

Глубинное обучение

Перенос обучения. Сегментация, локализация, детекция*

Даниил Водолазский

ВШЭ

28 июля 2021 г.



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Содержание

1 Перенос обучения (transfer learning)

Что выучивают нейросети
Крокодил learning

2 Семантическая сегментация

Анпулинг
Полносвёрточные сети
U-Net (2015)

3 Локализация

Bounding box
Примеры

4 Детекция объектов

R-CNN (2013)
IoU
Non-maximum suppression

Fast R-CNN
RoI pooling layer

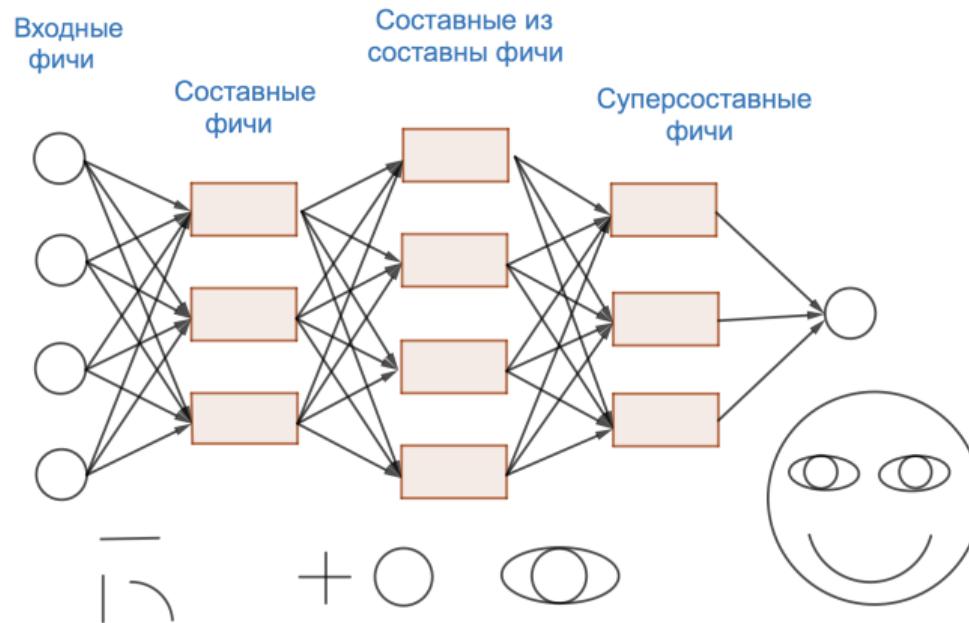
Faster R-CNN
RPN

Mask R-CNN
RoI Align

Перенос обучения (transfer learning)

- На практике тяжелые сети с нуля обучают только огромные компании.
- Это происходит из-за ограниченности ресурсов.
- Уже обученные архитектуры пытаются адаптировать под новые задачи, это называется **transfer learning**.

Что выучивают нейросети



Что выучивают нейросети



Layer 1



Layer 2



Layer 3



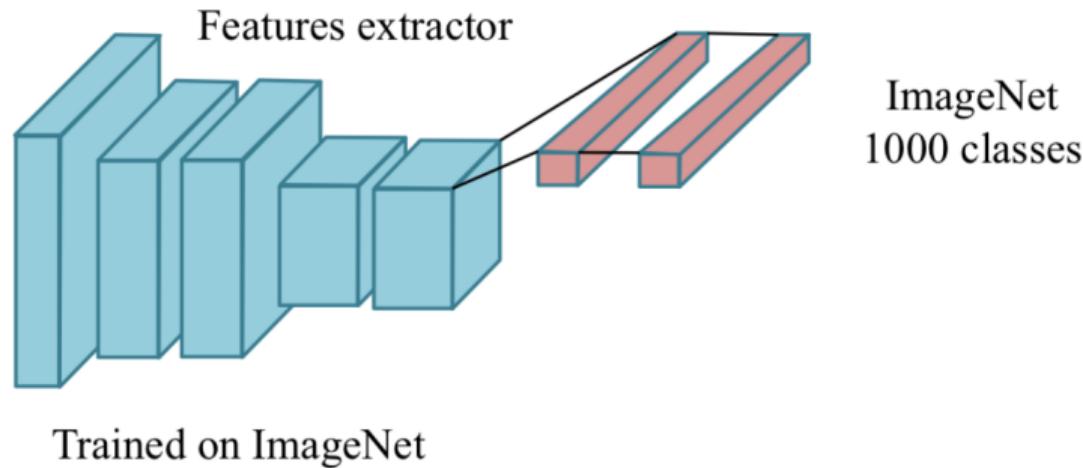
Layer 4



Layer 5

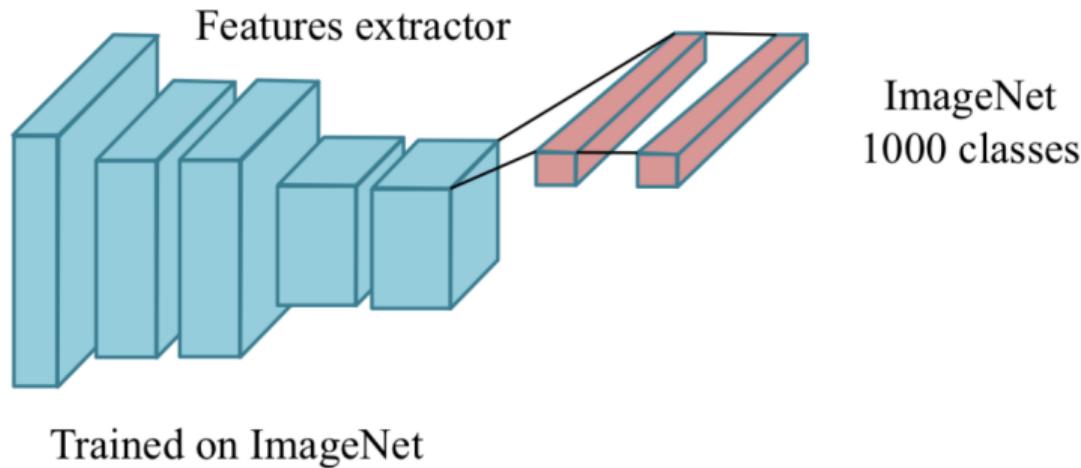
Перенос обучения (transfer learning)

- Глубокие сети извлекают из изображений сложные признаки, но для их обучения нужно много данных...



Перенос обучения (transfer learning)

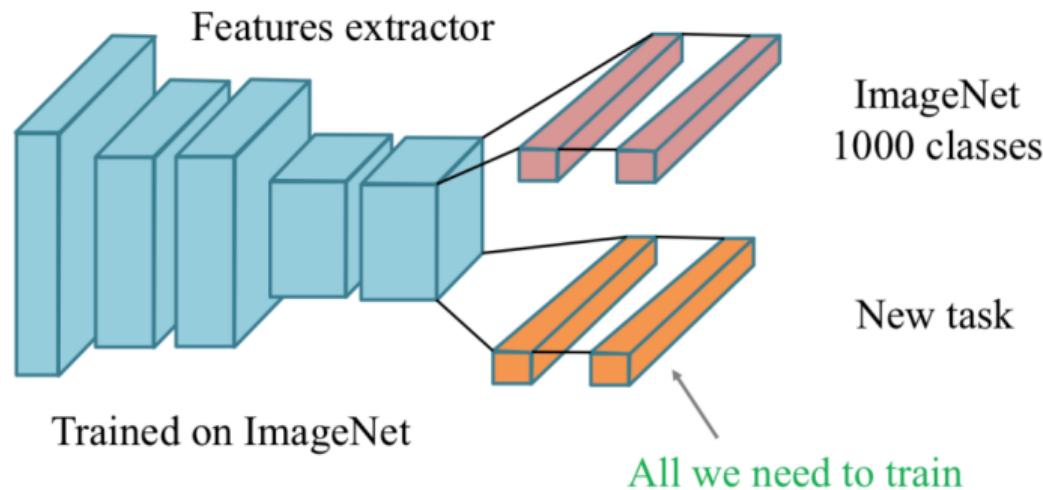
- Глубокие сети извлекают из изображений сложные признаки, но для их обучения нужно много данных...



- Давайте повторно использовать уже обученную сеть!

Перенос обучения (transfer learning)

- Глубокие сети извлекают из изображений сложные признаки, но для их обучения нужно много данных...

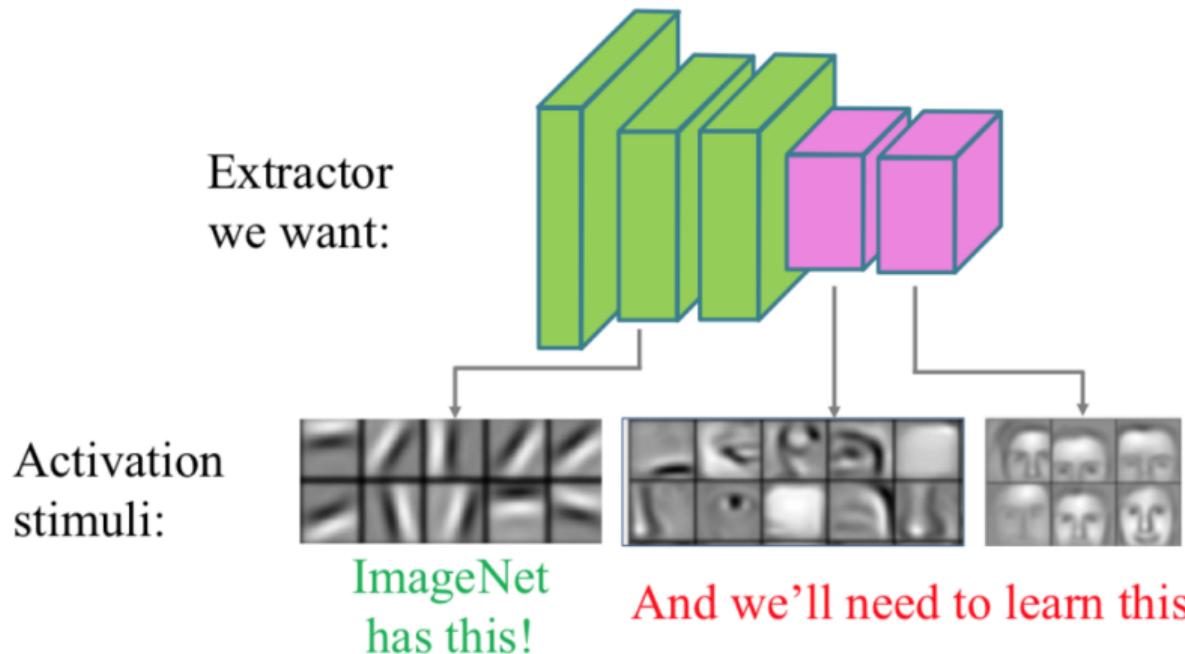


- Давайте повторно использовать уже обученную сеть!

