

Глубинное обучение для компьютерного зрения

Куликов В.А.

(в соавторстве с Лемпицким В.С.)

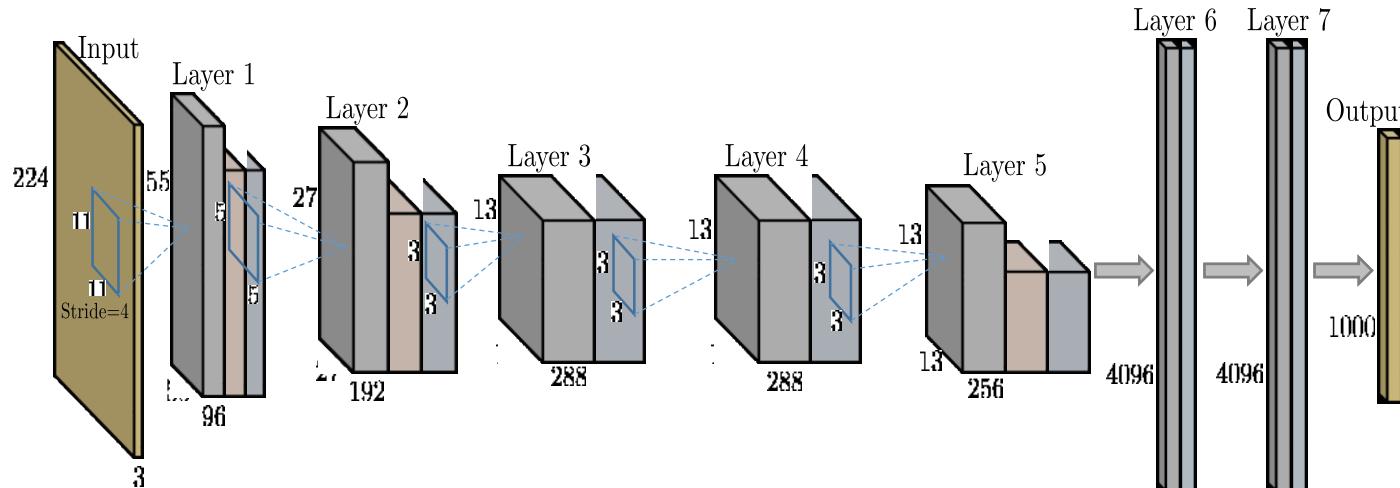


v.kulikov@skoltech.ru

Классификация изображений



Что на изображении?



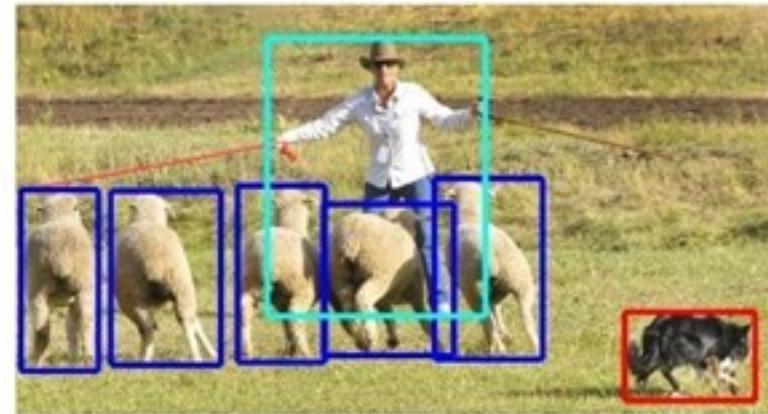
Кошка

Собака

Задачи компьютерного зрения



Классификация



Обнаружение объектов



Семантическая сегментация



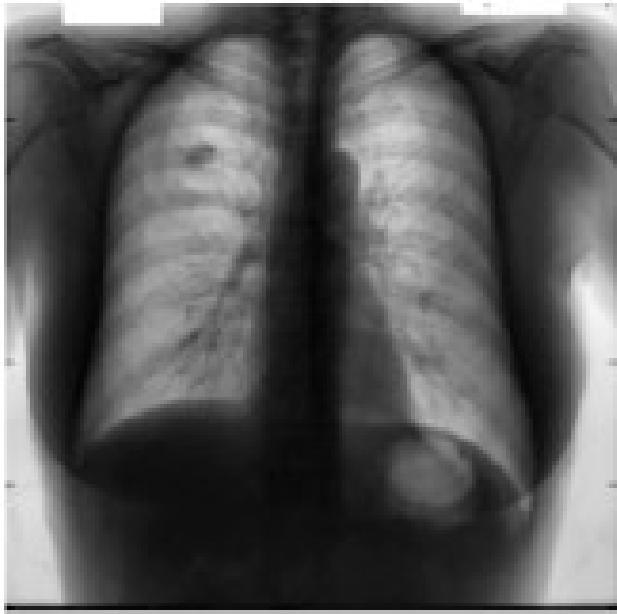
Сегментация объектов

[Lin et al. 2015]

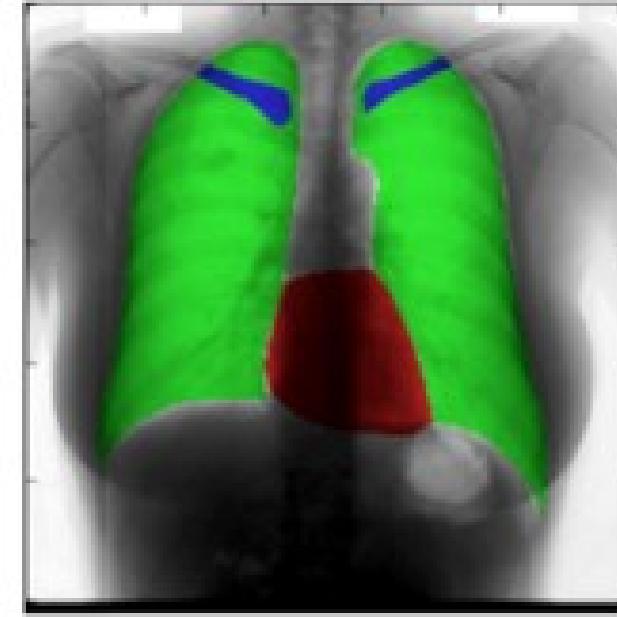
Семантическая сегментация для автопилота



Применение семантической сегментации в биомедицине



Input Image



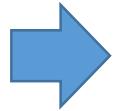
Segmented Image

Справа результат сегментации (зеленым отмечены легкие, красным сердце, синим – ключицы)

Постановка задачи семантической сегментации



Изображение $W \times H \times 3$



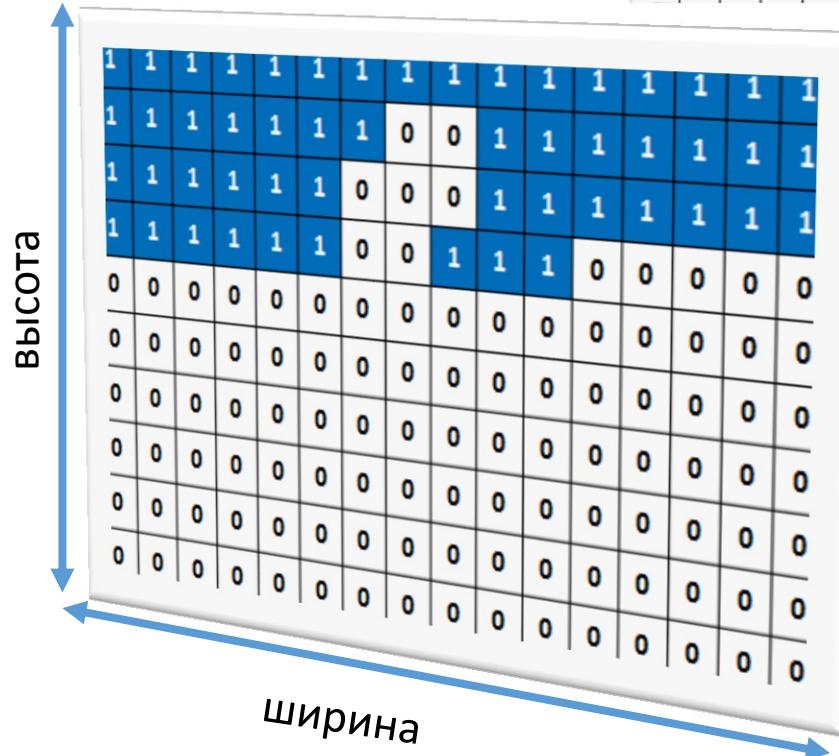
1 – небо, 2 – человек, 3 – земля, 4 – мотоцикл

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 4 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 4 | 4 | 4 | 4 | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 2 | 2 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 2 | 4 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |

Индексы семантических классов $W \times H \times 1$

Представление для машинного обучения

Трехмерный тензор $W \times H \times Nc$, где Nc – количество классов

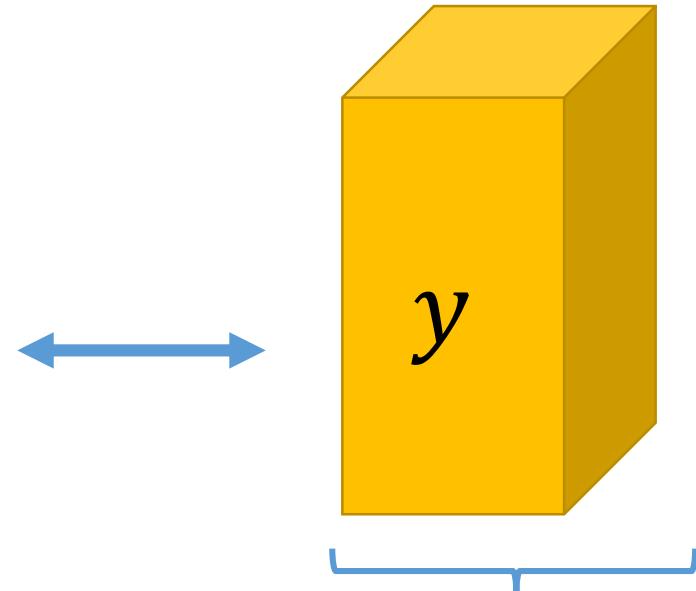
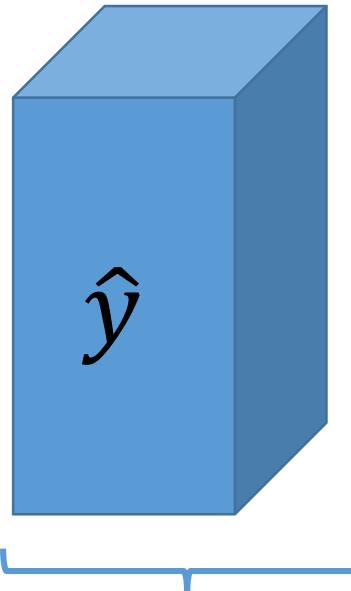


Для каждого пикселя one-hot вектор

Обучение/Функции потерь



$$\hat{y} = \Psi_W(x)$$



Кросс Энтропия

$$-\frac{1}{W \times H} \sum_{i,j} \sum_c y_{c,i,j} \log(\hat{y}_{c,i,j})$$

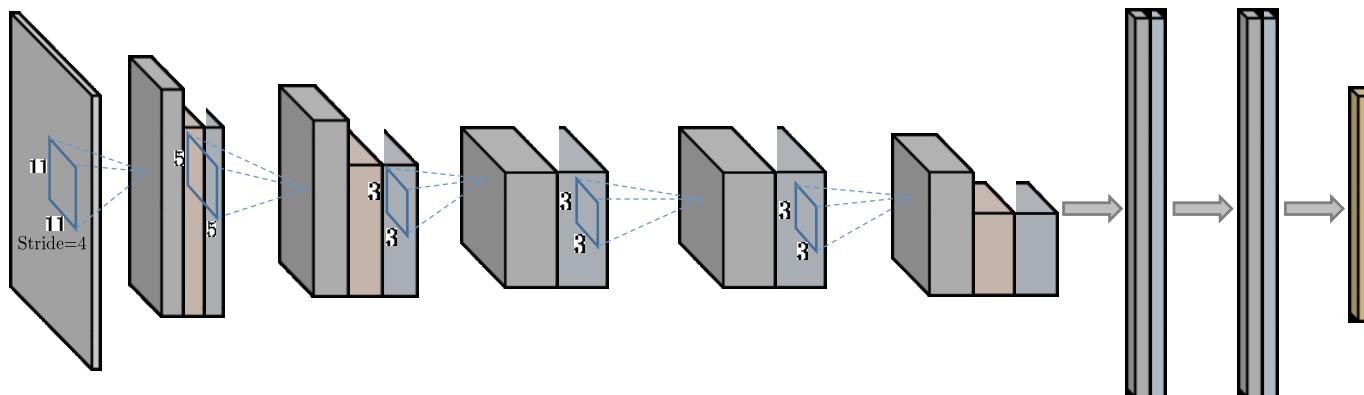
Оценки вероятностей
для каждого пикселя
 $Nc \times W \times H$

SoftDice

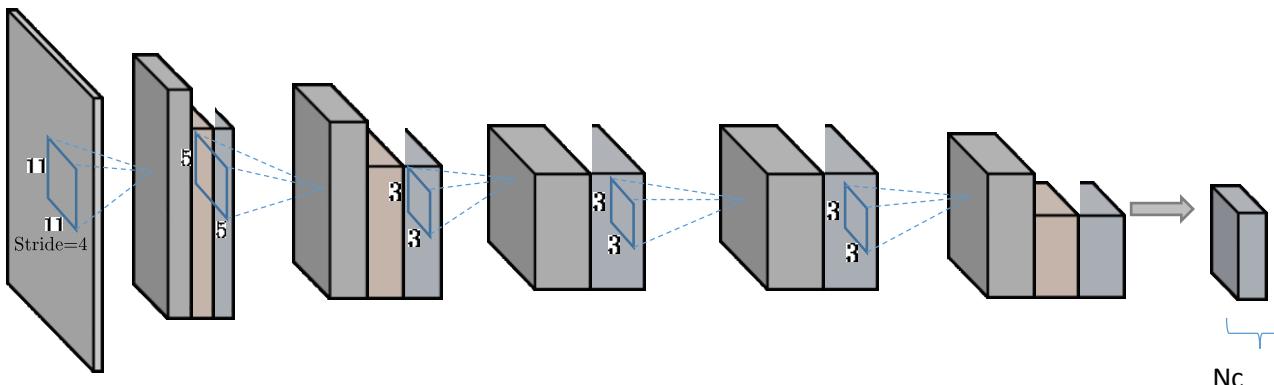
$$\frac{1}{Nc} \sum_c 1 - 2 \frac{\sum_{i,j} y_{c,i,j} \hat{y}_{c,i,j}}{\sum_{i,j} y_{c,i,j}^2 + \sum_{i,j} \hat{y}_{c,i,j}^2}$$

