

Введение в искусственный интеллект.

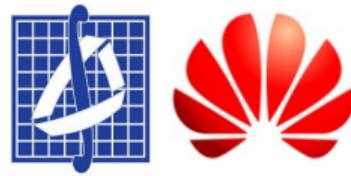
Современное компьютерное зрение

Лекция 6. Методы обнаружения объектов

Бабин Д.Н., Иванов И.Е., Петюшко А.А.

кафедра Математической Теории Интеллектуальных Систем

30 марта 2021 г.



План лекции

- ① Постановка задачи
- ② Метрики качества
- ③ Двухпроходные методы обнаружения
- ④ Однопроходные методы обнаружения



Недостатки классификации



Недостатки классификации

- 1 Классификация затруднена, если на одном изображении есть несколько объектов разных типов



Недостатки классификации

- ➊ Классификация затруднена, если на одном изображении есть несколько объектов разных типов
- ➋ Классификация не говорит о том, есть ли на изображении несколько объектов одного типа



Недостатки классификации

- ① Классификация затруднена, если на одном изображении есть несколько объектов разных типов
- ② Классификация не говорит о том, есть ли на изображении несколько объектов одного типа
- ③ Только классификации недостаточно для большинства приложений

Обнаружение объектов

Недостатки классификации

- ① Классификация затруднена, если на одном изображении есть несколько объектов разных типов
- ② Классификация не говорит о том, есть ли на изображении несколько объектов одного типа
- ③ Только классификации недостаточно для большинства приложений

Классификация vs Обнаружение

Главное отличие задачи детектирования от задачи классификации в том, что надо не только сказать есть ли объект на изображении, но и локализовать область нахождения объекта. Если объектов несколько, то требуется найти их все.



Формальное определение обнаружения объектов

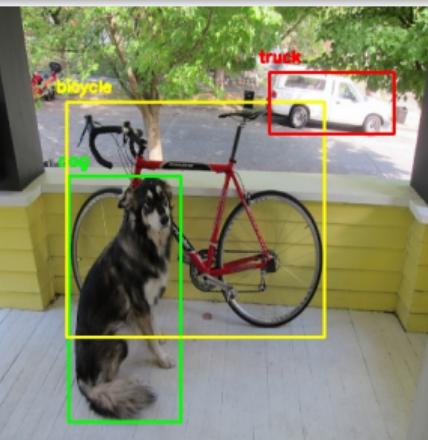
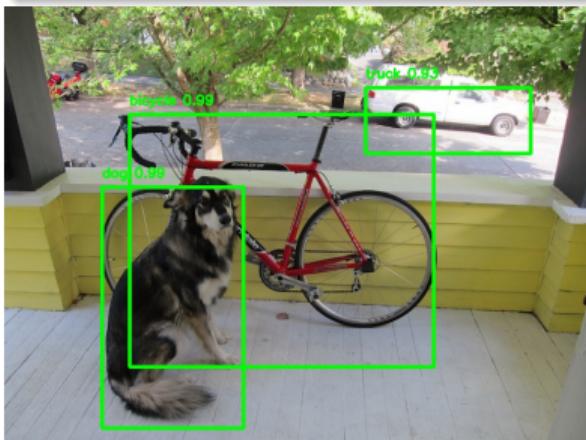
Формальное определение

- Вход: изображение
- Выход (soft): набор троек <тип объекта (object type), уверенность(confidence), прямоугольник (bounding box)>
- Для реальных приложений важнее hard выход: набор пар <тип объекта, прямоугольник>, но для сравнения детекторов нужен soft-выход.

Формальное определение обнаружения объектов

Формальное определение

- Вход: изображение
- Выход (soft): набор троек <тип объекта (object type), уверенность(confidence), прямоугольник (bounding box)>
- Для реальных приложений важнее hard выход: набор пар <тип объекта, прямоугольник>, но для сравнения детекторов нужен soft-выход.



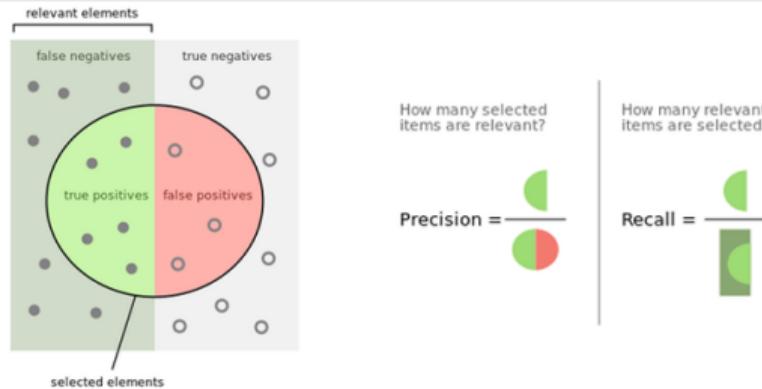
Как понять, что один один метод обнаружения лучше другого?



