



Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского
Институт информационных технологий, математики и механики

Образовательный курс
«Современные методы и технологии глубокого
обучения в компьютерном зрении»

Детектирование объектов на изображениях

При поддержке компании Intel

Гетманская Александра, Кустикова Валентина

Содержание

- ❑ Цель лекции
- ❑ Постановка задачи детектирования объектов на изображениях
- ❑ Открытые наборы данных
- ❑ Показатели качества детектирования
- ❑ Глубокие модели для детектирования объектов
- ❑ Сравнение моделей детектирования объектов
- ❑ Заключение



Цель лекции

- **Цель** – рассмотреть модели глубокого обучения для задачи детектирования объектов



ПОСТАНОВКА ЗАДАЧИ ДЕТЕКТИРОВАНИЯ ОБЪЕКТОВ



Постановка задачи (1)

- ❑ Задача детектирования объектов состоит в том, чтобы определить положение прямоугольника, окаймляющего объект заданного класса

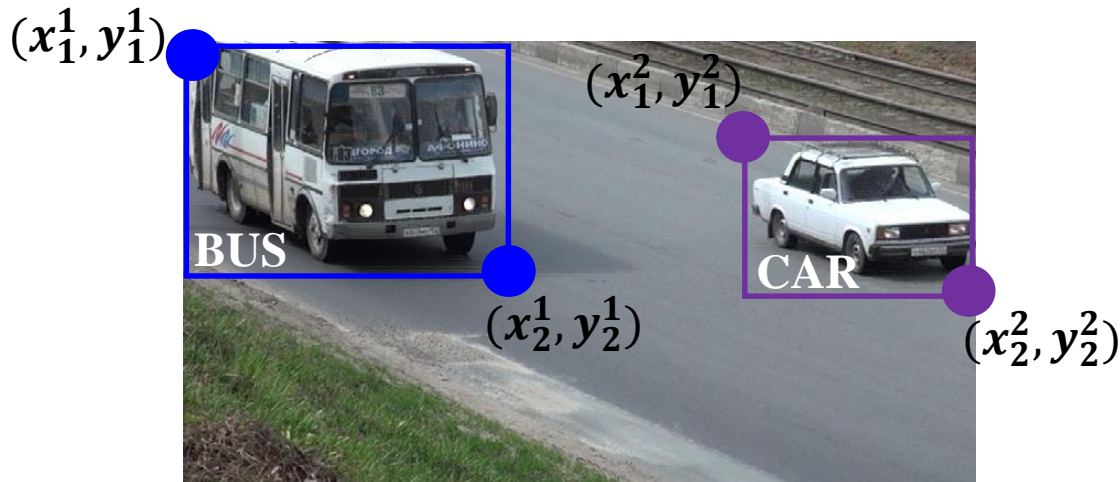


Постановка задачи (2)

- Задача детектирования состоит в том, чтобы каждому изображению I поставить в соответствие множество положений объектов B интересующих классов:

$$\varphi: I \rightarrow B, \quad B = \{b_k, k = \overline{0, |B| - 1}\},$$

где $b_k = ((x_1^k, y_1^k), (x_2^k, y_2^k)[, s^k, c^k])$, $s^k \in \mathbb{R}$ – достоверность,
 c^k – класс объектов («СТОЛ», «ПЕШЕХОД», «АВТОМОБИЛЬ», «АВТОБУС» и т.п.)



ОТКРЫТЫЕ НАБОРЫ ДАННЫХ



Наборы данных (1)

Набор данных	Размер тренировочного множества		Размер валидационного множества		Размер тестового множества		Кол-во классов
	Изображения	Объекты	Изображения	Объекты	Изображения	Объекты	
Детектирование объектов реальной жизни							
PASCAL VOC 2007 [http://host.robots.ox.ac.uk/pascal/VOC/voc2007]	2 501	6 301	2 510	6 307	4 952	12 032	20
PASCAL VOC 2012 [http://host.robots.ox.ac.uk/pascal/VOC/voc2012]	5 717	13 609	5 823	13 841	N/A	N/A	20
MS COCO [http://cocodataset.org]	165 482	N/A	81 208	N/A	81 434	N/A	91



Наборы данных (2)

Набор данных	Размер тренировочного множества		Размер валидационного множества		Размер тестового множества		Кол-во классов
	Изображения	Объекты	Изображения	Объекты	Изображения	Объекты	
Детектирование объектов реальной жизни							
Open Images Dataset [https://storage.googleapis.com/openimages/web/index.html]	~1,7 млн.	~1,4 млн.	~40 тыс.	~204 тыс.	~125 тыс.	~625 тыс.	600
Детектирование и распознавание лиц							
WIDER FACE [http://shuoyang1213.me/WIDERFACE]	12 881	~157 тыс.	3 220	~39 тыс.	16 102	~196 тыс.	1



Наборы данных (3)

Набор данных	Размер тренировочного множества		Размер валидационного множества		Размер тестового множества		Кол-во классов
	Изображения	Объекты	Изображения	Объекты	Изображения	Объекты	
Детектирование и распознавание лиц							
LFW [http://vis-www.cs.umass.edu/lfw]	11 910	~5 тыс.	0	0	1 323	~700	1
AFLW [https://www.tugraz.at/institute/icg/research/team-bischof/lrs/downloads/aflw]	25 тыс.	25 тыс. * 21	—	—	—	—	21
IMDB-WIKI [https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki]	523 051	~500 тыс.	—	—	—	—	1



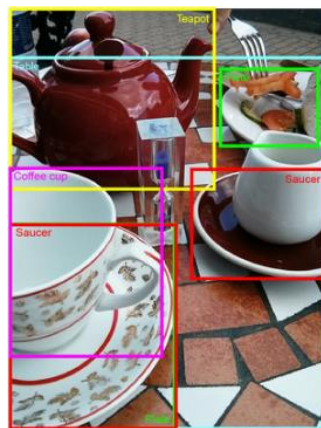
Наборы данных (4)

Набор данных	Размер тренировочного множества		Размер валидационного множества		Размер тестового множества		Кол-во классов
	Изображения	Объекты	Изображения	Объекты	Изображения	Объекты	
Детектирование пешеходов							
Caltech [http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians]	~57 видео по 1 мин	~ 175 тыс.	—	—	~ 47 видео по 1 мин	~ 175 тыс.	1
Wider Person [http://www.cbsr.jia.ac.cn/users/sfzhang/WiderPerson]	8 000	~ 240 тыс.	1 000	~ 30 тыс.	4 382	~ 130 тыс.	1



Open Images Dataset

- ❑ 15 851 536 объектов, принадлежащих 600 категориям
- ❑ Если более 5 экземпляров объектов одного класса сильно перекрывают друг друга, они заключаются в один прямоугольник с меткой «группа объектов»
- ❑ Все прямоугольники размечены вручную



* Open Images Dataset [<https://storage.googleapis.com/openimages/web/index.html>].

** Kuznetsova A., Rom H., Alldrin N., Uijlings J., Krasin I., Pont-Tuset J., Kamali S., Popov S., Mallocci M., Kolesnikov A., Duerig T., Ferrari V. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. – 2020. – [<https://arxiv.org/pdf/1811.00982.pdf>].

WIDER FACE

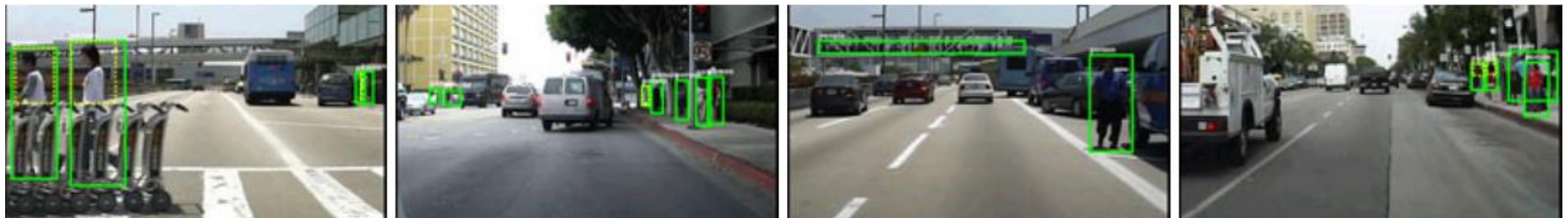
- ❑ WIDER FACE – бенчмарк для сравнения качества работы методов детектирования лиц на изображениях
- ❑ 32 203 изображения, на которых отмечено 393 703 лица с высокой степенью изменчивости в масштабе, позе и перекрытии



* WIDER FACE [<http://shuoyang1213.me/WIDERFACE>].

Caltech Pedestrian Dataset

- ❑ 10 часов видео с разрешением 640x480 и частотой 30 Гц. Видео получено с видеорегистратора, который установлен на автомобиле, движущемся в городских условиях
- ❑ ~250 000 аннотированных кадров (в 137 отрезках длиной около минуты), содержащих 350 000 окаймляющих прямоугольников и 2 300 уникальных пешеходов



* Caltech Pedestrian Dataset [http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians].



ПОКАЗАТЕЛИ КАЧЕСТВА ДЕТЕКТИРОВАНИЯ ОБЪЕКТОВ



Рассматриваемые показатели качества

- ❑ Показатель числа истинных срабатываний (true positive rate)
- ❑ Показатель числа ложных срабатываний (false detection rate)
- ❑ Количество ложных срабатываний, в среднем приходящихся на изображение (average false positives per frame)
- ❑ Средняя точность предсказания (average precision)



Показатель числа истинных срабатываний

- ❑ **Показатель числа истинных срабатываний** (true positive rate) – отношение количества правильно обнаруженных объектов TP к общему числу размеченных объектов $TP + FN$

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

- ❑ Считается, что объект обнаружен правильно, если доля перекрытия обнаруженного (detection, d) и размеченного (groundtruth, g) окаймляющих прямоугольников $IoU = \frac{S_{d \cap g}}{S_{d \cup g}}$ превышает некоторое пороговое значение τ

- ❑ Порог τ выбирается в промежутке от 0.5 до 0.7

		Предсказание	
		True	False
Разметка	True	TP	FN
	False	FP	TN



Показатель числа ложных срабатываний

- ❑ **Показатель числа ложных срабатываний** (false detection rate) – отношение количества ложных срабатываний к общему числу срабатываний детектора

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

- ❑ Объект считается обнаруженным правильно при выполнении тех же условий, что и для предыдущего показателя
- ❑ Обнаруженный прямоугольник принимается за ложное срабатывание, если ему не нашлась пара из разметки

		Предсказание	
		True	False
Разметка	True	TP	FN
	False	FP	TN



Количество ложных срабатываний, в среднем приходящихся на изображение

- ❑ **Количество ложных срабатываний, в среднем приходящихся на изображение** (average false positives per frame) – отношение количества ложных срабатываний FP к общему числу обработанных изображений N

$$FP_{perFrame} = \frac{FP}{N}$$

- ❑ Объект считается обнаруженным правильно при выполнении тех же условий, что и для предыдущих показателей
- ❑ Показатель представляет интерес при обработке потока изображений (например, видео)

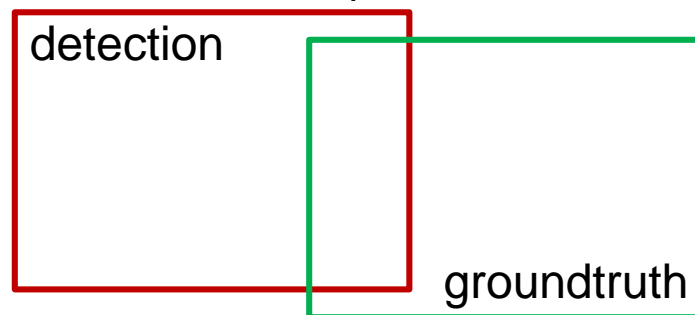
		Предсказание	
		True	False
Разметка	True	TP	FN
	False	FP	TN



