



УНИВЕРСИТЕТ ИТМО

# Задача детектирования объектов

## Одностадийная детекция

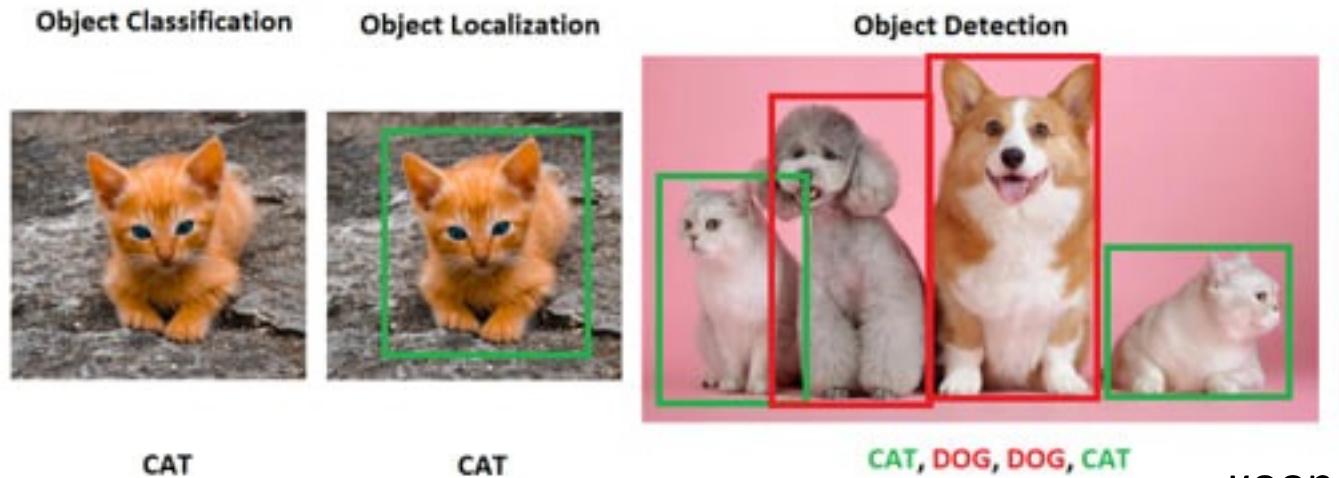
Ефимова Валерия Александровна

[vefimova@itmo.ru](mailto:vefimova@itmo.ru)

15.10.2022

- Задача детекции и оценка ее качества
- Одностадийные детекторы (YOLO, SSD, RetinaNet)
- Уточнение детекции
- Двустадийные детекторы (Mask R-CNN)
- Детекция на основе точек и EfficienDet
- Распознавание текста

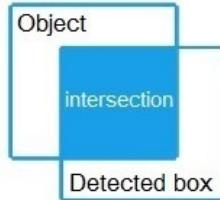
- Много объектов разных категорий.



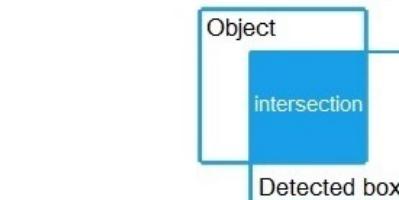
- Нужно не только определить класс, но и место на изображении – координаты центра объекта, ширина и высота окна, метка класса и степень уверенности:  $b_x, b_y, b_h, b_w, label, p$ .



# Intersection over Union (IoU, Jaccard Index)



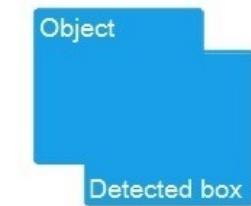
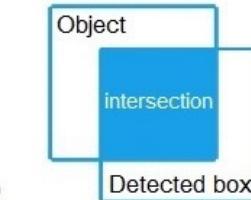
Precision =  $\frac{\text{Area of Overlap}}{\text{Area of Union}}$



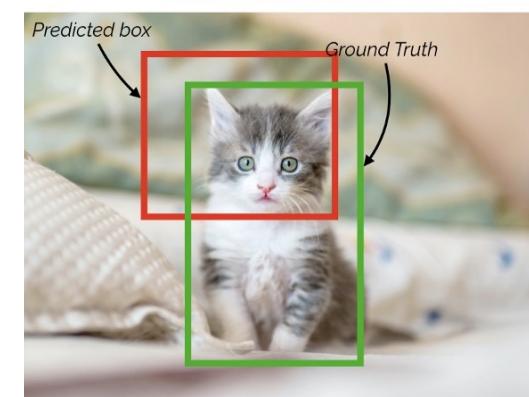
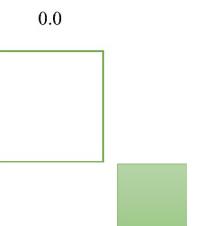
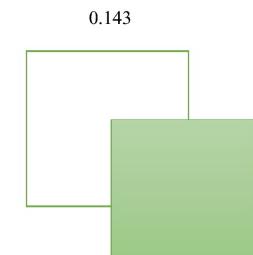
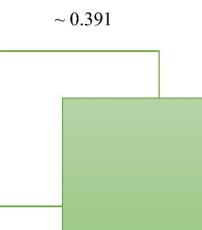
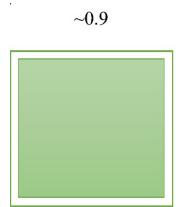
Recall =  $\frac{\text{Area of Overlap}}{\text{Area of Union}}$



IoU =  $\frac{\text{Area of Overlap}}{\text{Area of Union}}$

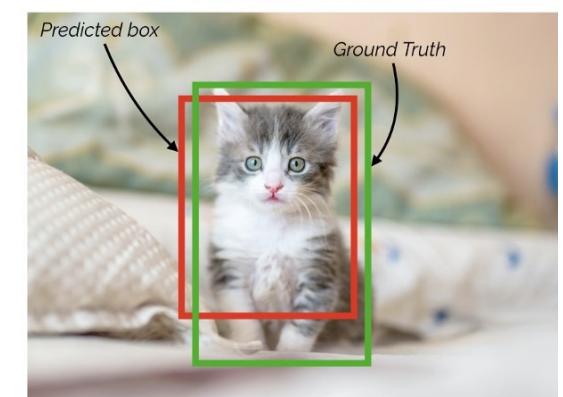


IoU – мера  
перекрытия  
баундинг боксов



*False Positive (FP)*

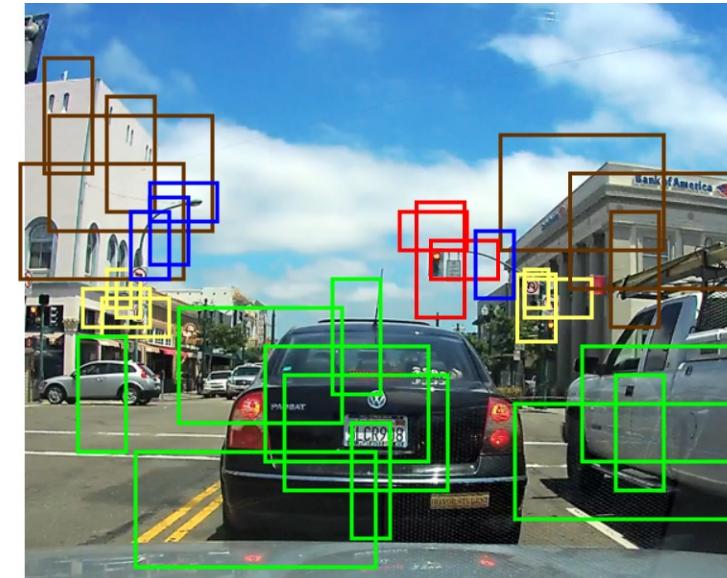
$IoU = \sim 0.3$



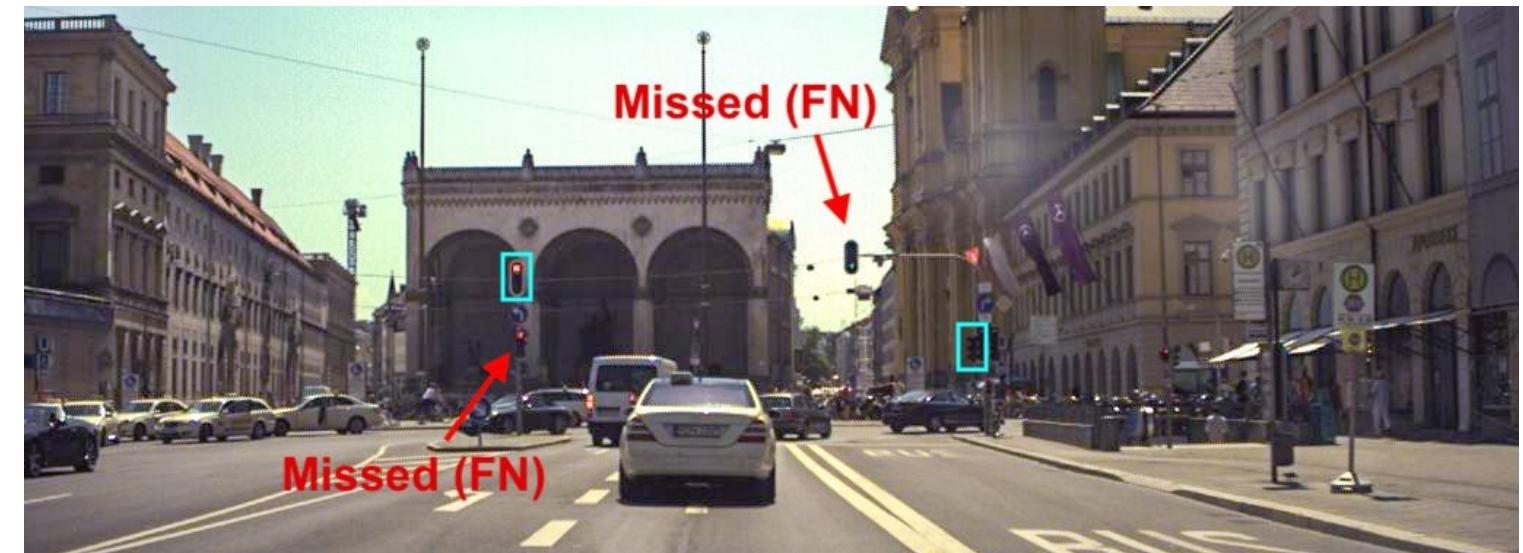
*True Positive (TP)*

$IoU = \sim 0.7$

- Высокий recall, но низкий precision:



- Высокий precision, но низкий recall:

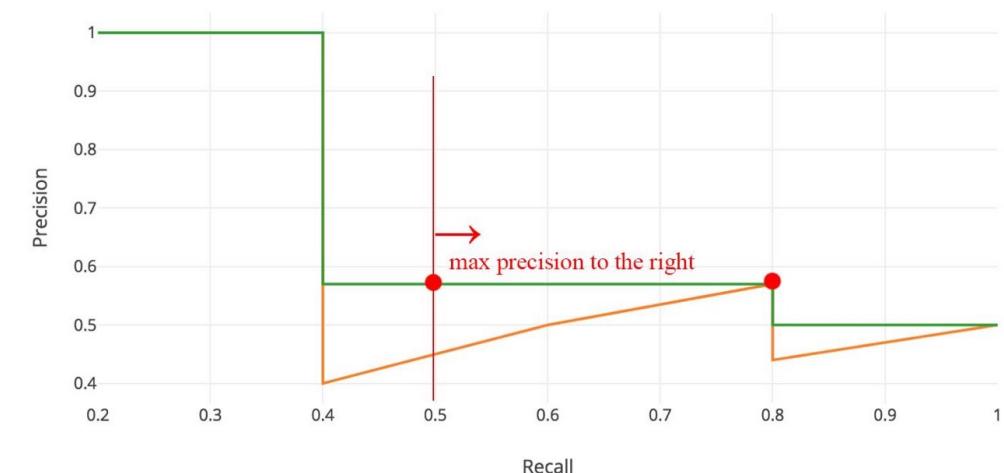
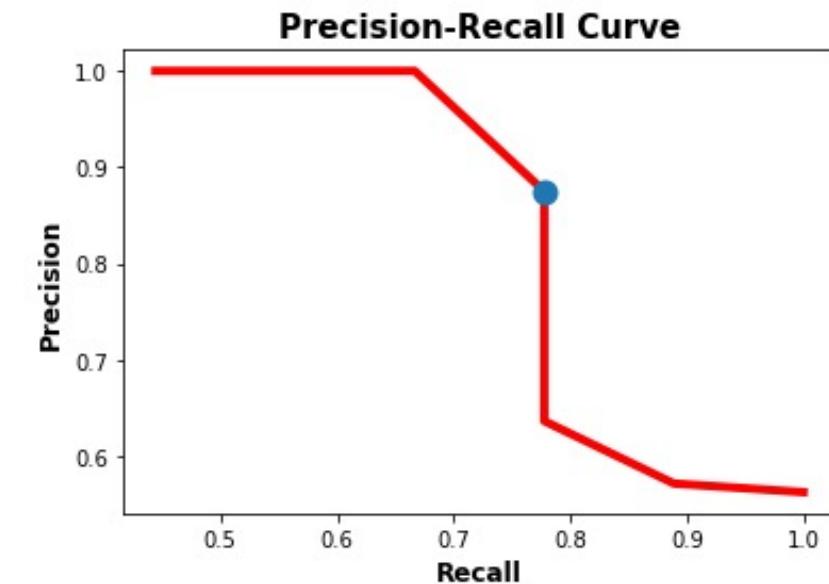


- Average Precision (AP) – площадь под PR-кривой.

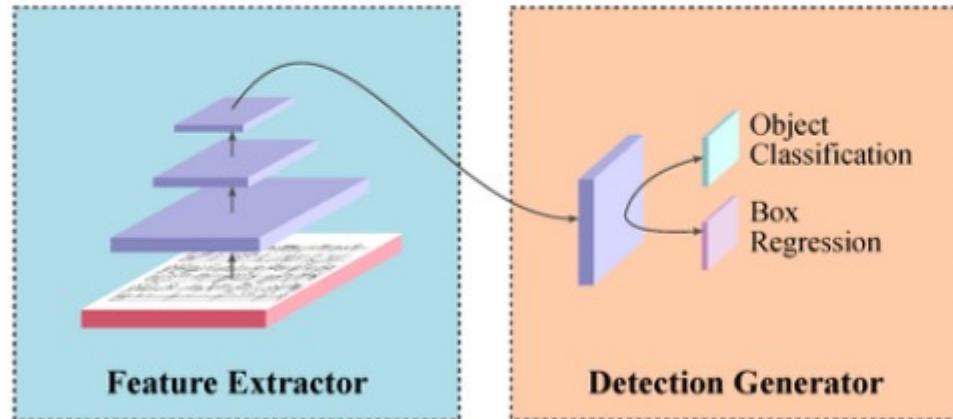
$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

- Кривая Precision-Recall получается путем построения значений точности и полноты модели в зависимости от порога достоверности модели – инкапсулирует компромисс между обеими метриками и максимизирует эффект обеих метрик.
- Mean Average Precision (mAP) – среднее Average Precision, где N – число классов.

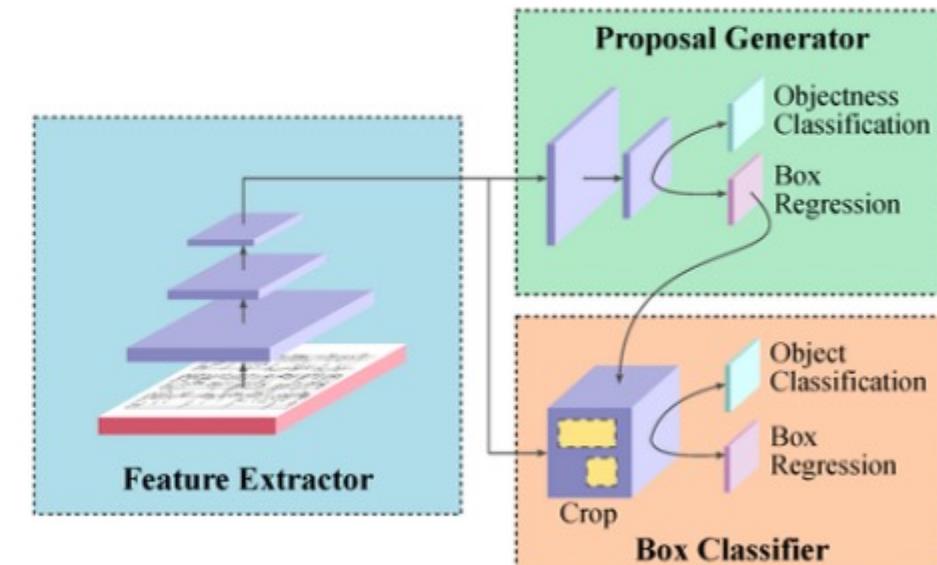
$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$



- Одностадийные (One-Stage) – YOLO, SSD, RetinaNet.
- Двустадийные (Two-Stage) – R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN.
- На основе точек (Point-based) – CenterNet, CornerNet.

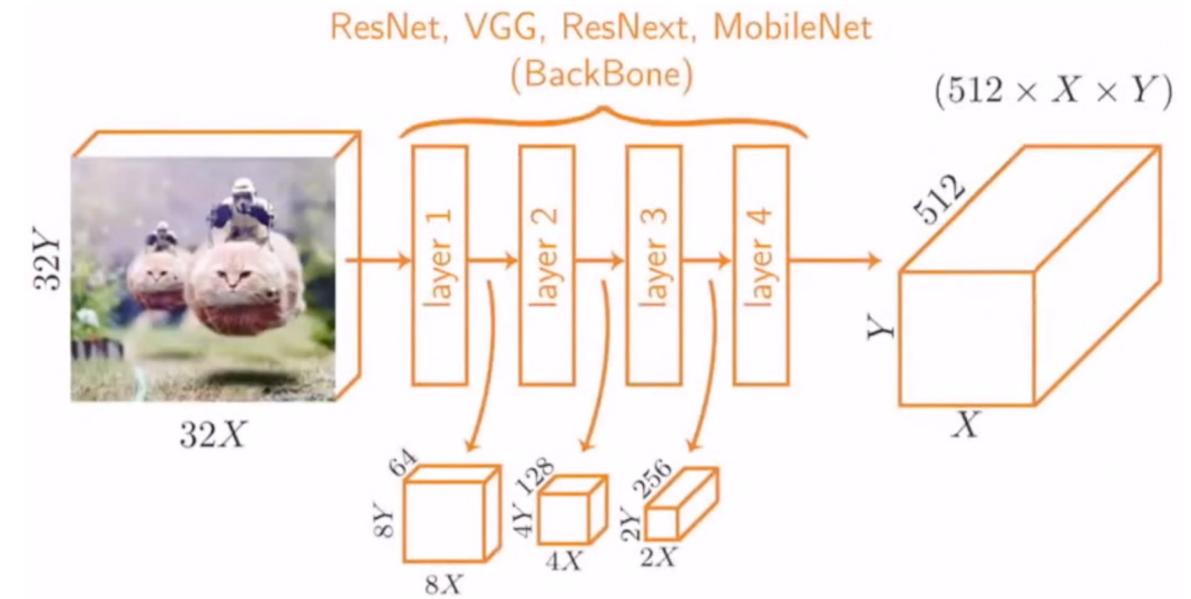
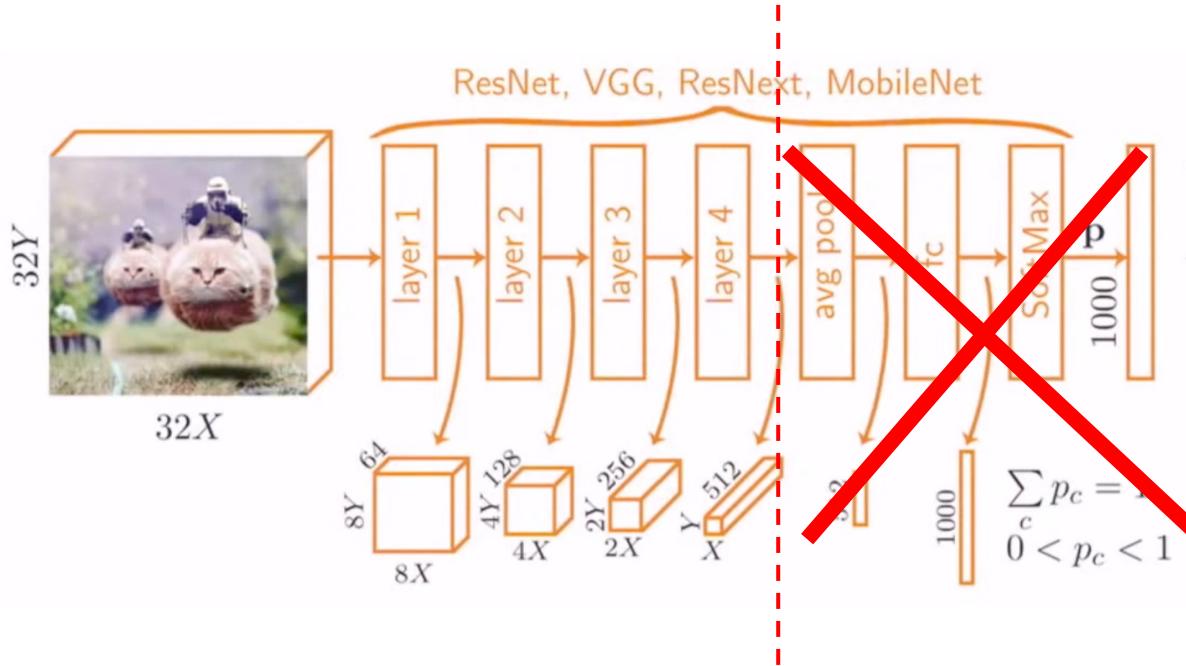


Одностадийный детектор

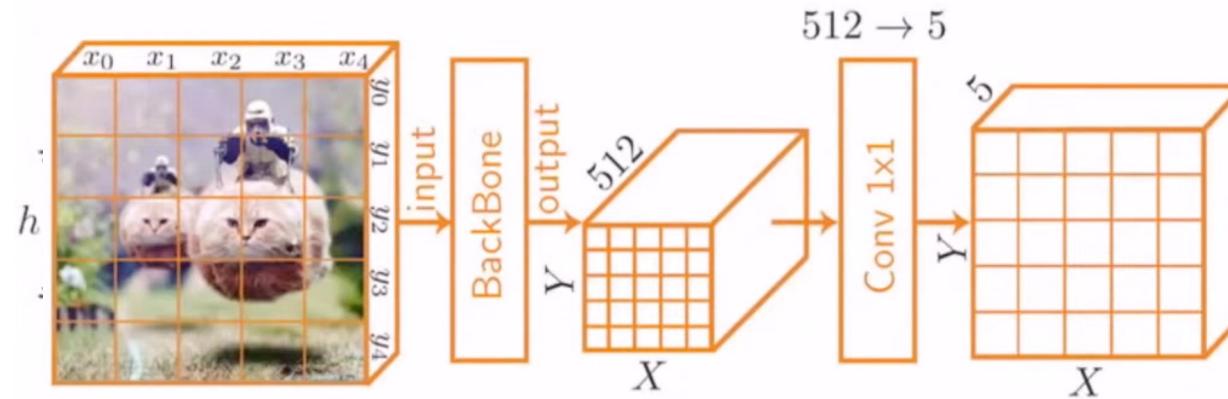


Двустадийный детектор

1. Извлечь признаки изображения с помощью нейросети backbone (ResNet, VGG, MobileNet).