Перенос обучения (transfer learning)

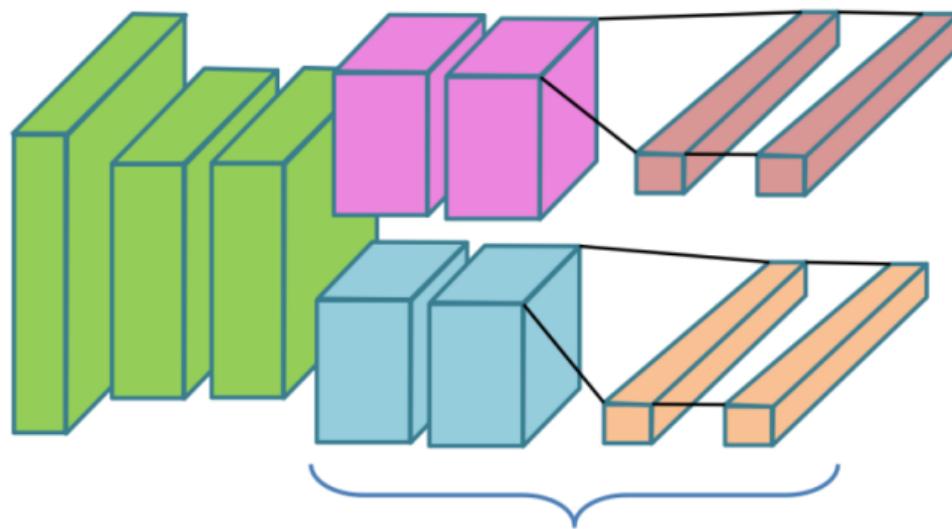
- Нужно меньше данных для обучения, так как нас интересуют лишь последние слои.
- Это работает, если наша задача похожа на ту, для которой обучалась используемая сетка.
- Например, если мы хотим распознавать эмоции, в датасете для нашей сетки должны были быть человеческие лица.

Перенос обучения (transfer learning)



Перенос обучения (transfer learning)

ImageNet features extractor



ImageNet
1000 classes

New task

Крокодил learning

cifar X



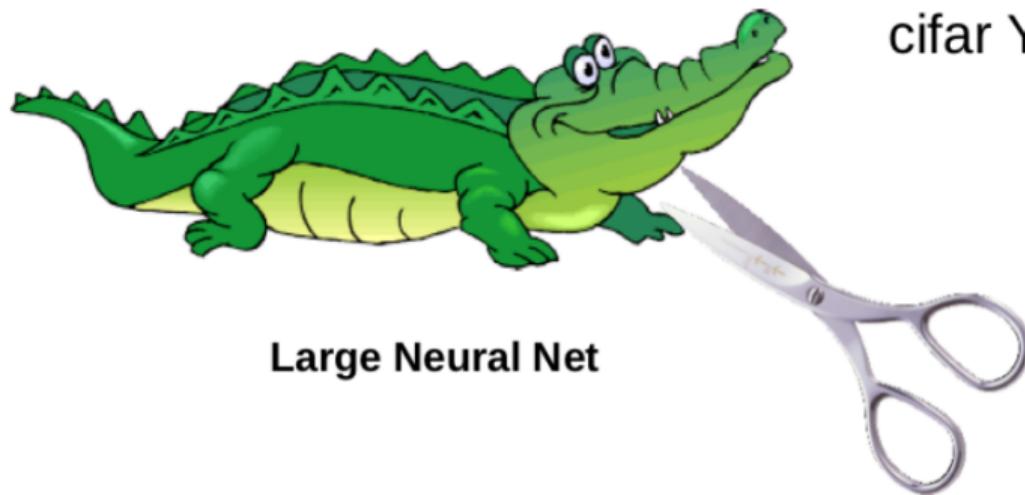
cifar Y

Large Neural Net

Крокодил learning

cifar X

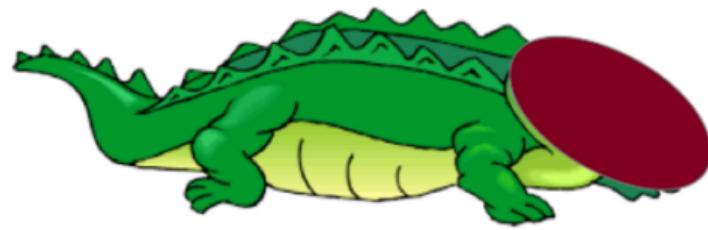
cifar Y



Крокодил learning

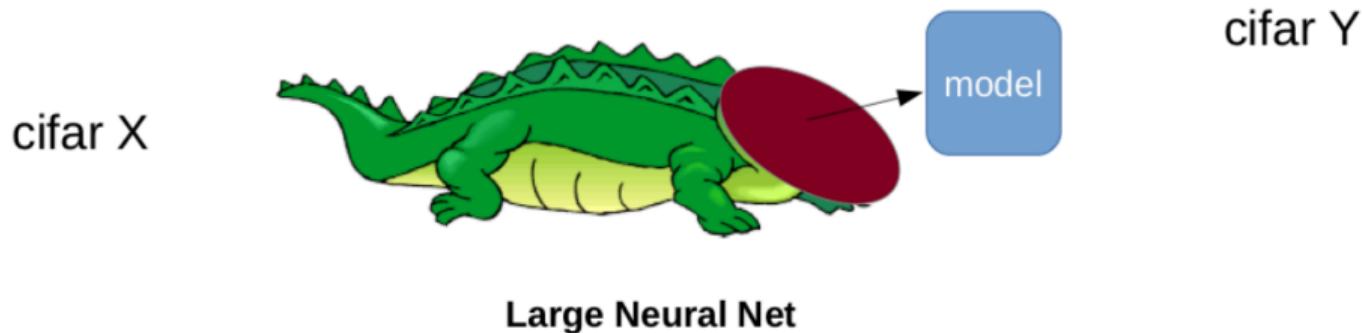
cifar X

cifar Y

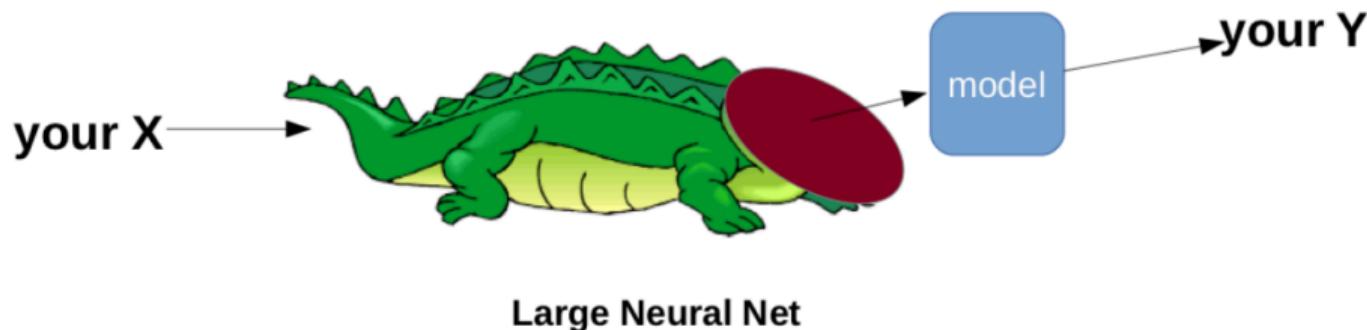


Large Neural Net

Крокодил learning



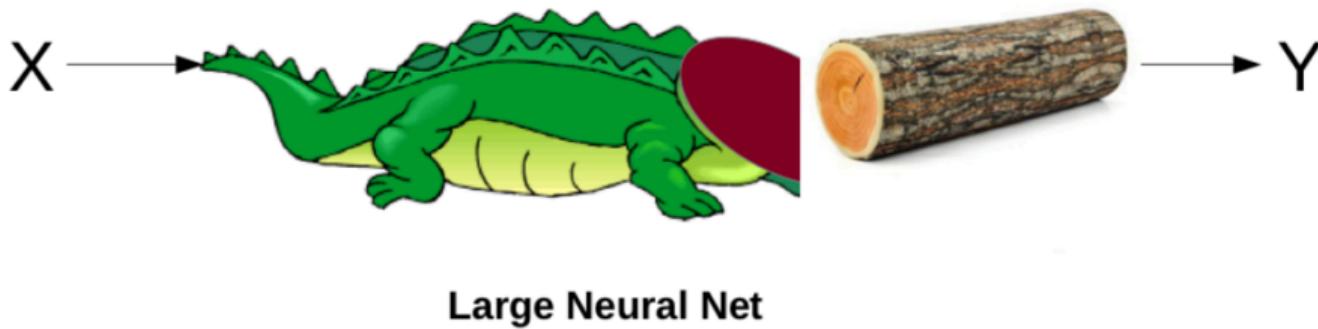
Крокодил learning



Крокодил learning

- Отрезали крокодилу голову.
- Используем тело крокодила как экстрактор признаков.
- Вместо головы крокодила можно прикрепить что угодно (как правило, это многослойный перцептрон).
- Можно брать даже случайный лес и бустинг.
- В экстракторе признаков веса модели обычно не дообучают, дообучение касается только новой головы крокодила.
- А если и дообучают, то с меньшим темпом обучения и только слои, находящиеся ближе к голове — иначе модель может забыть выученные ранее закономерности.

Крокодил learning



Зоопарки моделей

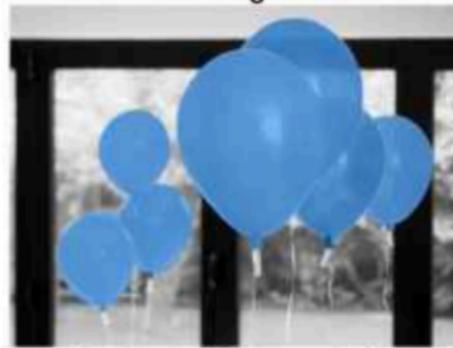
- Перед тем как решать задачу с нуля, убедитесь, что готового решения ещё нет.
- Зоопарк моделей в TorchVision: <https://pytorch.org/vision/stable/models.html>.
- Другой большой зоопарк: <https://modelzoo.co/>.

