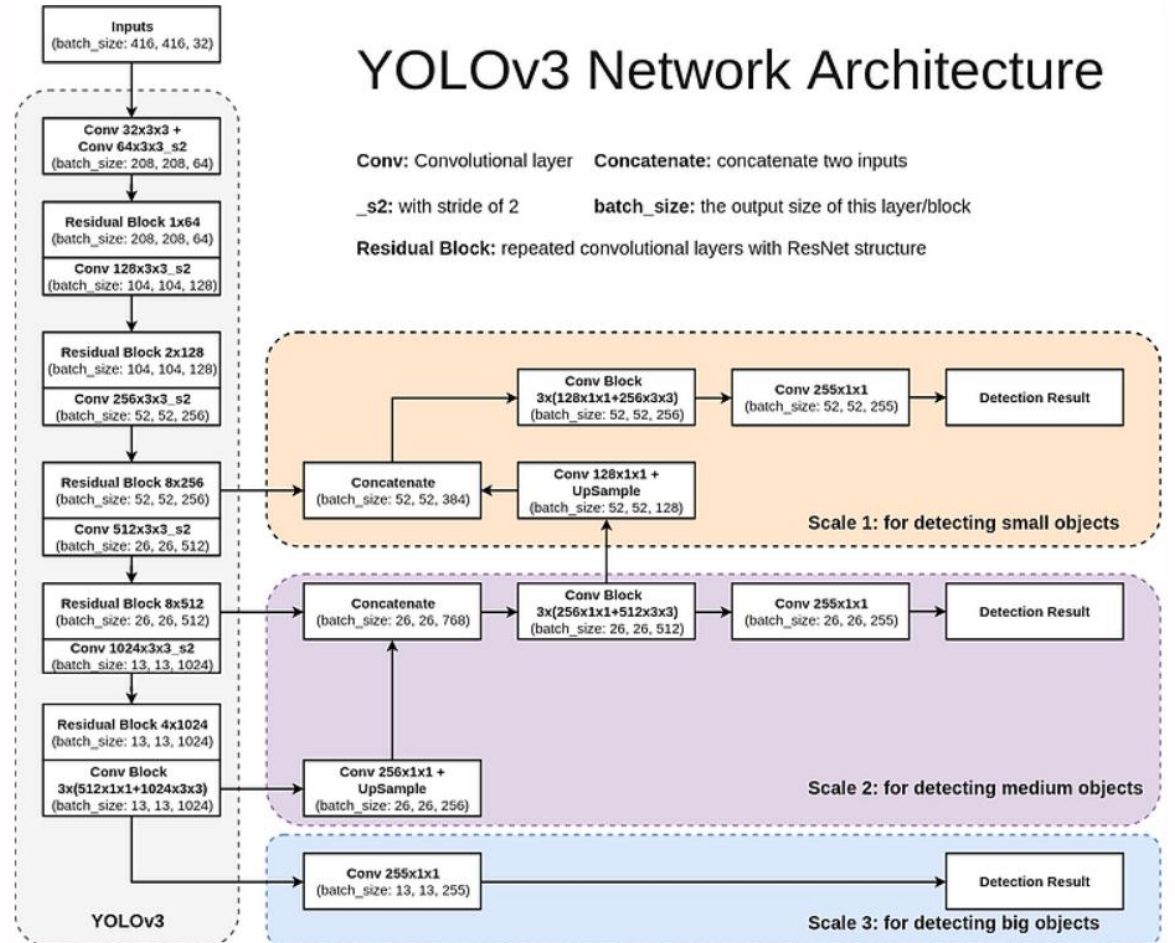


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an even grid and simultaneously predicts bounding boxes, confidence in those boxes, and class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.