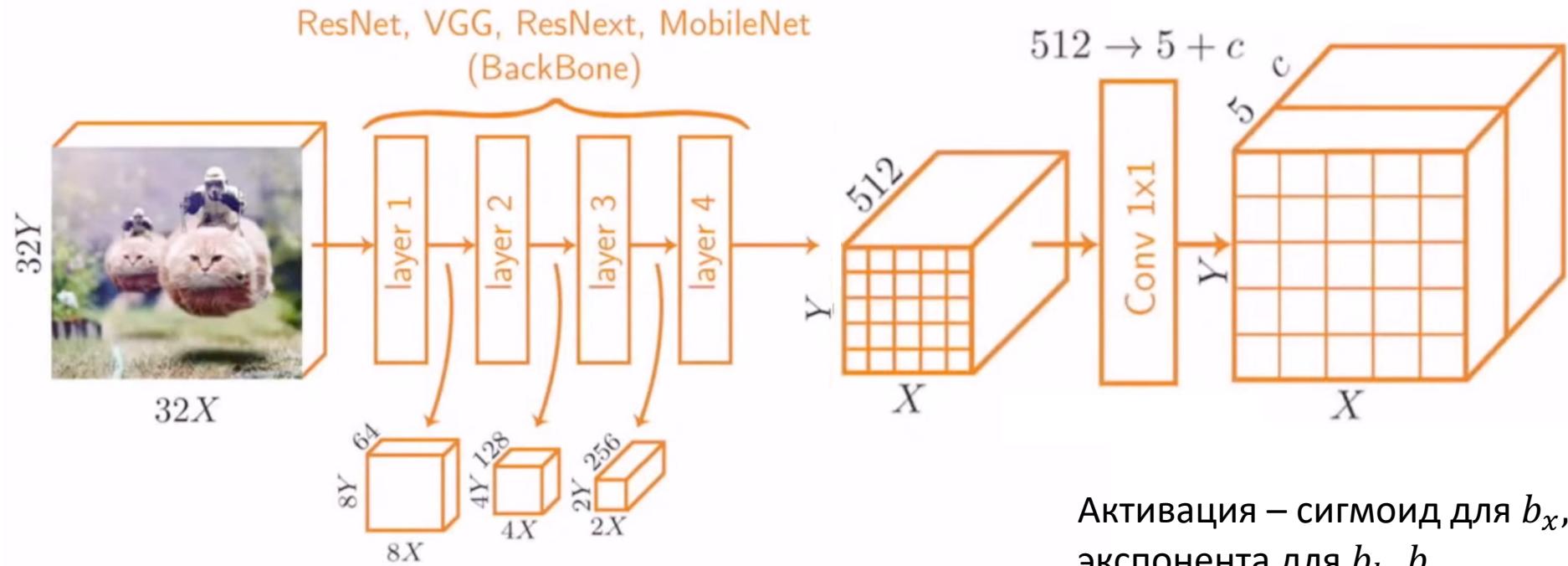


1. Извлечь признаки изображения с помощью нейросети backbone (ResNet, VGG, MobileNet).
2. По признакам изображения предсказать баундинг боксы и уверенность в них.



#1  $\rightarrow \sigma \rightarrow p$   
#2  $\rightarrow \sigma \rightarrow \Delta x$   
#3  $\rightarrow \sigma \rightarrow \Delta y$   
#4  $\rightarrow \exp \rightarrow w$   
#5  $\rightarrow \exp \rightarrow h$

1. Извлечь признаки изображения с помощью нейросети backbone (ResNet, VGG, MobileNet).
2. По признакам изображения предсказать баундинг боксы, метки классов и уверенность в них.
3. Постпроцессинг предсказаний.



Активация – сигмоид для  $b_x, b_y, p$ ,  
экспонента для  $b_h, b_w$ ,  
Softmax для метки класса.

$$\mathcal{L} = \sum_{pix} \mathcal{L}_{pix},$$
$$\mathcal{L}_{pix} = BCE(P, \tilde{I}) + \tilde{I} \cdot [BCE(b_x, \tilde{b_x}) + BCE(b_y, \tilde{b_y}) + |\log b_w - \log \tilde{b_w}| + |\log b_h - \log \tilde{b_h}| - \log p_c],$$

где  $\tilde{I}$  – индикатор наличия объекта,

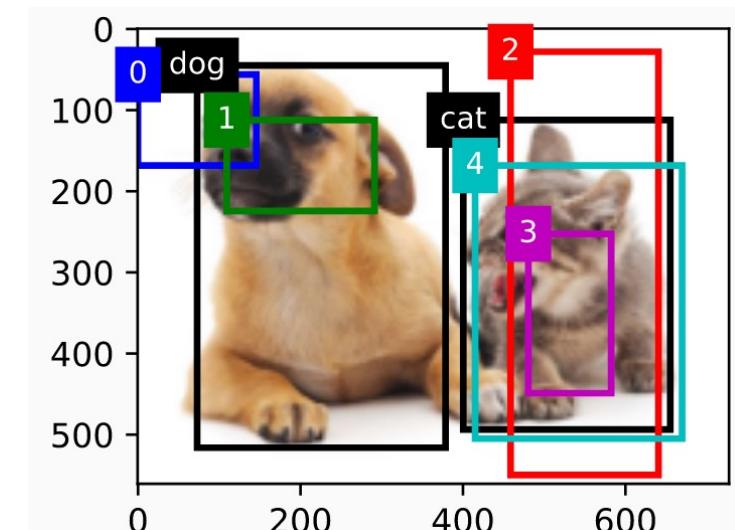
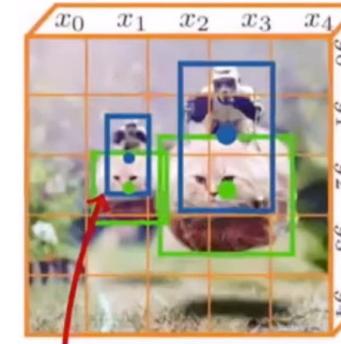
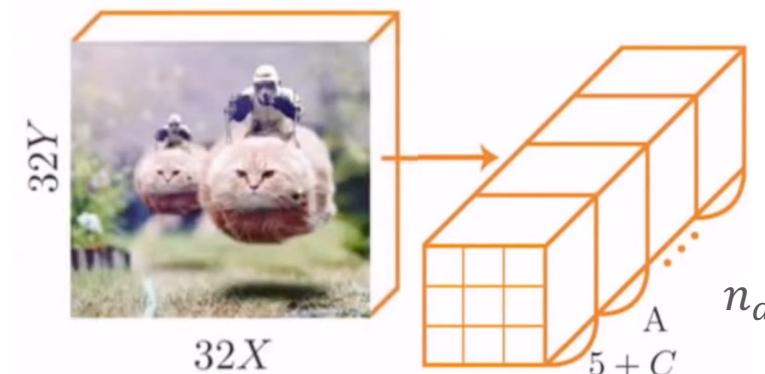
$P$  – предсказанная вероятность, что объект есть в ячейке,

$BCE(\cdot, \cdot)$  – бинарная кросс-энтропия,

$p_c$  – вероятность принадлежности к классу.

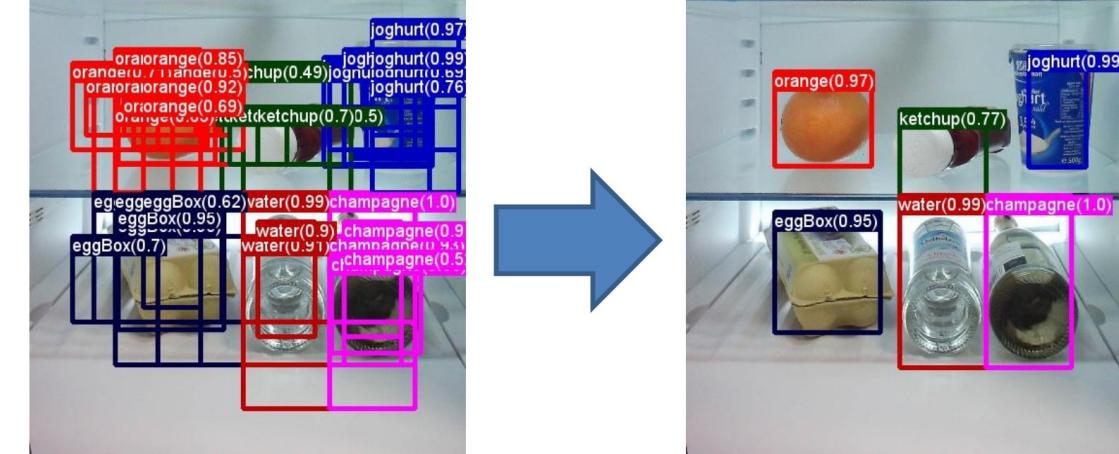
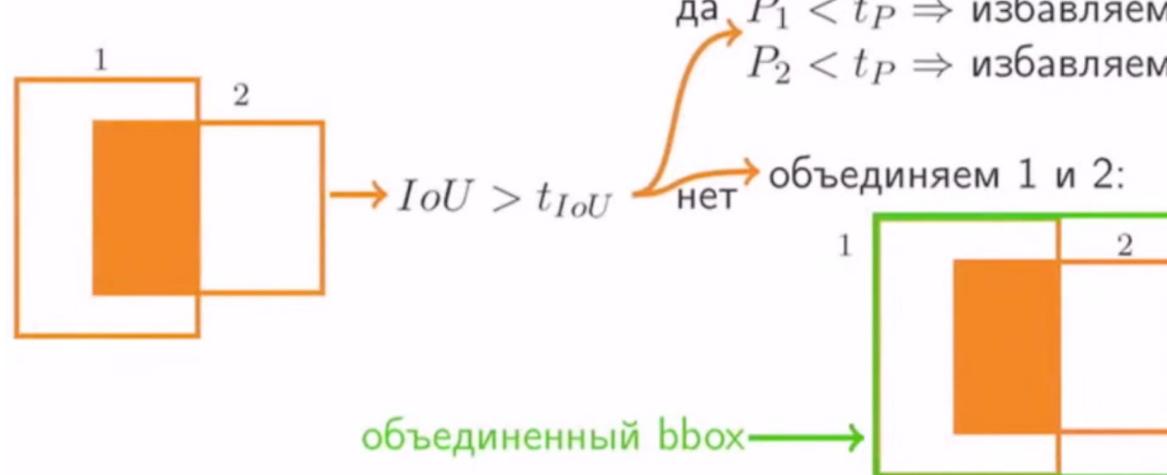
Нужно побрать коэффициенты.