Основные задачи компьютерного зрения

Classification



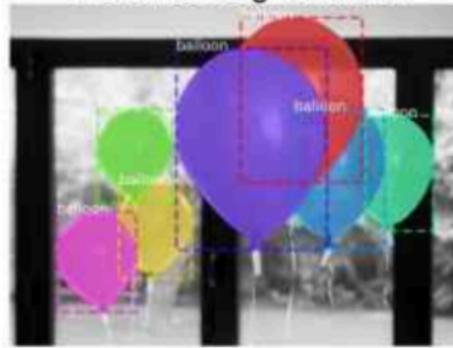
Semantic Segmentation



Object Detection



Instance Segmentation



Содержание

1 Перенос обучения (transfer learning)

Что выучивают нейросети
Крокодил learning

2 Семантическая сегментация

Анпулинг
Полносвёрточные сети
U-Net (2015)

3 Локализация

Bounding box
Примеры

4 Детекция объектов

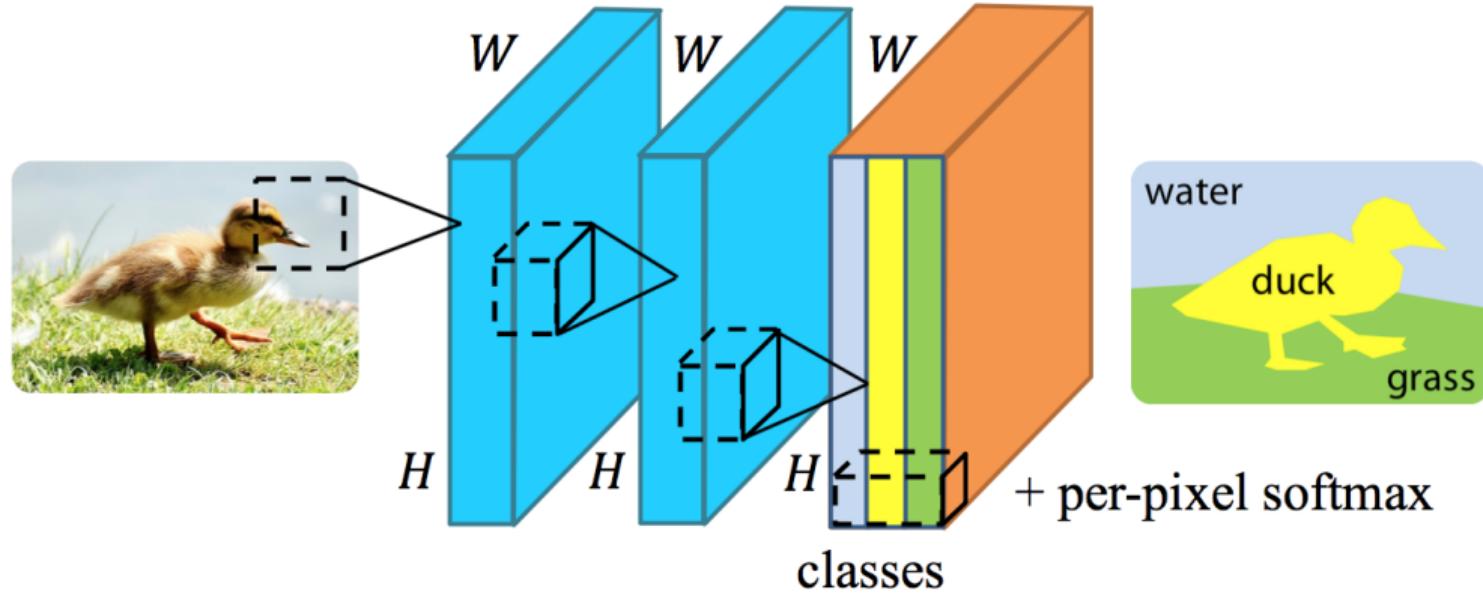
R-CNN (2013)
IoU
Non-maximum suppression

Fast R-CNN
RoI pooling layer

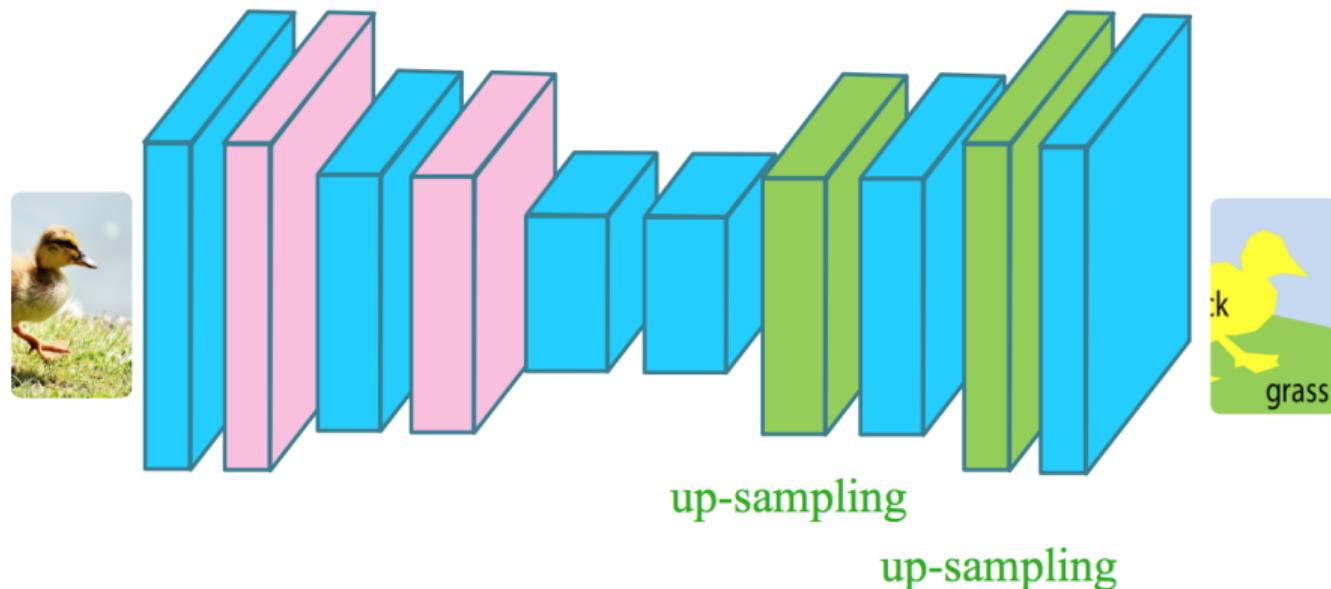
Faster R-CNN
RPN

Mask R-CNN
RoI Align

Семантическая сегментация

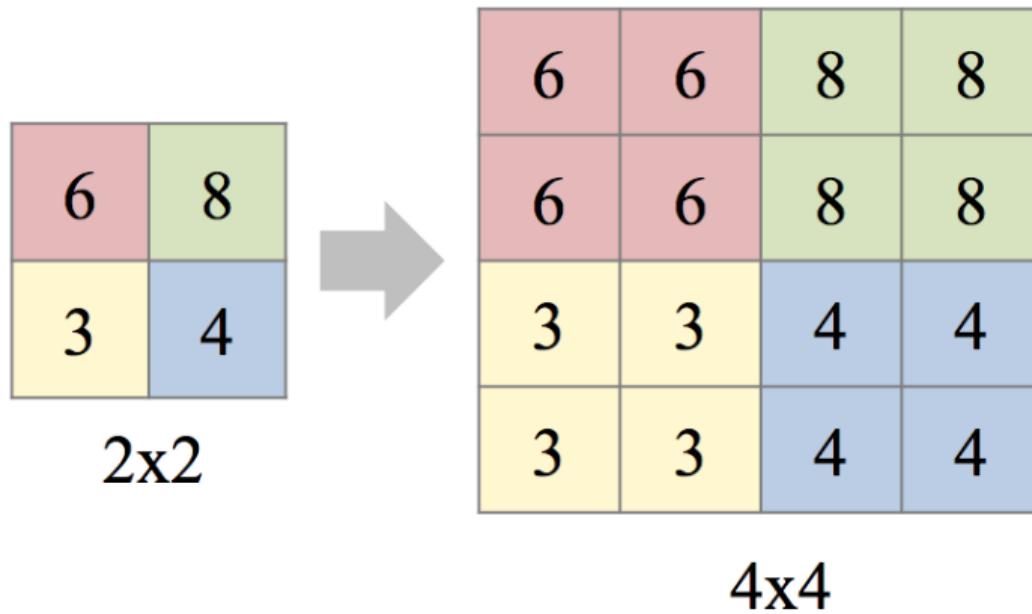


- Нам нужно научиться классифицировать каждый пиксель.
- Куча свёрток и попиксельный софтмакс без пулинга (наивный подход).

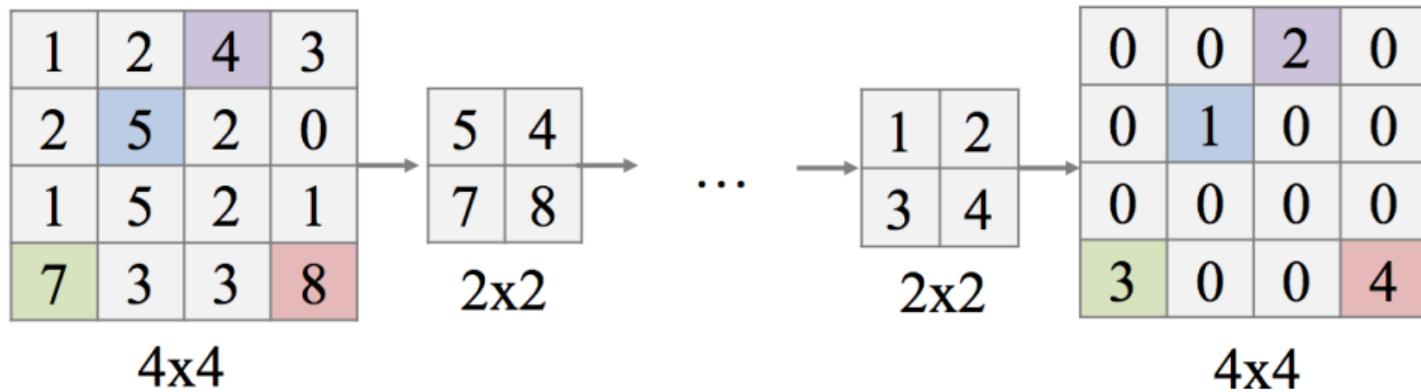


- Если захотим добавить пулинг, придётся делать анпулинг!

Анпулинг. Nearest neighbor unpooling

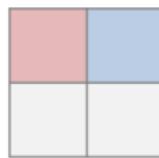


Анпулинг. Max unpooling

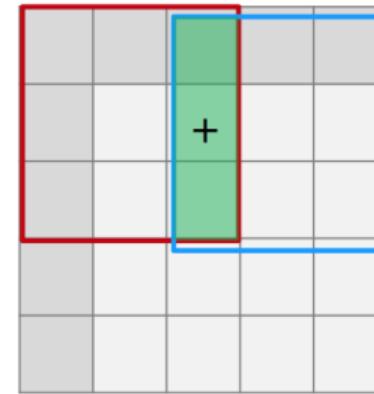


Анпулинг. Транспонированная свёртка

Input: 2x2



Input gives
weight for
filter

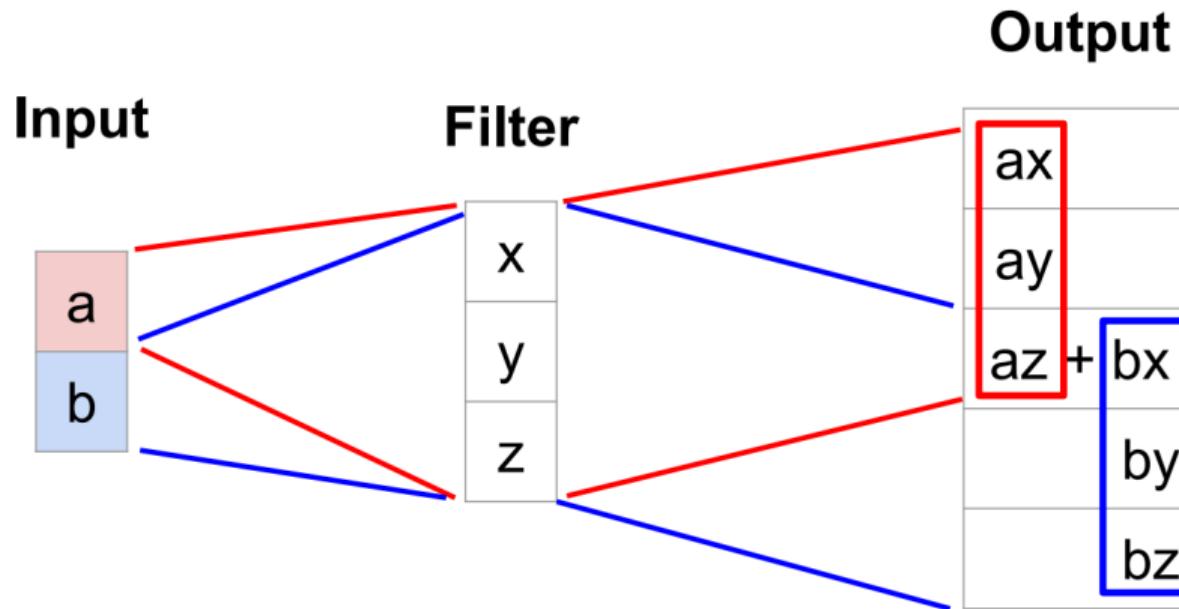


Stride: 2

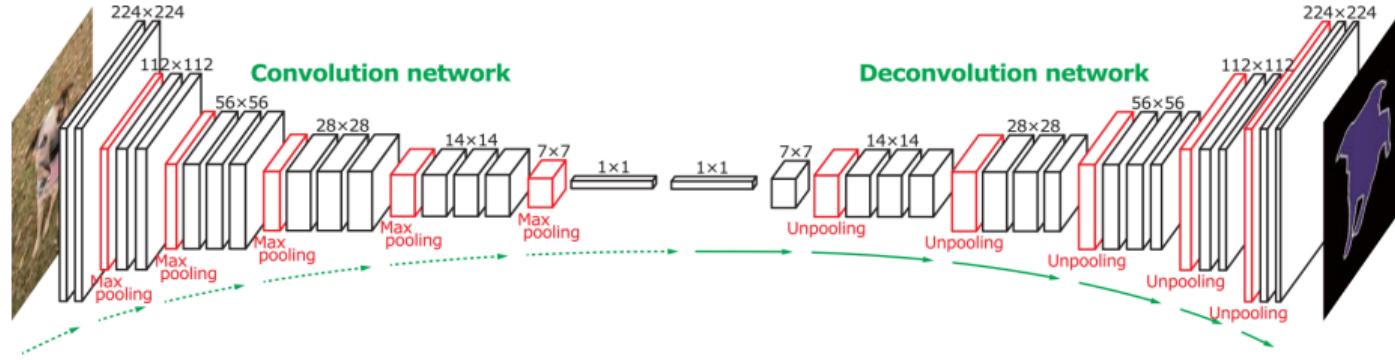
Output: 4x4

- Каждую клетку надо распаковать в 4 клетки \Rightarrow свёртка 3×3 со сдвигом 2.
- Веса такой свёртки обучаются во время тренировки сети.

Анпулинг. Транспонированная свёртка. Пример



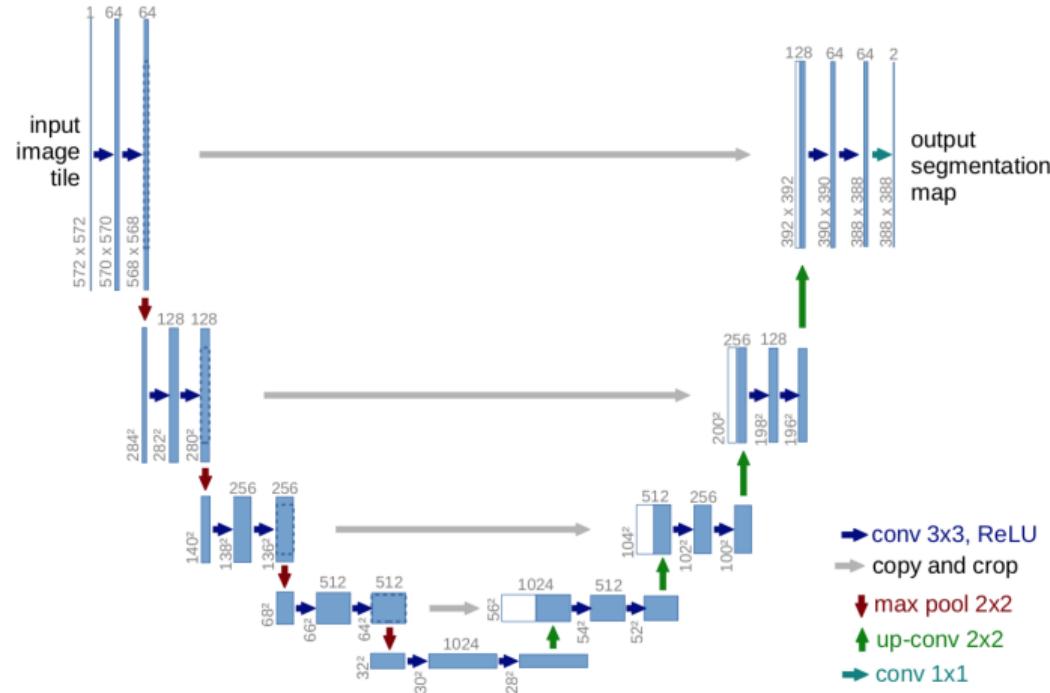
Полносвёрточные сети



- Перевели в скрытое представление, развернули, спрогнозировали.
- Всегда сохраняется локальная пространственная информация.
- Можно работать с изображениями любого размера (но не меньше определённого).
- Вместо полносвязных слоёв используют свёртки 1×1 : $c_{in} \times c_{out} \Rightarrow 1 \times 1 \times c_{in} \times c_{out}$.

U-Net (2015)

- Можно добавить связи между слоями, отражающими одинаковую абстракцию.



Содержание

1 Перенос обучения (transfer learning)

Что выучивают нейросети
Крокодил learning

2 Семантическая сегментация

Анпулинг
Полносвёрточные сети
U-Net (2015)

3 Локализация

Bounding box
Примеры

4 Детекция объектов

R-CNN (2013)
IoU
Non-maximum suppression

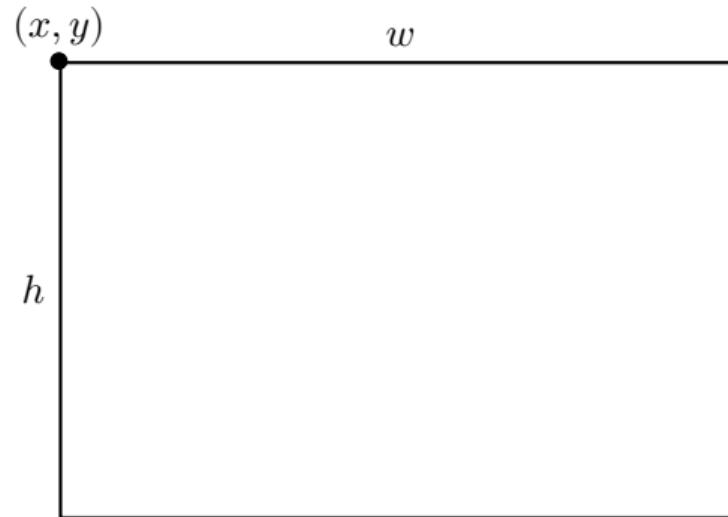
Fast R-CNN
RoI pooling layer

Faster R-CNN
RPN

Mask R-CNN
RoI Align

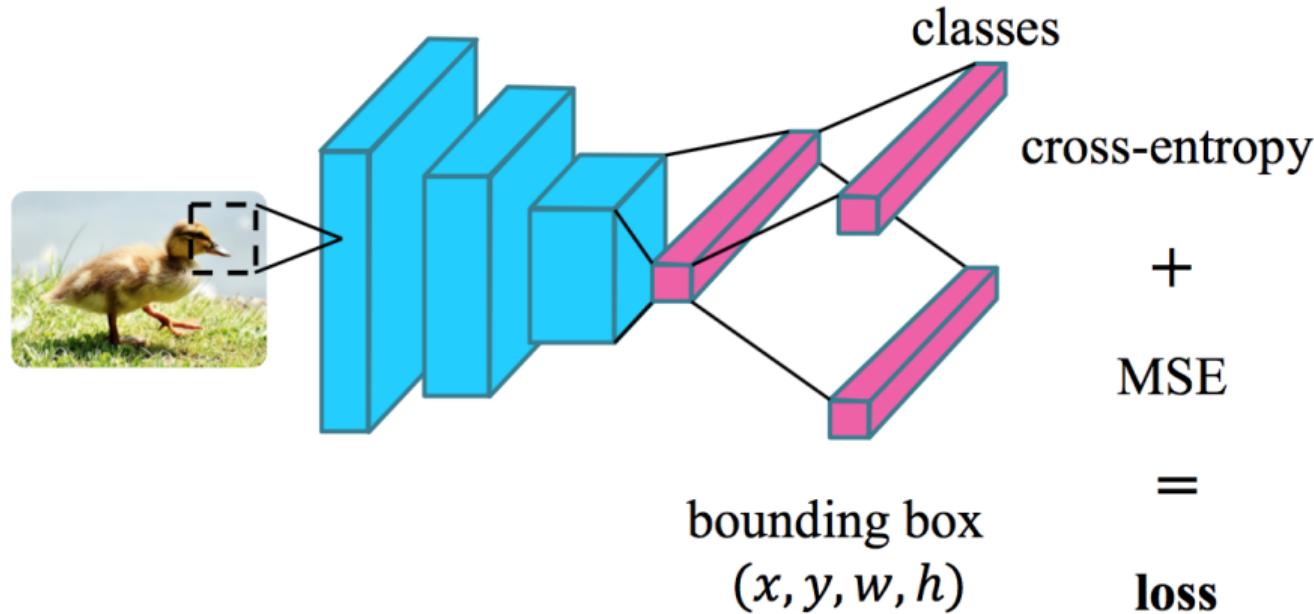
Bounding box

- Для локализации объекта нужно нащупать рамочку, в которой он находится.

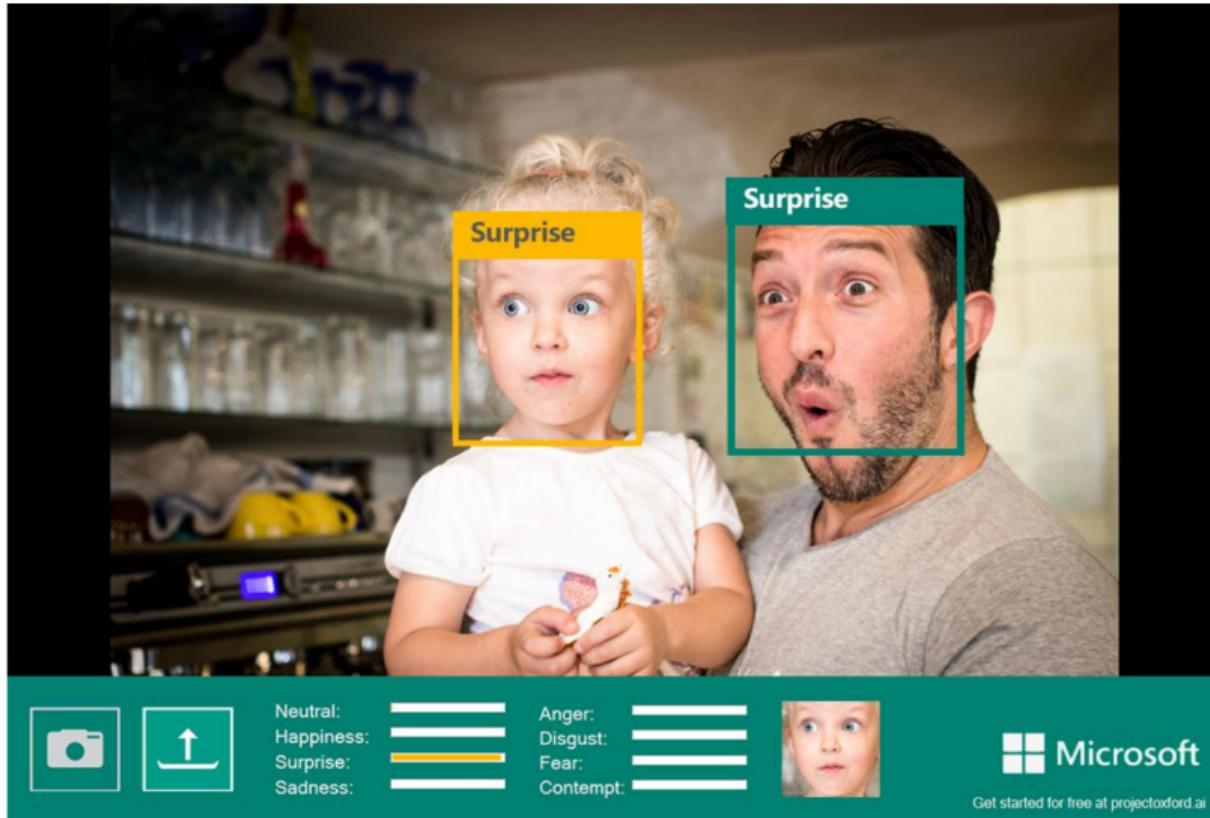


- Рамочка описывается параметрами (x, y, w, h) .

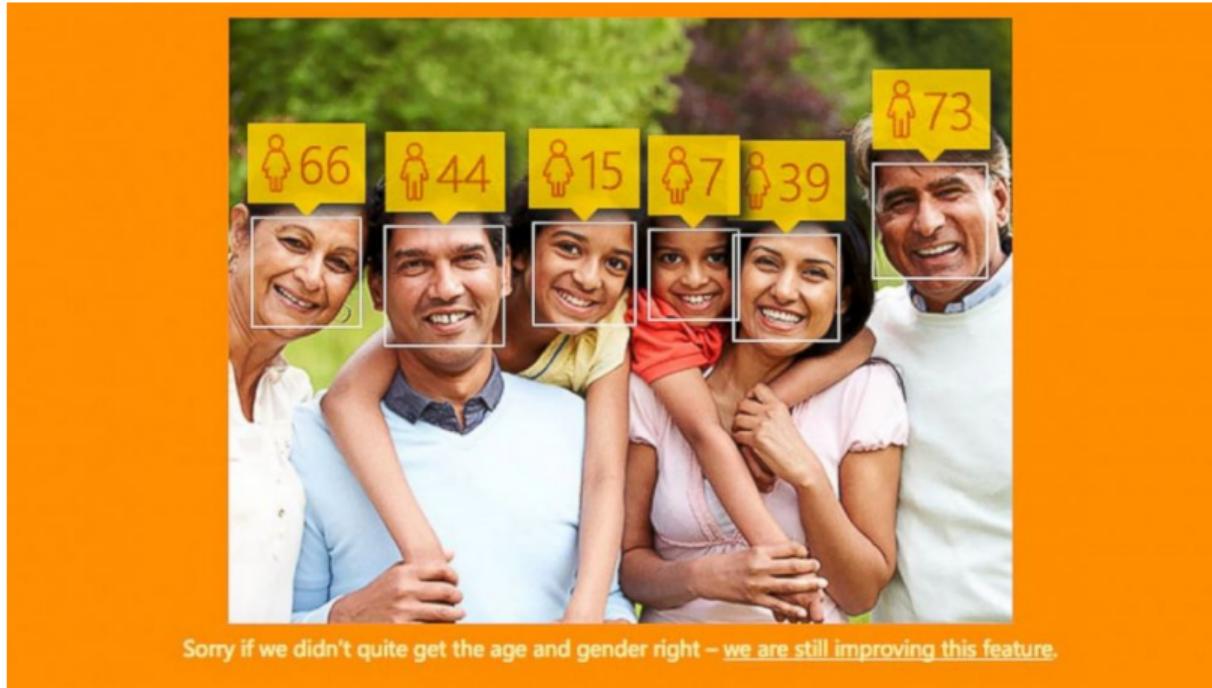
Bounding box



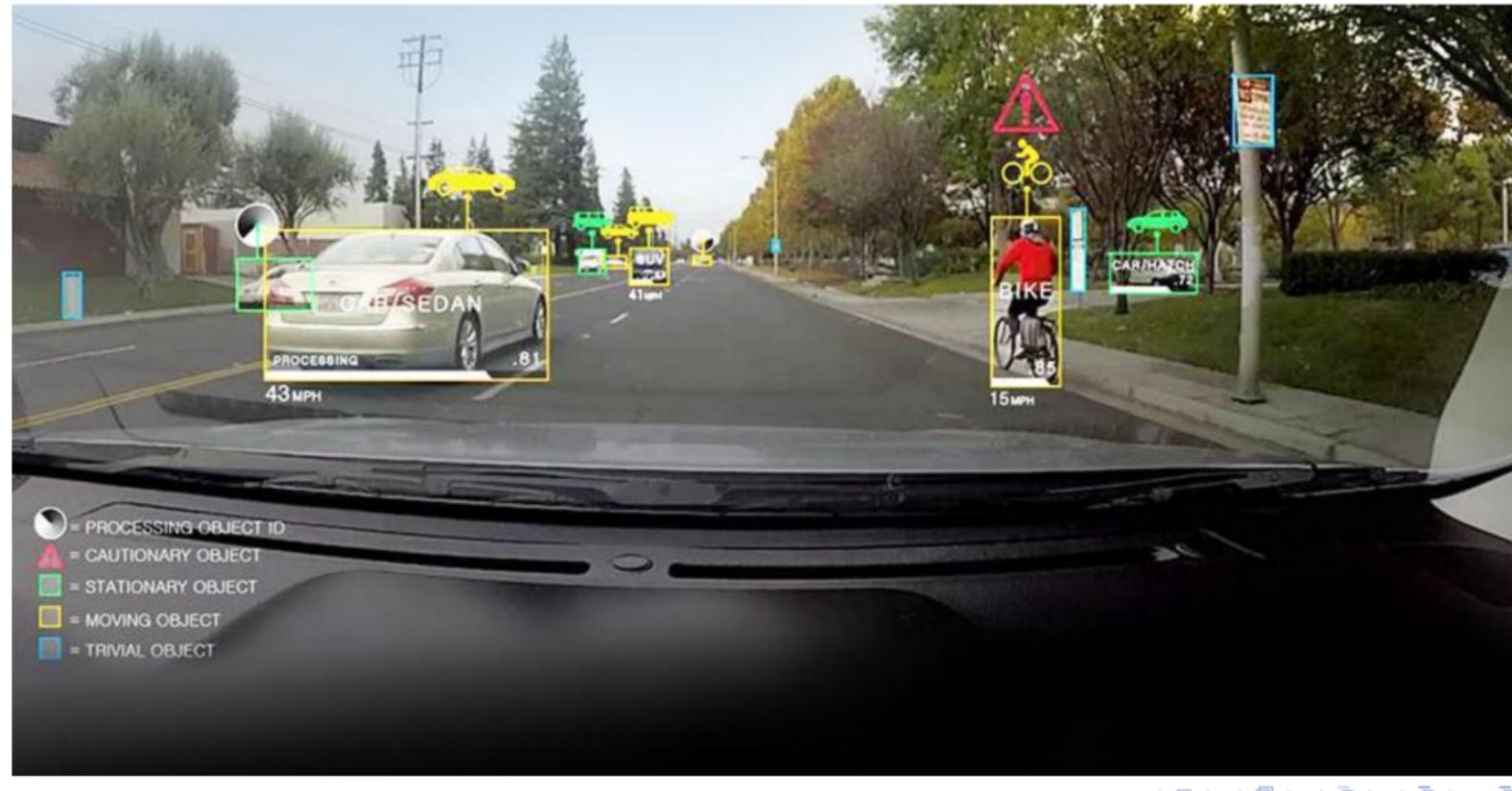
Примеры



Примеры



Примеры



Содержание

1 Перенос обучения (transfer learning)

Что выучивают нейросети
Крокодил learning

2 Семантическая сегментация

Анпулинг
Полносвёрточные сети
U-Net (2015)

3 Локализация

Bounding box
Примеры

4 Детекция объектов

R-CNN (2013)
IoU
Non-maximum suppression

Fast R-CNN

RoI pooling layer

Faster R-CNN

RPN

Mask R-CNN

RoI Align

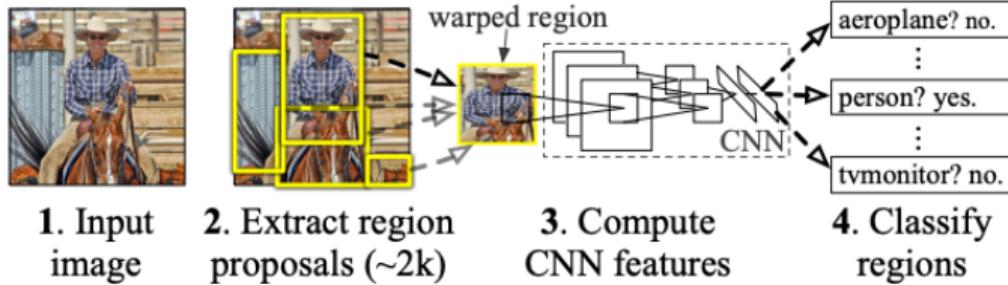
Детекция объектов. Краткая история

- Вначале подход был следующим — мы вручную придумываем признаки из картинки, потом каким-либо алгоритмом придумываем, как выделить на картинке объект.
- Например, мы хотим научиться выделять велосипед — нам нужно сначала научиться выделять части велосипеда, потом скользящим окном определять область, где больше всего частей было выделено.
- Такой подход хороший — мы много раз используем один и тот же алгоритм на одной картинке, а это непрактично.
- Приход нейронных сетей дал возможность использовать «нейронные знания» об изображении.

Нельзя просто взять CNN и решить задачу детекции!

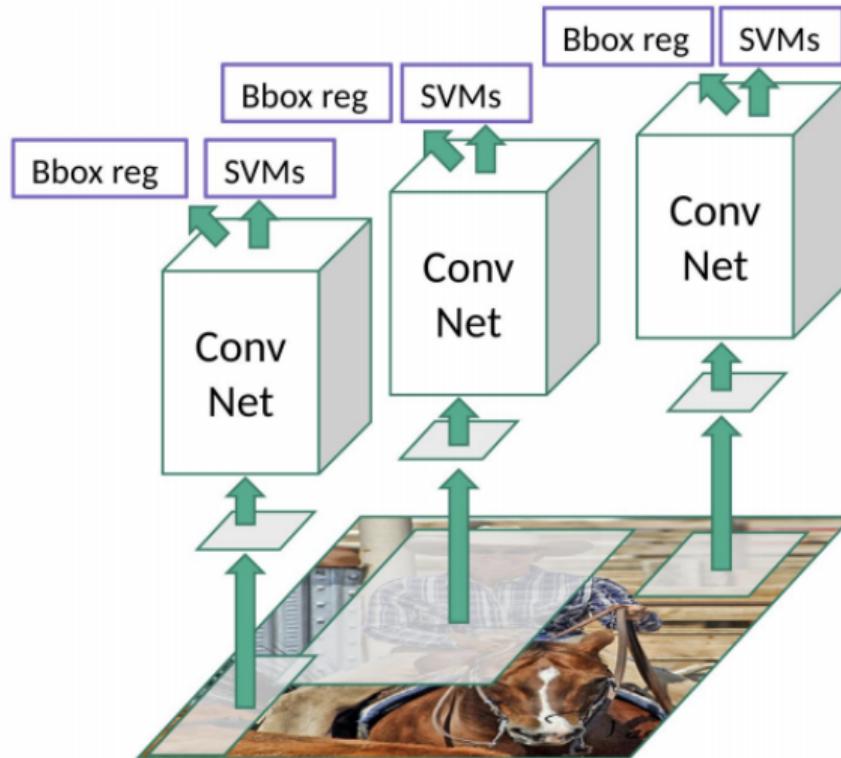
- Количество объектов неизвестно заранее — это переменная величина.
- Комбинаторная сложность перебора всех возможных рамок — $O(h^2w^2)$.
- Что делать с очень похожими рамками?
- Как измерить качество модели?
- Насколько быстро работает алгоритм?

R-CNN: *Regions with CNN features*



- ① Находить регионы-кандидаты алгоритмом **selective search**:
 - ① Generate initial sub-segmentation, we generate many candidate regions.
 - ② Use greedy algorithm to recursively combine similar regions into larger ones.
 - ③ Use the generated regions to produce the final candidate region proposals.
- ② Привести регионы к размеру 227×227 .
- ③ Получить вектор изображения размером 4096.
- ④ Классифицировать векторы с SVM.
- ⑤ Линейная регрессия для определения рамок.

R-CNN (2013)



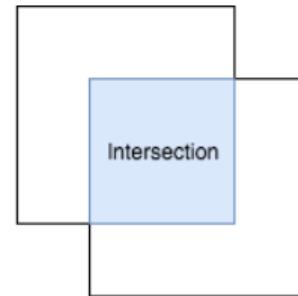
Особенности:

- Использование 16-пиксельной границы на первой стадии для добавления контекста.
- IoU, чтобы отбирать гипотезы.
- Non-maximum suppression.

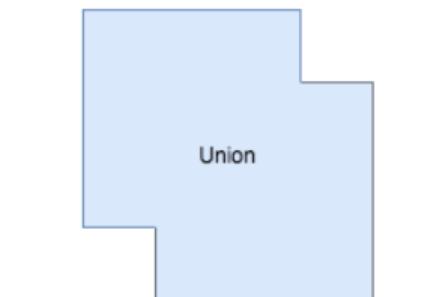
Недостатки:

- Всё ещё требует много времени, чтобы сгенерировать и классифицировать 2000 регионов.
- Невозможно использовать в реальном времени, поскольку обработка одного изображения занимает 47 секунд.
- Алгоритм selective search статический. Обучения на этой стадии не происходит. Это может привести к генерации неудачных кандидатов.

