

DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs

Mikhailov Nikita, 161

Higher School of Economics

01.03.2019

Примеры задач компьютерного зрения

- Распознавание
(текста)

Примеры задач компьютерного зрения

- Распознавание
(текста)
- Идентификация
(личности)

Примеры задач компьютерного зрения

- Распознавание
(текста)
- Идентификация
(личности)
- Обнаружение
(лиц, машин,)

Примеры задач компьютерного зрения

- Распознавание
(текста)
- Идентификация
(личности)
- Обнаружение
(лиц, машин,)

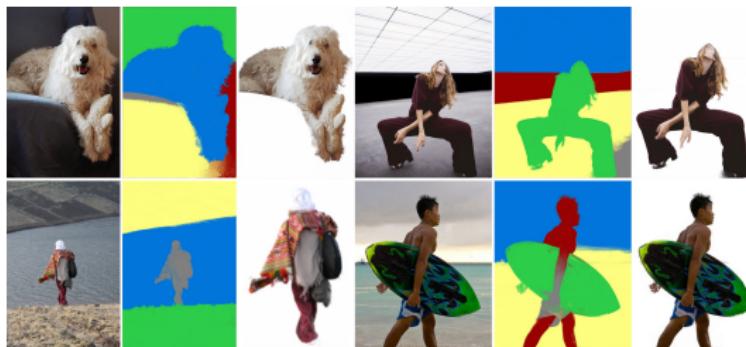


Сегментация изображений

Сегментация

Разделение изображения на сегменты (множества пикселей)

- Графы: использование MST, минимального разреза и т.д.
- k-Means
- Нейросетевой подход



Семантическая сегментация

Семантическая сегментация

Не просто выделение областей на изображении, но и его понимание.



Человек переходит дорогу.

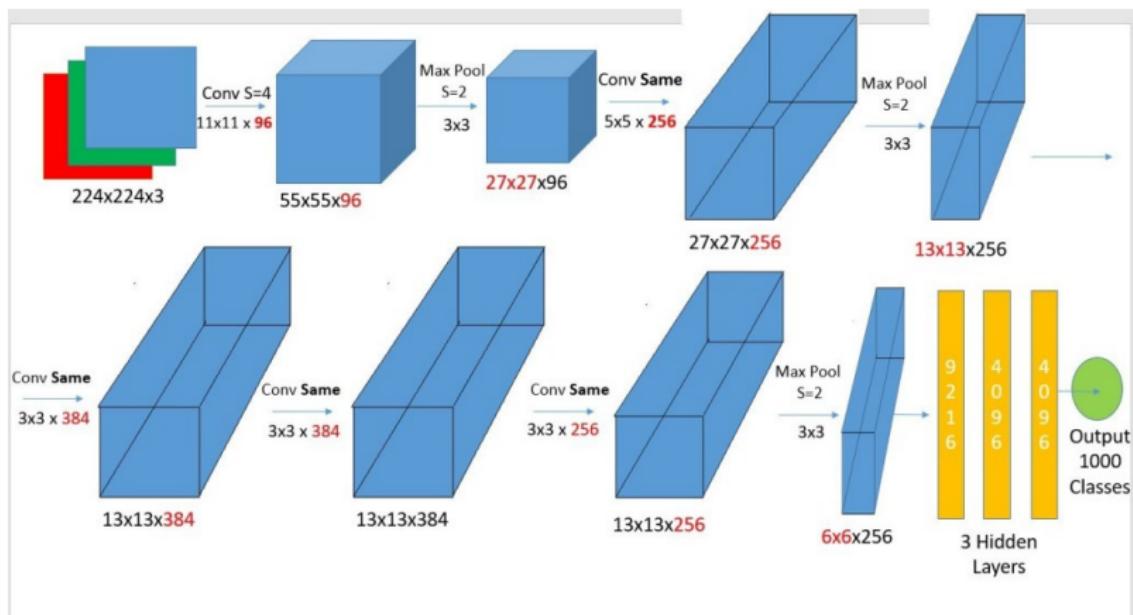
Минусы донейросетевых подходов

- Качество
- Использование эвристик
- Низкая универсальность

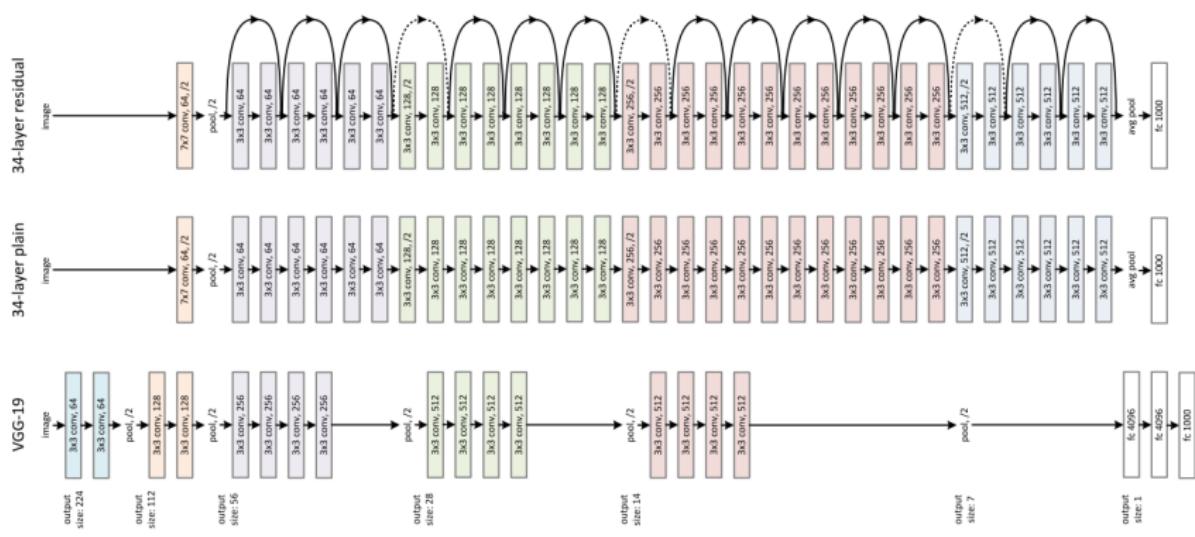
Текущие нейронные сети для классификации

- AlexNet
- ResNet
- VGG

AlexNet



VGG и ResNet



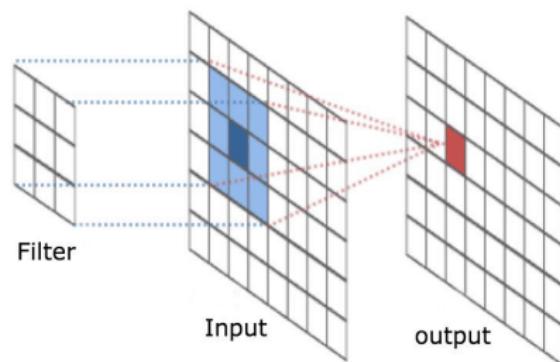
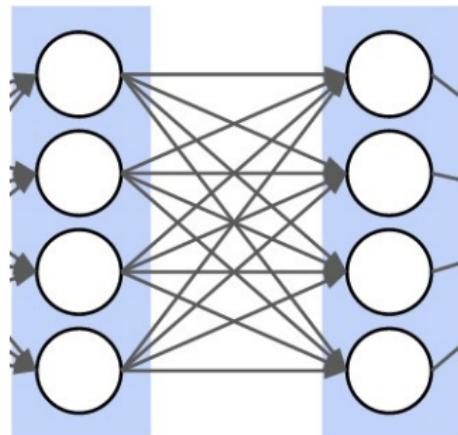
Особенности

- Сначала идут свертки, а в конце полносвязные слои
- Все полносвязные слои принимают на вход тензоры фиксированного размера
- Часто в сетях есть пулинг
- Сети обучаются так, чтобы быть устойчивыми к искажениям

Из классификации в сегментацию

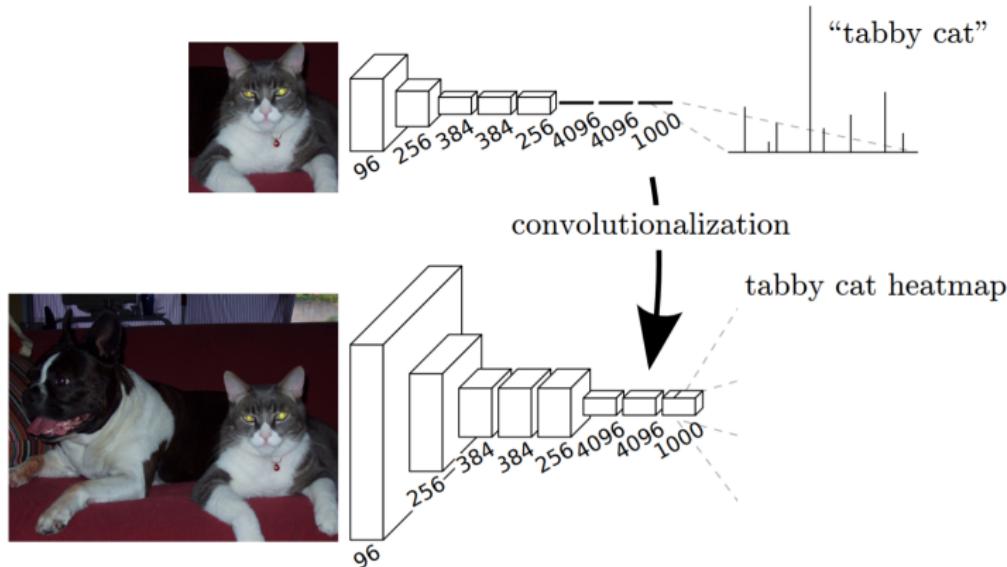
FC → Conv

Полносвязный слой после свертки есть свертка с ядром, покрывающим все изображение.



Из классификации в сегментацию

Если заменить все полносвязные слои на сверточные, то сеть будет принимать изображения любых размеров и классифицировать каждый пиксель.



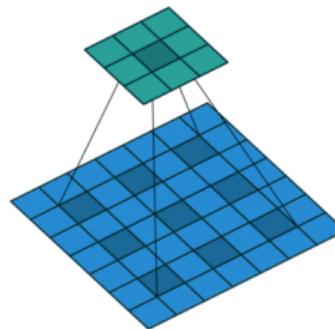
Atrous Convolutional

Пример разряженной свертки на одномерном случае

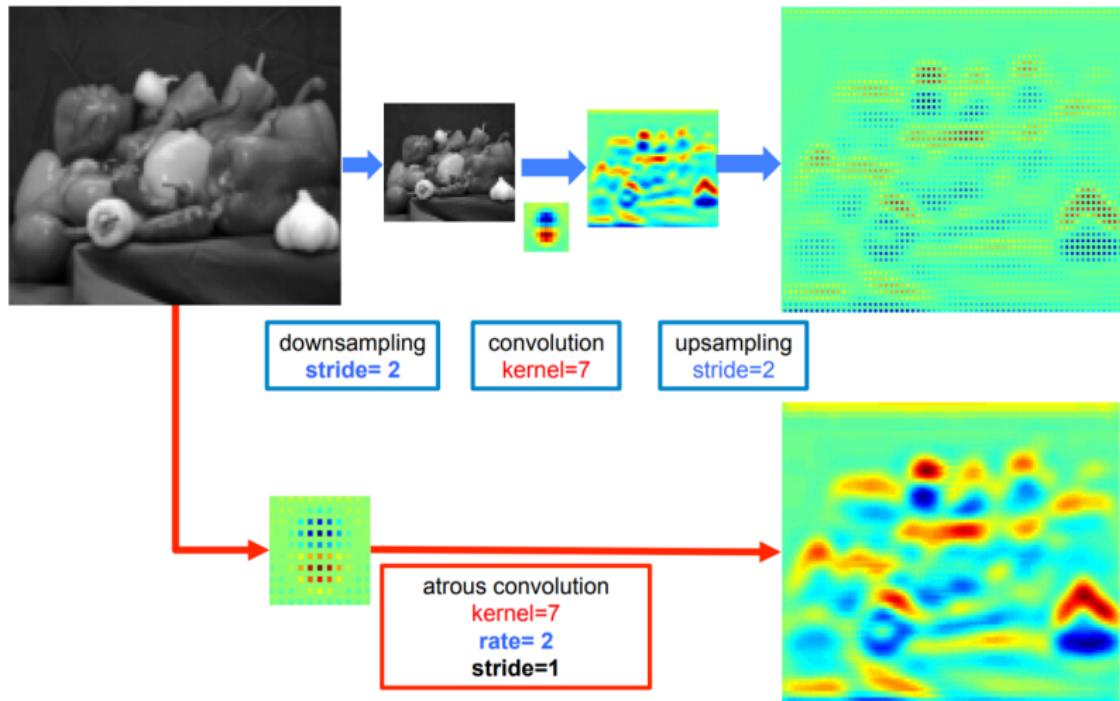
$$y[i] = \sum_{k=1}^K x[i+k - \frac{K-1}{2}] \cdot w[k]$$
$$y[i] = \sum_{k=1}^K x[i + rk - \frac{r(K-1)}{2}] \cdot w[k]$$

При $r = 1$ выполняется обычная свертка.

На примере ниже $r = 2$ – пропуск каждого второго пикселя



Отсутствие потерь деталей



Atrous Convolutional

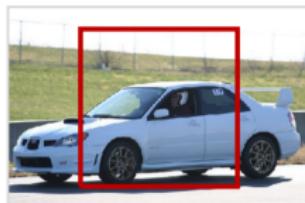
Плюсы использования разреженных сверток:

- Увеличение размера фильтра без увеличения числа параметров
- Возможность строить карту признаков с высоким разрешением, но это требует больше вычислительной мощности
- Обрабатывать разные масштабы одним ядром(далее)

Spatial pyramid pooling

Сеть, классифицирующая изображение имеет ограниченный вход.
Способы подгона изображения под окно:

- Обрезание (не то, о чем вы подумали)
- Сжатие/растяжение
- Проход скользящим окном



crop

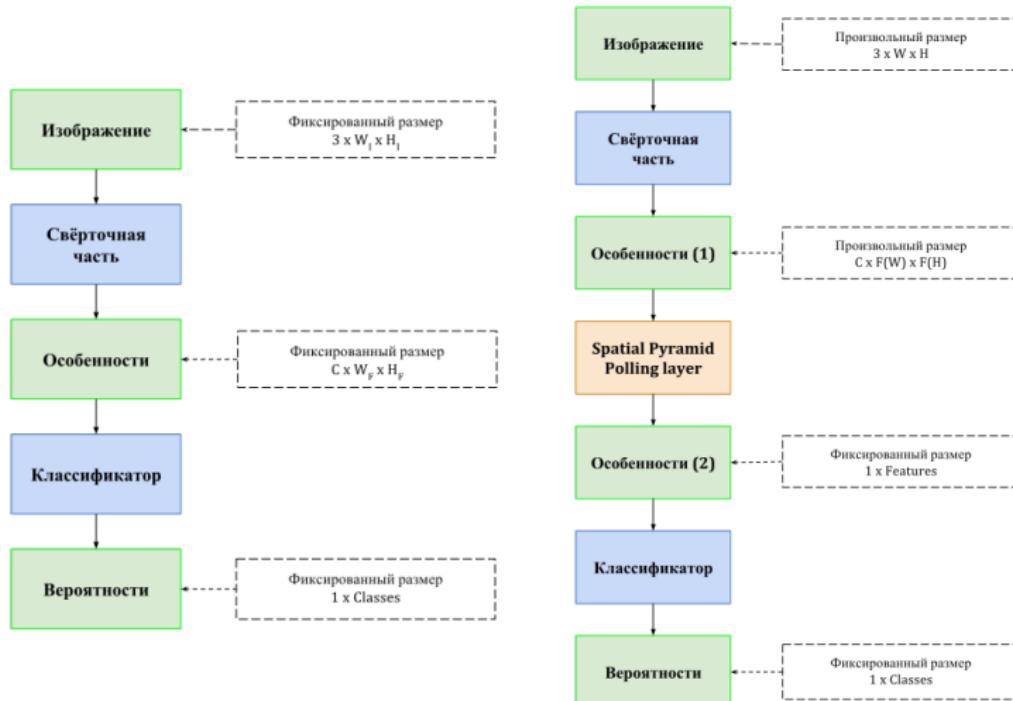


warp



Теряется или искажается объект, что приводит к уменьшению
классификации

Spatial pyramid pooling



Spatial pyramid pooling

В классической ситуации для обобщающего слоя мы задаем:

- Размер окна (W_{win}, H_{win})
- Размер сдвига (stride) по горизонтали и вертикали (W_{str}, H_{str})
- Ширину бордюра добавляемого (padding) вокруг входной матрицы (W_{pad}, H_{pad})
- Обобщающую функцию (в основном используется максимум).

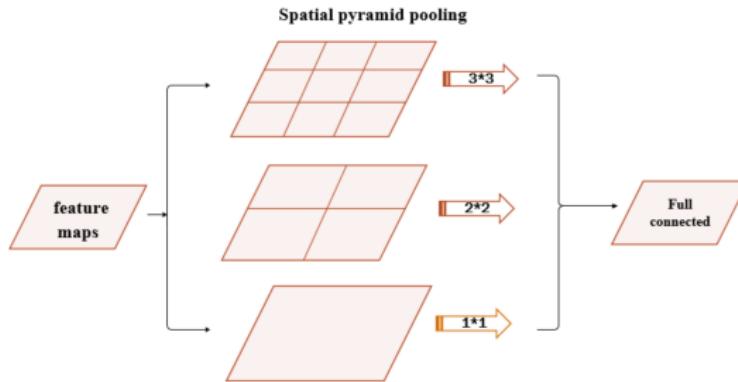
Тогда мы можем рас считать размер выхода (W_{out}, H_{out}).

Spatial pyramin pooling

Аналогично с классическим случаем мы можем задать (W_{out}, H_{out}) и по нему расчитать (W_{win}, H_{win})

Теперь можем обрабатывать изображения любого размера.

На примере ниже сделано несколько пуллинг слоев, которые потом объединяются и передаются в полносвязный слой.



Atrous Spatial Pyramid Pooling

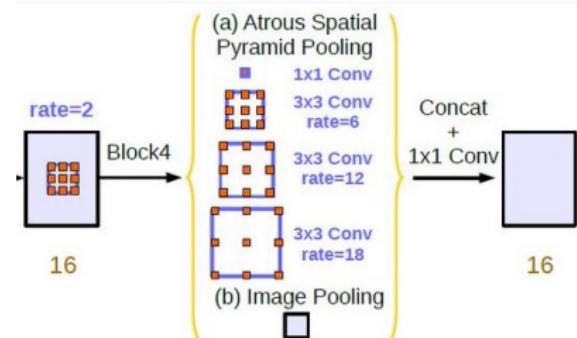
Сеть должна быть устойчива к масштабированию. Обычный метод:

- Сделать из исходной картинки несколько отмасштабированных изображений
- Подать на вход сети

Но предлагается Atrous Spatial Pyramid Pooling

Atrous Spatial Pyramid Pooling

Как ни странно, общего немного, но позволяет рассмотреть изображение под разными масштабами сразу



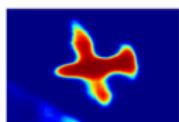
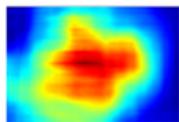
Postprocessing

После замены всех полносвязных слоев на сверточные мы получим сегментацию.

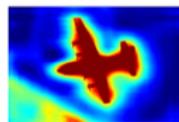
Она достаточно хорошая, но ошибается на границах объектов. DeepLab предлагает использовать fully connected CRF.



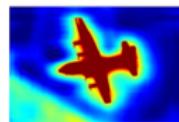
Image/G.T.



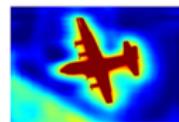
DCNN output



CRF Iteration 1



CRF Iteration 2



CRF Iteration 10

fully connected Conditional Random Field

- Представляем изображение в виде полного графа.
- Пусть $\{\mathbb{X} = X_1, \dots, X_N\}$ – множество переменных, отвечающих за класс пикселя
- $\mathbb{I} = \{I_1, \dots, I_N\}$ – множество признаковых описаний пикселя. В нашем случае RGB-вектор

fully connected Conditional Random Field

Тогда можем найти (очевидно)

$$Pr(\mathbb{X}|\mathbb{I}) = \frac{1}{Z(\mathbb{I})} \exp(-E(\mathbb{X}|\mathbb{I}))$$

$$E(x) = \sum_i^N \theta_i(x_i) + \sum_{i < j}^N \theta_{ij}(x_i, x_j),$$

где $\theta_i(x_i)$ – есть предсказание нашей сети для пикселя, то есть вектор вероятностей для классов

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \left[w_1 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_1^2}\right) - \frac{\|I_i - I_j\|^2}{2\sigma_2^2} \right] + w_2 \exp\left(\frac{\|p_i - p_j\|^2}{2\sigma_1^3}\right),$$

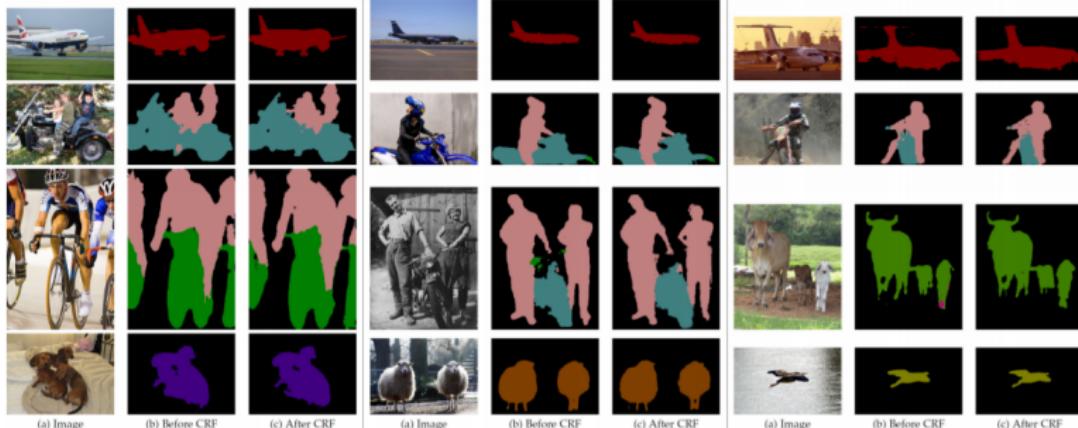
где p_i положение i -пикселя

fully connected Conditional Random Field

- в первом ядре штрафуем за то, что близкие пиксели с одинаковым цветом отнеслись к разным классам
- а во втором просто за ошибку на близких пикселях.

fully connected Conditional Random Field

Before and after fcCRF



Results

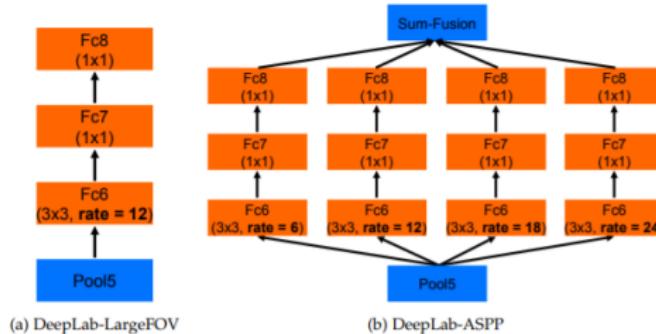


Fig. 7: DeepLab-ASPP employs multiple filters with different rates to capture objects and context at multiple scales.

| Method | before CRF | after CRF |
|----------|------------|-----------|
| LargeFOV | 65.76 | 69.84 |
| ASPP-S | 66.98 | 69.73 |
| ASPP-L | 68.96 | 71.57 |

Выводы

- Для сегментации подходят сети, обученные на классификацию
- Полносвязный слой можно представить в кач-ве сверточного
- Пулинг хорошо для классификации, но плохо сказывается на сегментации границ объектов.
- Разряженные свертки сохраняют разрешение на карте признаков
- Crop и Warp – плохо
- CRF – помогает сгладить результаты