

# Нейронные сети

## Задача сегментации

Лазарева Елизавета



Не забывайте  
отмечаться и  
оставлять  
отзыв



# Содержание лекции

1. Задача сегментации и ее виды
2. Семантическая сегментация
3. Объектная сегментация



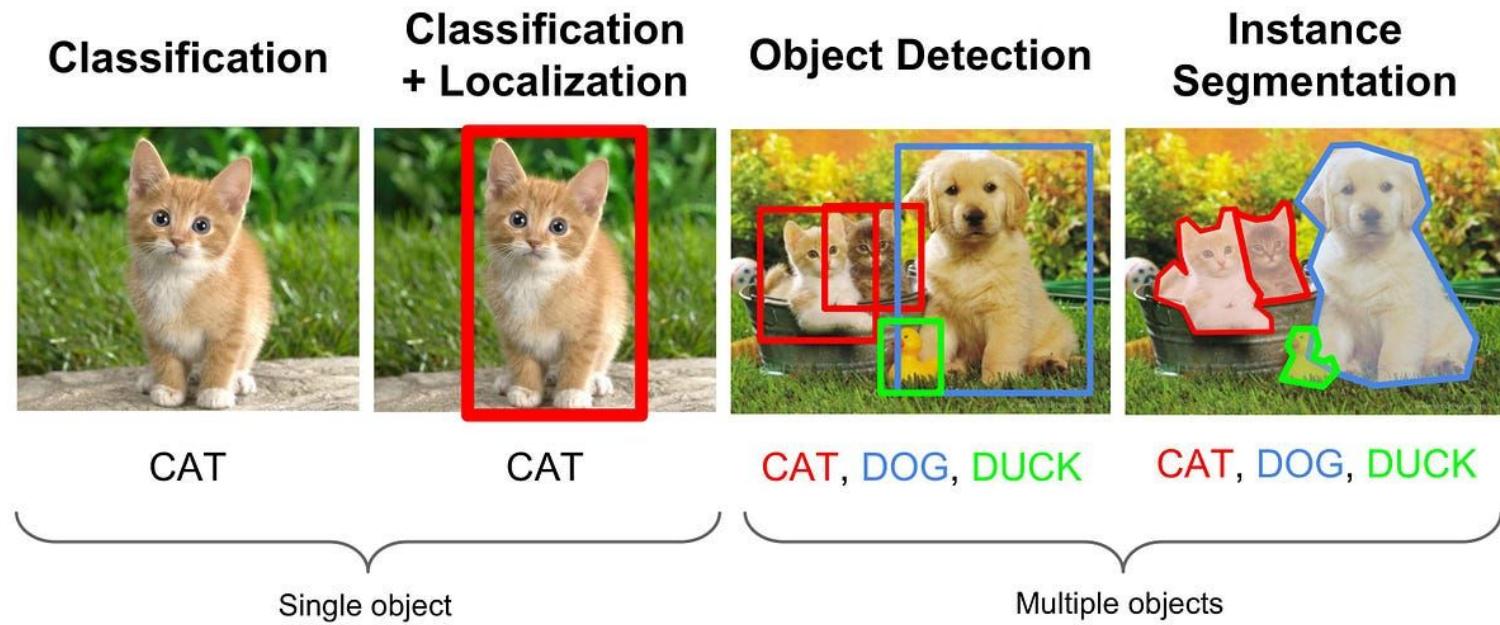
# Сегментация

• • •

# Локализация объектов

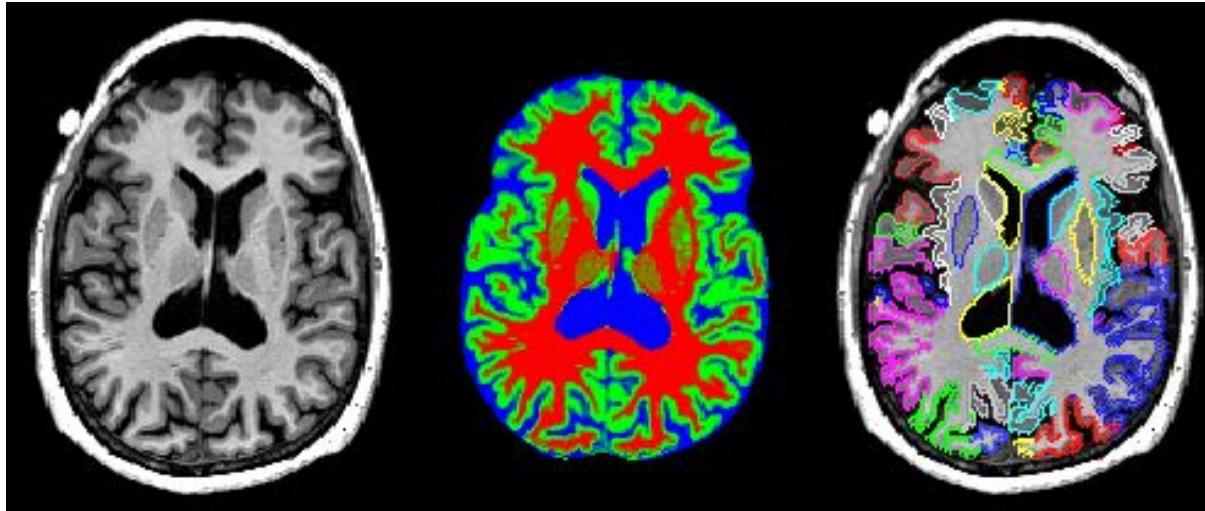
- На прошлых лекциях рассматривали задачу **детектирования объектов** (object detection)
  - Класс объектов
  - Положение объектов (через bounding boxes)
- Это - частный случай **задачи локализации**

# Локализация объектов



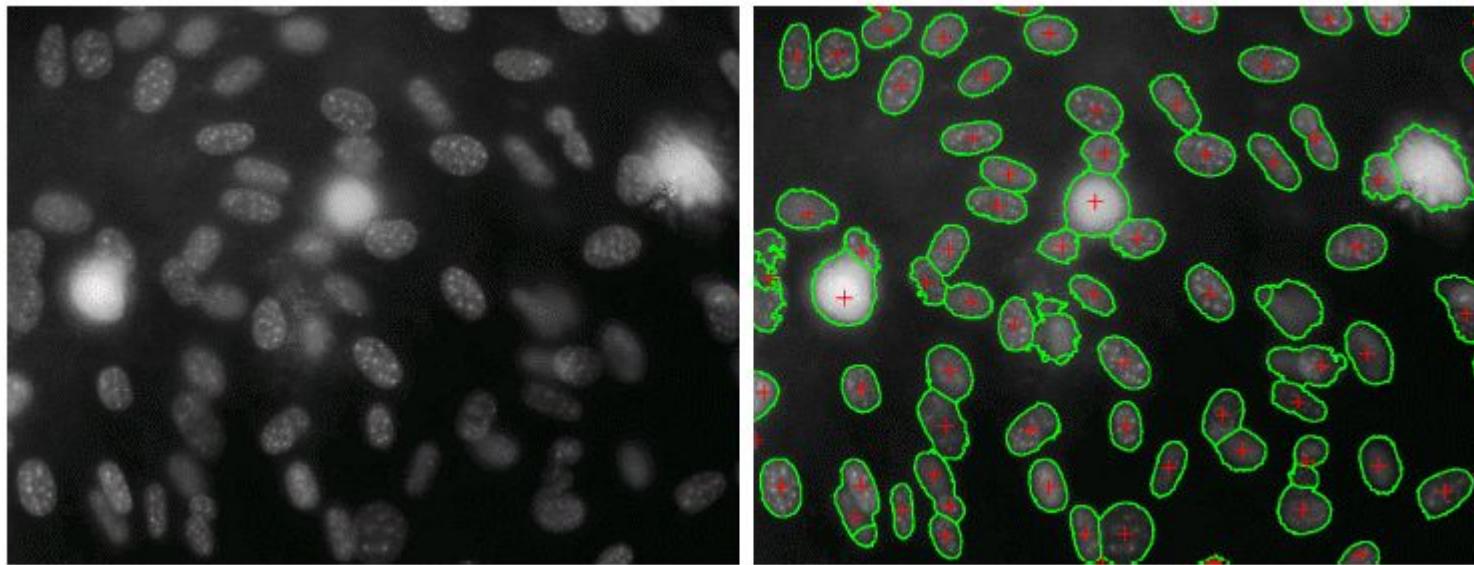
- Другой частный случай локализации – **сегментация**
- В сегментации положение объекта задается **попиксельной маской**

## Сегментация - примеры



Томографические снимки

# Сегментация - примеры



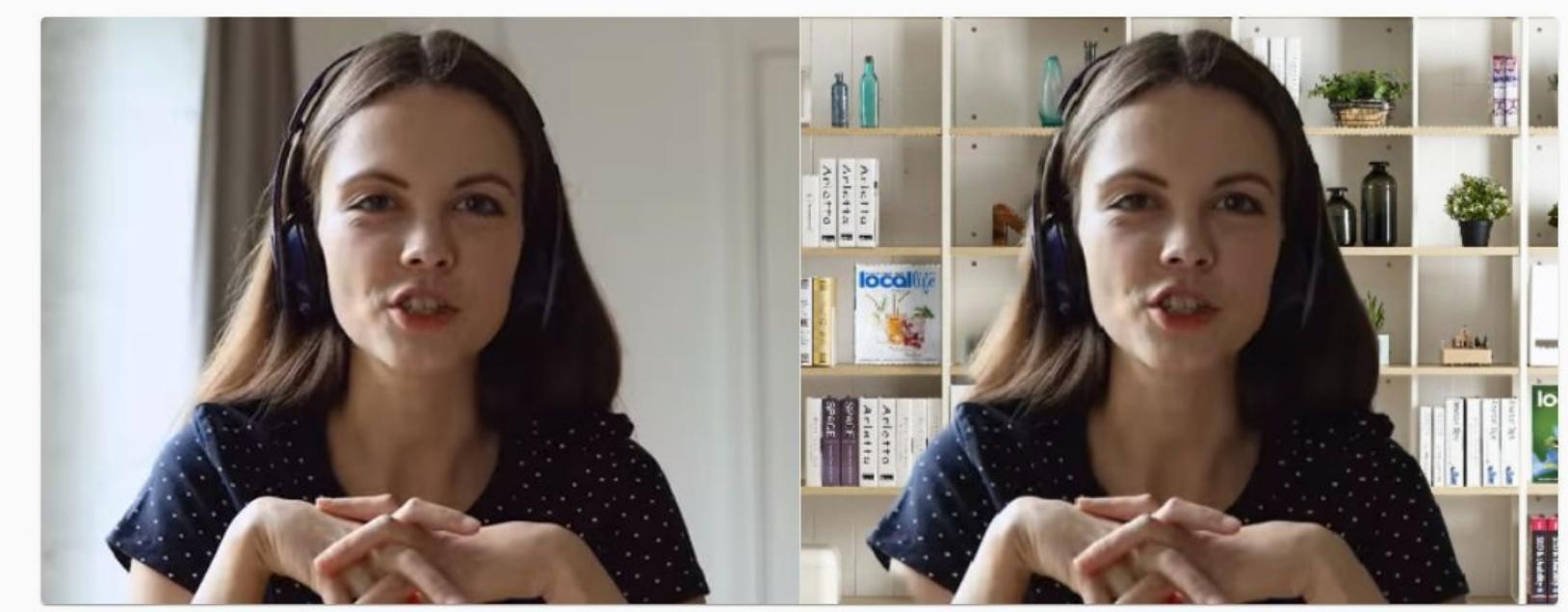
Снимки с микроскопа

# Сегментация - примеры



Спутниковые снимки / аэрофото

# Сегментация - примеры



Сегментация "фона"