IoU

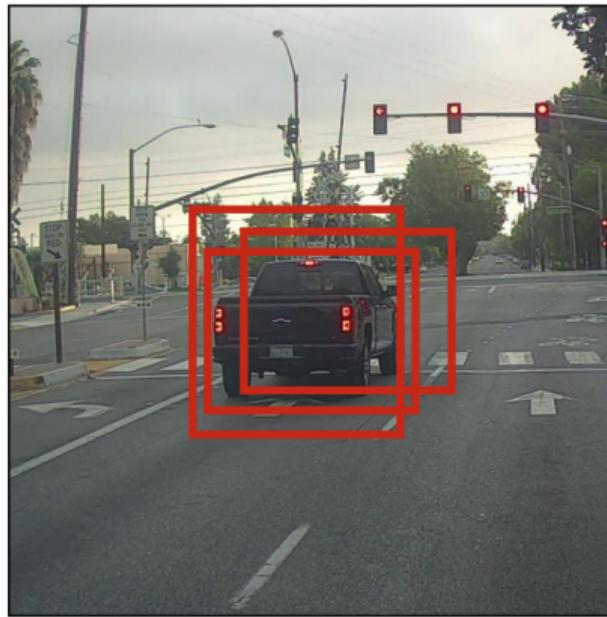


IoU =



Non-maximum suppression

Before non-max suppression



Non-Max
Suppression



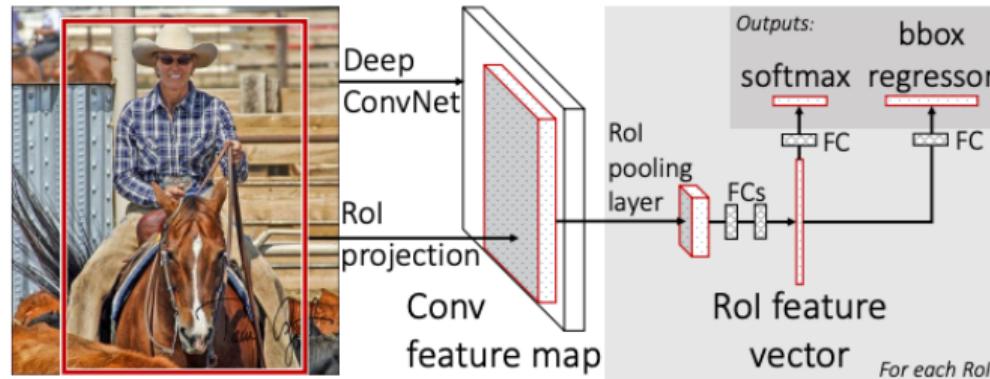
After non-max suppression



Algorithm 1 Non-Max Suppression

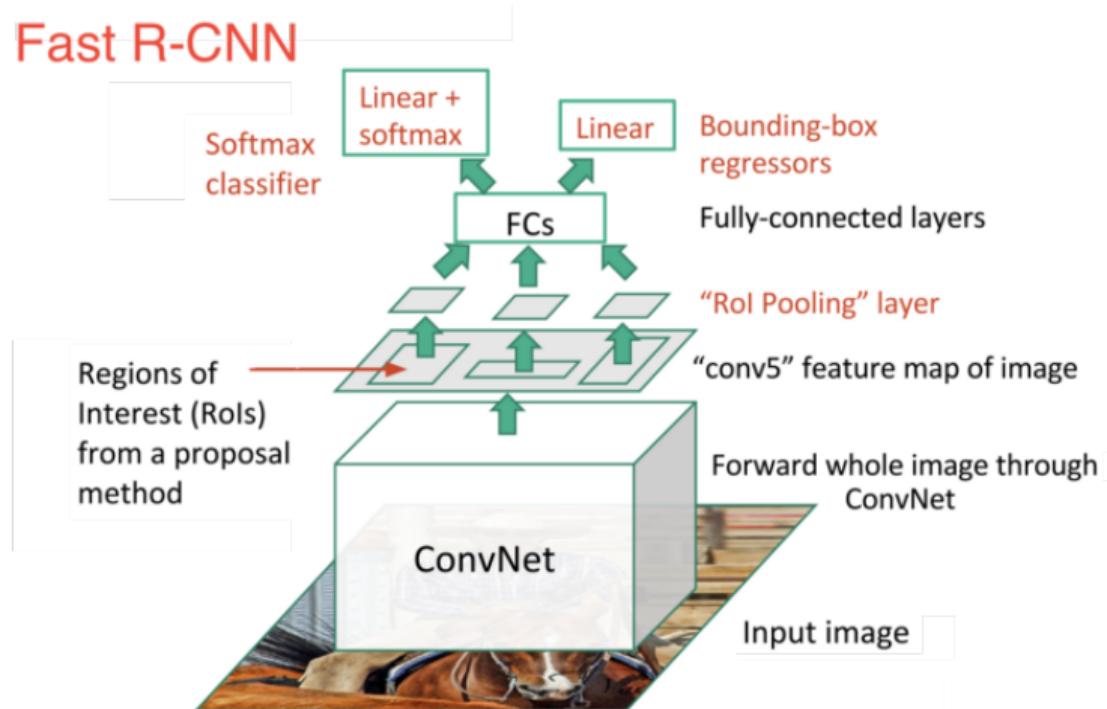
```
1: procedure NMS( $B, c$ )
2:    $B_{nms} \leftarrow \emptyset$  Initialize empty set
3:   for  $b_i \in B$  do  $\Rightarrow$  Iterate over all the boxes
4:      $discard \leftarrow \text{False}$  Take boolean variable and set it as false. This variable indicates whether  $b(i)$  should be kept or discarded
5:     for  $b_j \in B$  do Start another loop to compare with  $b(i)$ 
6:       if same( $b_i, b_j$ )  $> \lambda_{nms}$  then If both boxes having same IOU
7:         if score( $c, b_j$ )  $>$  score( $c, b_i$ ) then
8:            $discard \leftarrow \text{True}$  Compare the scores. If score of  $b(i)$  is less than that of  $b(j)$ ,  $b(i)$  should be discarded, so set the flag to True.
9:         if not  $discard$  then Once  $b(i)$  is compared with all other boxes and still the discarded flag is False, then  $b(i)$  should be considered. So add it to the final list.
10:         $B_{nms} \leftarrow B_{nms} \cup b_i$ 
11:    return  $B_{nms}$  Do the same procedure for remaining boxes and return the final list
```

Fast R-CNN



- 1 Используется всё изображение для получения карты признаков.
- 2 Регионы-кандидаты ищутся через selective search.
- 3 Гипотезам ставятся в соответствие координаты.
- 4 Классификация каждой гипотезы.

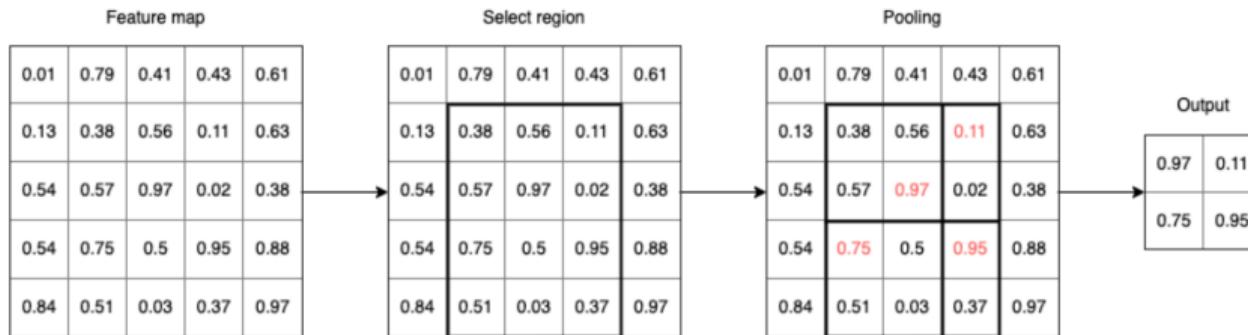
Fast R-CNN



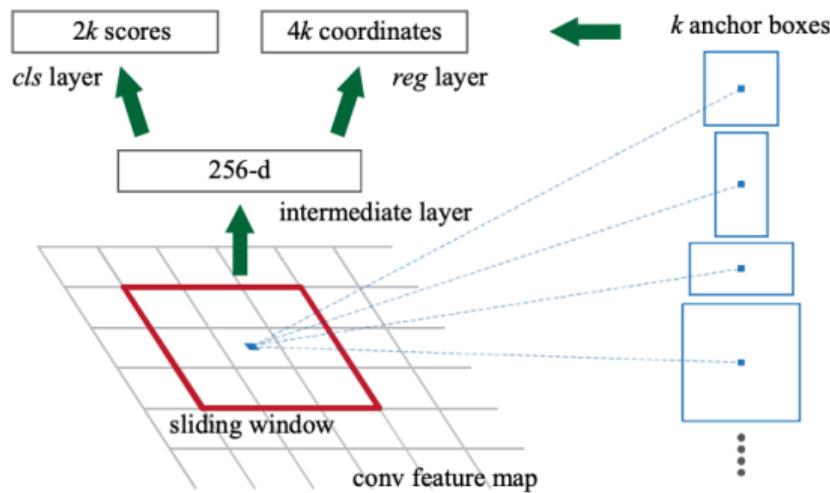
Особенности:

- Слой пулинга RoI (Regions of Interest).
- Софтмакс вместо классификаторов SVM.
- Multi-task loss.

