

Машинное обучение (Machine Learning)

Сегментация и детекция изображений

Уткин Л.В.

Санкт-Петербургский политехнический университет Петра Великого



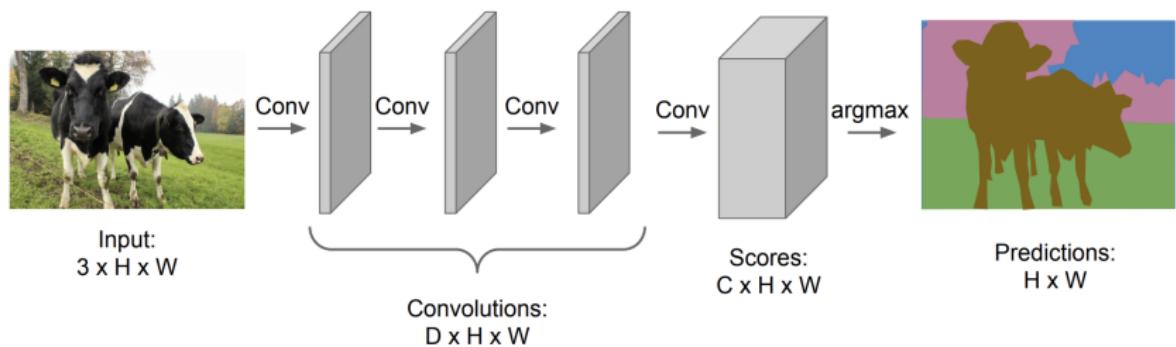
Semantic Segmentation, классификация на уровне пикселов

- Просто соединить набор сверточных слоев, где захватываются локальные элементы в изображениях.
- CNN может кодировать изображение как компактное представление его содержимого
- Отображение между входным изобр. и соответствующим выходом сегментации через иерархическое представление

Patch-by-patch scanning - Основная идея

- Метка класса определяется для каждого пикселя: задача рассматривается как задача “попиксельной” классификации
- Окно (patch) с центром в каждом пикселе подается на вход сверточной сети для генерации его класса
- Точность сегментации повышается с увеличением размера окна, так как оно охватывает больше контекстной информации
- Главный недостаток - большое перекрытие соседних окон и большой объем избыточных вычислений.

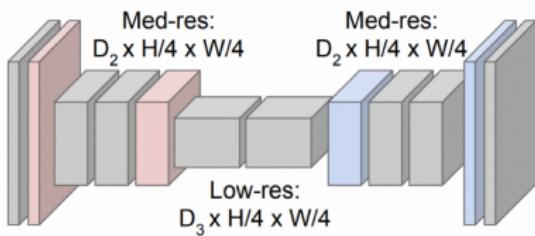
Semantic Segmentation, классификация на уровне пикселов



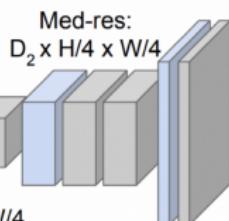
Semantic Segmentation, классификация на уровне пикселов



Input:
 $3 \times H \times W$



High-res:
 $D_1 \times H/2 \times W/2$



High-res:
 $D_1 \times H/2 \times W/2$

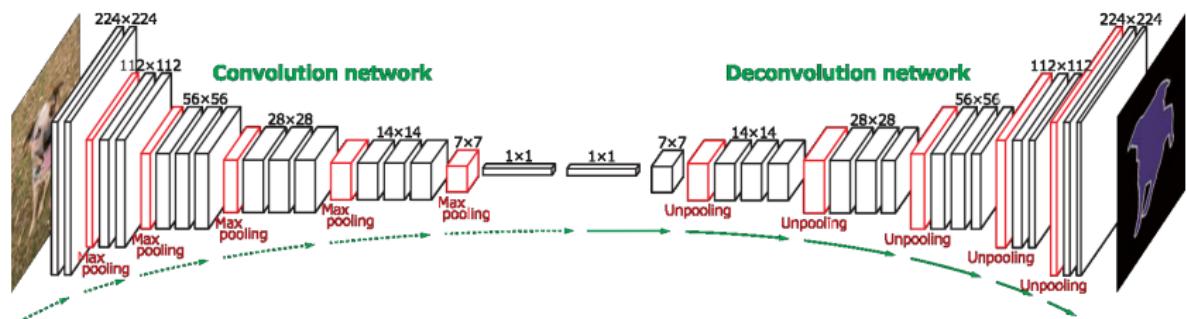


Predictions:
 $H \times W$

Up sampling
 Down sampling

Deconvolution network (Deconvnet)