Средняя точность предсказания (1)

□ Обозначения:

- $IoU = \frac{S_{d \cap g}}{S_{d \cup g}}$ – доля перекрытия обнаруженного (detection) и размеченного (groundtruth) окаймляющих прямоугольников (Intersection over Union), $IoU \in [0; 1]$
- TP – количество объектов, для которых доля перекрытия не меньше некоторого порога τ (т.е. считается, что объект обнаружен правильно – true positive)
- FP – количество обнаруженных объектов с долей перекрытия, меньшей τ (объект найден ошибочно), или объект обнаружен более одного раза (false positives)
- FN – количество необнаруженных объектов (false negatives)



Средняя точность предсказания (2)

- Пороговое значение τ , как правило, выбирается равным 0.5
- **Точность** (precision) – отношение количества правильно обнаруженных прямоугольников к общему числу срабатываний детектора

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

- **Отклик** (recall) – отношение количества правильно обнаруженных прямоугольников к общему числу объектов

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



Средняя точность предсказания (3)

- **Средняя точность предсказания** (average precision) – математическое ожидание точностей

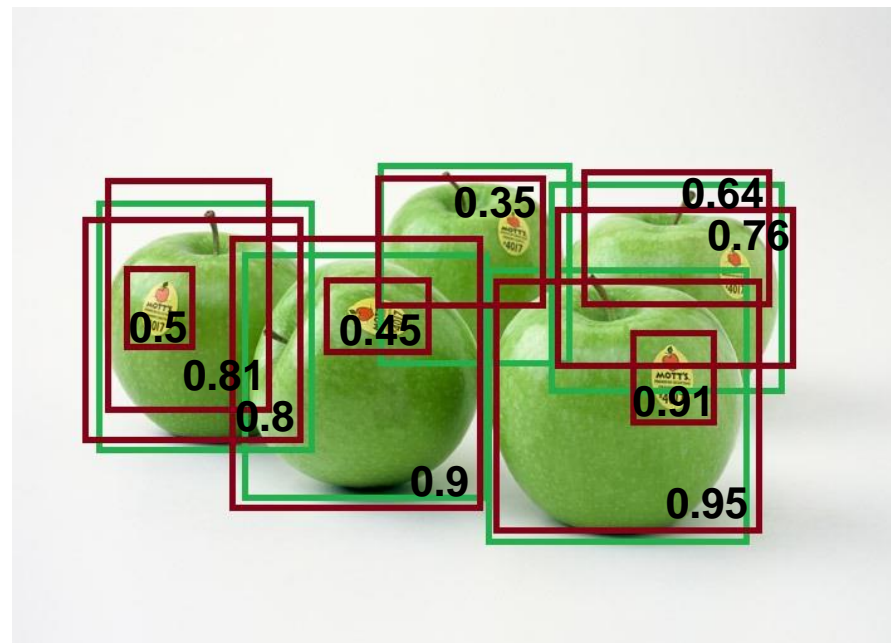
$$AP = \int_0^1 p(r) dr$$

- Схема вычисления:
 - Обнаруженные окаймляющие прямоугольники сортируются в порядке убывания достоверности наличия в них объектов
 - Для каждого обнаруженного прямоугольника выполняется поиск соответствия из разметки согласно условию $IoU \geq \tau$
 - Выполняется вычисление точности и отклика
 - Строится зависимость точности от отклика
 - Вычисляется площадь под графиком построенной зависимости



Средняя точность предсказания (4)

- Пример вычисления средней точности предсказания:
 - Исходное изображение – фотография яблок из набора данных ImageNet [<http://www.image-net.org>]
 - Разметка содержит окаймляющие прямоугольники для 5 яблок (зеленые прямоугольники)
 - Алгоритм детектирования обнаруживает 10 яблок (красные прямоугольники)
 - Для определенности предполагается, что достоверности различны, чтобы далее однозначно идентифицировать прямоугольники

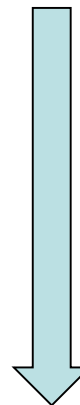


Средняя точность предсказания (5)

- Пример вычисления средней точности предсказания:
 - Сортировка прямоугольников, вычисление точности и отклика

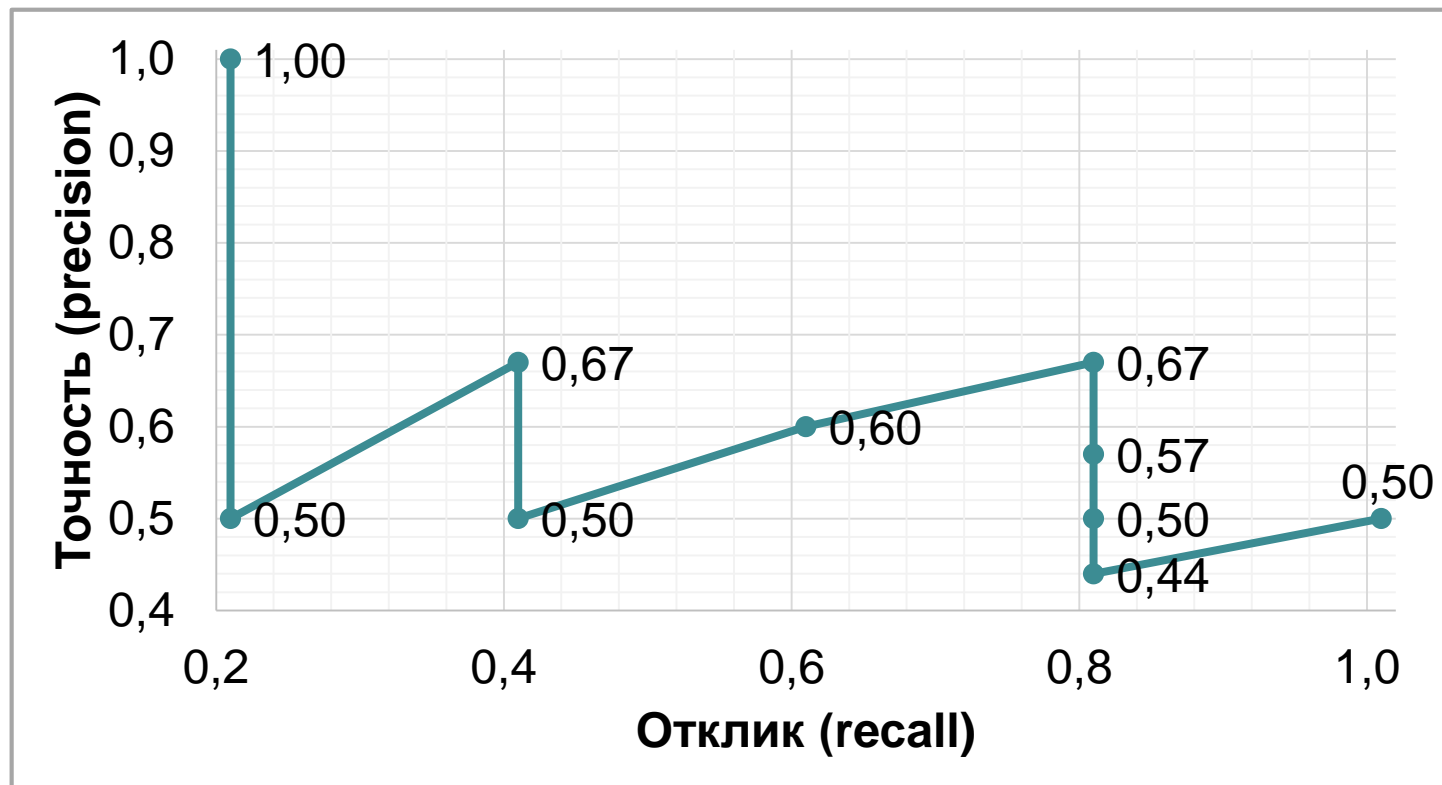
№	Достоверность	Объект?	Точность	Отклик
1	0.95	Да	$1/1 = 1.0$	$1/5 = 0.2$
2	0.91	Нет	$1/2 = 0.5$	$1/5 = 0.2$
3	0.9	Да	$2/3 \approx 0.67$	$2/5 = 0.4$
4	0.81	Нет	$2/4 = 0.5$	$2/5 = 0.4$
5	0.8	Да	$3/5 = 0.6$	$3/5 = 0.6$
6	0.76	Да	$4/6 \approx 0.67$	$4/5 = 0.8$
7	0.64	Нет	$4/7 \approx 0.57$	$4/5 = 0.8$
8	0.5	Нет	$4/8 = 0.5$	$4/5 = 0.8$
9	0.45	Нет	$4/9 \approx 0.44$	$4/5 = 0.8$
10	0.35	Да	$5/10 = 0.5$	$5/5 = 1.0$

Отклик
нарастает



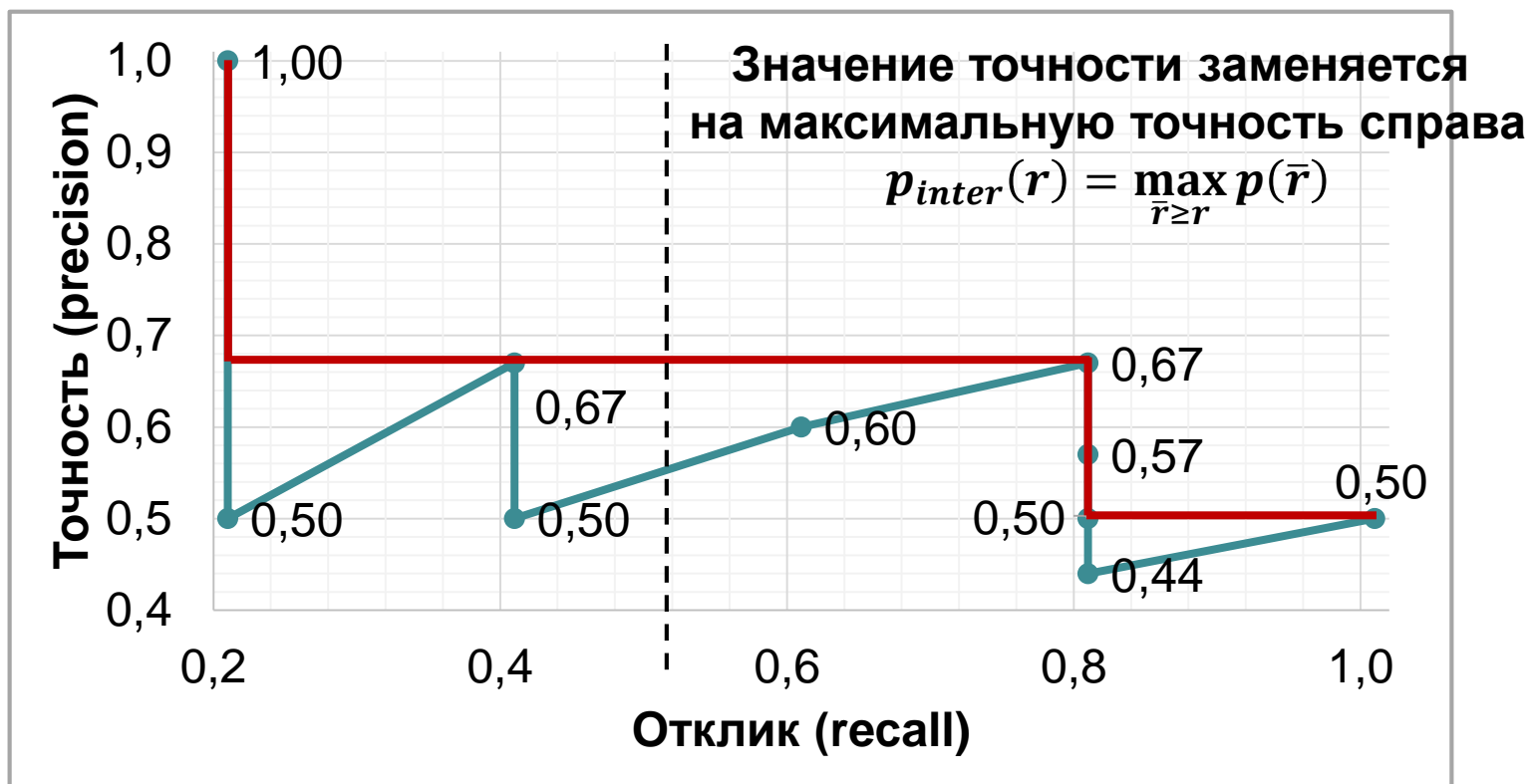
Средняя точность предсказания (6)