- Проблема: два объекта попадают в одну ячейку.
- Можно делать по сети для каждого размера окна, но можно добавить якоря.
- Якоря – характерные размеры или пропорции объектов.
- В пределах одного суперпикселя предсказываем  $n_a$  объектов. Тогда последняя свёртка будет:  $512 \rightarrow (5 + c) \cdot n_a$ , где  $n_a$  – число якорей.
- Предсказываем размер объектов относительно якоря.
- Для таргета считаем IoU якоря и баундинг бокса.

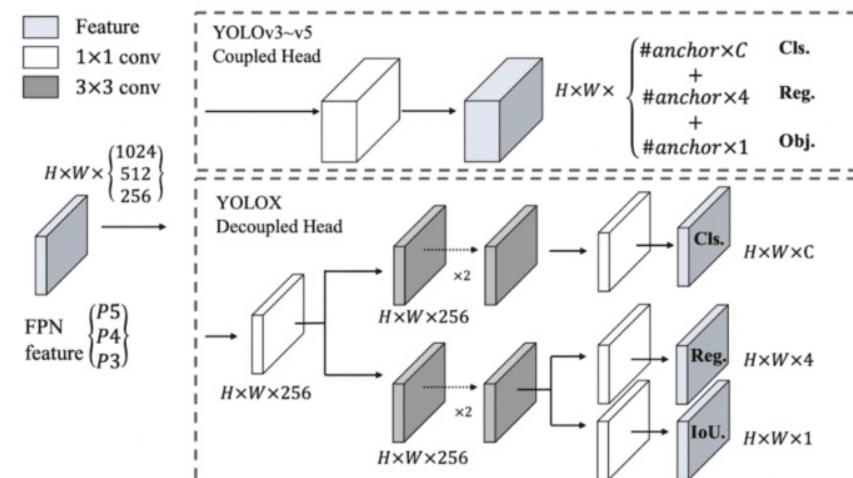


- Нейронные сети для детекции генерируют очень много баундинг боксов, нужно отсечь лишние, чтобы получить по одному на объект:
  - Можно выбрать порог для уверенности в детекции  $p$  (например 0,6).
  - Можно использовать алгоритм Non-maximum suppression.

$t_{IoU}$  – порог для IoU (например 0,5)



- YOLOv2 (2017) – добавлена нормализация батчей, более высокое разрешение, якоря.
- YOLOv3 (2018) – оценка предсказаний баундинг боксов, связи со слоями backbone, предсказание на разных уровнях.
- YOLOv4 (2020) – новые backbone на основе DenseNet, агрегация признаков, мозаичная аугментация данных, само-состязательное обучение.
- YOLOv5 (2020) – реализовано на PyTorch, аугментация данных, 16-битная точность.
- YOLOX (2021) – decoupled head, аугментация данных.
- YOLOv6 (2022) – EfficientRep Backbone и шея Rep-PAN, покональная дистилляция.
- YOLOv7 (2022) – улучшение агрегации слоев и многое другое.



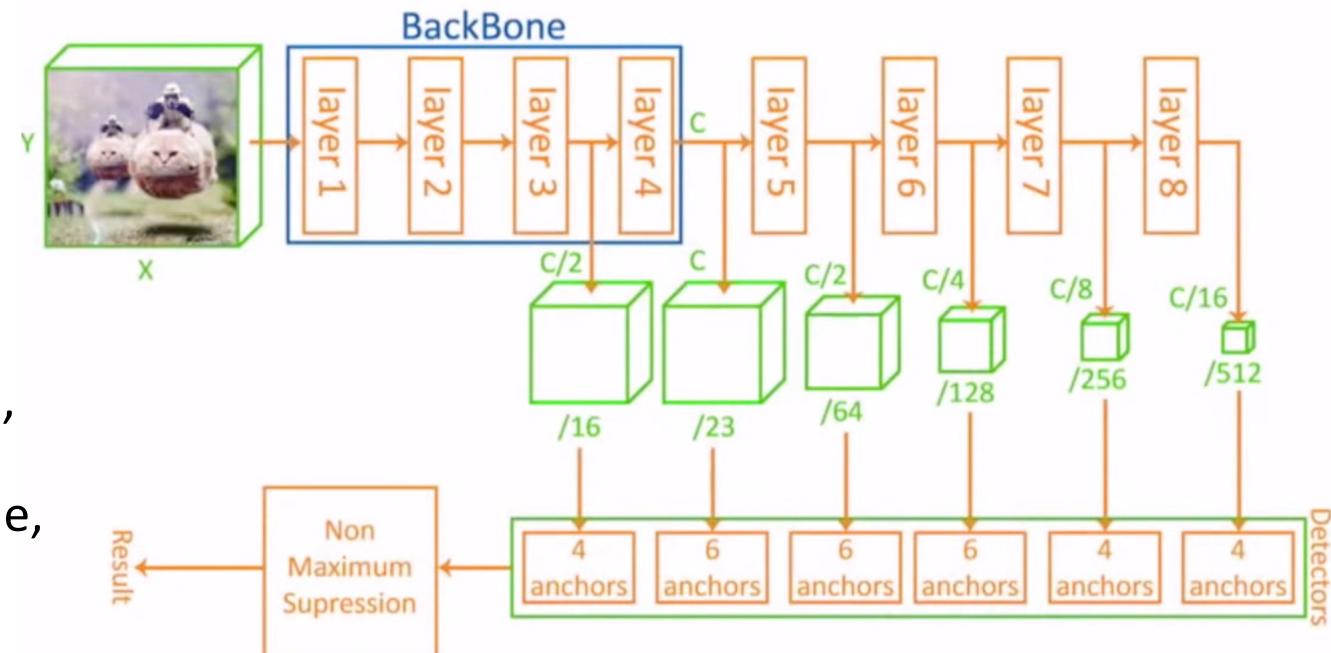
На вход обычно подается изображение фиксированного размера.

Предсказываем координаты центра объекта, ширину и высоту окна, метку класса и степень уверенности:  $b_x, b_y, b_h, b_w, label, p$ .

1. Извлечь признаки изображения с помощью нейросети backbone (ResNet, VGG, MobileNet).
2. По признакам изображения предсказать баундинг боксы, метки классов и уверенность в них.
3. Постпроцессинг предсказаний.

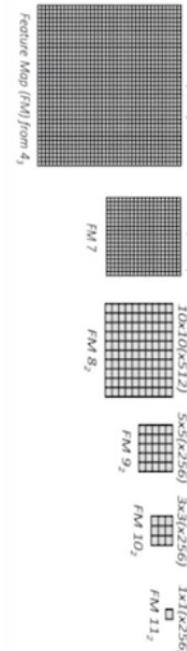
В YOLO предсказывали только на основании последней карты признаков.

К backbone добавляем 4 блока по две свертки, из них получаем карты признаков и еще 2 карты признаков с последних слоев backbone, итого 6 карт признаков.



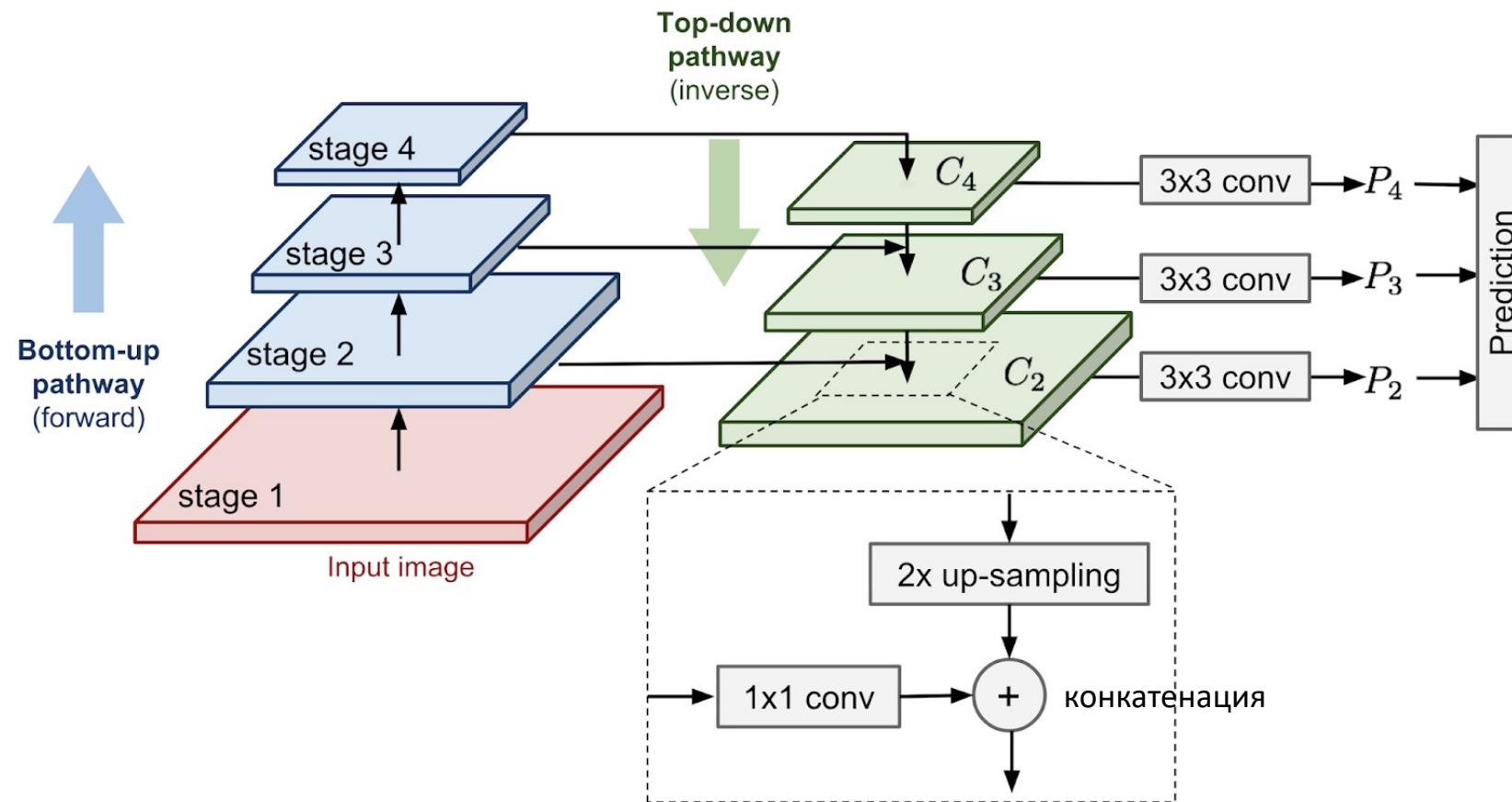
- Для каждой ячейки каждой карты признаки предсказываем баундинг боксы.

Чем раньше карта  
признаков, тем мельче  
объекты предсказываем.