H. Noh S. Hong B. Han, Learning Deconvolution Network for Semantic Segmentation

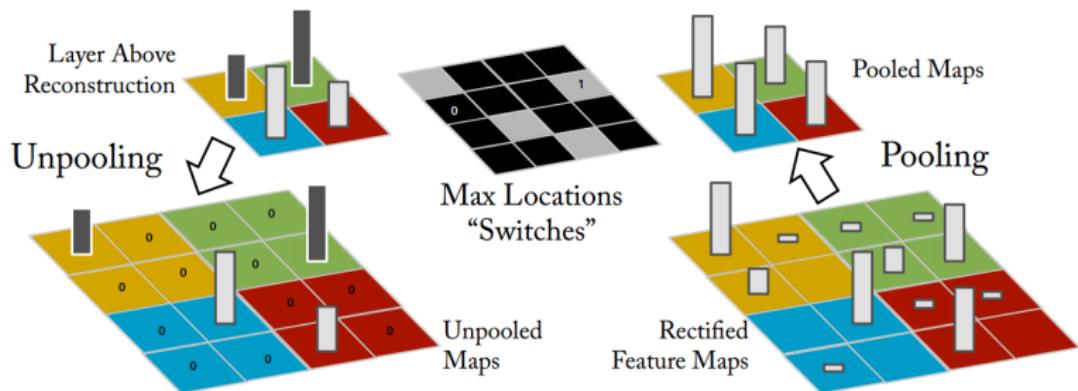


Deconvolution network

Реализует две основные процедуры

- ① Unpooling
- ② Deconvolution

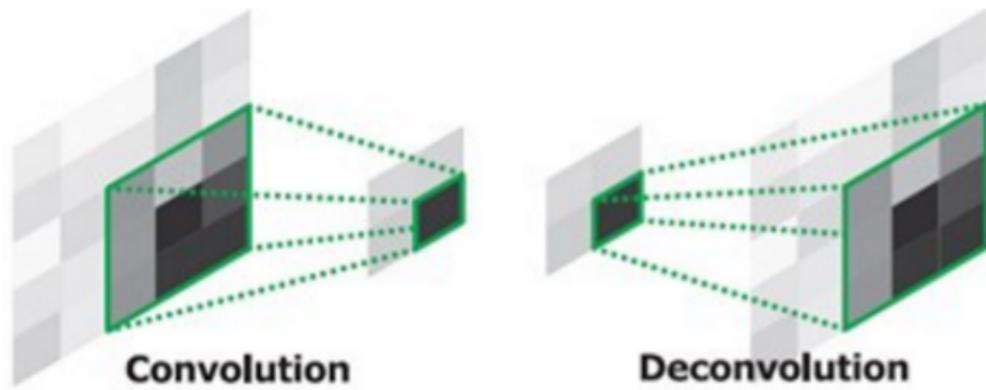
Unpooling (upsampling)



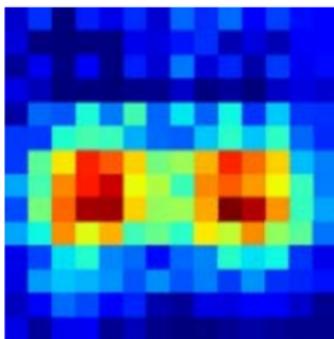
Deconvolution

- Deconvolution слой `layers` уплотняет рареженный слой, полученный в `unpooling`, с использованием операций свертки
- В отличие от сверточных слоев, которые соединяют множественные входные значения активации нейронов внутри окна фильтра в одно значение, обратная операция свртки ассоциирует одиночное входное значение активации с множественным выходом

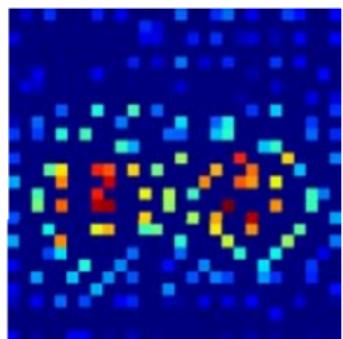
Deconvolution (схема)