Mean Average Precision (mAP): Precision, Recall¹

Определение

$$Precision = \frac{TP}{TP+FP}, \quad Recall = \frac{TP}{TP+FN}$$

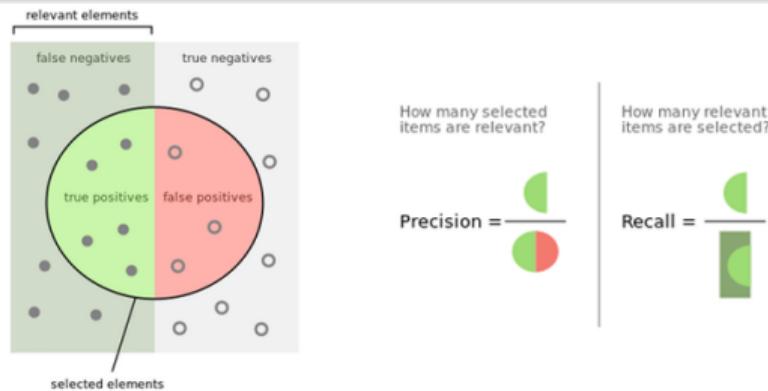


¹https://en.wikipedia.org/wiki/Precision_and_recall

Mean Average Precision (mAP): Precision, Recall¹

Определение

$$Precision = \frac{TP}{TP+FP}, \quad Recall = \frac{TP}{TP+FN}$$



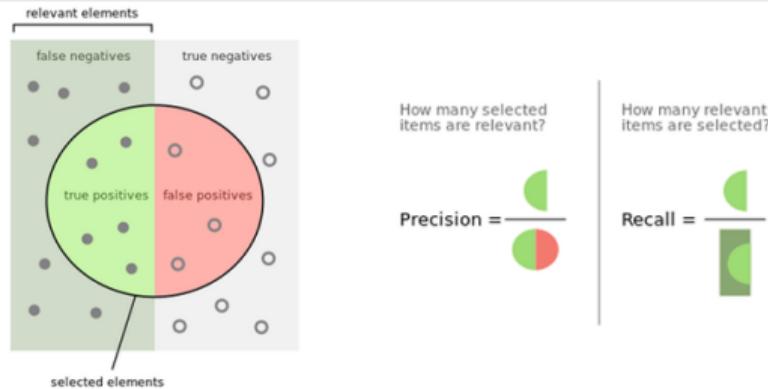
- В случае диагностики заболевания важна полнота (recall), то есть процент найденных заболевших.

¹https://en.wikipedia.org/wiki/Precision_and_recall

Mean Average Precision (mAP): Precision, Recall¹

Определение

$$Precision = \frac{TP}{TP+FP}, \quad Recall = \frac{TP}{TP+FN}$$



- В случае диагностики заболевания важна полнота (recall), то есть процент найденных заболевших.
- В случае распознавания номеров автомобиля для выписывания штрафа важнее точность (precision), чтобы было меньше споров и разбирательств

¹https://en.wikipedia.org/wiki/Precision_and_recall

Mean Average Precision (mAP)

Определение

$$p = \frac{TP}{TP + FP}$$

$$r = \frac{TP}{TP + FN}$$

Определение mAP

$$AP = \int_0^1 p(r)dr$$

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i,$$

где n — количество классов, а AP_i — AP для i -го класса.

TP, FP и FN для задачи обнаружения²

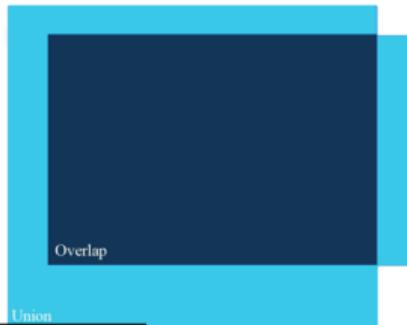
Проблема

Для того, чтобы воспользоваться формулами для mAP необходимо определить TP, FP и FN



- █ Ground truth
- █ Prediction

$$IoU = \frac{\text{area of overlap}}{\text{area of union}}$$



Intersection over Union

$$IoU = \frac{\text{area of overlap}}{\text{area of union}}$$

True Positive

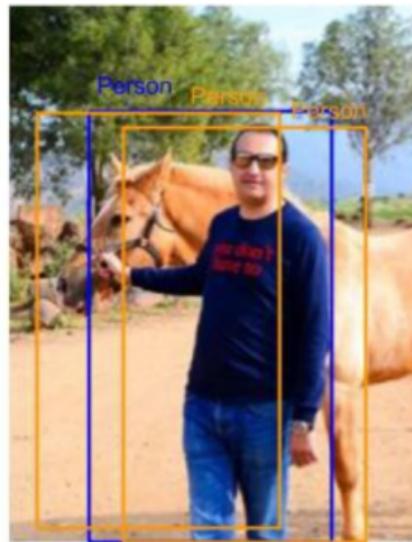
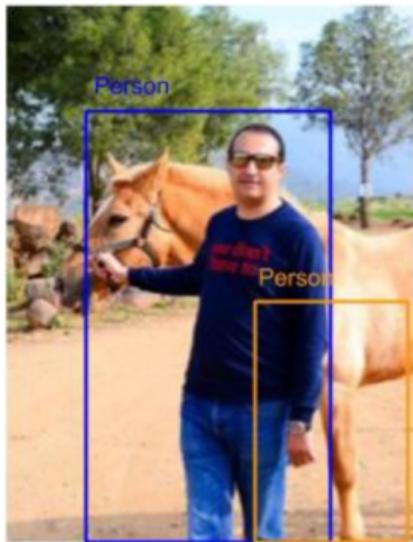
$$IoU > 0.5$$

²https://medium.com/@ionathan_hui/map-mean-average-precision-for-object-detection-45c121a31173
Бабин Д.Н., Иванов И.Е., Петюшко А.А.

