

Задачи распознавания

Задачи CV

- задачи распознавания изображений
 - основные
 - классификация (classification)
 - локализация (localization)
 - детекция (object detection)
 - сегментация (segmentation)
 - дополнительные
 - ...
- задачи генерации изображений
 - ...

План

1. постановка задачи
2. метрики оценки качества
3. модели
4. функции потерь для DL моделей
5. подробнее про DL модели

Классификация

Задача классификации

Классификация: какому классу принадлежит объект на изображении?

cat

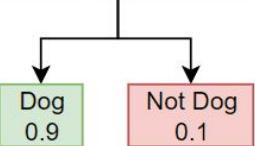


dog

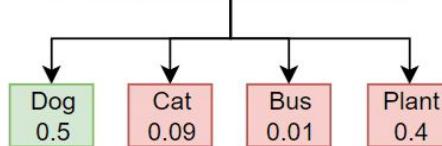


Виды классификации

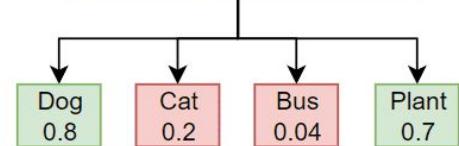
Binary Classification



Multiclass Classification



Multilabel Classification



source: <https://www.mathworks.com/help/deeplearning/ug/multilabel-image-classification-using-deep-learning.html>

Метрики

- Accuracy
- Balanced accuracy
- F1-score
- ROC-AUC, PR-AUC

Модели

- классические
- CNNs
- трансформеры

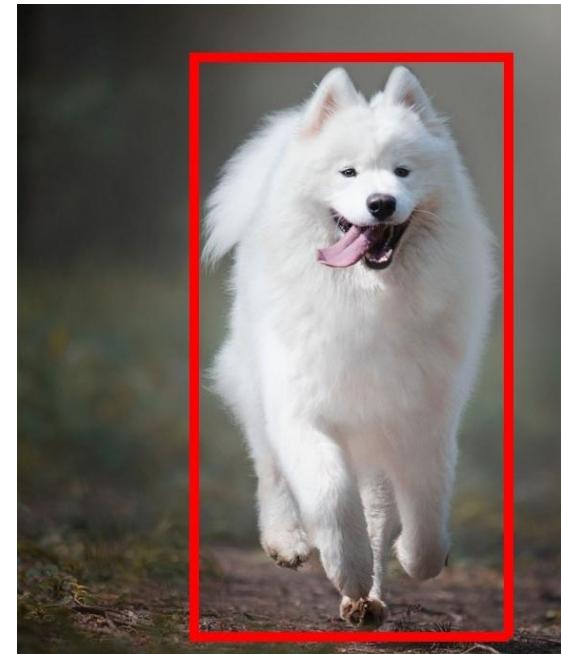
Функции потерь

- Кросс-энтропия
- Взвешенная кросс-энтропия

Детекция & локализация

Задача локализации и детекции

Локализация: выделить область картинки, содержащую объект

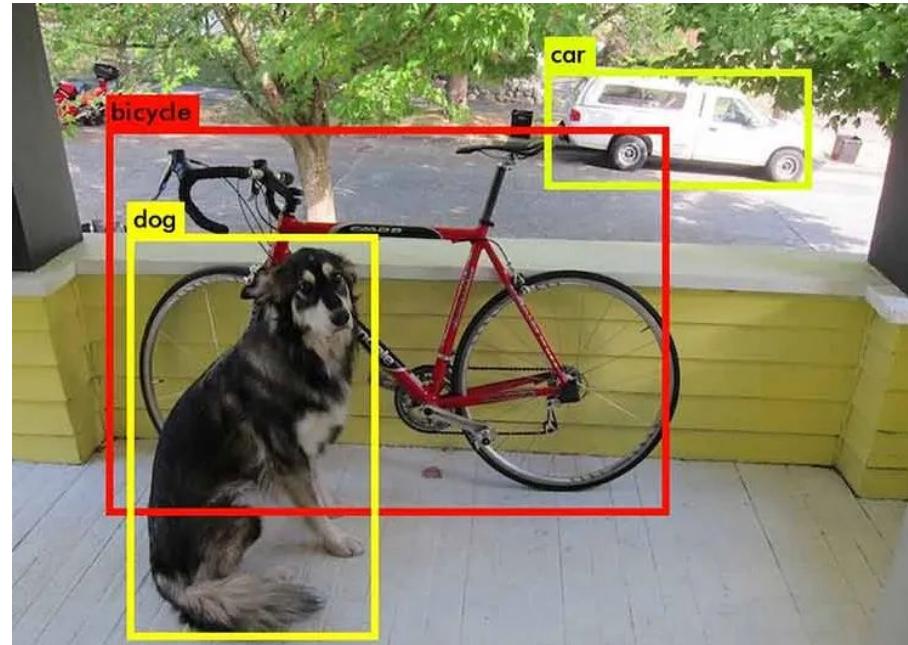


source: <https://www.madpaws.com.au/wp-content/uploads/2021/03/Dog-Breeds-Samovred.jpg>

Задача локализации и детекции

Детекция:

- найти все [важные] объекты на картинке
- определить класс каждого из них
- локализовать каждый из них



source:<https://towardsdatascience.com/deep-learning-method-for-object-detection-r-cnn-explained-ecdadd751d22>

Метрики оценки качества

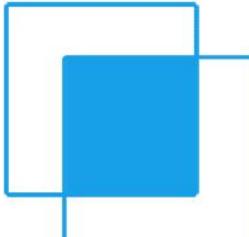
- mAP = mean Average Precision
- COCO mAP

Промежуточные шаги

- IoU = Intersection over Union
- AP = Average Precision

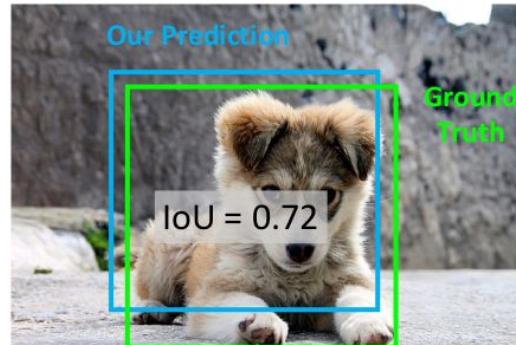
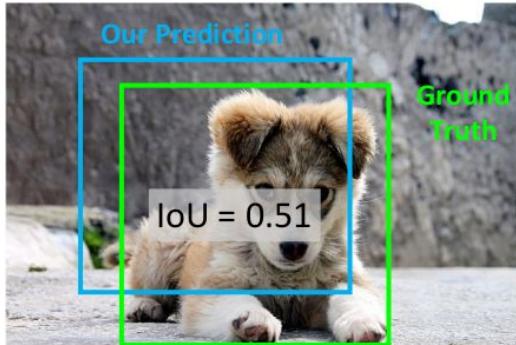
IoU

IoU = Intersection over Union

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


IoU

- $\text{IoU} > 0.5$ - “decent”
- $\text{IoU} > 0.7$ - “pretty good”
- $\text{IoU} > 0.9$ - “almost perfect”



AP

Дано:

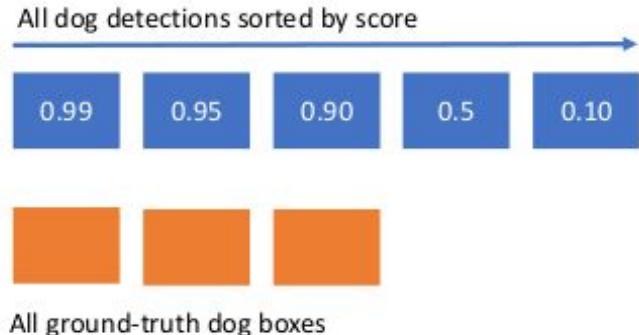
- зафиксируем рассматриваемый класс
- $G = [g_j]$ - список groundtruth боксов
- $P = [p_i]$ - список предсказанных боксов
- $c(p)$ - confidence score предсказанных боксов
- t - трешхолдинг порог

Опр. AP = Average Precision - площадь под PR кривой

Алгоритм построения PR кривой:

```
while len(P) > 0 and len(G) > 0:
```

1. достаем (с удалением) из P элемент p_i с наибольшим $c(p_i)$
1. $j = \operatorname{argmax}_h \text{IoU}(g_h, p_i)$
2. $\text{IoU}(g_j, p_i) > t$?
 - a. если да, то считаем предсказание верным и удаляем g_j из G
 - b. если нет, то считаем предсказание неверным
3. добавляем точку на PR кривую



AP

Дано:

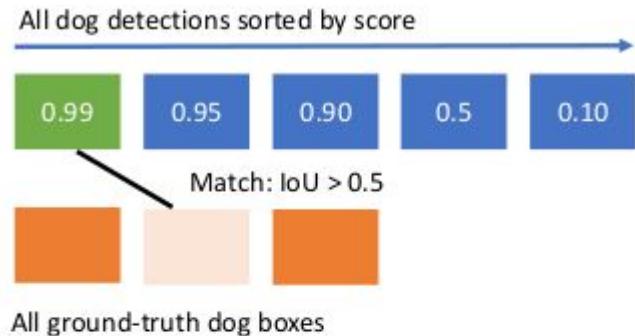
- зафиксируем рассматриваемый класс
- $G = [g_j]$ - список groundtruth боксов
- $P = [p_i]$ - список предсказанных боксов
- $c(p_i)$ - confidence score предсказанных боксов
- t - трешхолдинг порог

Опр. AP = Average Precision - площадь под PR кривой

Алгоритм построения PR кривой:

```
while len(P) > 0 and len(G) > 0:
```

1. достаем (с удалением) из P элемент p_i с наибольшим $c(p_i)$
1. $j = \operatorname{argmax}_h \text{IoU}(g_h, p_i)$
2. $\text{IoU}(g_j, p_i) > t$?
 - a. если да, то считаем предсказание верным и удаляем g_j из G
 - b. если нет, то считаем предсказание неверным
3. добавляем точку на PR кривую



AP

Дано:

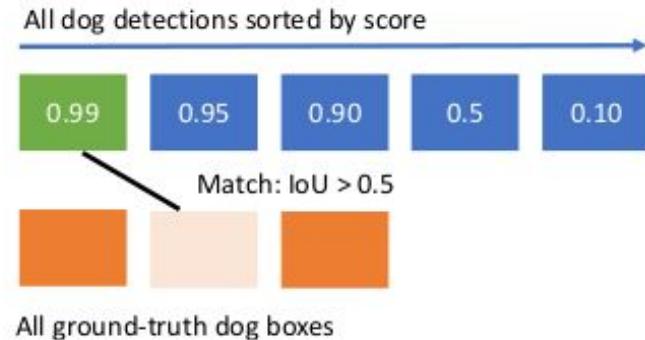
- зафиксируем рассматриваемый класс
- $G = [g_j]$ - список groundtruth боксов
- $P = [p_i]$ - список предсказанных боксов
- $c(p)$ - confidence score предсказанных боксов
- t - трешхолдинг порог

