МИСИС
УНИВЕРСИТЕТ

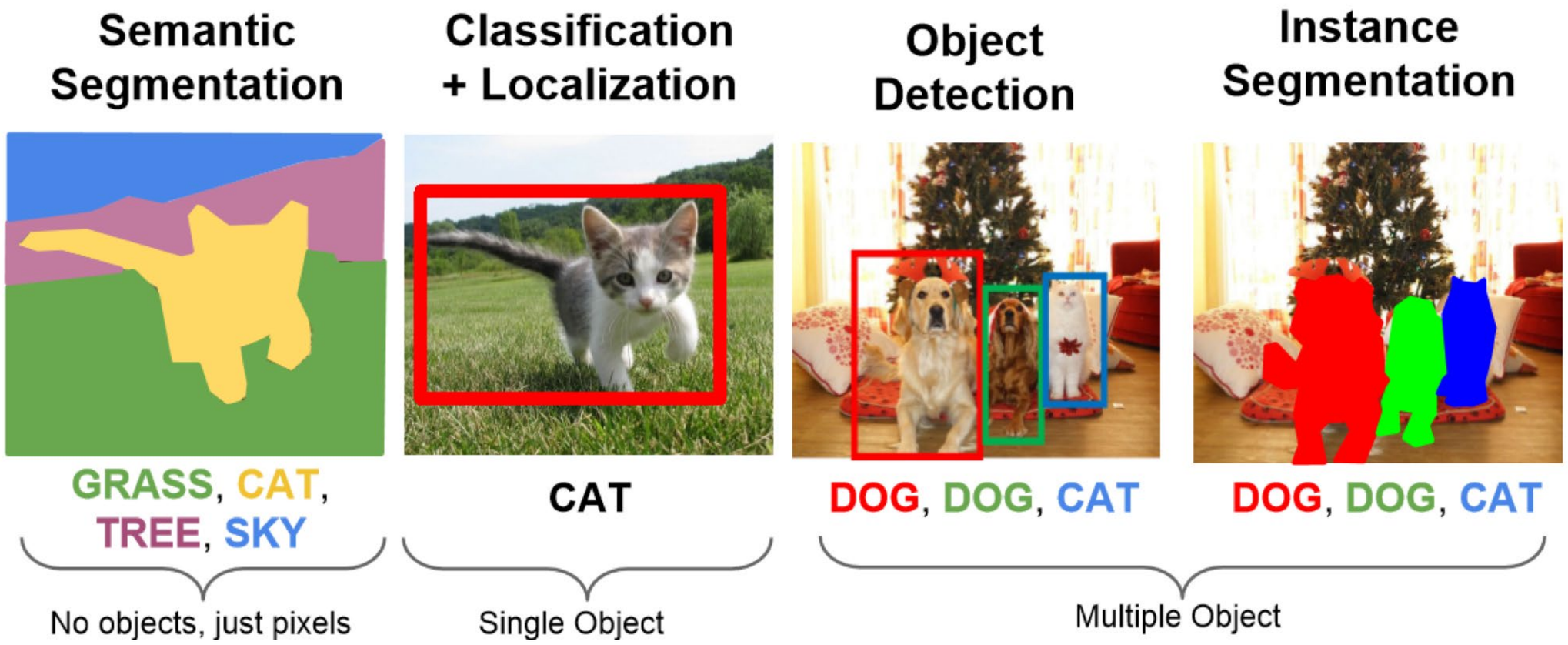
Детекция и сегментация

Корнет Мария Евгеньевна
старший преподаватель кафедры
Инженерная кибернетика

План

- Задачи компьютерного зрения
- Двухстадийные детекторы
- Одностадийные
- Сегментация

Задачи CV



https://github.com/nikkkkhil/SemanticSegmenteation

Компьютерное зрение

Классификация = что изображено?

Локализация = поиск местоположения одного объекта.

Детекция = Классификация + Локализация (для нескольких объектов).

Семантическая сегментация — присвоение каждому пикселю метки класса.

Инстанс-сегментация — разделение отдельных экземпляров объектов одного класса.

Ограничивающая рамка (bounding box) – координаты, ограничивающие определенную область изображения, чаще всего в форме прямоугольника.

Датасеты

Датасет	Год	Тренировочные	Кол-во объектов	Валид.	Тест	Особенности
VOC-2007	2007	2,501	6,301	2,510	4,952	20 классов, простые сцены
VOC-2012	2012	5,717	13,609	5,823	10,991	Расширенный VOC с новыми аннотациями
ILSVRC-2014	2014	456,567	478,807	20,121	—	Детекция на основе ImageNet
ILSVRC-2017	2017	456,567	478,807	55,502	65,500	Финальные соревнования ImageNet Detection
MS COCO-2015	2015	82,783	607,907	40,504	81,434	Многообъектные сцены, 80 классов
MS COCO-2018	2018	118,287	860,001	5,000	40,670	Используется для сегментации и keypoint detection
OID-2018 (Open Images)	2018	1,743,042	14,610,229	204,621	125,436	Огромный масштаб, более 600 классов

Датасеты

Название	Год / версия	Основные характеристики
LVIS (Large Vocabulary Instance Segmentation)	2019 (версия v1.0)	≈160 k изображений, ~2 млн аннотированных экземпляров, ~1200 классов (длинный хвост). Инстанс-сегментация + детекция. (
iSAID (Instance Segmentation in Aerial Images)	2019	Аэроснимки: 2 806 изображений, 655 451 экземпляр, 15 классов. Инстанс-сегментация/детекция.
TTPLA (Transmission Towers & Power Lines Aerial Image Dataset)	2020	1 100 изображений (разрешение 3840×2160), 8 987 экземпляров башен и линий электропередачи, задачи детекции и сегментации.
Argoverse 2 (Sensor Dataset for Autonomous Driving)	~2021/2022	1 000 сценариев с 3D-аннотациями объектов (лидар + камеры + HD-карты). 26+ категорий, задача детекции (3D), трекинга, прогнозирования.
Open Images V4 (обновлённая версия)	~2019	Миллионы изображений, сотни классов; новая версия включает маски сегментации (выпуск 2019).

Структура датасетов

```
dataset_name/
│
├── images/                    # Папка с изображениями
│   ├── train/                 # Обучающая выборка
│   ├── val/                   # Валидационная
│   └── test/                  # Тестовая
│
├── annotations/               # Разметка
│   ├── instances_train.json   # Разметка инстанс-сегментации
│   ├── instances_val.json
│   ├── bbox_train.json        # Для детекции
│   ├── keypoints.json         # Для поз или скелетов
│   └── panoptic_train.json    # Для паноптической сегментации
│
├── masks/                     # (не всегда) Пиксельные маски для сегментации
│   ├── image_0001.png
│   └── image_0002.png
│
└── classes.txt / labels.txt   # Список классов (по одному в строке)
```

Структура датасетов

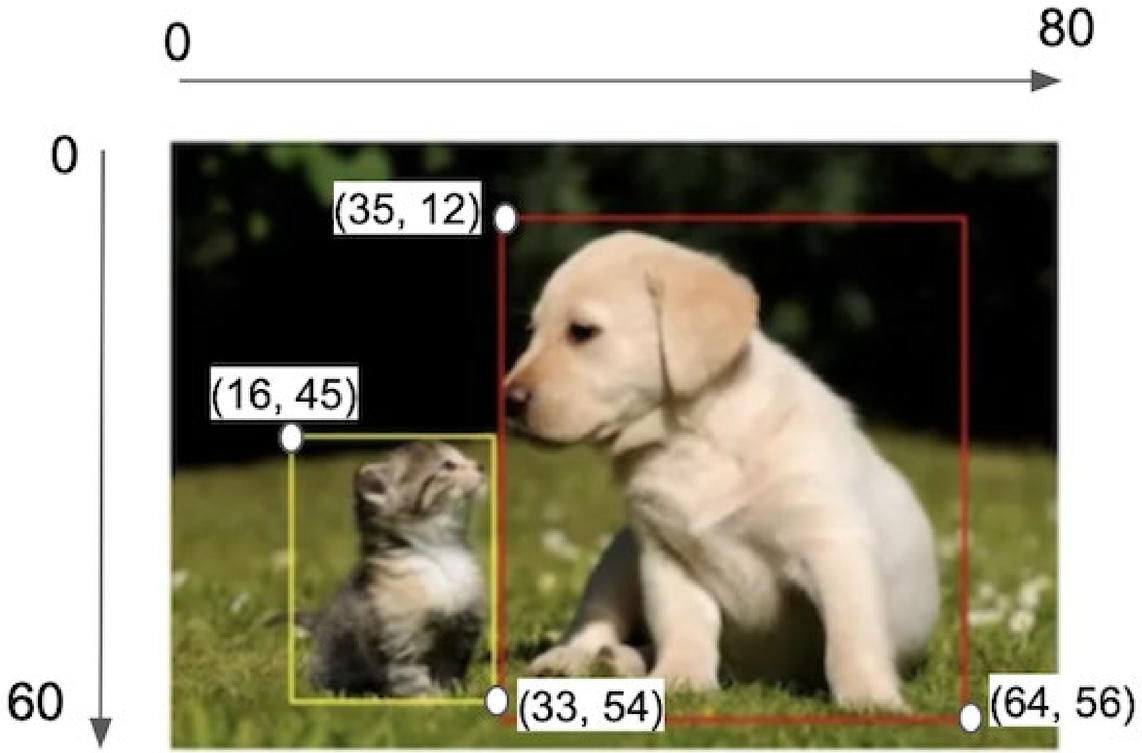
MS COCO

```
coco/
├── images/
│   ├── train2017/
│   ├── val2017/
│   └── test2017/
└── annotations/
    ├── instances_train2017.json
    ├── captions_train2017.json
    └── person_keypoints_train2017.json
```

Pascal VOC

```
VOCdevkit/
└── VOC2012/
    ├── JPEGImages/          # Изображения
    ├── SegmentationClass/   # Цветные маски классов
    ├── Annotations/         # XML-файлы с bboxes
    └── ImageSets/Main/      # train/test списки
```

Пример



Набор пар (*класс, координаты bounding box*).

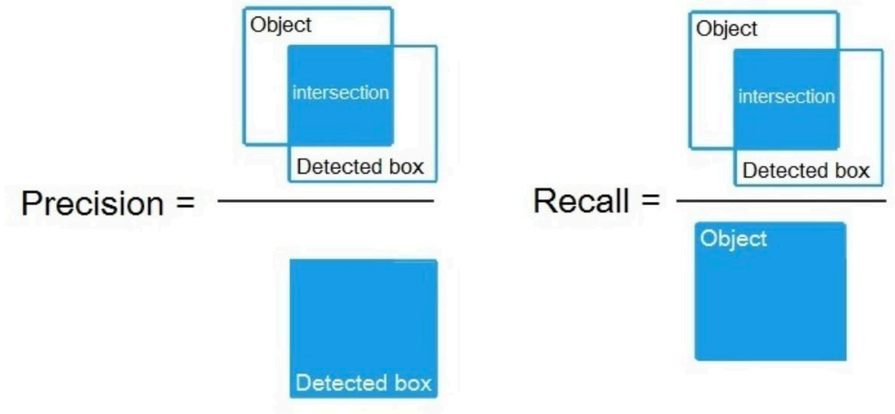
bounding box — это пара [x1, y1], [x2, y2],
где [x1, y1] — координаты левого верхнего угла,
[x2, y2] — координаты правого нижнего угла.

Пример:
(собака, [35, 12], [64, 56]);
(кошка, [16, 45], [33, 54]);

Основные метрики детекции

IoU (Intersection over Union) — отношения площади пересечения предсказанной и истинной рамки к площади их объединения.

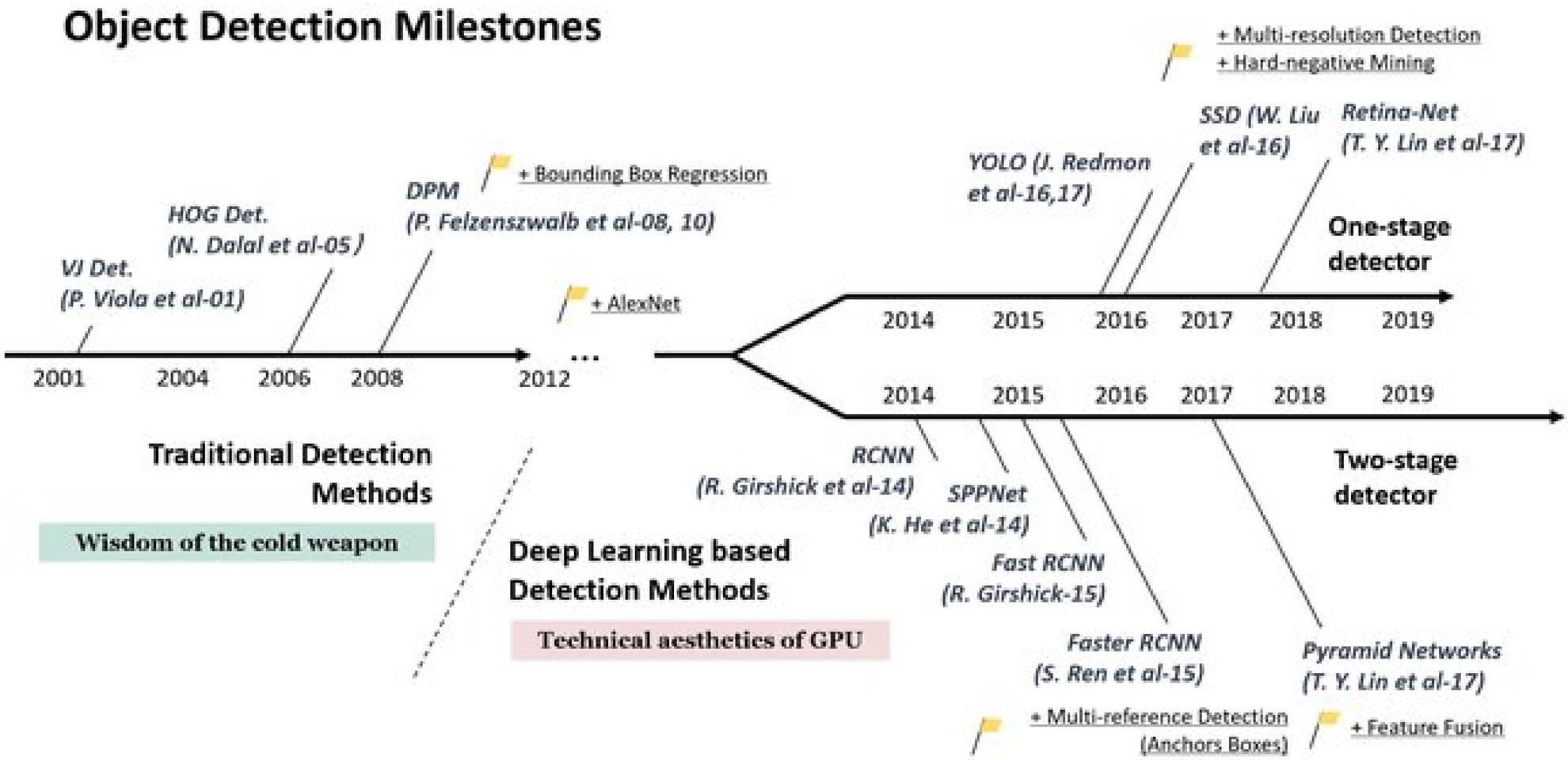
$$IoU = \frac{S_{\text{пересечения}}}{S_{\text{объединения}}} \geq \alpha$$



Precision — доля верных предсказаний среди всех найденных.

Recall — доля найденных объектов среди всех присутствующих.

AP/mAP — средняя точность по всем порогам IoU и классам.

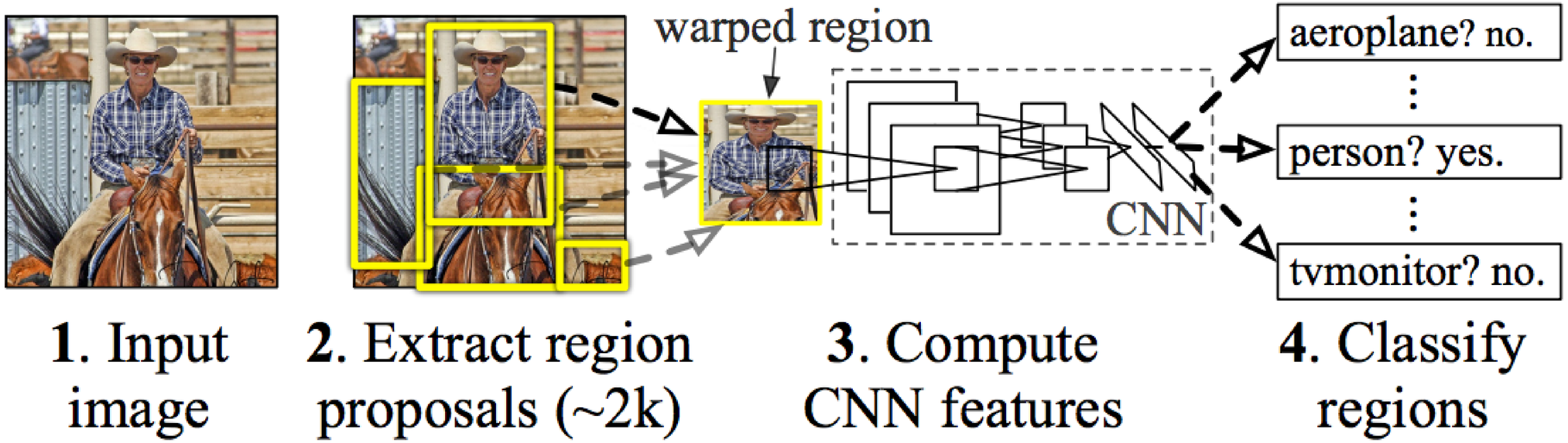


https://arxiv.org/abs/1905.05055

Двухстадийные детекторы

Основная идея: разделить изображение на множество регионов и классифицировать каждый из них.

R-CNN (Region Convolution Neural Network), Fast R-CNN, Fasten R-CNN

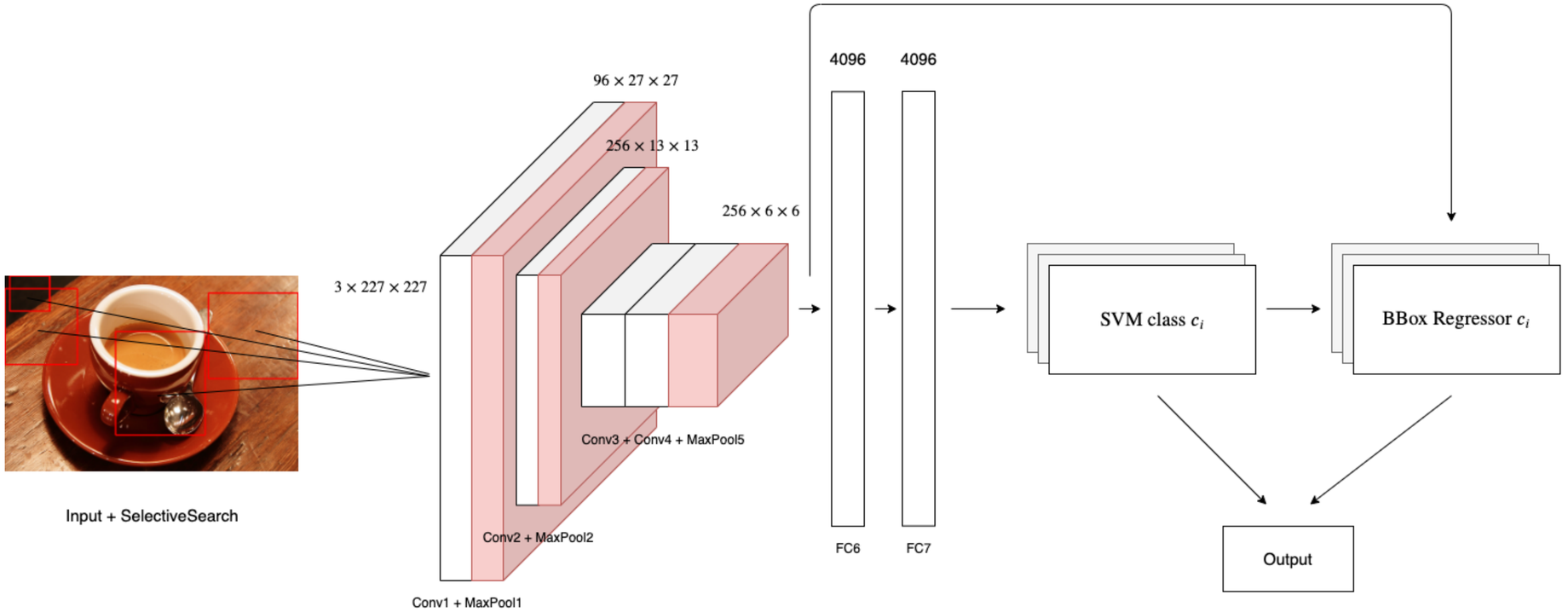


Двухстадийные детекторы

1. **Сгенерировать регионы-кандидаты:** Найти на изображении области, которые с большой вероятностью содержат объекты (не важно, какие именно) - гипотезы.
   Пример: ~2000 регионов от алгоритма Selective Search или Region Proposal Network (RPN).

1. **Каждый регион подготовить:** Привести все регионы-кандидаты к единому размеру (например, с помощью **RoI Pooling** или **RoI Align**), чтобы их можно было подать на вход полносвязным слоям нейросети.
2. **Произвести классификацию:** Определить, к какому классу принадлежит объект в каждом регионе (например, "кошка", "собака", "фон").
3. **Регрессия bounding box:** Уточнить координаты ограничивающей рамки для каждого региона, чтобы она точнее обрамляла объект.
4. **Устранить дубликаты:** Убрать лишние, пересекающиеся рамки, которые указывают на один и тот же объект, с помощью алгоритма **Non-Maximum Suppression (NMS)**.

R-CNN

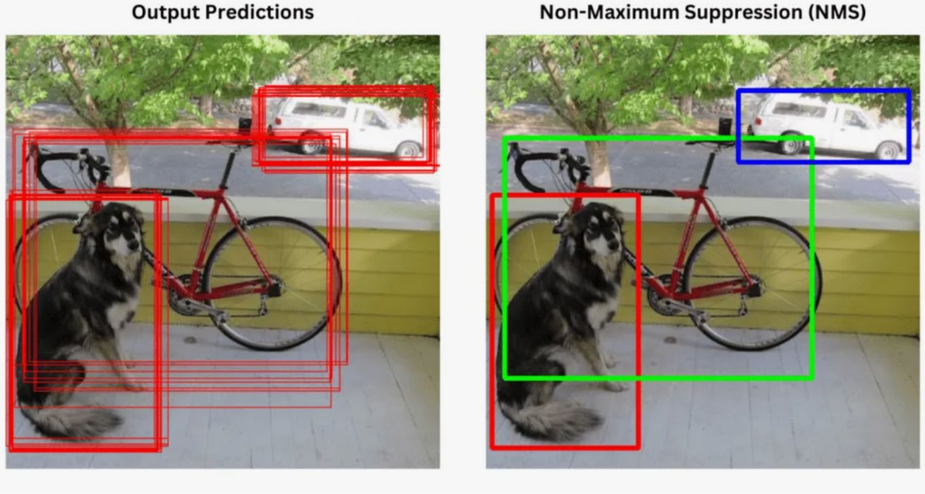


1. **Выбор 2000 регионов**

2. **Каждый регион подготовить:** 224×224
3. **Произвести классификацию:** Support Vector Machine
4. **Регрессия bounding box**
5. **Non-Maximum Suppression (NMS)**.

- Регионы могут дублировать друг друга, при этом каждый обрабатывается отдельно.
- Долгое обучение.
- Блок выделения регионов не подлежит обучению.

Non-Maximum Suppression




```python
def nms(boxes, scores, iou_thr=0.5):
    order = argsort(scores, descending=True)
    keep = []
    while len(order) > 0:
        i = order[0]                    # индекс лучшего бокса
        keep.append(i)
        ious = IoU(boxes[i], boxes[order[1:]])
        mask = ious < iou_thr
        order = order[1:][mask]         # оставляем только с малым IoU
    return keep
```

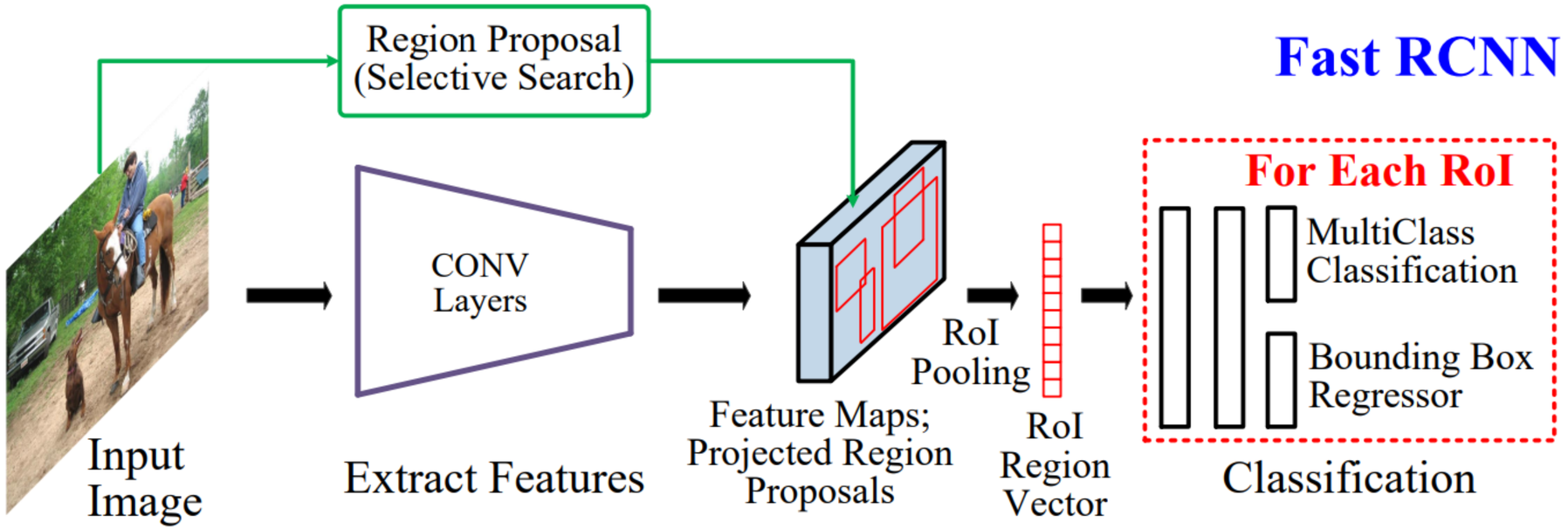
Вход: списки `boxes = [x1,y1,x2,y2]`, `scores`, порог `iou_thr`.

Выход: индексы оставшихся боксов.

Алгоритм:

1. Отсортировать боксы по `scores` по убыванию.
2. Пока список не пуст:
   - взять бокс `b*` с наибольшим score → добавить в результат;
   - удалить из списка все боксы с `IoU(b*, b) ≥ iou_thr`.
3. Вернуть сохранённые боксы.

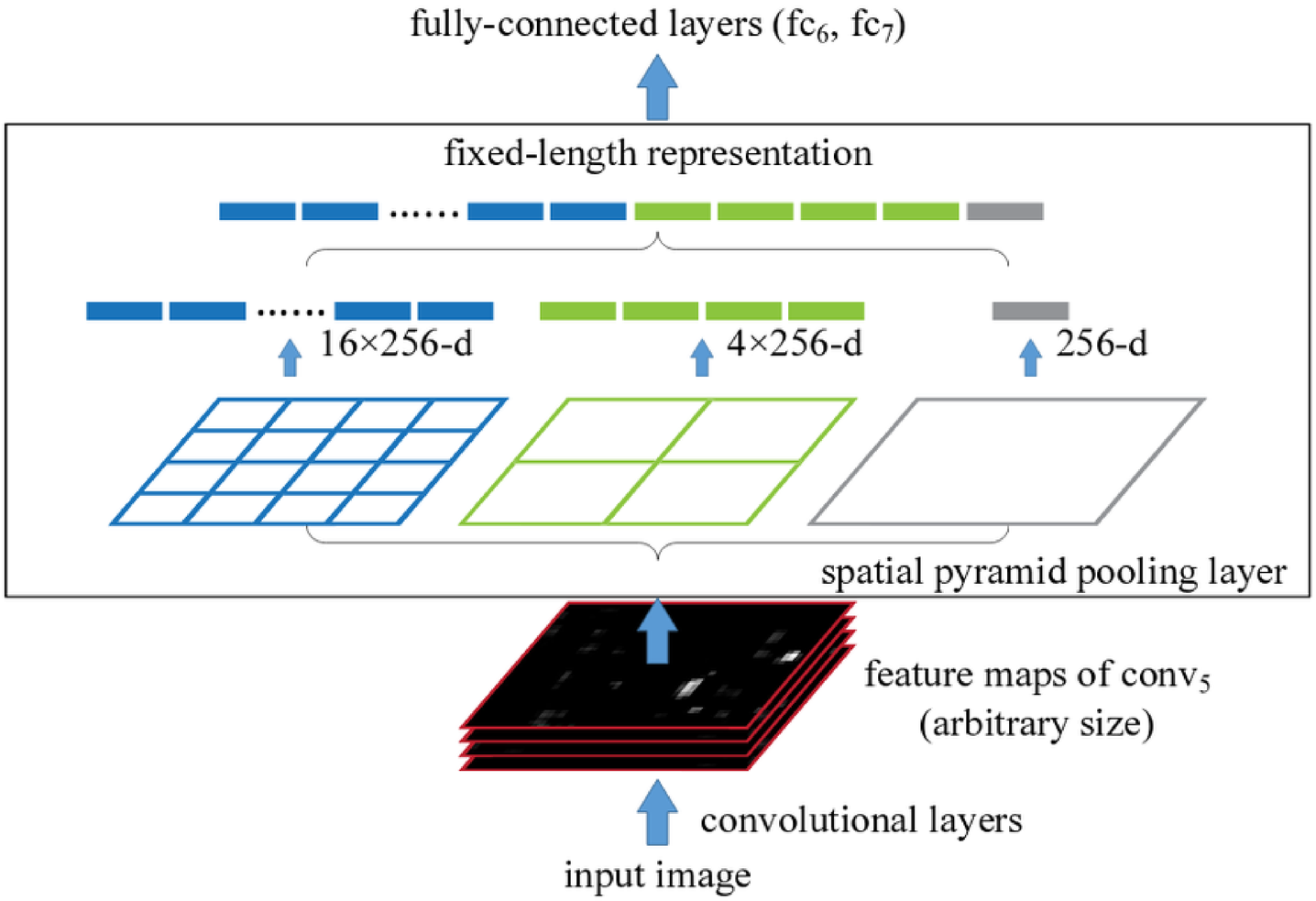
Fast R-CNN



Spatial Pyramid Pooling (SPP) – специальный слой пулинга, который позволяет обрабатывать изображения разного размера

https://arxiv.org/abs/1504.08083

Spatial Pyramid Pooling



Проблема:

Полносвязной НС нужен вход определенного размера

1.Применяем несколько уровней пулинга одновременно:

Уровень 1: пулинг до размера 4x4

Уровень 2: пулинг до размера 2x2

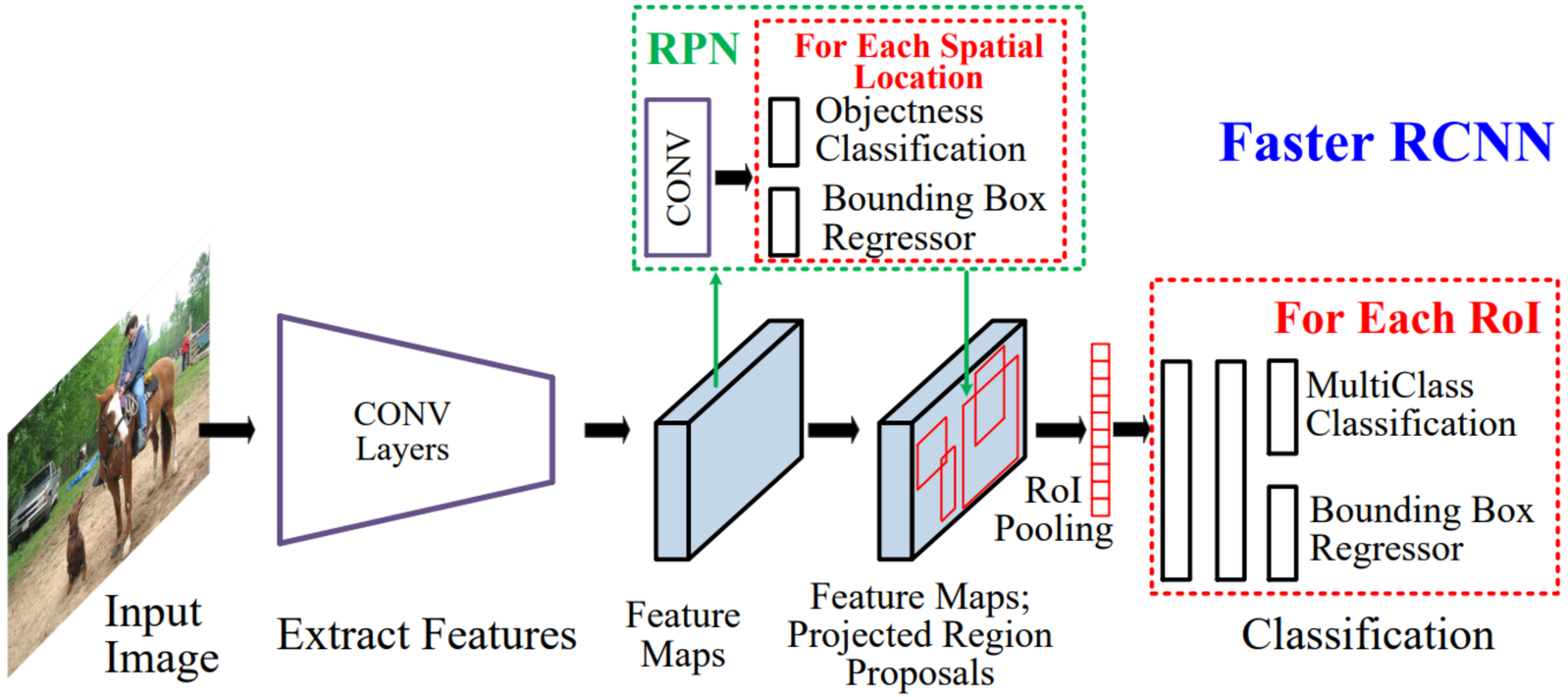
Уровень 3: пулинг до размера 1x1

Размеры могут варьироваться

2."Вытягиваем" и объединяем

в один большой вектор фиксированной длины

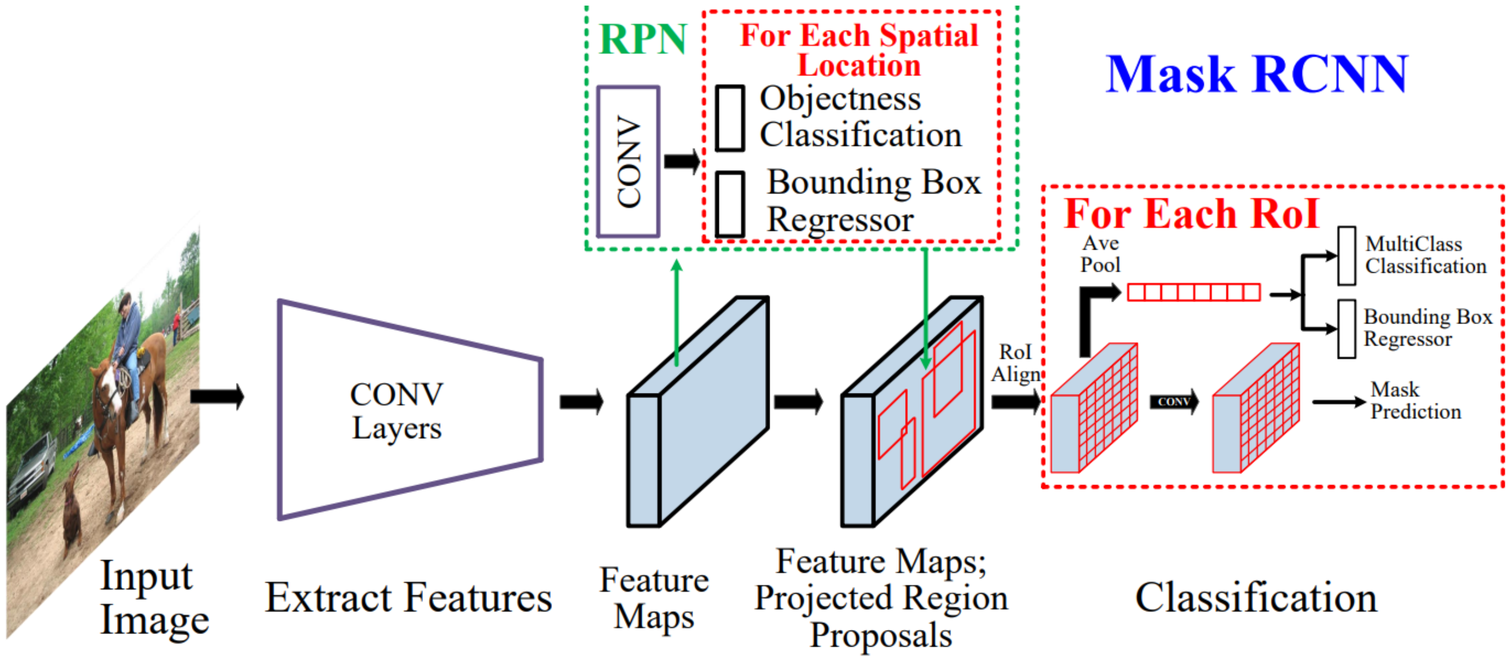
Faster R-CNN



Ключевое нововведение — **RPN (Region Proposal Network)**

https://arxiv.org/abs/1504.08083

Mask R-CNN



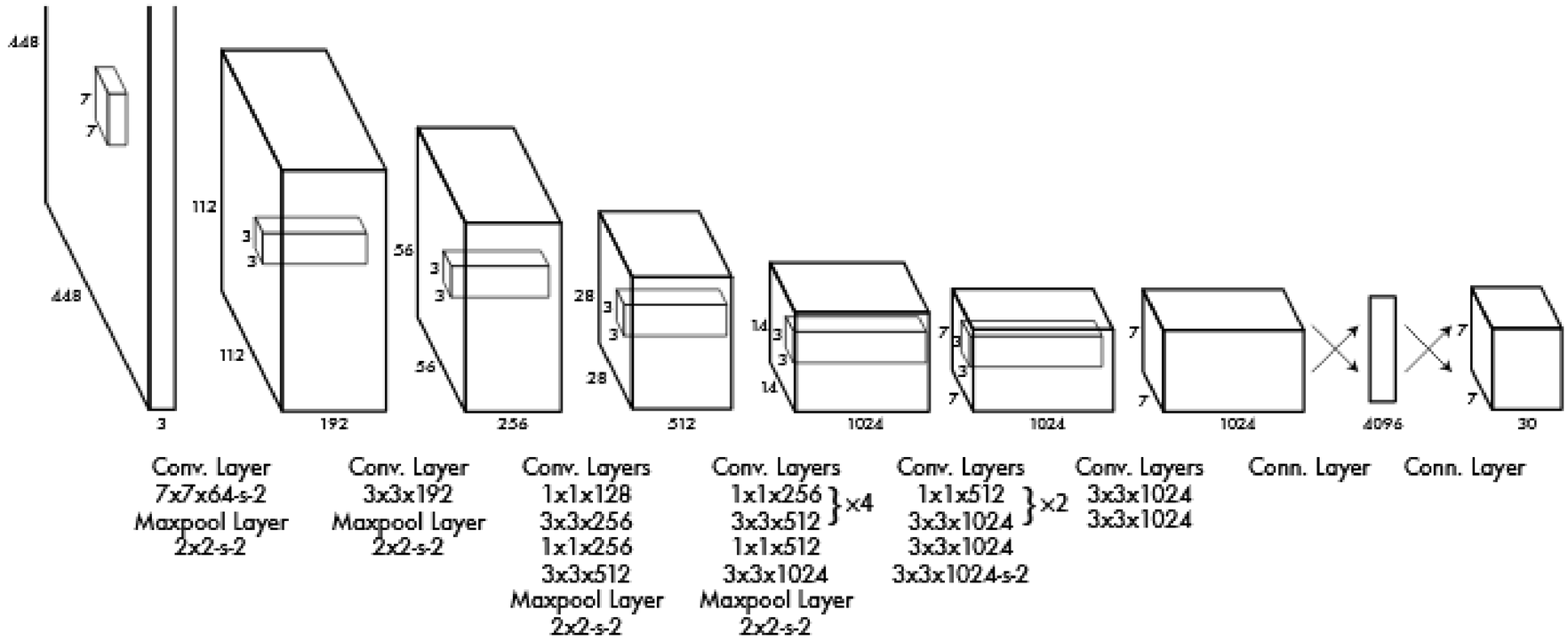
Ключевые нововведения: **RoI Align** (Region of Interest Align) и **добавление параллельной ветки для предсказания масок**.

https://arxiv.org/abs/1504.08083

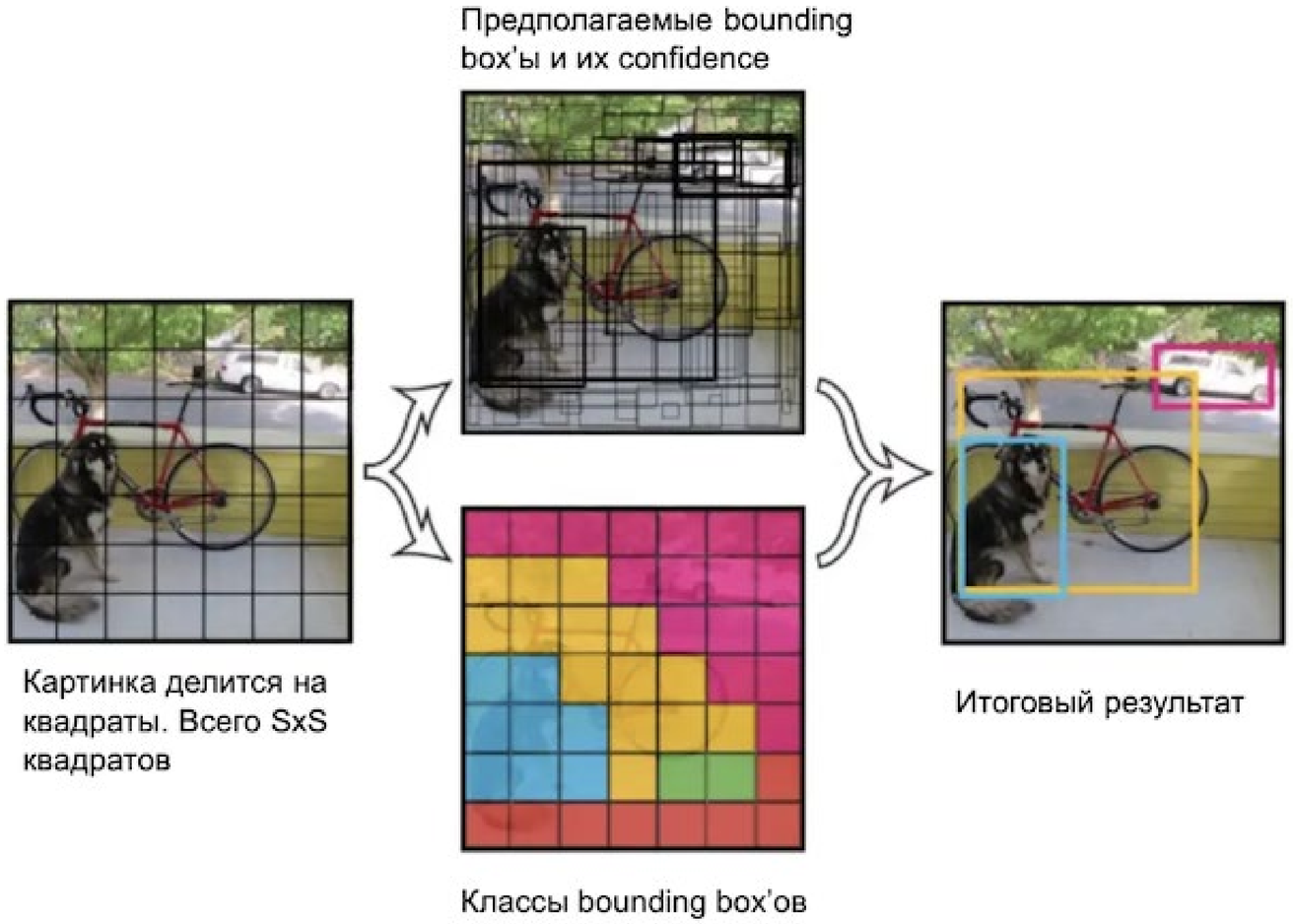
Одностадийные детекторы

YOLO (You Only Look Once, 2016)

Название означает «ты смотришь только один раз» — изображение проходит через нейросеть за один проход.

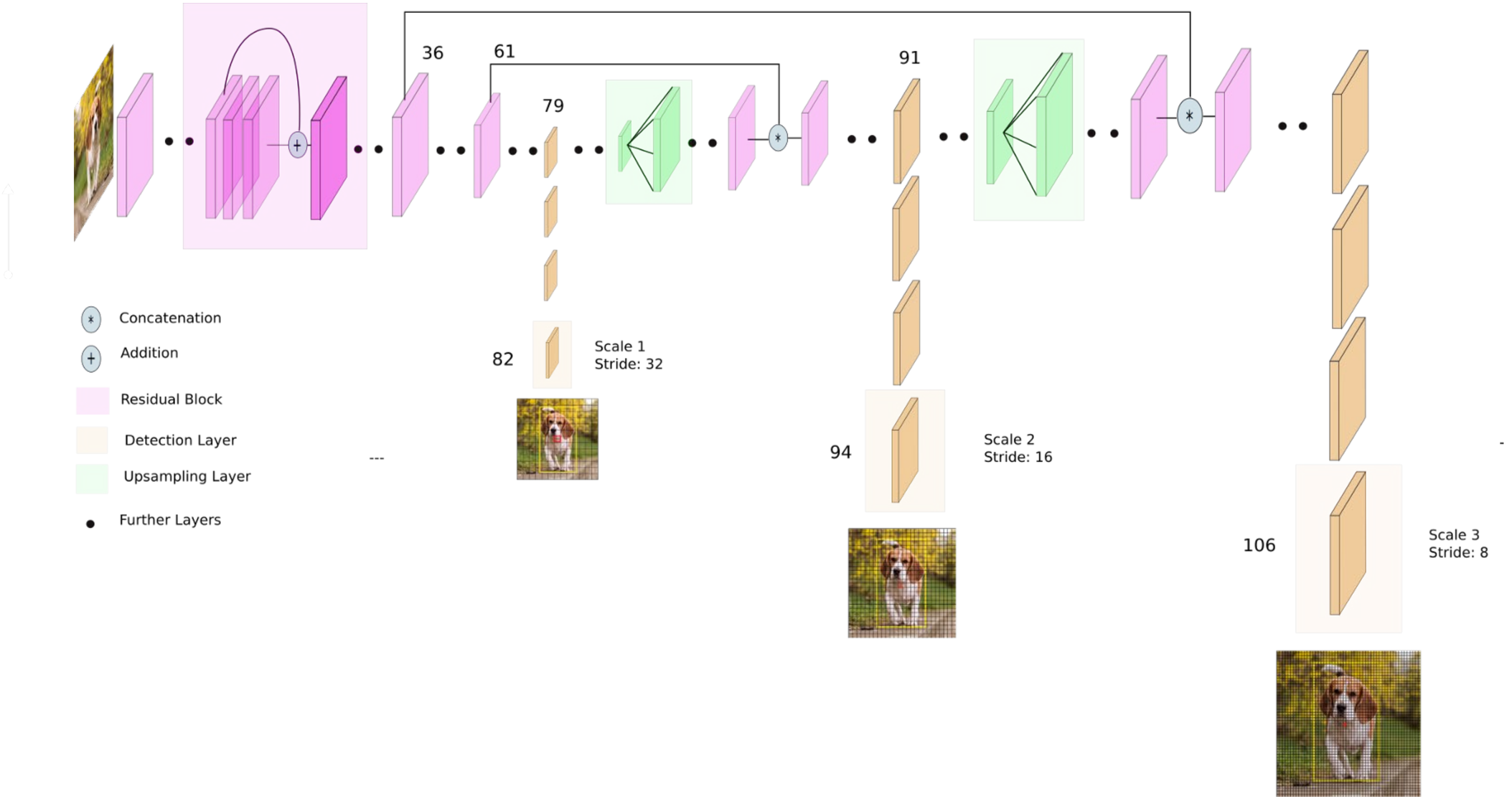


Yolo

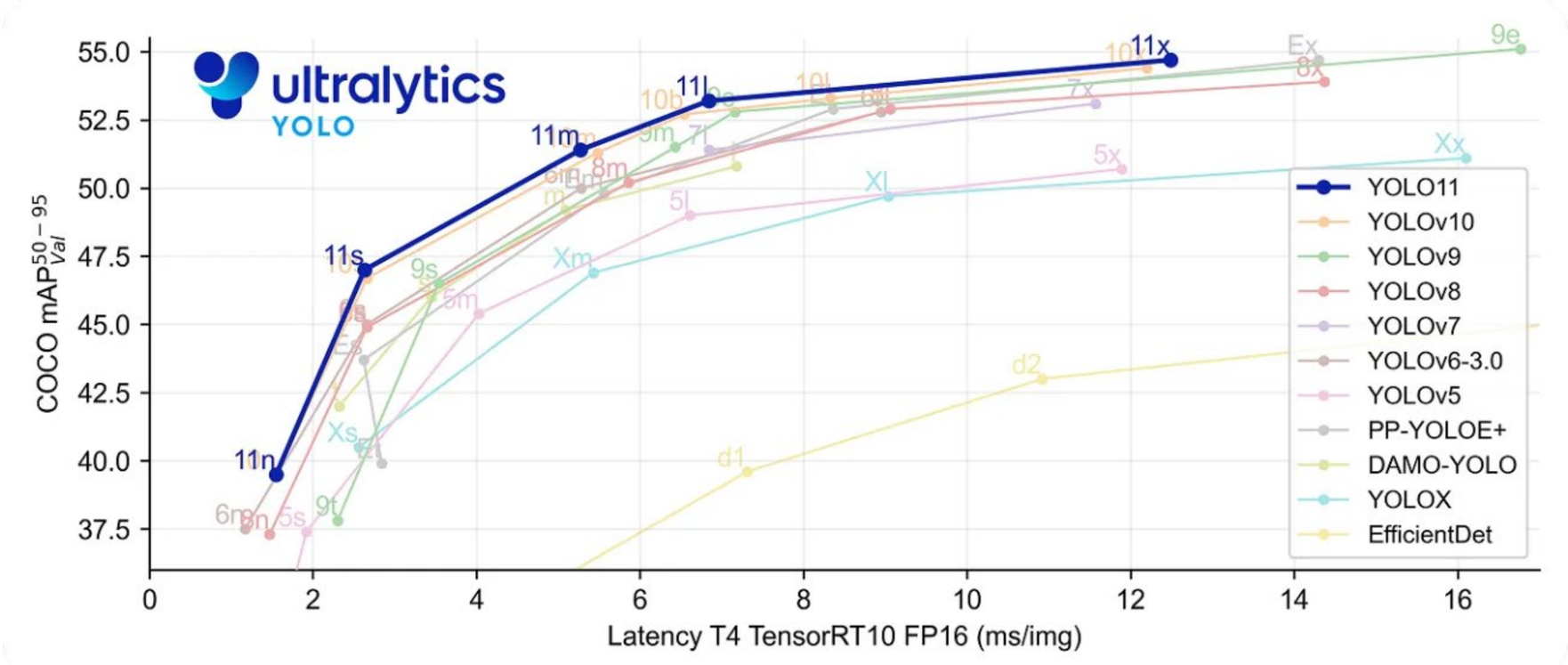


Каждая ячейка – свой класс, для каждого бокса своя вероятность что там есть объект.

Yolo v3

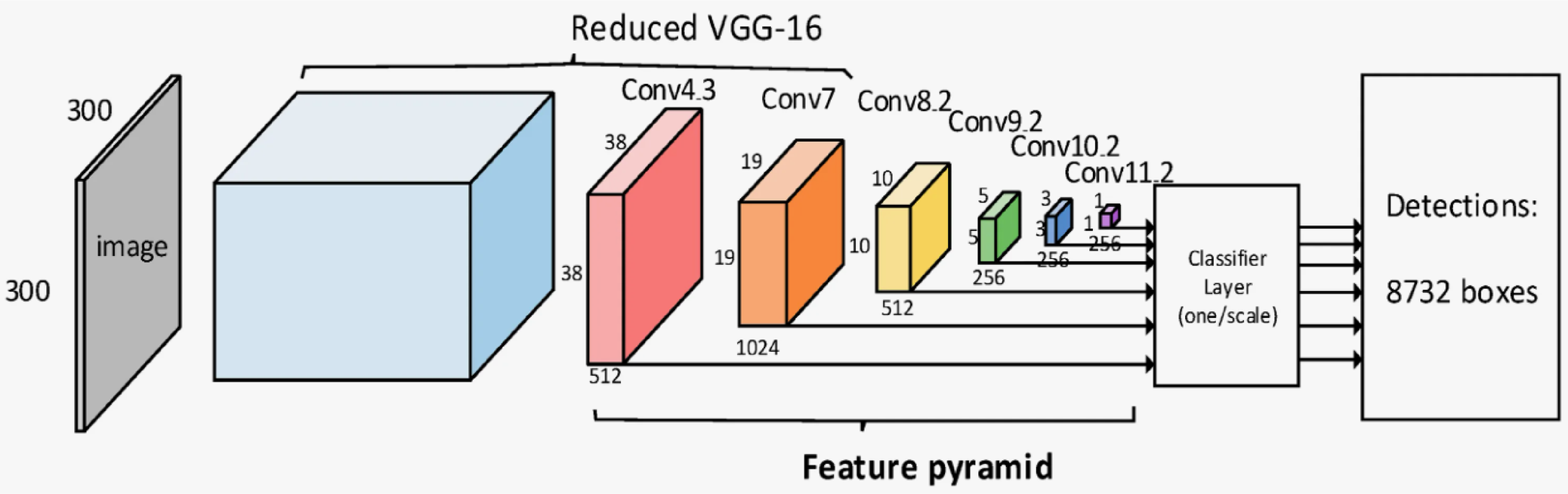

36
61
79
91
Concatenation
Addition
Residual Block
Detection Layer
Upsampling Layer
Further Layers
82
Scale 1
Stride: 32
94
Scale 2
Stride: 16
106
Scale 3
Stride: 8

Сравнение Yolo



https://www.ultralytics.com/ru/blog/comparing-ultralytics-yolo11-vs-previous-yolo-models

SSD (Single Shot Multibox Detector)



Single shot: это означает, что задачи локализации объектов и классификации выполняются в рамках одного прохода по сети

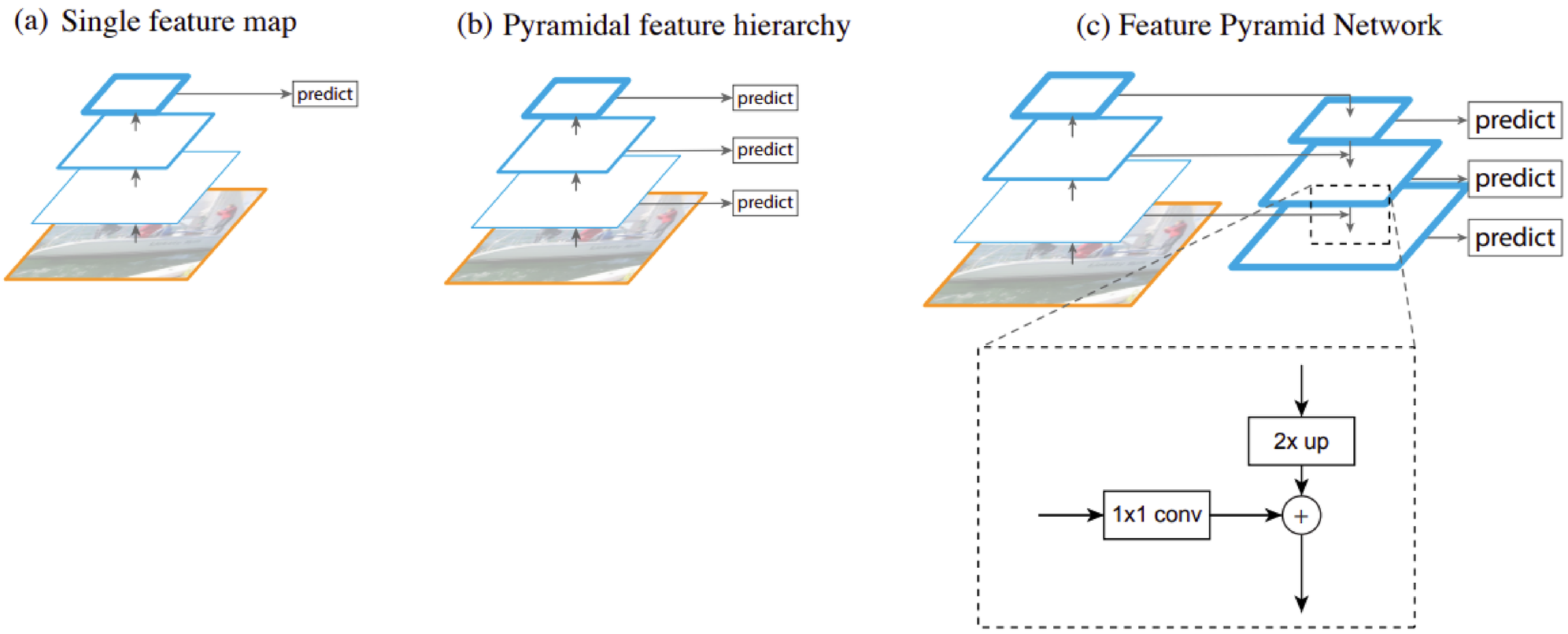
MultiBox: технология регрессии bounding boxes

Detector: сеть является детектором объектов и может классифицировать эти обнаруженные объекты

https://www.mdpi.com/2079-9292/12/7/1515/xml

Feature Pyramid Network

Архитектуры извлечения признаков в моделях YOLO (a), SSD (b) и в FPN (c)



https://deepmachinelearning.ru/docs/Neural-networks/Object-detection/Feature-Pyramid-Network

Современные тенденции

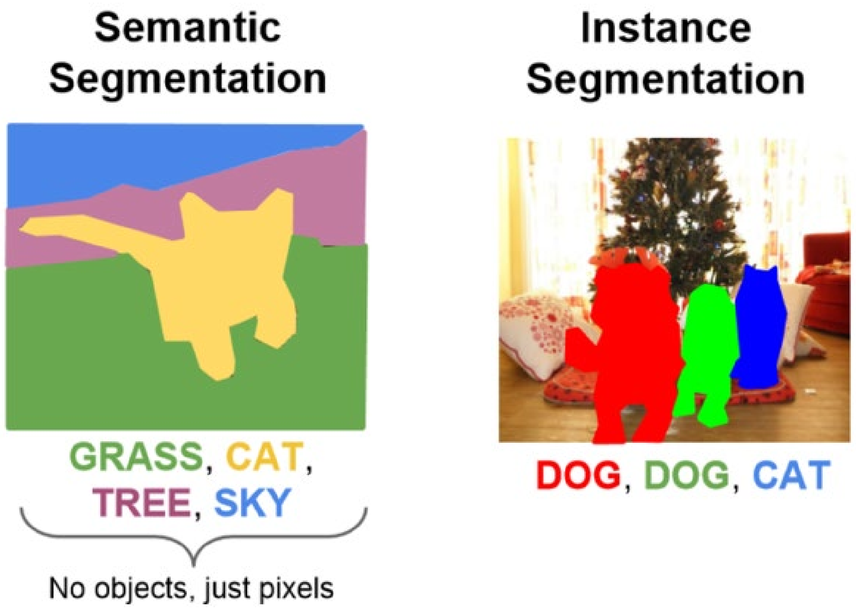
Современные архитектуры сочетают идеи:

- Используют **Feature Pyramid Networks (FPN)** для объединения признаков разных уровней.
- Применяют **attention-механизмы** и трансформеры (**DETR**, **Deformable DETR**) для отказа от эвристик вроде NMS и anchors.
- Переходят к **end-to-end обучению**, где вся система обучается на единой функции потерь.

Сегментация

Семантическая сегментация: Присвоение каждой точке изображения метки класса (например, "человек", "здание", "трава"). При этом разные объекты одного класса не различаются.

Инстанс-сегментация: Более сложная задача, которая не только классифицирует пиксели, но и разделяет отдельные экземпляры объектов within the same class (например, выделение каждого человека в толпе).



Архитектуры

Семантическая сегментация

FCN (Fully Convolutional Network) — первая CNN без полносвязных слоев, работающая с картами признаков произвольного размера.

U-Net — использует симметричную архитектуру «кодировщик-декодер» и skip-соединения, особенно успешна в медицине.

DeepLab — добавляет atrous convolutions (атриусные свертки) и Conditional Random Field для уточнения границ.

Инстанс-сегментация

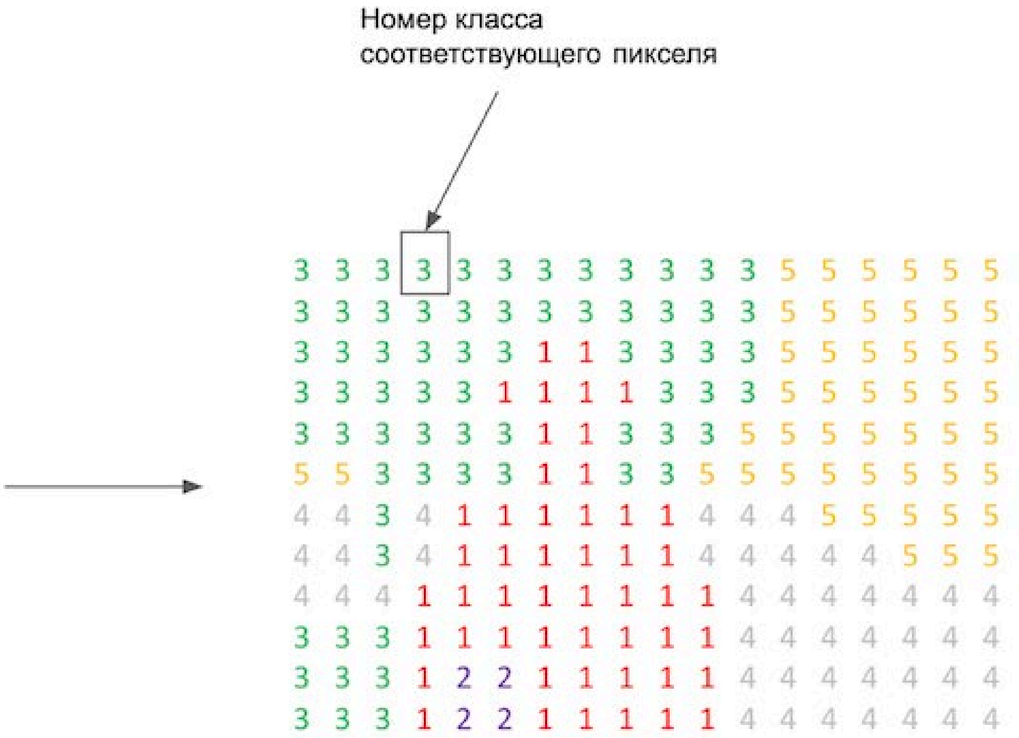
Объединяет детекцию и сегментацию.
Главный пример — **Mask R-CNN**

Постановка

- **0. Фон**
- 1. Люди
- 2. Сумки
- 3. Трава
- 4. Дорога
- 5. Здания

Вход

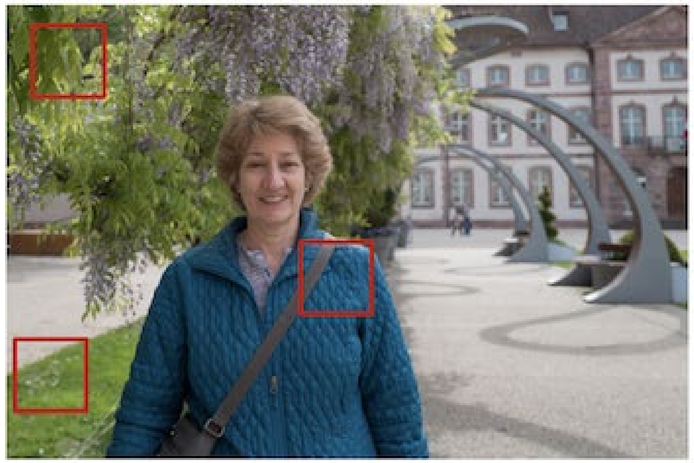
$h \times w \times 3$

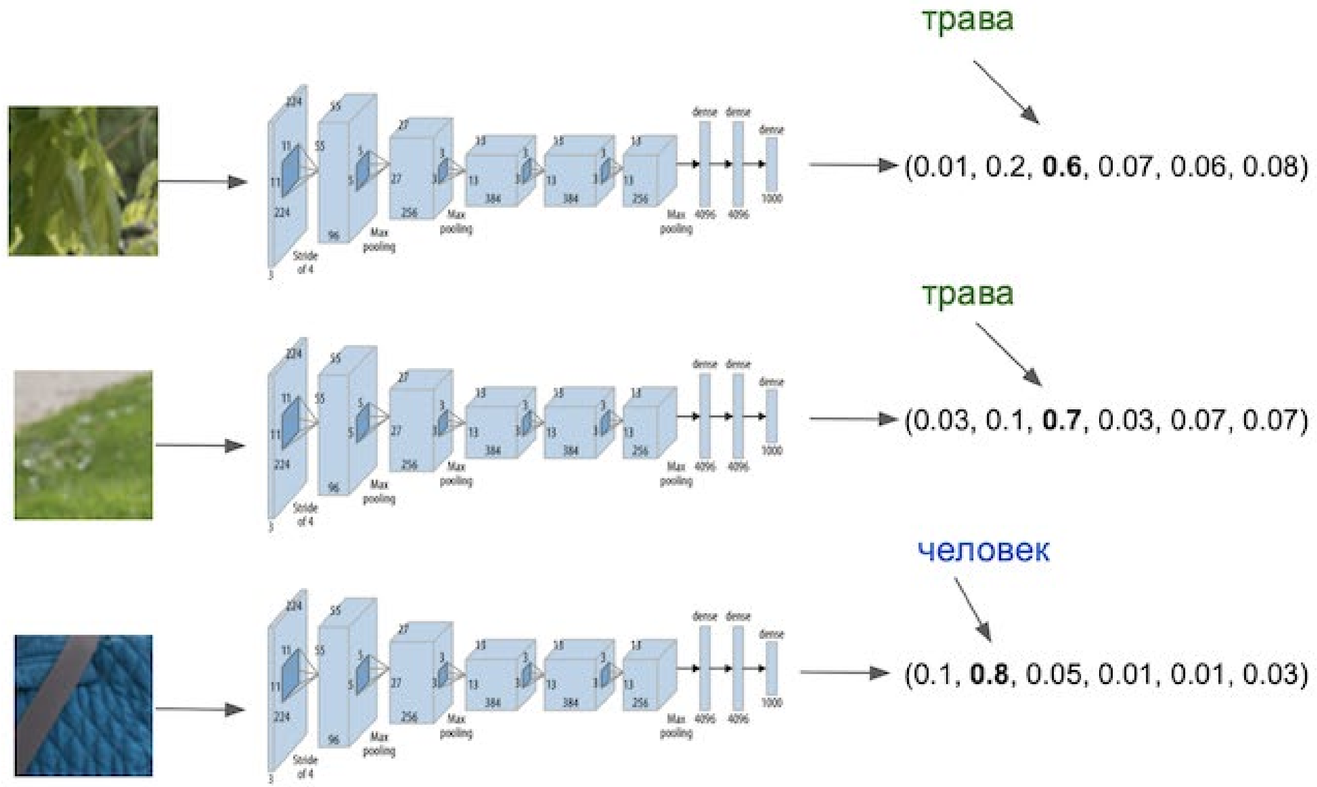


Выход

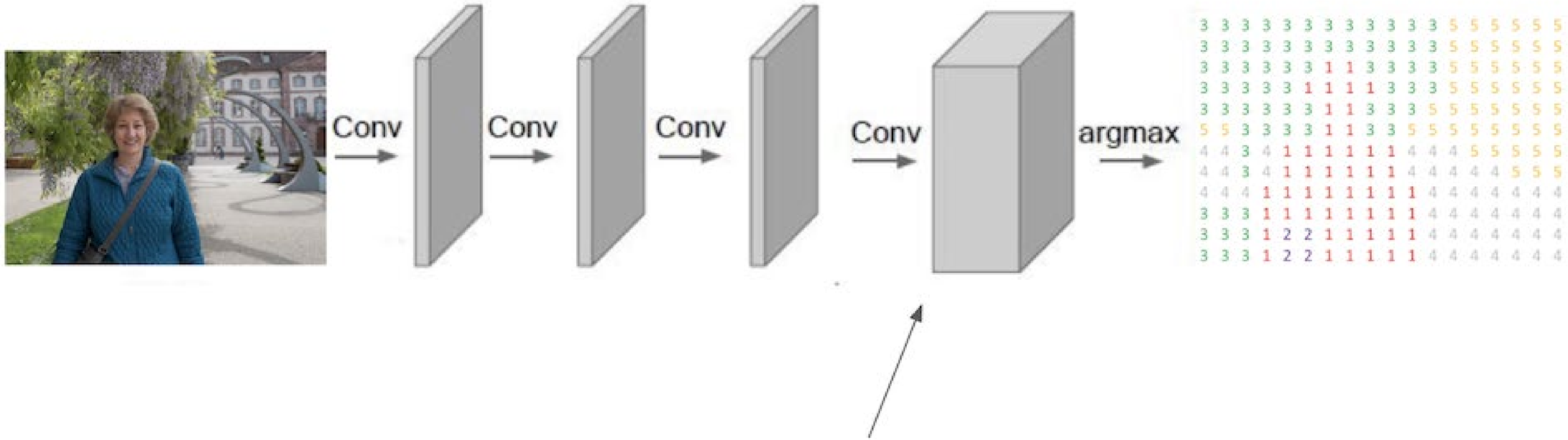
$h \times w$

Скользящее окно

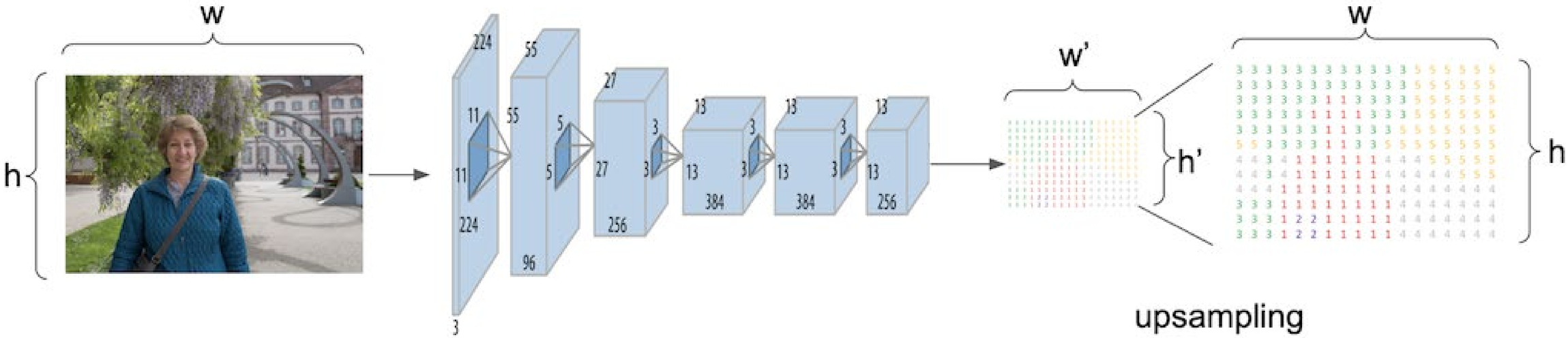




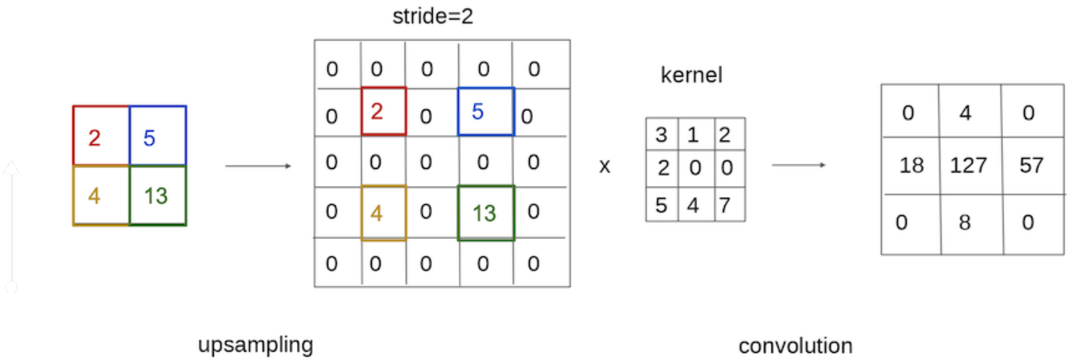
Fully Convolution Network

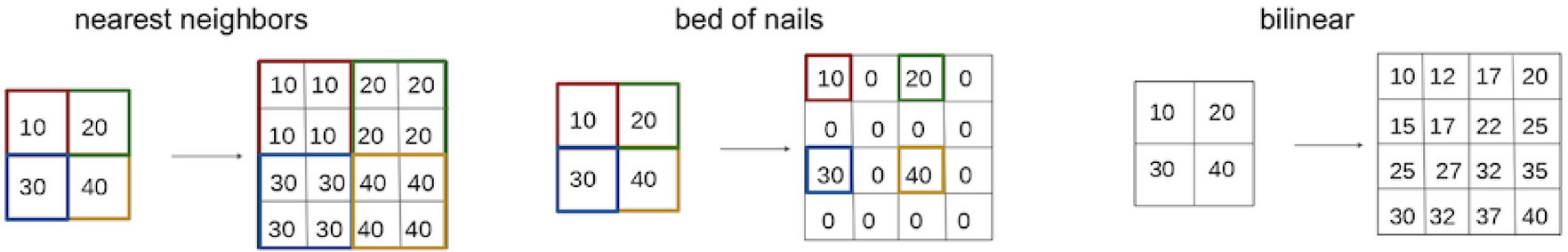


Upsampling

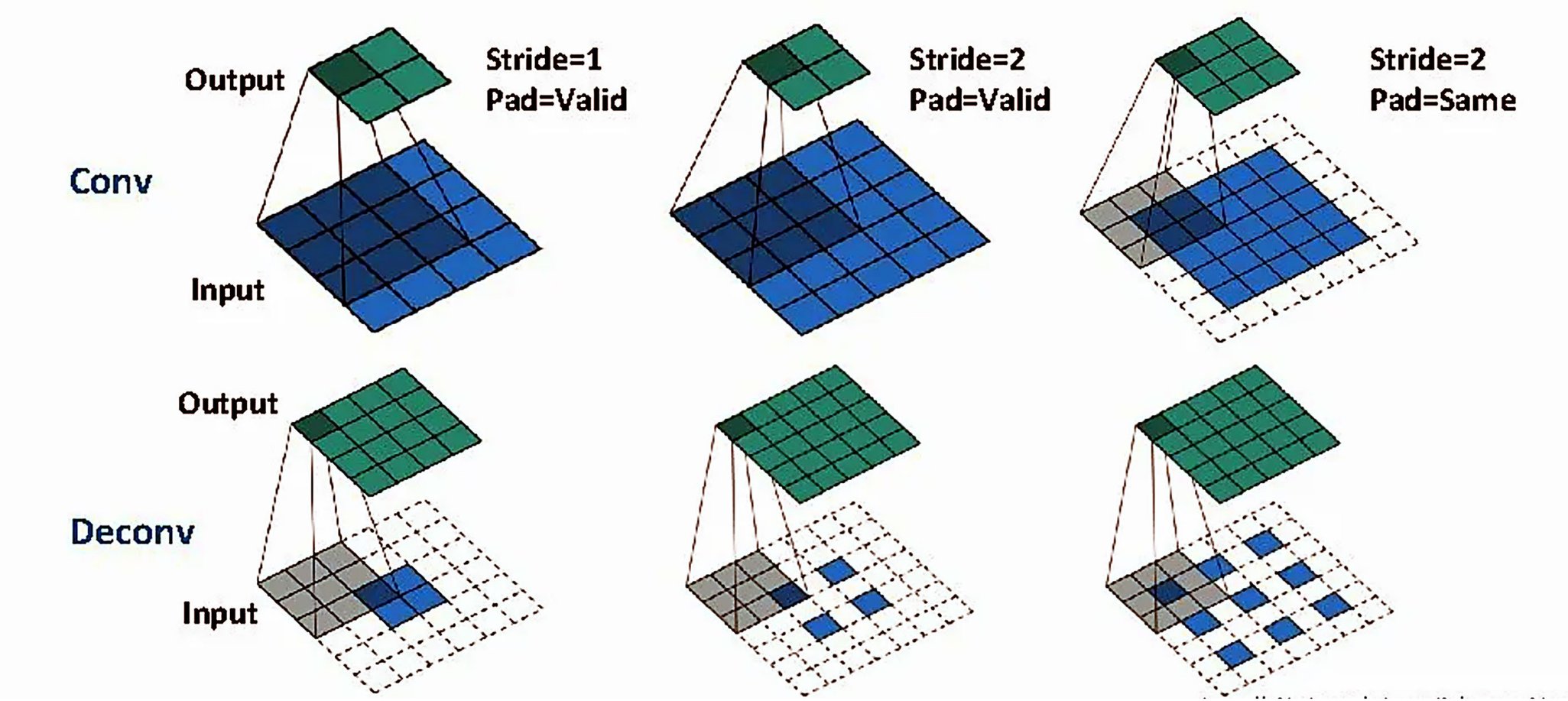

w
h
224
224
3
11
11
55
55
96
5
5
27
27
256
3
3
13
13
384
3
3
13
13
384
3
3
13
13
256
w'
h'
w
h
upsampling

Техники upsampling

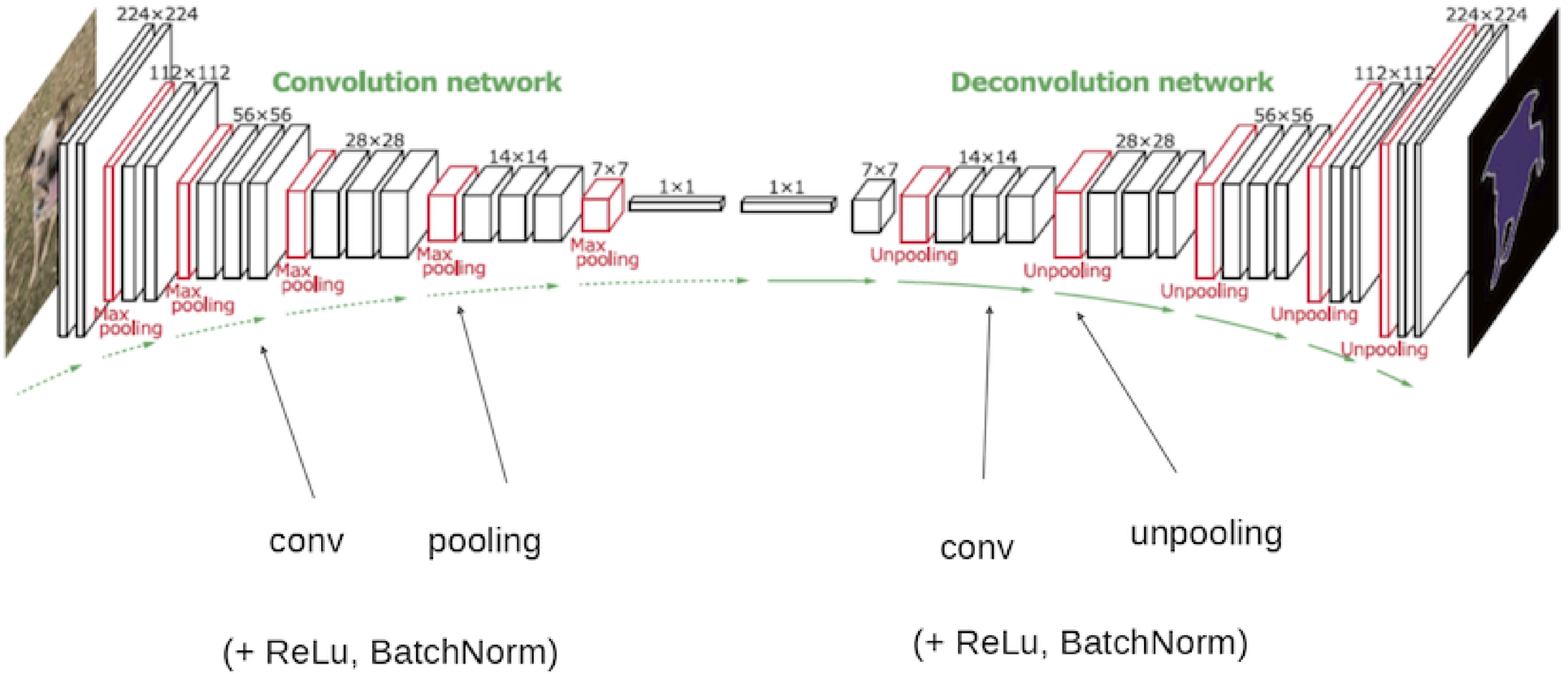




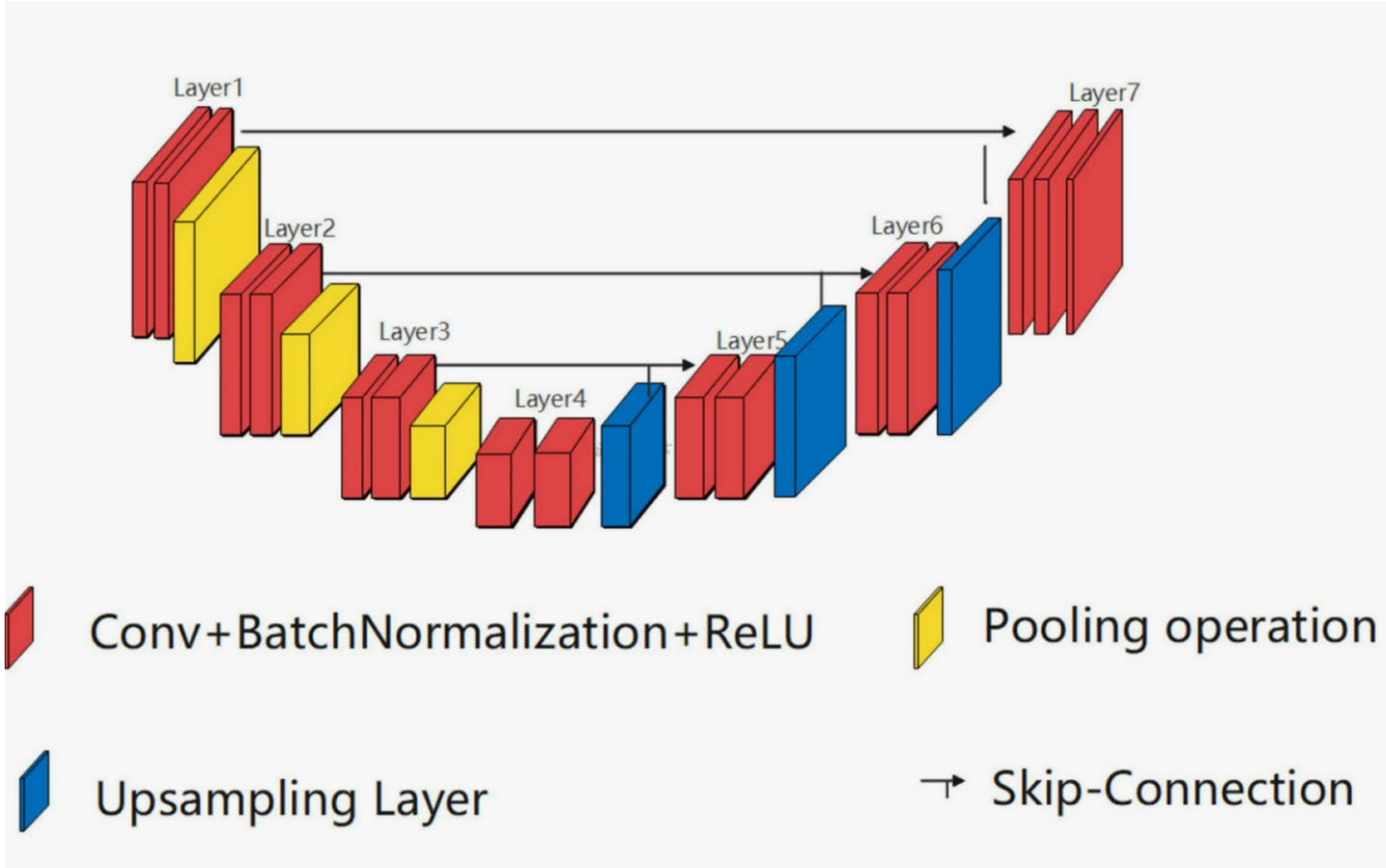
Transposed convolution



SegNet


224×224
112×112
56×56
28×28
14×14
7×7
1×1
1×1
7×7
14×14
28×28
56×56
112×112
224×224
Convolution network
Deconvolution network
Max pooling
Max pooling
Max pooling
Max pooling
Max pooling
Unpooling
Unpooling
Unpooling
Unpooling
Unpooling
conv
pooling
conv
unpooling
(+ ReLu, BatchNorm)
(+ ReLu, BatchNorm)

U-Net



https://arxiv.org/abs/1505.04597

DeepLab

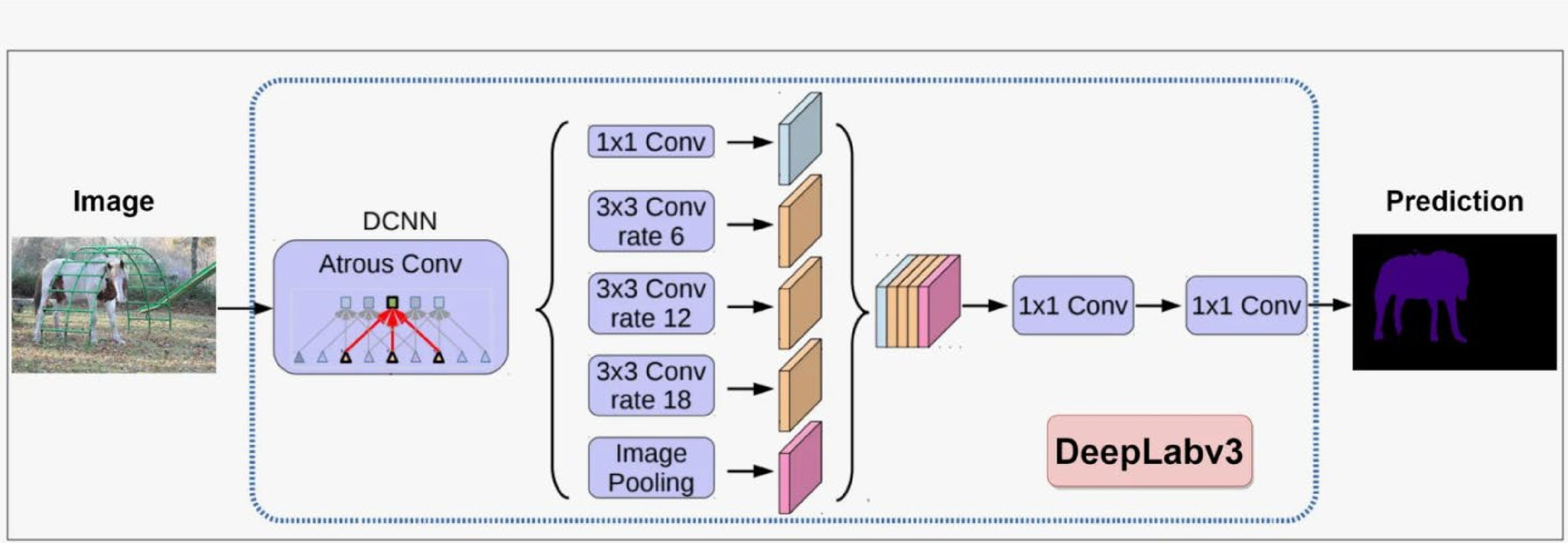


Image
DCNN
Atrous Conv
1x1 Conv
3x3 Conv rate 6
3x3 Conv rate 12
3x3 Conv rate 18
Image Pooling
1x1 Conv
1x1 Conv
Prediction
DeepLabv3

МИСИС
УНИВЕРСИТЕТ

Спасибо
за внимание!