TP, FP и FN для задачи обнаружения

False Positive

- $\text{IoU} < 0.5$ или нет пересечения
- Дубликат



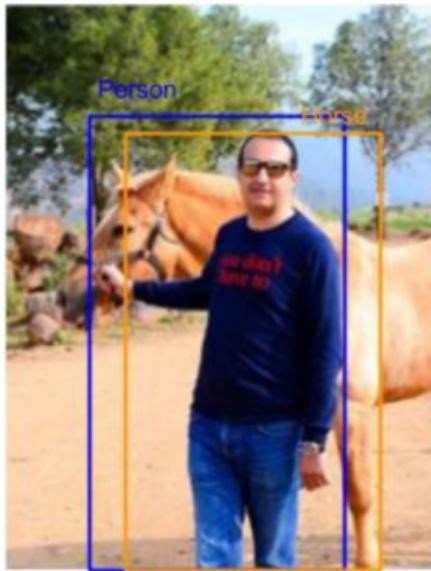
Duplicate BB are considered as FP



No IoU

False Negative

- $\text{IoU} > 0.5$ и неправильно определён класс объекта



Пример расчета AP

Пример

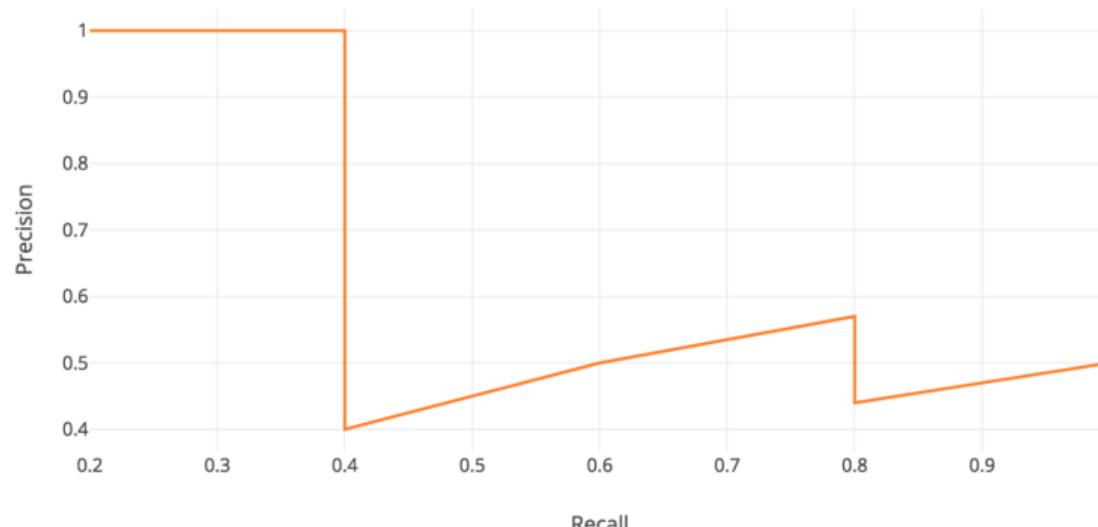
На фотографии изображено 5 яблок. Детектор выдал 10 прямоугольников. Отсортируем их в порядке убывания уверенности алгоритма обнаружения. Получилась таблица, для каждой строчки которой, можно посчитать точность и полноту

Rank	Correct?	Precision	Recall
1	True	1.0	0.2
2	True	1.0	0.4
3	False	0.67	0.4
4	False	0.5	0.4
5	False	0.4	0.4
6	True	0.5	0.6
7	True	0.57	0.8
8	False	0.5	0.8
9	False	0.44	0.8
10	True	0.5	1.0

Пример расчета AP

Пример

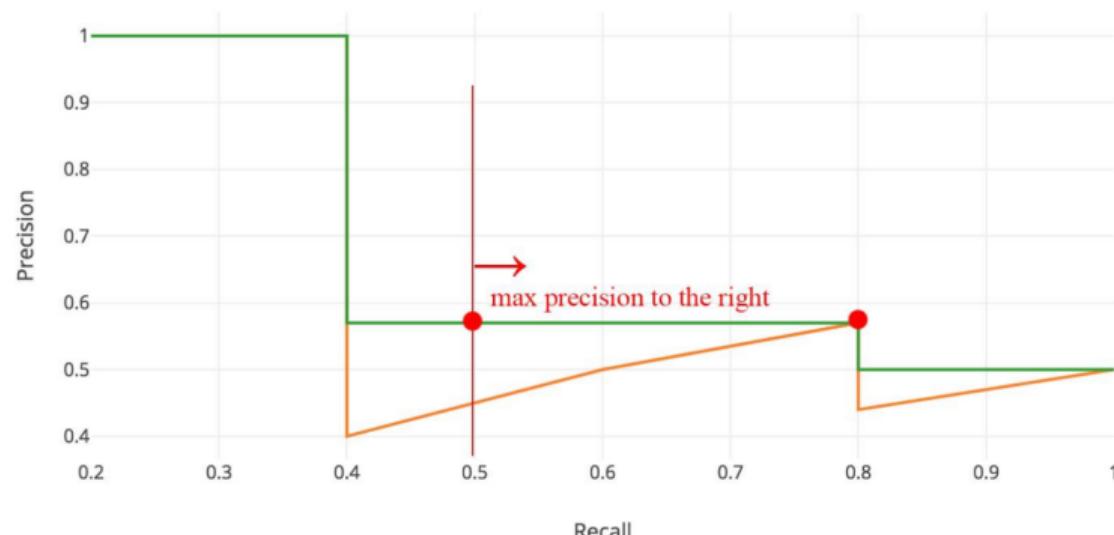
На фотографии изображено 5 яблок. Детектор выдал 10 прямоугольников. Отсортируем их в порядке убывания уверенности алгоритма обнаружения. Получилась таблица, для каждой строчки которой, можно посчитать точность и полноту



Пример расчета AP

Пример

На фотографии изображено 5 яблок. Детектор выдал 10 прямоугольников. Отсортируем их в порядке убывания уверенности алгоритма обнаружения. Получилась таблица, для каждой строчки которой, можно посчитать точность и полноту