Deconvolution and unpooling



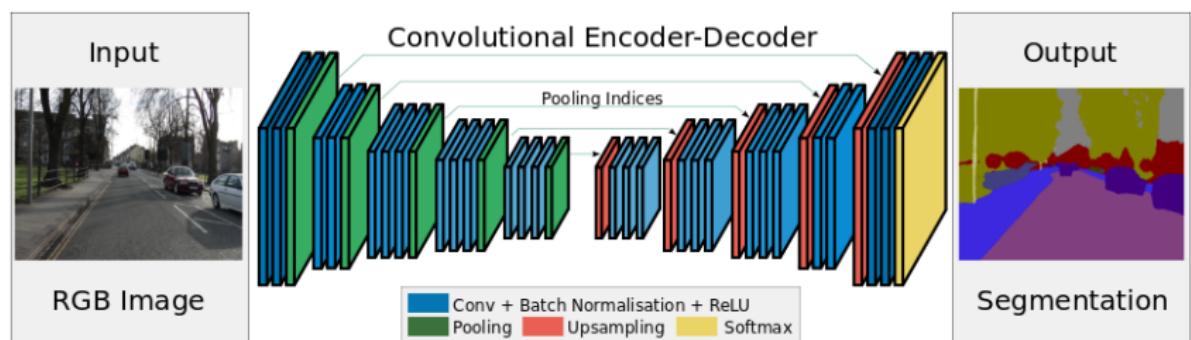
Deconv: 14x14



Unpool: 28x28

SegNet

V. Badrinarayanan, A. Kendall and R. Cipolla "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation." arXiv preprint arXiv:1511.00561, 2015



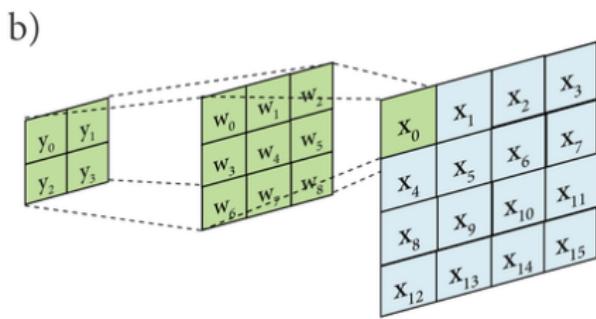
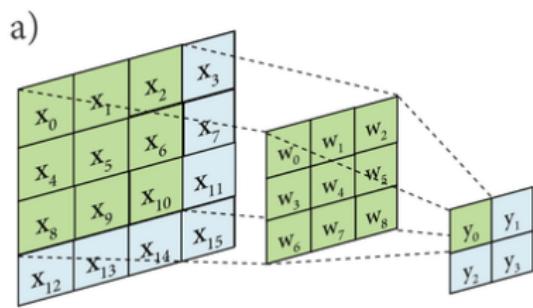
Особенность SegNet

- Сеть состоит из :
 - последовательности слоев нелинейного преобразования (кодер) и
 - соответствующего множества декодеров, за которыми следует “попиксельный” классификатор
- **Ключевой элемент** SegNet - декодер использует индексы, полученные на этапах max-pooling в кодере, для реализации upsampling
- Индексы - местоположения признаков с максимальными значениями в каждом окне pooling (запоминаются для каждой карты признаков в кодере)
- Это позволяет избежать обучения процедуры upsampling.
- Сеть в целом обучается, используя градиентный спуск

Semantic Segmentation, классификация на уровне пикселов

- Сжатие изображения посредством использования пулинга и сверток
- Эта конфигурация кодера для задач классификации, так как она заботилась только о содержании изображения, а не о его местоположении
- Однако для задачи сегментации необходимо иметь маску с полным разрешением для пиксельного прогнозирования

Semantic Segmentation, классификация на уровне пикселов



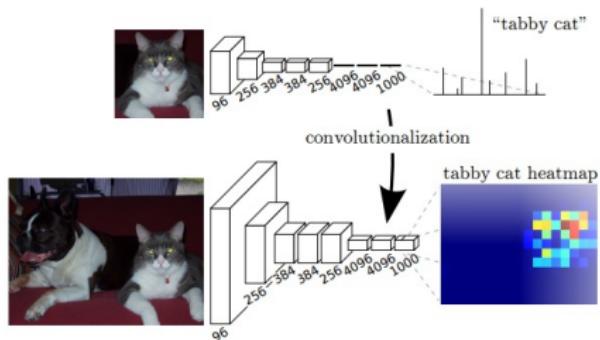
Fully Convolutional Network (FCN) - Полносверточная сеть

Полносверточная сеть была предложена для устранения patch-by-patch scanning и для повышения эффективности FCN заменяет полносвязанные слои в сверточной сети на 1×1 сверточные ядра.

