**Обзор семейства сетей типа Region-CNN.**

План

1 Разработка нейросети R-CNN.

2 Возможности сетей R-CNN

3 Этапы развития сети R-CNN.

4 Практика применения сети R-CNN. Какие задачи может решать

5 Результаты тестирования сети R-CNN

6 Популярность сети R-CNN

7 Примеры использования

8 Список используемых материалов

9 Используемые термины и сокращения

1 Разработка нейросети R-CNN.

1.1 Свёрточная нейронная сеть

Это специальная архитектура [искусственных нейронных сетей](https://ru.wikipedia.org/wiki/Искусственная_нейронная_сеть), предложенная [Яном Лекуном](https://ru.wikipedia.org/wiki/Лекун,_Ян) в 1988 году[[1]](https://ru.wikipedia.org/wiki/Свёрточная_нейронная_сеть" \l "cite_note-1) и нацеленная на эффективное [распознавание образов](https://ru.wikipedia.org/wiki/Теория_распознавания_образов). Использует некоторые особенности [зрительной коры](https://ru.wikipedia.org/wiki/Зрительная_кора), в которой были открыты так называемые простые клетки, реагирующие на прямые линии под разными углами, и сложные клетки, реакция которых связана с активацией определённого набора простых клеток. Таким образом, идея свёрточных нейронных сетей заключается в чередовании свёрточных слоёв ([англ.](https://ru.wikipedia.org/wiki/Английский_язык) convolution layers) и субдискретизирующих слоёв (subsampling layers или pooling layers, слоёв подвыборки).

Архитектура сети R-CNN (Regions With CNNs) была разработана командой из UC Berkley для применения Convolution Neural Networks к задаче object detection.

Дальнейшее развитие этого подхода привело к идее обработки входного изображения по частям. Входное изображение предварительно анализировалось и разбивалось на 2000 отдельных картинок. Каждое изображение анализировалось отдельно. Это улучшило качество обработки изображения.

Для анализа каждой картинки использовалась специальная предварительно обученная нейросеть. Данная нейросеть обобщала данные не картинке и выдавала признаки по которым можно было проводить дальнейшую обработку информации.

В дальнейшем этот подход был модифицирован и разбиение изображения стали выполнять, используя выходные данные нейросети CNN.

Это позволило ускорить работу нейросети.

1.2 Особенности нейросети R-CNN.

Обработка изображения происходит в несколько этапов.

Выделение регионов-кандидатов при помощи Selective Search.  
Преобразование региона в размер, принимаемый CNN CaffeNet.  
Получение при помощи CNN 4096-размерного вектора признаков.  
Проведение N бинарных классификаций каждого вектора признаков при помощи N линейных SVM.  
Линейная регрессия параметров рамки региона для более точного охвата объекта

Данная нейросеть состоит из нескольких групп слоев.

Каждая группа слоев решает свою задачу.

1 Группа слоев решает задачу нахождения координат прямоугольных областей, где предположительно может находится объект.

Изначально использовался алгоритм Selective Search.

2 Найденные области отдельно обрабатываются более сложной группой слоев, решающих задачу классификации объекта.

В результате данная нейросеть имеет низкую скорость обработки изображений и требует большой датасет для обучения нейросети.

Данная нейросеть так же чувствительна к размеру изображений и их качеству .

2 Возможности нейросетей типа R-CNN

**Classification** — классификация изображения по типу объекта, которое содержит

* **Semantic segmentation** — определение всех пикселей объектов определённого класса или фона на изображении. Если несколько объектов одного класса перекрываются, их пиксели никак не отделяются друг от друга;
* **Object detection** — обнаружение всех объектов указанных классов и определение охватывающей рамки для каждого из них;
* **Instance segmentation** — определение пикселей, принадлежащих каждому объекту каждого класса по отдельности;

3 Развитие сетей типа R-CNN

3.1 Fast R-CNN

Авторы данной нейросети предложили следующие пути ускорения работы.

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.

Бинарные SVM не использовались, вместо этого выбранные признаки передавались на полносвязанный слой, а затем на два параллельных слоя: softmax и bounding box regressor.