Интерполированная точность

На последнем графике была представлена интерполированная точность, которая формально определяется формулой

$$p_{interp}(r) = \max_{\hat{r}: \hat{r} > r} p(\hat{r})$$



Интерполированная точность

На последнем графике была представлена интерполированная точность, которая формально определяется формулой

$$p_{interp}(r) = \max_{\hat{r}: \hat{r} > r} p(\hat{r})$$

Интуиция: более стабильная значение при небольших колебаниях уверенности



Варианты подсчета AP

Интерполированная точность

На последнем графике была представлена интерполированная точность, которая формально определяется формулой

$$p_{interp}(r) = \max_{\hat{r}: \hat{r} > r} p(\hat{r})$$

Интуиция: более стабильная значение при небольших колебаниях уверенности

Приближенное вычисление интеграла

Иногда используют следующую формулу для подсчета AP (PASCAL VOC)

$$AP = \frac{1}{11} \sum_{r \in \{0.0, 0.1, \dots, 1.0\}} p_{interp}(r)$$

Варианты подсчета AP

Интерполированная точность

На последнем графике была представлена интерполированная точность, которая формально определяется формулой

$$p_{interp}(r) = \max_{\hat{r}: \hat{r} > r} p(\hat{r})$$

Интуиция: более стабильная значение при небольших колебаниях уверенности

Приближенное вычисление интеграла

Иногда используют следующую формулу для подсчета AP (PASCAL VOC)

$$AP = \frac{1}{11} \sum_{r \in \{0.0, 0.1, \dots, 1.0\}} p_{interp}(r)$$

В более поздних версиях соревнований использовалась точная формула

Усреднение AP по различным порогам IoU

$$AP = \frac{1}{10}(AP_{0.5} + AP_{0.55} + AP_{0.6} + AP_{0.65} + AP_{0.7} + AP_{0.75} + AP_{0.8} + AP_{0.85} + AP_{0.9} + AP_{0.95})$$



Average Recall (AR)

AR

AR — это среднее значение полноты в зависимости от IoU

$$AR = 2 \int_{0.5}^{1.0} R(IoU) d(IoU)$$



Идея

Детектор состоит из следующих модулей:

- ❶ Предсказыватель объектов (около 2000 кандидатов на изображение)

³R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014, <https://arxiv.org/pdf/1311.2524.pdf>

Идея

Детектор состоит из следующих модулей:

- ① Предсказыватель объектов (около 2000 кандидатов на изображение)
- ② Классификатор (без классификационной головы) для извлечения признаков для каждого кандидата

³R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014, <https://arxiv.org/pdf/1311.2524.pdf>

Нейросетевой детектор: R-CNN (Regions with CNN features)³

Идея

Детектор состоит из следующих модулей:

- ① Предсказыватель объектов (около 2000 кандидатов на изображение)
- ② Классификатор (без классификационной головы) для извлечения признаков для каждого кандидата
- ③ Линейный SVM для классификации кандидата

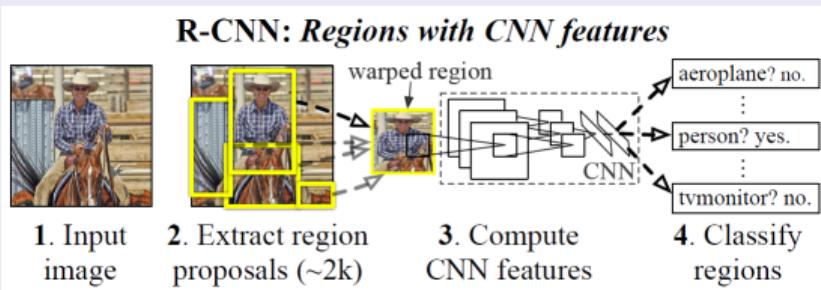
³R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014, <https://arxiv.org/pdf/1311.2524.pdf>

Нейросетевой детектор: R-CNN (Regions with CNN features) 3

Идея

Детектор состоит из следующих модулей:

- 1 Предсказыватель объектов (около 2000 кандидатов на изображение)
 - 2 Классификатор (без классификационной головы) для извлечения признаков для каждого кандидата
 - 3 Линейный SVM для классификации кандидата



³R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014, <https://arxiv.org/pdf/1311.2524.pdf>

R-CNN: Bounding Box Regression

Уточнение обнаружения: постановка задачи

Пусть $P^i = (P_x^i, P_y^i, P_w^i, P_h^i)$ — результаты алгоритма предсказывания детекций. Тогда будем искать

$$\begin{aligned}\hat{G}_x &= P_w d_x(P) + P_x, \quad \hat{G}_y = P_h d_y(P) + P_y, \\ \hat{G}_w &= P_w \exp(d_w(P)), \quad \hat{G}_h = P_h \exp(d_h(P)),\end{aligned}$$

где $d_*(P) = w_*^T \varphi(P)$ — линейное преобразование на признаках для P

Решение: гребневая регрессия

$$w_* = \arg \min_{\hat{w}_*} \sum_i (t_*^i - \hat{w}_*^T \varphi(P^i))^2 + \lambda \|\hat{w}_*\|^2,$$

где $t_x = \frac{G_x - P_x}{P_w}$, $t_y = \frac{G_y - P_y}{P_h}$, $t_w = \log(\frac{G_w}{P_w})$, $t_h = \log(\frac{G_h}{P_h})$