- Пример вычисления средней точности предсказания:
 - Построение зависимости точности от отклика
 - Результат – зигзагообразная кривая



Средняя точность предсказания (7)

- Пример вычисления средней точности предсказания:
 - Вычисление площади под зигзагообразной кривой – интерполяция и вычисление площади под «ступенькой»



ГЛУБОКИЕ МОДЕЛИ ДЛЯ ДЕТЕКТИРОВАНИЯ ОБЪЕКТОВ



Классификация глубоких моделей для детектирования объектов

- **Двухстадийные модели** формируют набор гипотез, которые потом классифицируются и уточняются границы
 - R-CNN
 - Fast R-CNN
 - Faster R-CNN
 - R-FCN

- **Одностадийные модели** предполагают формирование набора прямоугольников при проходе нейронной сети
 - SSD
 - YOLOv1, *v2, *v3



Рассматриваемые модели (1)

□ ***R-CNN (2014)***

- Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. – 2014. –

[<https://arxiv.org/pdf/1311.2524.pdf>],

[<https://ieeexplore.ieee.org/abstract/document/6909475>] (опубликованная версия).

□ ***Fast R-CNN (2015)***

- Girshick R. Fast R-CNN. – 2015. – [<https://arxiv.org/pdf/1504.08083.pdf>], [<https://ieeexplore.ieee.org/document/7410526>] (опубликованная версия).

□ ***Faster R-CNN, R-FCN (2016)***

- Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. – 2016. –

[<https://arxiv.org/pdf/1506.01497.pdf>], [<https://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf>]

(опубликованная версия).

Рассматриваемые модели (2)

- Dai J., Li Y., He K., Sun J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. – 2016. – [<https://arxiv.org/pdf/1605.06409.pdf>], [<https://papers.nips.cc/paper/6465-r-fcn-object-detection-via-region-based-fully-convolutional-networks.pdf>] (опубликованная версия).

❑ **SSD (2016)**

- Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A.C. SSD: Single Shot MultiBox Detector. – 2016. – [<https://arxiv.org/pdf/1512.02325.pdf>], [https://link.springer.com/chapter/10.1007/978-3-319-46448-0_2] (опубликованная версия).

❑ **YOLOv1 (2015), *v2 (2016)**

- Redmon J., Divvala S., Girshick R., Farhadi A. You only look once: Unified, real-time object detection. – 2015. – [<https://arxiv.org/pdf/1506.02640.pdf>], [<https://ieeexplore.ieee.org/document/7780460>] (опубликованная версия).
- Redmon J., Farhadi A. YOLO9000: Better, Faster, Stronger. – 2016. – [<https://arxiv.org/pdf/1612.08242.pdf>], [<https://pjreddie.com/darknet/yolo>].

Одностадийные модели

Рассматриваемые модели (3)

❑ **YOLOv3 (2018)**

- Redmon J., Farhadi A. YOLOv3: An Incremental Improvement. – 2018. – [<https://pjreddie.com/media/files/papers/YOLOv3.pdf>].



- ❑ **Примечание:** на данный момент значительное количество нейронных сетей, которые демонстрируют хорошие результаты детектирования на открытых наборах данных, являются модификациями перечисленных моделей



R-CNN (1)

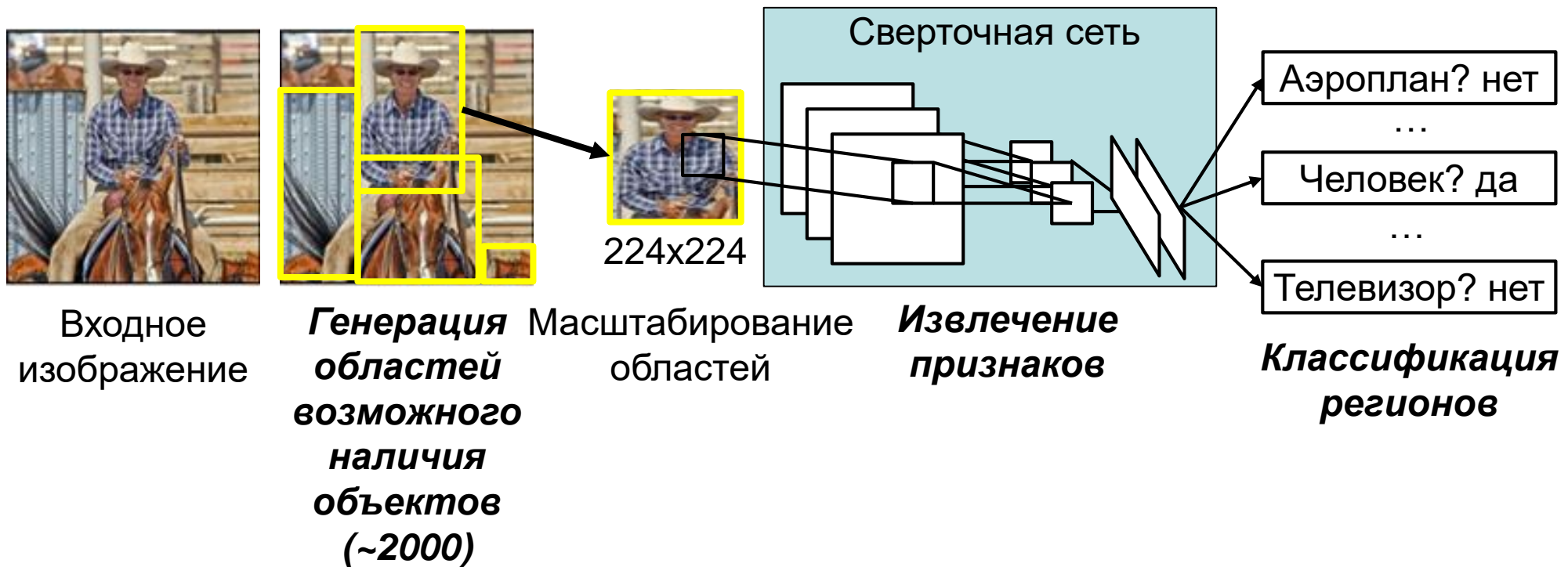
- ❑ R-CNN (Region-based Convolutional Neural Network) – одна из первых моделей, которая позволила получить высокие показатели качества детектирования на PASCAL VOC 2012
- ❑ Схема работы модели:
 - Генерация областей возможного наличия объектов – гипотез (~2000 областей)
 - Извлечение признаков для каждой сгенерированной области
 - Классификация построенных областей
 - Построение окаймляющих прямоугольников

* Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. – 2014. – [<https://arxiv.org/pdf/1311.2524.pdf>], [<https://ieeexplore.ieee.org/abstract/document/6909475>].



R-CNN (2)

□ Схема R-CNN:



* Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. – 2014. – [<https://arxiv.org/pdf/1311.2524.pdf>], [<https://ieeexplore.ieee.org/abstract/document/6909475>].

R-CNN (3)

- ❑ Генерация областей возможного наличия объектов – гипотез (~2000 областей)
 - Сканирование изображения
 - Выделение областей интереса с использованием метода выборочного поиска (selective search algorithm)
- ❑ Выделение признаков для каждой сгенерированной области
 - Обработка каждой построенной области с использованием сверточной нейронной сети посредством выполнения прямого прохода
 - В реализации R-CNN используется модель AlexNet (5 сверточных слоев и 2 полносвязных, на выходе – вектор признаков размера 4096 элементов)

* Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. – 2014. – [<https://arxiv.org/pdf/1311.2524.pdf>], [<https://ieeexplore.ieee.org/abstract/document/6909475>].



R-CNN (4)

- ❑ Классификация областей
 - Получение выхода сети и его перенаправление на вход метода опорных векторов (Support Vector Machine, SVM)
 - Использование набора бинарных SVM-классификаторов, каждый из которых определяет принадлежность определенному классу объектов
- ❑ Построение окаймляющих прямоугольников
 - Получение выхода сети
 - Перенаправление выхода сети на вход линейной регрессии для определения границ окаймляющего прямоугольника
 - Применение алгоритма подавления немаксимумов (greedy non-maximum suppression) для каждого класса объектов

* Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. – 2014. – [<https://arxiv.org/pdf/1311.2524.pdf>], [<https://ieeexplore.ieee.org/abstract/document/6909475>].



R-CNN (5)

- ❑ Основной недостаток R-CNN – вывод (inference) модели работает очень медленно, поэтому модель не может быть использована в системах реального времени
 - Для каждой сгенерированной гипотезы на входном изображении требуется прямой проход по сверточной сети, что составляет ~2000 прямых проходов на изображение
- ❑ Недостаток построения модели – необходимость обучения или тонкой настройки (fine-tuning) трех групп моделей:
 - Сверточная сеть для извлечения признаков (настройка)
 - Набор бинарных SVM-классификаторов (обучение)
 - Линейная регрессия для уточнения границ окаймляющих прямоугольников (обучение)

* Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. – 2014. – [<https://arxiv.org/pdf/1311.2524.pdf>], [<https://ieeexplore.ieee.org/abstract/document/6909475>].



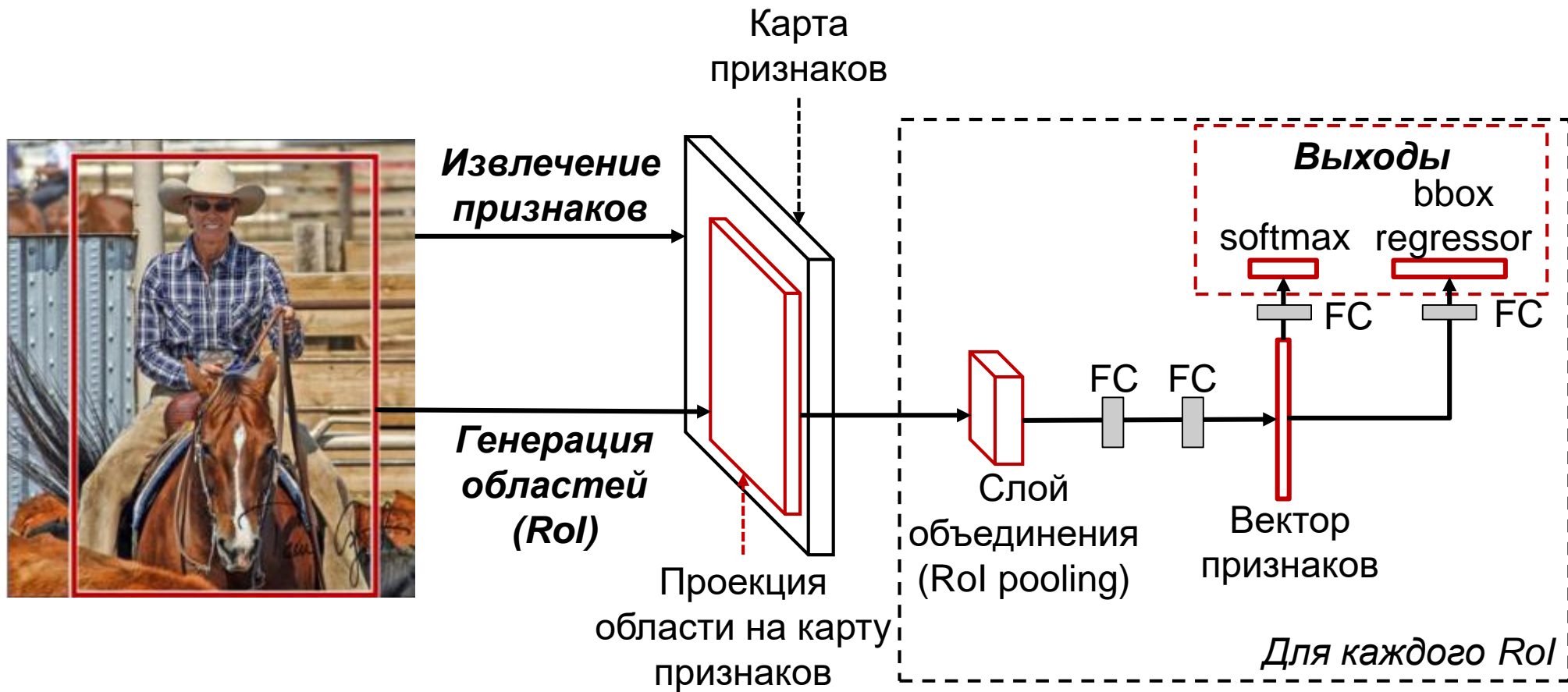
Fast R-CNN (1)