Архитектура семантической сегментации

Классификационная
Сеть AlexNet



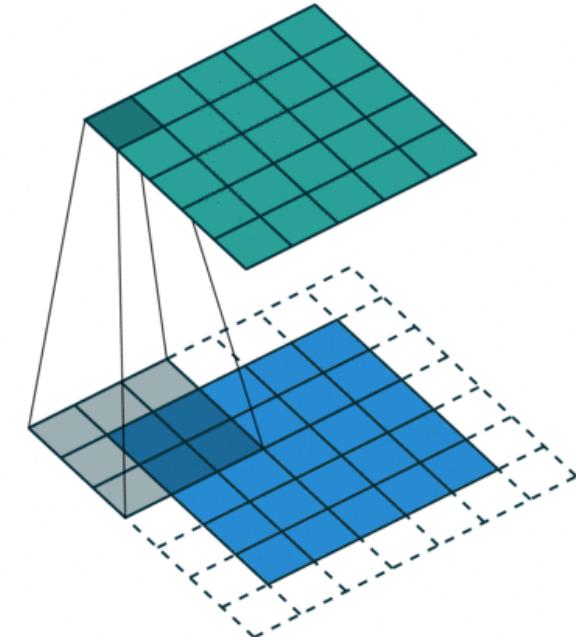
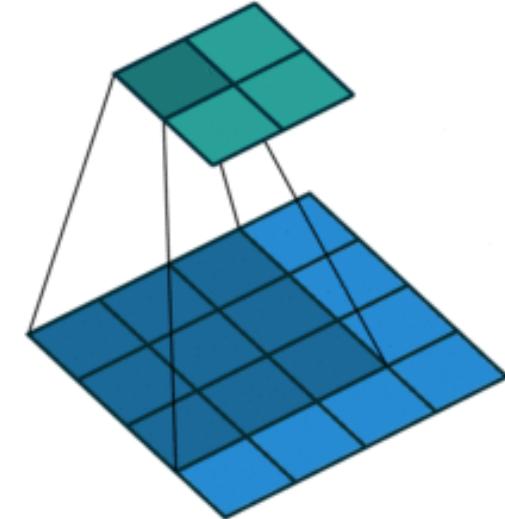
Сегментационная
Сеть на основе AlexNet



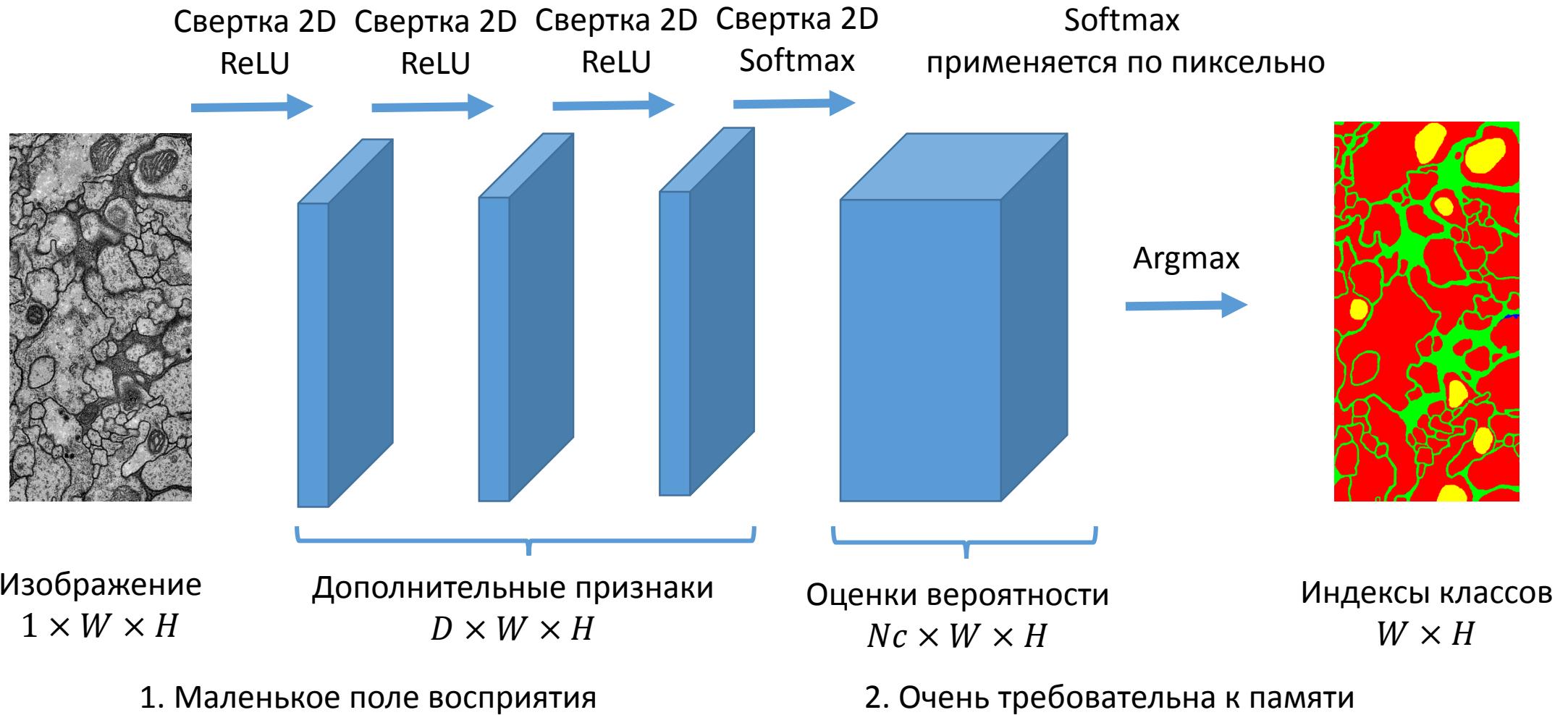
1. Маленькое разрешение
2. Ограниченнное поле восприятия

Поле восприятия

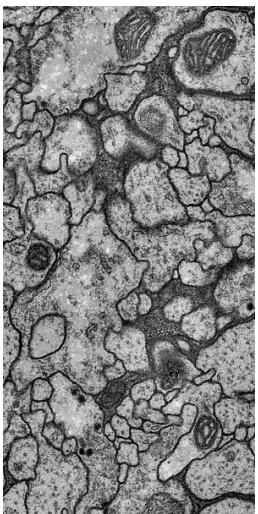
- Поле восприятия зависит от
 - Размера фильтра
 - Сдвига (stride)
 - Дилатации (Dilations)
 - Pooling
 - Количество сверток
- Границные эффекты (Paddings)
 - Valid
 - Same (0 – padding, mirror)



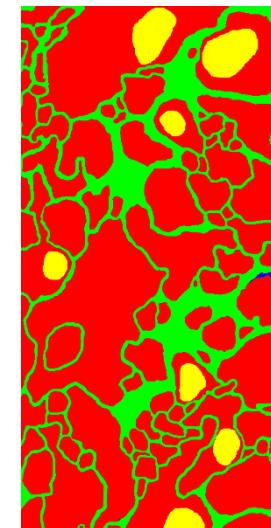
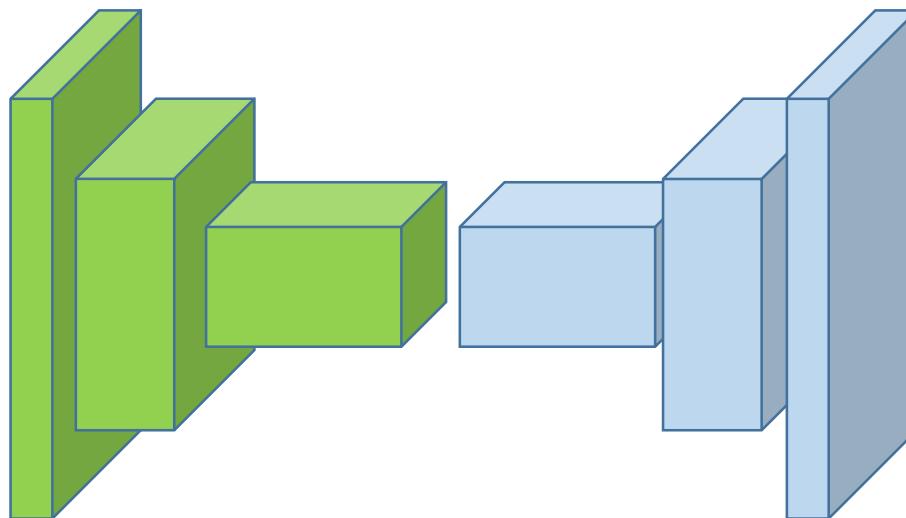
Архитектура семантической сегментации



Архитектура Encoder/Decoder



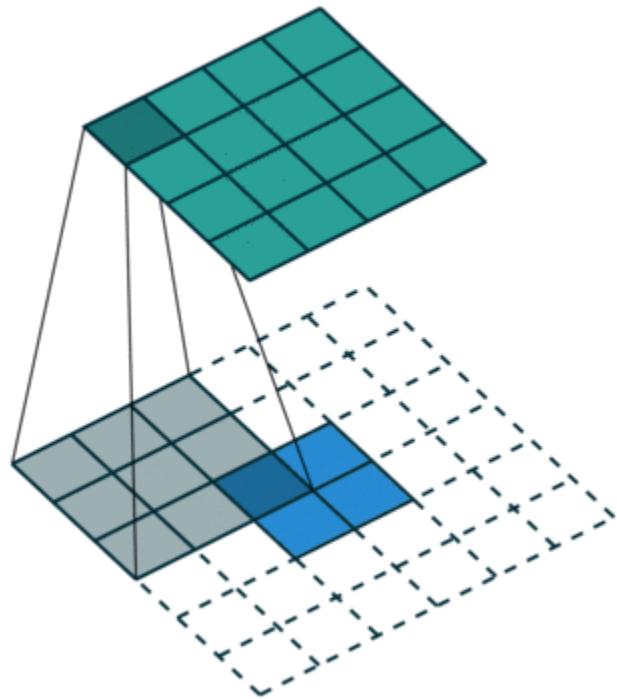
Изображение
 $1 \times W \times H$



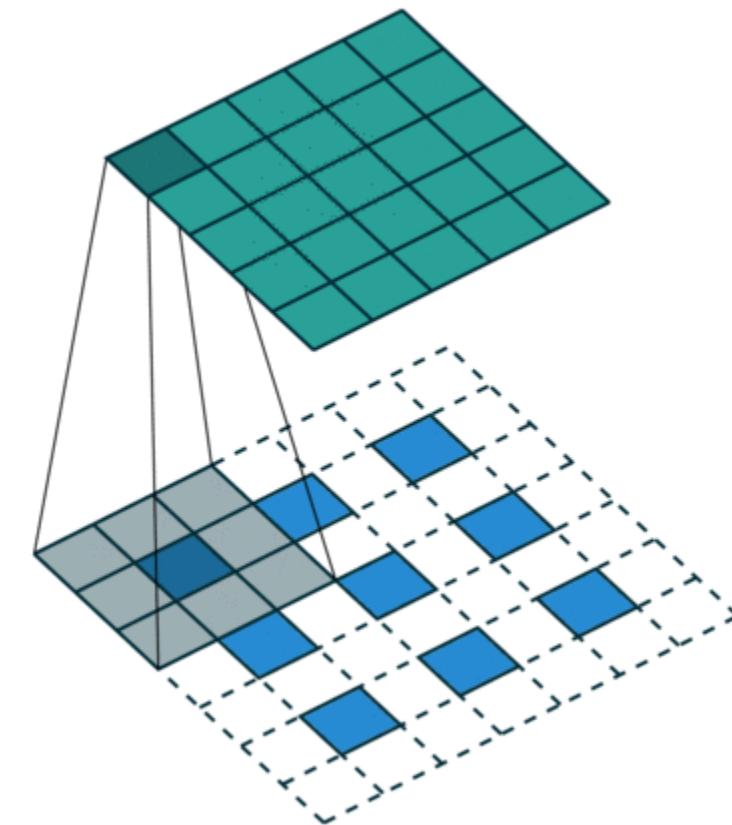
Индексы классов
 $W \times H$

Уменьшение пространственного разрешения увеличивает поле восприятия сети и экономит память

Обратная свертка (Transposed convolution)

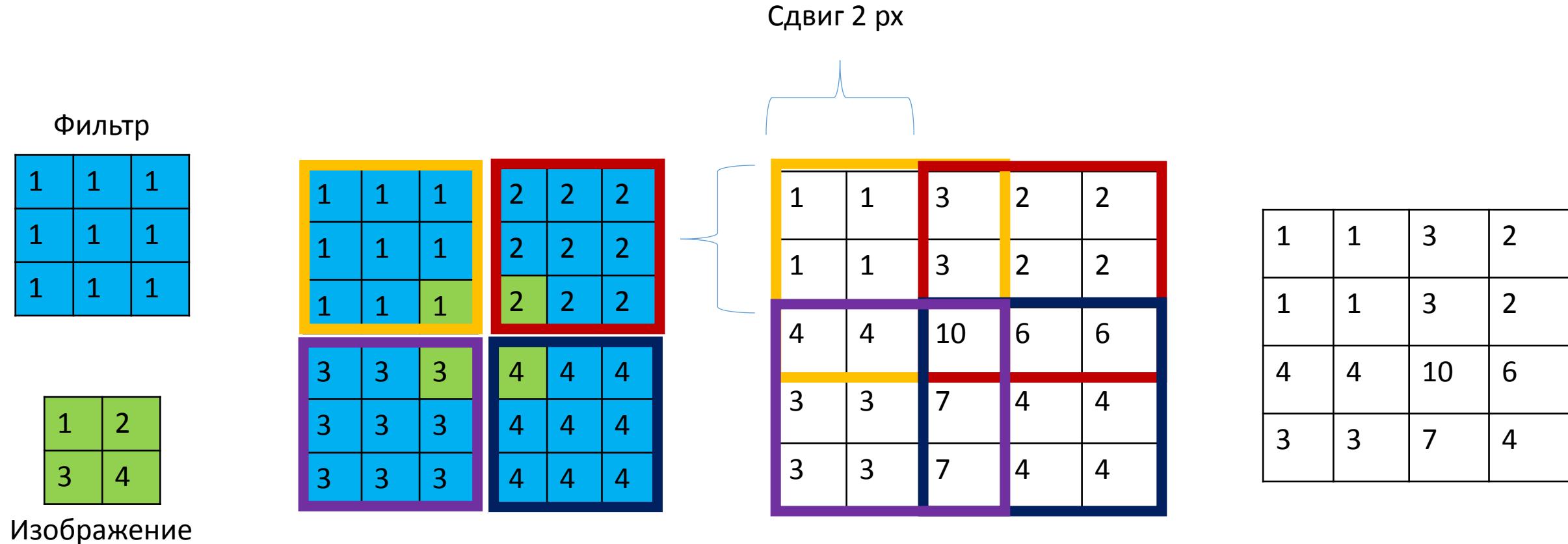


Без сдвига

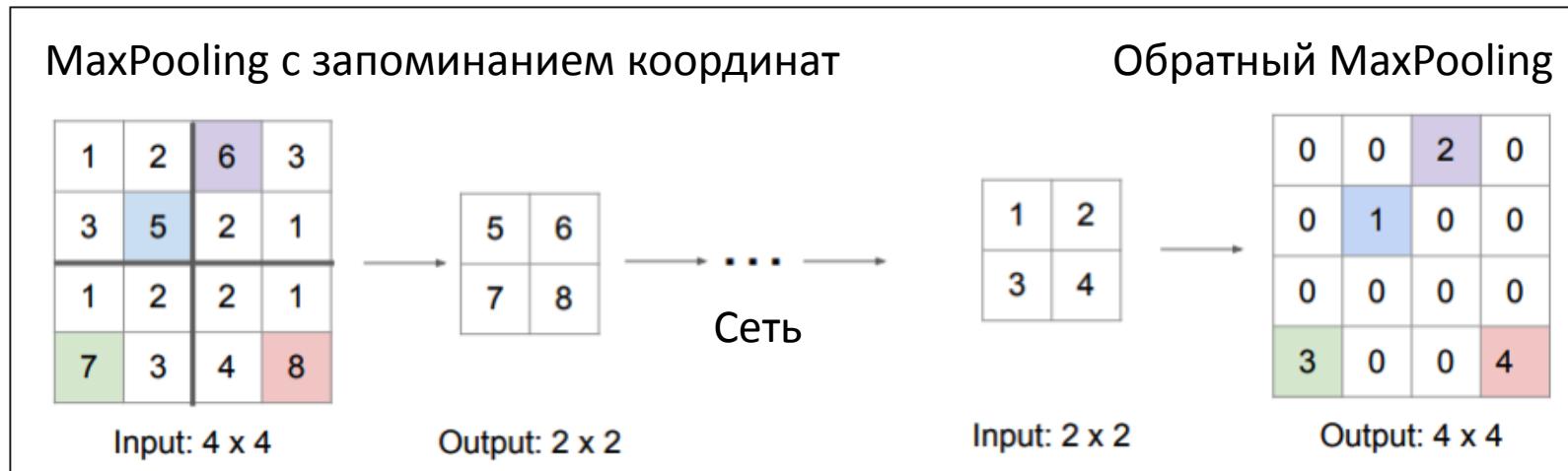


Со сдвигом 2

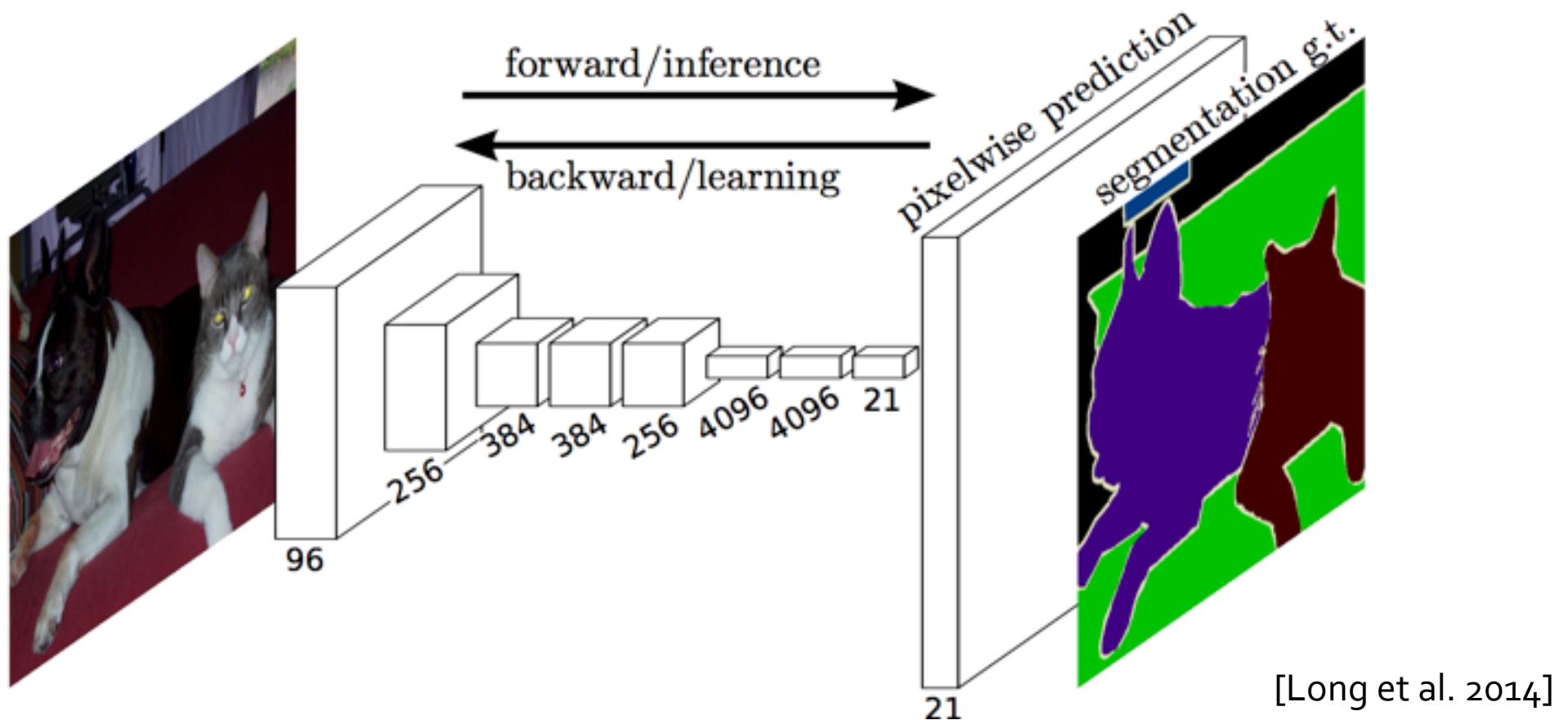
Как работает обращенная свертка с шагом 2



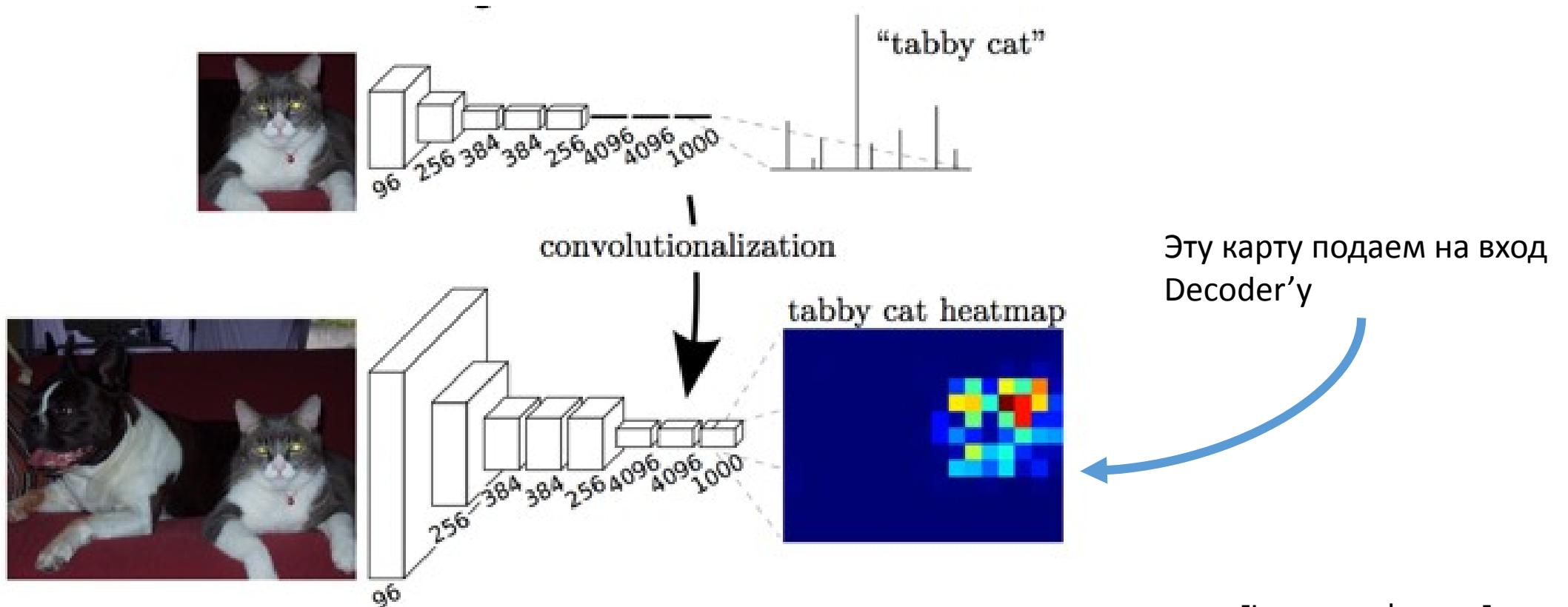
Другие методы повышения разрешения



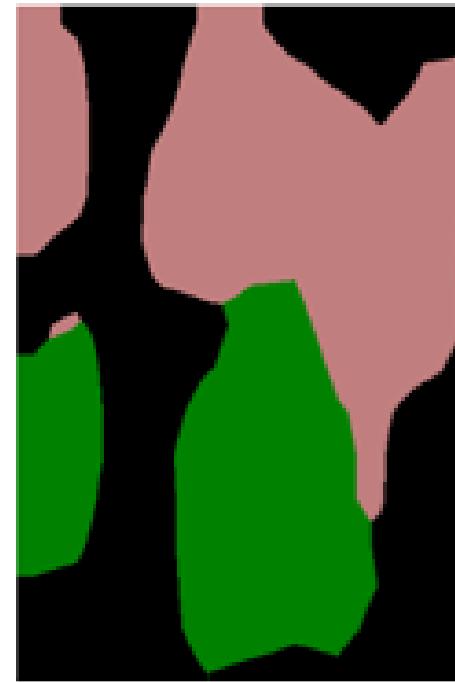
Использование Alexnet в качестве Encoder и Transposed Convolution



Превращаем полносвязанные линейные слои в свертки 1x1

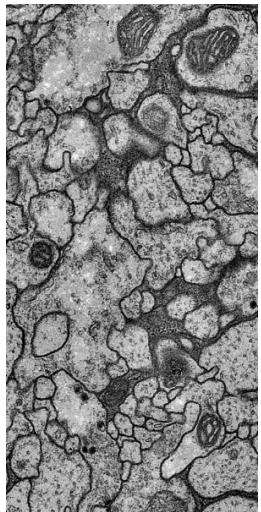


Проблемы с детализацией объектов

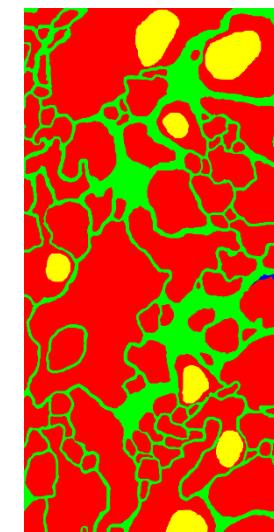
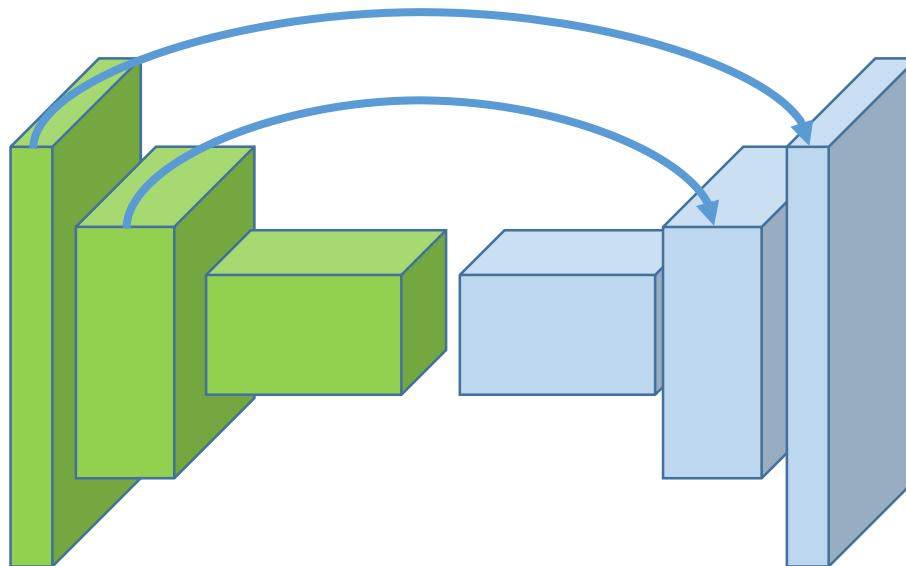


[Long et al. 2014]

Архитектура Encoder/Decoder



Изображение
 $1 \times W \times H$

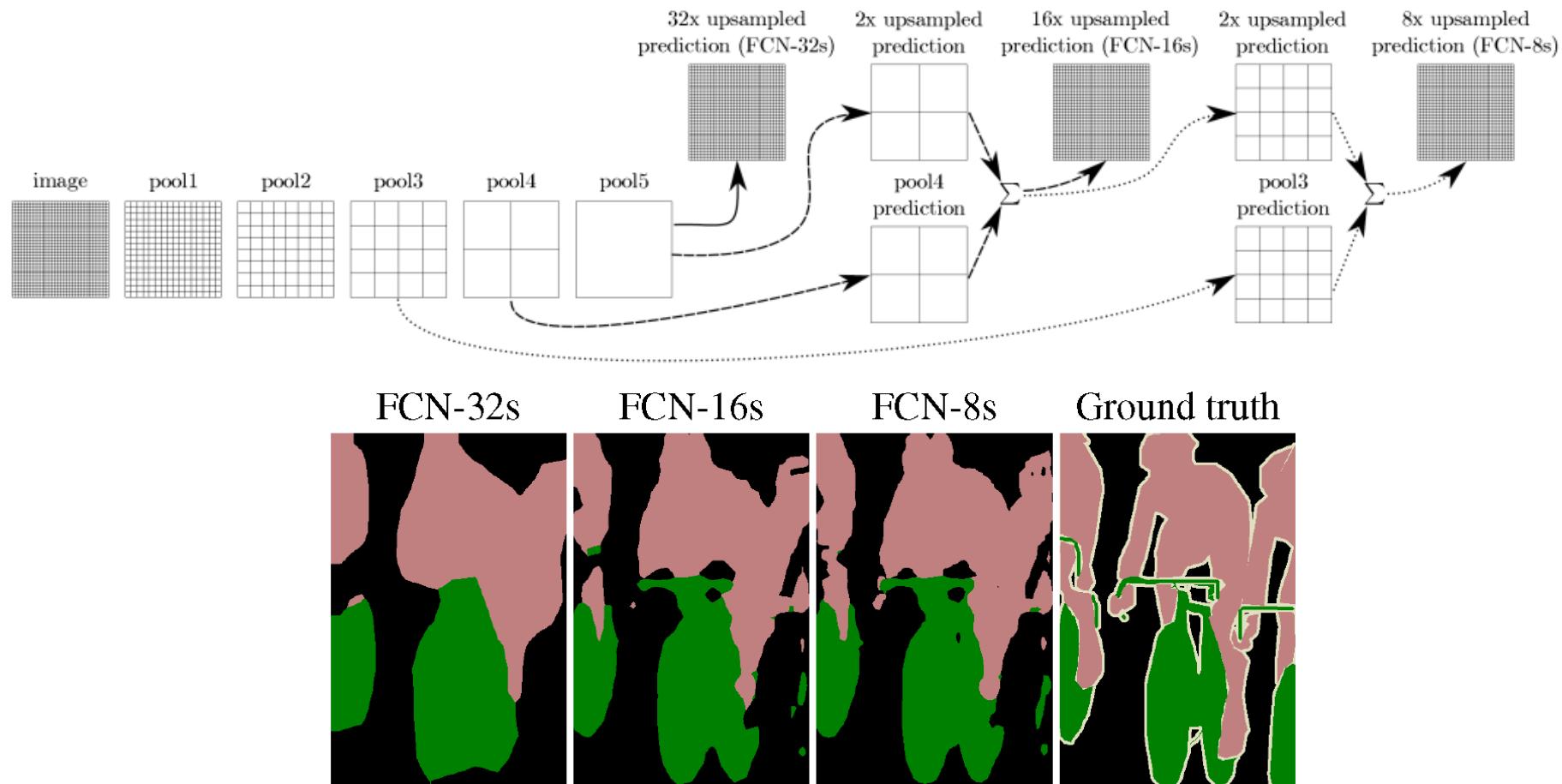


Индексы классов
 $W \times H$

Уменьшение пространственного разрешения увеличивает поле восприятия сети и экономит память

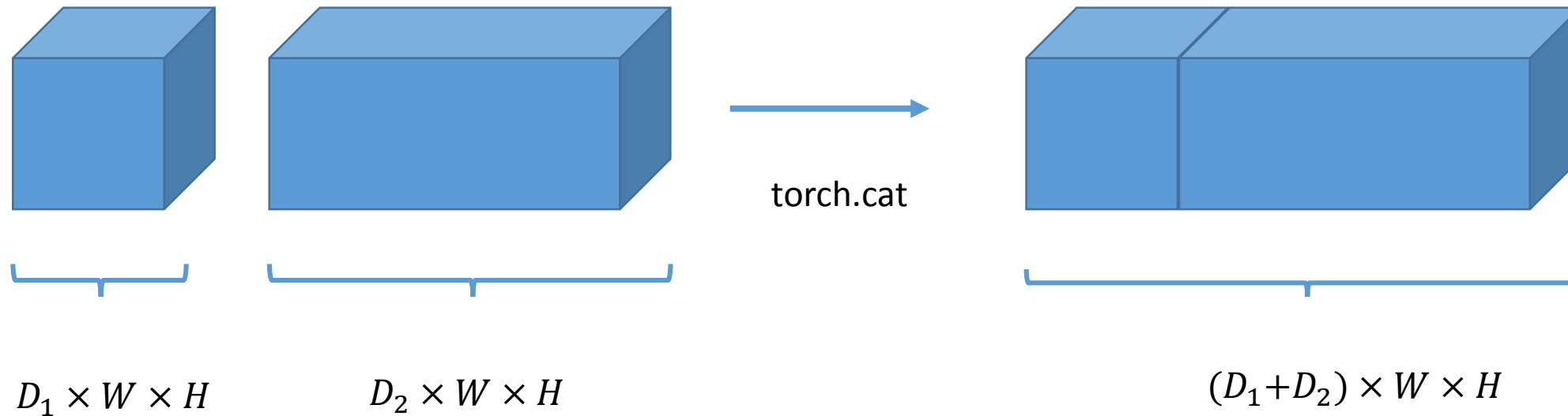
Пропускные связи (skip connections) позволяют сохранить исходную детализацию

Добавление пропускных связей

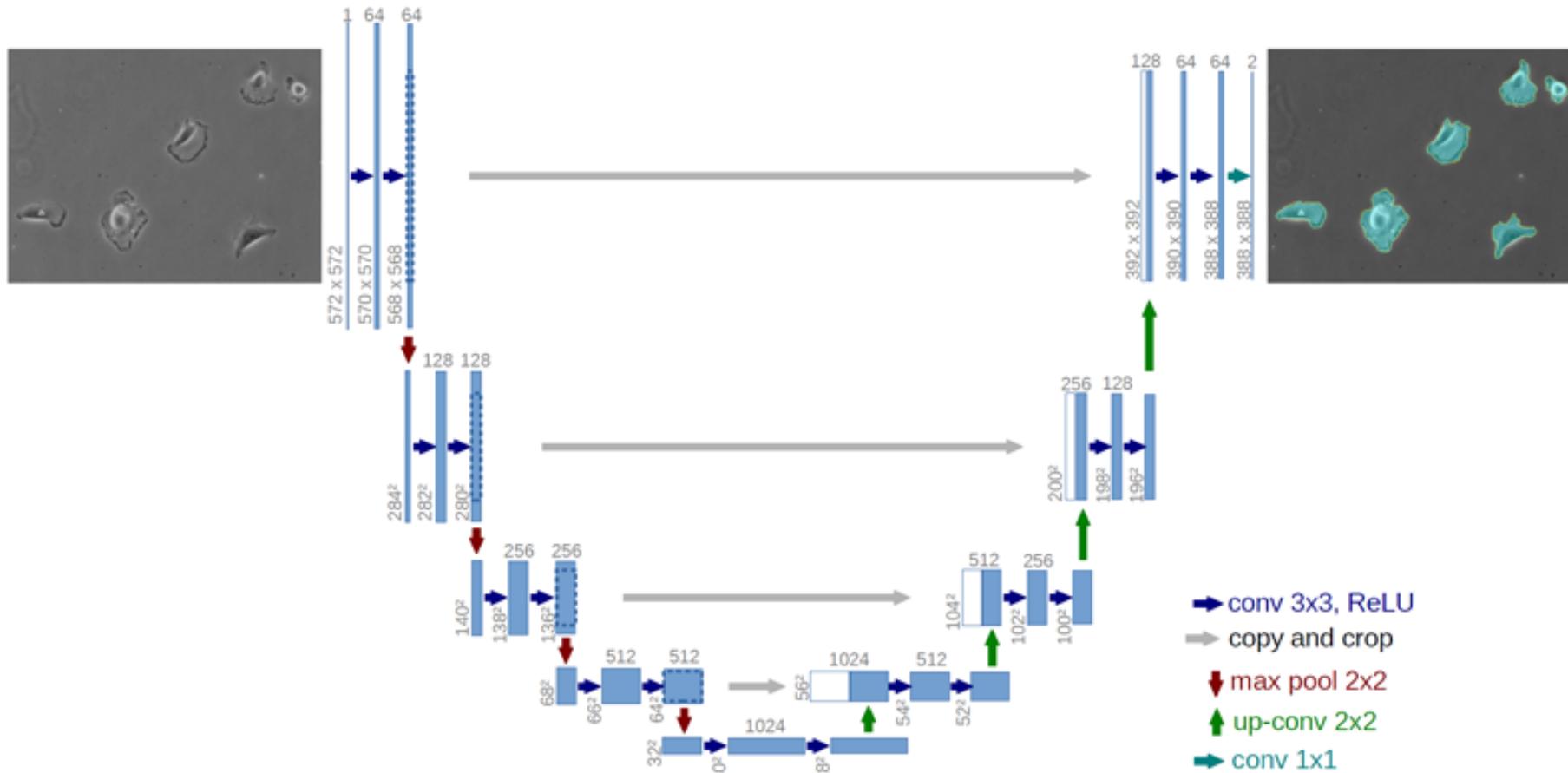


Реализация пропускных связей

- Сумма
- Конкатенация

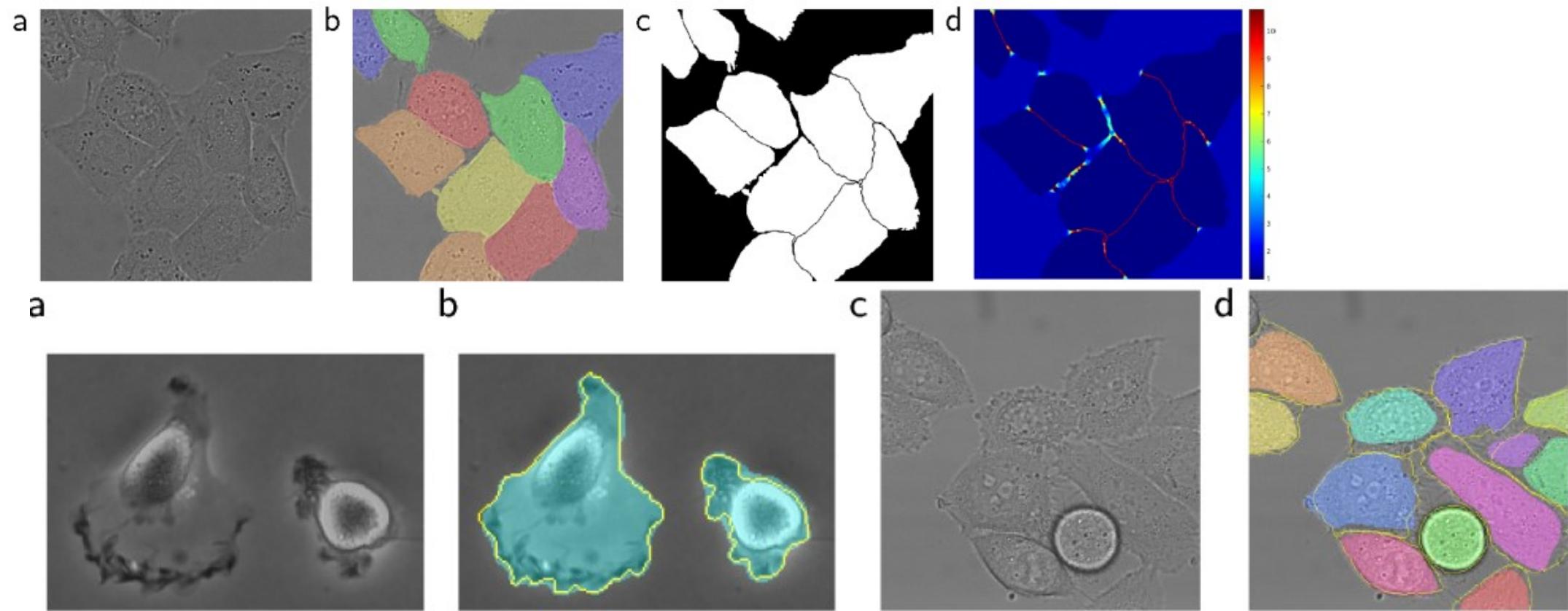


U-Net



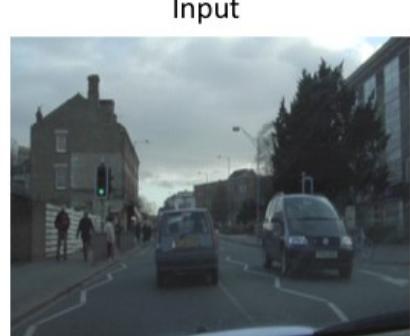
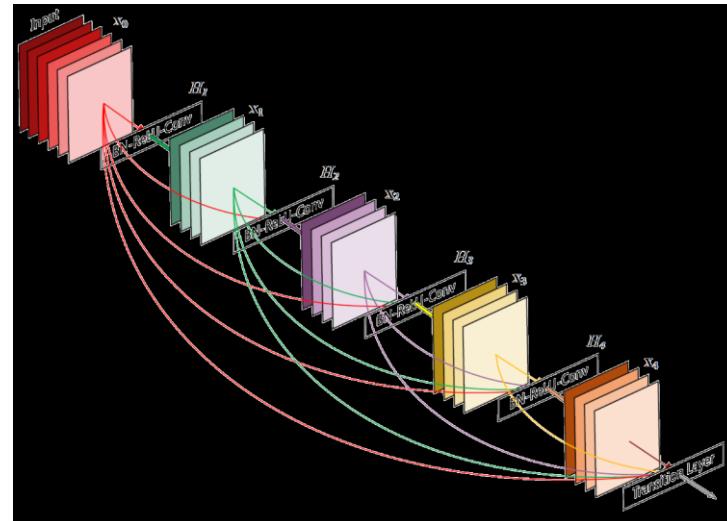
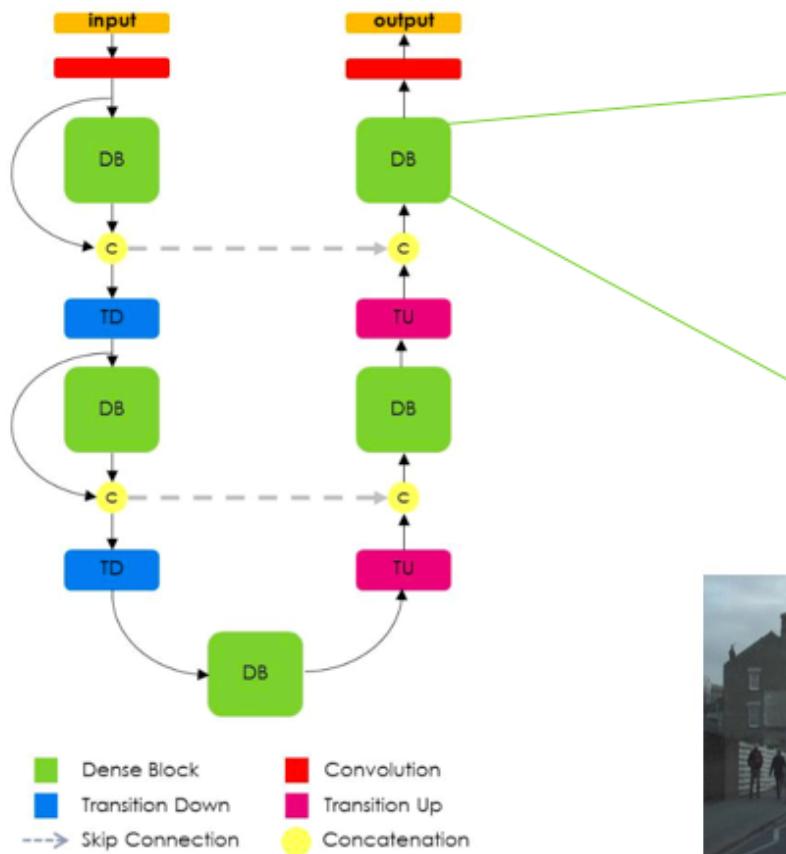
U-Net для выделения объектов

- Три класса: клетка, фон и границы. Все классы имеют разные веса и разный вклад в функцию потерь



[Ronnerberger et al. MICCAI15]

Объединение U-Net и DenseNet



Input



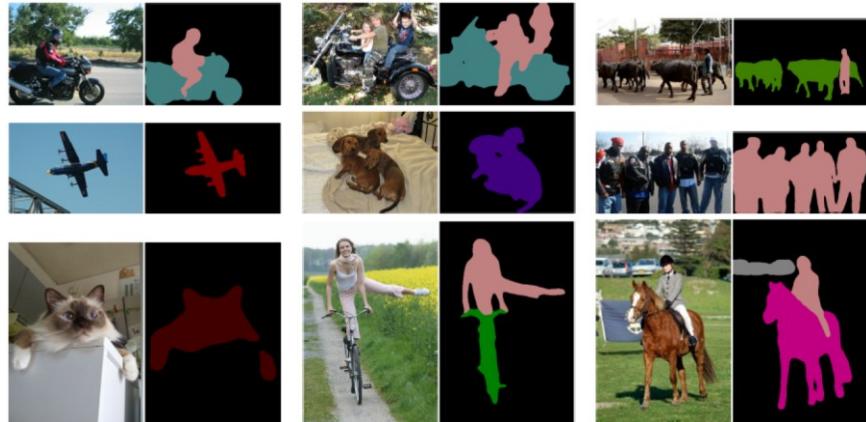
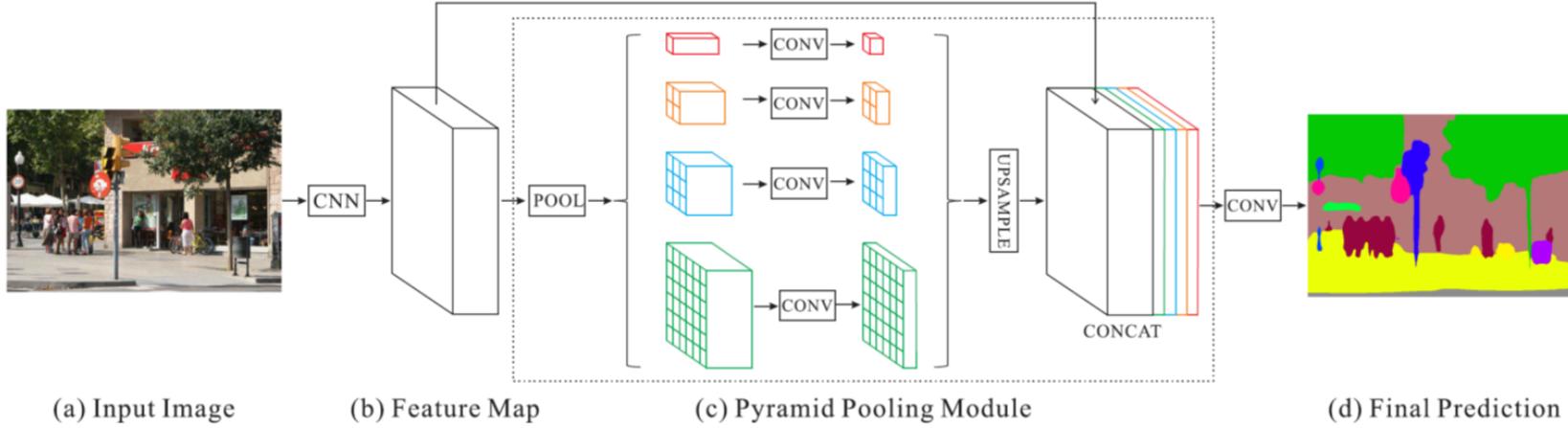
Ground truth



Prediction

[Jegou et al 17]

PSPNet



| Method | Mean IoU(%) | Pixel Acc.(%) |
|------------------------------|--------------|---------------|
| ResNet50-Baseline | 37.23 | 78.01 |
| ResNet50+B1+MAX | 39.94 | 79.46 |
| ResNet50+B1+AVE | 40.07 | 79.52 |
| ResNet50+B1236+MAX | 40.18 | 79.45 |
| ResNet50+B1236+AVE | 41.07 | 79.97 |
| ResNet50+B1236+MAX+DR | 40.87 | 79.61 |
| ResNet50+B1236+AVE+DR | 41.68 | 80.04 |

Метрики качества семантической сегментации

- IoU (Jaccard index) (отношение пересечения к объединению)

$$\bullet IoU = \frac{\text{target} \cap \text{prediction}}{\text{target} \cup \text{prediction}}$$

- Accuracy (Процент верно классифицированных пикселей)

$$\bullet accuracy = \frac{TP+TN}{TP+NP+FP+FN}$$

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



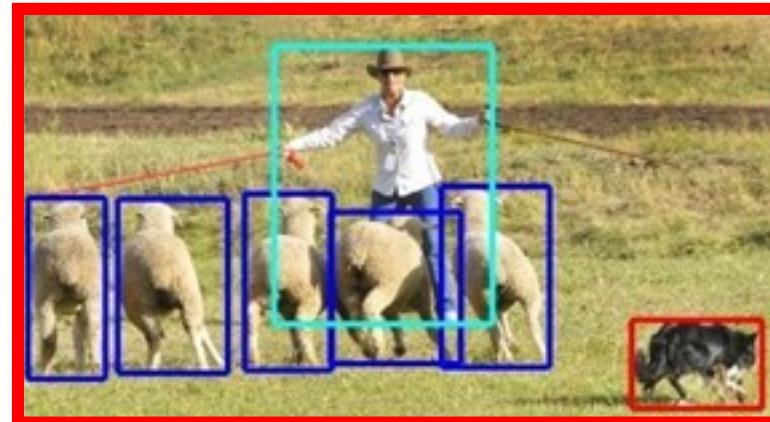
Практика семантическая сегментация

- На сервере <http://107.178.212.42>
- В терминале:
 - cd ostrov2018
 - git commit –а –м “”
 - git pull
- Если нету папки ostrov2018
 - git clone <https://github.com/kulikovv/ostrov2018>
- Открыть Notebook:
 - 7_SemanticSegmentation.ipynb

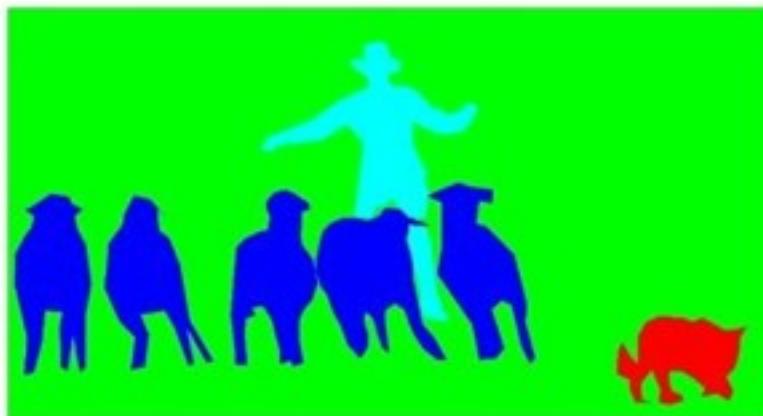
Задачи компьютерного зрения



Классификация



Обнаружение объектов



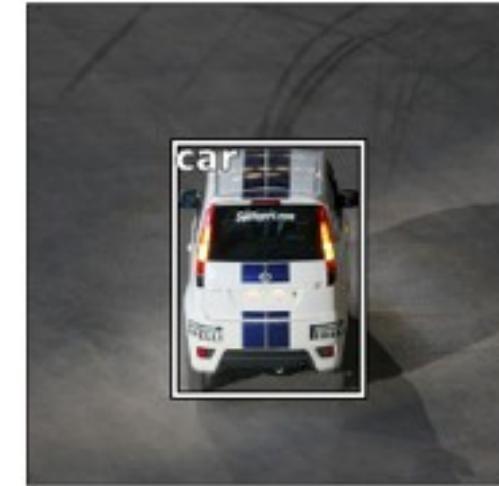
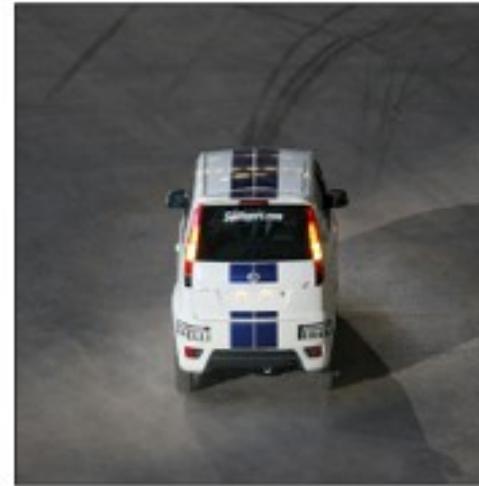
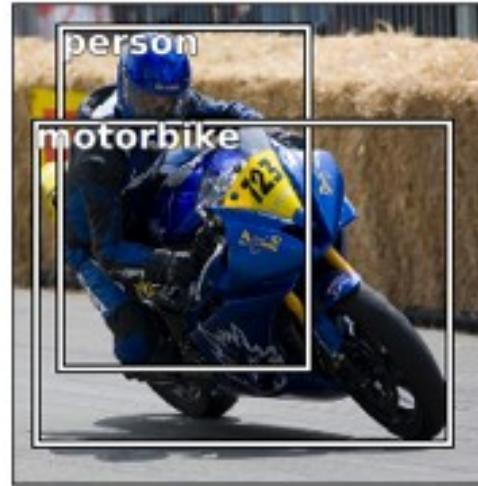
Семантическая сегментация



Сегментация объектов

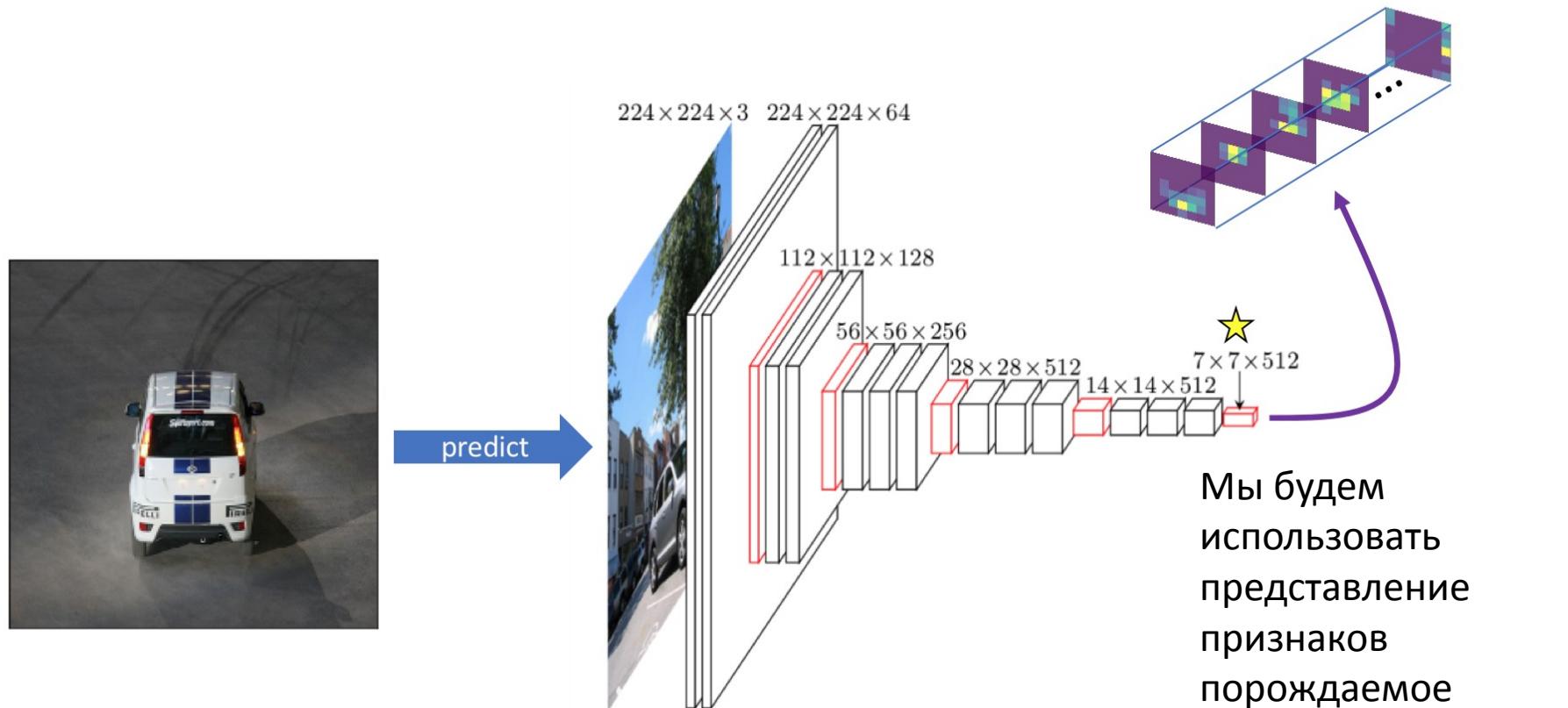
[Lin et al. 2015]

Задача обнаружения объектов



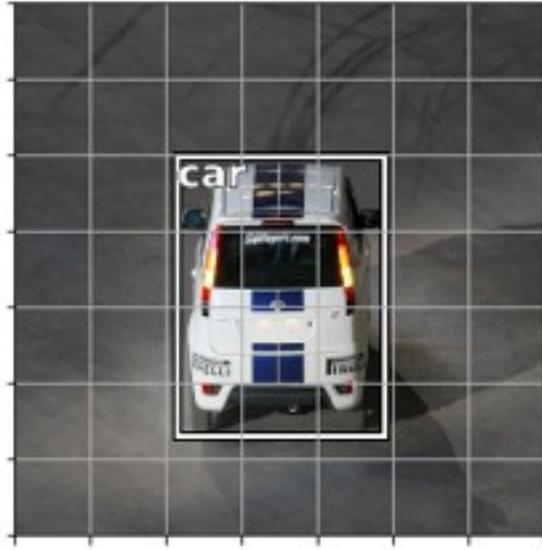
Задача: обнаружить объекты заранее заданных классов и найти для них описывающие прямоугольники.

Использование передобученной нейронной сети для получения описания



Slide credits: jeremyjordan.me

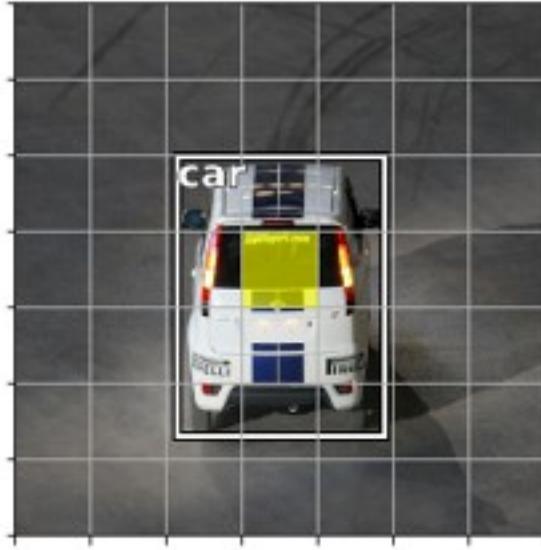
Подготовка исходных данных



Входные данные
должны иметь
описывающий
прямоугольник для
каждого объекта



Нам нужны **координаты**
центра объекта на **сетке**,
полученной на
последнем слое

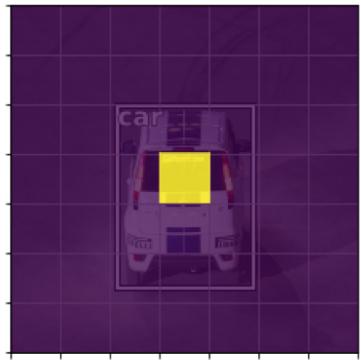


Эта ячейка будет
“отвечать” за
нахождение объекта на
изображении.

Slide credits: jeremyjordan.me

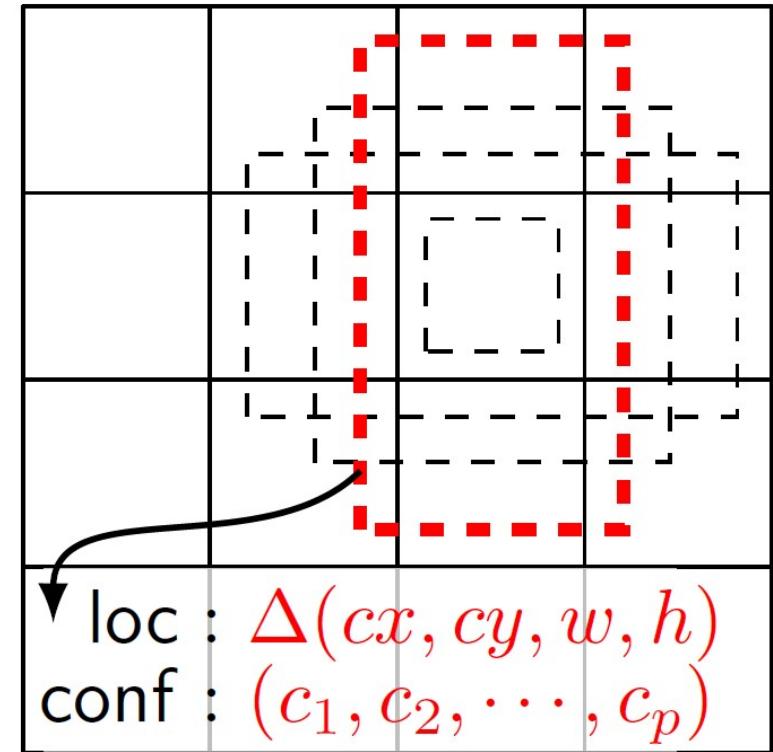
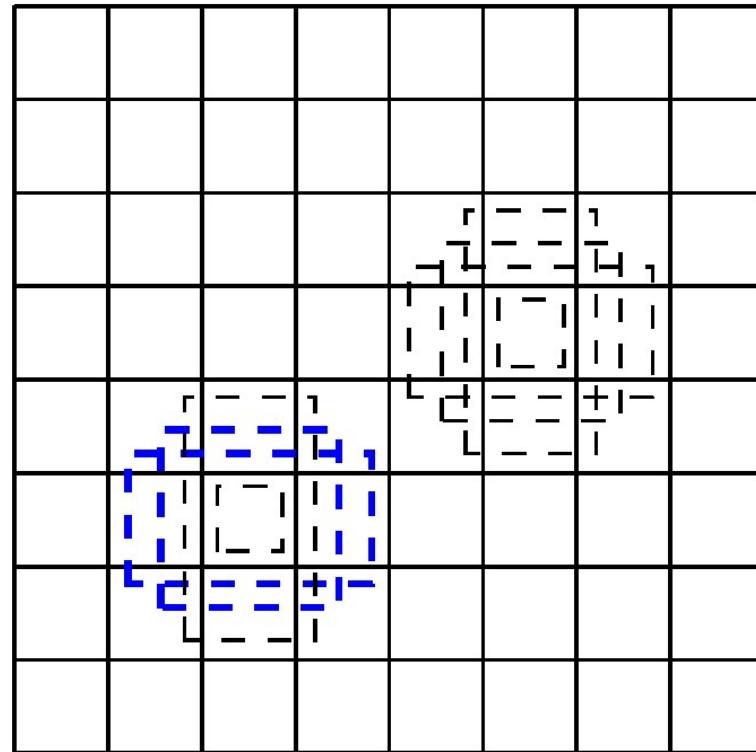
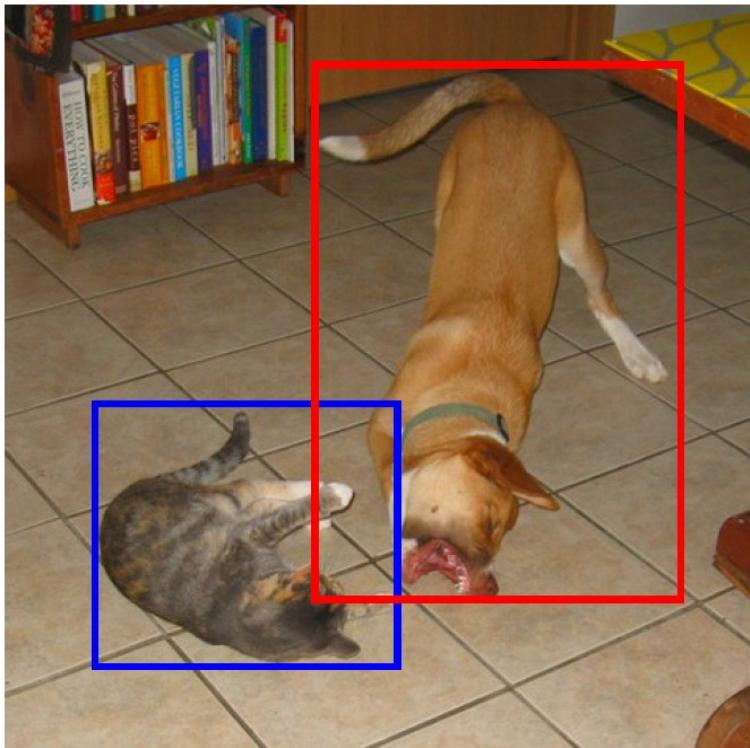
Кодировка данных

- t_x, t_y, t_w, t_h – координаты объекта относительно центра клетки, нормированные на размер изображения
- p_{obj} - вероятность того, что данный прямоугольник объект
- c_1, c_2, \dots, c_C - классификационная часть (вероятности принадлежать объекту $\sum c_i = 1$)



Описание одного объекта

Priors/Anchor boxes



loc : $\Delta(cx, cy, w, h)$
conf : (c_1, c_2, \dots, c_p)

[Liu et al. ECCV16]

ФУНКЦИЯ ПОТЕРЬ

Отклонение prior'a со смещением от истинного прямоугольника

$$F_{match}(x, l, g) = \frac{1}{2} \sum_{i,j} x_{i,j} \|l_i - g_j\|^2$$

Модифицированная бинарная кросс энтропия

$$F_{conf}(x, c) = - \sum_{i,j} x_{i,j} \log(c_i) - \sum_{i \in Neg} \log(c_0)$$

Суммарная функция потерь

$$F(x, l, c) = \alpha F_{match}(x, l, g) + F_{conf}(x, c)$$

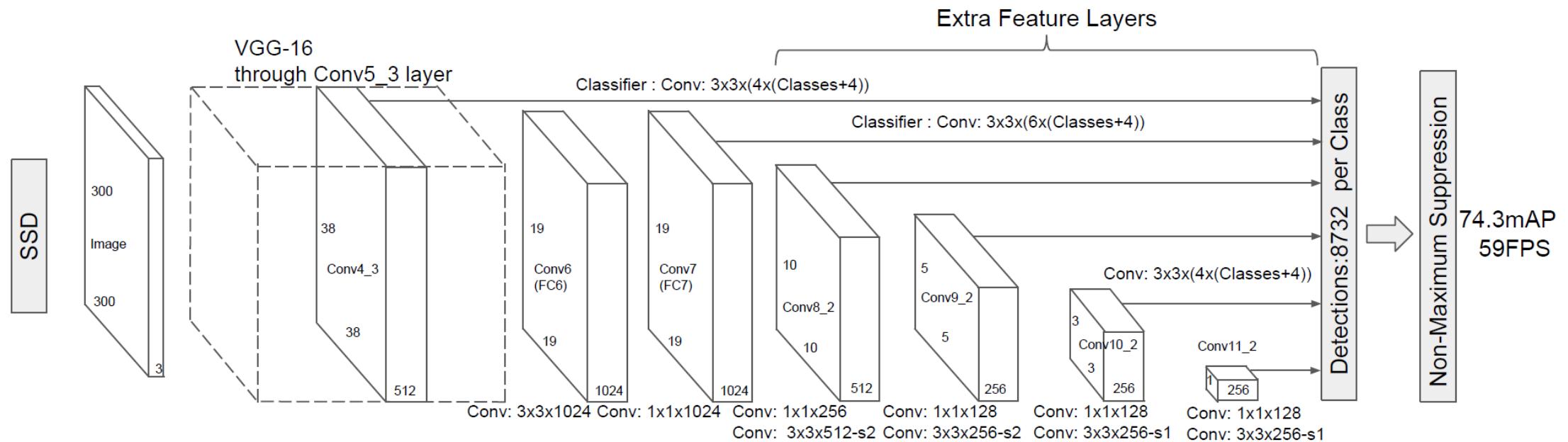
Ищем $\hat{x} = argmin_x F(x, l, c, g) | x_{i,j} \in \{0,1\}, \sum_i x_{i,j} = 1$

Шаг 1. Считаем прямой проход получаем оценки вероятностей и координаты для каждого класса

Шаг 2. Решаем задачи оптимального распределения ресурсов венгерским алгоритмом с prior и получаем x

Шаг 3. Для наилучших объектов вычисляем функцию потерь, таким образом чтобы каждый объект входил только один раз

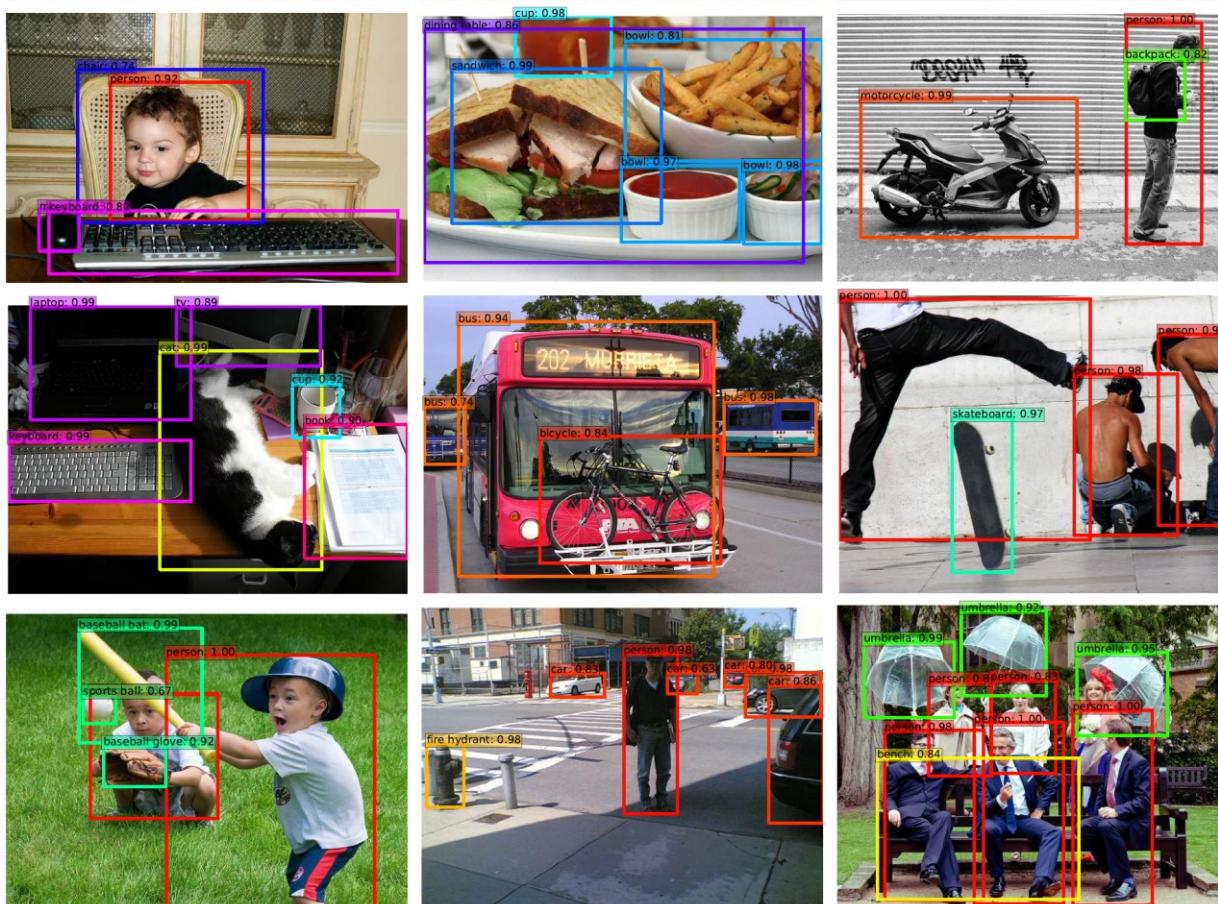
SSD detector



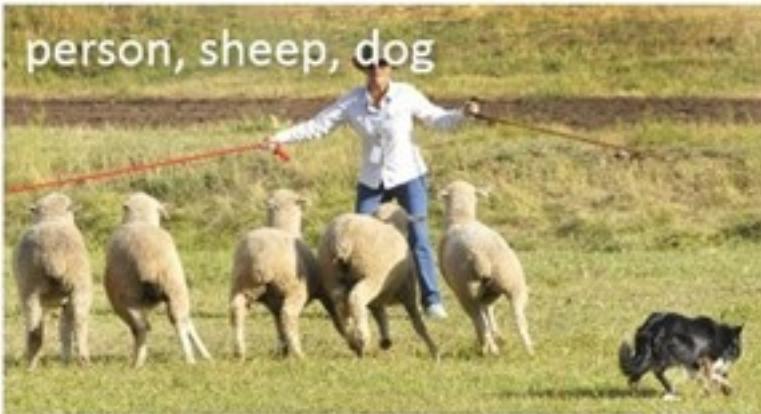
<https://github.com/amdegroot/ssd.pytorch>

[Liu et al. ECCV16]

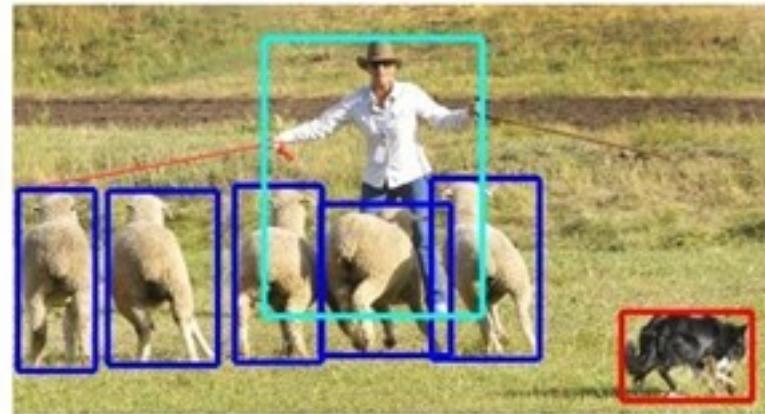
Примеры SSD



Задачи компьютерного зрения



Классификация



Обнаружение объектов



Семантическая сегментация



Сегментация объектов

[Lin et al. 2015]

Сегментация объектов

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

0 - фон

1 - кот

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 5 | 5 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 | 5 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 | 5 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 | 5 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 | 5 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

0 - фон

N – кот #N

Сегментация объектов (Instance segmentation)

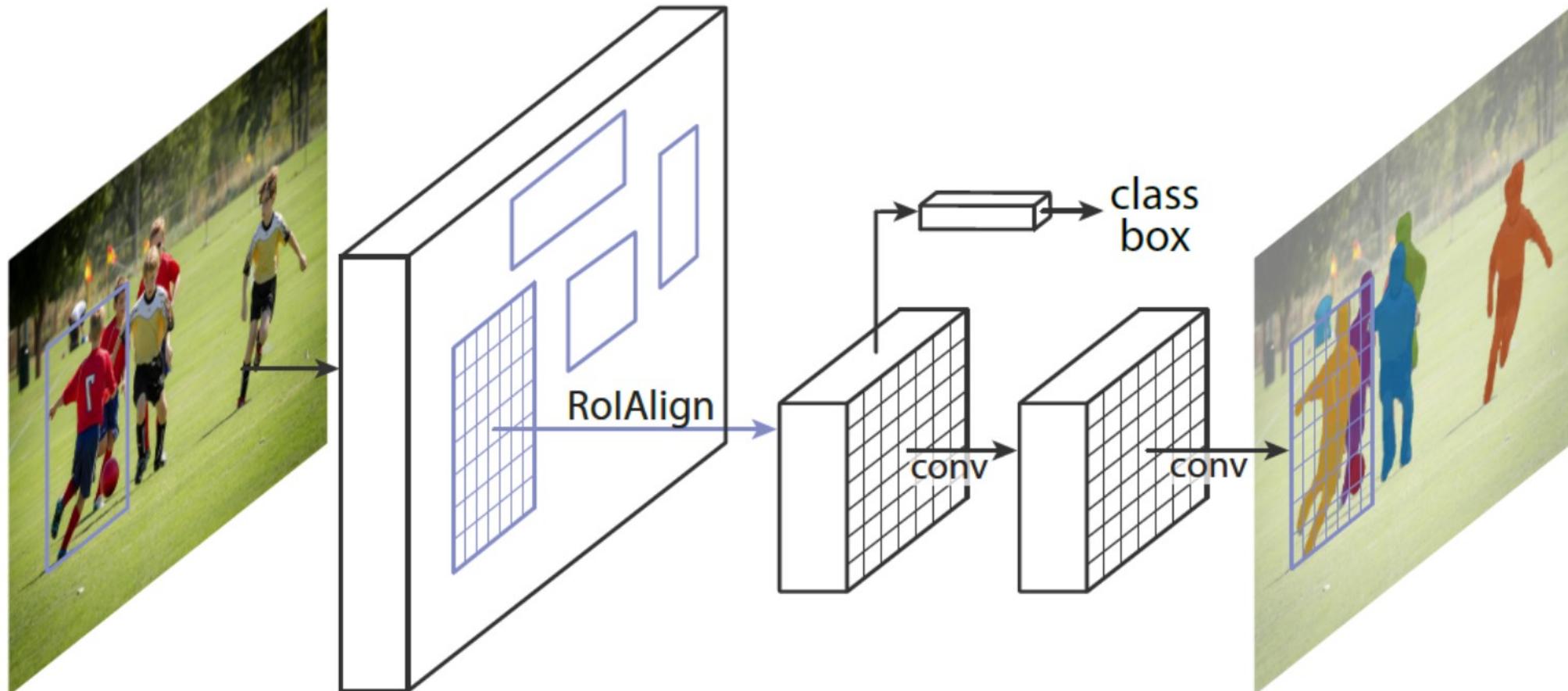
Семантическая сегментация



Сегментация объектов

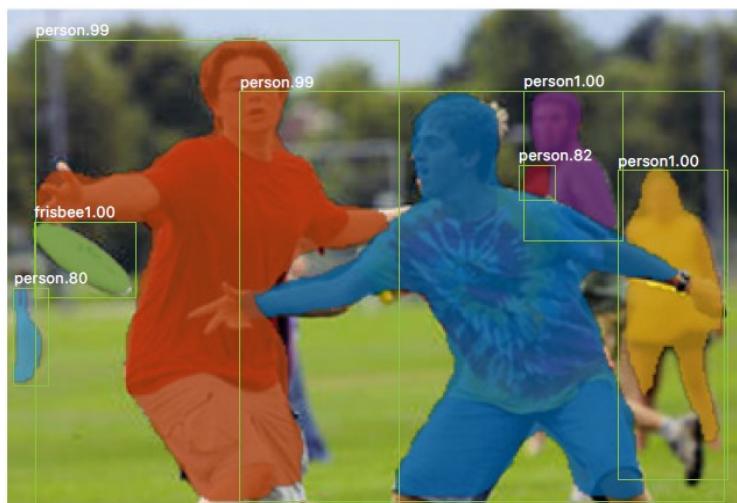
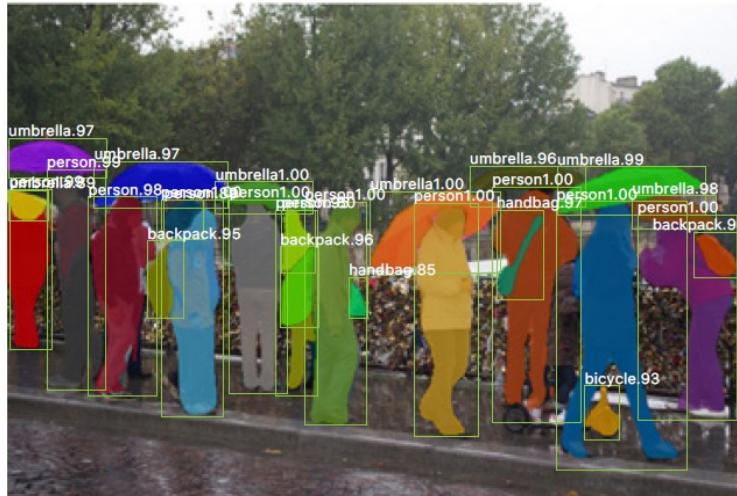
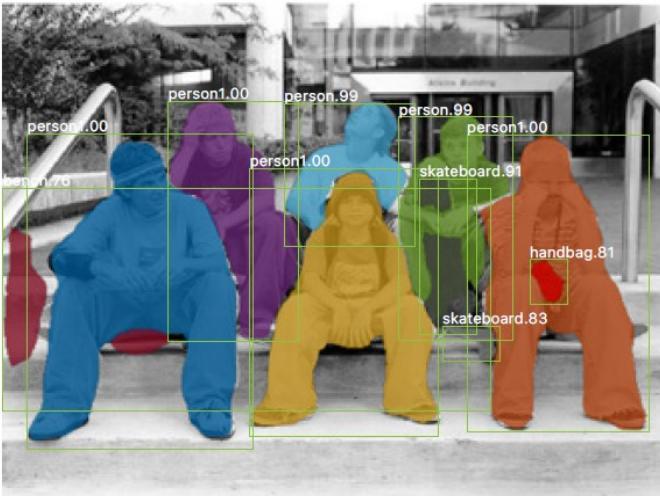


Выделение объектов



Две головы: Одна голова обнаруживает объект, затем прямоугольники вырезаются, приводятся к стандартному размеру и передаются на вход семантической сегментации

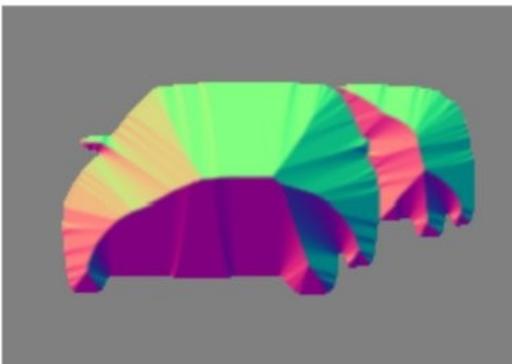
Результат Mask R-CNN



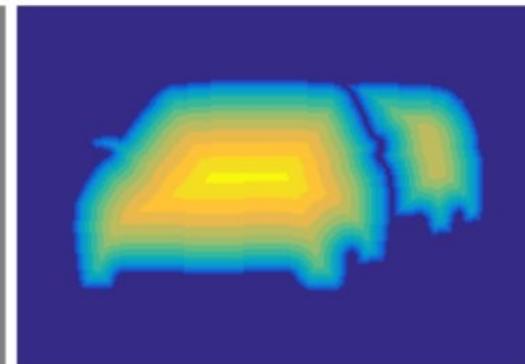
DeepWatershed



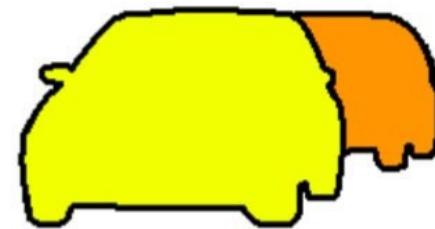
(a) Input Image



(b) GT angle of \vec{u}_p



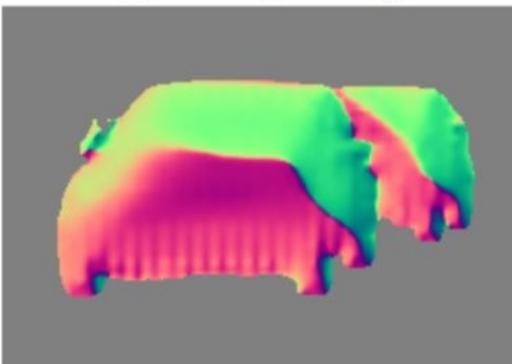
(c) GT Watershed Energy



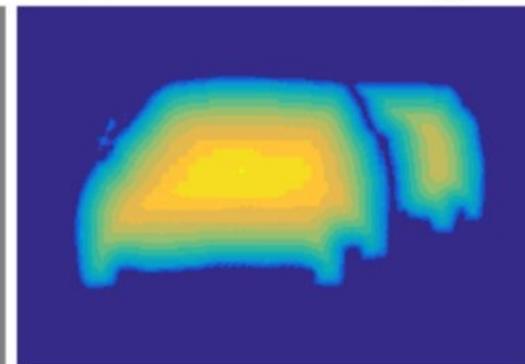
(d) GT Instances



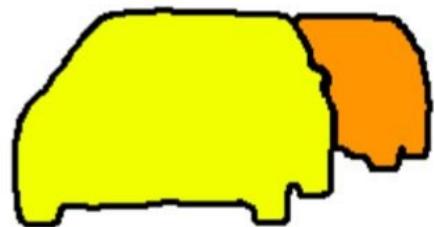
(e) Sem. Segmentation of [34]



(f) Pred. angle of \vec{u}_p



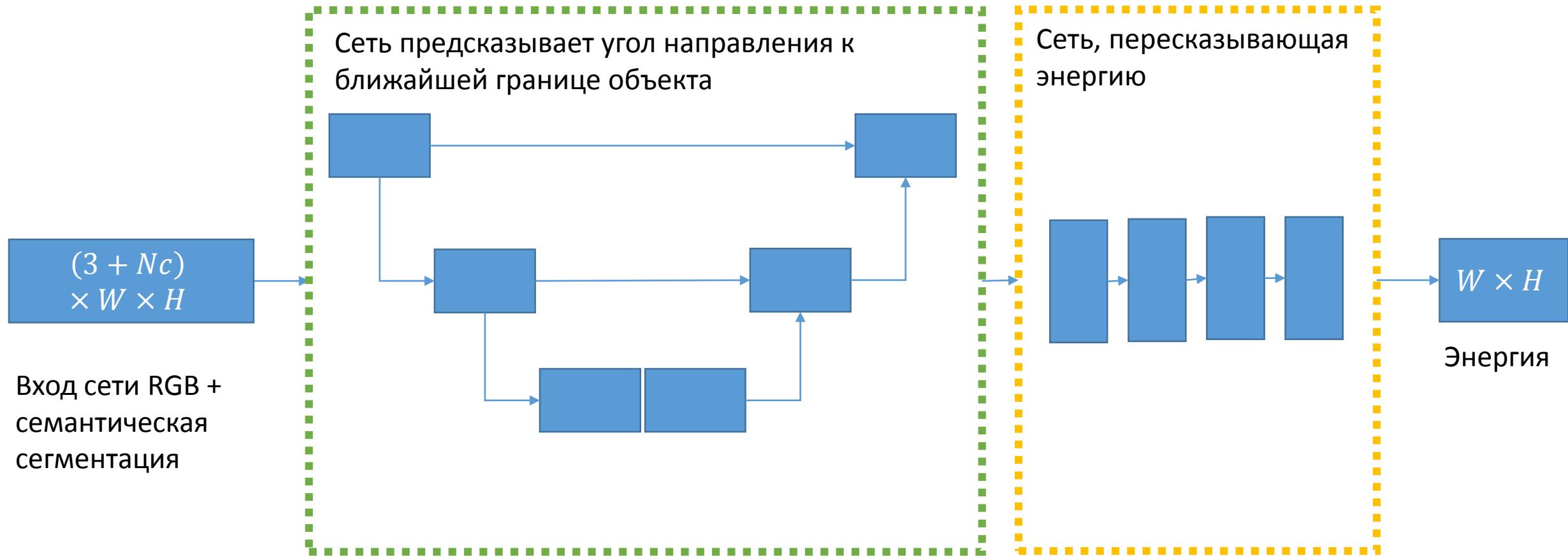
(g) Pred. Watershed Transform



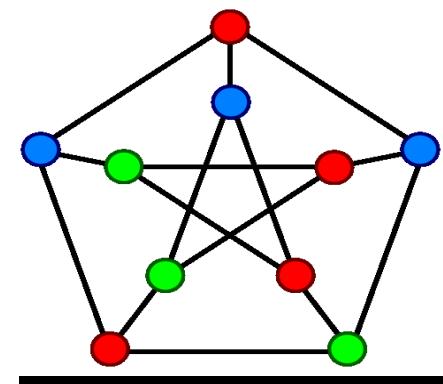
(h) Pred. Instances

[Bai et al. CVPR 2017]

DeepWatershed



Алгоритм DeepColoring



| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 5 | 5 | 5 | 0 | 0 | |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 0 | |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 0 | |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 0 | |
| 0 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 0 |
| 0 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 5 | 0 | 5 | 0 | |
| 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 4 | 0 | 4 | 5 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

0 - фон

N – кот #N

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 5 | 0 | 2 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 4 | 0 | 4 | 5 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 4 | 0 | 4 | 5 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

0 - фон

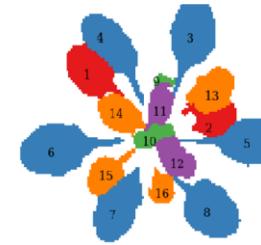
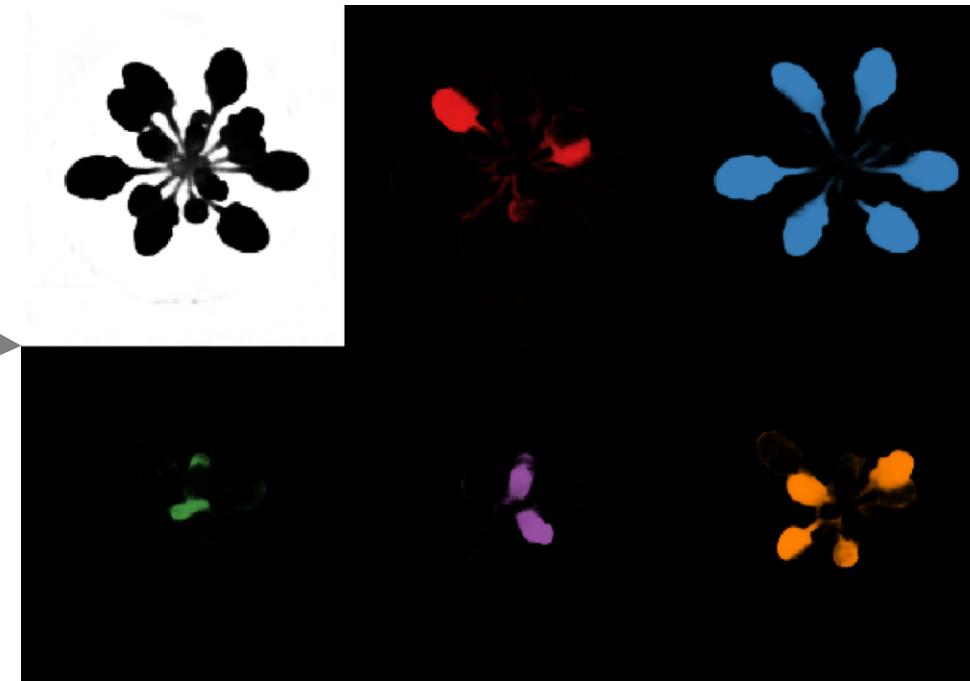
К – заданных групп объектов

DeepColoring



Применяем
сверточную сеть для
семантической
сегментации

Все объекты разделены на
непересекающиеся группы и разнесены на
разные карты



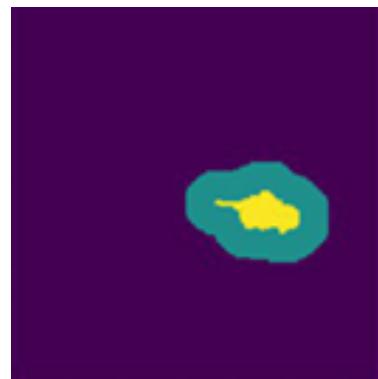
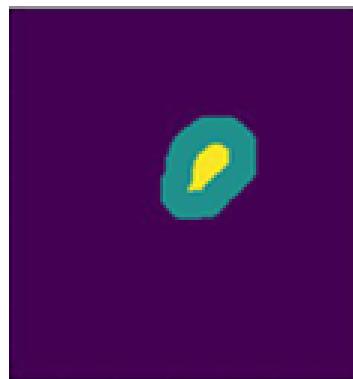
Попиксельно берем
argmax и применяем
метод поиска
связанных
компонент

Входные данные

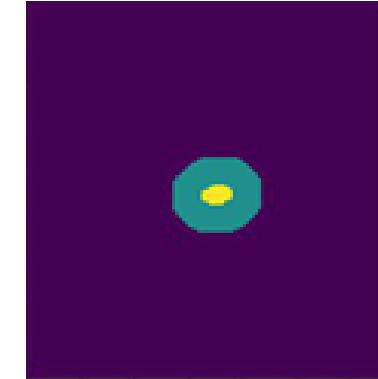
$$M_{halo}^k = dilate(M^k, m) \setminus M^k$$



Изображение



• • •

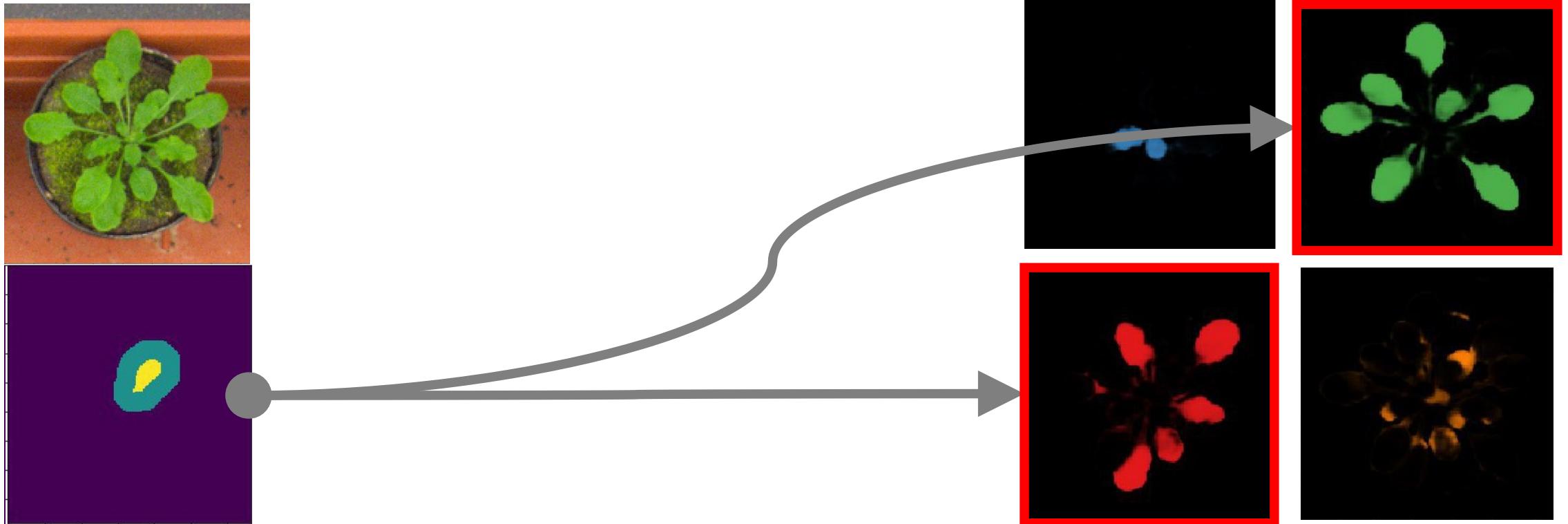


Маски для каждого объекта

Алгоритм вычисления функции потерь

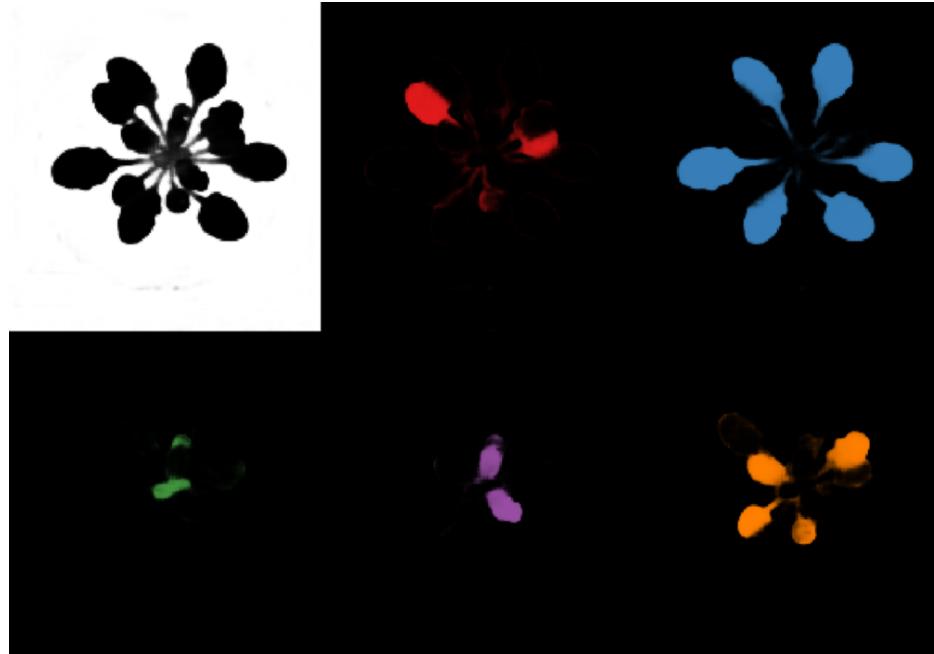
1. Для каждого объекта найти оптимальную карту на выходе из нейронной сети
2. Рассчитать суммарное качество размещения объектов по картам
3. Добавить качество сегментации фона

Шаг 1. Выбор карты для объекта



$$c_k = \operatorname{argmax}_{c=2}^C \left(\frac{1}{|M^k|} \sum_{p \in M^k} \log y[c, p] + \mu \frac{1}{|M_{halo}^k|} \sum_{p \in M_{halo}^k} \log(1 - y[c, p]) \right)$$

Шаг 2 и 3: вычисление функции потерь



$$L(x, w) = - \sum_{k=1}^K \sum_{p \in M^k} \log y[c_k, p] - \sum_{p \in Background} \log y[0, p]$$

Пример сегментации дорожного движения



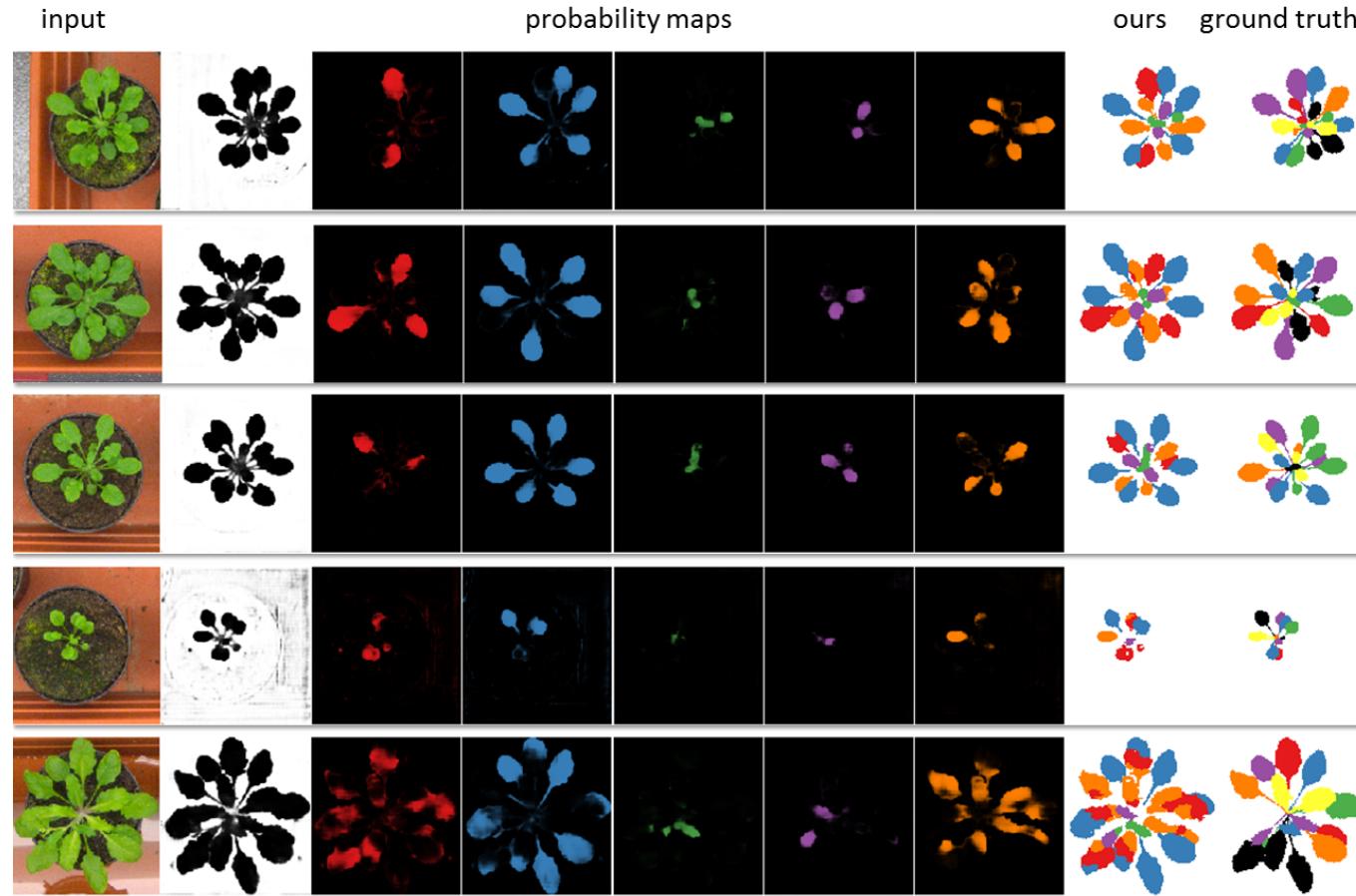
Пример сегментации дорожного движения в Москве



Применение для биомедицины



DeepColoring для фенотипирования растений



Стилизация изображений

A



B



C



D

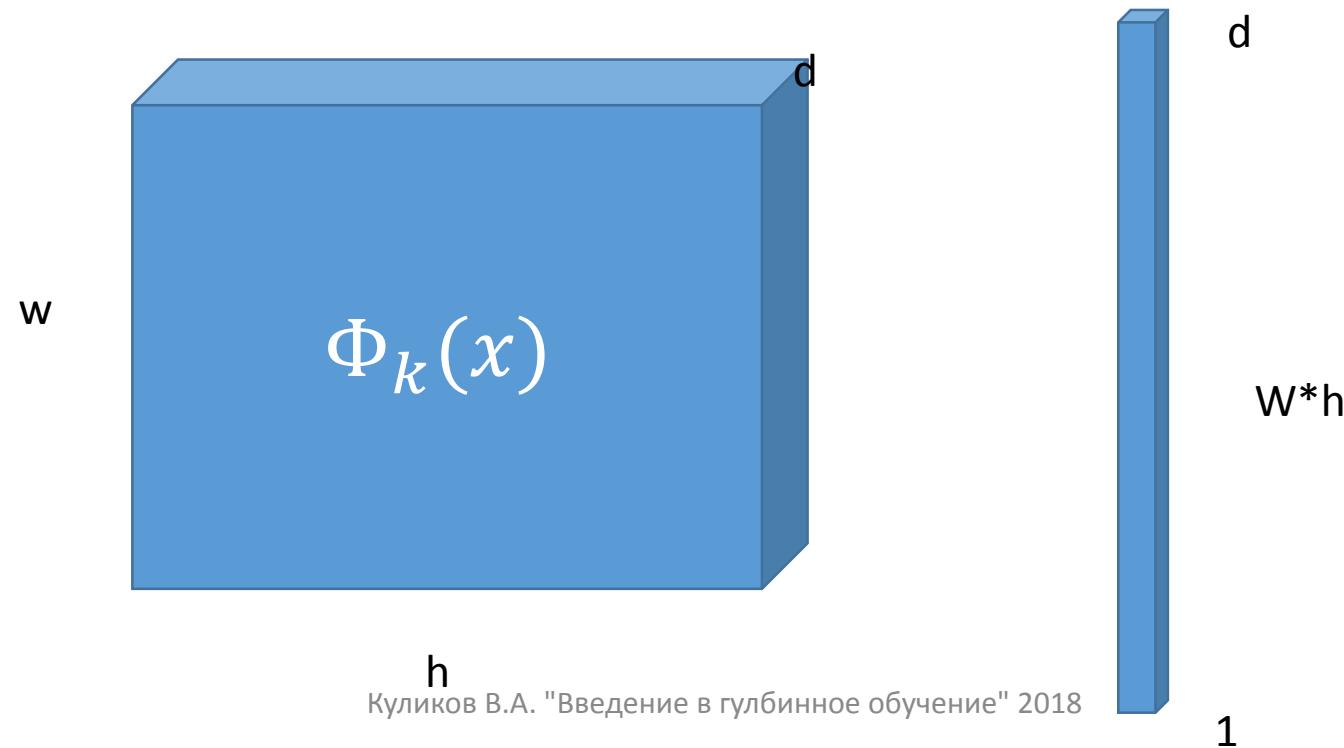


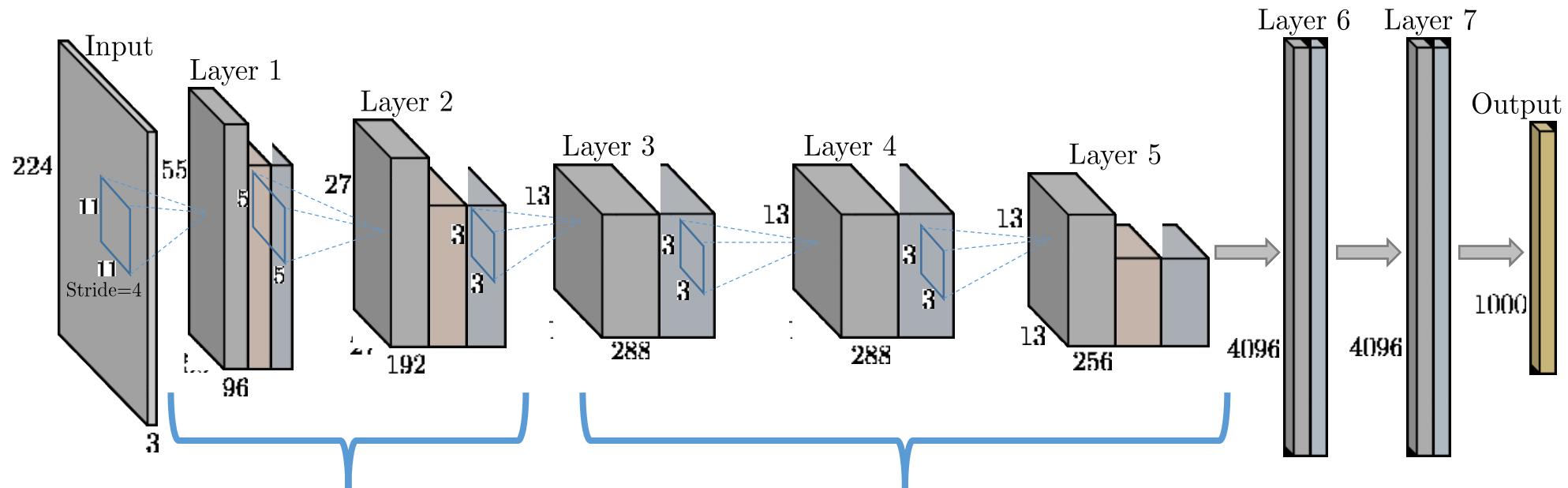
Куликов В.А. "Введение в глубинное обучение" 2018

[Gatys et al. 2015] 54

Матрица Грамма

$$\bullet G(x_1, \dots, x_n) = \begin{pmatrix} < x_1, x_1 > & \cdots & < x_n, x_1 > \\ \vdots & \ddots & \vdots \\ < x_n, x_1 > & \cdots & < x_n, x_n > \end{pmatrix}$$





$$L_{Style}(x, x_s) = \sum_{i \in Style} \|G(\Phi_i(x)) - G(\Phi_i(x_s))\|$$

$$L_{Content}(x, x_c) = \sum_{i \in Content} \|\Phi_i(x) - \Phi_i(x_c)\|$$

$$\hat{x} = \operatorname{argmin}_x L_{Style}(x, x_s) + \lambda L_{Content}(x, x_c)$$

Куликов В.А. "Введение в глубокое обучение" 2018

Стилизация изображений