Алгоритм предсказания детекций

Алгоритм selective search ^a не настраивается на данные

^aJ. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. IJCV, 2013

Алгоритм предсказания детекций

Алгоритм selective search ^a не настраивается на данные

^aJ. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. IJCV, 2013

Backbone для извлечения признаков

- ❶ В качестве начально инициализации берётся классификационная сеть обученная на 1000 классах из ImageNet
- ❷ Удаляется последний слой и заменяется на слой нужного размера со случайной инициализацией (добавляется новый класс фон)
- ❸ Обучение происходит тех изображениях, у которых $\text{IoU} > 0.5$ с разметкой, остальные рассматриваются, как негативные

Классификатор категорий

- ① Для каждого класса обучается свой линейный SVM (one vs rest)
- ② Для отделения негативных примеров используется другой порог — $\text{IoU} < 0.3$ (параметр подбирается по валидации и от него очень сильно зависит финальный результат)



Классификатор категорий

- ➊ Для каждого класса обучается свой линейный SVM (one vs rest)
- ➋ Для отделения негативных примеров используется другой порог — $\text{IoU} < 0.3$ (параметр подбирается по валидации и от него очень сильно зависит финальный результат)

Корректировка детекций (bounding box regression)

- ➌ Гребневая регрессия



R-CNN: Результаты

Detector	VOC2007, test, mAP
R-CNN (AlexNet)	54.2
R-CNN (AlexNet) BB	58.5
R-CNN (VGG)	62.2
R-CNN (VGG) BB	66.0

Достоинства

- ① Один из лучших методов на момент написания статьи
- ② Bounding box regression улучшает качество
- ③ Замена backbone на более продвинутый улучшает качество



Достоинства

- ① Один из лучших методов на момент написания статьи
- ② Bounding box regression улучшает качество
- ③ Замена backbone на более продвинутый улучшает качество

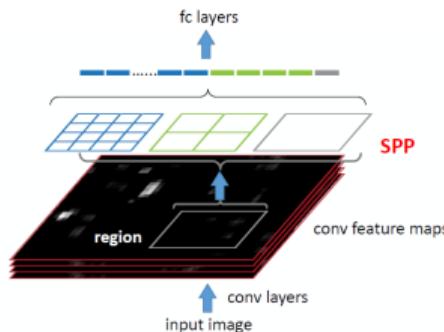
Недостатки

- ① Самый главный недостаток — скорость работы (огромный overhead по вычислениям, так как одни и те же куски картинки обрабатываются по много раз)
- ② Сложное многоэтапное обучение
- ③ Обучение требует много дискового пространства и вычислительных ресурсов
- ④ Selective search (алгоритм поиска кандидатов) — необучаемый алгоритм

Нейросетевой детектор: SPPnet⁴

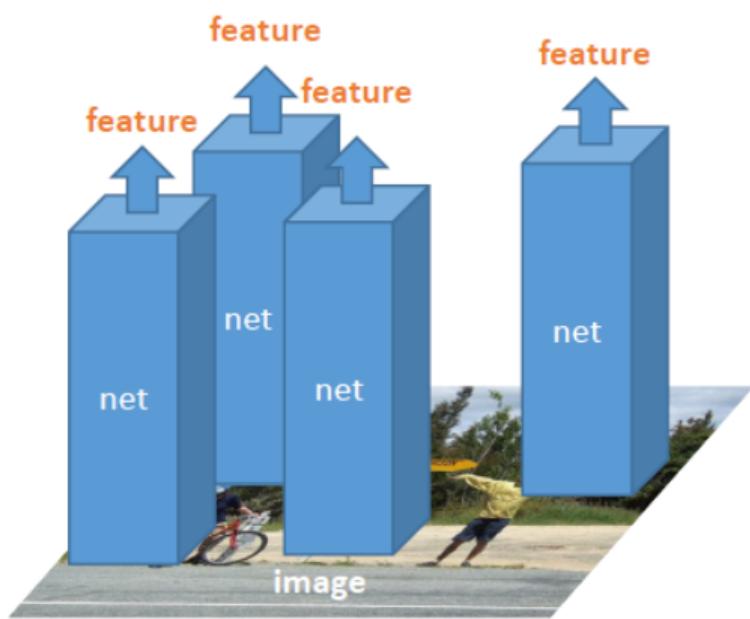
Идея

- ➊ Новый слой Spatial Pyramid Pooling (SPP) позволяет обрабатывать изображения разного размера
- ➋ Это позволяет делать обучение с разным размером входа
- ➌ Это даёт улучшение на задаче классификации и значительное ускорение при использовании R-CNN



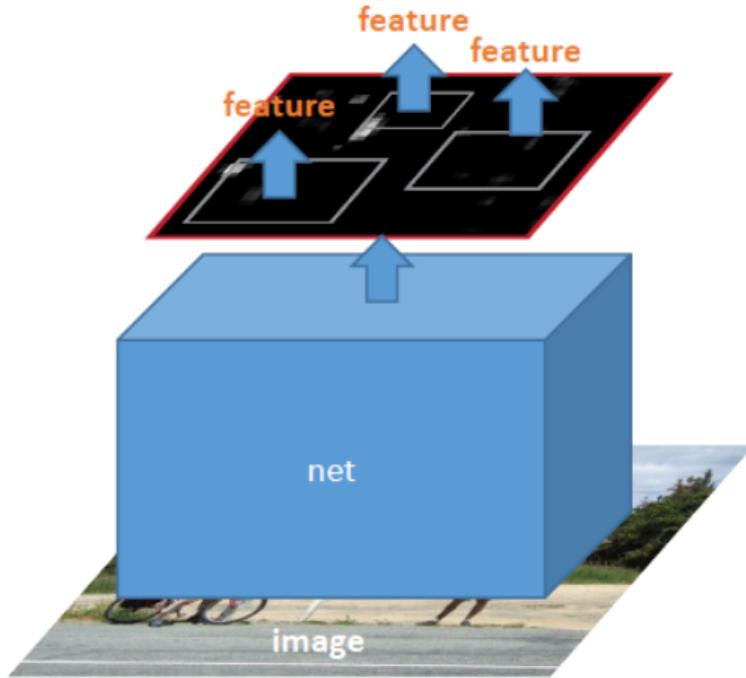
⁴K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV, 2014, <https://arxiv.org/pdf/1506.01497.pdf>

R-CNN vs SPPnet⁵



R-CNN

2000 nets on image regions



SPP-net

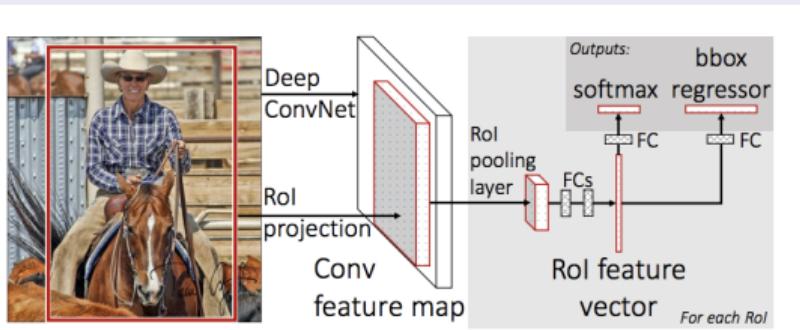
1 net on full image



Нейросетевой детектор: Fast R-CNN⁶

Описание алгоритма

- 1 Вход: изображение и предсказания объектов
- 2 Сначала картинка подаётся на несколько свёрточных слоев
- 3 После чего из полученных признаков и предсказаний объектов специальным слоем (ROI pooling) извлекаются признаки
- 4 По полученным признакам делается классификация и коррекция обнаружений



⁶R. Girshick, "Fast R-CNN," in IEEE International Conference on Computer Vision (ICCV), 2015,
<https://arxiv.org/pdf/1506.01497.pdf>



Детали обучения

В качестве лосса используется линейная комбинация классификационного лосса и регрессионного лосса для коррекций детекции



Fast R-CNN: Результаты

Detector	VOC2007, test, mAP
R-CNN (AlexNet)	54.2
R-CNN (AlexNet) BB	58.5
R-CNN (VGG)	62.2
R-CNN (VGG) BB	66.0
SPPnet BB	63.1
Fast R-CNN	66.9

Достоинства и недостатки Fast R-CNN

Достоинства

- ① По сравнению с R-CNN на порядок быстрее с более высоким качеством обнаружения
- ② Обучение стало происходить в один этап

Недостатки

- ① Двух-этапный метод обнаружения
- ② Selective search (алгоритм поиска кандидатов) — не настраиваемый на данные алгоритм



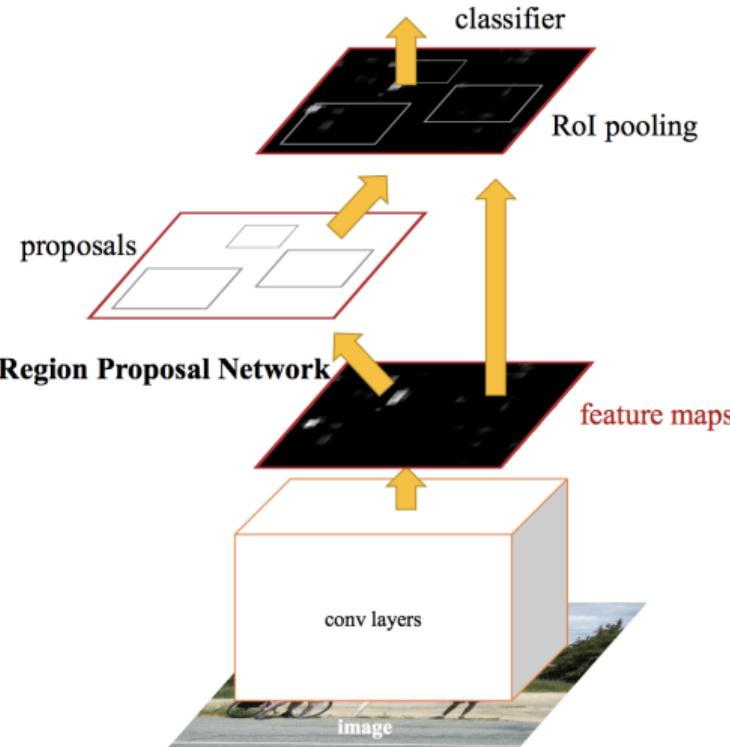
Идея

- ① Faster R-CNN = RPN + Fast R-CNN
- ② RPN (Region Proposal Network) — предсказывает регионы, где могут находиться объекты
- ③ Так как и RPN и Fast R-CNN содержат свёрточные слои, то на начальном этапе разумно использовать общие свёртки

⁷S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, <https://arxiv.org/pdf/1506.01497.pdf>



Схема Faster R-CNN⁸



⁸<https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>

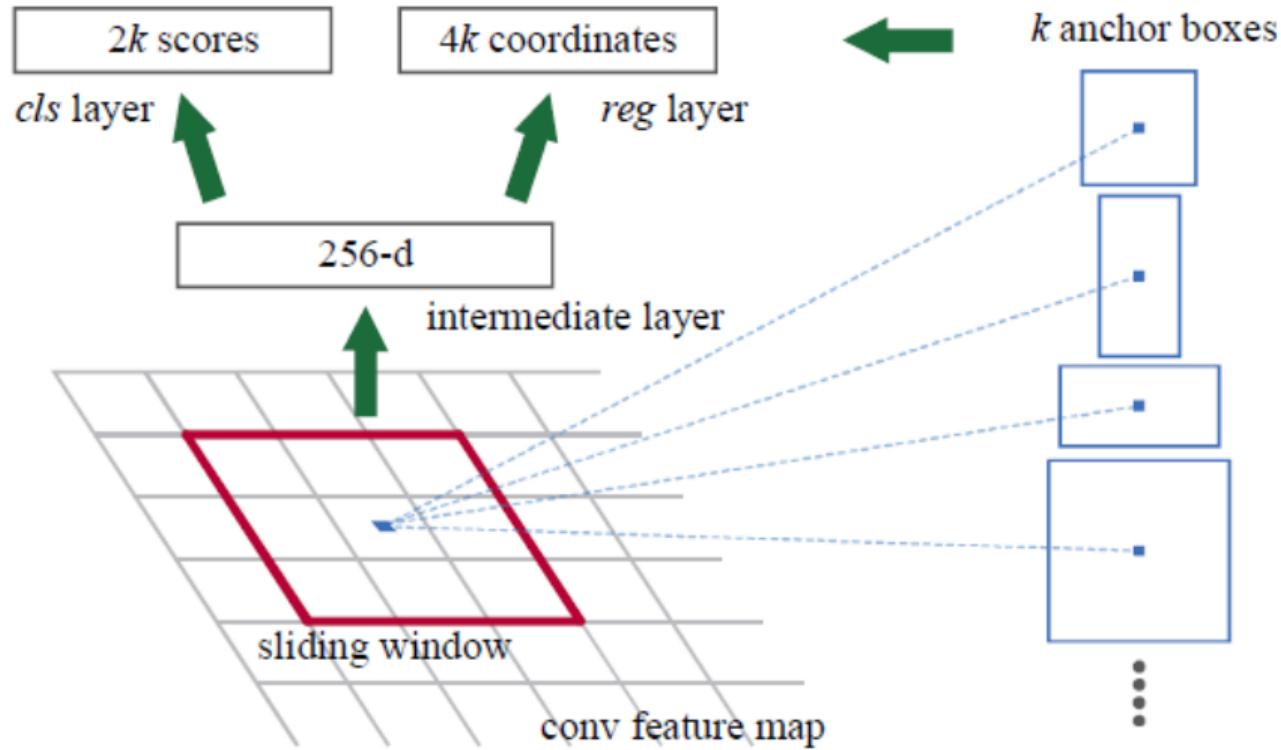
Faster R-CNN: Region Proposal Network

- ➊ Изображение подаётся на выход нескольким свёрточным слоям для извлечения признаков
- ➋ Затем скользящим окном для 9 размеров (3 соотношения сторон + 3 масштаба) генерируются признаки фиксированного размера для предсказания детекций
- ➌ Далее признаки подаются на следующий слой, где для каждого прямоугольника считается вероятность того, есть ли объект внутри
- ➍ Так же из полученных признаков считаются уточняющие коэффициенты для размеров прямоугольника

Функция потерь и обучение

- ➊ В качестве лосса используется линейная комбинация классификационного лосса и регрессионного лосса для коррекций детекции
- ➋ В батче имеются позитивные и негативные примеры в равном соотношении

Faster R-CNN: Region Proposal Network



Этапы обучения

- ① Обучение RPN из предобученной модели на ImageNet
- ② Обучение детекционной модели из предобученной модели на ImageNet (пока оба обучения независимы)
- ③ Инициализация общих слоёв весами из детекционной сети. Дообучение остальных слоёв RPN. Первые слои не меняются
- ④ Дообучение последних слоёв детекционной сети



Faster R-CNN: Результаты

Detector	VOC2007, test, mAP
R-CNN (AlexNet)	54.2
R-CNN (AlexNet) BB	58.5
R-CNN (VGG)	62.2
R-CNN (VGG) BB	66.0
SPPnet BB	63.1
Fast R-CNN	66.9
Faster R-CNN	69.9
Faster R-CNN(other training data)	78.8



Нейросетевой детектор: YOLO⁹

Идея

- 1 Изображение покрывается сеткой $S \times S$
- 2 Выход сети представляет собой тензор размера $S \times S \times (B * 5 + C)$, где B — количество детекций с центром в этой ячейке, C — количество классов
- 3 Детекция — это вектор $(x, y, w, h, confidence)$
- 4 Для всех детекций ячейки вычисляется только одно распределение по классам
- 5 Каждая ячейка предсказывает только один объект



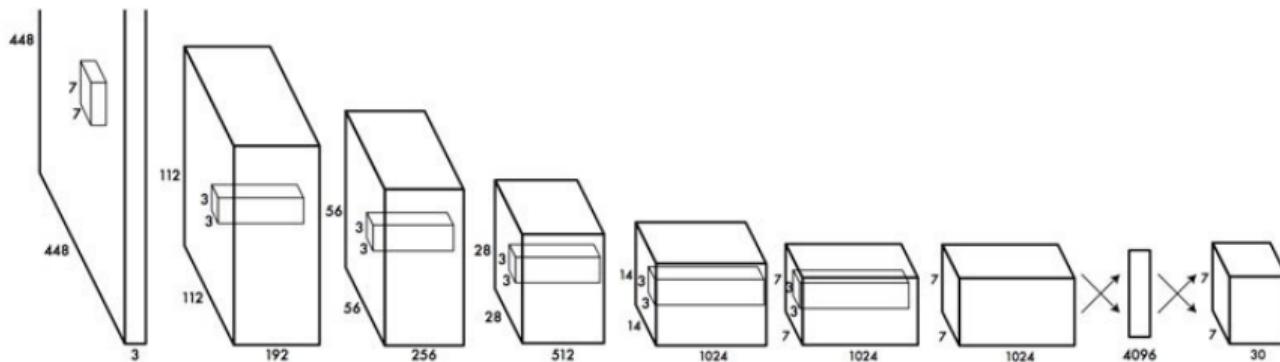
⁹ J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection, 2015, <https://arxiv.org/pdf/1506.02640.pdf>

YOLO: Архитектура

Pascal VOC

Типичные параметры: $S = 7$, $B = 2$, $C = 20$

Финальные предсказания — это тензор $7 \times 7 \times 30$



YOLO: функция потерь

$$\begin{aligned} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$



Достоинства и недостатки YOLO

Достоинства

- ① Быстрый алгоритм, может работать в режиме реального времени
- ② Все предсказания даёт одна нейронная сеть, которая обучается end2end
- ③ Использует для предсказания всё изображение
- ④ Обладает хорошей обобщающей способностью

Недостатки

- ① Ограниченнное количество обнаружений на ячейку



YOLO: Результаты

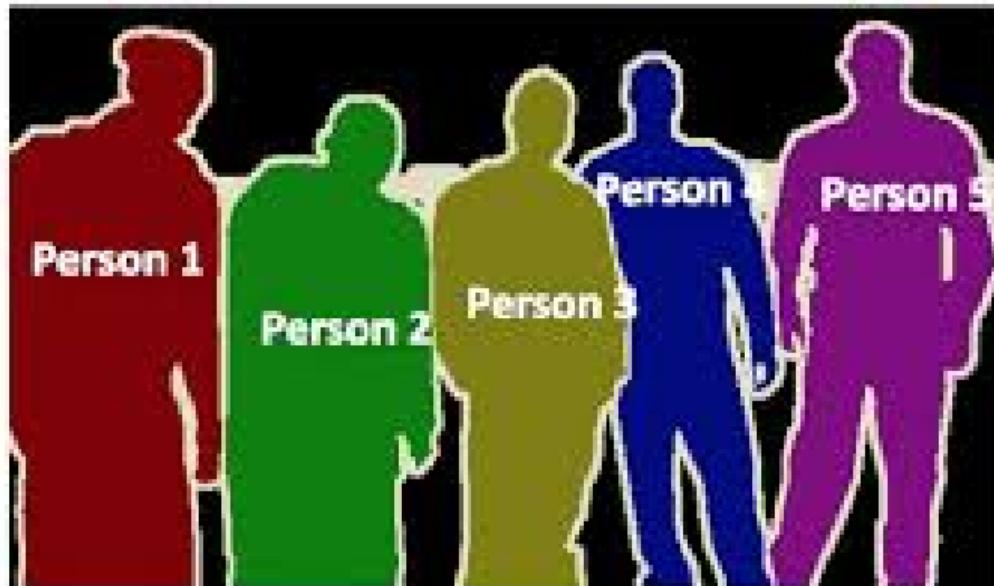
Detector	VOC2007, test, mAP
R-CNN (AlexNet)	54.2
R-CNN (AlexNet) BB	58.5
R-CNN (VGG)	62.2
R-CNN (VGG) BB	66.0
SPPnet BB	63.1
Fast R-CNN	66.9
Faster R-CNN	69.9
Faster R-CNN(other training data)	78.8
YOLO v1	63.4
YOLO v2	78.2

Postprocessing: Non-Maximal Suppression

YOLO, как и большинство детекторов, может дублировать обнаружения для одного и того же объекта. Чтобы исправить это, обычно применяют алгоритм *NMS* для удаления дубликатов. Такой постпроцессинг добавляет 2-3 % тАР.

Типичная реализация алгоритма выглядит следующим образом:

- ❶ Отсортировать предсказания по уверенности
- ❷ Начиная с самых больших показателей, идем по предсказаниям и удаляем обнаружения с тем же классом и $\text{IoU} > 0.5$ с предыдущими предсказаниями



Instance Segmentation

Развитие задачи обнаружения: Pose estimation / Key point detection



- ① Обнаружение объектов — важная с практической точки зрения и востребованная в индустрии задача

Заключение

- ① Обнаружение объектов — важная с практической точки зрения и востребованная в индустрии задача
- ② Современные подходы делятся на двухпроходные (Faster R-CNN) и однопроходные (YOLO)

Заключение

- ① Обнаружение объектов — важная с практической точки зрения и востребованная в индустрии задача
- ② Современные подходы делятся на двухпроходные (Faster R-CNN) и однопроходные (YOLO)
- ③ Существуют довольно быстрые и качественные решения, которые работают даже на мобильных телефонах



Заключение

- ➊ Обнаружение объектов — важная с практической точки зрения и востребованная в индустрии задача
- ➋ Современные подходы делятся на двухпроходные (Faster R-CNN) и однопроходные (YOLO)
- ➌ Существуют довольно быстрые и качественные решения, которые работают даже на мобильных телефонах
- ➍ Всё больше исследователей смотрят в сторону быстродействия, нежели улучшения качества

Заключение

- ➊ Обнаружение объектов — важная с практической точки зрения и востребованная в индустрии задача
- ➋ Современные подходы делятся на двухпроходные (Faster R-CNN) и однопроходные (YOLO)
- ➌ Существуют довольно быстрые и качественные решения, которые работают даже на мобильных телефонах
- ➍ Всё больше исследователей смотрят в сторону быстродействия, нежели улучшения качества
- ➎ Задача имеет варианты развития — обнаружение ключевых точек объекта, более точная локализация объекта (сегментация)

Спасибо за внимание!