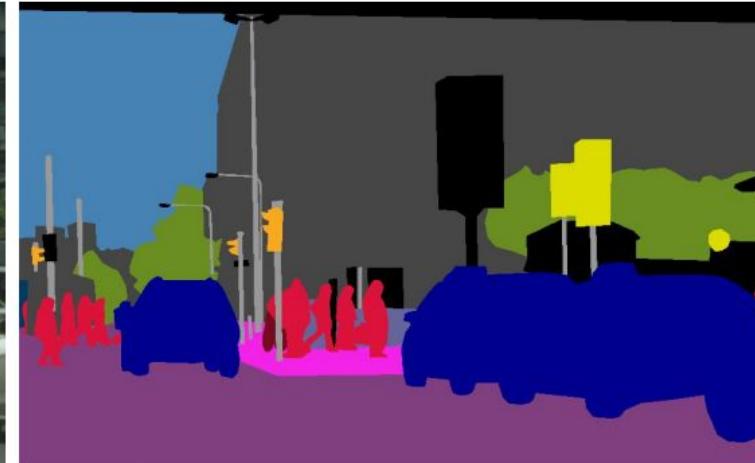
# Сегментация - виды

- Сегментация позволяет более точно **определять форму интересующих объектов**
- Есть несколько видов сегментации с **разной полнотой извлекаемой информации:**
  - Semantic Segmentation
  - Instance Segmentation
  - Panoptic Segmentation

# Сегментация - виды



(a) image



(b) semantic segmentation



(c) instance segmentation



(d) panoptic segmentation

# Сегментация - виды

- Семантическая (semantic) - каждому пикселю (кроме фона) ставится в соответствие номер класса
- Объектная (instance) - разделяются маски разных экземпляров одного класса
- Panoptic
  - для классов типа “небо”, “дорожное полотно” и т.д. (stuff) только класс,
  - для остальных (things) - отдельная маска

# Семантическая сегментация

...

# Семантическая сегментация

- Семантическая сегментация  $\sim$  пиксельная классификация
  - На входе в модель - изображение (RGB, RGBD, ...), облако точек, ...
  - На выходе - набор масок разных классов
- Рассмотрим бинарный случай - классы “фон” и “объект”

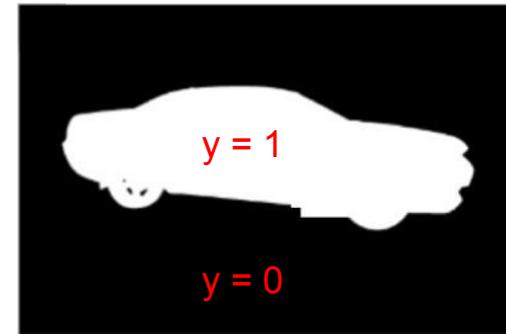
# Семантическая сегментация

RGB



# Семантическая сегментация

RGB



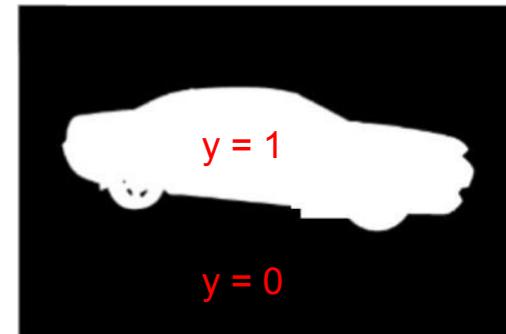
Ground-truth  
маска

# Семантическая сегментация

RGB



Модель



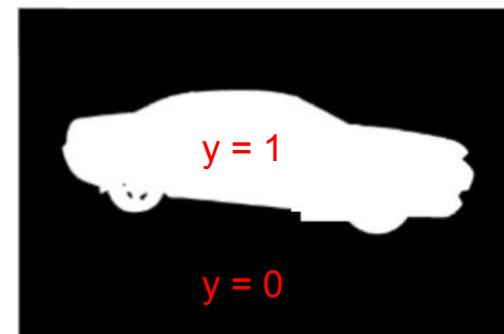
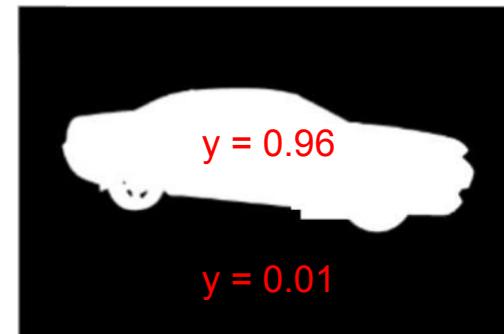
Ground-truth  
маска

# Семантическая сегментация

RGB

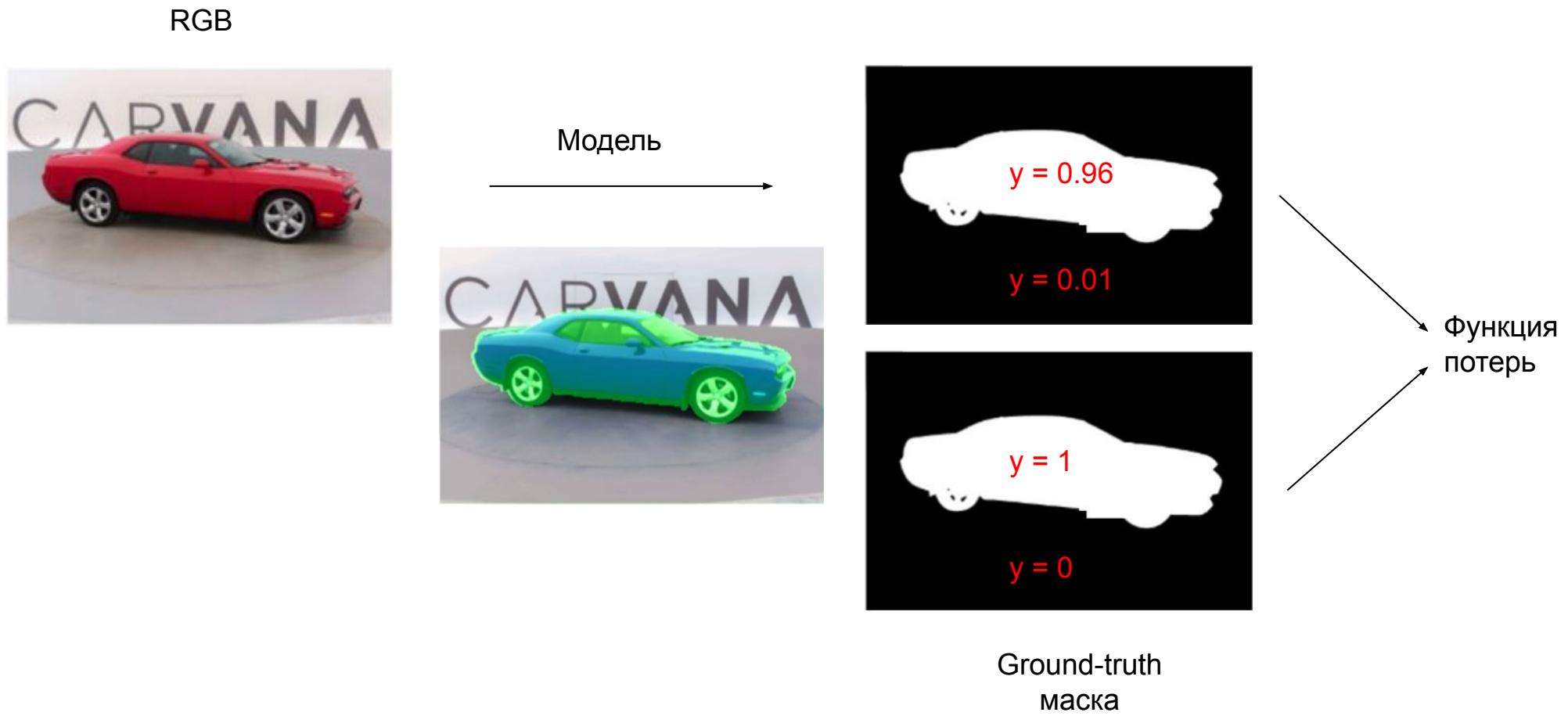


Модель



Ground-truth  
маска

# Семантическая сегментация



# Семантическая сегментация - функция потерь

- Что использовать в качестве лосса?
  - Попиксельная классификация -> Cross-Entropy?

$$BCE(y, \bar{y}) = -y \log (\bar{y}) - (1 - y) \log (1 - \bar{y})$$

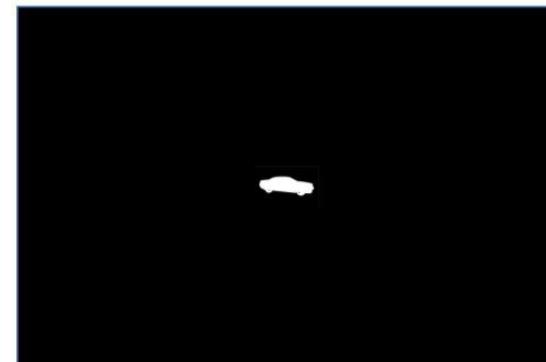
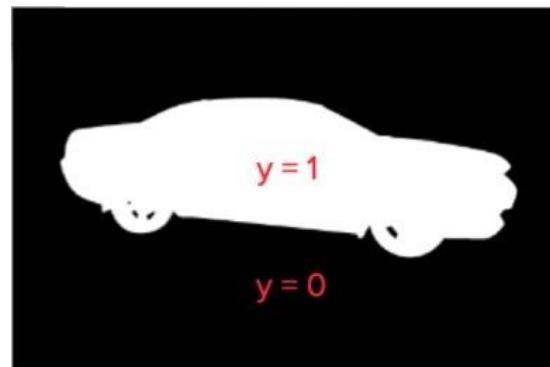
- Считаем в каждом пикселе, усредняем по площади

# Семантическая сегментация - функция потерь

- Что использовать в качестве лосса?
  - Попиксельная классификация -> Cross-Entropy?

$$BCE(y, \bar{y}) = -y \log (\bar{y}) - (1 - y) \log (1 - \bar{y})$$

- Считаем в каждом пикселе, усредняем по площади

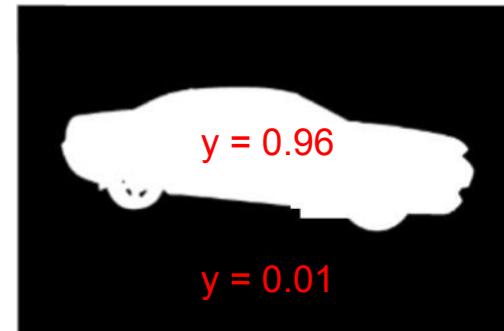


# Семантическая сегментация - дисбаланс классов

- Аналогично задаче детектирования, может быть сильный дисбаланс классов
- Возможные решения:
  - Веса для балансировки вклада разных пикселей
  - Focal Loss
  - Другие функции (см. дальше)

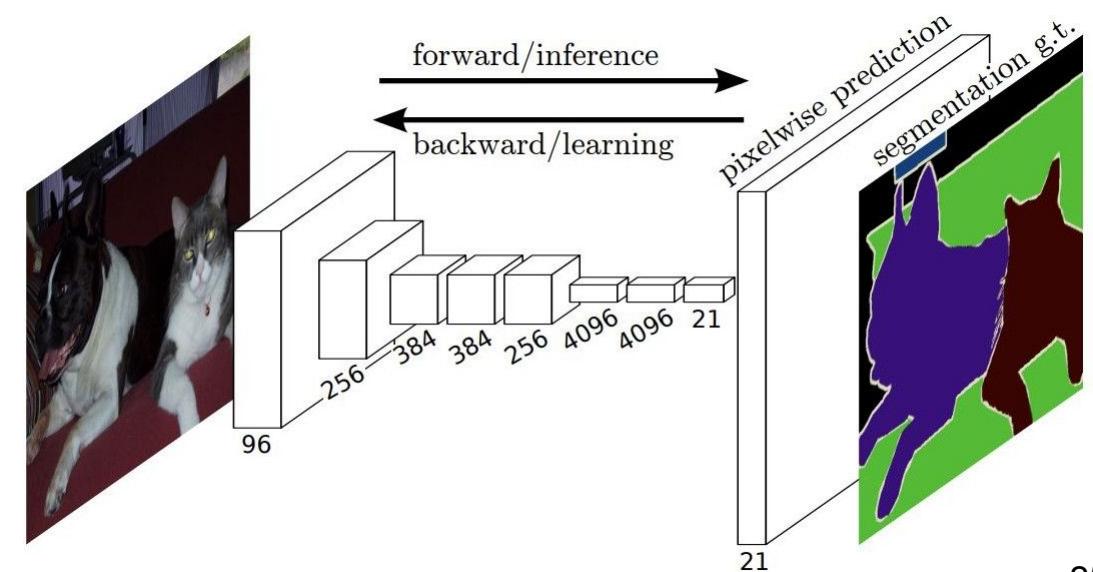
# Семантическая сегментация - архитектура

- Какой может быть архитектура модели?
- "Интерфейс" модели:
  - На входе - изображение,  $H \times W \times C$
  - На выходе - К масок,  $H \times W \times K$



# Fully-Convolutional Network (FCN) (2014)

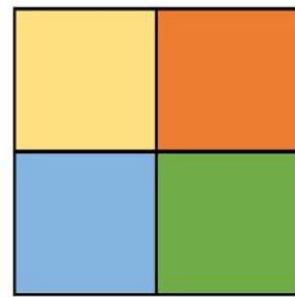
- Fully convolutional networks for Semantic Segmentation (2014)
  - Backbone (VGG)
  - Upsampling для приведения к исходному размеру
- Проблемы:
  - Узкое бутылочное горлышко
  - Резкое увеличение H/W



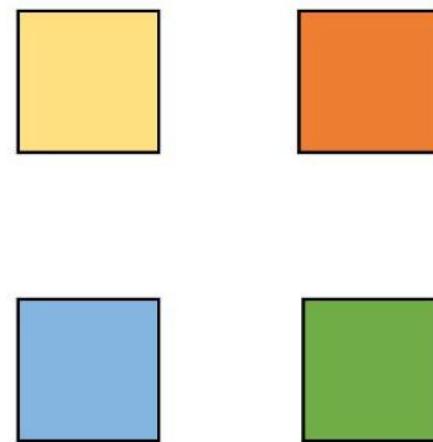
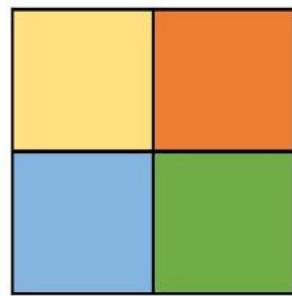
## Interlude

- Как увеличить размер ( $H, W$ ) карты активаций?
  - Интерполяция значений (Upsampling)
  - Транспонированная свертка (Transposed Conv)

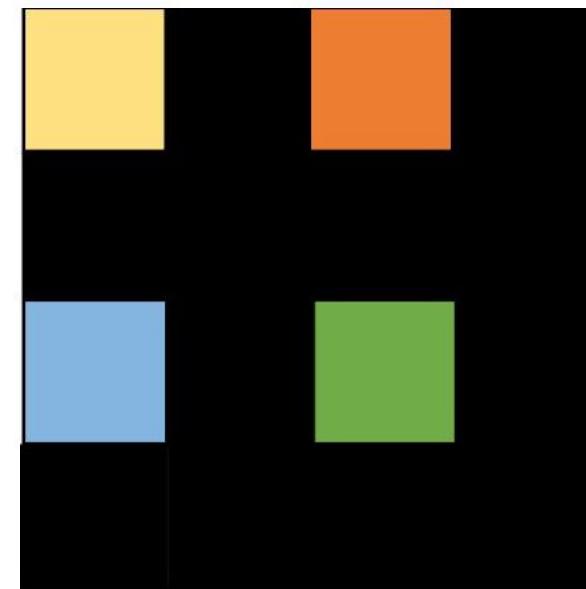
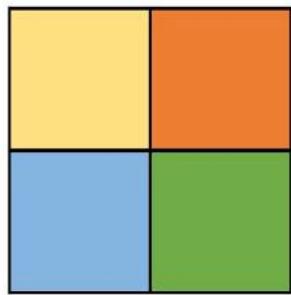
# Upsampling



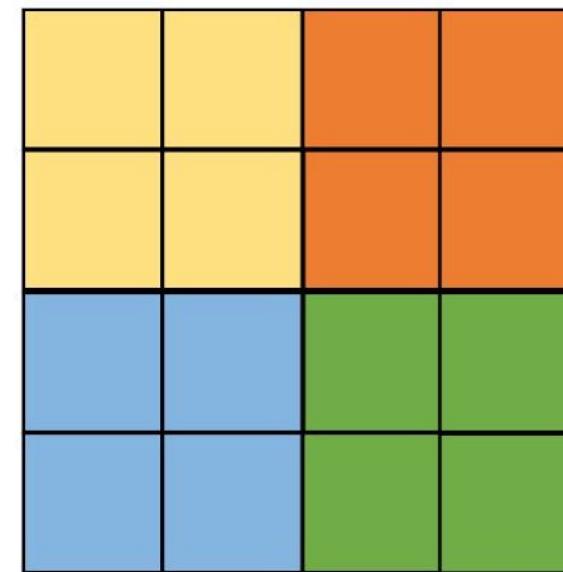
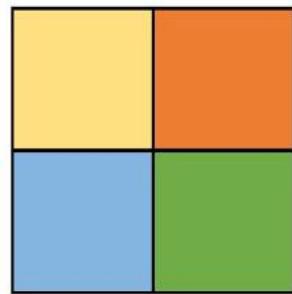
# Upsampling



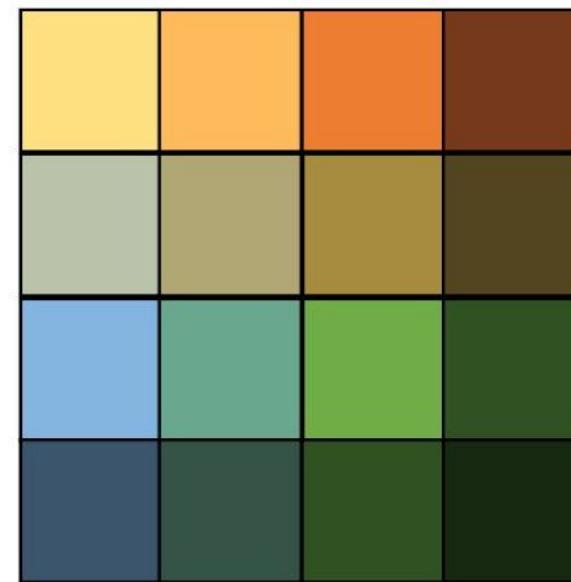
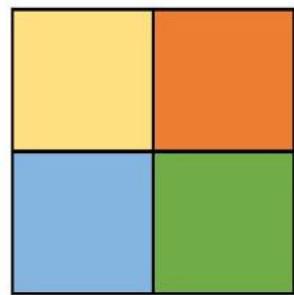
# Upsampling



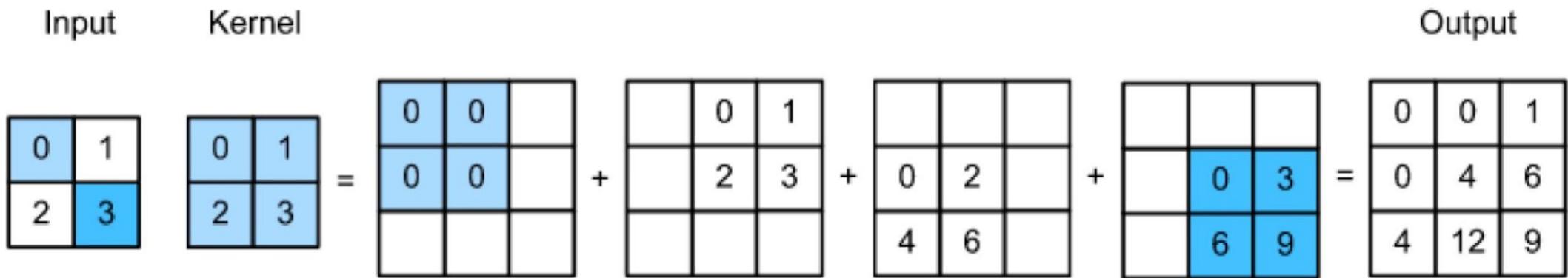
# Upsampling



# Upsampling



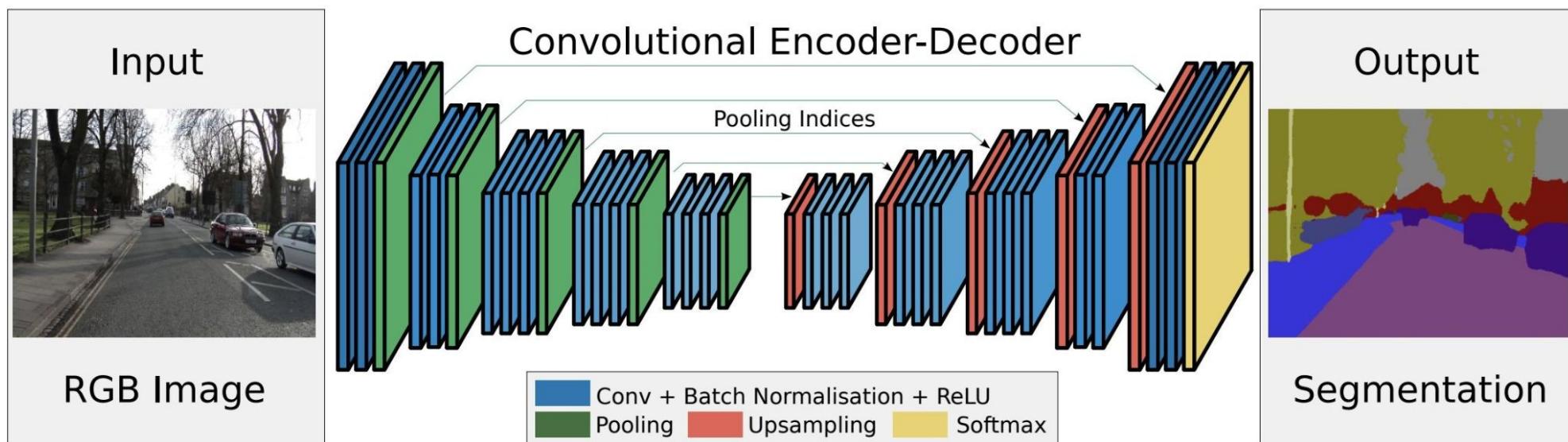
# Transposed Conv



- Вместо "сворачивания" с ядром (как в обычном ConvLayer) происходит "разворачивание" входного сигнала
- Значения входного сигнала выступают весами перед ядром транспонированной свертки