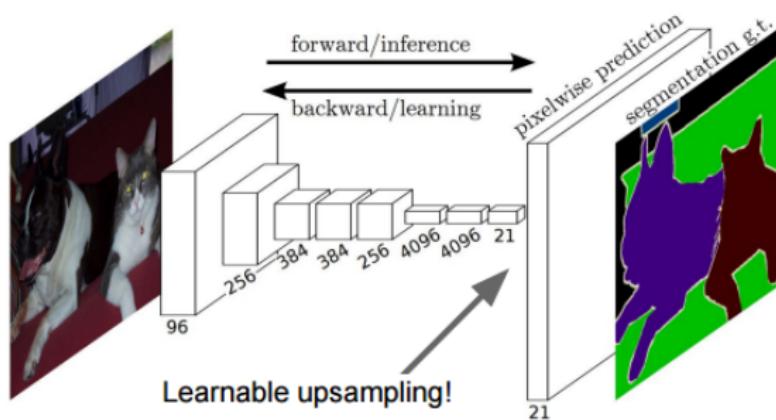
FCN в качестве входа использует всю картинку и получает сегментационную карту на выходе одним проходом прямого распространения

Fully Convolutional Network (FCN)

- Преобразование полносвязанных слоев в сверточные, охватывающих всю входную область
- Каждая из этих сверток будет выводить coarse heatmap меток

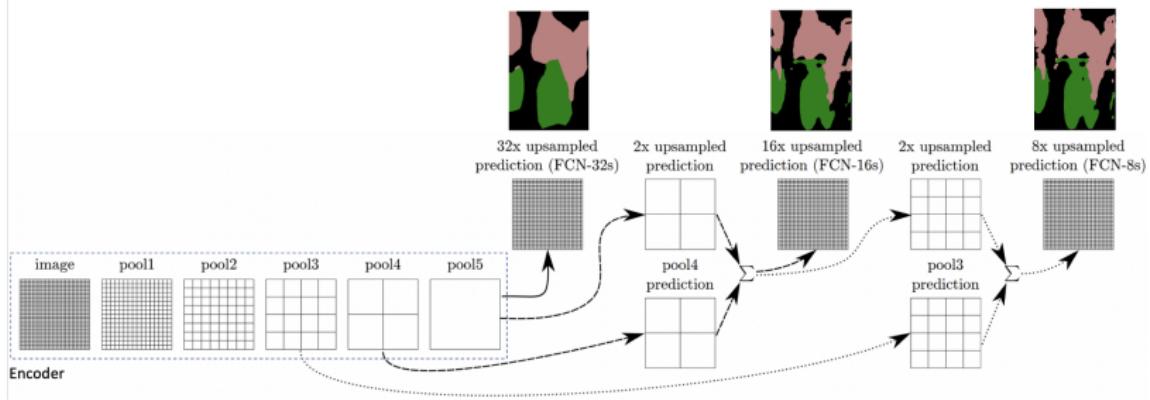


FCN



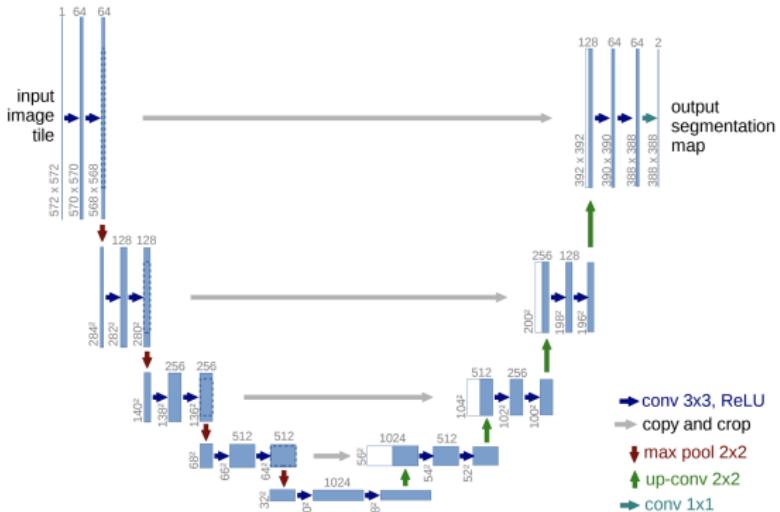
Кодер сжимает изображение в изображение с более низким разрешением, что приводит к грубой сегментации после операции повышения дискретизации (upsampling)

FCN



Положение на слоях более высокого уровня соответствуют положениям на изображении, с которыми они связаны путями, называемыми *рецептивными полями*