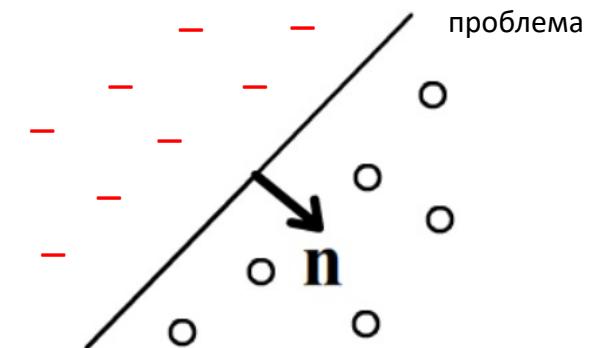


Feature Map From	Feature Map Dimensions	Prior Scale	Aspect Ratios	Number of Priors per Position	Total Number of Priors on this Feature Map
conv4_3	38, 38	0.1	1:1, 2:1, 1:2 + an extra prior	4	5776
conv7	19, 19	0.2	1:1, 2:1, 1:2, 3:1, 1:3 + an extra prior	6	2166
conv8_2	10, 10	0.375	1:1, 2:1, 1:2, 3:1, 1:3 + an extra prior	6	600
conv9_2	5, 5	0.55	1:1, 2:1, 1:2, 3:1, 1:3 + an extra prior	6	150
conv10_2	3, 3	0.725	1:1, 2:1, 1:2 + an extra prior	4	36
conv11_2	1, 1	0.9	1:1, 2:1, 1:2 + an extra prior	4	4
<b>Grand Total</b>	-	-	-	-	<b>8732 priors</b>

- В SSD мелкие объекты предсказываются из ранних карт признаков, но в них нет информации о контексте (маленький receptive field)!
- Контекст может быть очень важен (автомобиль на дороге, а не на дереве).



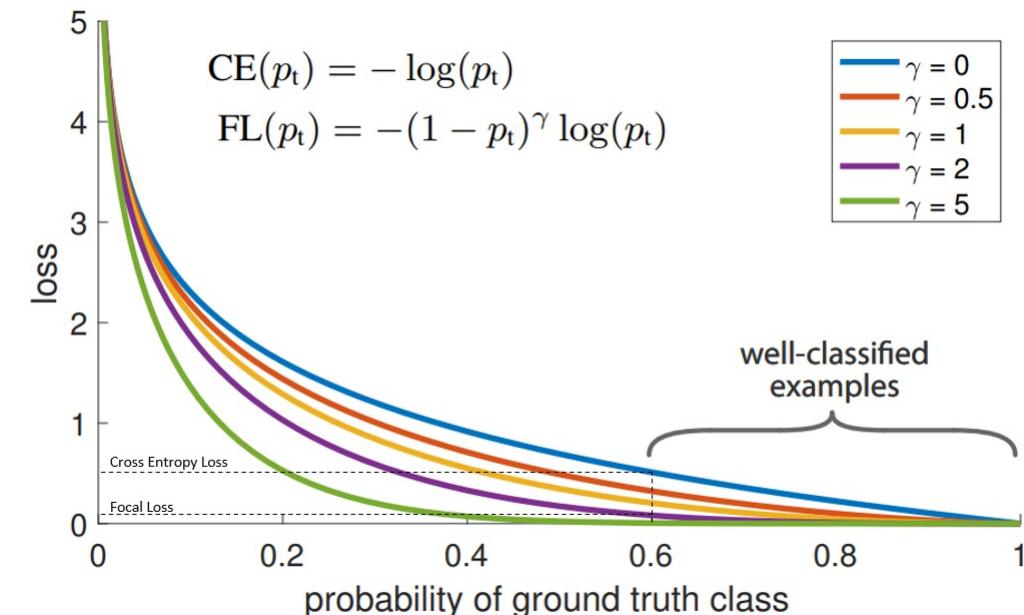
- При решении задачи детекции очень мало положительных объектов и много отрицательных!
- Если уже умеем предсказывать правильный класс, то уменьшим влияние хорошо классифицированных объектов.



$p_t$  – вероятность верного класса.

$CE = -\log(p_t)$  – ошибка предсказания правильного класса, но только ее недостаточно.

$$FL = (1 - p_t)^\gamma \cdot \log(p_t)$$





УНИВЕРСИТЕТ ИТМО

# Задача детекции

## Двустадийная детекция

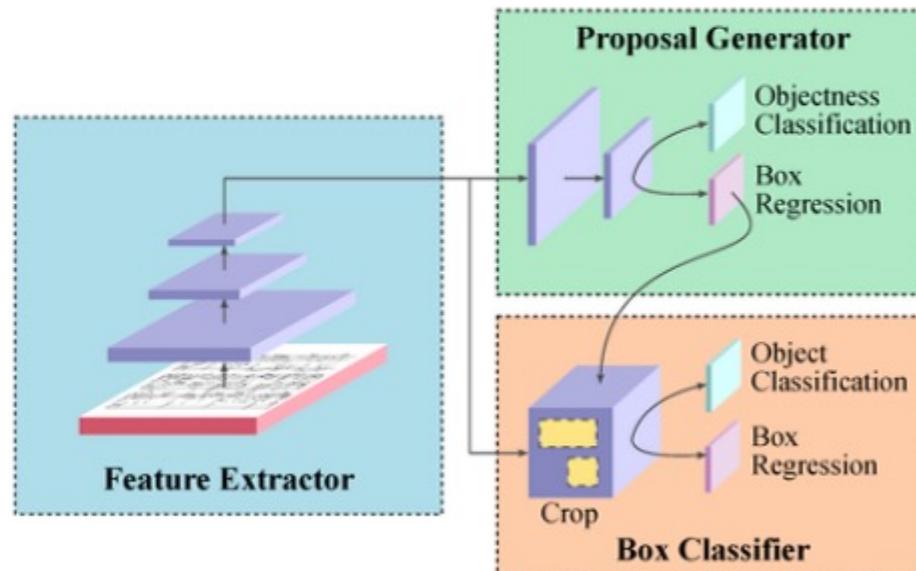
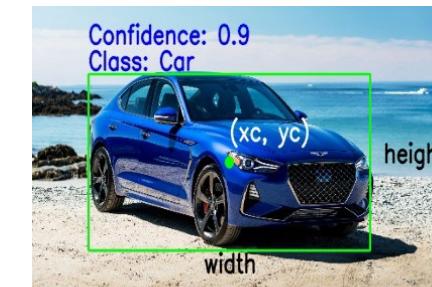
Ефимова Валерия Александровна

[vefimova@itmo.ru](mailto:vefimova@itmo.ru)

12.10.2022

# Задача двустадийной детекции

Найти объекты интересующих классов и их расположение: координаты центра объекта, ширина и высота окна, метка класса и степень уверенности:  $b_x, b_y, b_h, b_w, label, p$ .

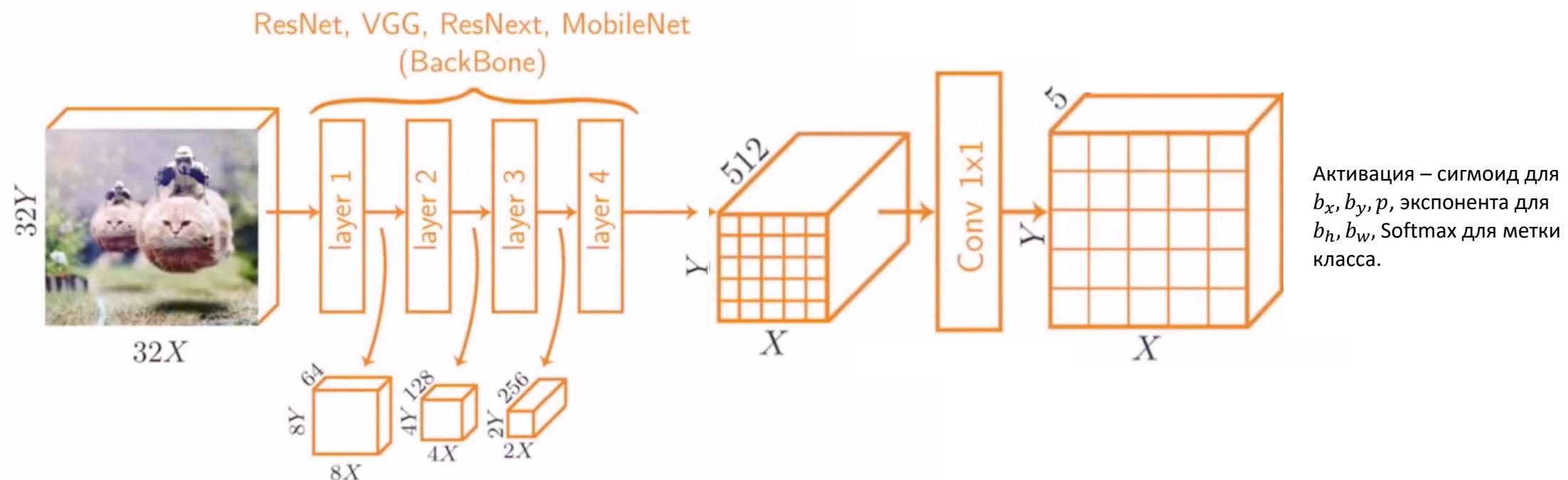


Определяем интересующие области

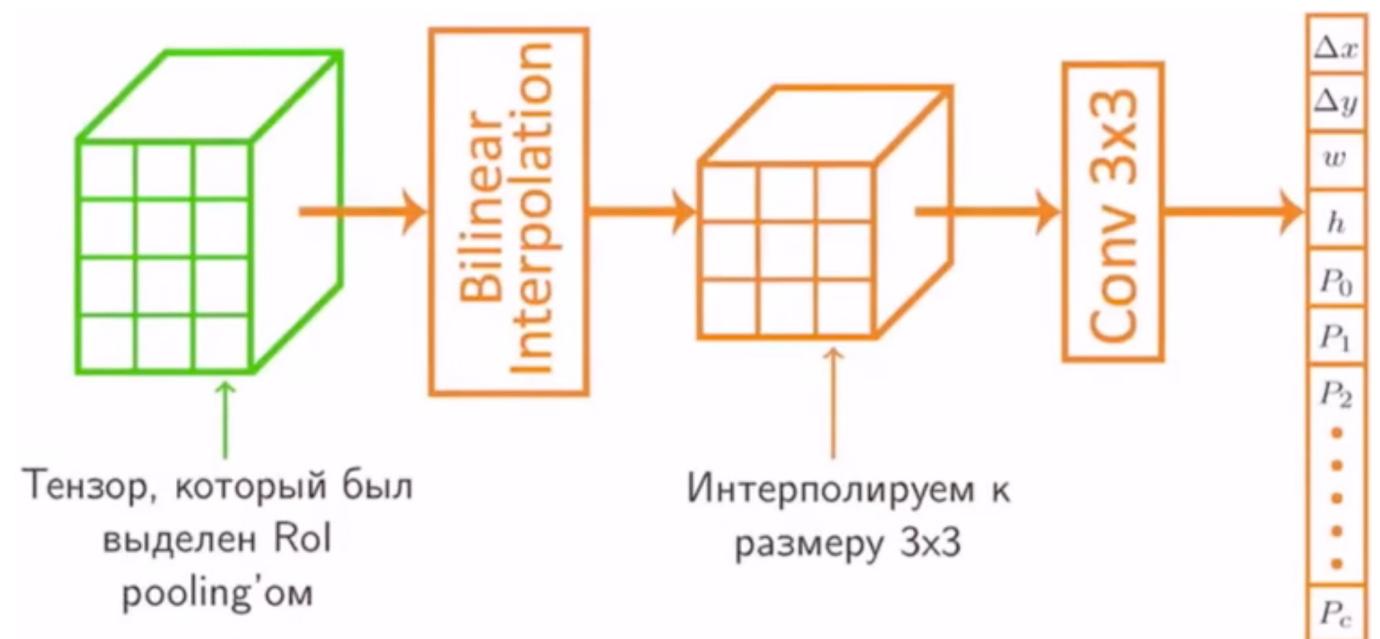
«Рассматриваем под микроскопом» интересующие регионы.