<https://pjreddie.com/media/files/papers/yolo.pdf>

200 эпох:



300 эпох:



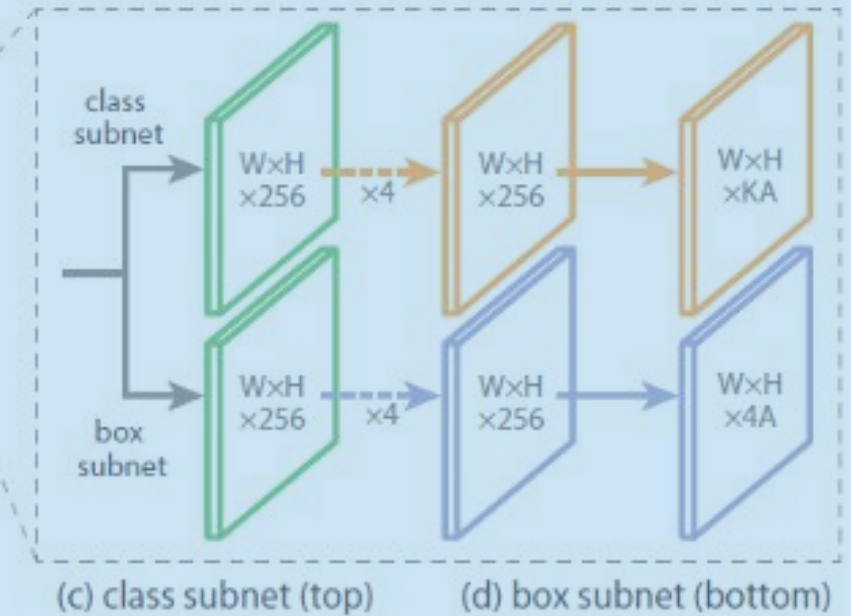
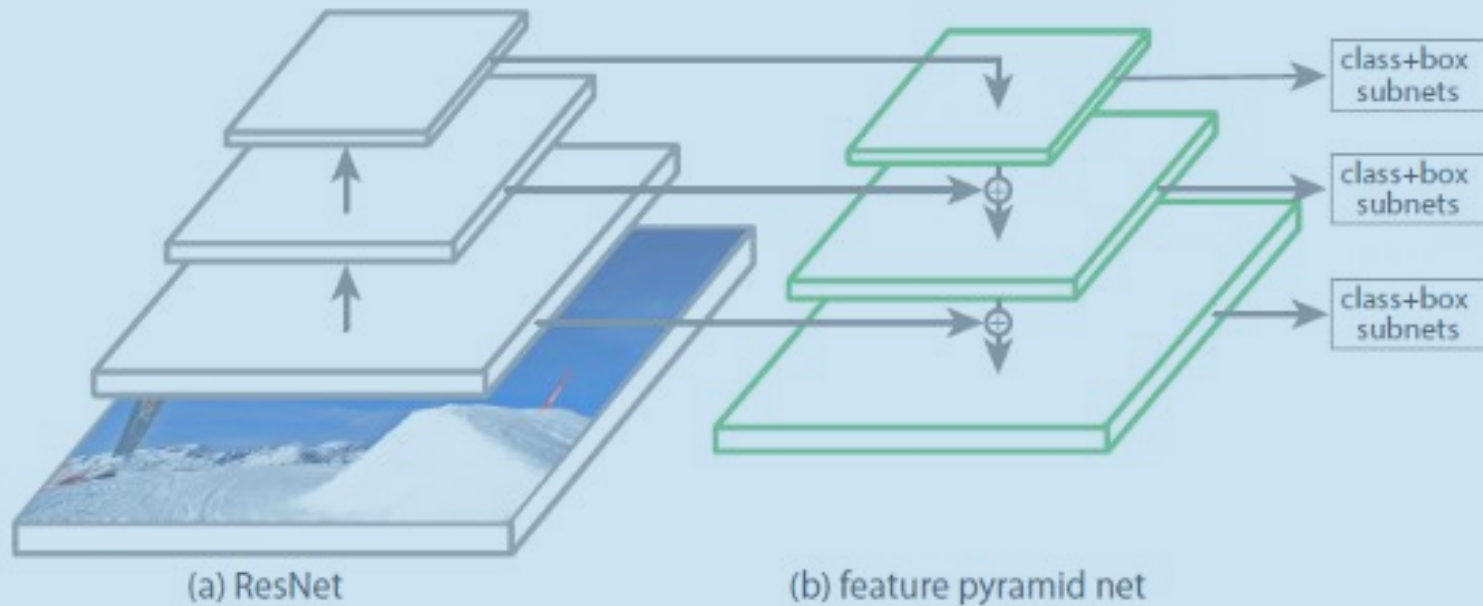
400 эпох:



500 эпох:



RetinaNET



<https://arxiv.org/pdf/1708.02002.pdf>