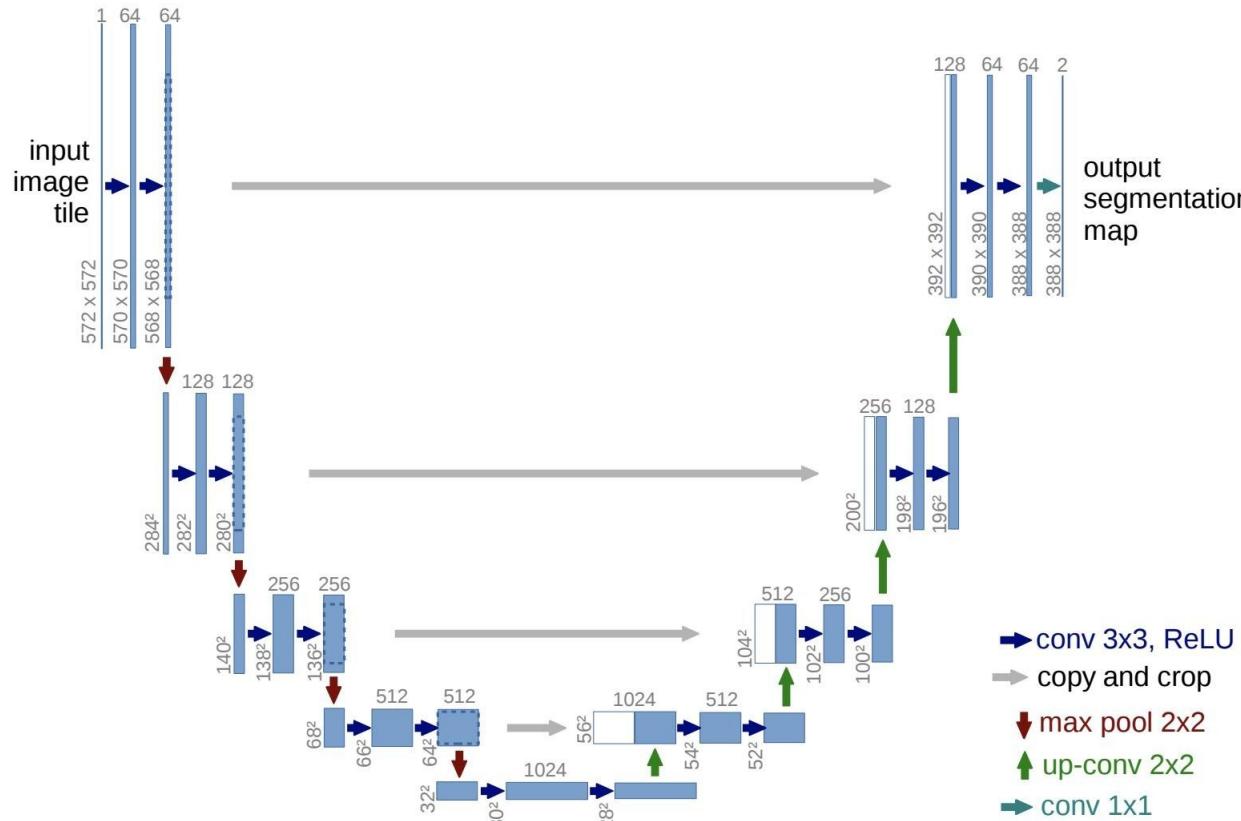
# SegNet (2015)

- SegNet: A Deep Convolutional Encoder-Decoder Architecture
  - Симметричная архитектура вида Encoder-Decoder
  - Постепенный Upsampling



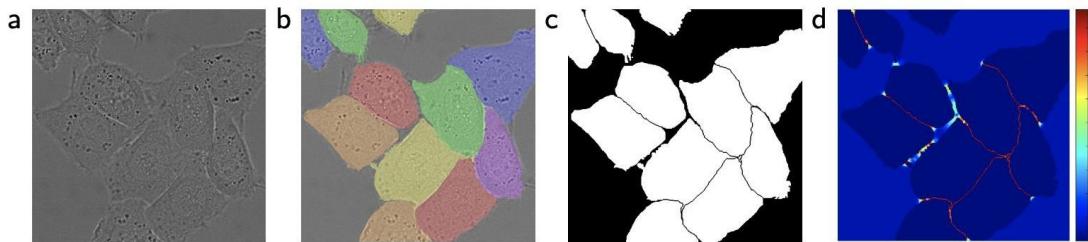
# UNet (2015)

- U-Net: CNNs for Biomedical Image Segmentation
  - Добавили горизонтальные связи к Encoder-Decoder

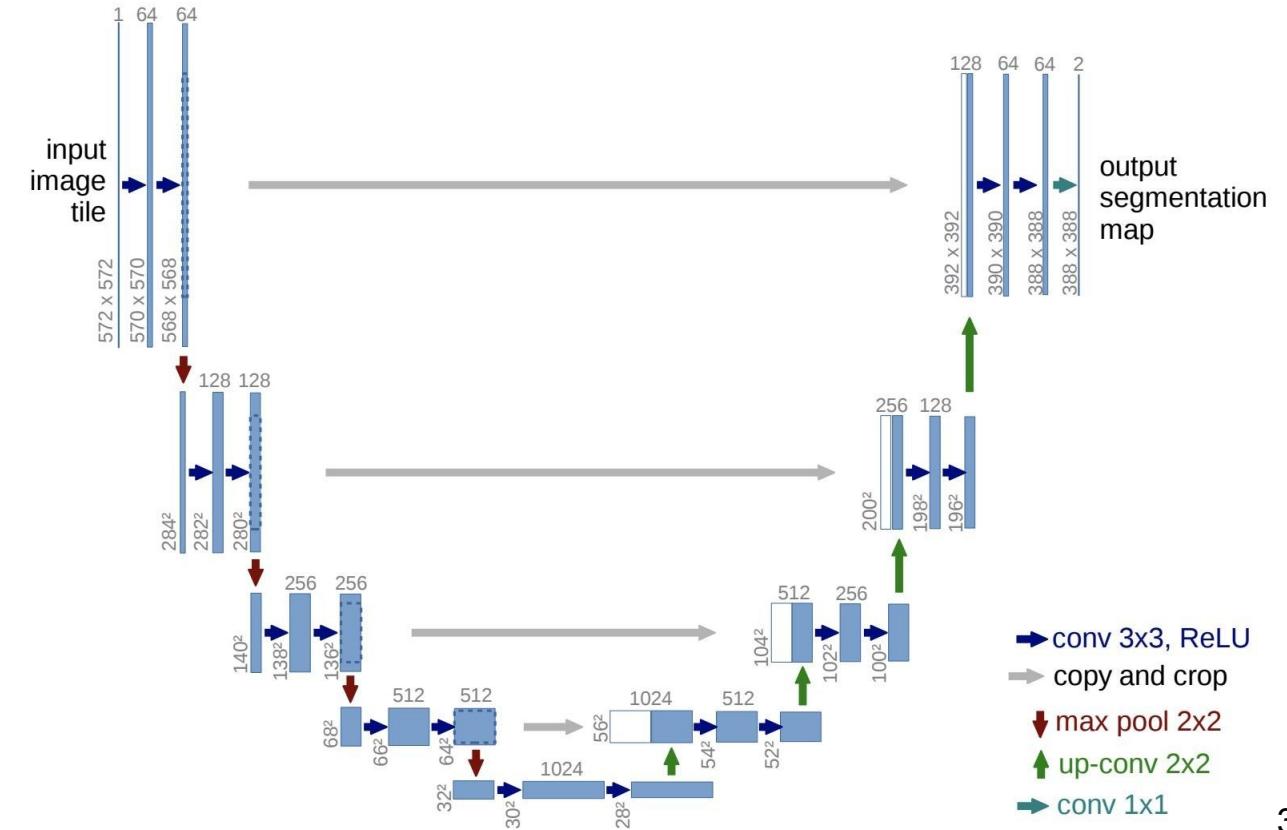


# UNet (2015)

- Сильное улучшение сегментации на границах объектов
- Всего лишь 30 изображений 512x512 для обучения!



**Fig. 3.** HeLa cells on glass recorded with DIC (differential interference contrast) microscopy. (a) raw image. (b) overlay with ground truth segmentation. Different colors indicate different instances of the HeLa cells. (c) generated segmentation mask (white: foreground, black: background). (d) map with a pixel-wise loss weight to force the network to learn the border pixels.

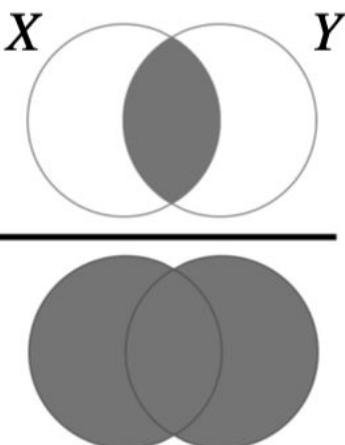


# UNet

- Из конкретной архитектуры для сегментации UNet давно превратился в "подход" для задач image-to-image:
  - Сегментация (subj)
  - Колоризация (предсказание цветных каналов для grayscale-входа)
  - InPainting ("закрашивание" пустот)
  - ...
- UNet-like сеть =
  - encoder (resnet, efficientnet, ...) +
  - decoder

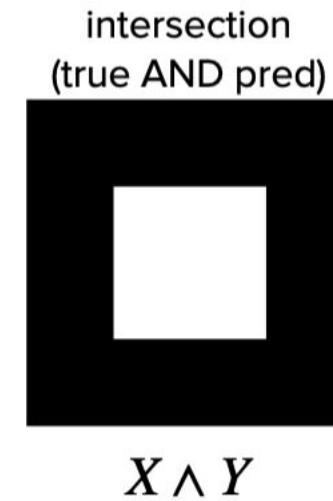
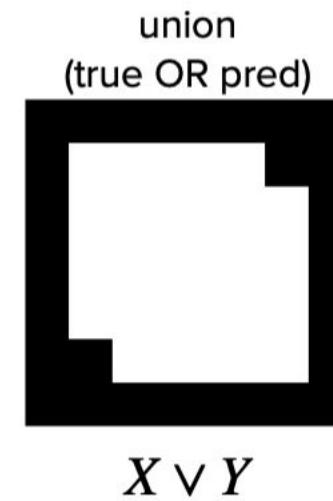
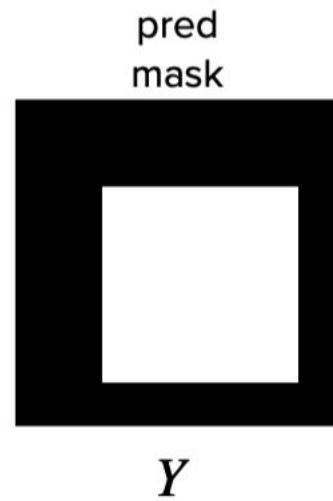
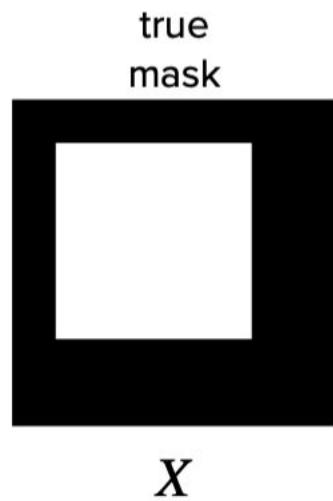
# Beyond BCE