Двустадийный детектор

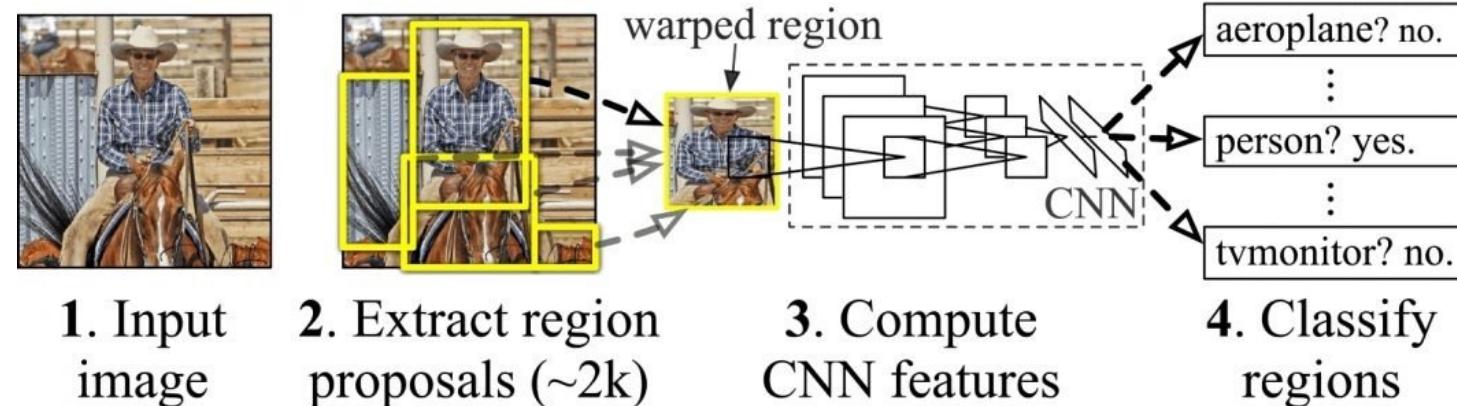
- Сначала RPN основывался на классическом CV.
- То же, что в YOLO, но без классификации.
- Функция ошибки:  $\mathcal{L} = \sum_{pix} \mathcal{L}_{pix}, \quad \mathcal{L}_{pix} = BCE(P, \tilde{I}) + \tilde{I} \cdot (BCE(b_x, \tilde{b}_x) + BCE(b_y, \tilde{b}_y) + |\log b_w - \log \tilde{b}_w| + |\log b_h - \log \tilde{b}_h| - \log p_c)$ .
- Получаем очень много кандидатов (не обязательно верных) на объект, применим NMS.



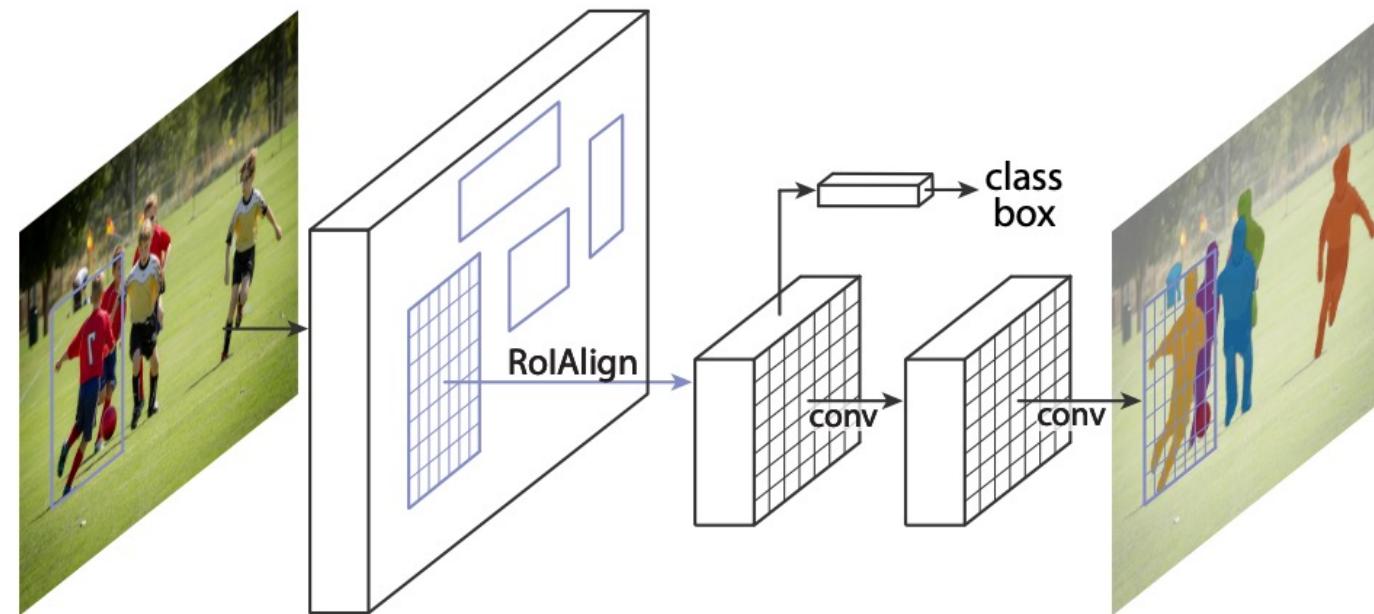
- RPN отбрала интересные области.
  - В идеале: рассмотрим области по одной, уточним и классифицируем.
- Переиспользуем предсказанные RPN баундинг боксы: вырежем их в тензорах.
  - Проблема: углы предсказанных баундинг боксов не обязательно в углу ячейки.
  - Расширяем баундинг боксы по сетке.
- Билинейная интерполяция тензора к *фиксированному размеру (например, 3x3)*.
- Несколько сверток, например, 3x3.
- Предсказание центра, размеров, класса.



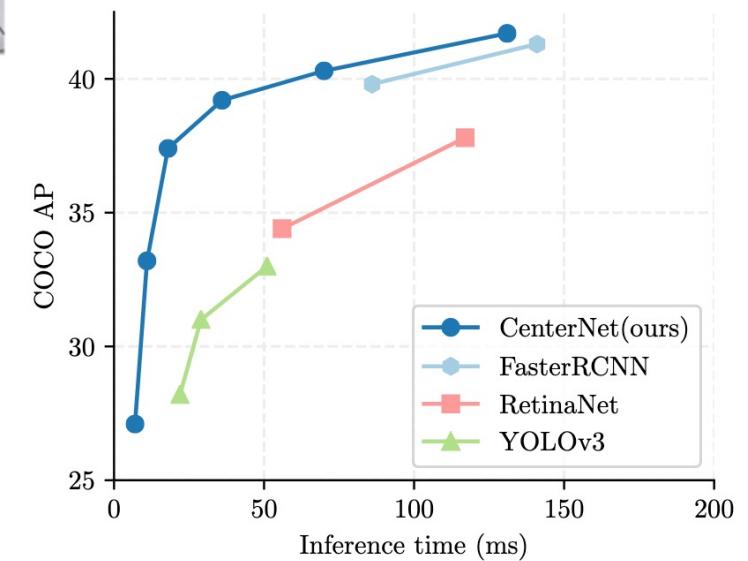
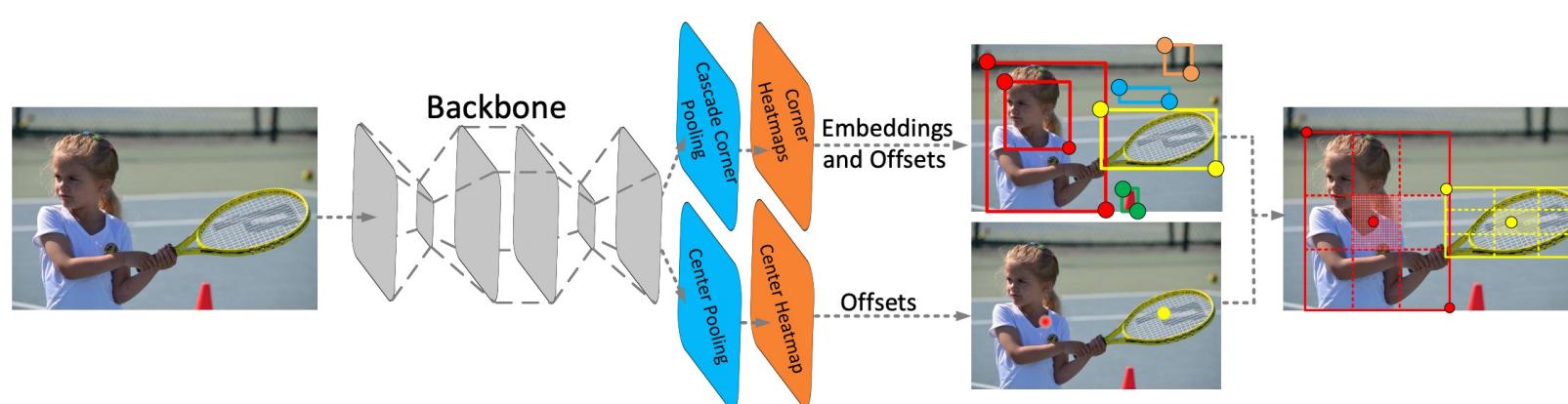
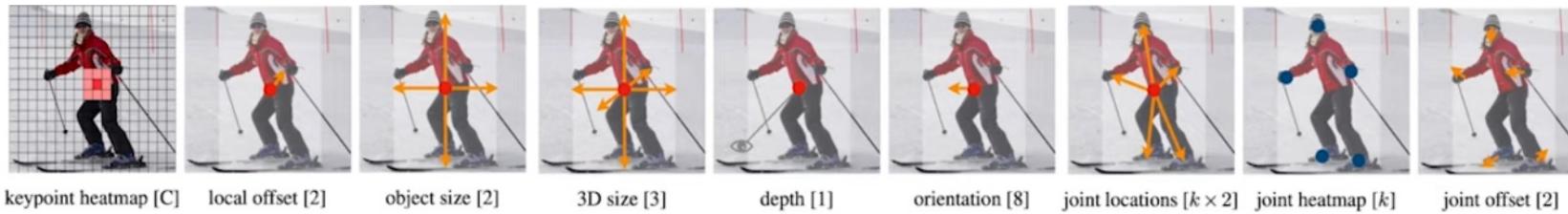
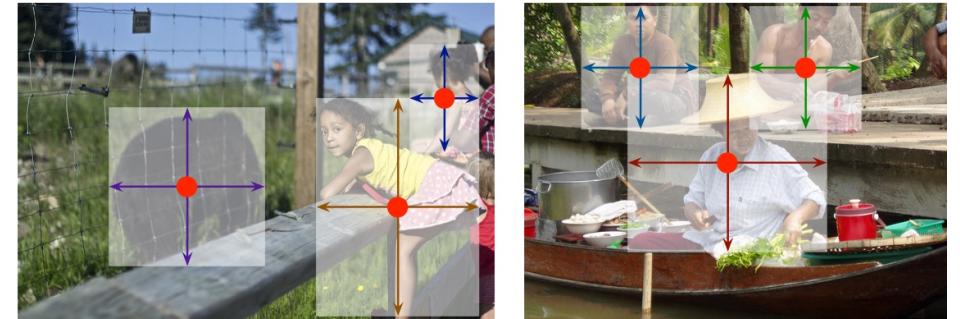
- RPN обучается как YOLO.
- Проблема: операция RoI пулинга (округления) не дифференцируема → нужно обучать отдельно.
- Эволюция: R-CNN (RPN классическими методами CV) → Fast R-CNN (RPN – нейронная сеть) → Faster R-CNN (объединили признаки RPN и RoI пулинга) → Mask R-CNN (предсказываем еще маску).