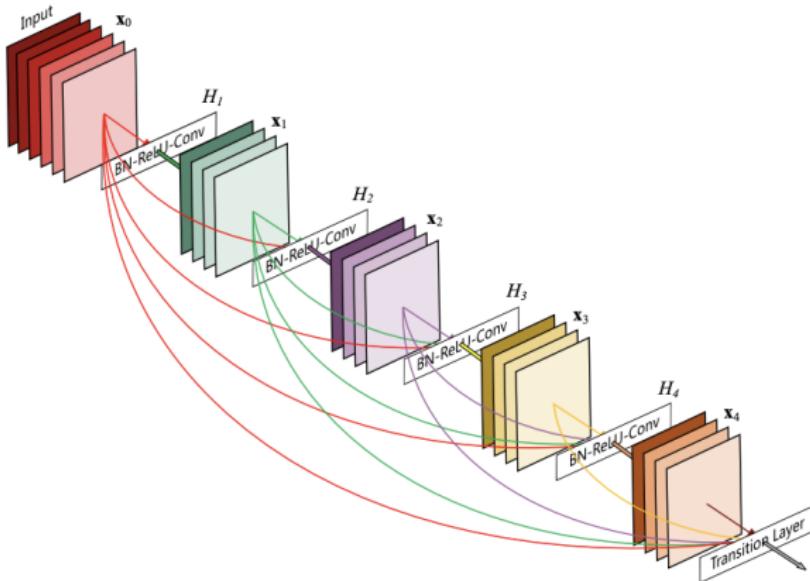
U-Net



U-Net

- Первая часть состоит из обычных сверток, ReLU и пулинга 2×2 с шагом 2, который приводит к понижающей дискретизации (downsampling)
- Вторая часть состоит из последовательности повышающей дискретизации карты (upsampling), конкатенации соответствующей карты признаков, которая необходима из-за потери границ пикселей после каждой свертки, свертки и применения ReLU
- Как и FCN, признаки высокого разрешения из пути сжатия объединяются с upsampled выходом, который затем подается на ряд сверточных слоев

FC-DenseNet

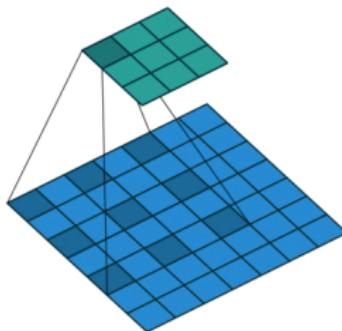


FC-DenseNet

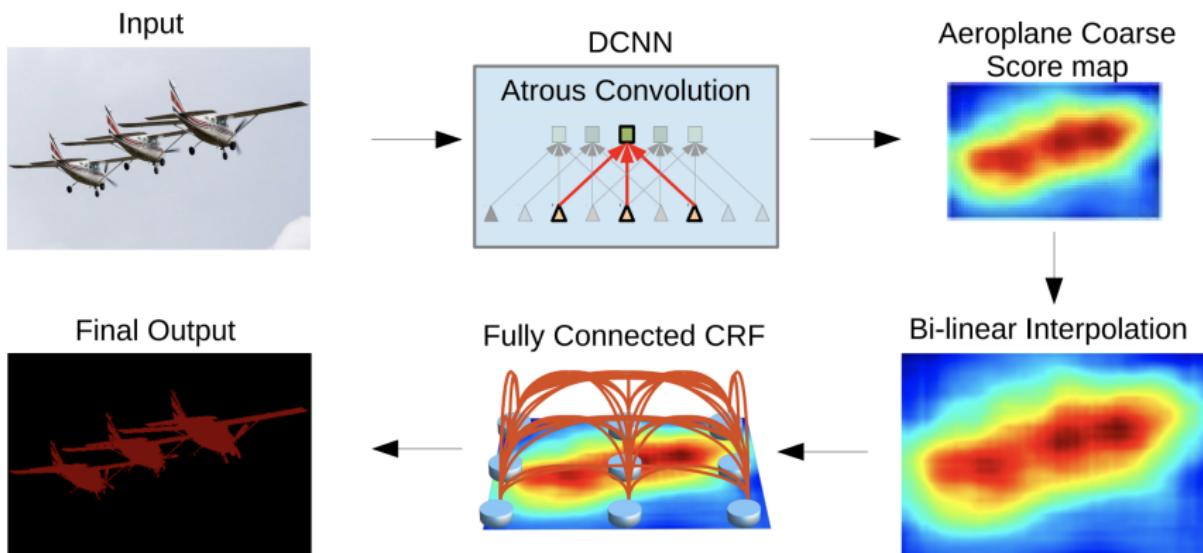
- FC DenseNet или 100 Layers Tiramisu - это метод сегментации, основанный на архитектуре DenseNet
- DenseNet основан на парадигме, в которой “перемычки” с ранних уровней устанавливаются на более поздние уровни и все слои связаны друг с другом
- Каждый слой передает свои карты признаков всем последующим слоям
- Если ResNet использует поэлементное суммирование для объединения признаков, то DenseNets - конкетенацию
- Каждый уровень получает совокупный набор “знаний” от всех предыдущих уровней

DeepLab

- DeepLab v1 основан на двух идеях: Atrous Convolution и полносвязанное условное случайное поле (FC CRF)
- Atrous convolution с французского “a trous” - дыра (hole), также называется “расширяющаяся (dilated) свертка”
- “Расширяющаяся (dilated) свертка” - это стандартная свертка, через которую пропускается некоторое число пикселей в двух измерениях



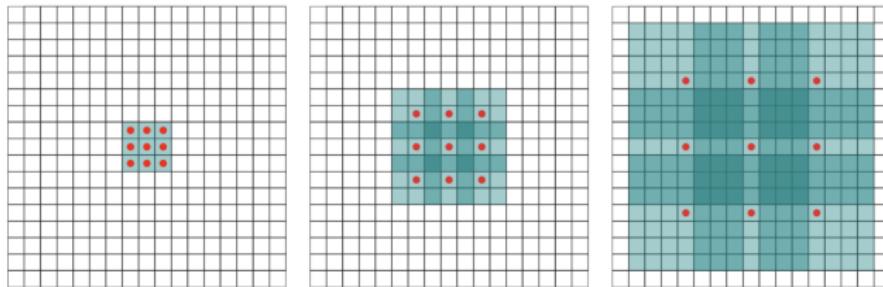
DeepLab



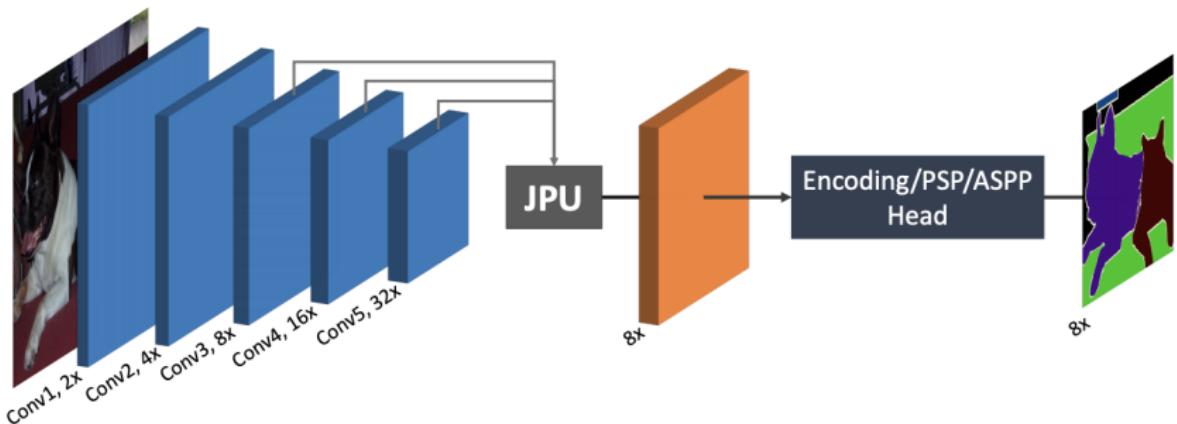
FC CRF is applied to refine the segmentation result

DeepLab

- Вместо типового пулинга DeepLab использует расширенные слои для решения проблемы балансировки.
- Контролируя поле зрения в dilated свертках, можно найти лучший компромисс между точной локализацией (малое поле зрения) и ассилияцией контекста (большое поле зрения)