RoI pooling layer

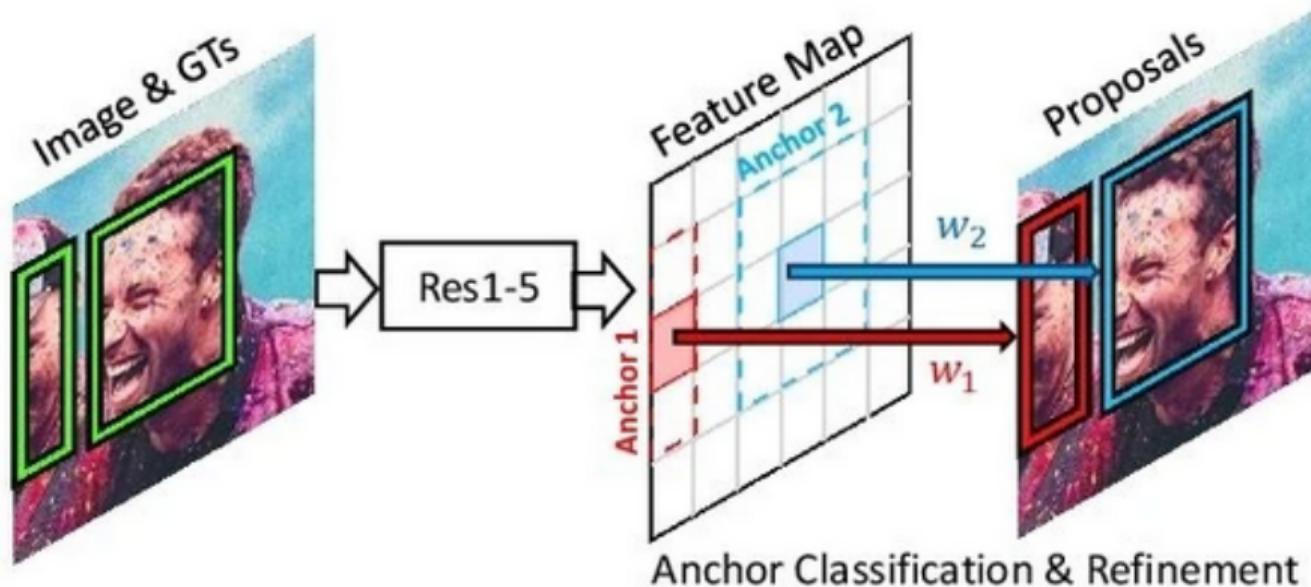


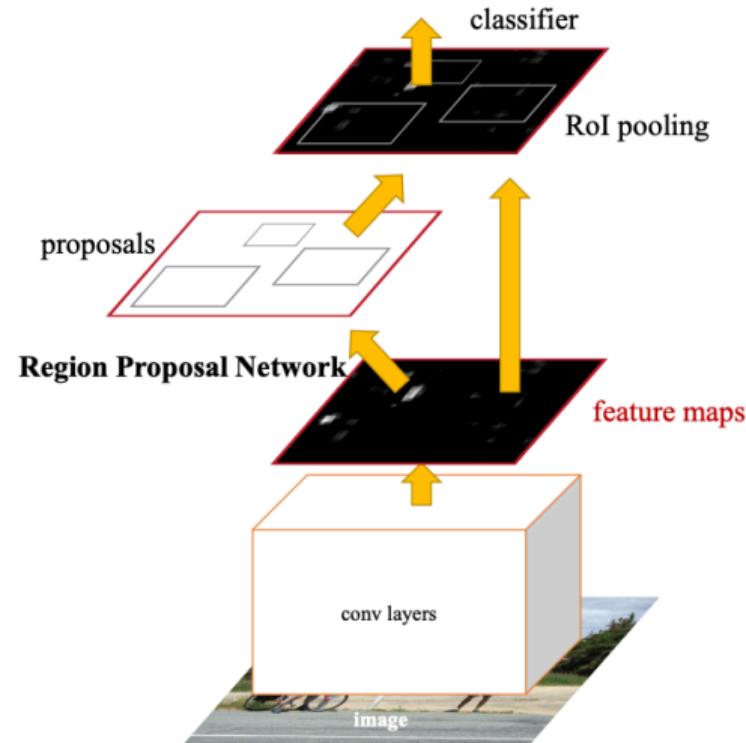
Faster R-CNN



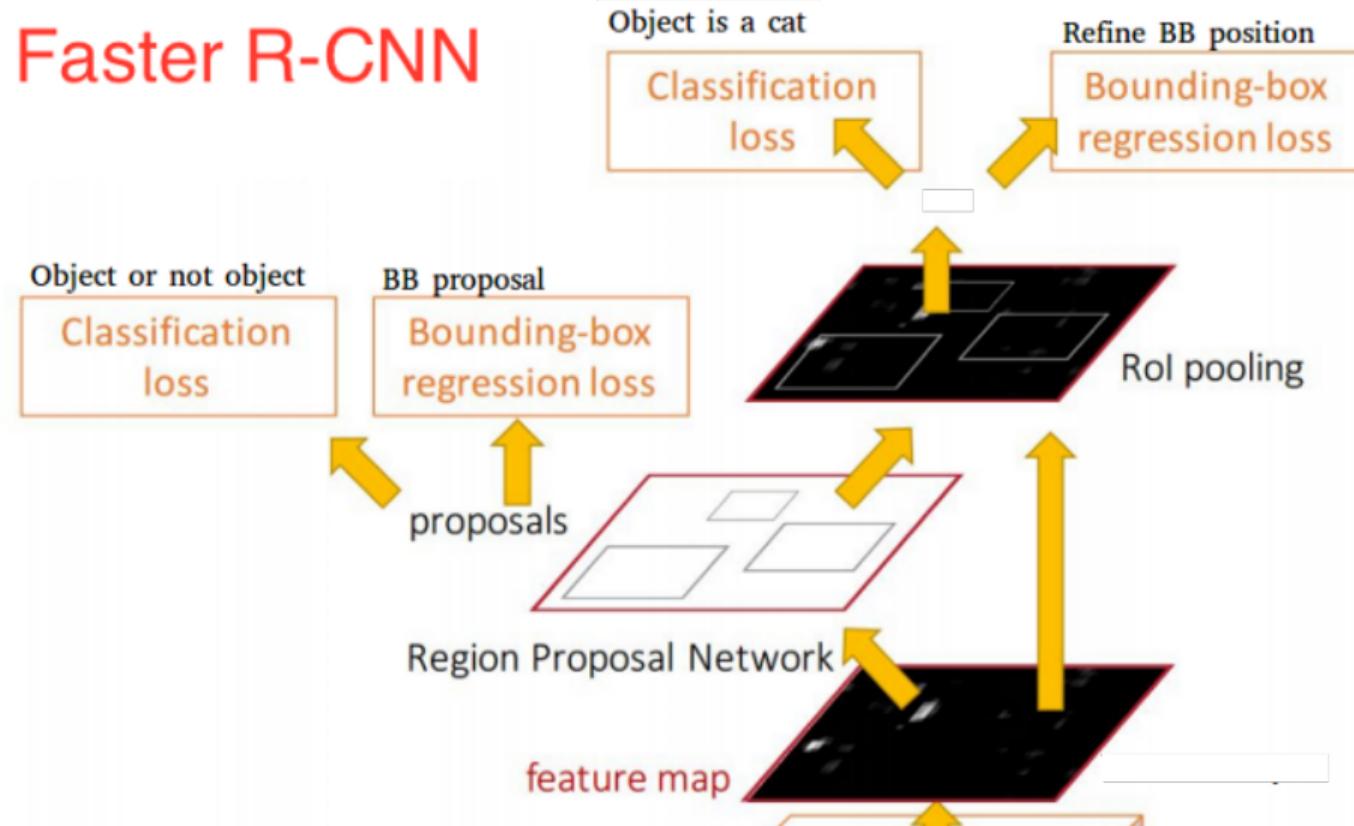
- Карта признаков извлекается через RPN (Region Proposal Network) вместо selective search.
- RoI-pooling (уже видели).

Faster R-CNN





Faster R-CNN



Mask R-CNN

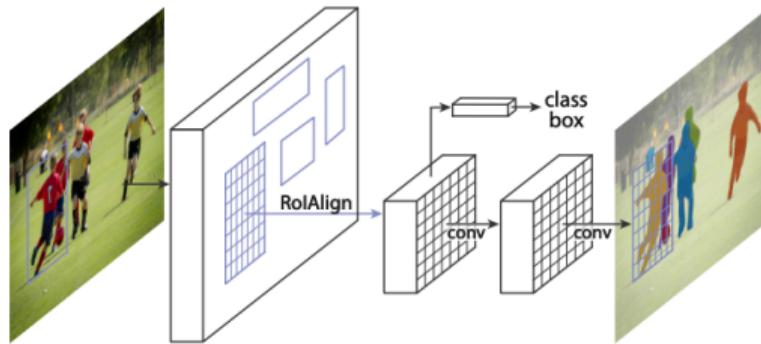
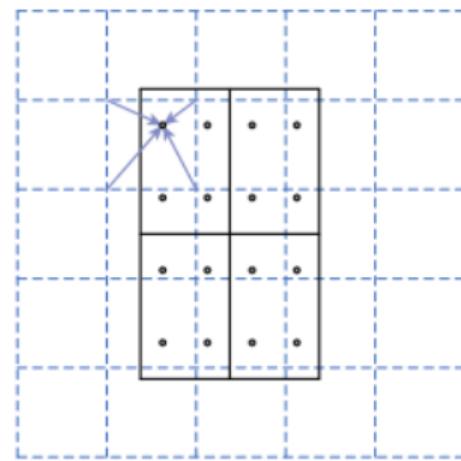


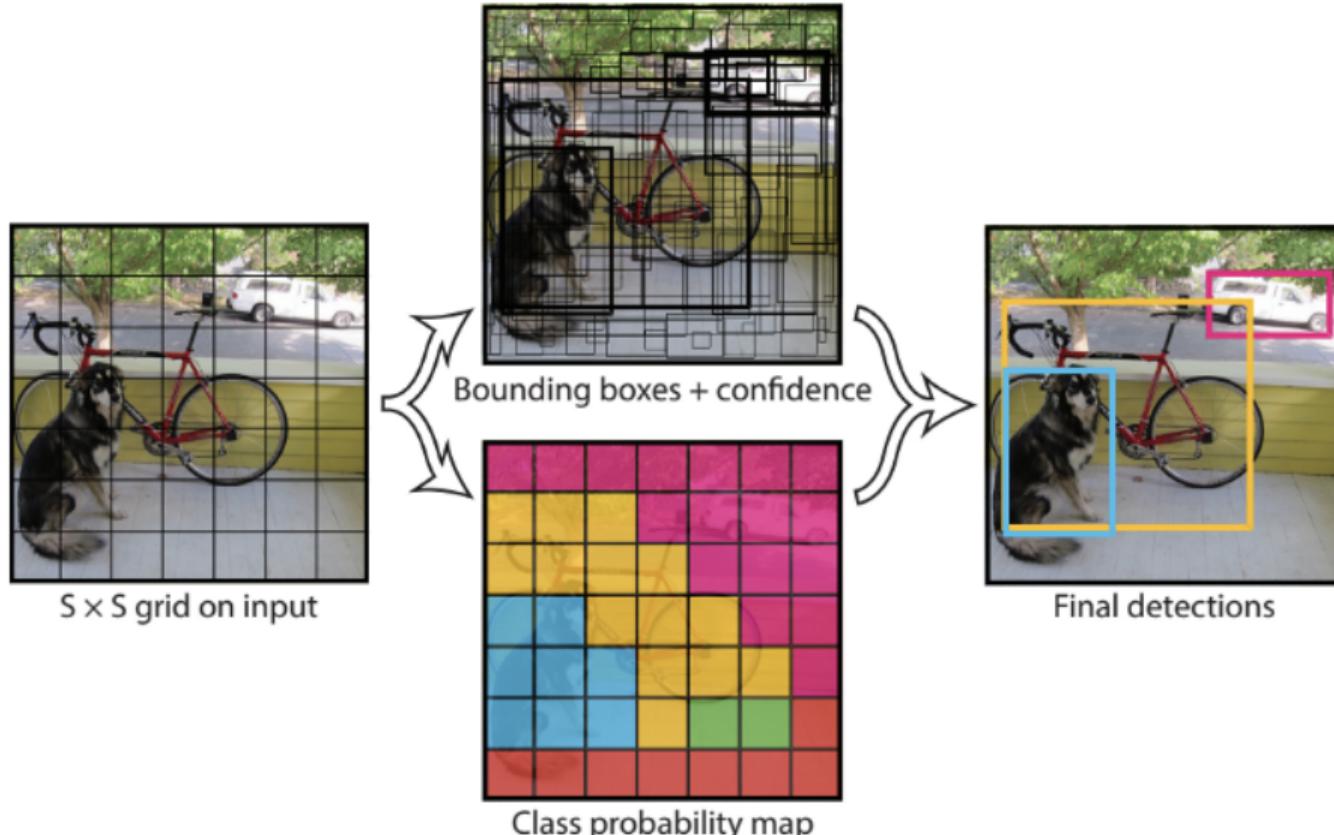
Figure 1. The **Mask R-CNN** framework for instance segmentation.

- RPN.
- RoI-Align.
- CNN-слои.
- Классификация, регрессия на координаты боксов и сегментация маски для каждой гипотезы.

RoI-Align

Region of Interest Align, or **RoI-Align**, is an operation for extracting a small feature map from each RoI in detection and segmentation based tasks. It removes the harsh quantization of RoI Pool, properly aligning the extracted features with the input. To avoid any quantization of the RoI boundaries or bins (using $\frac{x}{16}$ instead of $[\frac{x}{16}]$), RoIAlign uses bilinear interpolation to compute the exact values of the input features at four regularly sampled locations in each RoI bin, and the result is then aggregated (using max or average).





Содержание

1 Перенос обучения (transfer learning)

Что выучивают нейросети
Крокодил learning

2 Семантическая сегментация

Анпулинг
Полносвёрточные сети
U-Net (2015)

3 Локализация

Bounding box
Примеры

4 Детекция объектов

R-CNN (2013)
IoU
Non-maximum suppression

Fast R-CNN
RoI pooling layer

Faster R-CNN
RPN

Mask R-CNN
RoI Align

Что узнали

- Какие еще бывают задачи в компьютерном зрении: сегментация, локализация, детекция.
- Как использовать полносвёрточные нейронные сети.
- Подходы к детекции объектов.