- Применение кросс-энтропии может сломаться о дисбаланс классов (и не только в сегментации)
- Вспомним, что в детекторах объектов говорили про понятие **Intersection-over-Union (IoU)** (синоним - Jaccard Index):

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

$$Jaccard(X, Y) = \frac{|X \wedge Y|}{|X \vee Y|}$$

# Jaccard Index для сегментации

- По аналогии определим Jaccard Index для пары сегментационных масок:



$$Jaccard(X, Y) = \frac{|X \wedge Y|}{|X \vee Y|}$$

## Jaccard Index для сегментации

- Напрямую оптимизировать Jaccard Index нельзя (**почему?**)
- Но можно аппроксимировать его, например:

$$J_{seg} = \frac{\sum_{x,y} M_{gt}(x,y) * M_{pred}(x,y)}{\sum_{x,y} M_{gt}(x,y) + M_{pred}(x,y) - M_{gt}(x,y) * M_{pred}(x,y)}$$

- Получить из этого лосс можно, например, так:

$$Loss_J = 1 - \log(J_{seg})$$

- Часто комбинируют с BCE:

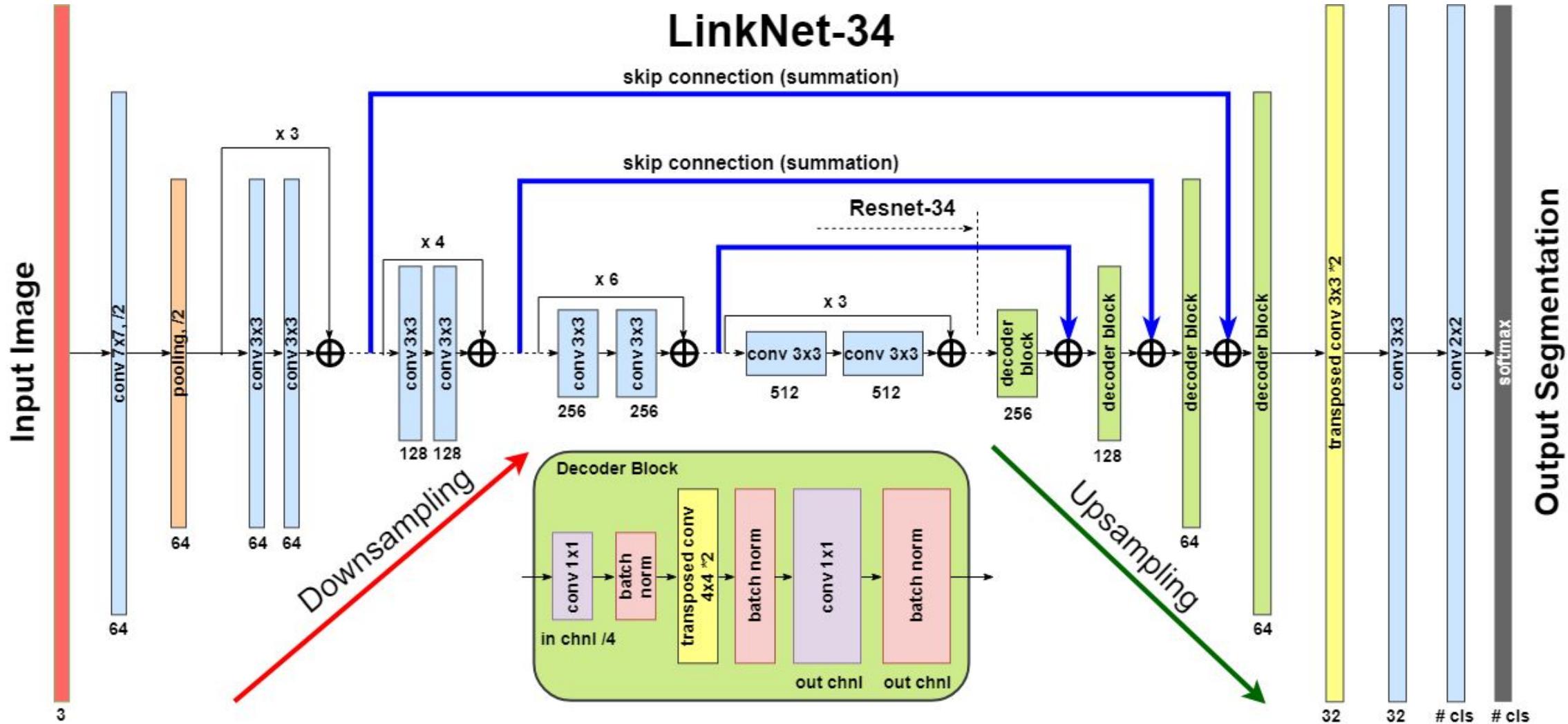
$$Loss = \alpha * Loss_{BCE} + (1 - \alpha) * Loss_J$$

# Segmentation Losses

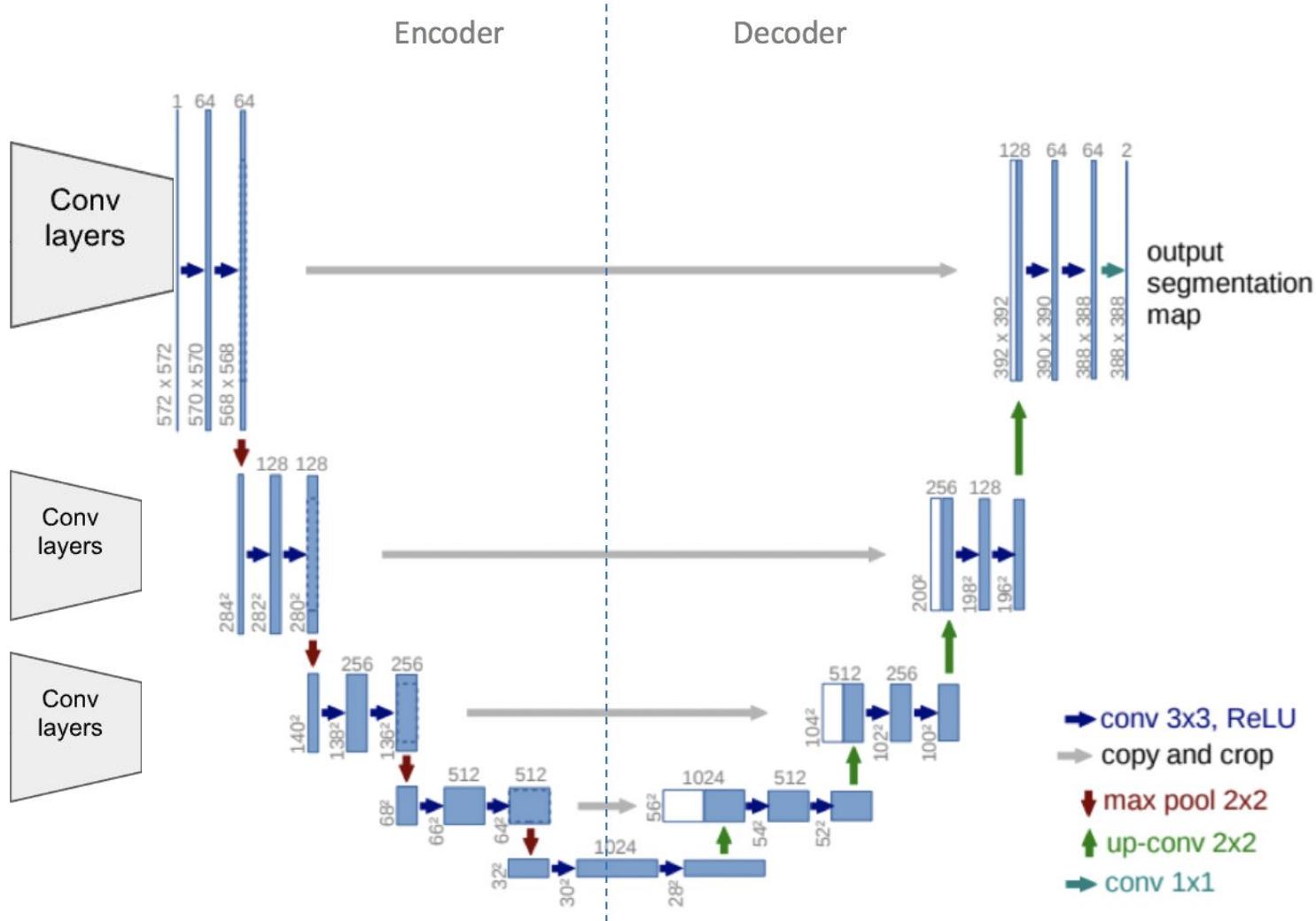
- Jaccard Loss vs Dice Loss - постоянная [путаница](#)
- [A survey of loss functions for semantic segmentation \(2020\)](#)

# Ternaus Net

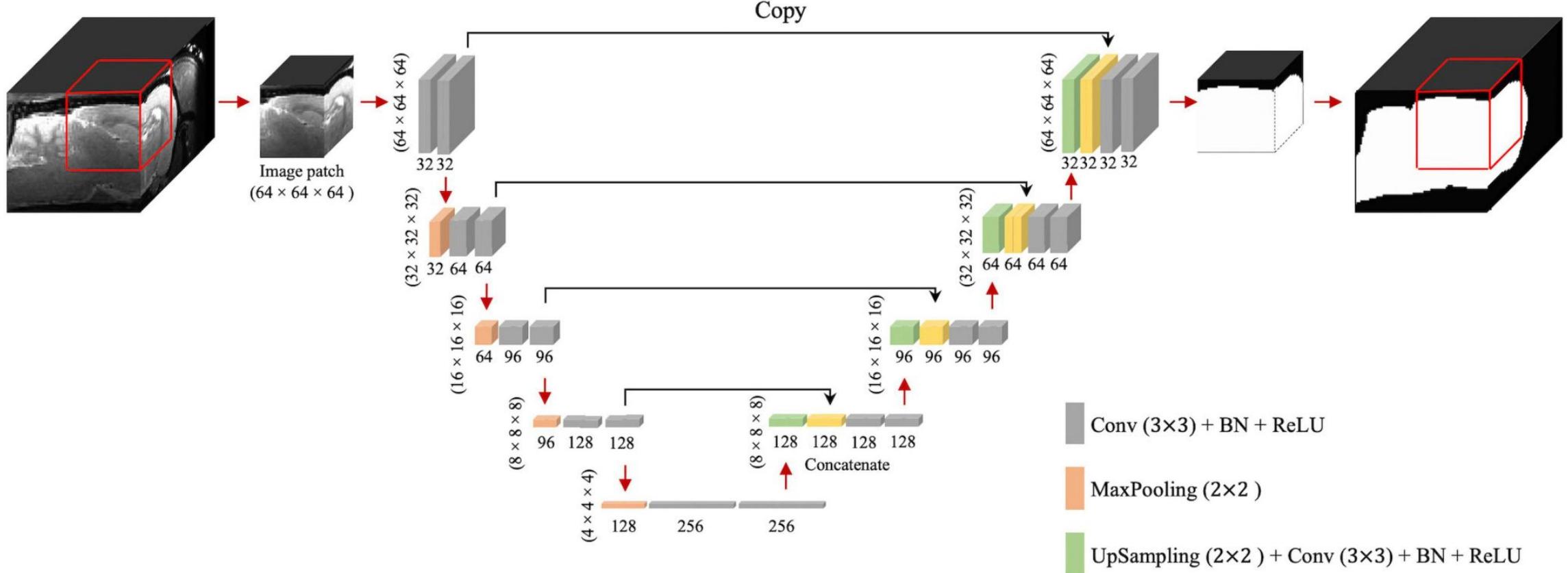
## LinkNet-34



# Multi-Input U-Net

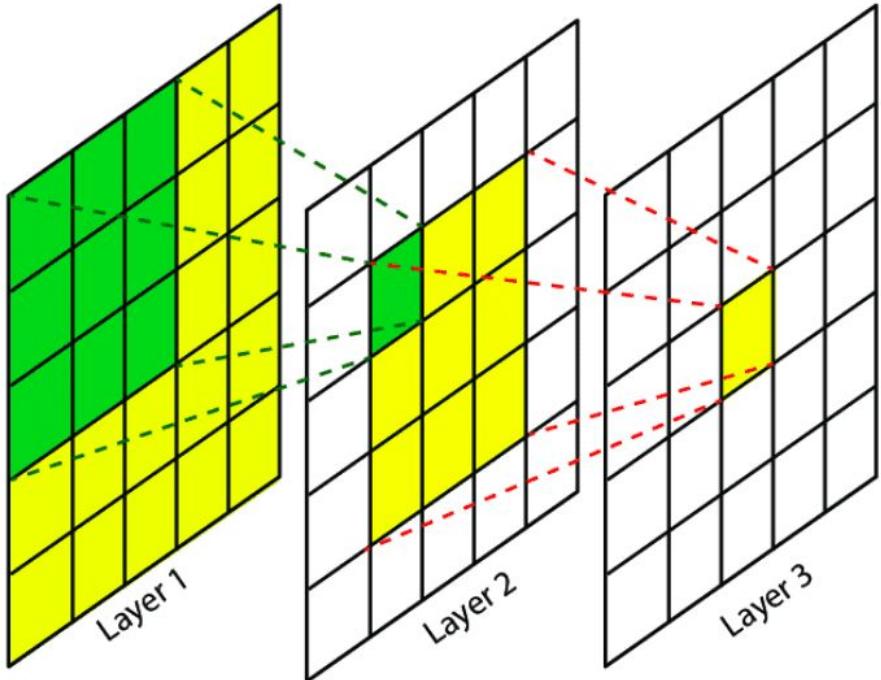


# 3D U-Net

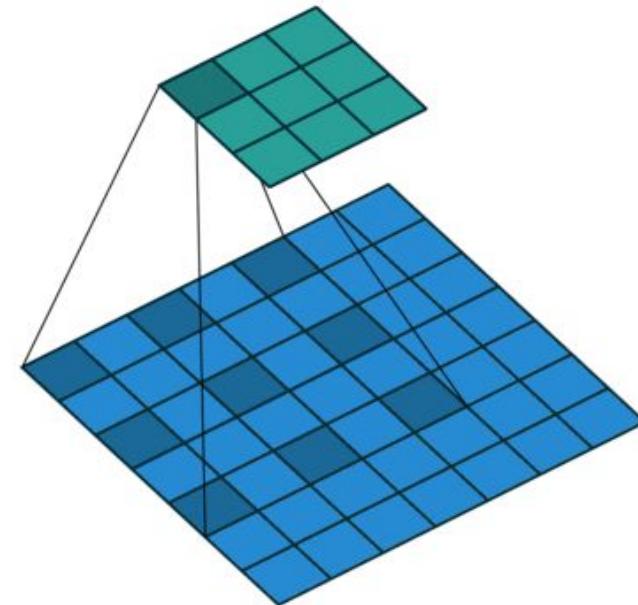


# Receptive field

Conv(3) -> Conv(3)



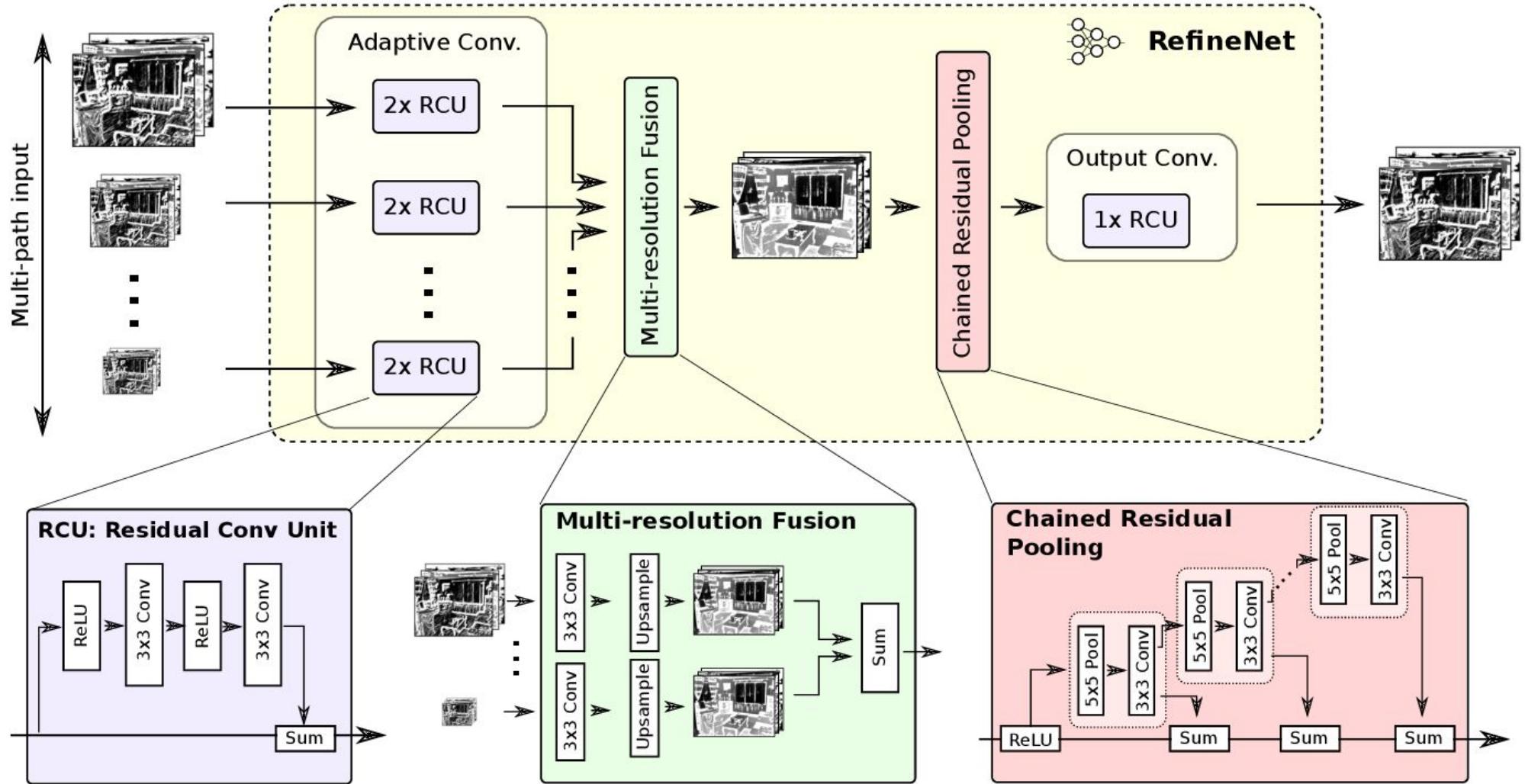
Conv(3, dilation=2)



Задача: определить размер receptive field для

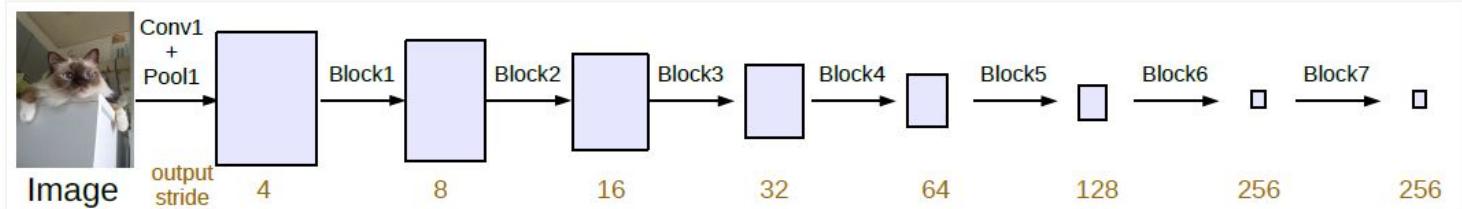
- 1) Conv(3) -> Conv(3, stride=2) -> Conv(3)
- 2) Conv(3) -> Conv(3, dilation=2)