В результате мы получили  
Обучение современных моделей, таких как VGG16 идет в 9 раз быстрее по сравнению с R-CNN, и в 3 раза быстрее, чем SPPnet.

Во время теста работает в 200 раз быстрее, чем R-CNN, и в 10 раз быстрее, чем SPPnet.

Имеет значительно более высокое значение MAP для POSCAL VOC по сравнению с R-CNN и SPPnet.

Fast R-CNN выпускается под лицензией MIT  
Автор Ross Girshick.

3.2 Faster R-CNN  
После улучшений, сделанных в Fast R-CNN, самым узким местом нейросети оказался механизм генерации регионов-кандидатов. В 2015 команда из Microsoft Research смогла сделать этот этап значительно более быстрым. Они предложили вычислять регионы не по изначальному изображению, а опять же по карте признаков, полученных из CNN. Для этого был добавлен модуль под названием Region Proposal Network (RPN)

В рамках RPN по извлечённым CNN признакам скользят «мини-нейросетью» с небольшим (3х3) окном. Полученные с её помощью значения передаются в два параллельных полносвязанных слоя: box-regression layer (reg) и box-classification layer (cls). Выходы этих слоёв базируются на так называемых anchor-ах: k рамках для каждого положения скользящего окна, имеющих разные размеры и соотношения сторон. Reg-слой для каждого такого anchor-а выдаёт по 4 координаты, корректирующие положение охватывающей рамки; cls-слой выдаёт по два числа – вероятности того, что рамка содержит хоть какой-то объект или что не содержит.  
Процесс обучения reg и cls слоёв объединённый; loss-функцию они имеют общую, представляющую собой сумму loss-функций каждого из них, с балансирующим коэффициентом.  
  
Оба слоя RPN выдают только предложения для регионов-кандидатов. Те из них, которые имеют высокую вероятность содержания какого-либо объекта, передаются дальше в модуль детектирования объектов и уточнения охватывающей рамки, который по-прежнему реализован как Fast R-CNN.  
  
Для того, чтобы разделять признаки, получаемые в CNN, между RPN и модулем детектирования, процесс обучения всей сети построен итерационно, с использованием нескольких шагов:

1. Инициализируется и обучается на определение регионов-кандидатов RPN-часть.
2. С использованием предлагаемых RPN регионов заново обучается Fast R-CNN часть.
3. Обученная сеть детектирования используется, чтобы инициализировать веса для RPN. Общие convolution-слои, однако, фиксируются и производится донастройка только слоёв, специфичных для RPN.
4. С зафиксированными convolution-слоями окончательно донастраивается Fast R-CNN.

Работает данная нейросеть значительно быстрее и точнее предшествующих алгоритмов.

3.3 Mask R-CNN

Mask R-CNN развивает архитектуру Faster R-CNN путём добавления ещё одной ветки, которая предсказывает положение маски, покрывающей найденный объект, и, таким образом решает уже задачу instance segmentation. Маска представляет собой просто прямоугольную матрицу, в которой 1 на некоторой позиции означает принадлежность соответствующего пикселя объекту заданного класса, 0 — что пиксель объекту не принадлежит.

Авторы документа условно разделяют разработанную архитектуру на CNN-сеть вычисления признаков изображения, называемую ими backbone, и head — объединение частей, отвечающих за предсказание охватывающей рамки, классификацию объекта и определение его маски.

Помимо высоких результатов в задачах instance segmentation и object detection, Mask R-CNN оказалась пригодной для определения поз людей на фотографии (human pose estimation). Ключевой момент здесь — выделение опорных точек (keypoints), таких как левое плечо, правый локоть, правое колено и т.п., по которым можно нарисовать каркас позиции человека

В экспериментах по Mask R-CNN, наряду с обычной CNN ResNet-50/101 в качестве backbone, также проводились исследования целесообразности использования Feature Pyramid Network (FPN). Они показали, что применение FPN в backbone даёт Mask R-CNN прирост как в точности, так и в производительности. Это делает полезным описание так же и данного улучшения, несмотря на то, что ему посвящён отдельный документ и с серией рассматриваемых статей он связан мало.  
Назначение Feature Pyramids, как и image pyramids, — улучшение качества детектирования объектов с учётом большого диапазона их возможных размеров.

3.4 Dinamic R-CNN

Несмотря на то, что в последние годы двухступенчатые объектные детекторы постоянно совершенствовали самые современные характеристики, сам процесс обучения далек от идеала. В этой работе мы сначала укажем на проблему несоответствия между фиксированными сетевыми настройками и процедурой динамического обучения, которая сильно влияет на производительность. Например, стратегия назначения фиксированных меток и функция потерь регрессии не могут соответствовать изменению распределения предложений и, таким образом, вредны для обучения высококачественных детекторов. Следовательно, мы предлагаем Dynamic R-CNN настроить параметры назначения меток (порог IoU) и форму функции потери регрессии (параметры потери SmoothL1) автоматически на основе статистики предложений во время обучения. Этот динамичный подход позволяет лучше использовать обучающие образцы   
В частности, наш метод улучшает базовый уровень ResNet-50-FPN с AP на 1,9% и AP на 5,5%90в наборе данных MS COCO без дополнительных затрат.

3.5 Siam R-CNN

Используется сиамская архитектура повторного обнаружения.

Так же используется динамический подход на основе треклетов, который использует преимущества повторного обнаружения как прогноза шаблона первого кадра, так и прогноза предыдущего кадра, чтобы смоделировать полную историю отслеживаемого объекта и потенциальных объектов отвлекающего объекта. Это позволяет нашему подходу принимать лучшие решения по отслеживанию, а также повторно обнаруживать отслеживаемые объекты после длительной окклюзии. Наконец, мы предлагаем новую сложную стратегию майнинга, чтобы улучшить устойчивость Siam R-CNN к похожим объектам. Siam R-CNN достигает наилучшей на данный момент производительности по десяти контрольным показателям отслеживания, с особенно сильными результатами для долгосрочного отслеживания.

8 Список используемых материалов

8.1 *R-CNN, Fast R-CNN, Faster R-CNN, YOLO - Алгоритмы обнаружения объектов*

*Понимание алгоритмов обнаружения объектов*

[Рохит Ганди](https://towardsdatascience.com/@grohith327?source=post_page-----36d53571365e----------------------) [9 июля 2018 г](https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e?source=post_page-----36d53571365e----------------------)

https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

8.2 Свёрточная нейронная сеть

[https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%91%D1%80%D1%82%D0%BE%D1%87%D0%BD%D0%B0%D1%8F\_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F\_%D1%81%D0%B5%D1%82%D1%8C](https://ru.wikipedia.org/wiki/Свёрточная_нейронная_сеть)

8.3 АНАЛИЗ АЛГОРИТМОВ ПОИСКА ОБЪЕКТОВ НА ИЗОБРАЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ РАЗЛИЧНЫХ МОДИФИКАЦИЙ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

<http://www.vestnik.vsu.ru/pdf/analiz/2019/03/2019-03-12.pdf>

8.4 Сравнение некоторых методов решения задачи детектирования лиц на изображениях

https://pdfs.semanticscholar.org/dcd7/43fa5df146626c2c7909c761da0f4c0c9303.pdf

8.5 Оригинальные документы с исследованиями на arXiv.org:  
1. R-CNN: <https://arxiv.org/abs/1311.2524>  
2. Fast R-CNN: <https://arxiv.org/abs/1504.08083>  
3. Faster R-CNN: <https://arxiv.org/abs/1506.01497>  
4. Mask R-CNN: <https://arxiv.org/abs/1703.06870>  
5. Feature Pyramid Network: <https://arxiv.org/abs/1612.03144>

8.6 Dynamic Objects Segmentation for Visual Localization in Urban Environments

https://arxiv.org/pdf/1807.02996.pdf

8.7 Siam R-CNN: визуальное отслеживание путем повторного обнаружения

https://arxiv.org/abs/1911.12836