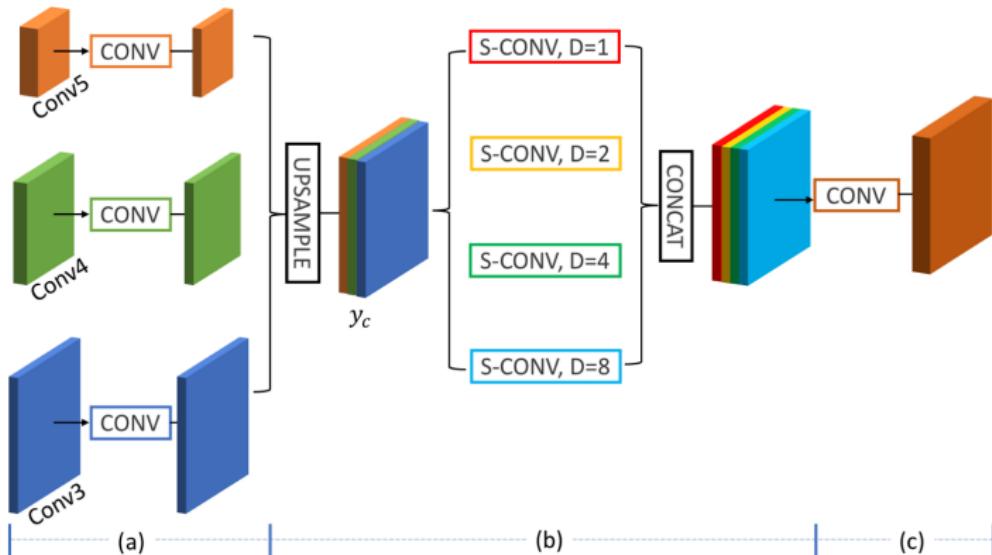
Fast FCN



Fast FCN

- Принцип расширенной свертки был заменен совместным пирамидальным upsampling (JPU)
- Различия между Fast FCN и DilatedFCN заключаются в последних двух стадиях свертки
- Как правило, карта входных объектов сначала обрабатывается обычным слоем свертки, а затем серией расширенных сверток.
- Fast FCN концептуально обрабатывает входную карту признаков сверткой, а затем использует несколько сверток для генерации выходных данных
- Это снижает вычислительную сложность по сравнению с DilatedFCN
- JPU создан, чтобы упростить процесс оптимизации.

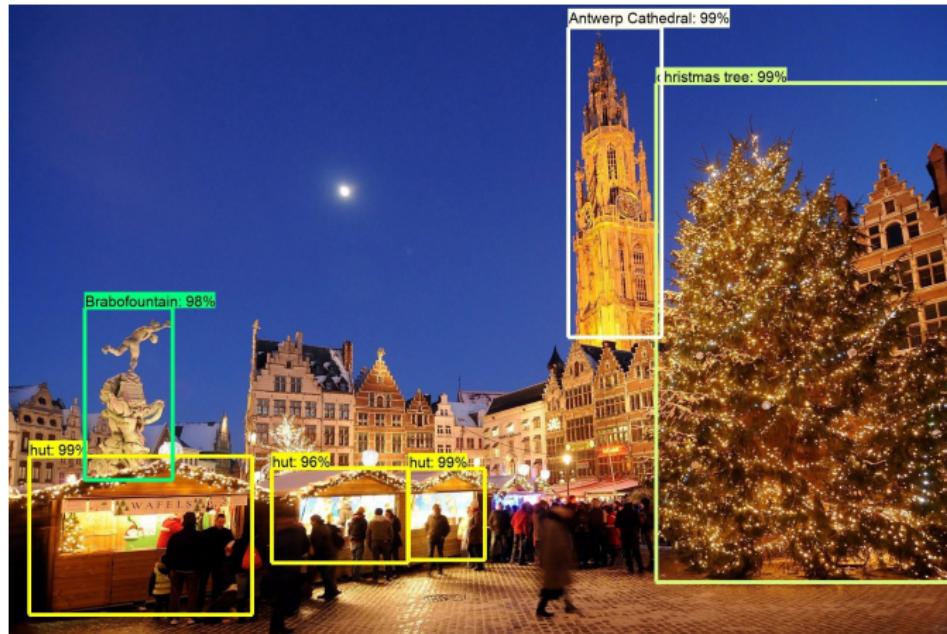
Fast FCN - архитектура JPU



Fast FCN - архитектура JPU

- Каждая карта признаков проходит через обычный сверточный блок
- Затем карты признаков подвергаются дискретизации и конкатенации, которые затем проходят через четыре свертки с различными степенями расширения
- Результаты свертки снова объединяются и проходят через слой окончательной свертки
- Где ASPP использует только информацию из последней карты признаков, JPU извлекает информацию о всем контексте из карт многоуровневых объектов, что приводит к повышению производительности

Detection обнаружение



Detection обнаружение

- При обучении детектора набор данных будет иметь для каждого изображения 0 или более ограничивающих прямоугольников (bounding box) и соответствующих им меток
- Т.е. модель имеет 2 выхода:
 - распределение вероятностей задачи классификации
 - прогнозирование bounding box
- Решение - детектор для обнаружения одного конкретного объекта по всему изображению. Один детектор для кошек, другой - для собак.
- **Другое решение** - использование скользящего окна в сочетании с простым классификатором изображений

Detection обнаружение

- ① Создать небольшое окно, вырезать область внутри окна и передать его в ConvNet.
- ② Сдвинуть окно дальше и продолжить передавать вырезанное содержимое в ConvNet.
- ③ Как только окно переместится по всему изображению, увеличить размер окна и повторите шаг 1 и шаг 2.
- ④ Конечным результатом является набор различных вырезанных изображений, где некоторые содержат метки и bounding box(es) объекта(ов).