Опр. AP = Average Precision - площадь под PR кривой

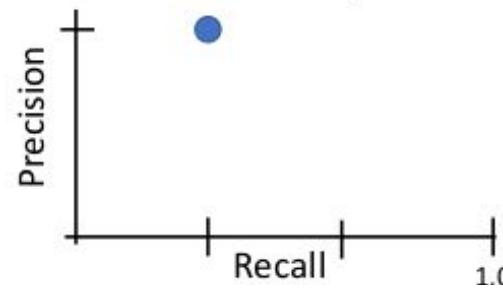
Алгоритм построения PR кривой:

```
while len(P) > 0 and len(G) > 0:
```

1. достаем (с удалением) из P элемент p_i с наибольшим $c(p_i)$
1. $j = \operatorname{argmax}_h \text{IoU}(g_h, p_i)$
2. $\text{IoU}(g_j, p_i) > t$?
 - a. если да, то считаем предсказание верным и удаляем g_j из G
 - b. если нет, то считаем предсказание неверным
3. добавляем точку на PR кривую



$$\begin{aligned} \text{Precision} &= 1/1 = 1.0 \\ \text{Recall} &= 1/3 = 0.33 \end{aligned}$$



AP

Дано:

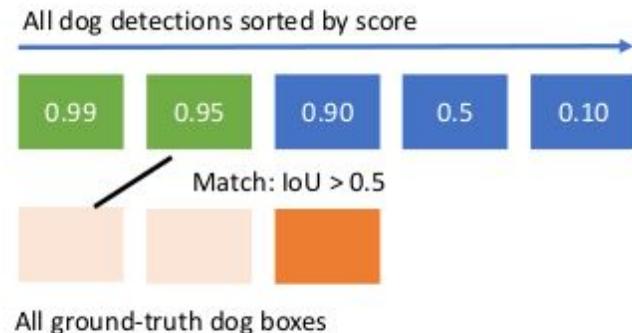
- зафиксируем рассматриваемый класс
- $G = [g_j]$ - список groundtruth боксов
- $P = [p_i]$ - список предсказанных боксов
- $c(p)$ - confidence score предсказанных боксов
- t - трешхолдинг порог

Опн. AP = Average Precision - площадь под PR кривой

Алгоритм построения PR кривой:

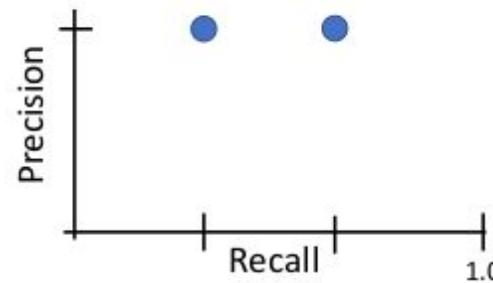
```
while len(P) > 0 and len(G) > 0:
```

1. достаем (с удалением) из P элемент p_i с наибольшим $c(p_i)$
1. $j = \operatorname{argmax}_h \text{IoU}(g_h, p_i)$
2. $\text{IoU}(g_j, p_i) > t$?
 - a. если да, то считаем предсказание верным и удаляем g_j из G
 - b. если нет, то считаем предсказание неверным
3. добавляем точку на PR кривую



$$\text{Precision} = 2/2 = 1.0$$

$$\text{Recall} = 2/3 = 0.67$$



AP

Дано:

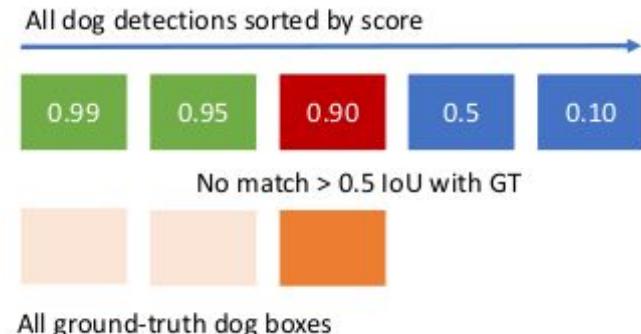
- зафиксируем рассматриваемый класс
- $G = [g_i]$ - список groundtruth боксов
- $P = [p_j]$ - список предсказанных боксов
- $c(p)$ - confidence score предсказанных боксов
- t - трешхолдинг порог

Опр. AP = Average Precision - площадь под PR кривой

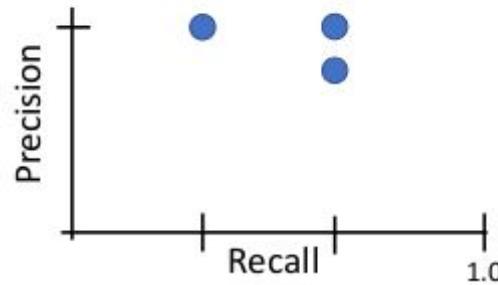
Алгоритм построения PR кривой:

```
while len(P) > 0 and len(G) > 0:
```

1. достаем (с удалением) из P элемент p_i с наибольшим $c(p_i)$
1. $j = \operatorname{argmax}_h \text{IoU}(g_h, p_i)$
2. $\text{IoU}(g_j, p_i) > t$?
 - a. если да, то считаем предсказание верным и удаляем g_j из G
 - b. если нет, то считаем предсказание неверным
3. добавляем точку на PR кривую



$$\text{Precision} = 2/3 = 0.67$$
$$\text{Recall} = 2/3 = 0.67$$



AP

Дано:

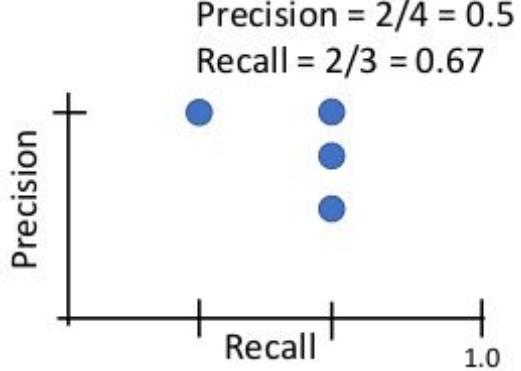
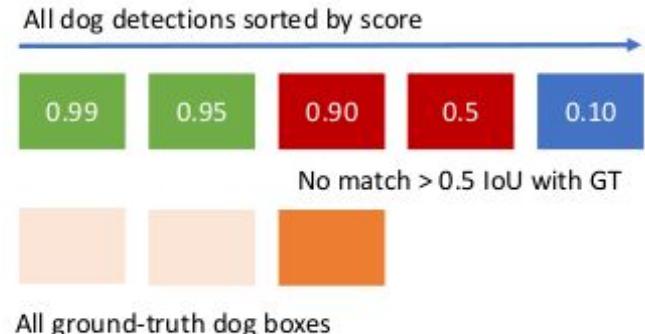
- зафиксируем рассматриваемый класс
- $G = [g_j]$ - список groundtruth боксов
- $P = [p_i]$ - список предсказанных боксов
- $c(p)$ - confidence score предсказанных боксов
- t - трешхолдинг порог

Опр. AP = Average Precision - площадь под PR кривой

Алгоритм построения PR кривой:

```
while len(P) > 0 and len(G) > 0:
```

1. достаем (с удалением) из P элемент p_i с наибольшим $c(p_i)$
1. $j = \operatorname{argmax}_h \text{IoU}(g_h, p_i)$
2. $\text{IoU}(g_j, p_i) > t$?
 - a. если да, то считаем предсказание верным и удаляем g_j из G
 - b. если нет, то считаем предсказание неверным
3. добавляем точку на PR кривую



AP

Дано:

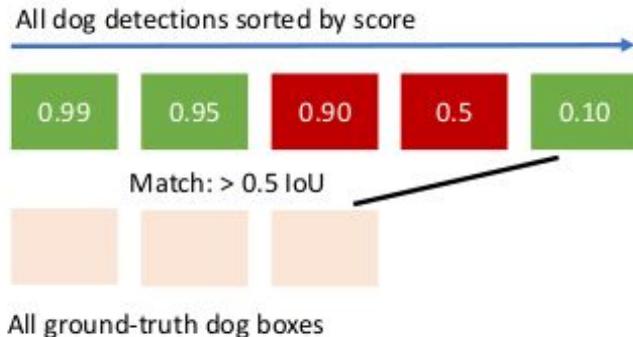
- зафиксируем рассматриваемый класс
- $G = [g_j]$ - список groundtruth боксов
- $P = [p_i]$ - список предсказанных боксов
- $c(p)$ - confidence score предсказанных боксов
- t - трешхолдинг порог

Опр. AP = Average Precision - площадь под PR кривой

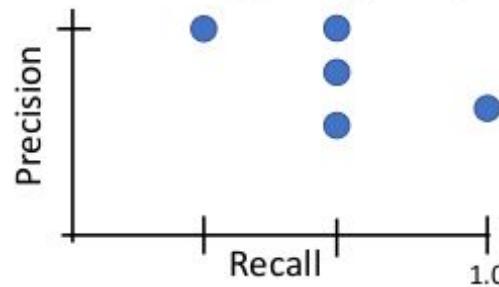
Алгоритм построения PR кривой:

```
while len(P) > 0 and len(G) > 0:
```

1. достаем (с удалением) из P элемент p_i с наибольшим $c(p_i)$
1. $j = \operatorname{argmax}_h \text{IoU}(g_h, p_i)$
2. $\text{IoU}(g_j, p_i) > t$?
 - a. если да, то считаем предсказание верным и удаляем g_j из G
 - b. если нет, то считаем предсказание неверным
3. добавляем точку на PR кривую



$$\begin{aligned} \text{Precision} &= 3/5 = 0.6 \\ \text{Recall} &= 3/3 = 1.0 \end{aligned}$$



AP

Дано:

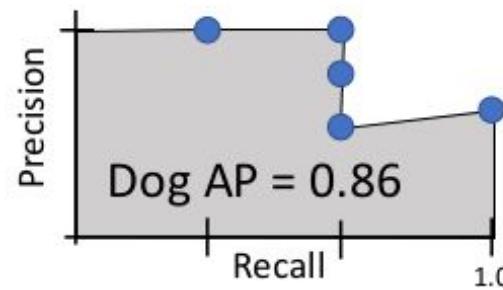
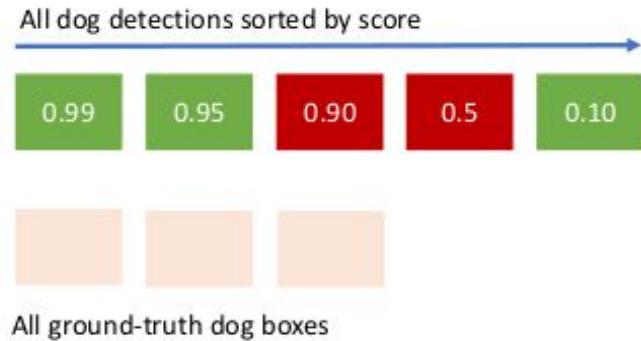
- зафиксируем рассматриваемый класс
- $G = [g_j]$ - список groundtruth боксов
- $P = [p_i]$ - список предсказанных боксов
- $c(p)$ - confidence score предсказанных боксов
- t - трешхолдинг порог

Опр. AP = Average Precision - площадь под PR кривой

Алгоритм построения PR кривой:

```
while len(P) > 0 and len(G) > 0:
```

1. достаем (с удалением) из P элемент p_i с наибольшим $c(p_i)$
1. $j = \operatorname{argmax}_h \text{IoU}(g_h, p_i)$
2. $\text{IoU}(g_j, p_i) > t$?
 - a. если да, то считаем предсказание верным и удаляем g_j из G
 - b. если нет, то считаем предсказание неверным
3. добавляем точку на PR кривую



mAP

mAP = mean Average Precision:
среднее значений AP для каждого класса

mAP

какое t выбрать:

- обычно 0.5
- COCO mAP:
 - посчитать mAP@t c t = 0.5, 0.55, 0.6, ..., 0.95
 - взять среднее

Модели

- классические (на основе признаков)
 - Viola-Jones
 - SIFT
 - ...
- CNNs
- трансформеры

Функции потерь

- Классификация
 - Кросс-энтропия
- Локализация (регрессия ббокса)
 - MSE

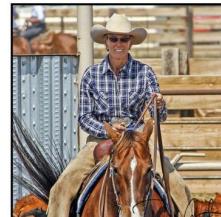
CNNs

- [2013] R-CNN
- [2015] Fast R-CNN
- [2015] Faster R-CNN

R-CNN

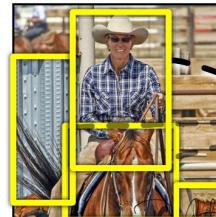
R-CNN = Region-based CNN

- Rols with selective search
- warp each RoI to 224x224
- pass through a CNN
- pass through heads
 - classification
 - bbox regression



1. Input image

R-CNN: *Regions with CNN features*



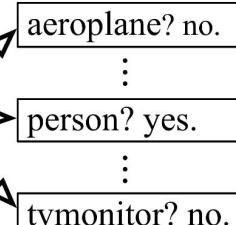
2. Extract region proposals (~2k)

warped region



CNN

3. Compute CNN features

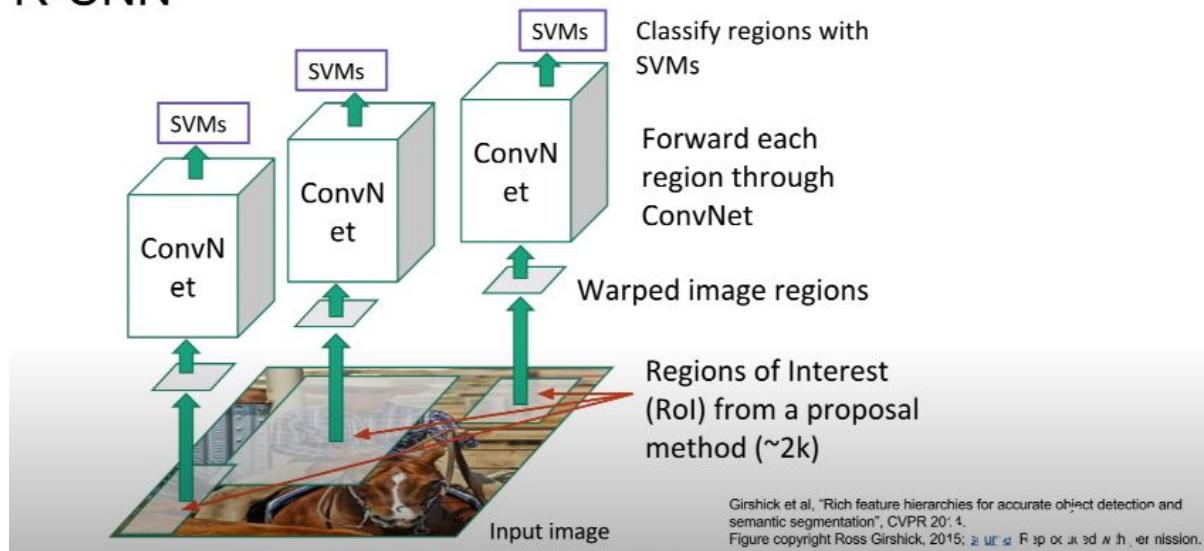


4. Classify regions

Fast R-CNN

Ускоряем обучение и инференс x10:
заменяем warp → CNN на CNN → warp

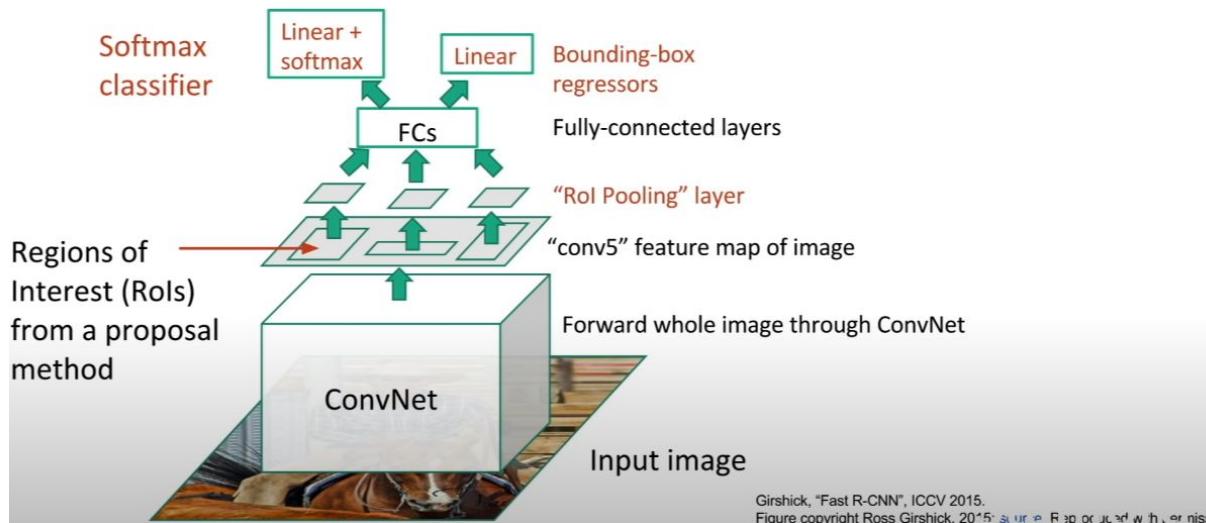
R-CNN



Fast R-CNN

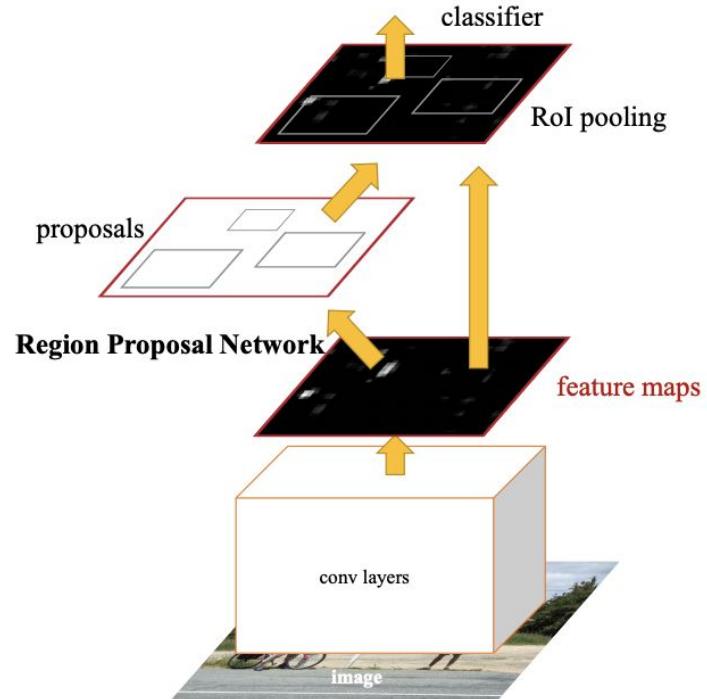
Ускоряем обучение и инференс x10:
заменим warp → CNN на CNN → warp

Fast R-CNN



Faster R-CNN

Ускоряем обучение и инференс еще x10:
заменяем selective search нейронкой
RPN (Region Proposal Network)



Two-Stage vs Single-Stage

- two-stage:
 - формируем пропозалы
 - каждый распознаем: класс + уточненный ббокс
- signle-stage:
 - сразу распознаем

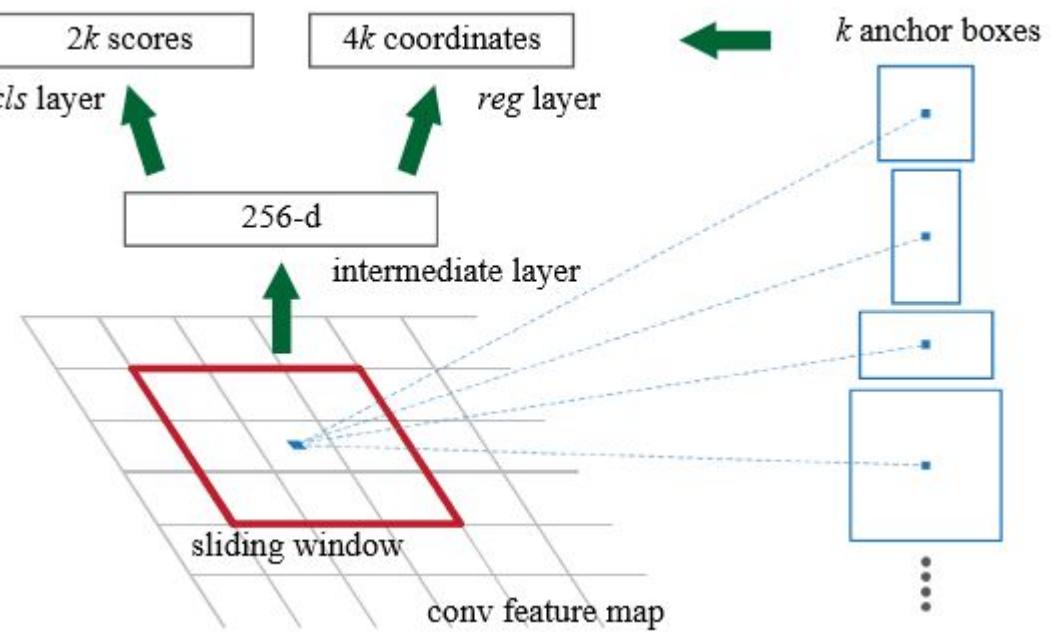
Anchor-based methods

- [2015] SSD
- [2015] YOLO

Anchor-based methods

Якоря (anchors) - заранее определенные образцы боксов, которые ищем

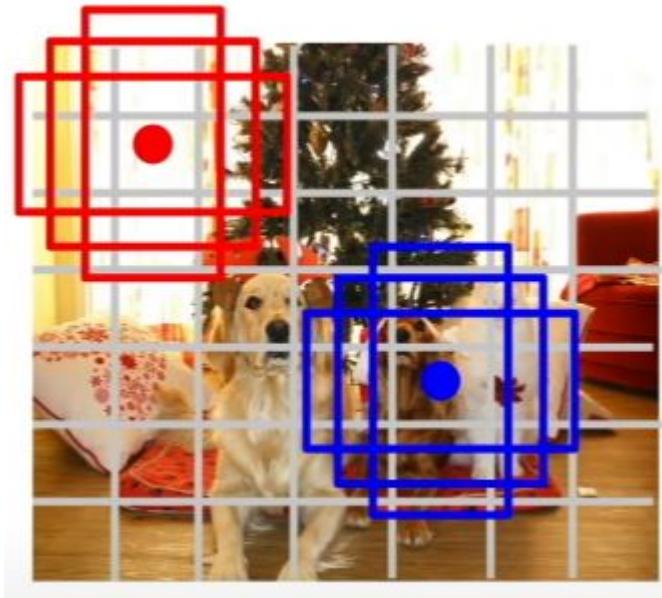
- размер
- соотношение сторон



Anchor-based methods

Пример YOLO:

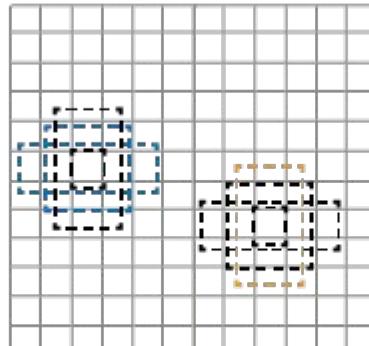
- разбиваем изображение на 7×7 грид
- пропускаем целиком через энкодер
- для образа каждого из блоков грида
для каждого из K якорей предсказываем
 - вероятности каждого из C классов
 - регрессию бокса на якорь



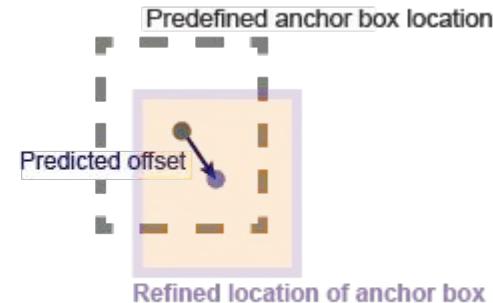
Anchor-based methods



Ground truth image and
bounding boxes



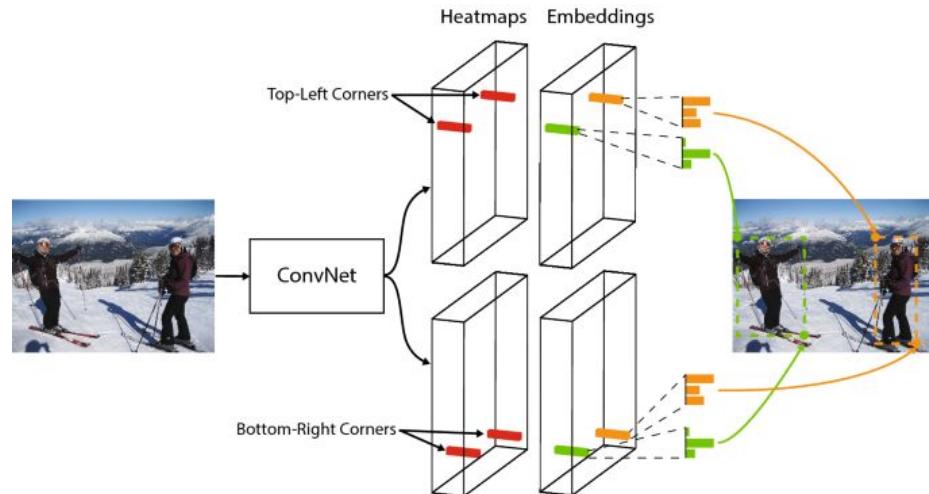
Anchor boxes at each predefined
location in each feature map



Key-point based methods

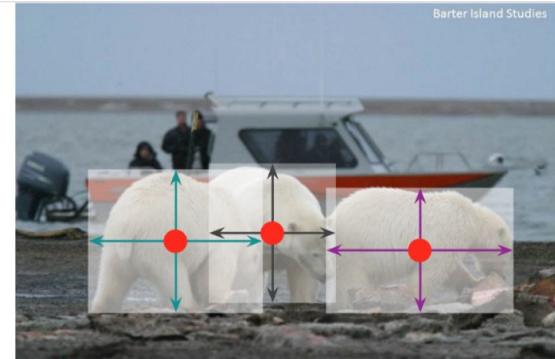
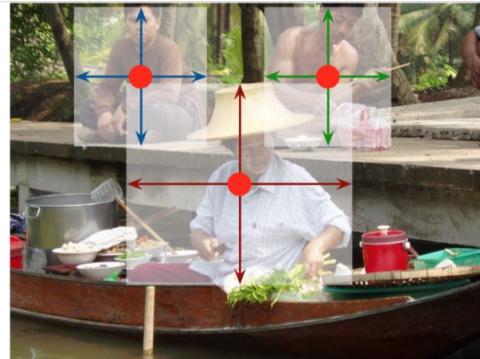
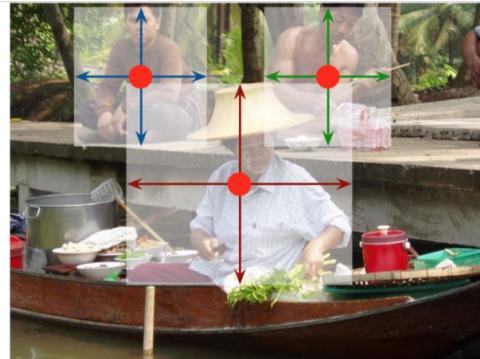
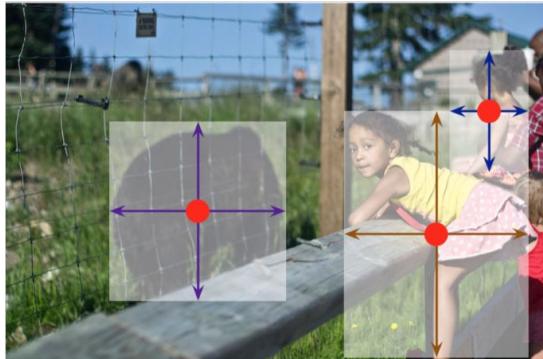
CornerNet 2018 [paper](#)

- вероятность top-left corner
- вероятность bottom-right corner
- эмбеддинг, чтобы сматчить 2 угла



CenterNet 2019 [paper](#)

- координаты центра
- размеры бокса



Barter Island Studies

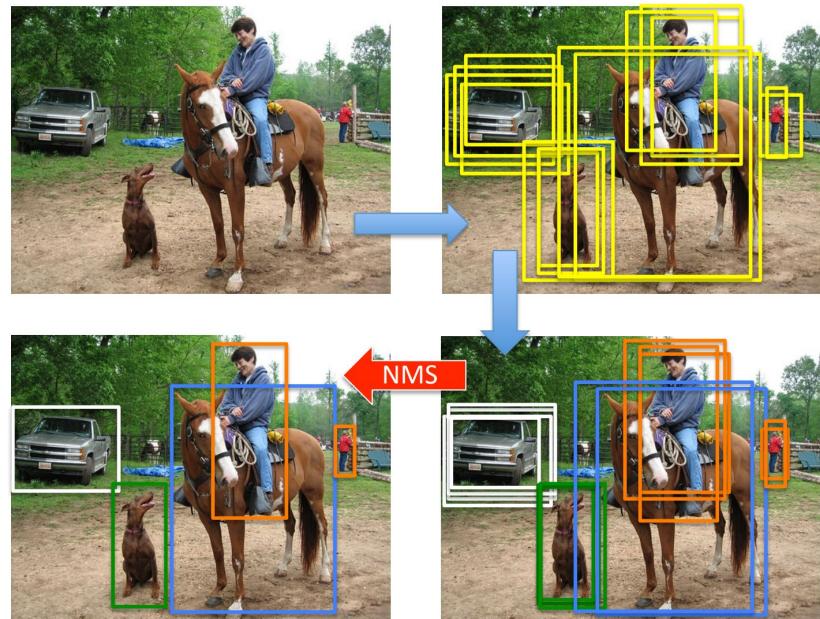
Non-maximum suppression

Дано: P - список предсказанных боксов

Алгоритм NMS:

```
while len( $P$ ) > 0
```

1. взять (с удалением) бокс p из P с наибольшим confidence score $c(p)$
2. удалить из P все боксы q : $\text{IoU}(p,q) > \text{threshold}$



Трансформеры

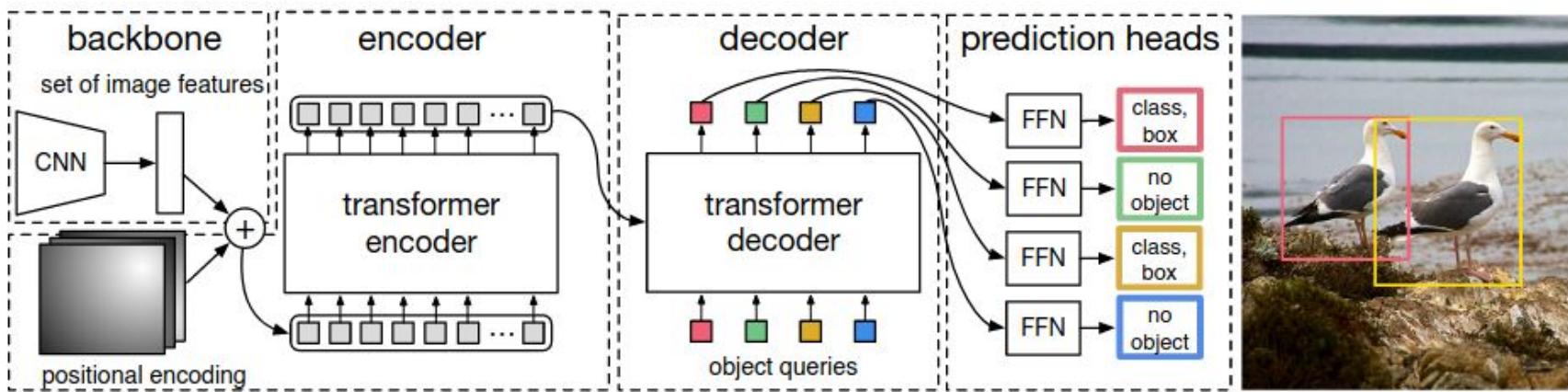
- DETR

DETR

DETR, 2020 [paper](#)

Архитектура

- CNN бэкбоун
- трансформер-энкодер
- трансформер-декодер: на вход N object queries ($N >$ max кол-во объектов на любом фото)
- N FFN головы

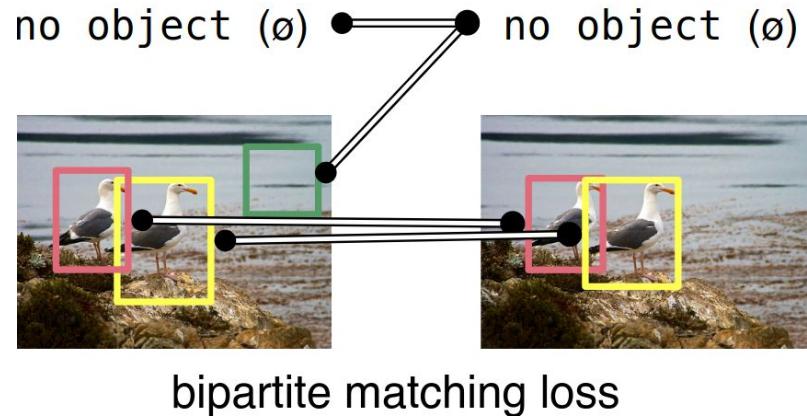


DETR

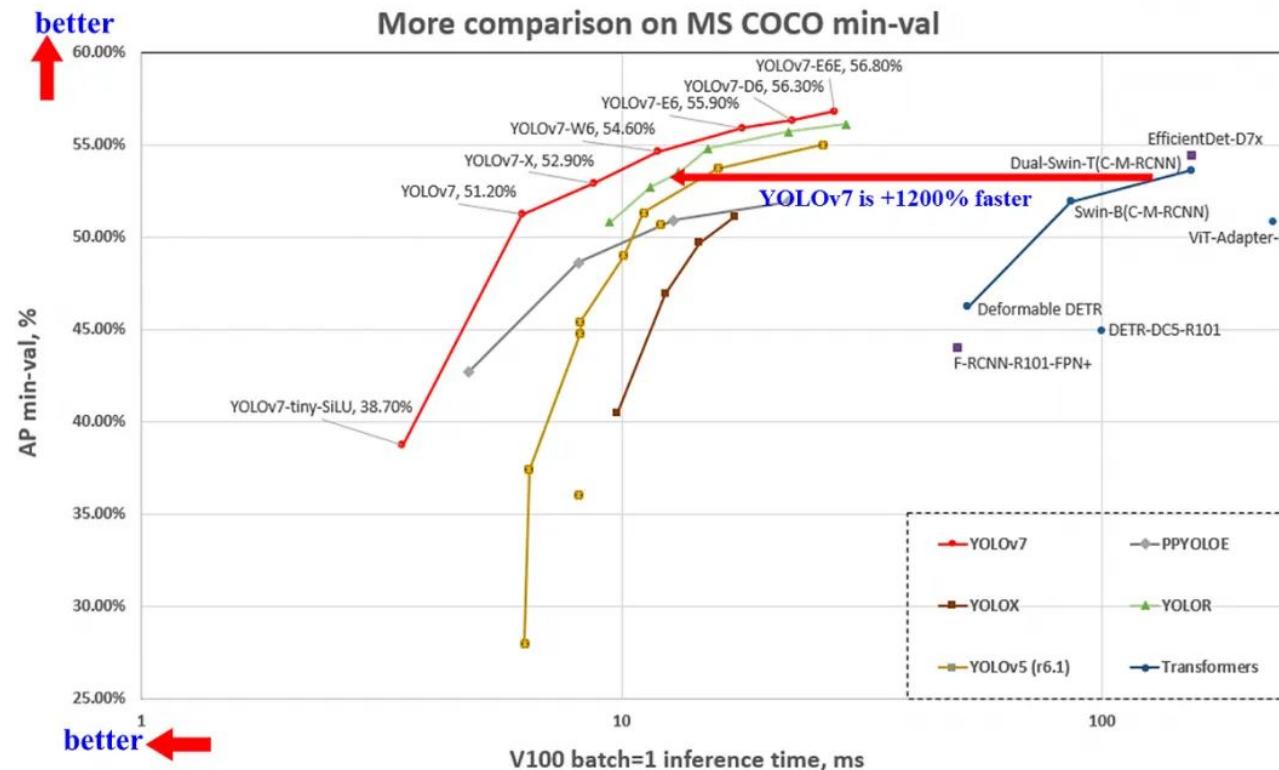
Вместо NMS - bipartite matching loss

- находим лучший матчинг gt и предиктов
- штрафуем за него

Note: количество сходится за счет объектов класса “no object”



CNNs vs Transformers



Сегментация

Виды сегментации

- **Semantic segmentation:** пиксельная классификация
- **Instance segmentation:** выделить пиксели каждого объекта [по отдельности] каждого класса
- **Panoptic segmentation** = semantic + instance:
 - things - instance
 - stuff - semantic

Semantic Segmentation vs. Instance Segmentation vs. Panoptic Segmentation



(a) Image



(b) Semantic Segmentation



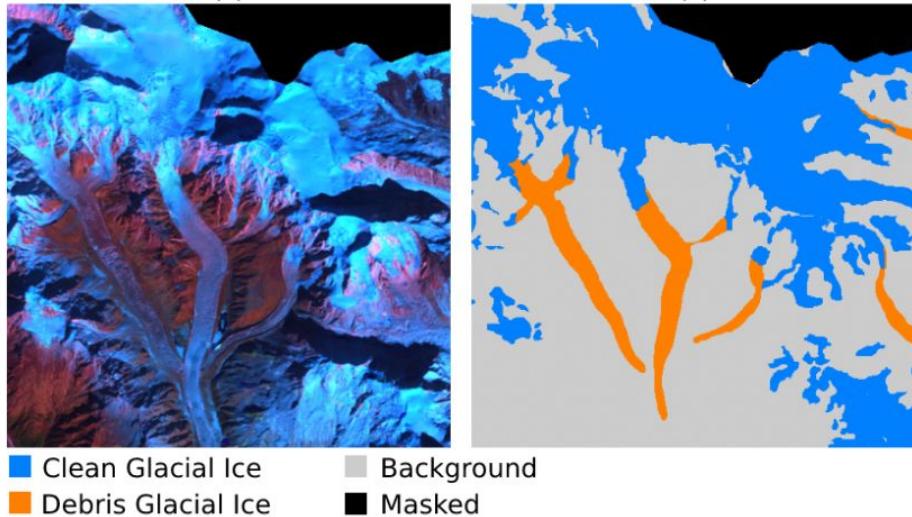
(c) Instance Segmentation



(d) Panoptic Segmentation

source: <https://www.v7labs.com/blog/panoptic-segmentation-guide>

Все виды полезные: semantic

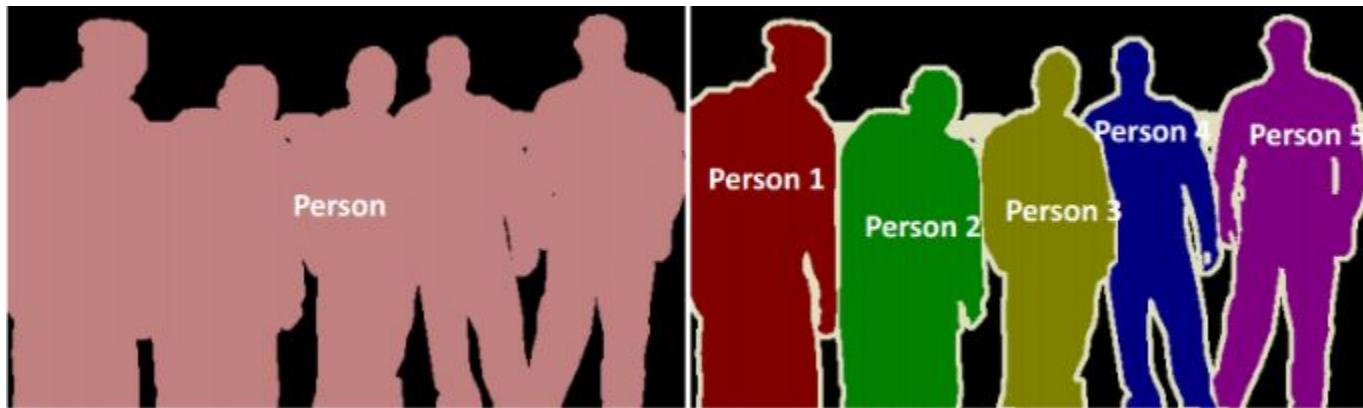


source: <https://www.width.ai/post/semantic-segmentation-vs-instance-segmentation>



Glacier image segmentation (Source: [Arval et al.](#))

Все виды полезные: instance



Все виды полезные: panoptic



Метрики

- Jaccard = IoU
- Dice = F1 бинарной маски

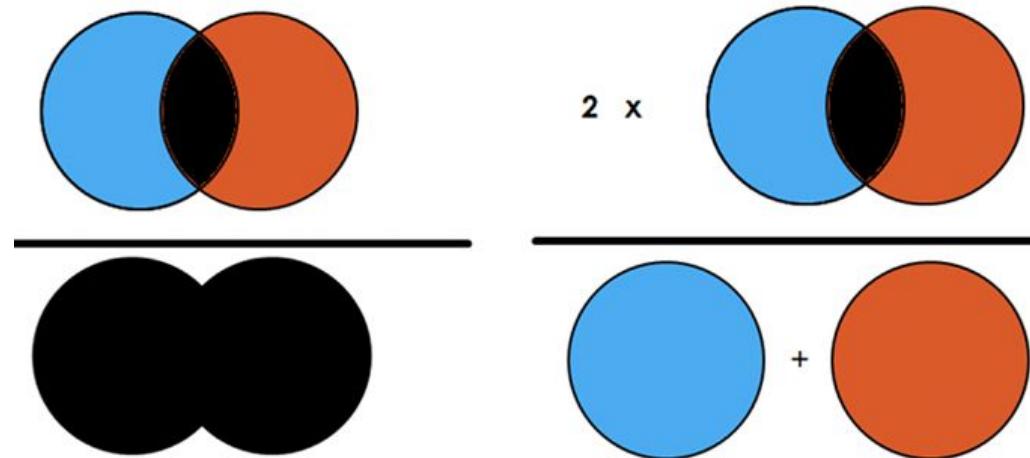


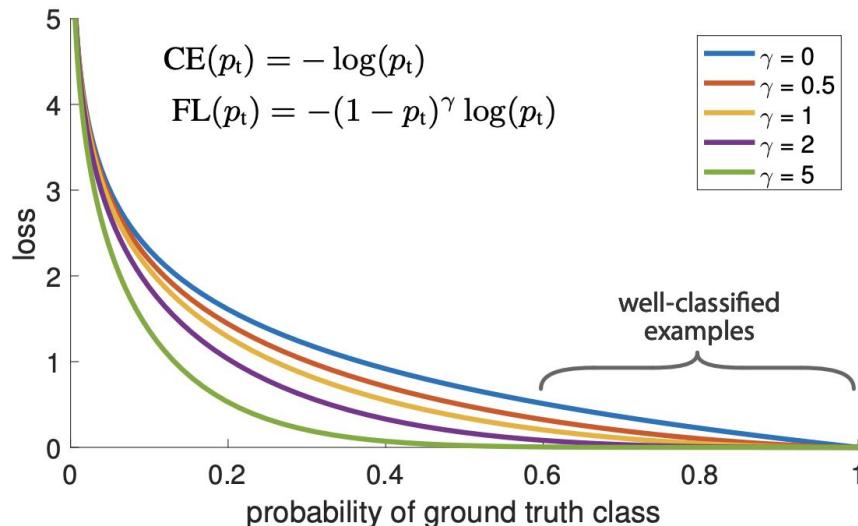
Illustration of IoU and Dice Coefficient.

Модели

- классические (на основе признаков)
 - трешхолдинг
 - кластеризация
 - ...
- CNNs
- трансформеры

ФУНКЦИИ ПОТЕРЬ

- Попиксельная кросс-энтропия
- focal loss - меньше награждать за true negative [paper](#)



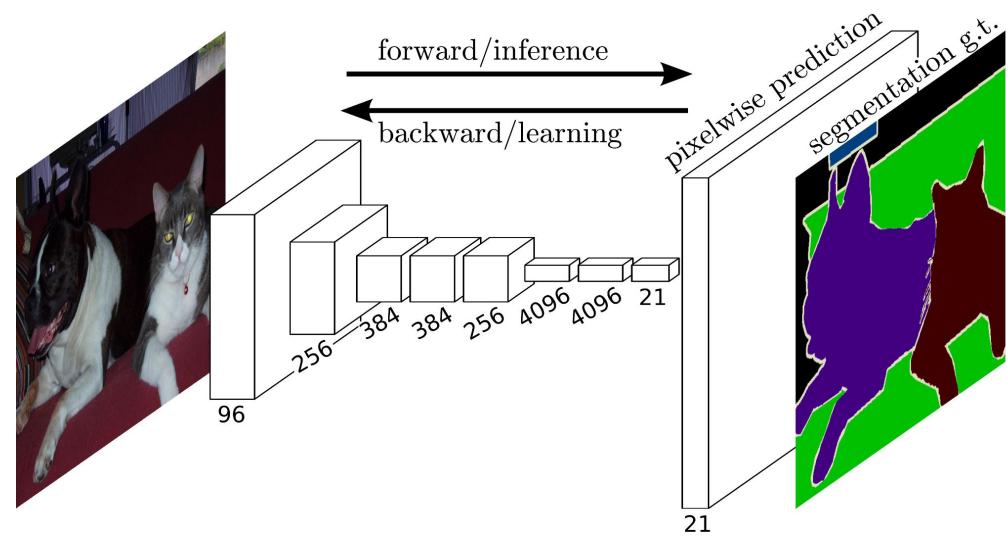
Важные CNN модели

- semantic segmentation
 - FCN
 - UNet
 - DeepLab
- instance segmentation
 - Mask R-CNN
- panoptic segmentation

FCN

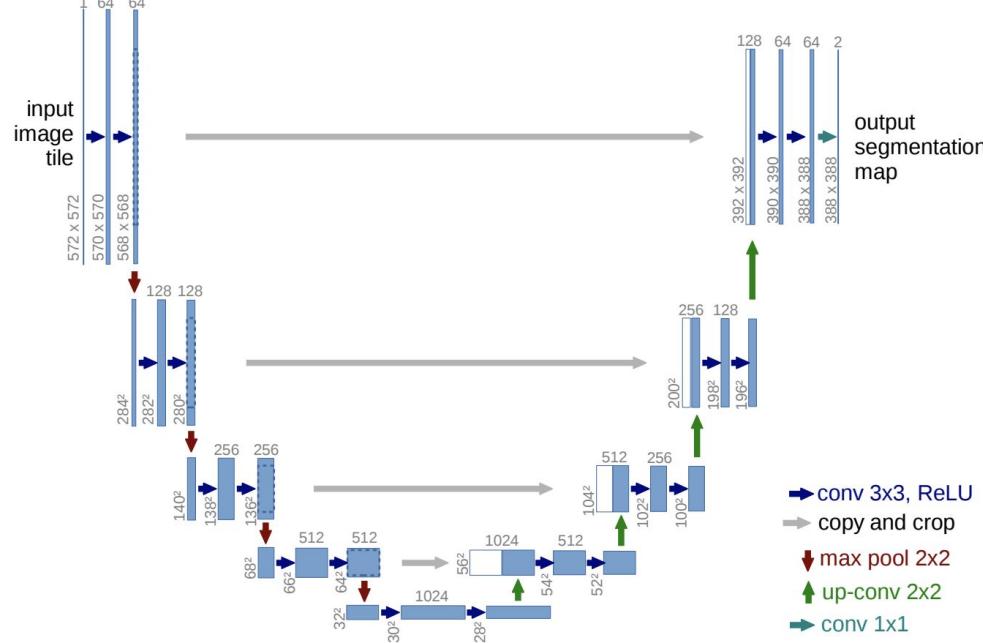
Fully Convolutional Networks 2014 [paper](#)

- энкодер-декодер
- skip-connections



U-Net

U-Net, 2015 [paper](#)

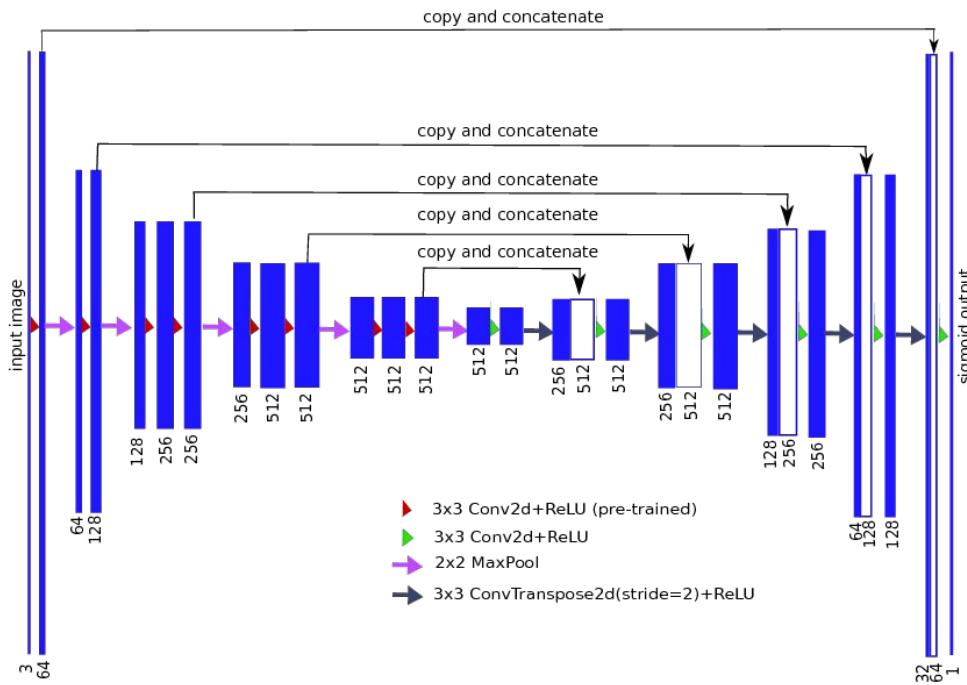


U-Net

TernausNet, 2018 [paper](#)

Идея трансфер лернинга для сегментации:
использовать предобученную на ImageNet
модель классификации в качестве энкодера
(vgg11)

Note: энкодер (encoder) = **backbone**



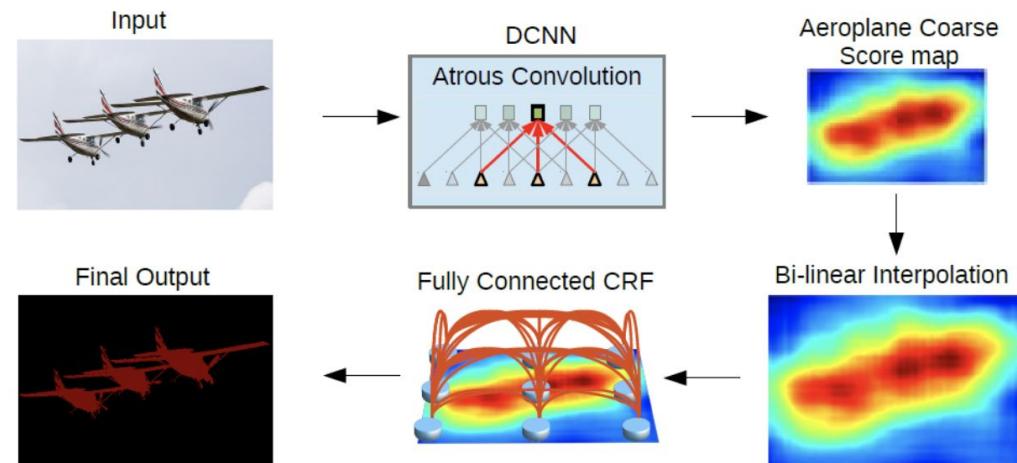
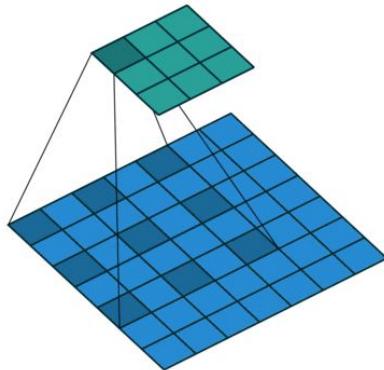
U-Net

- [UNet++](#)
- [Attention U-Net](#)
- SE-UNet
- ...

DeepLab

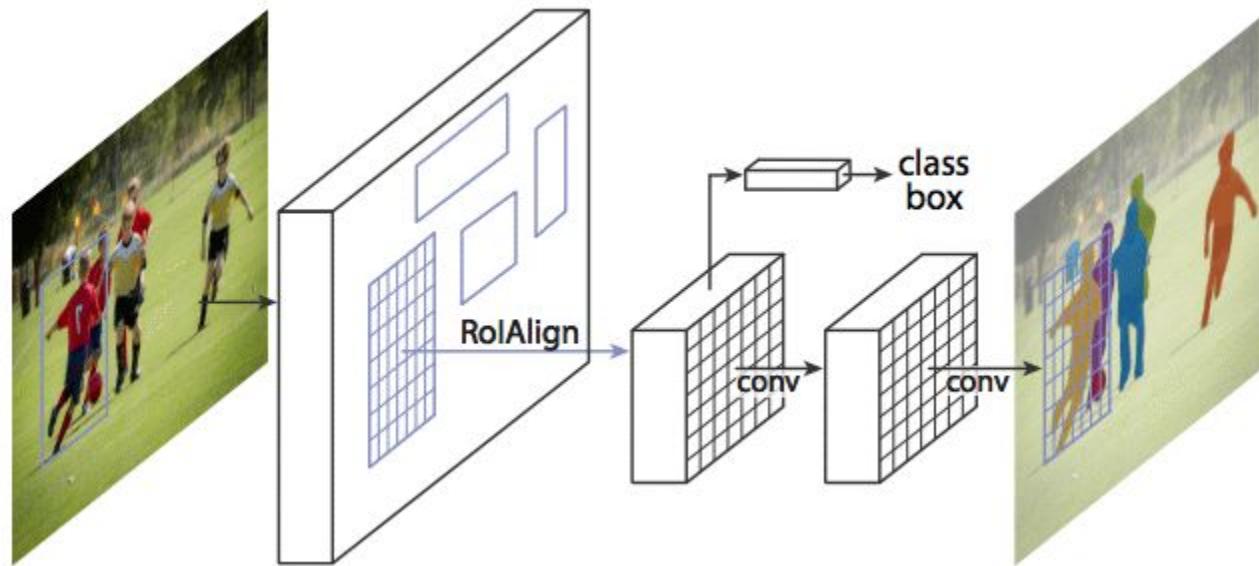
DeepLab 2014 [paper](#)

- нет энкодера-декодера
- dilated (atrous) convolutions
- CRF for refinement



Mask R-CNN

Mask R-CNN 2017 [paper](#)



Трансформеры для сегментации

- SETR
- SegFormer
- Mask2Former
- Segmenter
- CLIPSeg
- SAM
- ...

SETR

SETR 2020 [paper](#)

проблема FCNs: ограниченность receptive поля

решение: трансформер вместо CNN энкодера

архитектура SETR:

- [a] ViT-like трансформер
- декодер
 - [b] progressive upsampling (SETR-PUP)
 - [c] multi-level feature aggregation (SETR-MLA)

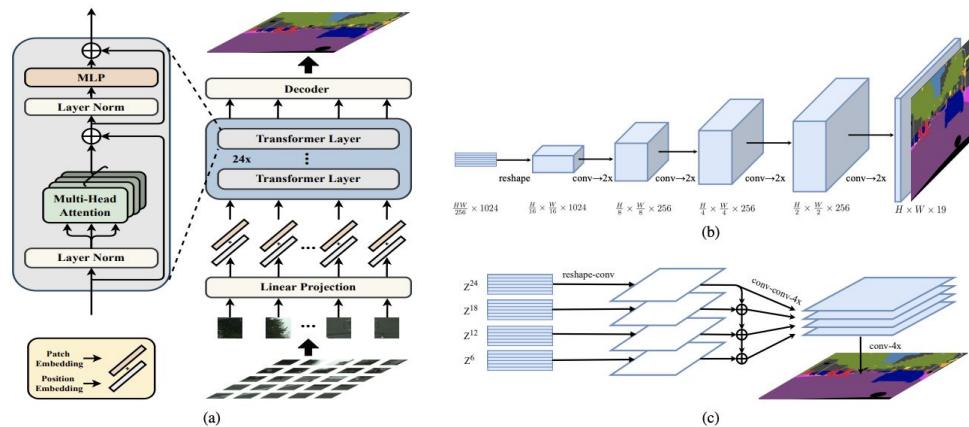


Figure 1. Schematic illustration of the proposed **Segmentation Transformer (SETR)** (a). We first split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. To perform pixel-wise segmentation, we introduce different decoder designs: (b) progressive upsampling (resulting in a variant called SETR-PUP); and (c) multi-level feature aggregation (a variant called SETR-MLA).

SegFormer

SegFormer 2021 [paper](#)

- энкодер - трансформер
- декодер - MLP

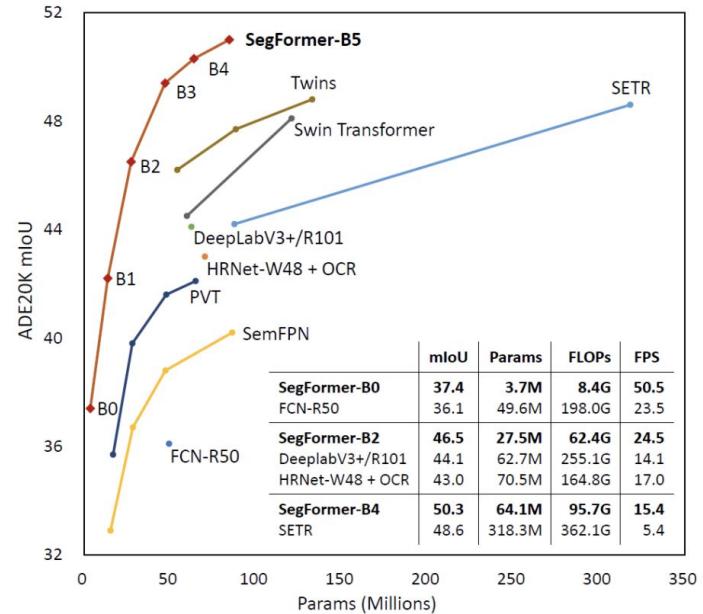
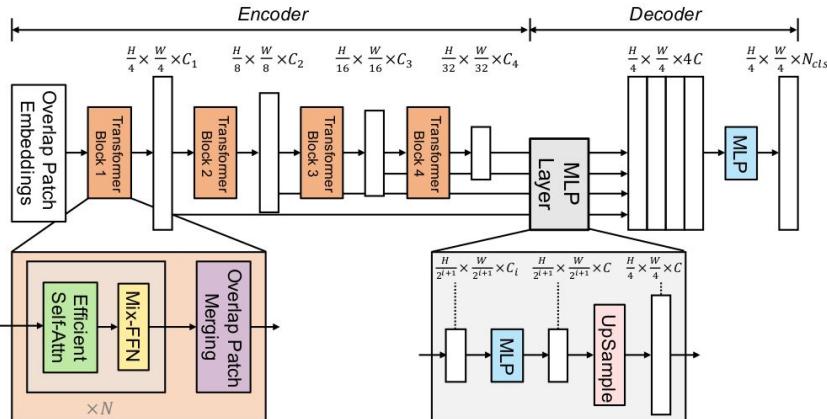


Figure 1: Performance of SegFormer-B0 to SegFormer-B5.

Any type segmentation

с 2020 тренд:

- одна архитектура
- можно обучить на любой тип сегментации (semantic, instance, panoptic)

DETR - panoptic segmentation

panoptic segmentation

- k things object queries добавляем stuff object queries
 - things - instances
 - stuff - background things: trees, sky, sand, grass, ...
- добавляем mask head

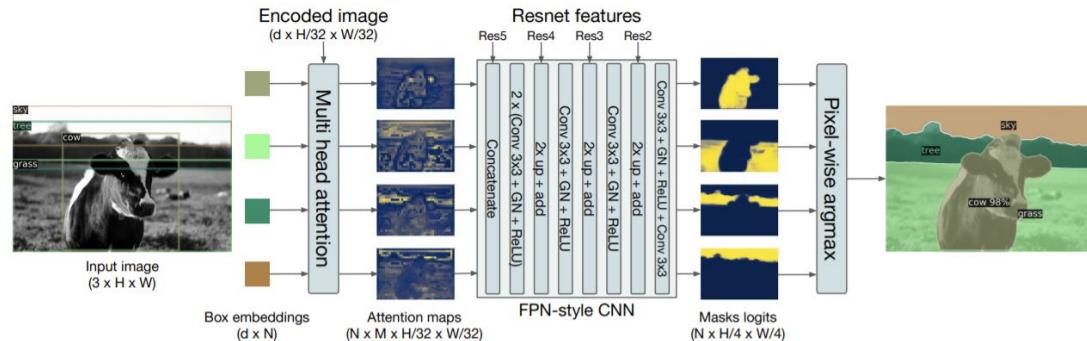
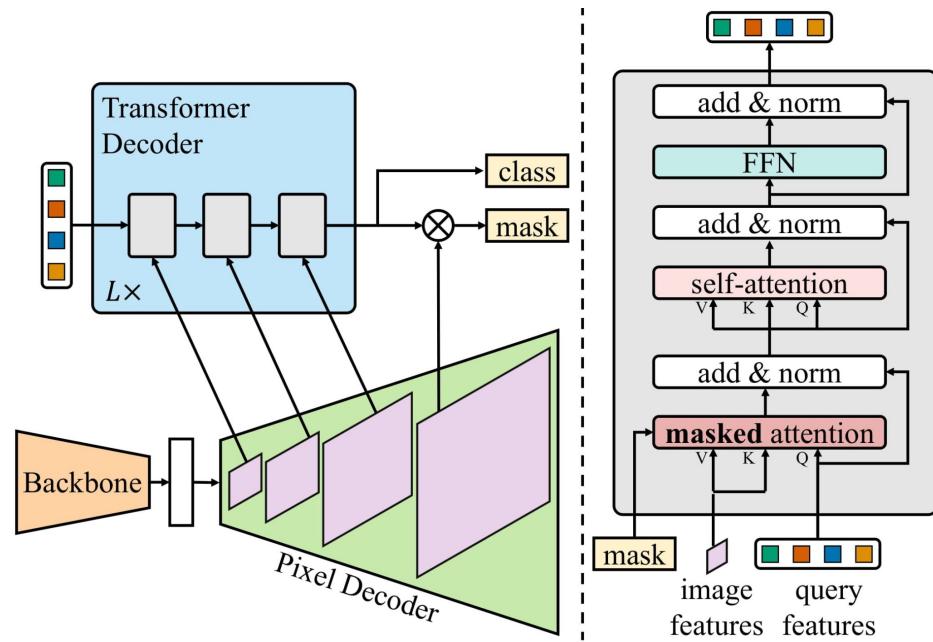


Fig. 8: Illustration of the panoptic head. A binary mask is generated in parallel for each detected object, then the masks are merged using pixel-wise argmax.

Mask2Former

Mask2Former, 2020 [paper](#)



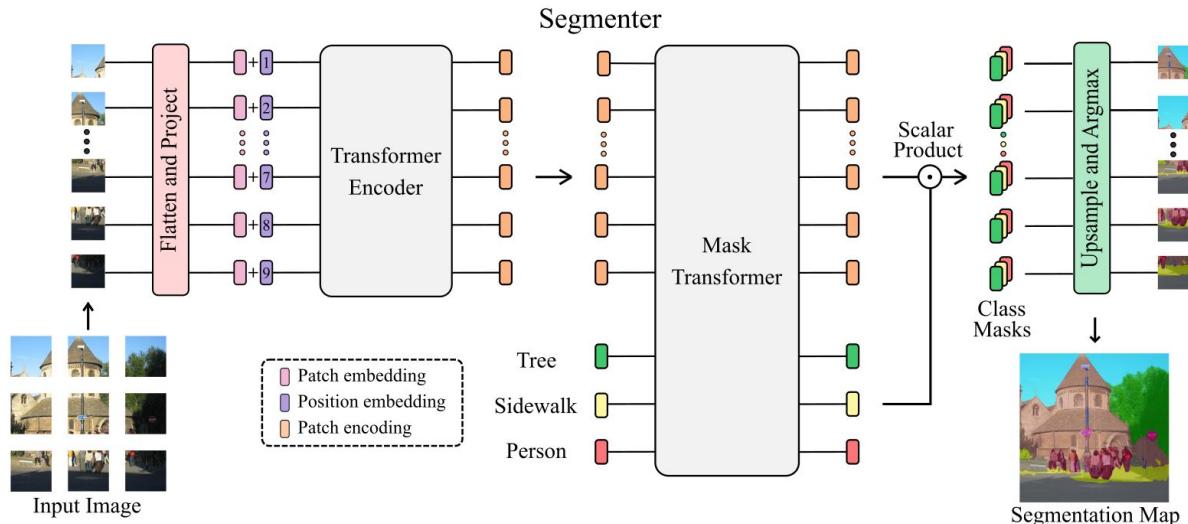
Segmenter

Segmenter 2021 [paper](#)

- полностью трансформер
- энкодер-декодер

Дополнительно

- 3 вида сегментации
- на каждый отдельное обучение

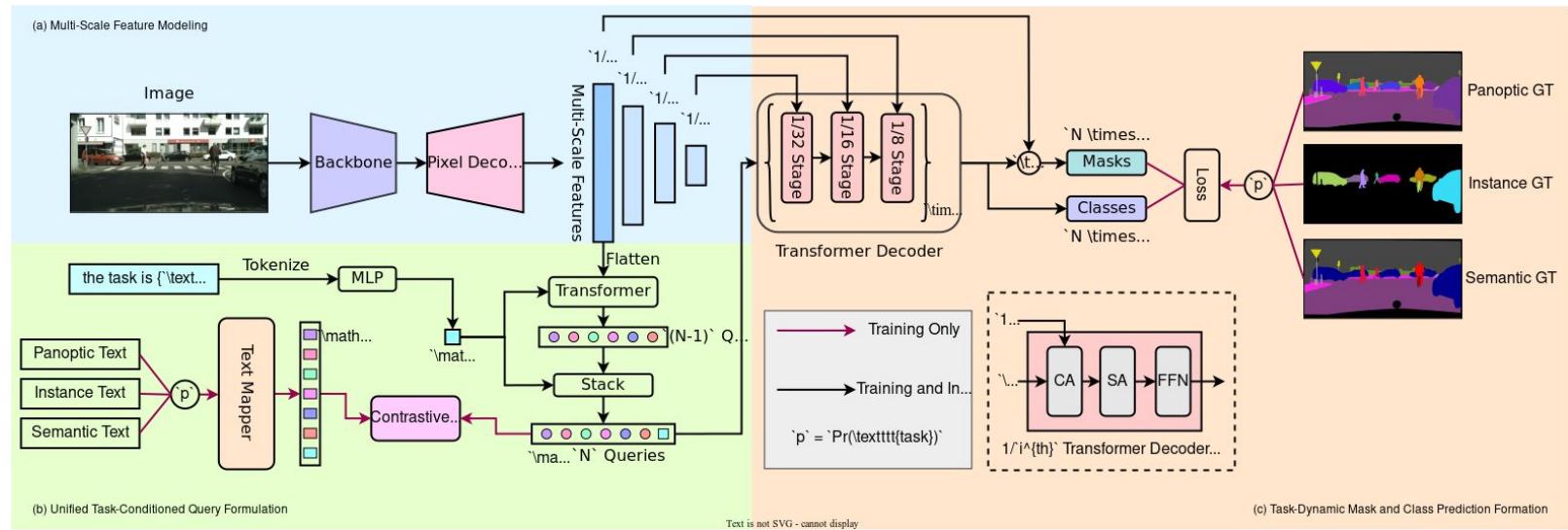


All types segmentation

Задача: делать любой тип сегментации одной моделью

Идея: передавать тип сегментации через промпт

Реализация - OneFormer 2022 [paper](#)

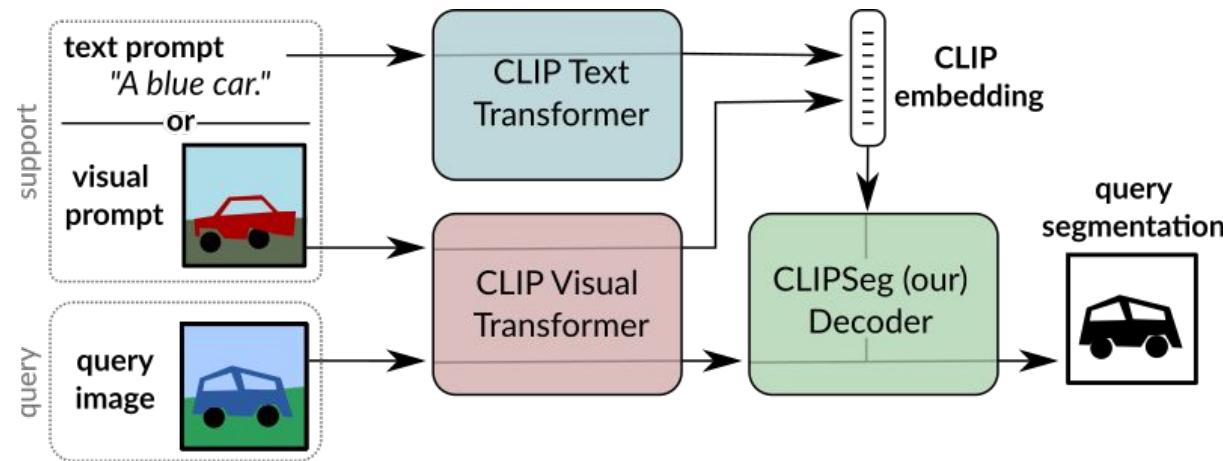


Open-World (0-shot) Segmentation

Задача: сегментим что угодно, а не predefined классы

Идея: использовать CLIP

Реализация - CLIPSeg 2021 [paper](#)



Foundation Segmentation Models

- EVA 2023 [github](#)
- SAM 2023 [github](#)

SAM

раньше 2 типа сегментации:

- автоматическая по заранее определенным классам
(с обучением по куче размеченных данных)
- интерактивная сегментация: сегментируют что угодно, но с участием человека, постепенно уточняющего маску подсказками

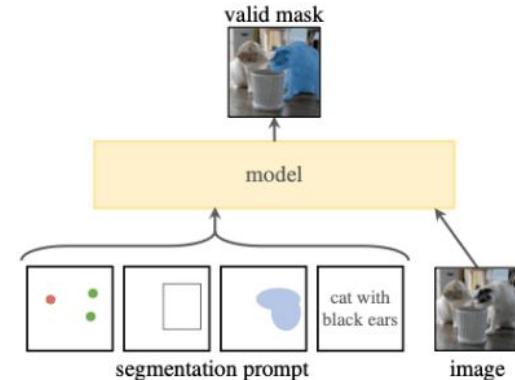
теперь: SAM сегментирует что угодно по одному промпту или вообще просто так

SAM

Функционал:

1. сегментировать любой объект по промпту

- sparse prompts
 - 1+ точки объекта
 - ббокс объекта
 - текстовый промпт
- dense prompts
 - примерная маска



SAM

Функционал:

1. сегментировать любой объект по промпту

- sparse prompts
 - 1+ точки объекта
 - ббокс объекта
 - текстовый промпт
- dense prompts
 - примерная маска



SAM

Функционал:

2. сегментировать все

- стартуют с грида точек
- по каждой как по промпту делают сегментацию

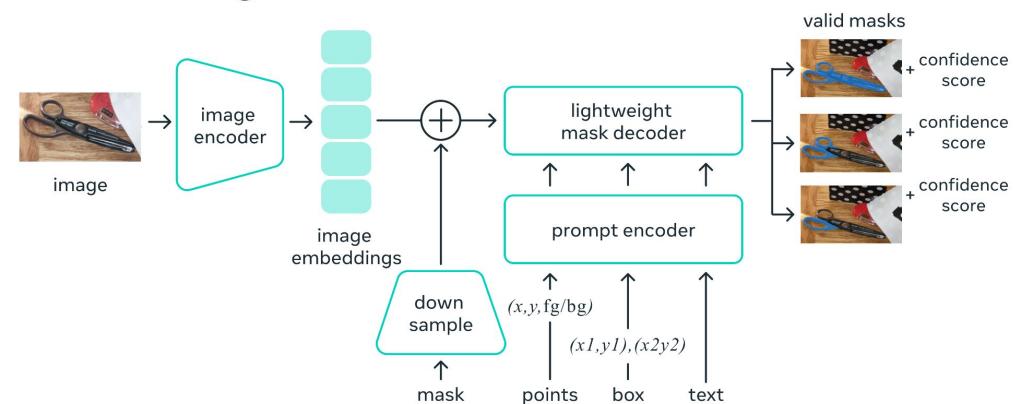


SAM

Архитектура

- image encoder (предобученный MAE)
- prompt encoder
- легковесный трансформер-декодер

Universal segmentation model



SAM

Как обучали: огромный датасет с 1b данных: SA-1B

Как собрали:

1. начали с качественной human-made аннотации
2. затем аннотация, размеченная людьми с помощью SAM, обученного на уже собранной аннотации
3. затем self-training

Спасибо за внимание!