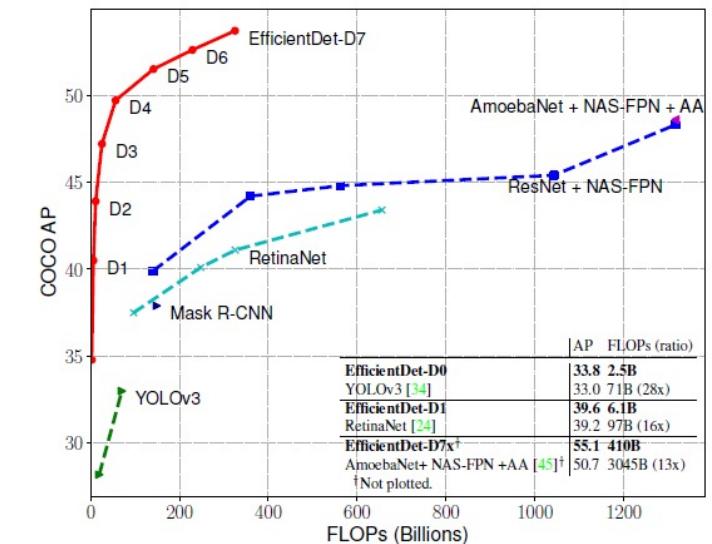
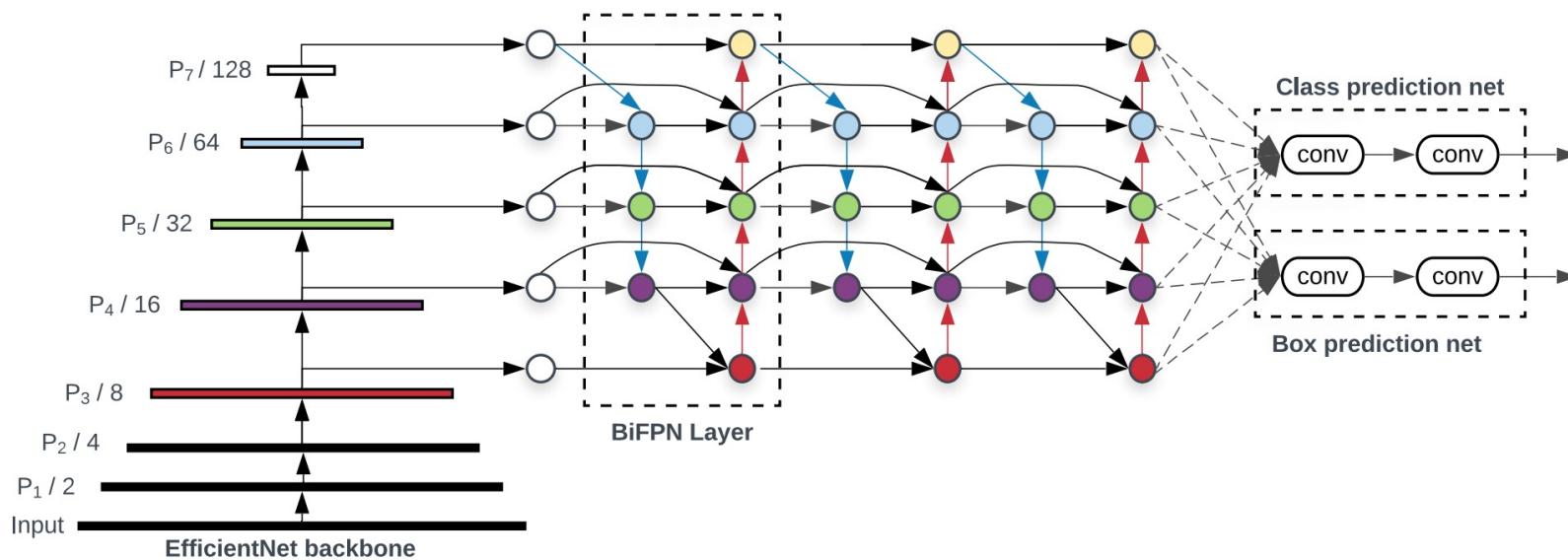
- Из вырезанного тензора можем предсказывать маску объекта.
- Сложнее разметка данных, так как нужны маски.
- Добавляется ошибка на сегментацию – позволило уточнить детекцию.



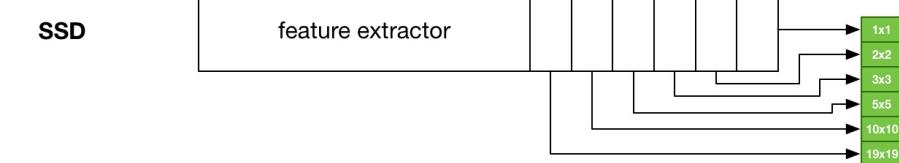
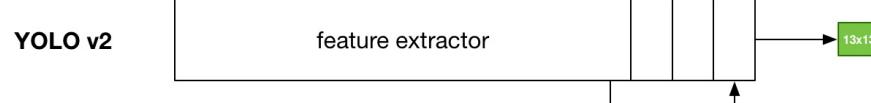
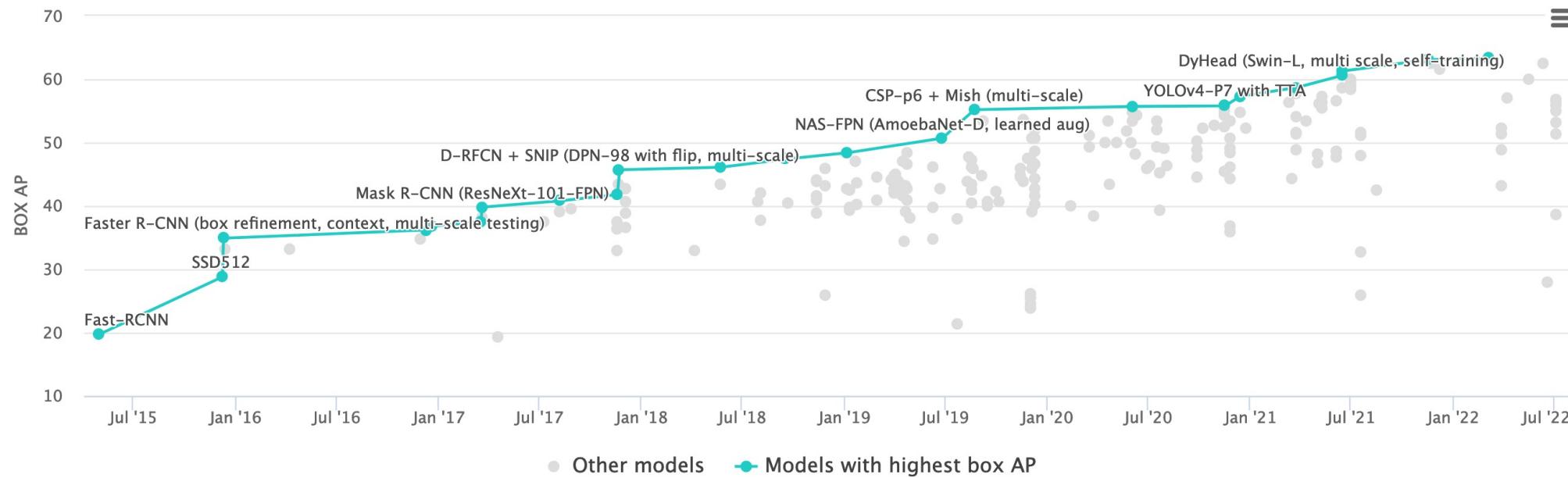
- Определение ключевых точек.
- Предсказываем центр бокса, ширину и высоту и вероятность нахождения тела в каждой точке.
- Предсказываем еще параметры: глубину, позу человека, положение суставов.



- EfficientNet + BiFPN

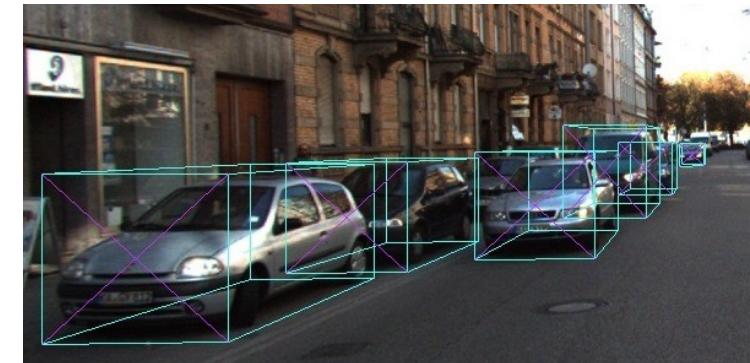
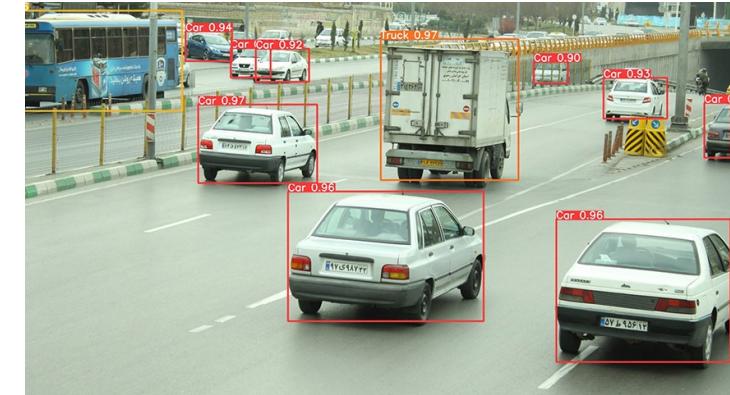
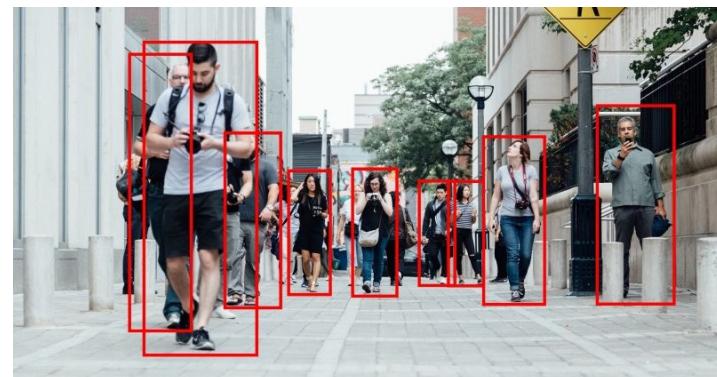
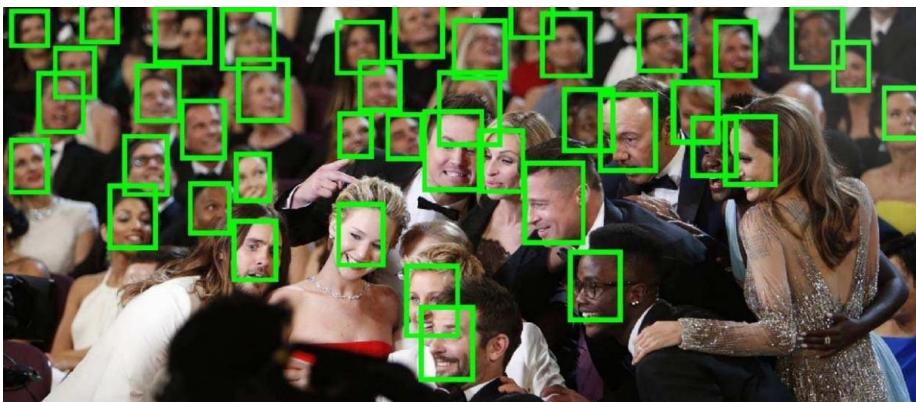


Рассмотрели задачу детекции.