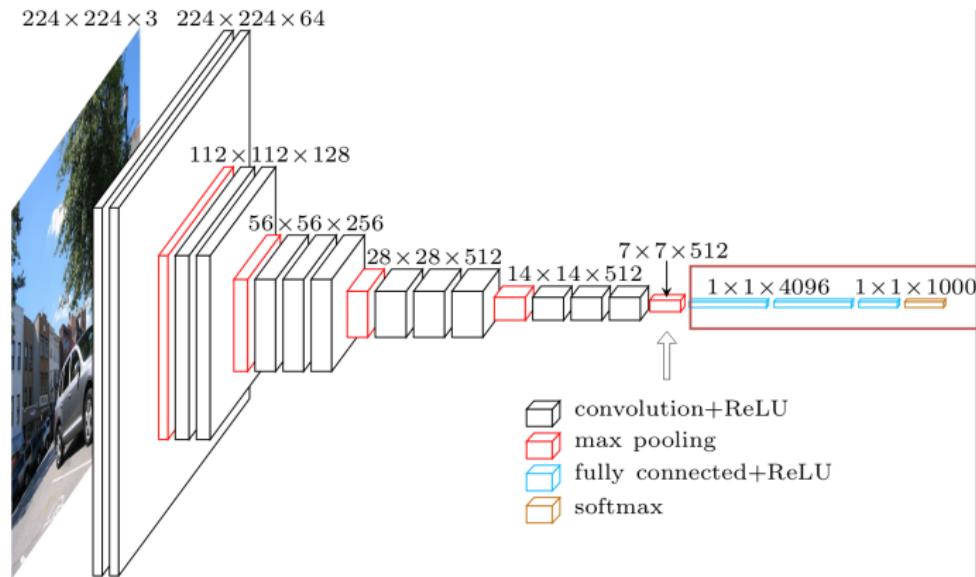
Минусы алгоритма:

- Скользящее окно имеет фиксированную прямоугольную форму - неточное обнаружение bounding box, если ни одно из скользящих окон не соответствует реальному объекту
- Постоянное вырезание изображений и подача их в ConvNet требует больших вычислительных ресурсов

One-stage Object Detection

- One-shot detectors обходят оба минуса, используя фиксированное множество детекторов на сетке (grid detectors)
- Каждый bounding box детектор находится в определенной позиции на изображении
- Подаем изображение в обычную сверточную сеть, которая называется основой детектора объектов.
- Эта сеть - есть не что иное, как классификатор изображений и ее можно предварительно обучать на огромных наборах данных (ImageNet)

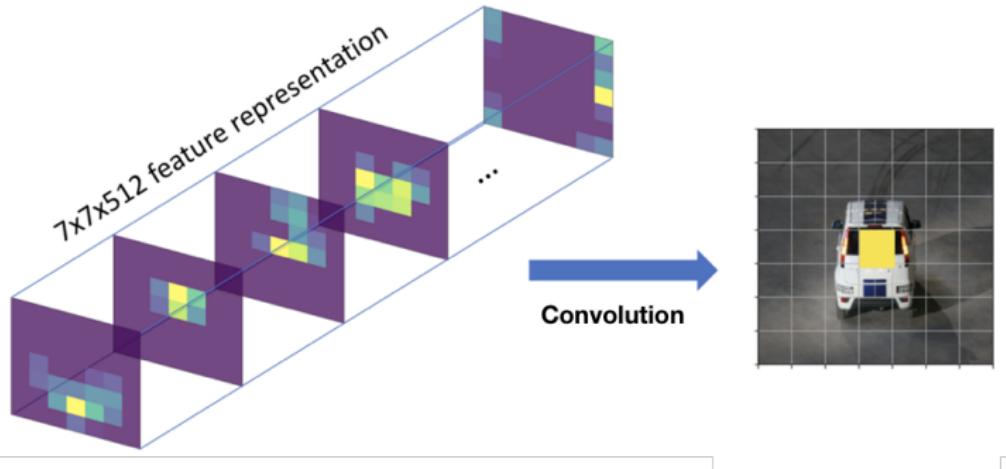
Detection обнаружение - основная сеть



Detection обнаружение

- Удаляем последние слои (красный прямоугольник), так как не надо классифицировать входное изображение
- Выход основной сети - изображение $7 \times 7 \times 512$, которое имеет низкое пространственное разрешение и высокое разрешение признаков
- Располагая эту карту признаков на исходном изображении, можно сопоставить ячейки сетки и входное изображение
- При таком отображении ячейка сетки, которая содержит центр bounding box, может быть аппроксимирована.

Detection обнаружение



Detection обнаружение

- Сетка объединит все 512 признаков, используя сверточные слои, чтобы обнаружить объект с помощью bounding box.
- Атрибуты, необходимые для описания обнаруженного объекта:
 - координаты x-y и ширина/высота bounding boxes (4)
 - вероятность ячейки сетки, содержащей объект (1)
 - класс из N классов, к которым принадлежит обнаруженный объект
- Каждая ячейка сетки выполняет $4+1+N$ сверток, чтобы обнаружить и нарисовать bounding box для объекта.

Вопросы

?