- ❑ Fast R-CNN – развитие R-CNN, направленное на ускорение вычислений в R-CNN
- ❑ Проблема – при генерации областей возможного наличия объектов области могут в значительной степени перекрываться, в результате чего сверточная сеть будет проходить по одинаковым фрагментам изображения, которые принадлежат разным гипотезам
- ❑ Решение – изменить порядок выполнения этапов извлечения признаков и генерации гипотез о возможном наличии объектов (алгоритмы используются те же, что и для R-CNN)
- ❑ Отличие – набор SVM-классификаторов и линейных регрессий заменяются на нейросетевые реализации

* Girshick R. Fast R-CNN. – 2015. – [<https://arxiv.org/pdf/1504.08083.pdf>], [<https://ieeexplore.ieee.org/document/7410526>].



Fast R-CNN (2)



* Girshick R. Fast R-CNN. – 2015. – [<https://arxiv.org/pdf/1504.08083.pdf>],
[<https://ieeexplore.ieee.org/document/7410526>].

Fast R-CNN (3)

- ❑ Слой объединения (RoI pooling layer) интегрирует информацию о построенной карте признаков входного изображения и сгенерированном регионе и формирует признаковое описание области
 - Вход:
 - Выходная карта признаков с последнего сверточного слоя нейронной сети, которая обеспечивает извлечение признаков
 - Координаты сгенерированной области в системе, связанной с входным изображением
 - Выход:
 - Признаковое описание области (размеры одинаковы для всех областей)



Fast R-CNN (4)

- ❑ Слой объединения (RoI pooling layer) интегрирует информацию о построенной карте признаков входного изображения и сгенерированном регионе и формирует признаковое описание области

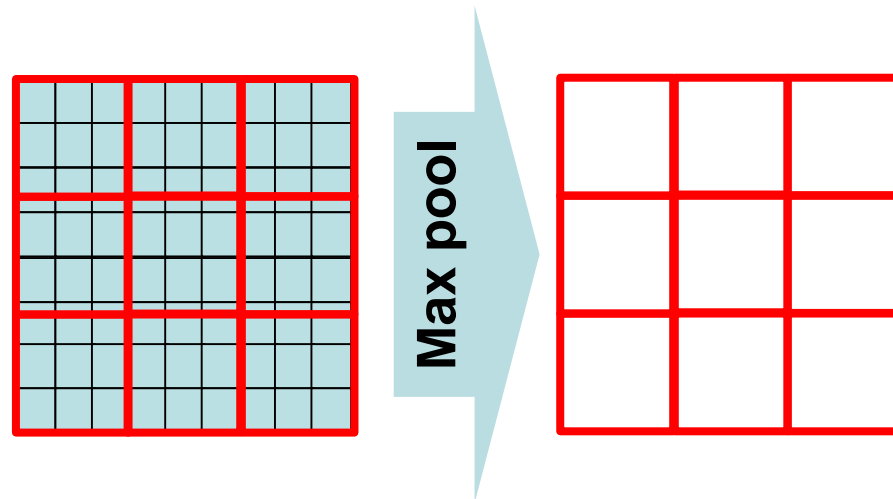
– Алгоритм:

- Координаты сгенерированной области преобразуются в систему, связанную со сверточной картой признаков, построенной для изображения
- Далее рассматривается фрагмент карты признаков размера $w \times h$, соответствующий сгенерированной области
- На полученный фрагмент накладывается сетка фиксированного размера $W \times H$ ($W = H = 7$)
- В каждой ячейке выполняется операция объединения по максимуму (max pooling), в результате чего формируется признаковое описание области пространственных размеров $W \times H$



Fast R-CNN (5)

- ❑ Слой объединения (RoI pooling layer) интегрирует информацию о построенной карте признаков входного изображения и сгенерированном регионе и формирует признаковое описание области
 - Пример для $w \times h = 9 \times 8$ и $W \times H = 3 \times 3$



Fast R-CNN (6)

- ❑ Общая часть классификатора и регрессора:
 - 2 полносвязных слоя
 - Выходной вектор используется в качестве входов для двух параллельных веток – классификатора и регрессора
- ❑ Классификатор – полносвязный слой + функция активации softmax
 - Количество элементов полносвязного слоя соответствует количеству классов с учетом фона
 - Выходной вектор отражает достоверность принадлежности каждому классу
- ❑ Регрессор – полносвязный слой
 - Выходной вектор – смещения прямоугольника для каждого возможного класса (per-class bounding-box regression offsets)



Faster R-CNN (1)

- ❑ Faster R-CNN – модификация Fast R-CNN, в которой для генерации областей возможного наличия объектов используется специальная нейронная сеть RPN (Region Proposal Network)
- ❑ По аналогии с Fast R-CNN изображение произвольного размера подается на вход сверточной нейронной сети (лучшие результаты на ResNet-101) для извлечения признаков
- ❑ Полученная карта признаков перенаправляется на вход RPN, цель которой состоит в том, чтобы обойти карту признаков скользящим окном (sliding window) и сформировать набор областей, а также карты достоверностей их принадлежности каждому из допустимых классов

* Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. – 2016. – [<https://arxiv.org/pdf/1506.01497.pdf>], [<https://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf>].

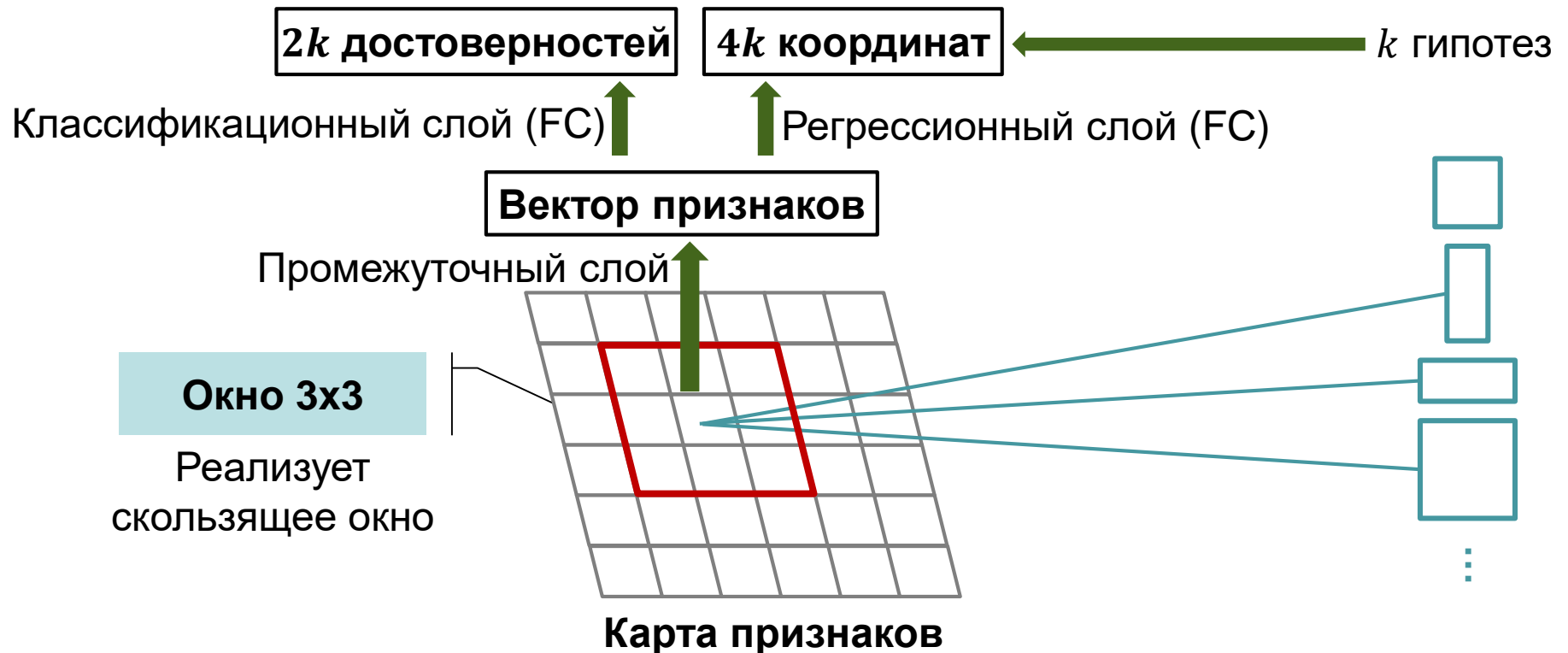
Faster R-CNN (2)

- ❑ RPN – сверточная нейронная сеть
- ❑ Вход:
 - Изображение произвольного разрешения
- ❑ Выход:
 - Набор прямоугольных регионов (гипотез) и соответствующих векторов достоверностей, отражающих степень принадлежности классу или фону
- ❑ RPN состоит из двух частей:
 - Последовательность сверточных слоев, наследуемая из широко известных моделей (например, ZF или VGG)
 - Небольшая сверточная сеть для генерации гипотез, которая реализует обход карты признаков, построенной с использованием сверточной сети (первая часть RPN)



Faster R-CNN (3)

- Сеть для генерации гипотез (Region Proposal Network):



* Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. – 2016. – [<https://arxiv.org/pdf/1506.01497.pdf>], [<https://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf>].