# Детектирование объектов на практике

- Автотранспорт
- Люди (подсчет, попадание в область, расстояние между людьми, контроль на заводах)
- Лица (маски, выявление определенных людей) – обычно маленькие, отдельные архитектуры.
- Текст (перевод по картинке, ценник в магазине)
- Трекинг
- Действия (драки, курение, передача из рук в руки)
- В 3D





УНИВЕРСИТЕТ ИТМО

# Спасибо за внимание!

ОБРАЗОВАТЕЛЬНЫЕ ПРОГРАММЫ В ОБЛАСТИ  
ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА



УНИВЕРСИТЕТ ИТМО

# Распознавание текста

Ефимова Валерия Александровна

[vefimova@itmo.ru](mailto:vefimova@itmo.ru)

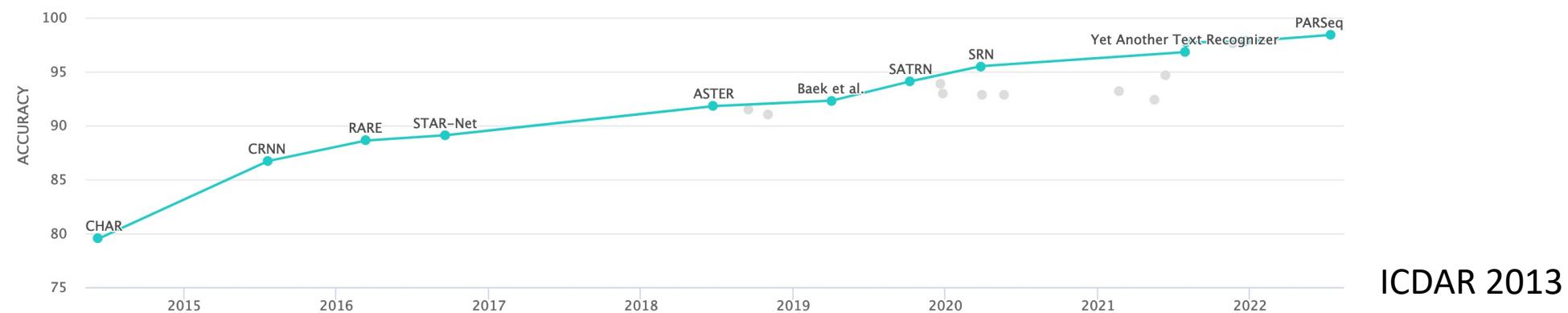
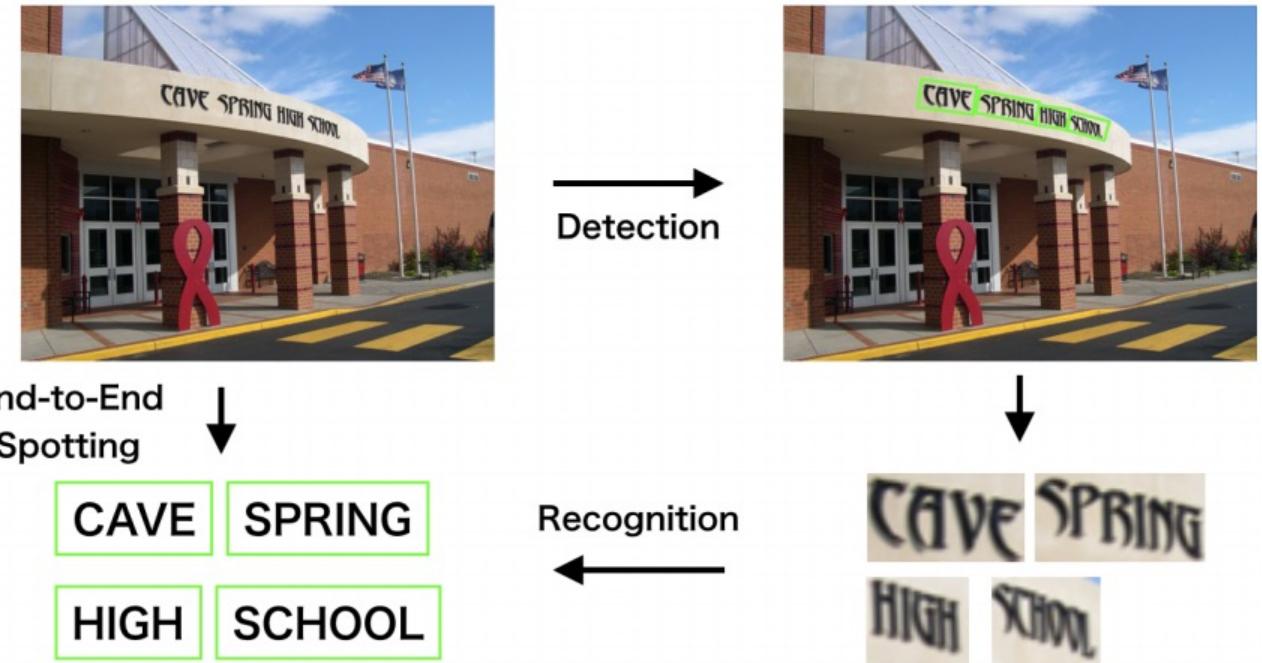
12.10.2022

Оптическое распознавание символов (англ. optical character recognition, OCR) – представление текста на изображении в виде электронных символов.

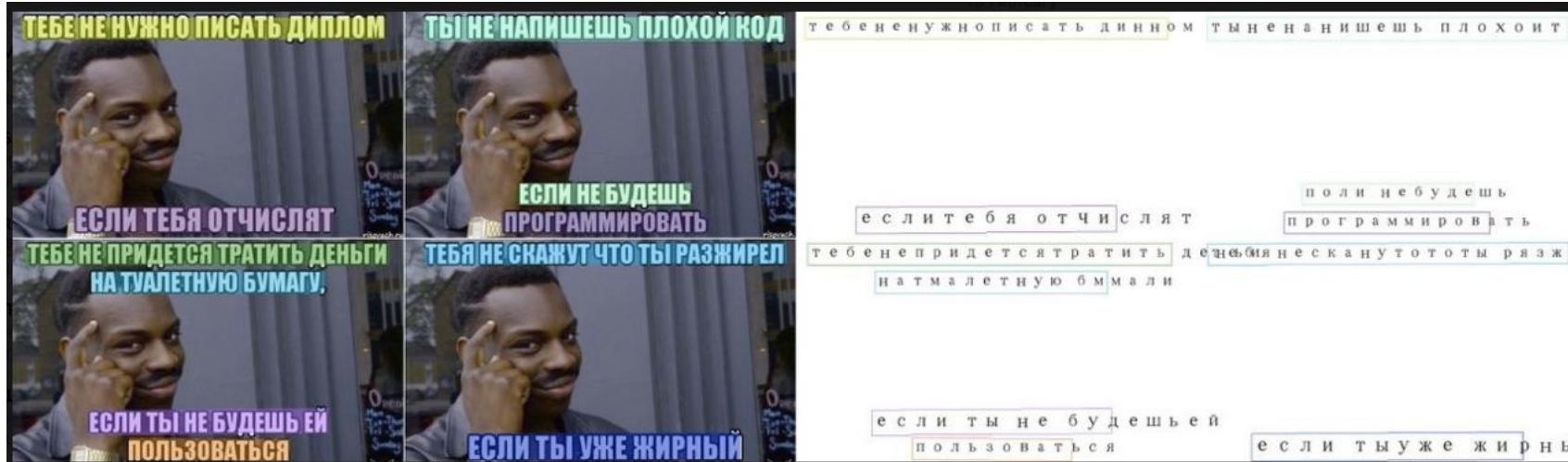
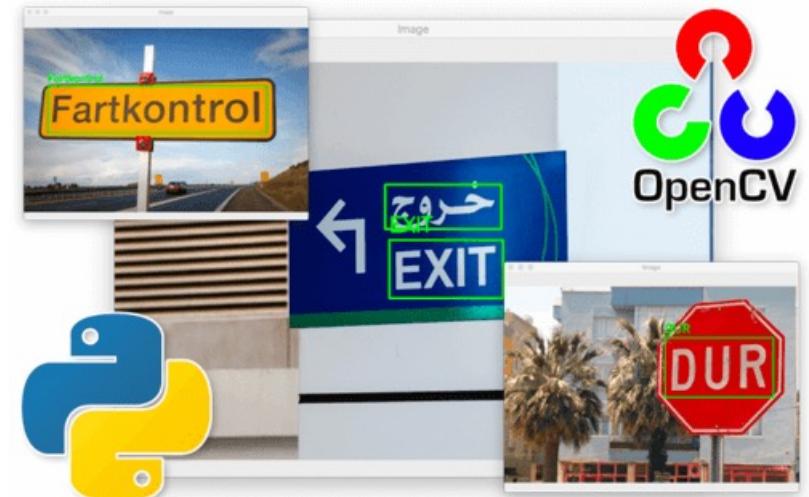
Разнообразие текста (язык, где, как написан, освещение, фон):



- Классическими методами
- Локализация
- Распознавание символов
- Сквозное распознавание (end-to-end)



- Tesseract OCR
  - <https://github.com/tesseract-ocr/tesseract>
- PaddleOCR
  - <https://github.com/PaddlePaddle/PaddleOCR>
- EasyOCR
  - <https://github.com/JaideAI/EasyOCR>
- Kraken
  - <https://github.com/mittagessen/kraken>



- Рассмотрели задачу детекции и методы оценки ее качества: IoU, mAP.
- Детектируются сотни баундинг боксов (для объектов разных размеров и форм для всех классов), но только единицы из них верные – выбираем с помощью Non-Maximum Suppression.
- Одностадийные детекторы позволяют достичь большей скорости детекции, чем двустадийные, но не очень хорошо находят маленькие объекты и незначительные различия.
- Текст на английском языке распознается довольно точно, но с распознаванием русского языка есть проблемы.

- <https://colab.research.google.com/drive/1laTDPPrboJlnz5noBNVmzmWJBBUsTP51S#scrollTo=sDQsemJkDWuL>
- [https://github.com/open-mmlab/mmdetection/blob/master/demo/MMDet\\_Tutorial.ipynb](https://github.com/open-mmlab/mmdetection/blob/master/demo/MMDet_Tutorial.ipynb)
- <https://colab.research.google.com/drive/1X9A8odmK4k6l26NDviiT6dd6TgR-piOa>



УНИВЕРСИТЕТ ИТМО

# Спасибо за внимание!

ОБРАЗОВАТЕЛЬНЫЕ ПРОГРАММЫ В ОБЛАСТИ  
ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА