Другие задачи типа «что и где изображено»

Классификация + локализация

«cat»

Семантическая сегментация

(Semantic Segmentation)

GRASS, CAT, TREE, SKY

Детектирование объектов

(Object Detection)

DOG, DOG, CAT

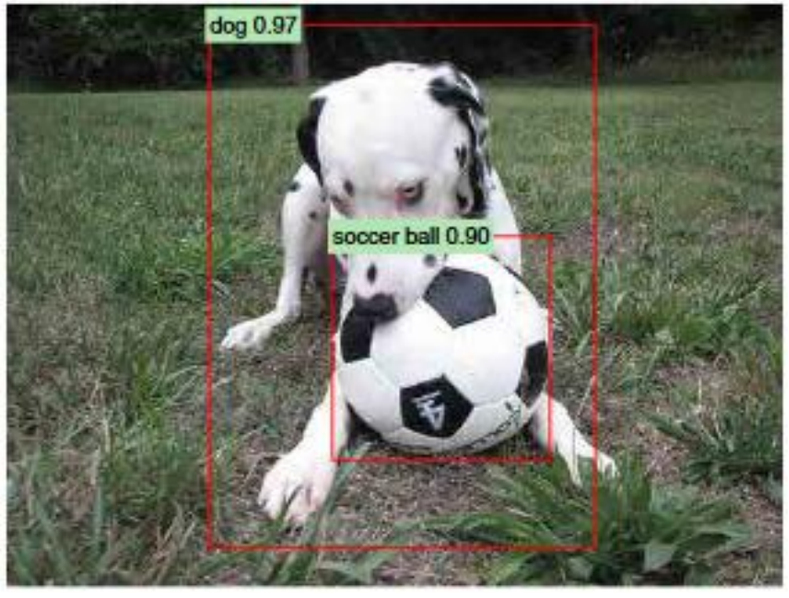
Сегментация объектов

(Instance Segmentation)

DOG, DOG, CAT

http://cs231n.stanford.edu/2017/syllabus.html

Детектирование объектов = Локализация + Классификация



Локализация (localization) объекта – где

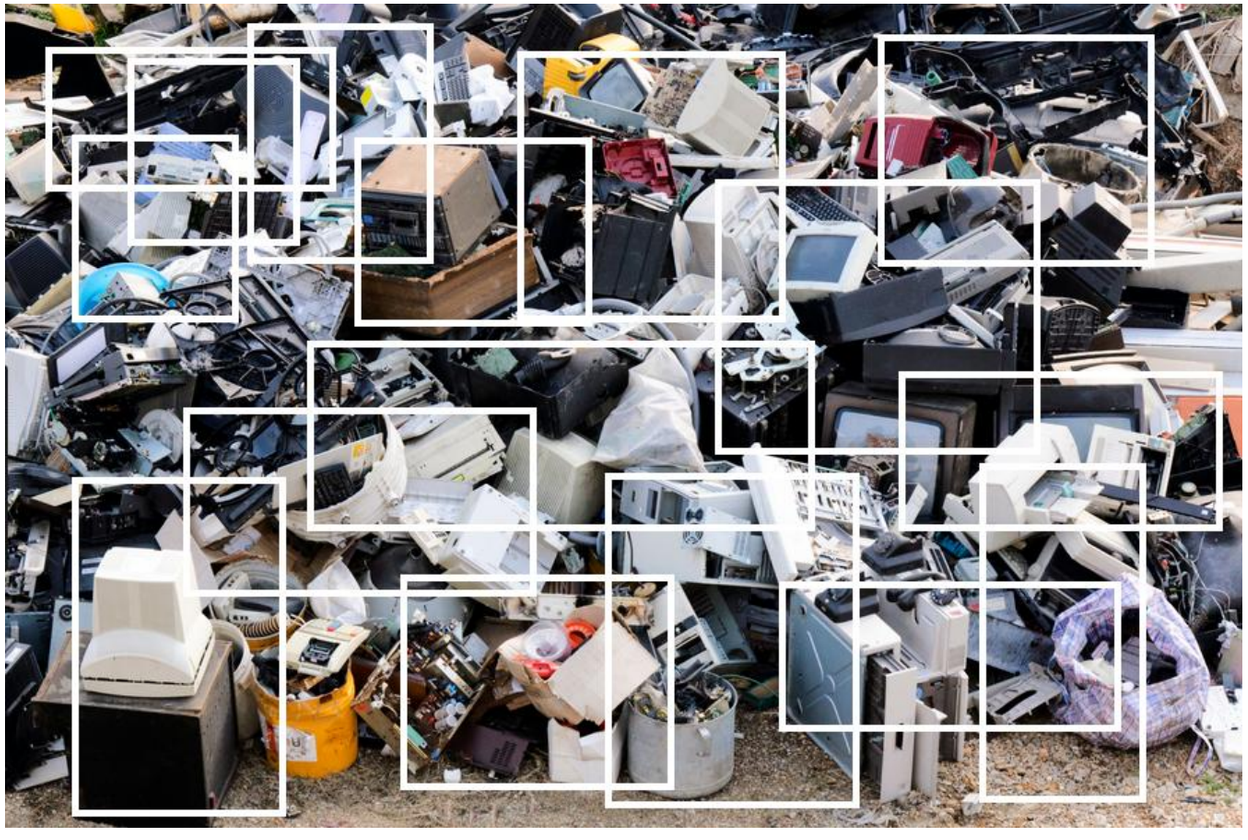
Классификация – что

м.б. ещё определяем параметры объекта

http://cv-tricks.com/object-detection/faster-r-cnn-yolo-ssd/

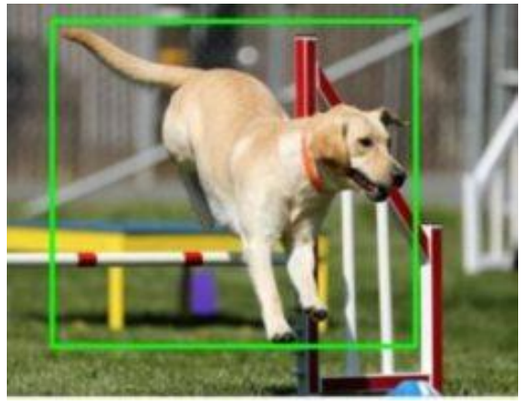
Детектирование объектов

Формально – перебрать разные локализации
(чаще всего используют прямоугольники)
и для каждого – классификация

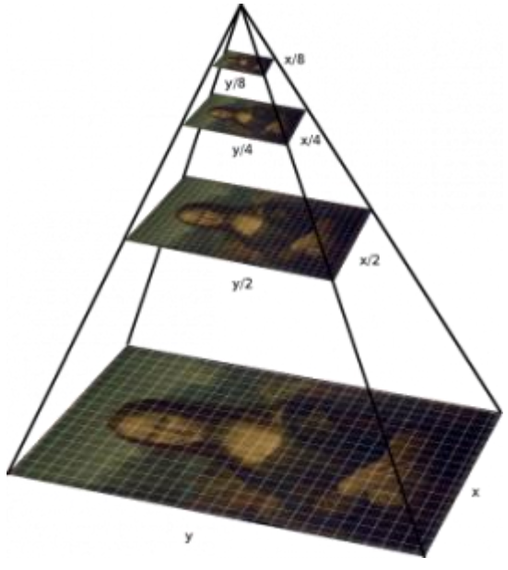


Детектирование объектов

Реально –
проблема размера



Решение –



Использовать изображения разных масштабов +
фиксированный размер области

R-CNN: работа сети

- **Выбор 2000 регионов (Selective Search)**
- **Регионы $\rightarrow$ 224×224**
- **регионы $\rightarrow$ CNN $\rightarrow$ SVM/Regression**
- **non maximum suppression (NMS) дальше**

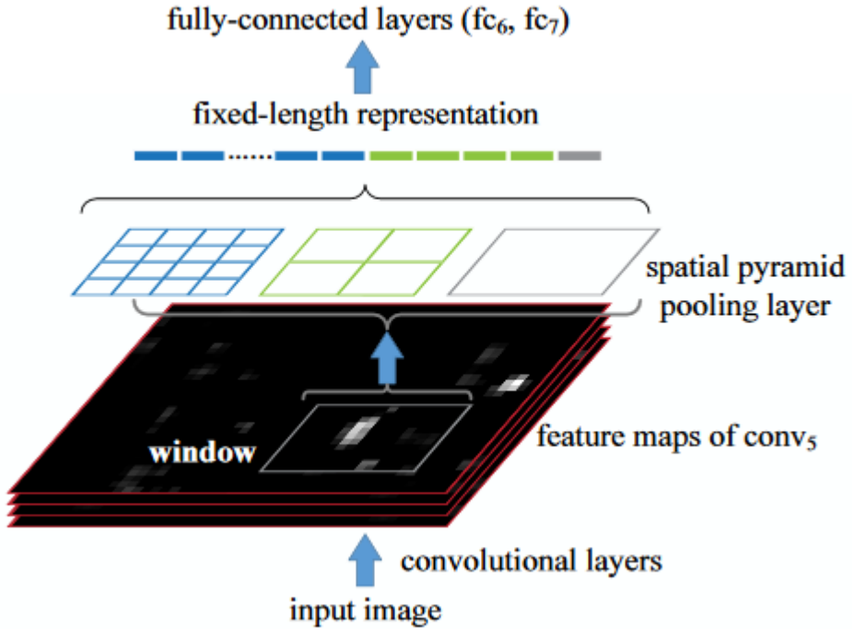
R-CNN: Недостатки

- **Дообучиваем CNN (log_loss), потом ещё SVM, потом bounding-box-регрессию…**

очень долгое обучение

- **Для каждого региона запускаем CNN**

The Spatial Pyramid Pooling Layer



Считаем CNN-представление для изображений один раз

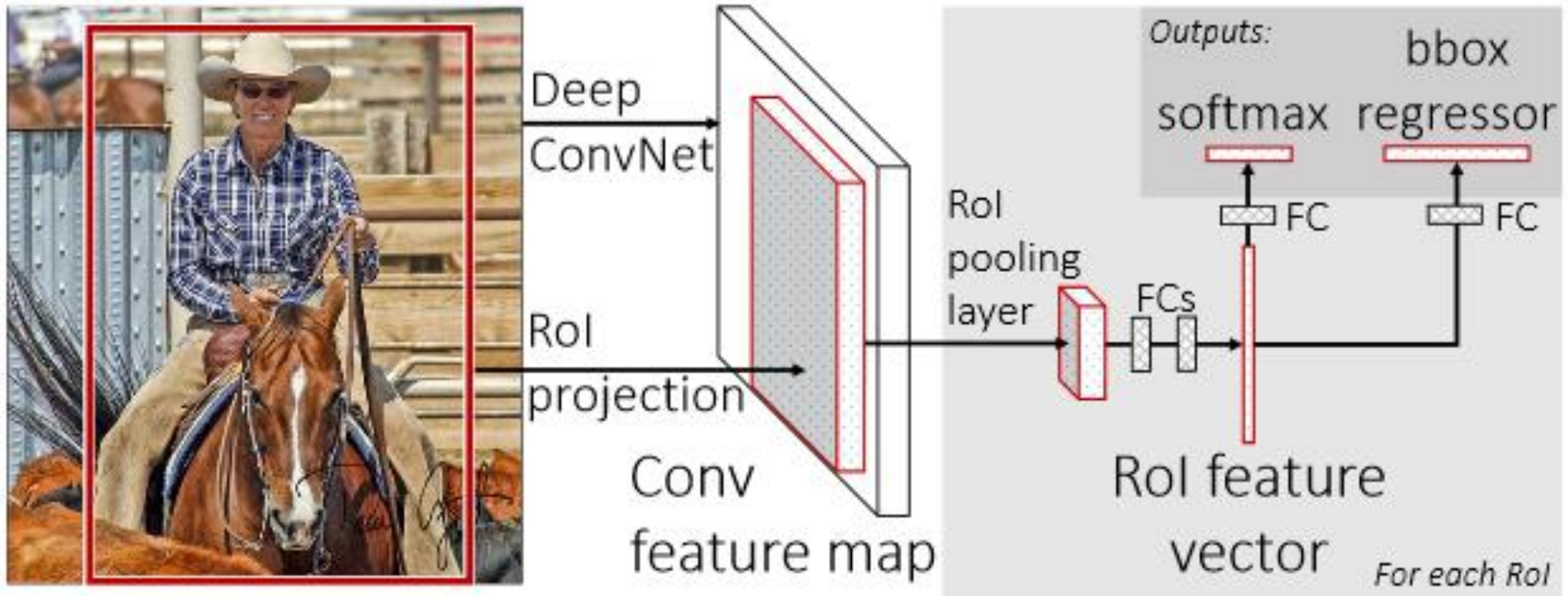
С помощью SPP-net для каждого региона получаем признаковое представление (фиксированной длины)

Быстрое решение проблемы того, что регионы разных размеров!
+ сначала-то мы считаем CNN-представление!
а SPP-net просто эффектный способ перевода его в признаки
Сеть не работает на каждом регионе! Сразу на изображении!
Выигрыш по скорости 10x – 100x!

Fast R-CNN

Fast R-CNN = R-CNN + SPP + регрессию встроили в НС

Обучение в одну стадию (раньше CNN → SVM → regression)!



Вход: изображение + параметры регионов

Регионы тоже с помощью SS

end2end: ошибка = сумма ошибок классификатора и регрессора

Fast R-CNN

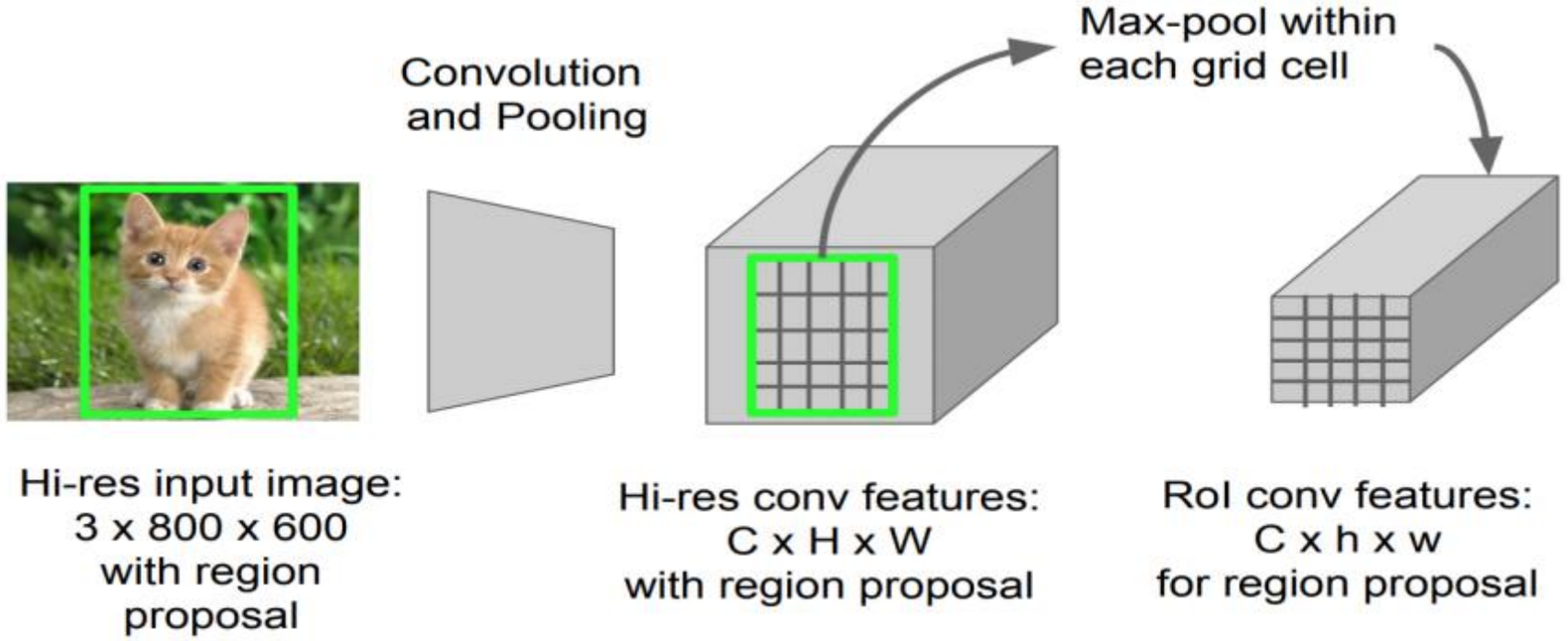
пропускаем через CNN-сеть изображение целиком

(а не каждый регион в отдельности, как раньше)

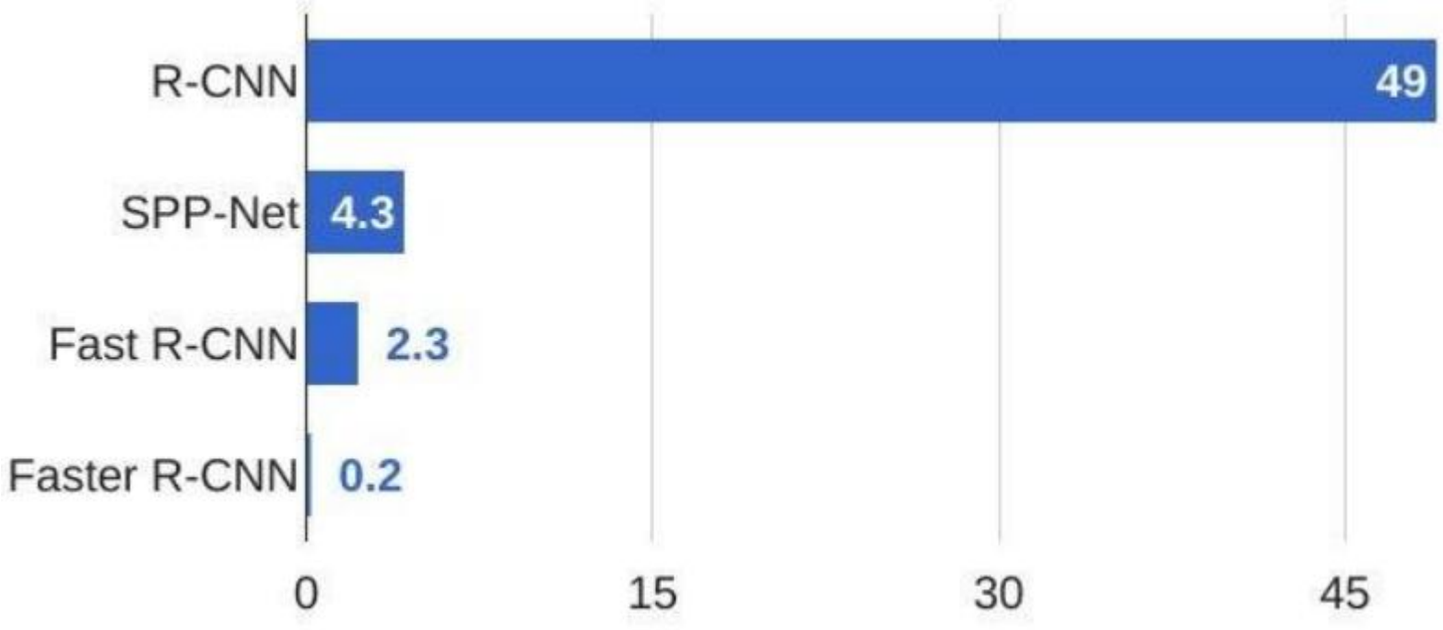
а регионы накладываются на полученную карту признаков

признаки из разных регионов приводятся в одну размерность с помощью RoIPooling

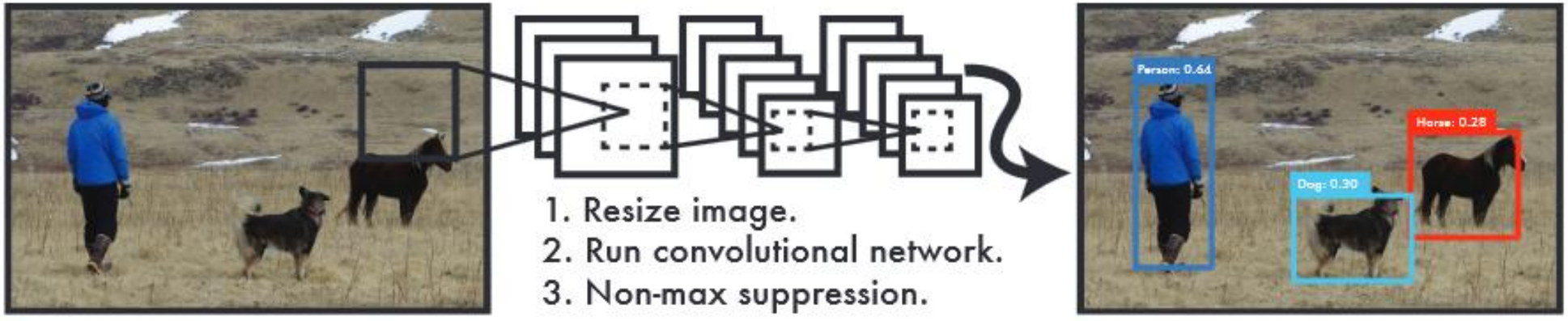
H×W → пулинг по сетке H/7×W/7 → сетка 7×7



Скорость *-R-CNN сетей



YOLO: You only ~~Live~~ Look Once



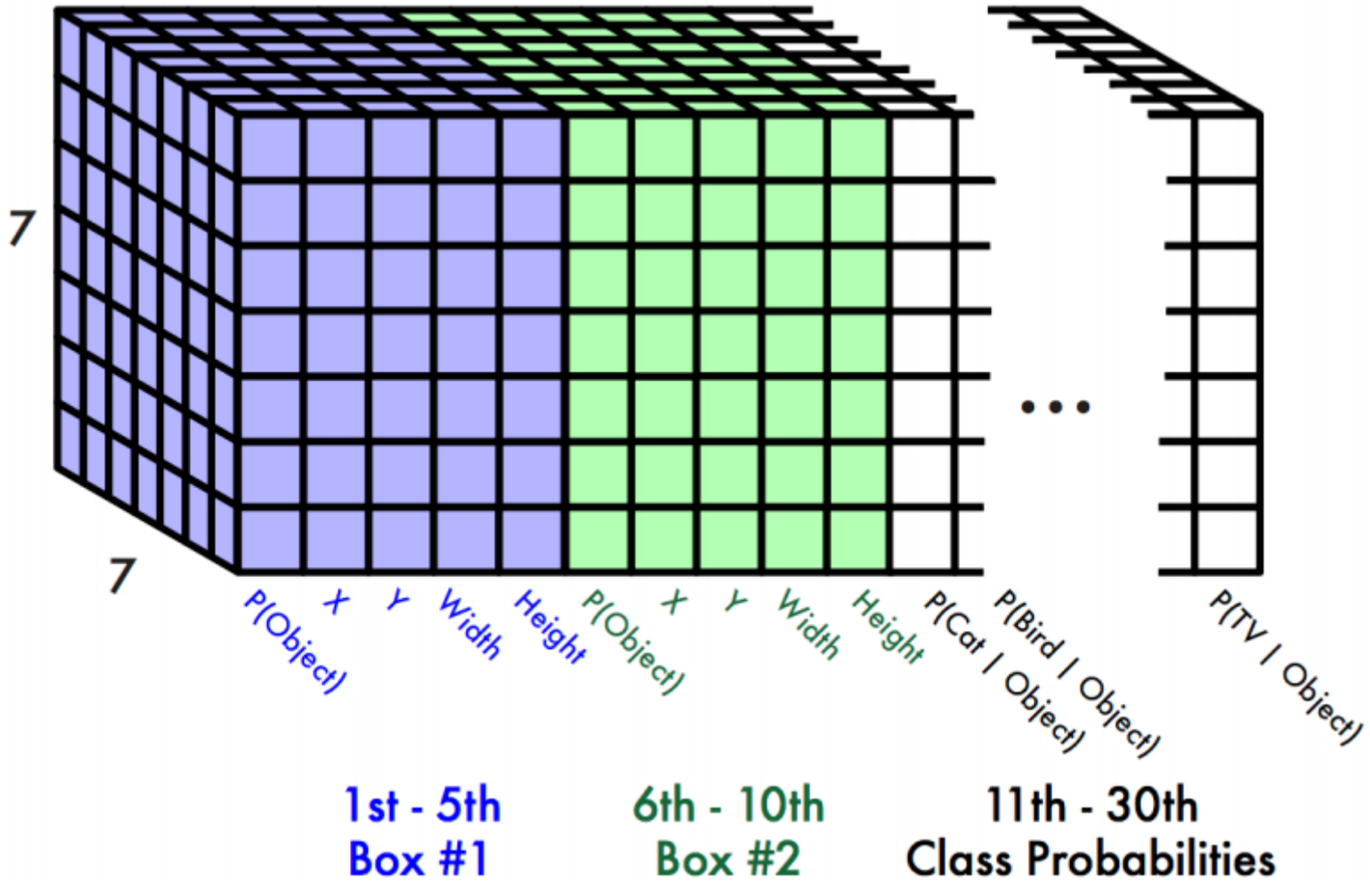
- **Изменение масштаба → 448×448**
- **CNN (одна!)**
- **Детектирование – задача регрессии;)**
- **Пороговое принятие решения**
- **Запуск сразу на всём изображении – очень быстро**

Сеть видит изображение целиком, а не регионами

Очень быстрая, но точность хуже (особенно для мелких объектов)

https://pjreddie.com/media/files/papers/yolo_1.pdf

YOLO: выход модели



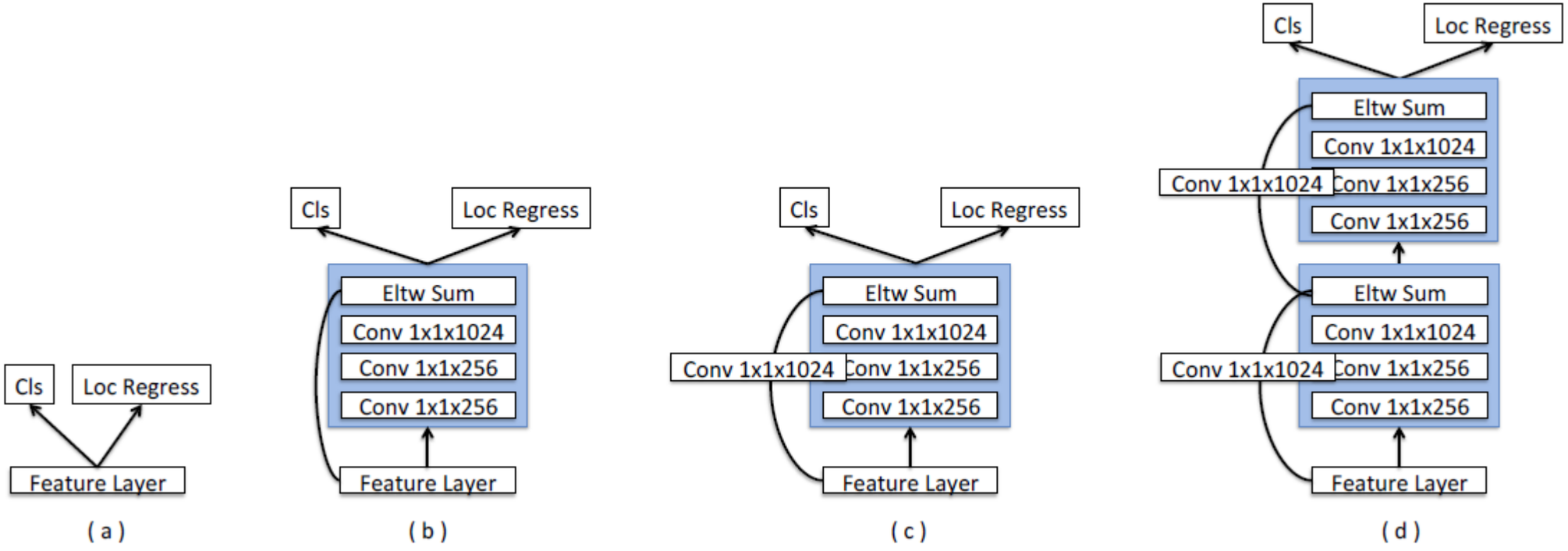
В тензоре 7×7×30 закодированы все регионы оценки за классы

7×7 – это сетка;)

30 = 5 + 5 +
(почему-то 2 региона для каждой ячейки)
+ 20 (# классов ~ Pascal VOC)

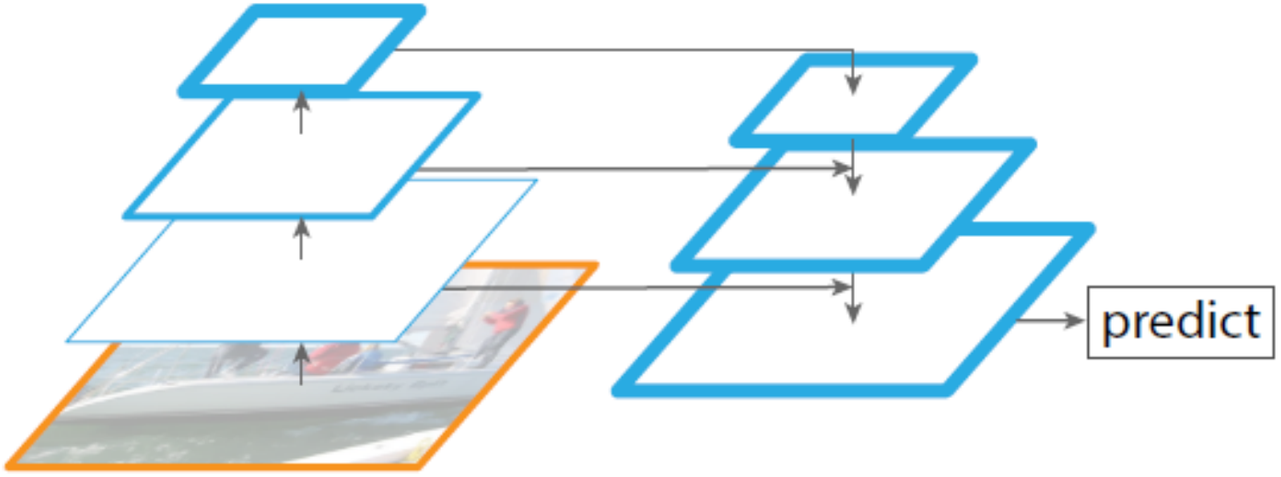
5 = |(x, y, w, h, c)|
x, y – координаты в центре соотв. ячейки
c – уверенность, что регион правильный

DSSD: Prediction Module



Были попробованы разные варианты

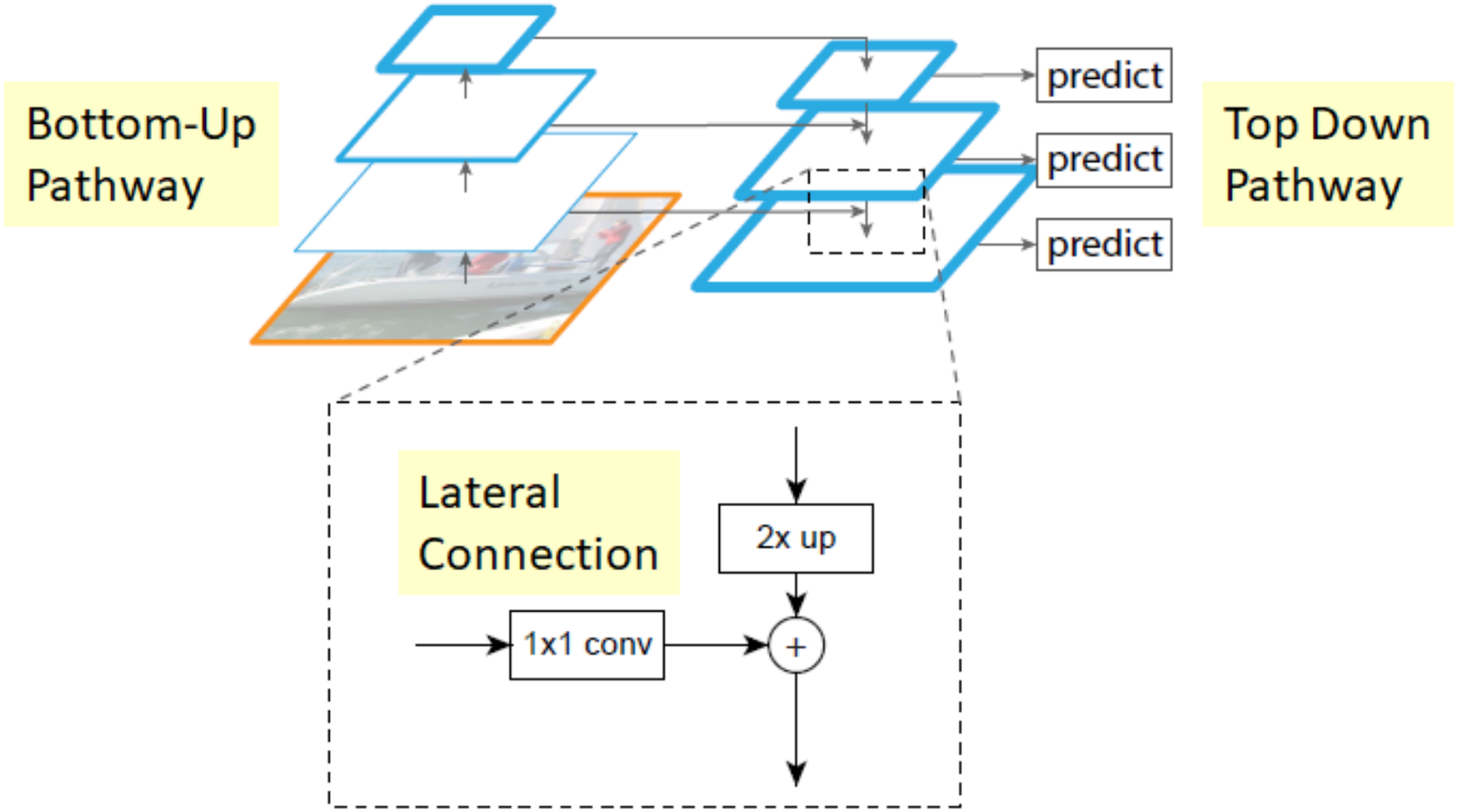
Разные архитектуры



(e) Similar Structure with (d)

Иногда применяется схожая архитектура

Feature Pyramid Networks (FPN)

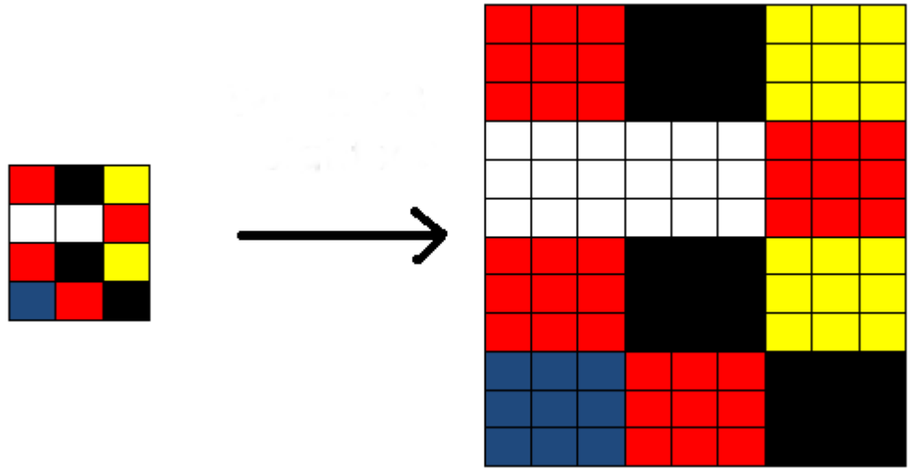


предсказания на разных уровнях

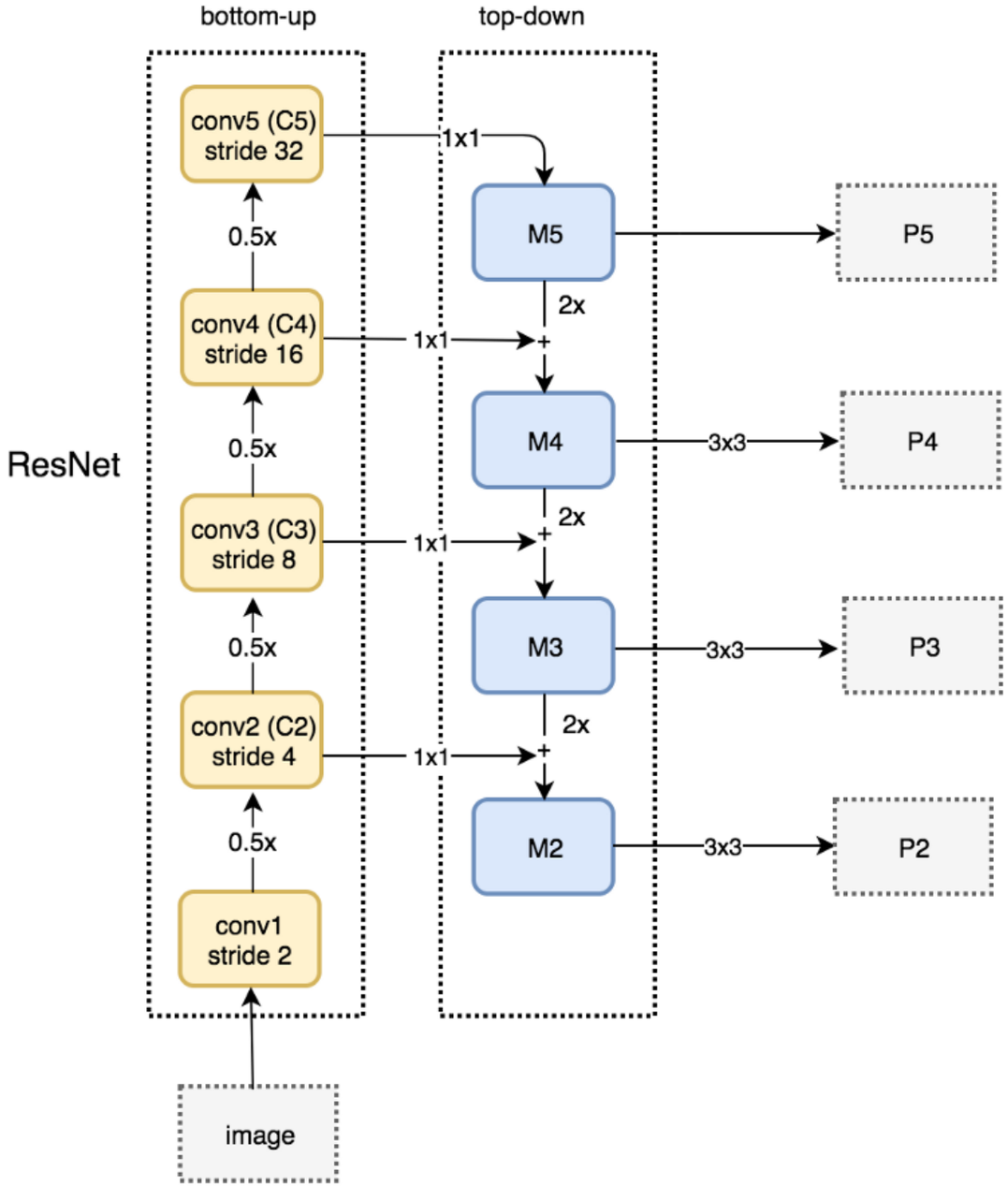
для одинаковости размера (256) свёртки 1×1

Top-down

upsampling × 2 с помощью ближайшего соседа



lateral connection складывает признаковые карты одинакового пространственного размера



Bottom-up

Используется ResNet

На уровнях пространственное разрешение уменьшается в 2 раза

P1 нет из-за слишком большой пространственной размерности

Это не object-detector, а построение признаков

План: задачи с изображениями

Классификация – что изображено
Локализация – где изображено
Детектирование – что и где

Сегментация – матрица меток сегментов
семантическая сегментация / сегментация объектов

Преобразование изображений
- **удаление шума**
- **стилизация**

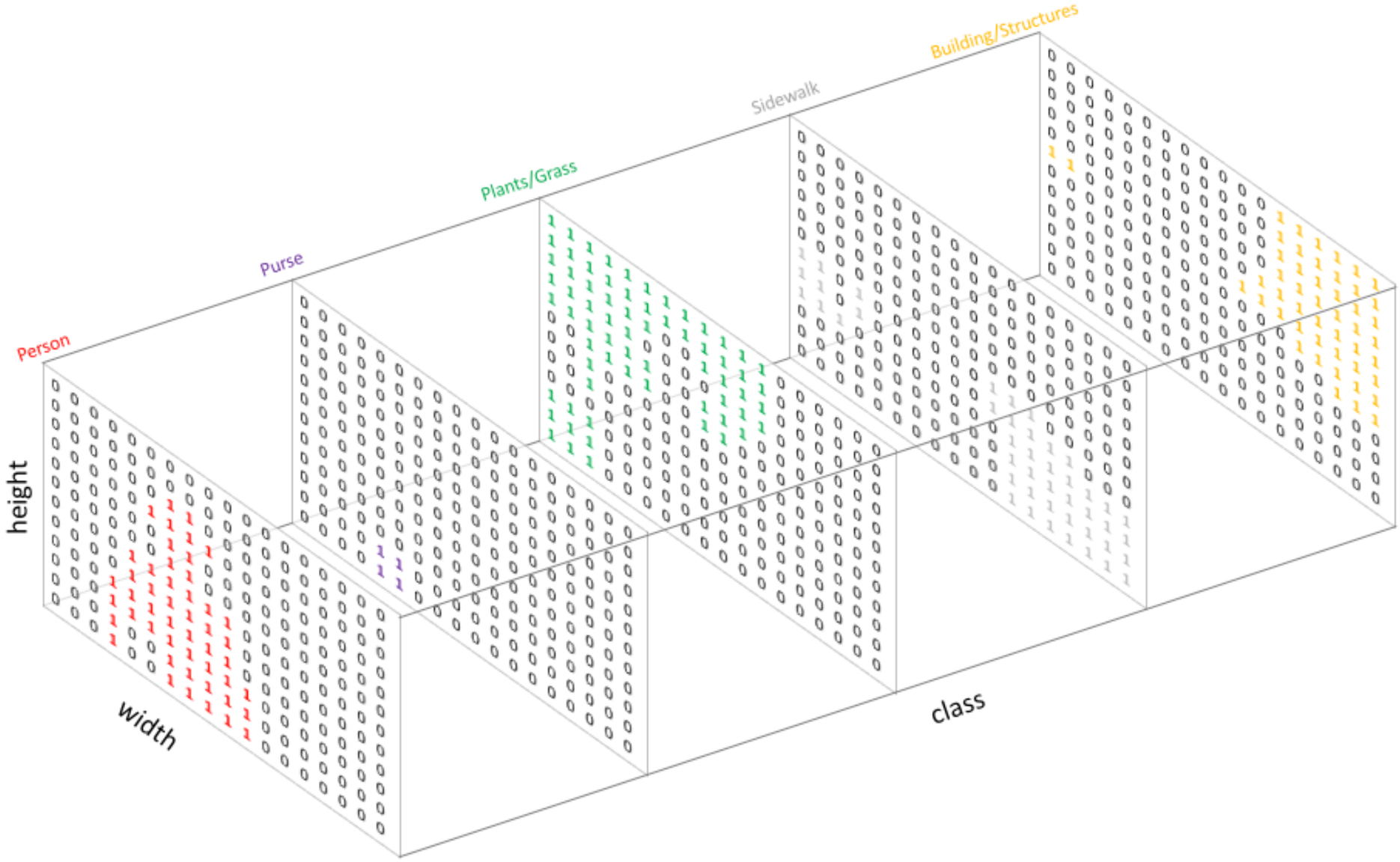
Восстановление объектов (ex: 3D-модели)

Семантическая сегментация: приложения

- беспилотное вождение
- медицинские изображения
- изображения со спутников
- для других задач (например ИИ)
- извлечение изображения отдельных объектов (киноиндустрия)

Семантическая сегментация

ОНЕ-кодирование целевого вектора



Обратная свёртка (upconvolution / conv-transpose)

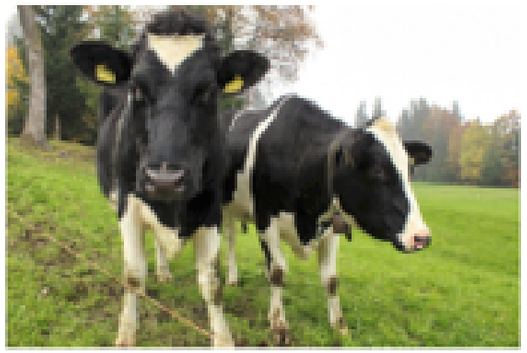
Можно...

$$\begin{pmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{pmatrix} *^{\mathrm{T}} \begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix} = H^{\mathrm{T}} \cdot \begin{pmatrix} z_{11} \\ z_{21} \\ z_{12} \\ z_{22} \end{pmatrix}$$

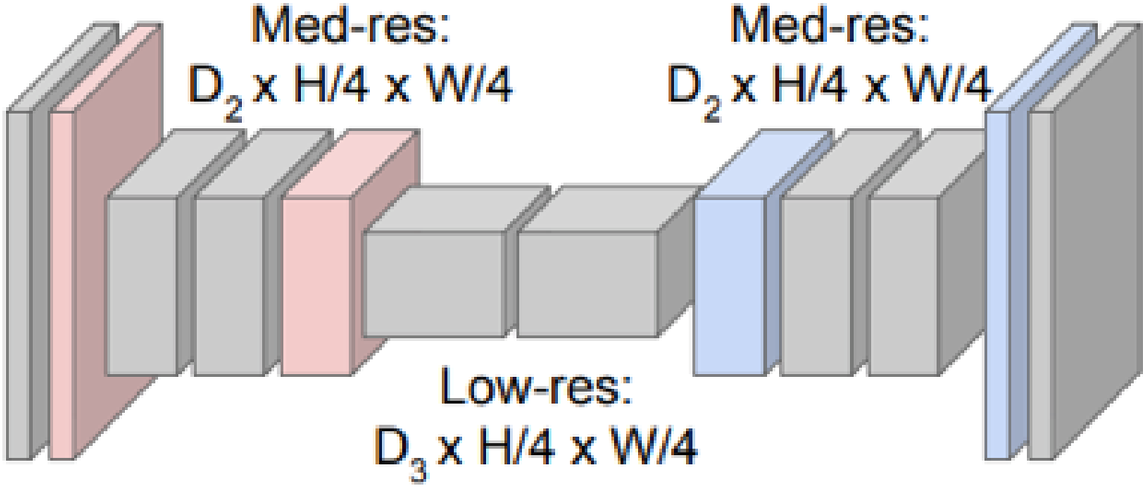
Обратная свёртка увеличивает пространственное разрешение...

Семантическая сегментация

В итоге что-то такое...



Input:
3 x H x W



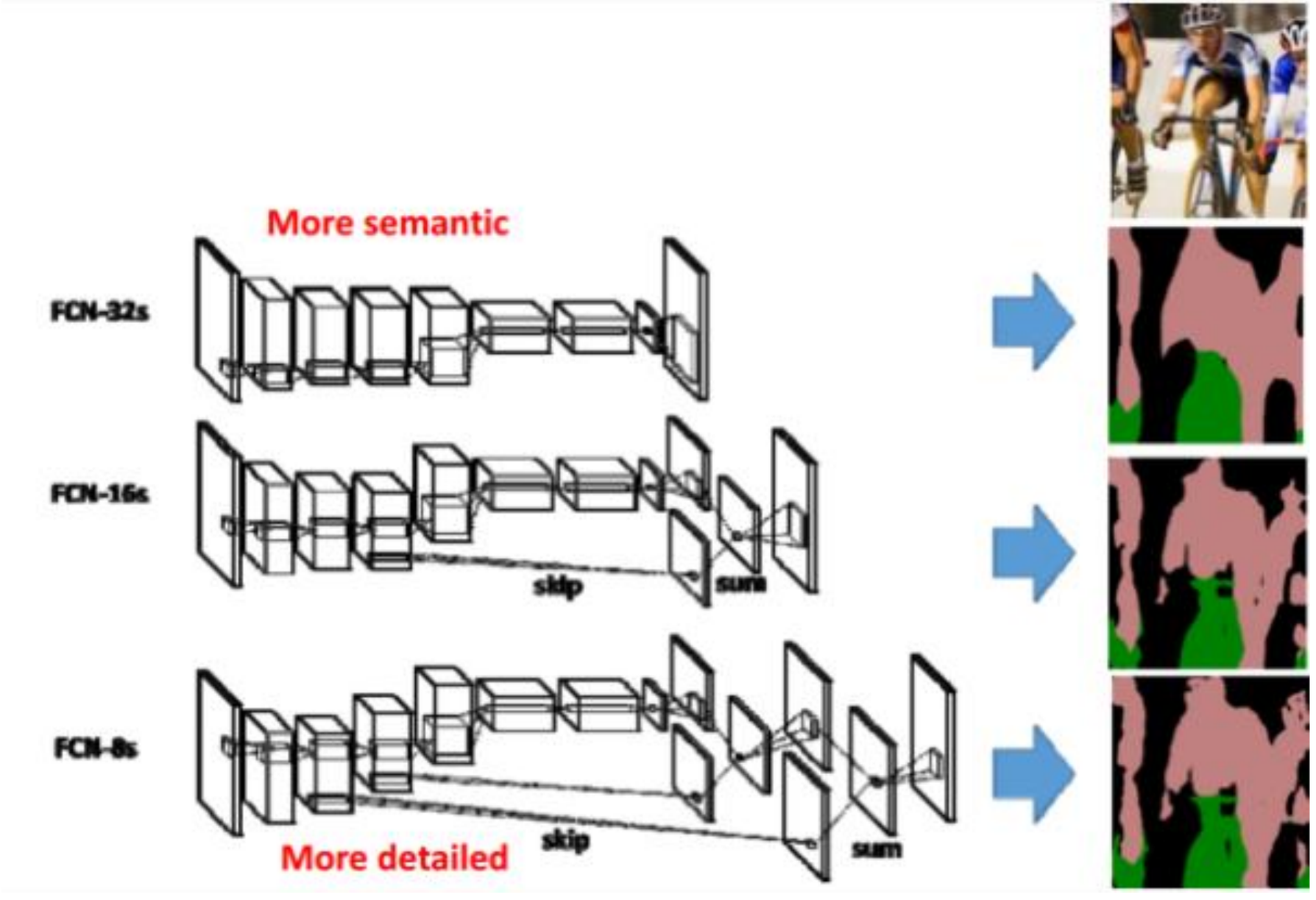
High-res:
D1 x H/2 x W/2

High-res:
D1 x H/2 x W/2

Predictions:
H x W

уменьшаются размеры, но увеличивается число каналов!

Эффект прокидывания связей



TernausNet

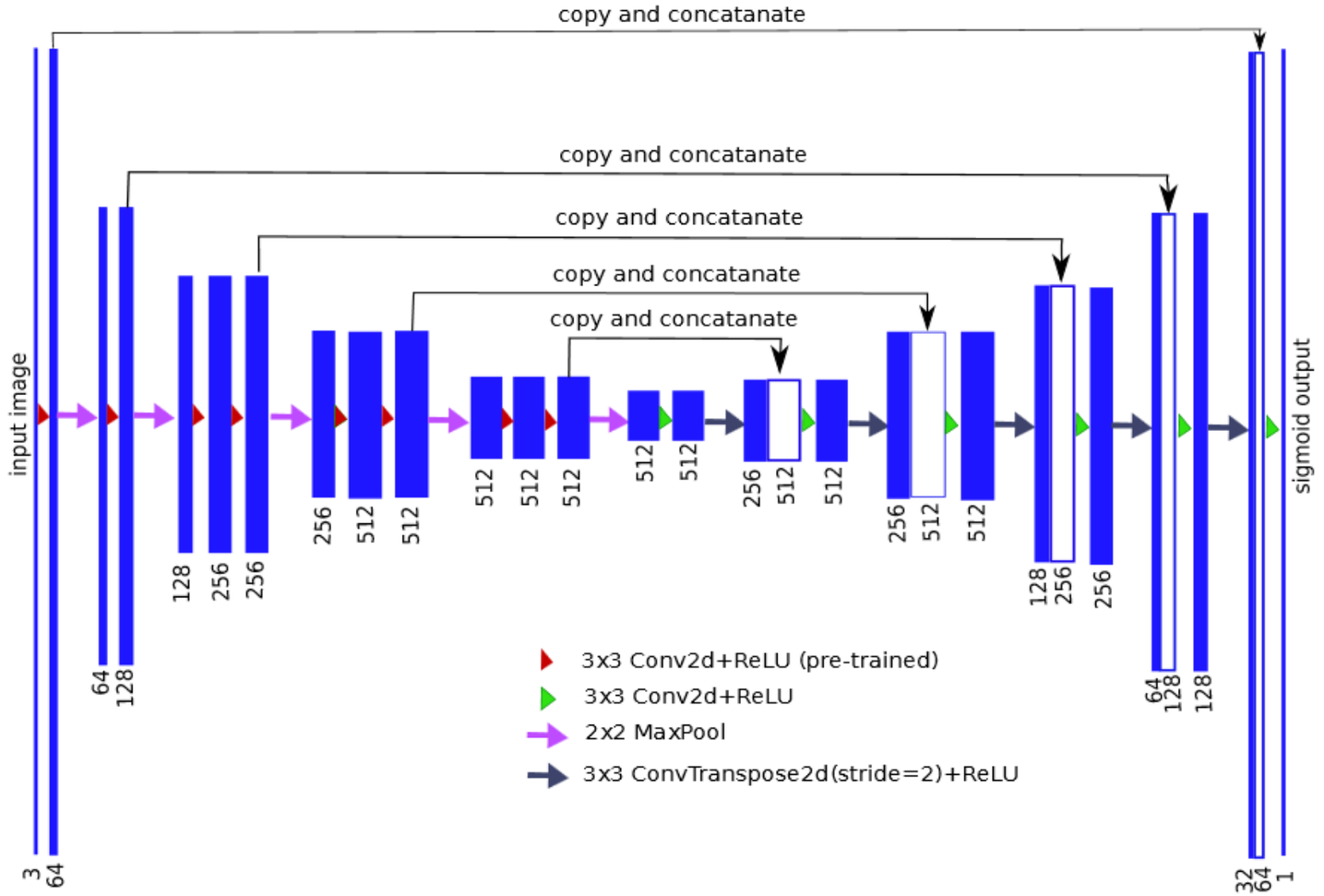
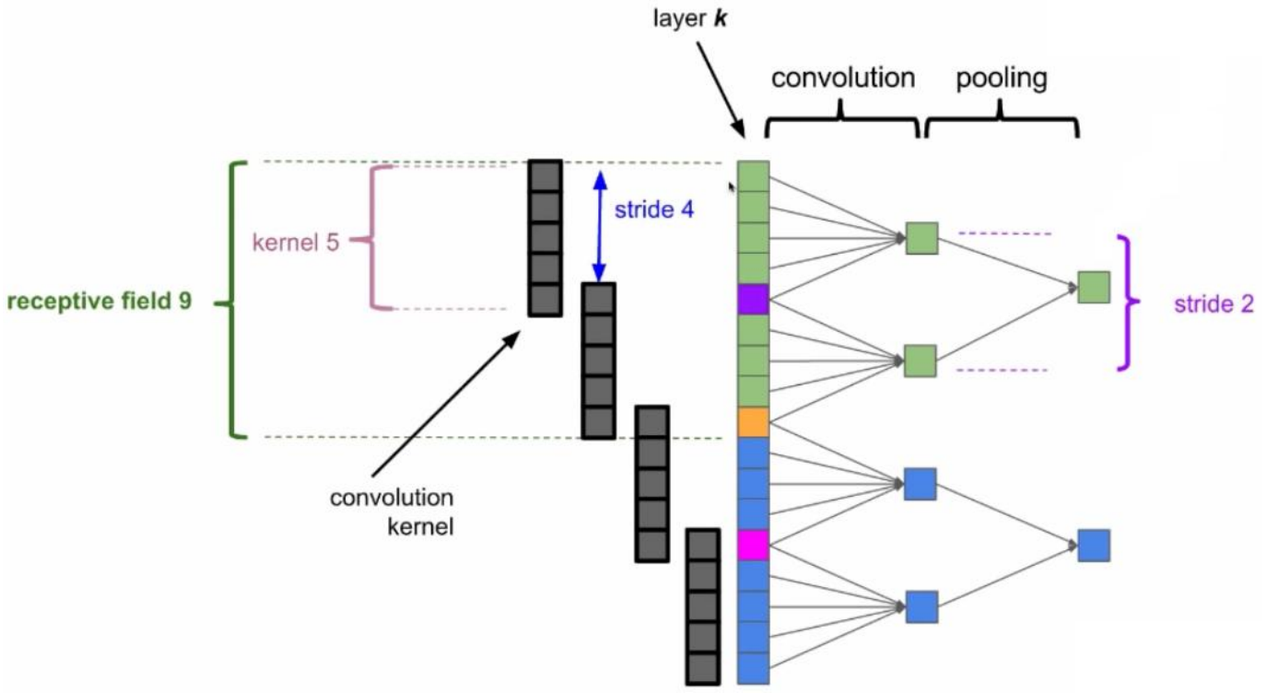
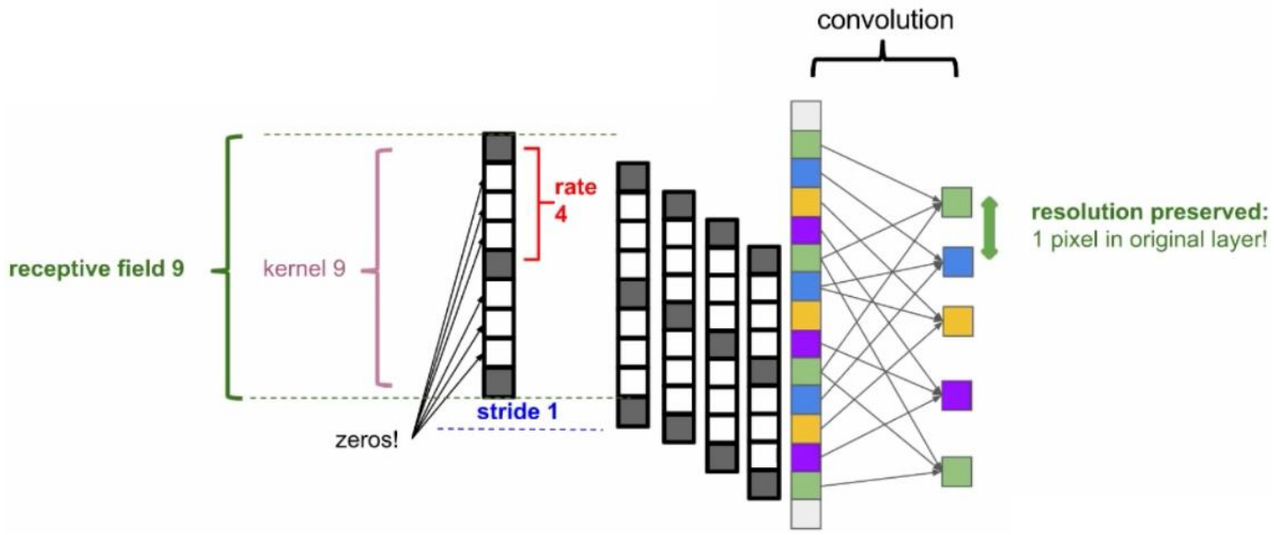


Fig. 1. Encoder-decoder neural network architecture also known as U-Net where VGG11 neural network without fully connected layers as its encoder. Each blue rectangular block represents a multi-channel features map passing through a series of transformations. The height of the rod shows a relative map size (in pixels), while their widths are proportional to the number of channels (the number is explicitly subscribed to the corresponding rod). The number of channels increases stage by stage on the left part while decrease stage by stage on the right decoding part. The arrows on top show transfer of information from each encoding layer and concatenating it to a corresponding decoding layer.

использование обученной VGG11 в качестве кодировщика https://arxiv.org/abs/1801.05746

Расширенные свёртки (Dilated convolutions / Atrous Convolutions)





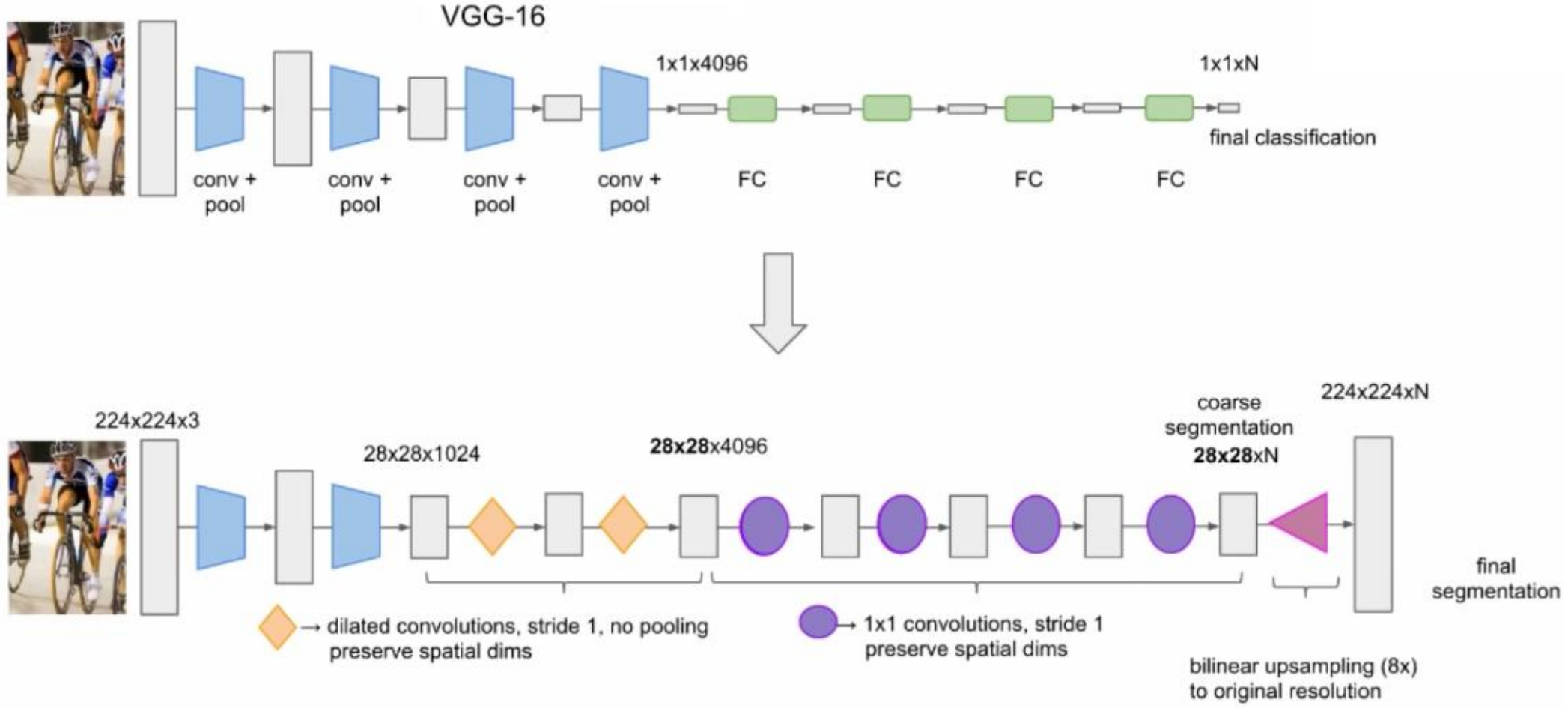
нет пулинга

получаем большую, но разреженную свёртку

вычислительная сложность ~ число ненулей

при одинаковом размере модели можем сделать любое рецептивное поле

DeepLabv1/2: использование расширенных свёрток

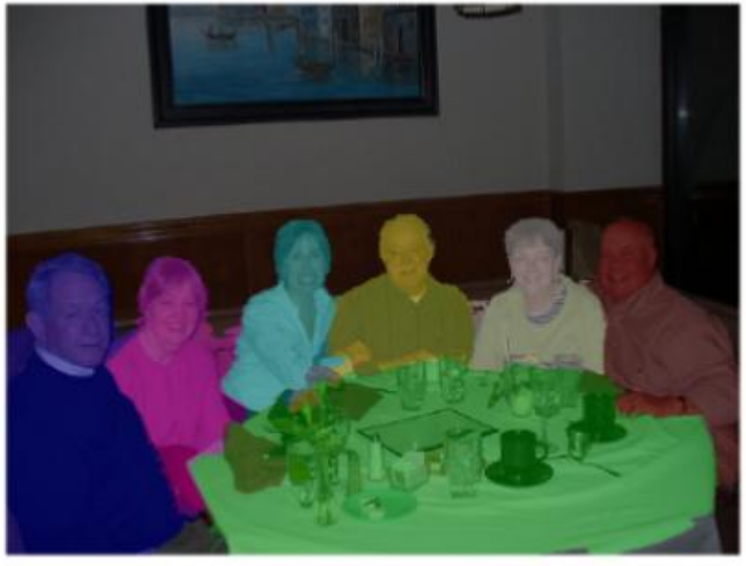


нет пулинга – последние свёртка + пулинг заменили на расширенные свёртки (stride=1)

постпроцессинг – CRF

Сегментация объектов (Instance segmentation)

Semantic Segmentation



Instance Segmentation

в семантической сегментации каждому пикселю – класс
в сегментации объектов надо различать группы пикселей формально одного класса, но принадлежащие разным объектам

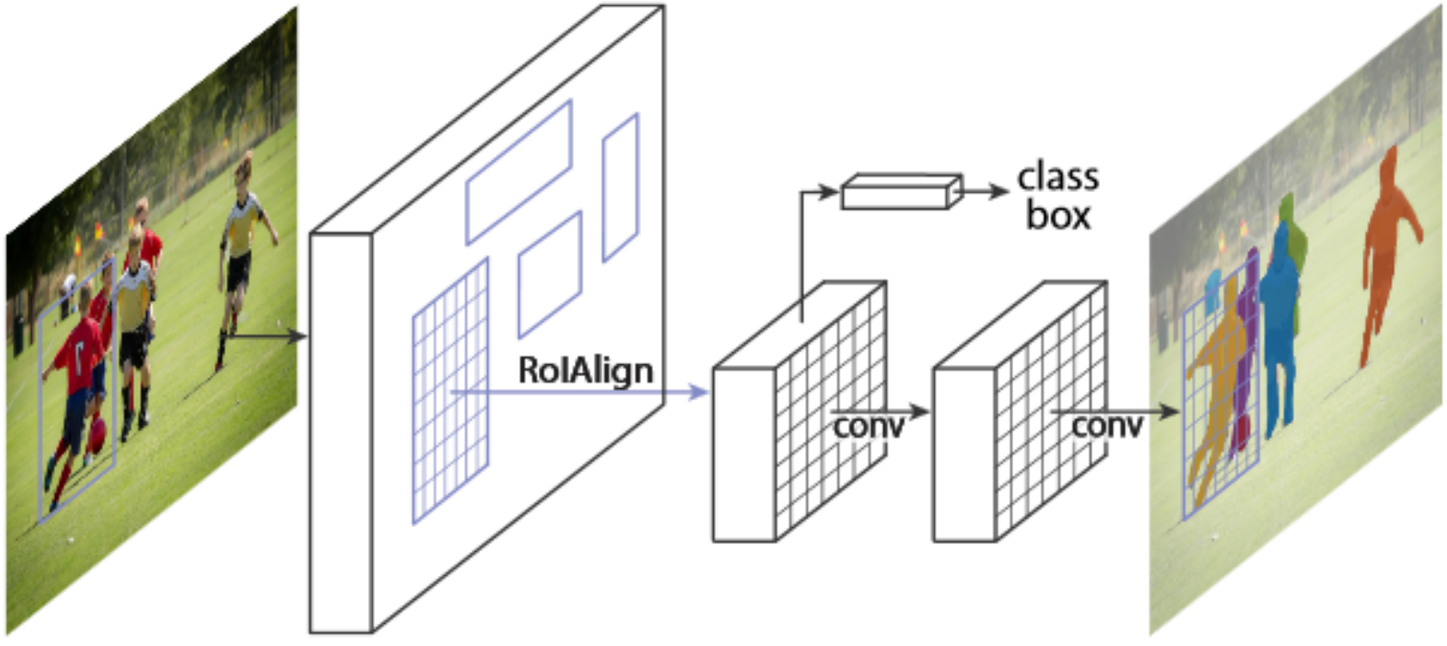
https://neurohive.io/ru/osnovy-data-science/semantic-segmention/

Сегментация объектов (Instance segmentation)

не просто найти рамку, а выделить объект!

Mask R-CNN https://arxiv.org/pdf/1703.06870.pdf

Mask R-CNN



Faster R-CNN + определение сегментационной маски (маленькая FCN)

+ борьба за более точное определение границ

Хороши также для определения позы

специальные маски для детектирования опорных точек (локоть, плечо и т.п.)

– один пиксель =1, остальные =0