Faster R-CNN (4)

- Сеть для генерации гипотез (Region Proposal Network):
 - Карта признаков обходится окном размера 3×3 , что соответствует рецептивному полю размера 171×171 для модели ZF и 228×228 для VGG
 - Для каждого положения окна формируется k гипотез о прямоугольных областях, содержащих объект. Центр области совпадает с центром окна, области отличаются соотношением сторон
 - Промежуточный слой – сверточный слой с ядром 3×3 + функция активация ReLU. В результате обхода скользящим окном формируется вектор размерности 256 для модели ZF и 512 для VGG
 - Классификационный и регрессионный слои реализуются посредством одномерных сверточных слоев



Faster R-CNN (5)

- Сеть для генерации гипотез (Region Proposal Network):
 - Выход регрессионного слоя – вектор размерности $4k$, по 4 координаты для каждой гипотезы, которые соответствуют сдвигам сторон прямоугольной области (shape offsets), являющейся гипотезой
 - Выход классификационного слоя – вектор достоверностей размерности $2k$, по 2 значения для каждой гипотезы, соответствующие достоверностям того, что область содержит объект некоторого класса или нет (реализуется бинарный классификатор)



Faster R-CNN (6)

□ Генератор гипотез:

- Каждый элемент входной карты признаков соответствует ведущему положению (anchor) набора гипотез
- Для генерации гипотез используется 2 параметра – масштаб и соотношение сторон прямоугольной области
 - VGG-16 уменьшает масштаб исходного изображения в 16 раз, 16 – шаг генерации гипотез в системе координат исходного изображения
 - Если масштабы $\{8, 16, 32\}$ и соотношения сторон $\{1/2, 1/1, 2/1\}$, то генерируется 9 гипотез для каждого ведущего положения
 - Чтобы получить следующее ведущее положение на исходном изображении, достаточно сдвинуть предыдущее положение на 16

- ## □ **Примечание:** модель обучается как единая нейронная сеть, функция ошибки – взвешенная функция потерь для двух веток, соответствующих классификации и регрессии

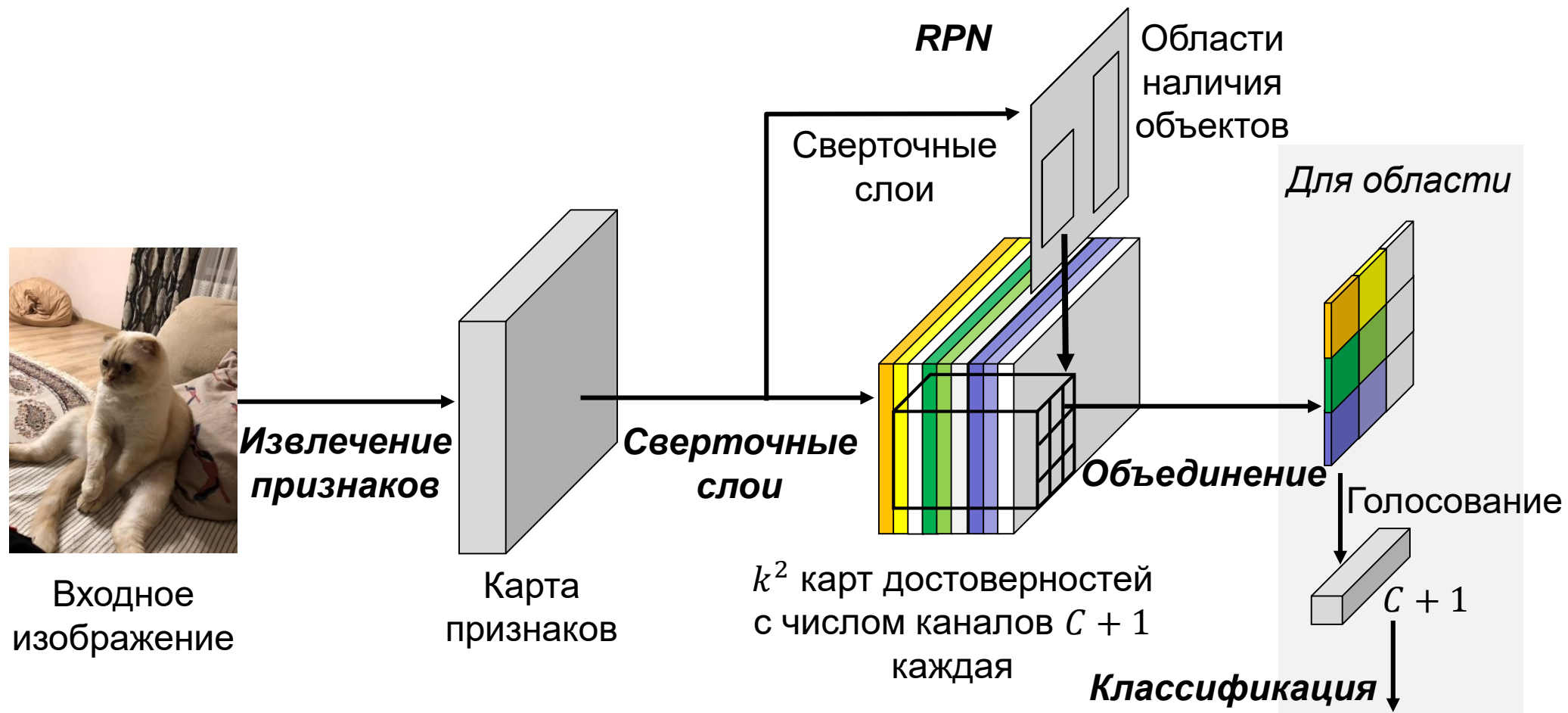


R-FCN (1)

- ❑ R-FCN (Region-based Fully Convolutional Network) является логическим продолжением развития метода Faster R-CNN
- ❑ Основная идея R-FCN состоит в том, чтобы на выходе сети сформировать **карты достоверностей принадлежности допустимым классам, которые чувствительны к расположению областей возможного наличия объектов** (position-sensitive score maps)

* Dai J., Li Y., He K., Sun J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. – 2016.
– [<https://arxiv.org/pdf/1605.06409.pdf>], [<https://papers.nips.cc/paper/6465-r-fcn-object-detection-via-region-based-fully-convolutional-networks.pdf>].

R-FCN (2)



* Dai J., Li Y., He K., Sun J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. – 2016. – [<https://arxiv.org/pdf/1605.06409.pdf>], [<https://papers.nips.cc/paper/6465-r-fcn-object-detection-via-region-based-fully-convolutional-networks.pdf>].

R-FCN (3)

□ Схема работы R-FCN:

- Извлечение признаков из исходного изображения посредством прямого прохода по некоторой сверточной нейронной сети
- Добавление сверточных слоев и формирование набора карт достоверностей принадлежности допустимым классам, которые чувствительны к расположению областей возможного наличия объектов
 - Количество таких карт – k^2 , что отвечает числу относительных положений объекта на пространственной сетке $k \times k$, которой разбивается каждая область возможного наличия объекта (если $k = 3$, то относительные положения «сверху слева», «сверху по центру», ..., «снизу справа»)
 - Глубина каждой карты $C + 1$, где C – количество категорий объектов
 - Глубина объединенной карты признаков – $k^2(C + 1)$



R-FCN (4)

□ Схема работы R-FCN:

- Генерация областей возможного наличия объектов с использованием полностью сверточной RPN
- Объединение карт достоверностей в соответствии с относительным положением в области (position-sensitive RoI pooling layer)
 - В соответствии с расположением области вырезается соответствующая часть набора карт признаков, отвечающих относительным позициям объекта
 - Полученные карты реорганизуются в соответствии с относительными позициями
- Классификация областей с помощью softmax-классификатора. Вход классификатора – вектор достоверностей принадлежности области каждому из допустимых классов, полученный посредством голосования



SSD (1)

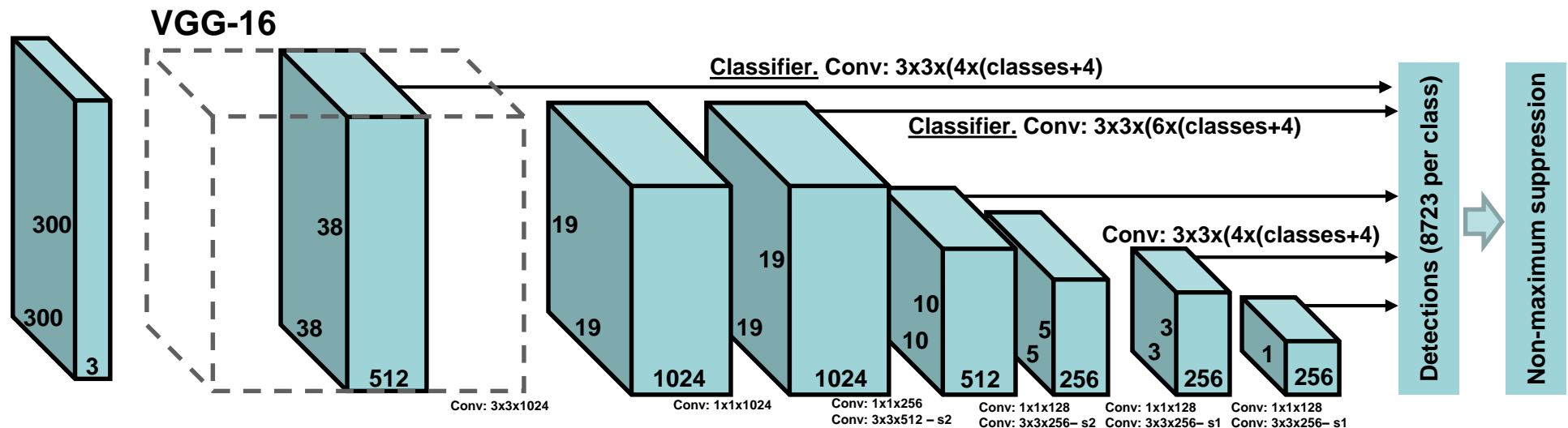
- ❑ SSD (Single Shot Multibox Detector) позволяет одновременно предсказывать размещение окаймляющих прямоугольников и классифицировать объекты, ограниченные этими прямоугольниками
- ❑ SSD представляет собой единую сверточную нейронную сеть, к промежуточным картам признаков которой применяются нейросетевые детекторы
- ❑ Разработаны архитектуры для разных размеров входа (SSD300 – 300x300, SSD512 – 512x512 и другие)

* Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A.C. SSD: Single Shot MultiBox Detector. – 2016. – [<https://arxiv.org/pdf/1512.02325.pdf>], [https://link.springer.com/chapter/10.1007/978-3-319-46448-0_2].



SSD (2.1)

□ Архитектура SSD300:



* Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A.C. SSD: Single Shot MultiBox Detector. – 2016. – [<https://arxiv.org/pdf/1512.02325.pdf>], [https://link.springer.com/chapter/10.1007/978-3-319-46448-0_2].

SSD (2.2)

Слой (num_filters w×h, stride[, pad])	Разрешение карты признаков на выходе слоя	Классификатор	Разрешение карты признаков на выходе классификатора	Количество гипотез
image:	300×300×3			
conv1_1: 64 3×3, 1, 1 + ReLU	300×300×64			
conv1_2: 64 3×3, 1, 1 + ReLU	300×300×64			
pool1: max 2×2, 2	150×150×64			
conv2_1: 128 3×3, 1, 1 + ReLU	150×150×128			
conv2_2: 128 3×3, 1, 1 + ReLU	150×150×128			
pool2: max 2×2, 2	75×75×128			
conv3_1: 256 3×3, 1, 1 + ReLU	75×75×256			
conv3_2: 256 3×3, 1, 1 + ReLU	75×75×256			
conv3_3: 256 3×3, 1, 1 + ReLU	75×75×256			
pool3: max 2×2, 2	38×38×256			
conv4_1: 512 3×3, 1, 1 + ReLU	38×38×512			
conv4_2: 512 3×3, 1, 1 + ReLU	38×38×512			
conv4_3: 512 3×3, 1, 1 + ReLU	38×38×512	conv_c1: $4(c+4) \times 3 \times 3, 1$	$38 \times 38 \times [4(c+4)]$	$38 \times 38 \times 4 = 5776$
pool4: max 2×2, 2	19×19×512			
conv5_1: 512 3×3, 1, 1 + ReLU	19×19×512			
conv5_2: 512 3×3, 1, 1 + ReLU	19×19×512			
conv5_3: 512 3×3, 1, 1 + ReLU	19×19×512			
pool5: max 3×3, 1, 1	19×19×512			
fc6: 1024 3×3, 1, 6 (dilation: 6) + ReLU	19×19×1024			
fc7: 1024 1×1, 1, 0 + ReLU	19×19×1024	conv_c2: $6(c+4) \times 3 \times 3, 1$	$19 \times 19 \times [6(c+4)]$	$19 \times 19 \times 6 = 2166$
conv6_1: 256 1×1, 1, 0 + ReLU	19×19×256			
conv6_2: 512 3×3, 2, 1 + ReLU	10×10×512	conv_c3: $6(c+4) \times 3 \times 3, 1$	$10 \times 10 \times [6(c+4)]$	$10 \times 10 \times 6 = 600$
conv7_1: 128 1×1, 1, 0 + ReLU	10×10×128			
conv7_2: 256 3×3, 2, 1 + ReLU	5×5×256	conv_c4: $6(c+4) \times 3 \times 3, 1$	$5 \times 5 \times [6(c+4)]$	$5 \times 5 \times 6 = 150$
conv8_1: 128 1×1, 1, 0 + ReLU	5×5×128			
conv8_2: 256 3×3, 1, 0 + ReLU	3×3×256	conv_c5: $4(c+4) \times 3 \times 3, 1$	$3 \times 3 \times [4(c+4)]$	$3 \times 3 \times 4 = 36$
conv9_1: 128 1×1, 1, 0 + ReLU	3×3×128			
conv9_2: 256 3×3, 1, 0 + ReLU	1×1×256	conv_c6: $4(c+4) \times 3 \times 3, 1$	$1 \times 1 \times [4(c+4)]$	$1 \times 1 \times 4 = 4$
Общее число гипотез о расположении объектов				8732

- Структура SSD300 (conv – сверточный слой, pool – пространственное объединение; количество фильтров классификатора $k(c+4)$, где k – количество прямоугольников по умолчанию, c – количество классов, 4 соответствует числу сторон прямоугольников (каждое значение – сдвиг стороны окаймляющего прямоугольника относительно стороны прямоугольника по умолчанию))

SSD (3)

□ Архитектура SSD300:

- Модель построена на базе модели VGG-16, в которой сверточные слои фигурируют без изменений, а полностью связанные слои заменены на полностью сверточные
- К картам признаков разного масштаба присоединяются классификационные сверточные слои, которые одновременно обеспечивают генерацию возможных положений объектов и их классификацию
- Для исключения дублирования окаймляющих прямоугольников, выполняется процедура подавления не-максимумов (non-maximum suppression)



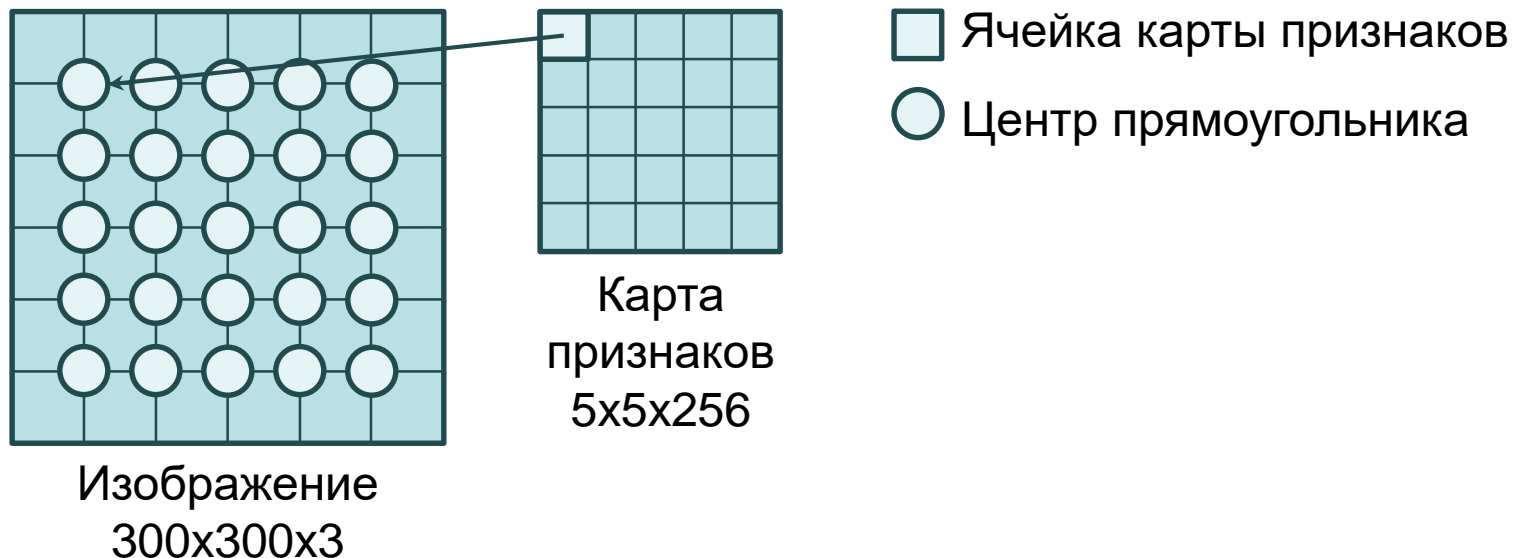
SSD (4)

- Классификационные сверточные слои:
 - Карта признаков на некотором уровне отвечает описанию изображения некоторого масштаба в целом, ячейка карты – описанию некоторой прямоугольной области изображения
 - Каждый классификационный слой определенной ячейке карты признаков ставит в соответствие набор окаймляющих прямоугольников по умолчанию (k штук)
 - Для каждого прямоугольника определяется вектор достоверностей принадлежности объекта допустимым классам (длины C) и вектор сдвигов сторон прямоугольника по умолчанию для уточнения его границ (вектор длины 4)
 - Если карта признаков имеет размеры $m \times n$ и каждой ячейке соответствует k -прямоугольников, то количество выходов на классификационном слое составляет $ktn(c + 4)$



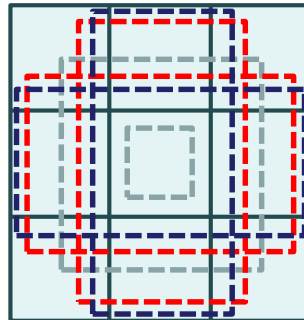
SSD (5)

- ❑ Формирование гипотез – прямоугольников, потенциально содержащих объект
 - Рассмотрим на примере SSD300 и карты признаков, построенной на четвертом сверточном слое



SSD (6)

- Формирование гипотез – прямоугольников, потенциально содержащих объект
 - Для каждого центра делается предположение о расположении объекта
 - Гипотез 4 или 6, что соответствует количеству прямоугольников, у которых центр расположен в выбранной точке: два квадрата разного масштаба, две пары прямоугольников с соотношением сторон $1/2$, $2/1$ и $1/3$, $3/1$



- На самой крупной и двух самых мелких картах признаков используется по 4 гипотезы

YOLOv1 (1)

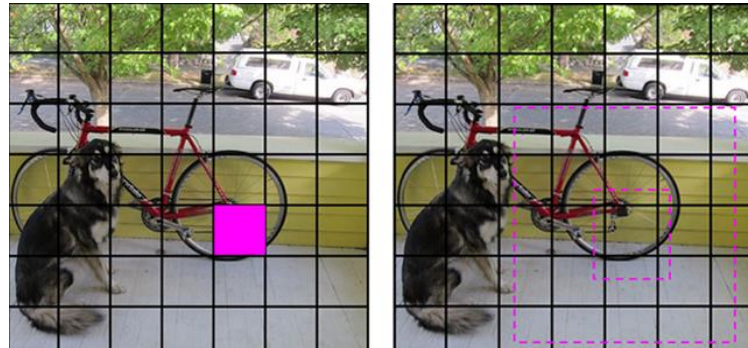
- ❑ YOLO (You Only Look Once) – еще одна модель детектирования объектов, которая представляется единой сверточной сетью, обеспечивающей построение окаймляющих прямоугольников и классификацию объектов в этих прямоугольниках
- ❑ Модель плохо обнаруживает объекты небольшого размера

* Redmon J., Divvala S., Girshick R., Farhadi A. You only look once: Unified, real-time object detection. – 2015.
– [<https://arxiv.org/pdf/1506.02640.pdf>], [<https://ieeexplore.ieee.org/document/7780460>].



YOLOv1 (2)

- ❑ Входное изображение делится на ячейки сеткой $S \times S$
- ❑ Каждая ячейка отвечает за предсказание B окаймляющих прямоугольников



- ❑ Для каждого окаймляющего прямоугольника предсказываются параметры x, y, w, h, c , где (x, y) – центр прямоугольника относительно границ ячейки, w и h – ширина и высота прямоугольника в системе координат изображения, c – достоверность присутствия объекта

* Redmon J., Divvala S., Girshick R., Farhadi A. You only look once: Unified, real-time object detection. – 2015. – [<https://arxiv.org/pdf/1506.02640.pdf>], [<https://ieeexplore.ieee.org/document/7780460>].

YOLOv1 (3)

- ❑ Достоверность присутствия объекта в ячейке определяется следующим образом:

$$c = P(Object) \cdot IoU_{pred}^{truth},$$

где $P(Object)$ – вероятность наличия объекта в окаймляющем прямоугольнике, IoU_{pred}^{truth} – отношение площади пересечения обнаруженного и размеченного прямоугольников

- ❑ Достоверность строится для ячейки независимо от количества соответствующих окаймляющих прямоугольников
- ❑ Если ячейка не содержит объект, то достоверность равна нулю
- ❑ В противном случае, значение достоверности принимается равным IoU_{pred}^{truth}



YOLOv1 (4)

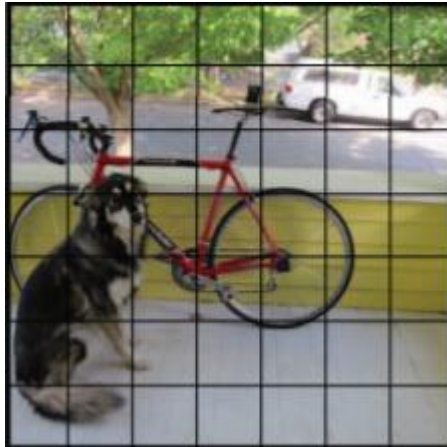
- Для каждого окаймляющего прямоугольника прогнозируется C условных вероятностей $P(Class_i | Object)$, где C – количество детектируемых классов объектов
- Условные вероятности классов умножаются на предсказания достоверности прямоугольника, что позволяет получить оценки достоверности для каждого прямоугольника, зависящие от класса:

$$P(Class_i | Object) \cdot P(Object) \cdot IoU_{pred}^{truth} = P(Class_i) \cdot IoU_{pred}^{truth}$$

- Построенные оценки отражают 2 аспекта:
 - Вероятность наличия объекта определенного класса в прямоугольнике
 - Степень соответствия предсказанного прямоугольника объекту



YOLOv1 (5)



Входное изображение,
разбитое на ячейки



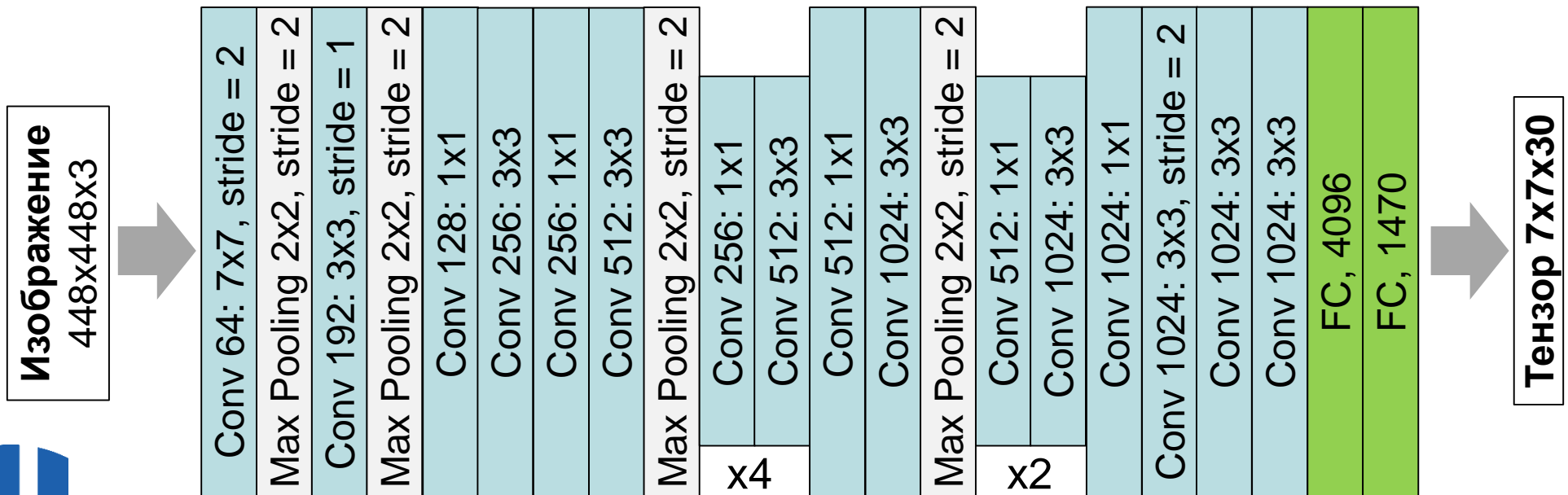
Карта достоверностей
(разный цвет – разные
классы объектов)

- Предсказание – тензор размерности $S \times S \times (B * 5 + C)$, для каждой ячейки сетки размерности $S \times S$ предсказывается B окаймляющих прямоугольников и C вероятностей принадлежности классам

* Redmon J., Divvala S., Girshick R., Farhadi A. You only look once: Unified, real-time object detection. – 2015. – [<https://arxiv.org/pdf/1506.02640.pdf>], [<https://ieeexplore.ieee.org/document/7780460>].

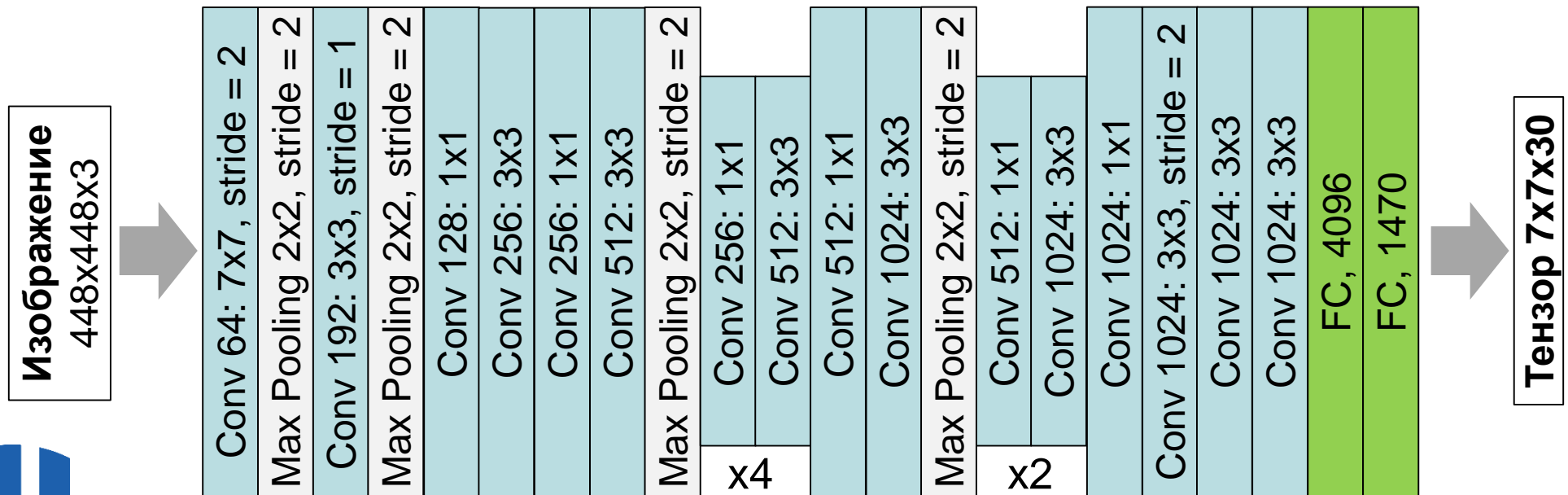
YOLOv1 (6)

- YOLOv1 базируется на модели GoogLeNet:
 - Сеть содержит 24 сверточных слоя и следующих за ними 2 полносвязных слоя, после каждого слоя функция активации ReLU
 - Вместо начальных inception-модулей используется сверточный слой, понижающий размерность изображения



YOLOv1 (7)

- YOLOv1 базируется на модели GoogLeNet:
 - При запуске на PASCAL VOC количество ячеек $S = 7$ при построении сети, количество прямоугольников – $B = 2$, количество классов объектов – $C = 20$
 - Сеть сначала обучается на ImageNet для настройки 20 сверточных и 1 полносвязного слоев на изображениях 224x224



YOLOv2 (1)

- ❑ YOLOv2 – модификация YOLOv1
- ❑ Основные изменения:
 - Пакетная нормализация входов каждого сверточного слоя
 - Предварительная настройка сверточных слоев на ImageNet осуществляется на изображениях разрешения 448x448, что дает возможность настроить фильтры для работы на высоком разрешении
 - Использование ведущих прямоугольников вместо прямого предсказания координат
 - Многомасштабное обучение – каждые 10 пачек случайно меняется разрешение изображения {320, 352,..., 608}
 - Увеличение количества категорий объектов (YOLO9000)

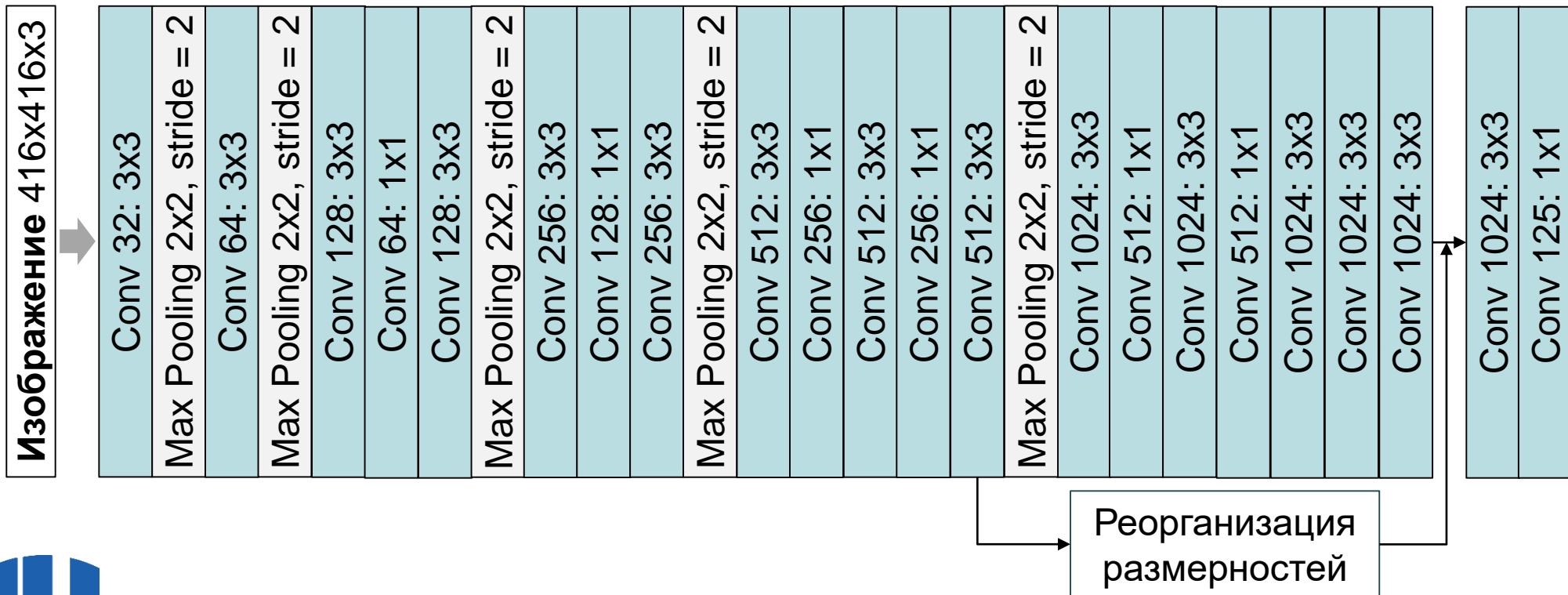
* Redmon J., Farhadi A. YOLO9000: Better, Faster, Stronger. – 2016. –
[<https://arxiv.org/pdf/1612.08242.pdf>], [<https://pjreddie.com/darknet/yolo>].



YOLOv2 (2)

□ Архитектура YOLOv2:

- Выходной тензор имеет размерность 13x13x125: 13x13 соответствует сетке разбиения исходного изображения, 5 прямоугольников в каждой ячейке (по 25 параметров)



YOLOv2 (3)

- ❑ Использование ведущих прямоугольников вместо прямого предсказания координат
- ❑ Генерация ведущих окаймляющих прямоугольников с помощью алгоритма кластеризации k-средних
 - Вместо ручного выбора B прямоугольников, запускается кластеризация k-средних на обучающей выборке окаймляющих прямоугольников для автоматического поиска хороших начальных приближений
 - Метрика расстояния между прямоугольником и кластером:
$$d(box, centroid) = 1 - IoU(box, centroid)$$
 - Экспериментально показано, что $k = 5$ – хороший компромисс между сложностью модели и высоким откликом
 - Для каждой ячейки входного изображения формируется 5 ведущих прямоугольников



YOLOv2 (4)

- Сеть для каждого прямоугольника предсказывает 5 компонент t_x , t_y , t_w , t_h и t_o и вектор принадлежности классам (20 классов)

$$b_x = \sigma(t_x) + c_x, b_y = \sigma(t_y) + c_y,$$

$$b_w = p_w e^{t_w}, b_h = p_h e^{t_h},$$

$$P(object) * IoU(b, object) = \sigma(t_o)$$

где (c_x, c_y) – смещение текущей ячейки относительно левого верхнего угла изображения,

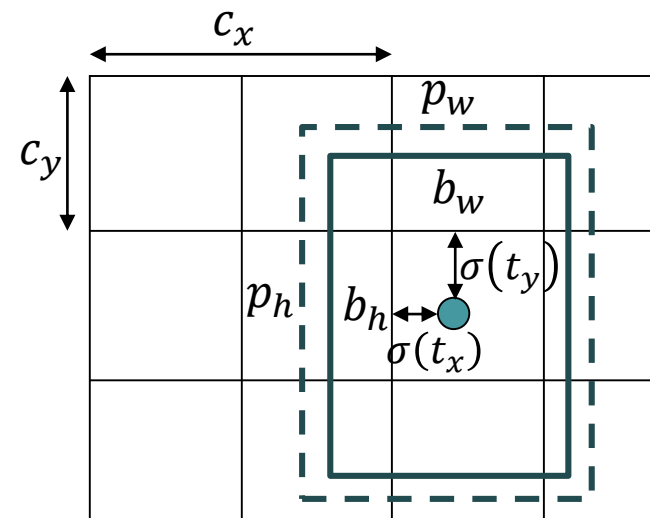
p_w, p_h – ширина и высота ведущего прямоугольника,

t_x – смещение по x ,

t_y – смещение по y ,

t_o – значение достоверности,

$\sigma(\cdot)$ – сигмоидальная функция активации



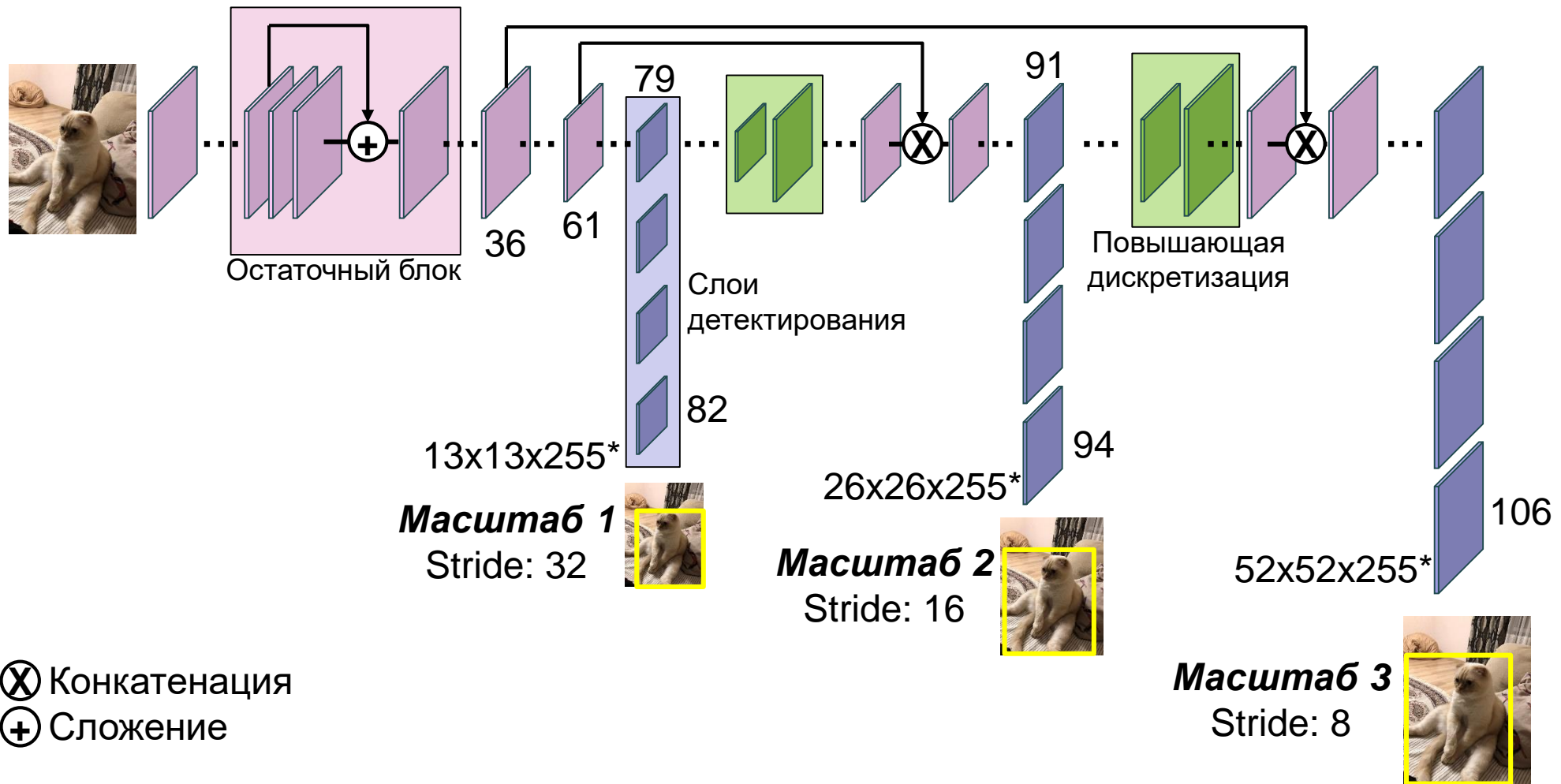
YOLOv3 (1)

- YOLOv3 – развитие YOLOv2
 - Наращивание глубины сети (106 полностью сверточных слоев)
 - Добавление остаточных связей
 - Детектирование объектов на трех разных масштабах признаков описаний
 - Использование трех ведущих прямоугольников на каждом масштабе вместо пяти

* Redmon J., Farhadi A. YOLOv3: An Incremental Improvement. – 2018. – [\[https://pjreddie.com/media/files/papers/YOLOv3.pdf\]](https://pjreddie.com/media/files/papers/YOLOv3.pdf).



YOLOv3 (2)



* $255 = B \times (5 + C) = 3 \times (5 + 80)$, 80 классов в наборе данных MS COCO.

** What's new in YOLO v3? [<https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b>].

СРАВНЕНИЕ МОДЕЛЕЙ ДЕТЕКТИРОВАНИЯ ОБЪЕКТОВ



Изменение качества детектирования (1)

- ❑ Тренировочный набор данных: PASCAL VOC 2012 + сторонние данные (обычно MS COCO)
- ❑ Тестовый набор данных: PASCAL VOC 2012
- ❑ Количество классов объектов: 20 классов
- ❑ Показатель качества: средняя точность предсказания (average precision)
- ❑ Модели:
 - R-CNN (R-CNN (bbox reg)*) – модель R-CNN, построенная на 16-слойной сверточной сети, которая обучена на ILSVRC 2012 и настроена на VOC 2012 trainval, SVM-детекторы обучены на VOC 2012 trainval

* Указано название модели, которое фигурирует в таблице с результатами детектирования объектов на данных PASCAL VOC 2012

[\[http://host.robots.ox.ac.uk:8080/leaderboard/displaylb_main.php?challengeid=11&compid=4\]](http://host.robots.ox.ac.uk:8080/leaderboard/displaylb_main.php?challengeid=11&compid=4).



Изменение качества детектирования (2)

- Faster R-CNN (Faster RCNN, ResNet (VOC+COCO)*) – развитие модели Faster RCNN. Построена на ResNet, обучена на ImageNet и настроена на MS COCO trainval, настроена на VOC 2007 trainval+test и VOC 2012 trainval
- R-FCN (R-FCN, ResNet (VOC+COCO)) – модель R-FCN, построенная на базе ResNet-101. Предварительно обучена на ImageNet, последовательно настроена на наборах MS COCO trainval, VOC 2007 trainval+test и VOC 2012 trainval
- SSD300ft (SSD300 VGG16 07++12+COCO) – SSD300, обученная на MS COCO trainval35k и настроенная на VOC07 trainval + test and VOC12 trainval
- YOLOv2 – базовая модель, рассмотренная в лекции
- ATLDENV2 – ансамбль из двух моделей, основанных на ResneXt152_32x8d (описание модели не опубликовано)



Изменение качества детектирования (3)

Модель	Год	mAP, %	AP, % (для некоторых классов)				
			bus	car	cat	person	train
R-CNN	2014	62.4	65.9	66.4	84.6	76.0	54.2
Faster R-CNN	2015	83.8	86.3	87.8	94.2	89.6	90.3
SSD300ft	2016	79.3	84.9	84.0	93.4	85.6	88.3
R-FCN	2016	85.0	86.7	89.0	95.8	91.1	92.0
YOLOv2	2017	75.4	81.2	78.2	92.9	88.6	88.8
ATLDETV2	2019	92.9	95.5	95.7	98.0	96.1	96.2

- ❑ С 2014 по 2019 гг. средняя точность детектирования увеличилась на 30% за счет применения рассмотренных подходов
- ❑ Легковесные модели (YOLOv2) показывают более низкие результаты качества
- ❑ Согласно результатам конкурса PASCAL VOC 2012* разрабатывается много модификаций рассмотренных моделей

* Detection Results: VOC2012

[http://host.robots.ox.ac.uk:8080/leaderboard/displaylb_main.php?challengeid=11&compid=4].



Сравнение качества и скорости работы моделей

- ❑ Тренировочные данные: PASCAL VOC 2007+2012
- ❑ Тестовые данные: PASCAL VOC 2007
- ❑ Показатель качества: средняя точность предсказания, усредненная по 20 классам (mean average precision)
- ❑ Инфраструктура: NVIDIA M40 или Titan X (сравнение качественное)

*Неплохое качество, но низкий FPS
(модель не работает в реальном времени)*

Высокий FPS, но низкое качество

Компромисс между качеством и скоростью работы

Модель	mAP, %	FPS
Fast R-CNN	70.0	0,5
Faster R-CNN VGG-16	73.2	7
Faster R-CNN ResNet	76.4	5
YOLO	63.4	45
SSD500	76.8	19
YOLOv2 544x544	78.6	40

* Redmon J., Farhadi A. YOLO9000: Better, Faster, Stronger. – 2016. –
[<https://arxiv.org/pdf/1612.08242.pdf>], [<https://pjreddie.com/darknet/yolo>].

Заключение

- ❑ Множество глубоких моделей для детектирования объектов не ограничивается рассмотренными в настоящей лекции
- ❑ Существует большое количество модификаций рассмотренных архитектур (в частности, Faster R-CNN и SSD), о чем свидетельствуют результаты широко известных конкурсов по детектированию объектов разных классов
- ❑ **Оптимальная модель – компромисс между точностью и скоростью**
 - Точность определяется требованиями к результатам решения задачи (результаты точности различаются в зависимости от тестовых данных!)
 - Скорость определяется имеющимися аппаратными возможностями (высокая скорость вывода на мощных GPU не всегда является хорошим показателем)



Основная литература (1)

- ❑ Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. – 2014. – [<https://arxiv.org/pdf/1311.2524.pdf>], [<https://ieeexplore.ieee.org/abstract/document/6909475>].
- ❑ Girshick R. Fast R-CNN. – 2015. – [<https://arxiv.org/pdf/1504.08083.pdf>], [<https://ieeexplore.ieee.org/document/7410526>].
- ❑ Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. – 2016. – [<https://arxiv.org/pdf/1506.01497.pdf>], [<https://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf>].



Основная литература (2)

- ❑ Dai J., Li Y., He K., Sun J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. – 2016. –
[\[https://arxiv.org/pdf/1605.06409.pdf\]](https://arxiv.org/pdf/1605.06409.pdf),
[\[https://papers.nips.cc/paper/6465-r-fcn-object-detection-via-region-based-fully-convolutional-networks.pdf\]](https://papers.nips.cc/paper/6465-r-fcn-object-detection-via-region-based-fully-convolutional-networks.pdf).
- ❑ Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A.C. SSD: Single Shot MultiBox Detector. – 2016. –
[\[https://arxiv.org/pdf/1512.02325.pdf\]](https://arxiv.org/pdf/1512.02325.pdf),
[\[https://link.springer.com/chapter/10.1007/978-3-319-46448-0_2\]](https://link.springer.com/chapter/10.1007/978-3-319-46448-0_2).
- ❑ Redmon J., Divvala S., Girshick R., Farhadi A. You only look once: Unified, real-time object detection. – 2015. –
[\[https://arxiv.org/pdf/1506.02640.pdf\]](https://arxiv.org/pdf/1506.02640.pdf),
[\[https://ieeexplore.ieee.org/document/7780460\]](https://ieeexplore.ieee.org/document/7780460).



Основная литература (3)

- ❑ Redmon J., Farhadi A. YOLO9000: Better, Faster, Stronger. – 2016. – [<https://arxiv.org/pdf/1612.08242.pdf>], [<https://pjreddie.com/darknet/yolo>].
- ❑ Redmon J., Farhadi A. YOLOv3: An Incremental Improvement. – 2018. – [<https://pjreddie.com/media/files/papers/YOLOv3.pdf>].



Авторский коллектив

- ❑ **Турлапов Вадим Евгеньевич**
д.т.н., профессор кафедры МОСТ ИИТММ ННГУ
vadim.turlapov@itmm.unn.ru
- ❑ **Васильев Евгений Павлович**
преподаватель кафедры МОСТ ИИТММ ННГУ
evgeny.vasiliev@itmm.unn.ru
- ❑ **Гетманская Александра Александровна**
преподаватель кафедры МОСТ ИИТММ ННГУ
getmanskaya.alexandra@gmail.com
- ❑ **Кустикова Валентина Дмитриевна**
к.т.н., доцент кафедры МОСТ ИИТММ ННГУ
valentina.kustikova@itmm.unn.ru

