Fast R-CNN

Курбатова Виктория

R-CNN описанная ранее имеет несколько недостатков, в основном связанных с высокими временными затратами.

1. Необходимо обучить три разные модели по отдельности - CNN для генерации объектов изображения, классификатор, который предсказывает класс, и модель регрессии для ужесточения ограничивающих рамок.

2. Требуется прямой проход CNN для каждой гипотезы для каждого отдельного изображения (это около 2000 прямых проходов на изображение!).
3. Непосредственно процесс детектирования и классификации требует существенного времени (авторы пишут о 47 секундах на одно изображение при использовании VGG16 в качестве свёрточной сети)

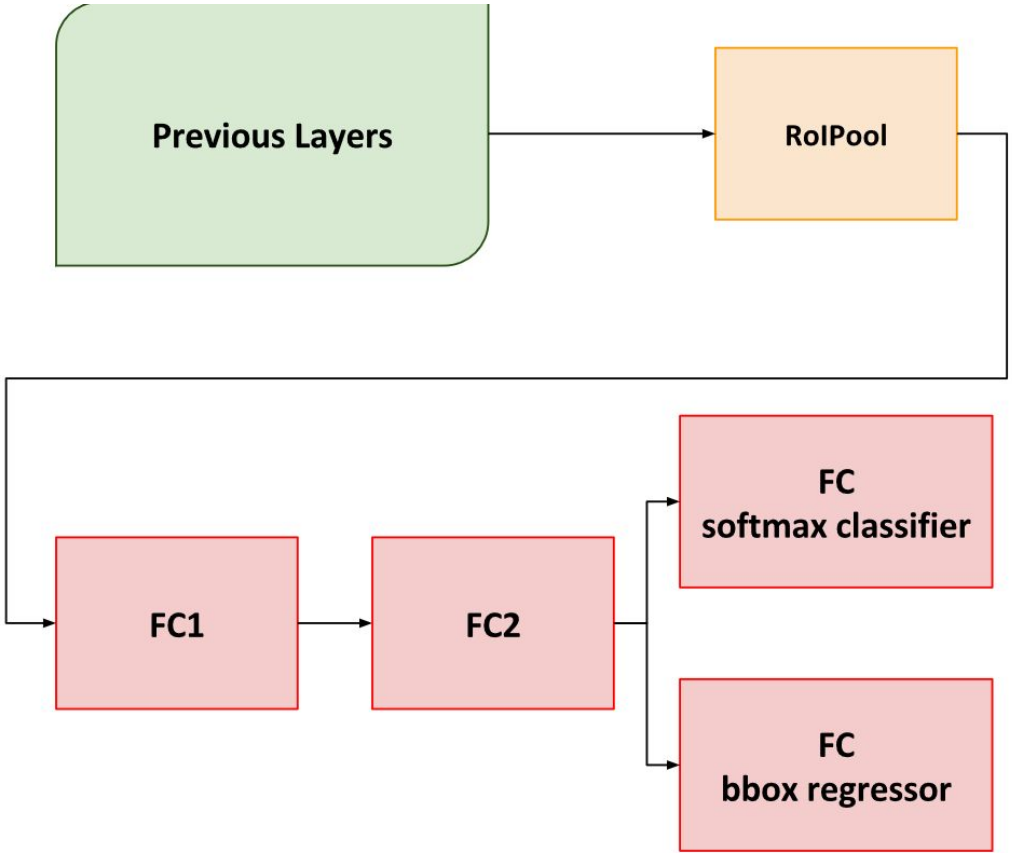
Авторы Fast R-CNN предложили ускорить процесс за счёт пары модификаций:

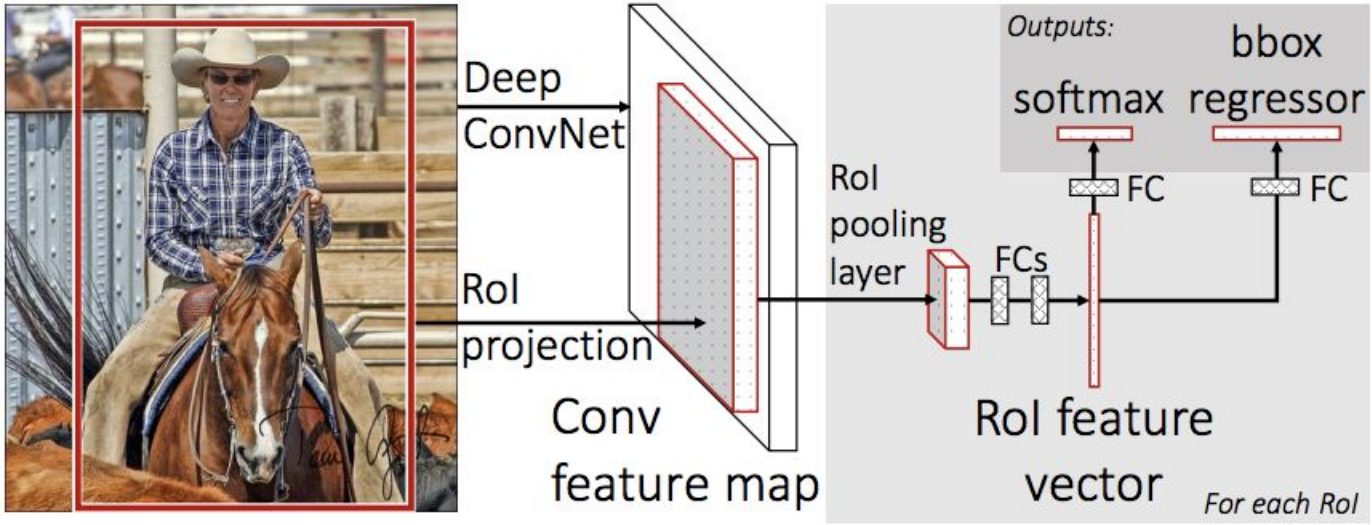
- Пропускать через CNN не каждый из 2000 регионов-кандидатов по отдельности, а всё изображение целиком. Предложенные регионы потом накладываются на полученную общую карту признаков;
- Вместо независимого обучения трёх моделей (CNN, SVM, bbox regressor) совместить все процедуры тренировки в одну.

- Авторы предлагают подавать на вход сети полное изображение, но при этом последний max-pool слой заменить на слой нового типа *RoI pooling*.

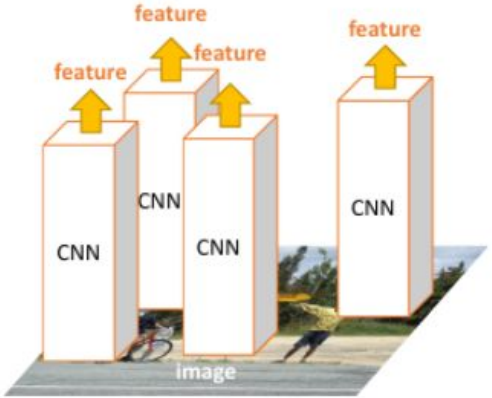
***RoI pooling* слой :**

- Принимает на вход карту особенностей, полученную от последнего свёрточного слоя нейронной сети, и RoI претендента;
- RoI преобразуется из координат изображения в координаты на карте особенностей и на полученный прямоугольник накладывается сетка *W x H* с наперед заданными размерами ;
- Делается *max pooling* по каждой ячейке этой сетки.


Previous Layers
RoIPool
FC1
FC2
FC
softmax classifier
FC
bbox regressor

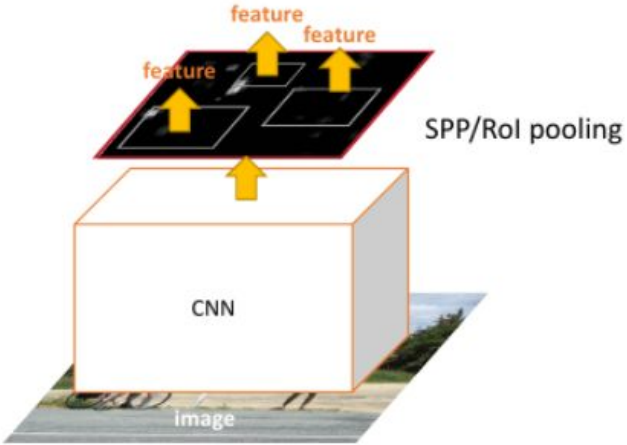

Deep
ConvNet
RoI
projection
Conv
feature map
RoI
pooling
layer
FCs
Outputs:
softmax
bbox
regressor
FC
FC
RoI feature
vector
For each RoI

R-CNN vs. Fast R-CNN (forward pipeline)



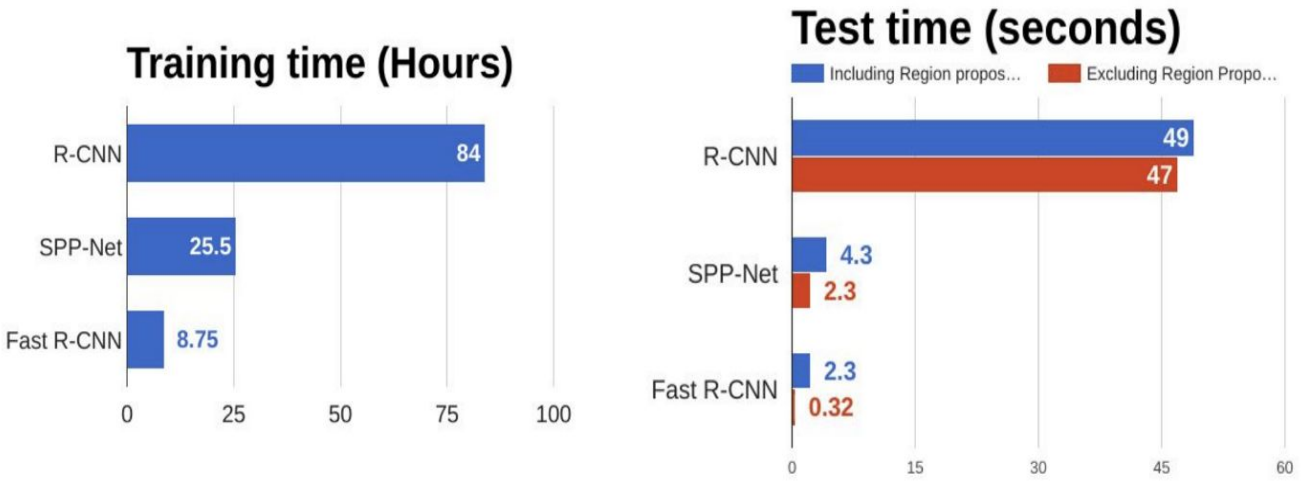
R-CNN

- Extract image regions
- 1 CNN per region (2000 CNNs)
- Classify region-based features



SPP-net & Fast R-CNN (the same forward pipeline)

- 1 CNN on the entire image
- Extract features from feature map regions
- Classify region-based features



Сравнение алгоритмов обнаружения объектов

Тренировка. Функция потерь.

Чтобы тренировать полученную сеть, необходимо ввести функцию потерь, объединяющую ошибки классификации и регрессии координат прямоугольника.

$p=(p_0,p_1,\ldots,p_N)$ - выход с *softmax* слоя - вектор вероятностей,что в соответствующем RoI содержится объект одного из $N+1$ классов.

Второй слой выдаст нам матрицу 4×N восстановленных сдвигов bbox-а объекта для каждого из N классов: $(t_x^n, t_y^n, t_w^n, t_h^n)$, $n=1,\ldots,N$.

Авторы предлагают следующую функцию потерь: $$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v)$$

Ошибка классификации вычисляется стандартно: $$L_{cls}(p, u) = -\log(p_u).$$

Во втором слагаемом в качестве функции потерь берется следующая сумма: $$L_{loc}(t^u, v) = \sum_{i=x,y,w,h} \Psi_1(t_i^u - v_i)$$

Через Ψ1(x) обозначена функция Хубера : $$\Psi_1(x) = \begin{cases} 0.5x^2, & \text{если} |x| < 1 \\ |x| - 0.5, & \text{иначе} \end{cases}$$

Минибатч

Чтобы уменьшить время тренировки, в минибатч лучше класть несколько претендентов из одного изображения.

Однако, не стоит злоупотреблять и составлять минибатч только из претендентов с одного изображения.

В статье предлагается следующая схема:

1. Выбирается случайным образом *N* изображений
2. На каждом выбирается *R / N* претендентов.

   Таким образом получаем минибатч составленный из *R* элементов.

Итоги

- Во-первых, в данном случае генерируется карта особенностей для всего изображения, а затем при помощи специального слоя из нее вычленяются карты для каждого из претендентов, что позволяет существенно сократить время вывода.
- Во-вторых, авторы отказались от тренировки отдельного SVM для каждого класса и научились тренировать сеть таким образом, чтобы использование softmax слоя давало результаты сопоставимые с использованием набора SVM.
- В-третьих, авторы отказались от использования отдельных линейных регрессоров и натренировали, для уточнения локализации объектов, отдельный слой нейронной сети.