# RefineNet

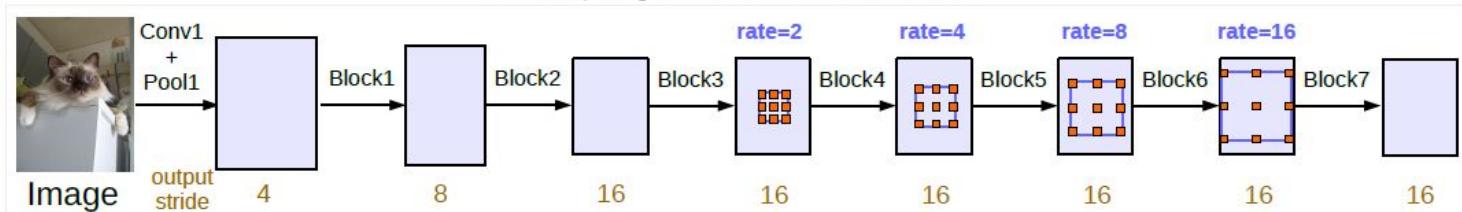


U-Net, но с очень специфической архитектурой кодировщика

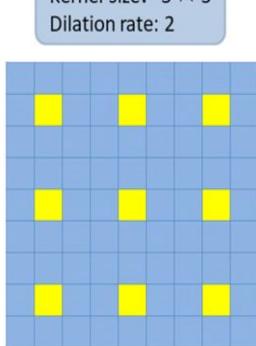
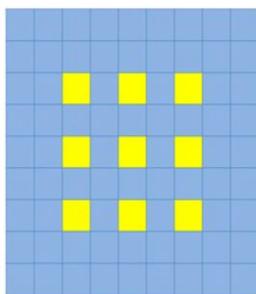
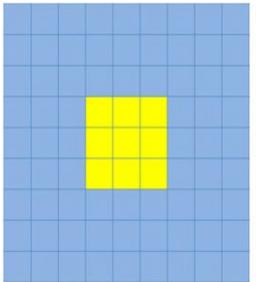
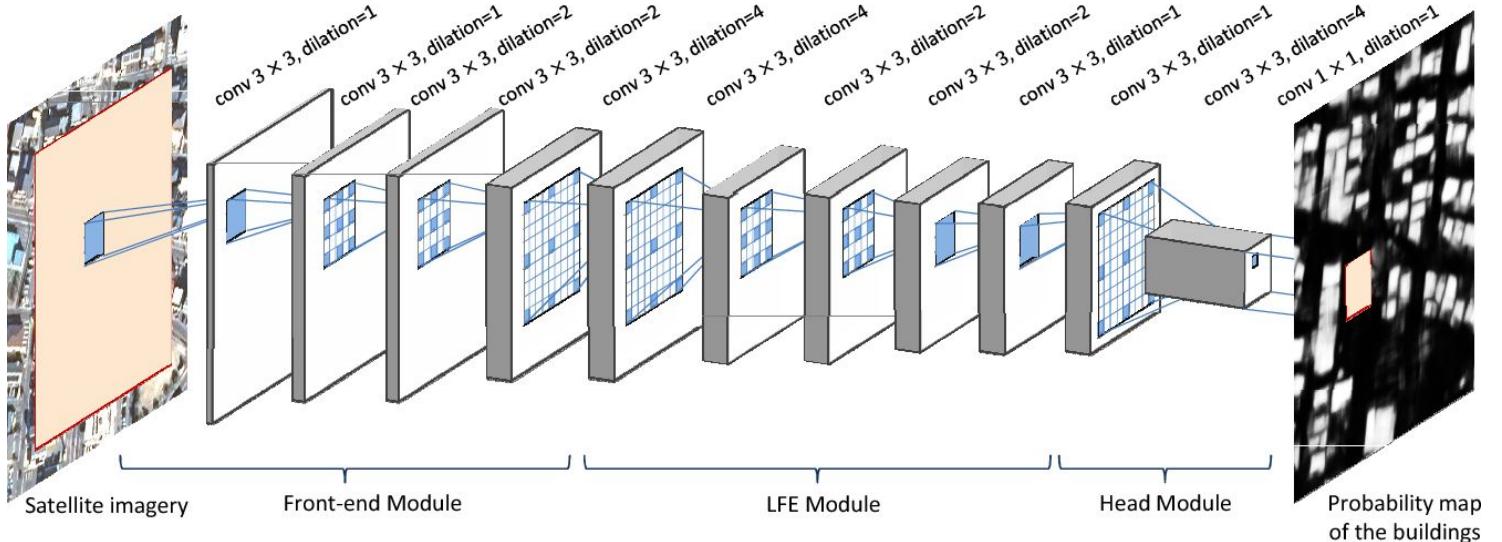
# Dilated Convolutions



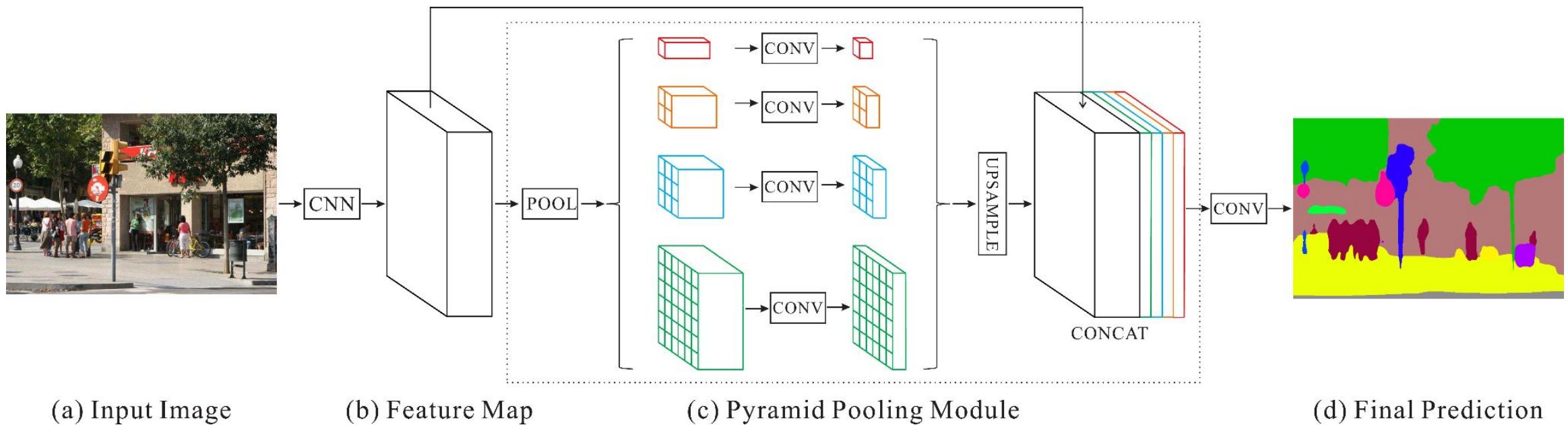
(a) Going deeper without atrous convolution.



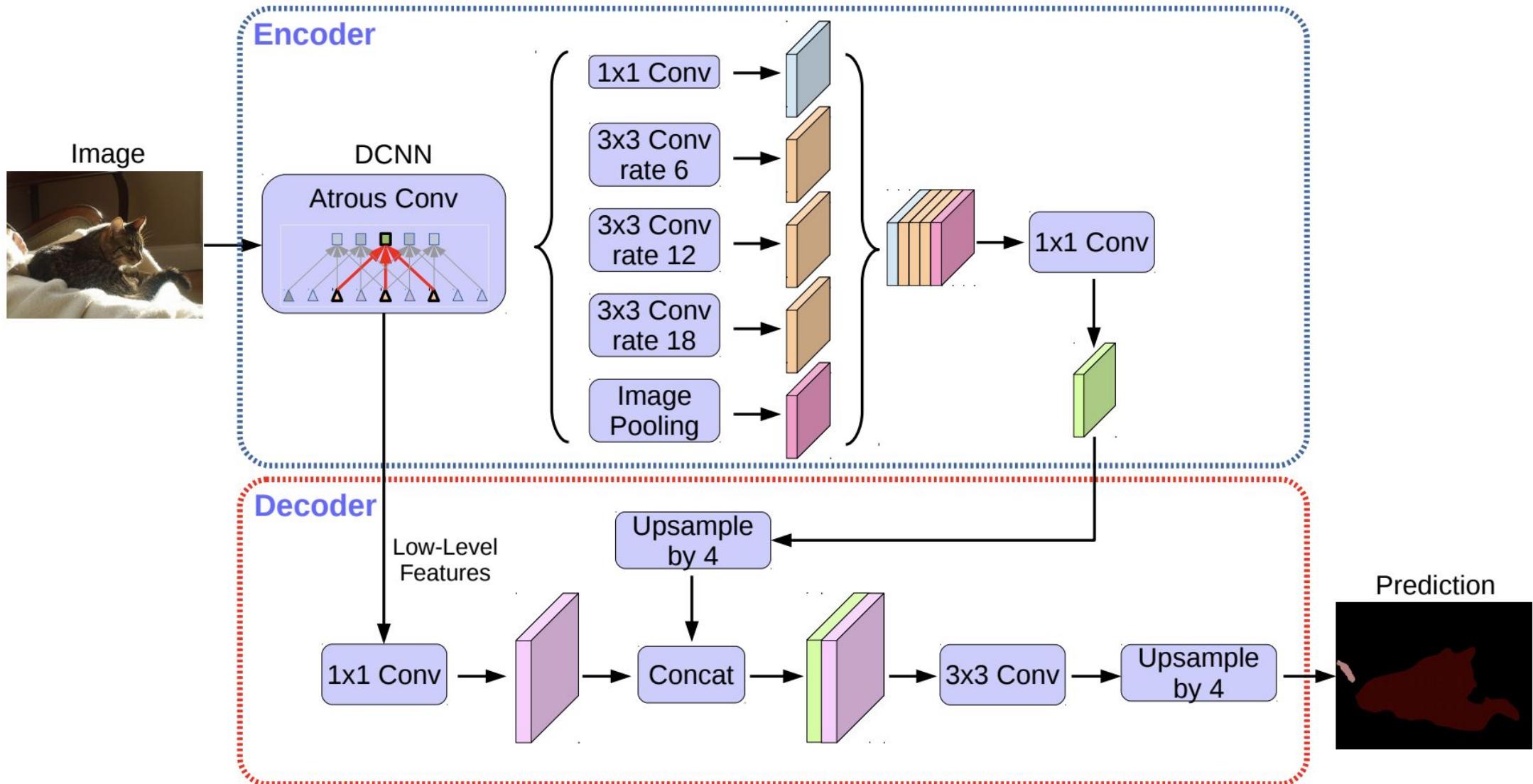
(b) Going deeper with atrous convolution. Atrous convolution with  $rate > 1$  is applied after block3 when  $output\_stride = 16$ .



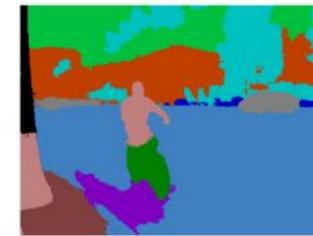
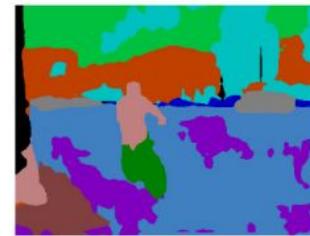
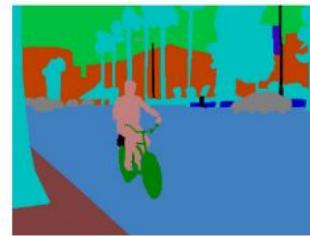
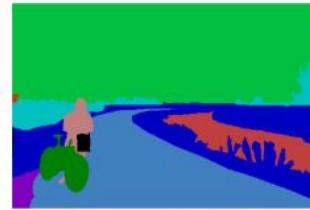
# PSP Net (2016)



# DeepLab V3+



# DeepLab postprocessing procedure



(a) Image

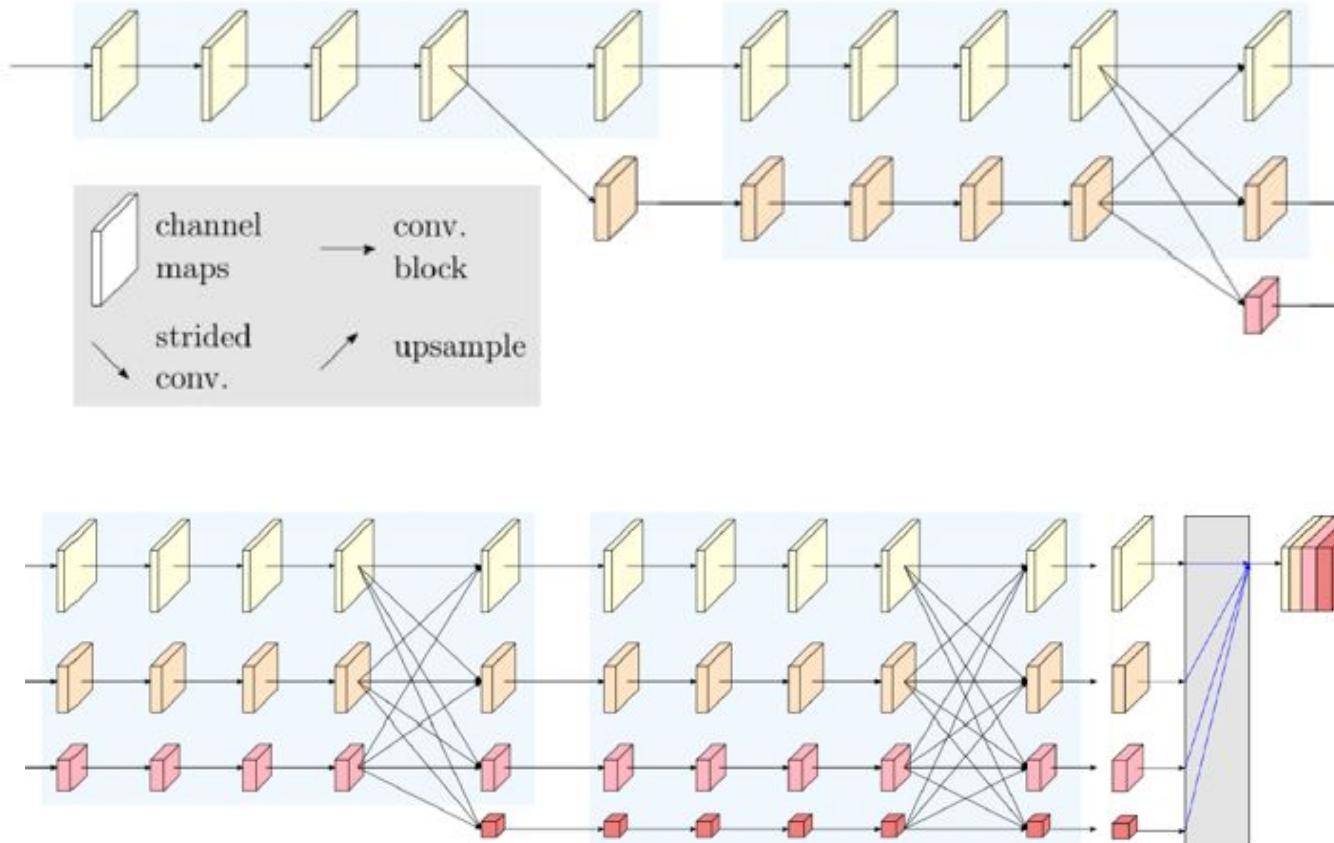
(b) G.T.

(c) Before CRF

(d) After CRF

Conditional Random Field (CRF)

# HRNet

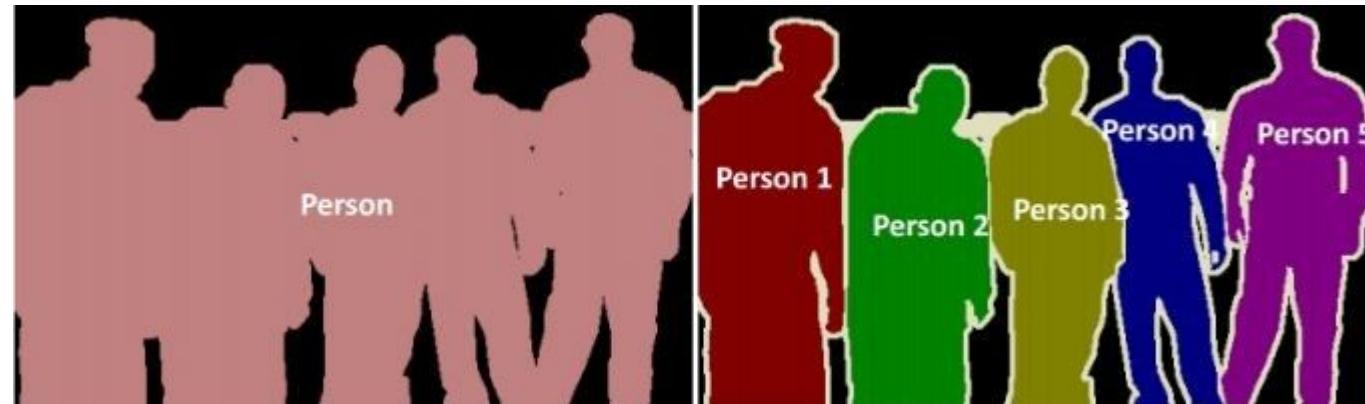


# Объектная сегментация

...

# Объектная сегментация

- Если объекты не “касаются” и не перекрывают друг друга, то разделить их маски не составит труда и после semantic segmentation
- ... Но так бывает не всегда

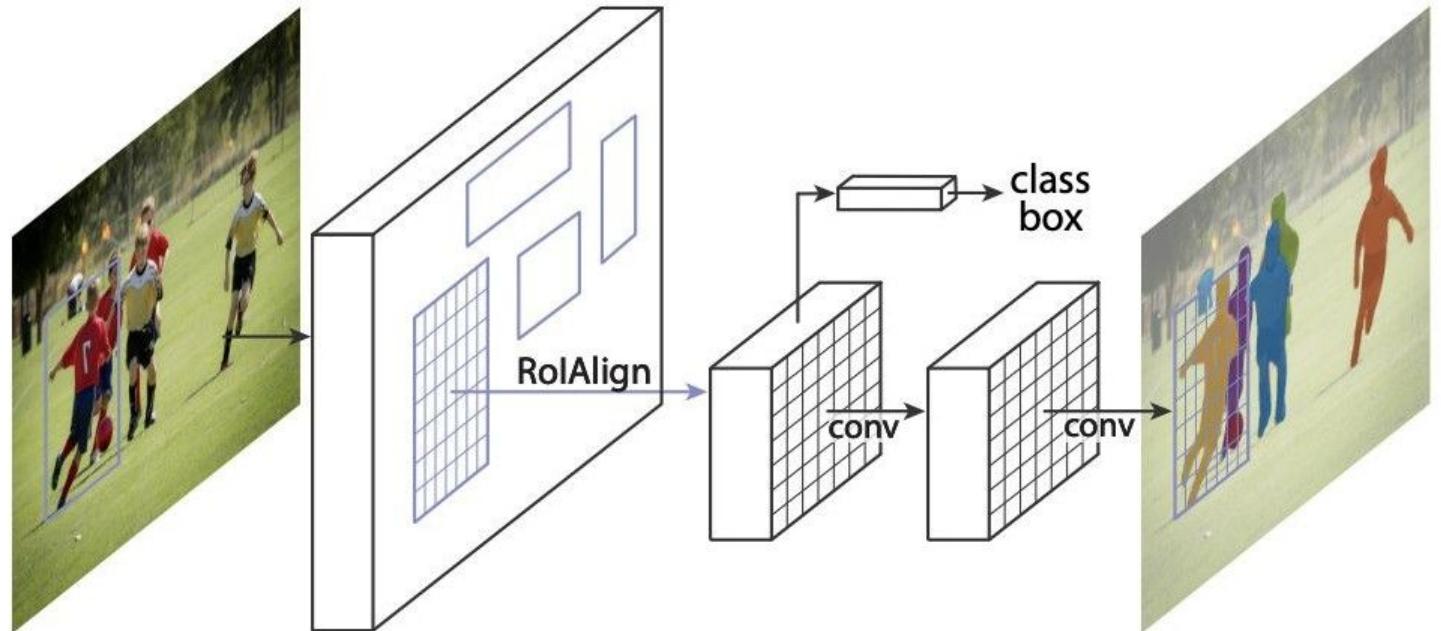


# Объектная сегментация

- Идея: встроить в детектор объектов семантическую сегментацию для бокса вокруг каждого объекта

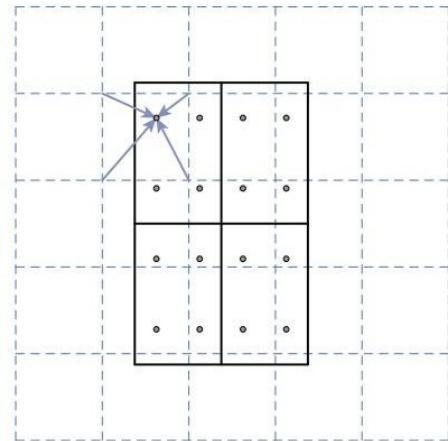
# Mask R-CNN (2017)

- Mask R-CNN (2017)
- В основе - Faster R-CNN
- Дополнительная ветвь для предсказания бинарной маски во всех proposals
  - Маски не зависят от классов (т.е. сегментация на 1 класс)
- Вместо RoIPool – RoIAvg



# Mask R-CNN (2017) - RoIAlign

- В Mask R-CNN используется операция RoIAlign: вместо грубого округления границ и пулинга значений используется интерполяция значений по сетке



**Figure 3. RoIAlign:** The dashed grid represents a feature map, the solid lines an ROI (with  $2 \times 2$  bins in this example), and the dots the 4 sampling points in each bin. RoIAlign computes the value of each sampling point by bilinear interpolation from the nearby grid points on the feature map. No quantization is performed on any coordinates involved in the ROI, its bins, or the sampling points.

# Mask R-CNN - примеры

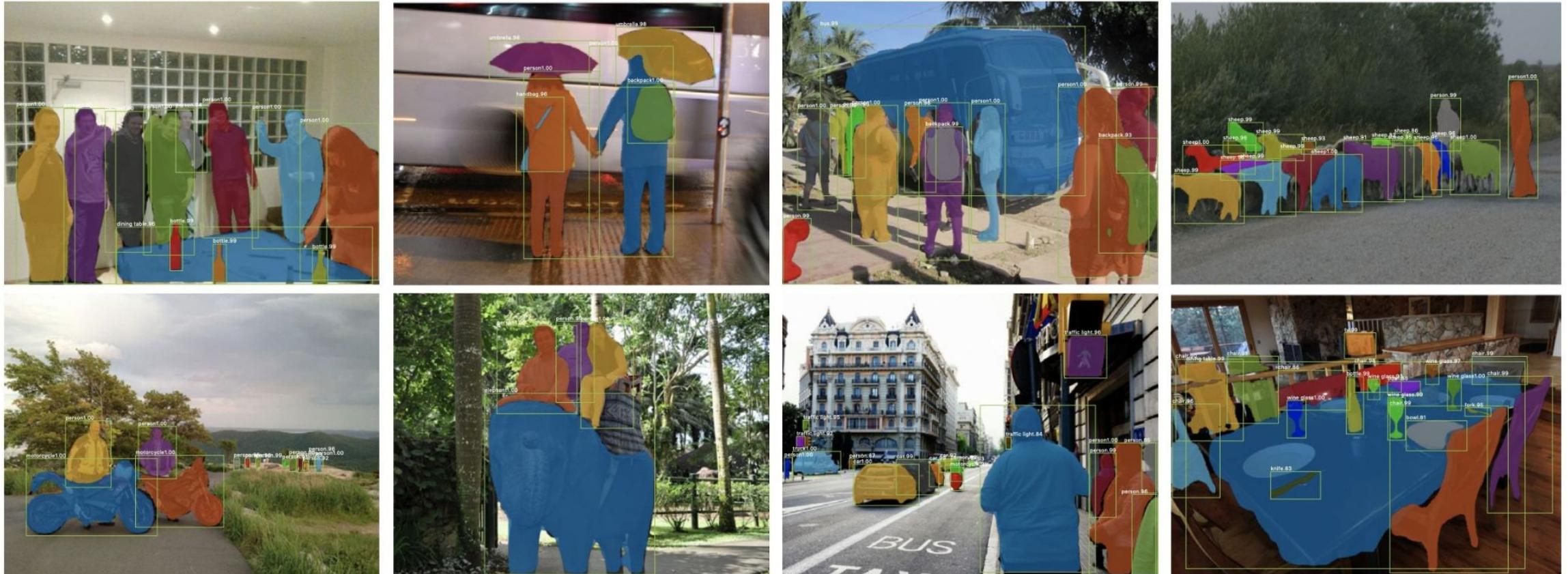


Figure 2. **Mask R-CNN** results on the COCO test set. These results are based on ResNet-101 [19], achieving a *mask AP* of 35.7 and running at 5 fps. Masks are shown in color, and bounding box, category, and confidences are also shown.

# Mask R-CNN

- Есть в PyTorch

```
from torchvision.models.detection.mask_rcnn import MaskRCNNPredictor
```

# Вопросы?



Спасибо  
за внимание!