Лаборатория компьютерной
графики и мультимедиа
ВМК МГУ имени М.В. Ломоносова

Курс «Компьютерное зрение»

«Сегментация изображений»

Антон Конушин и Тимур Мамедов

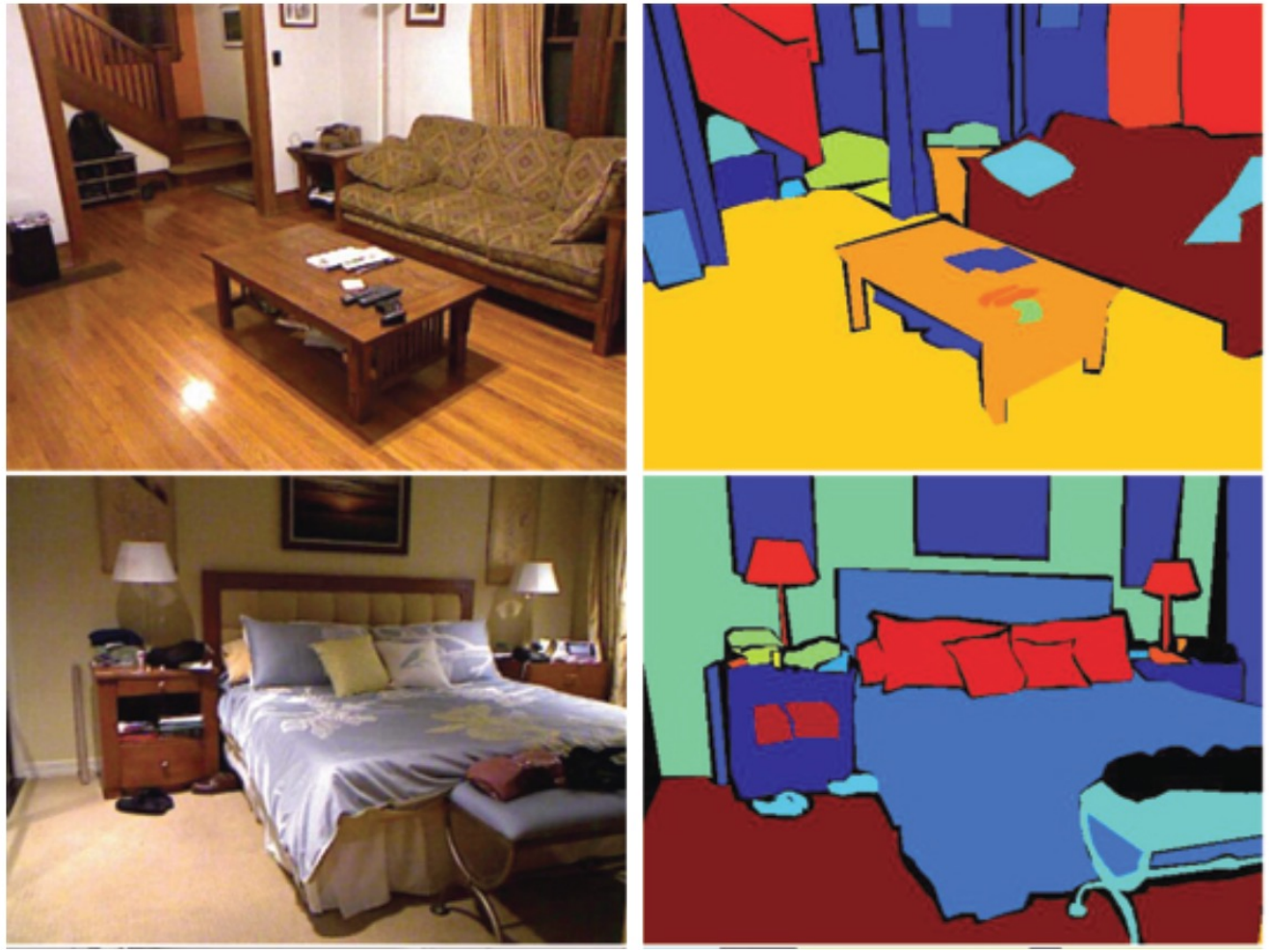
2025 год

План лекции

1. Такая разная сегментация
2. Пересегментация
3. Семантическая сегментация
4. Интерактивная
5. Сегментация экземпляров
6. Паноптическая сегментация
7. Оценка позы человека

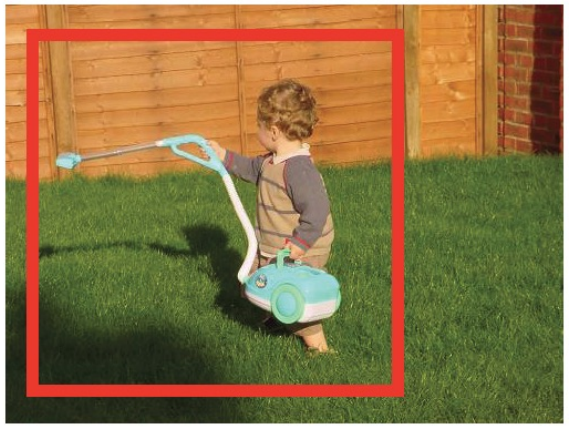
Задача сегментации

Разделение изображения на фрагменты(группы пикселов) по определённому критерию



Извлечение объекта

Выделение конкретного произвольного объекта, указанного пользователем или по другому заданного



Сегментация без учителя

Разделение изображения на регионы, однородные по своим визуальным характеристикам, и отличающихся от соседних регионов

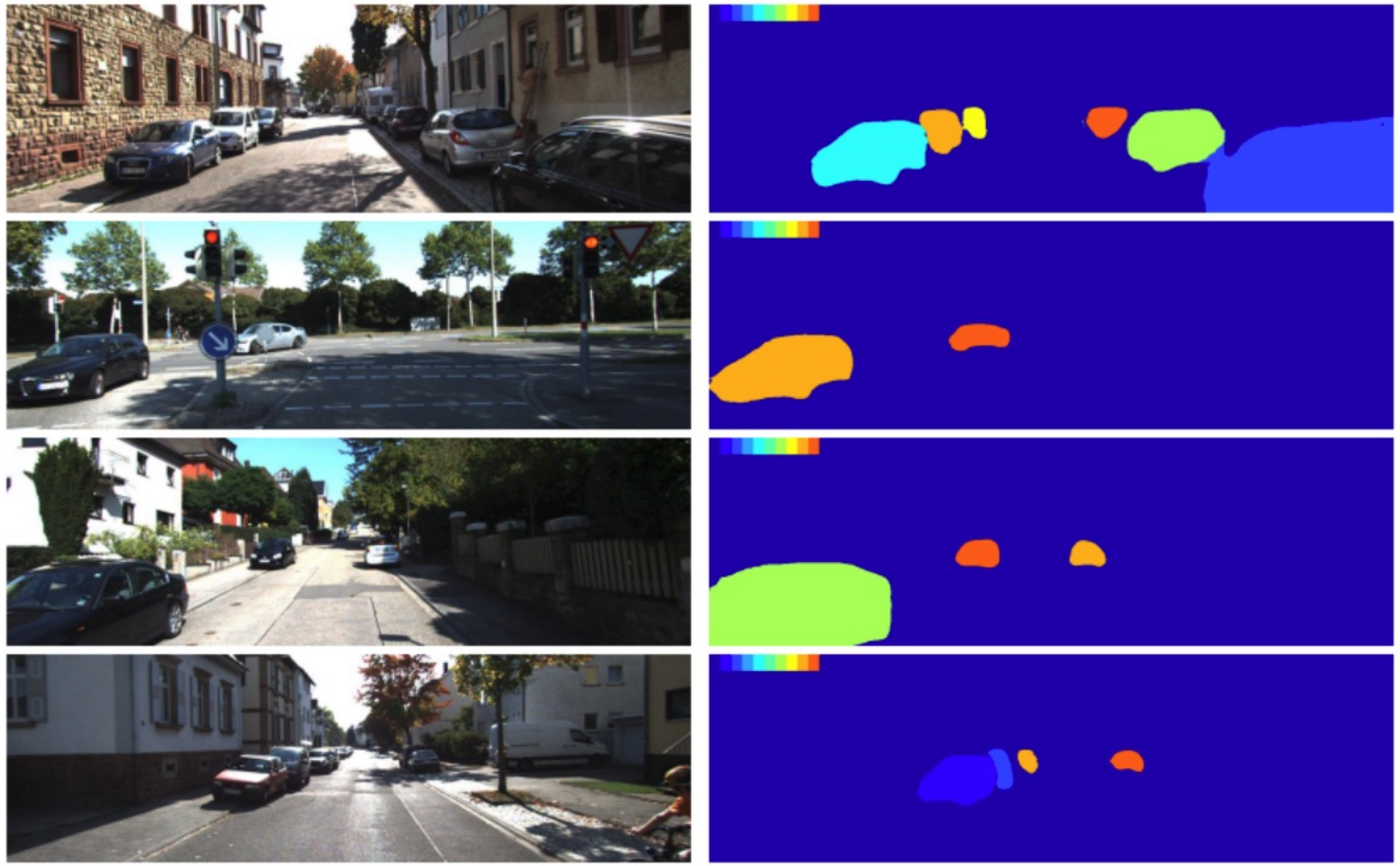
Семантическая сегментация

Попиксельная разметка изображения, где каждая метка соответствует определенному объекту

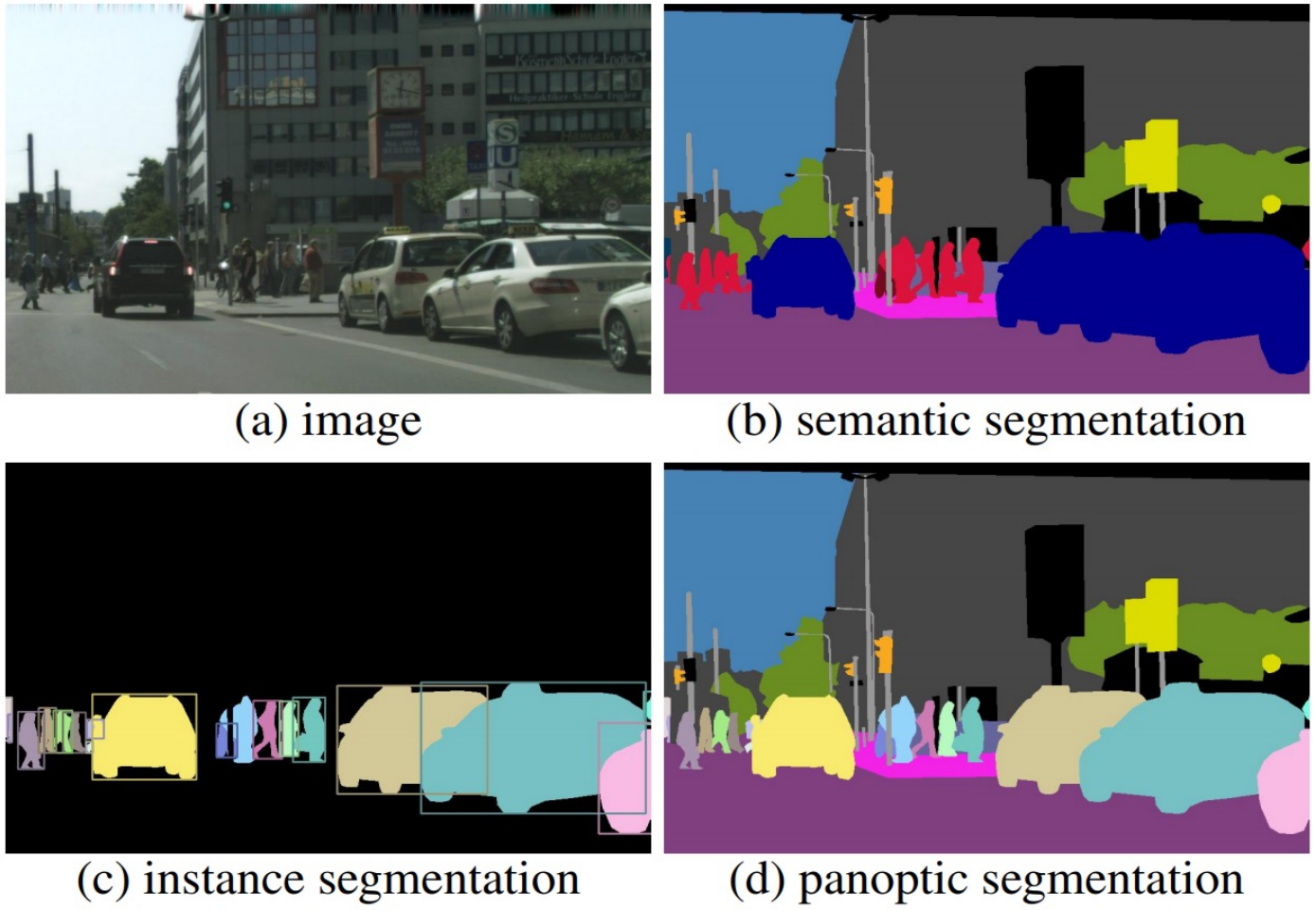
- Разные пиксели одного объекта существенно отличаются друг от друга по признакам (яркости, цвету, текстуре окрестности)
- Единственное, что у них общее – это «семантика»
- Поэтому задача сегментации объекта тесно связана с задачей распознавания

Instance segmentation

Выделение всех отдельных экземпляров объектов определённого класса. Каждый объект помечен своей меткой.



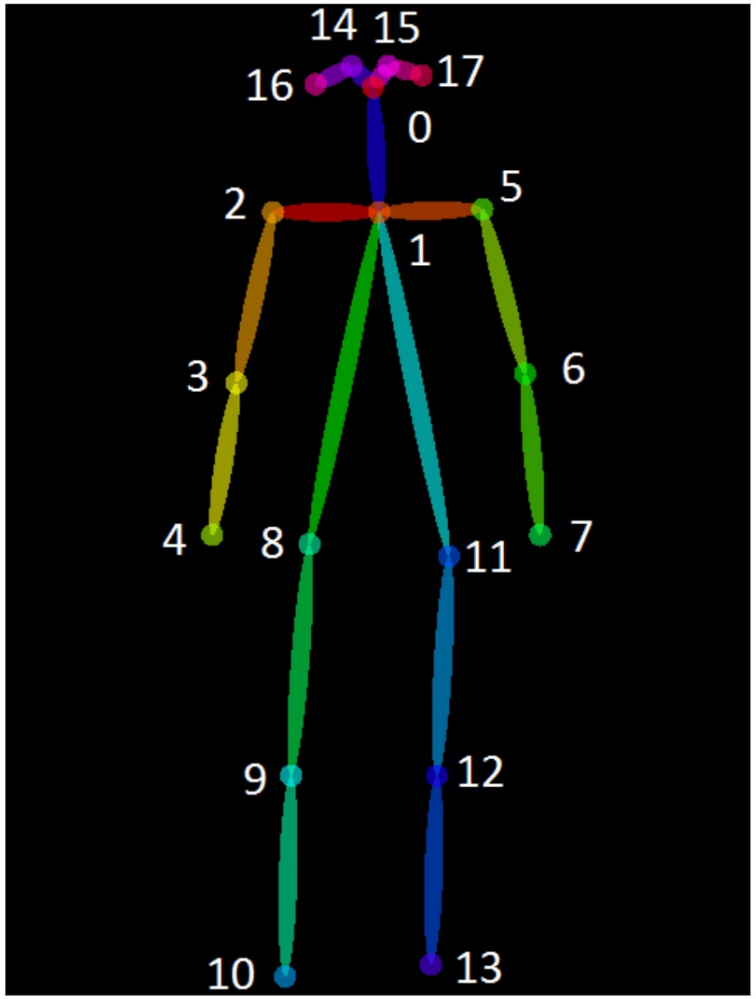
Panoptic segmentation



Объединение задача semantic и instance segmentation в единую модель

Source: https://arxiv.org/pdf/1801.00868.pdf

Оценка позы человека



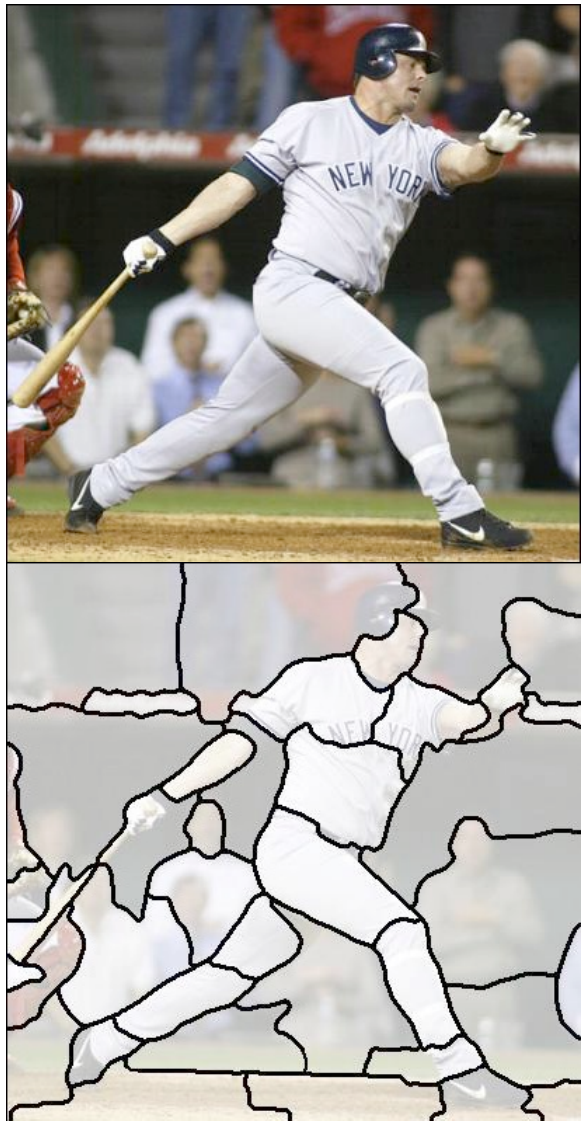
Photograph taken from Pexels

Какая связь с сегментацией?

План лекции

1. Такая разная сегментация
2. Визуальная сегментация
3. Семантическая сегментация
4. Интерактивная
5. Сегментация экземляров
6. Паноптическая сегментация
7. Оценка позы человека

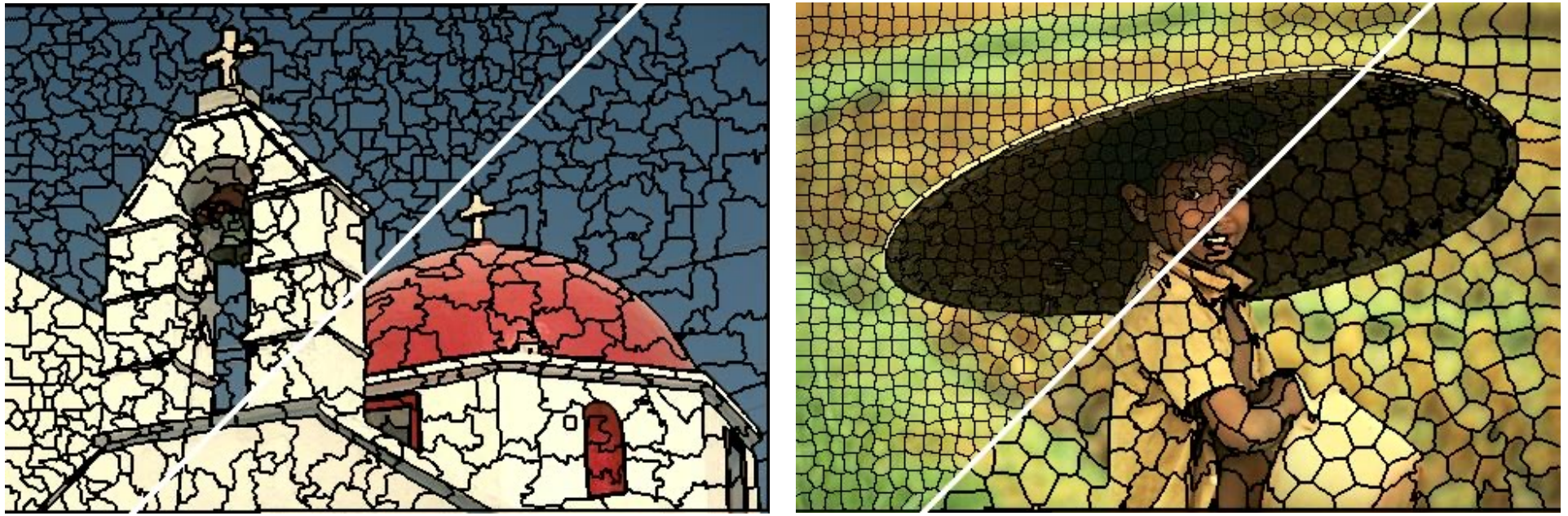
Требования к сегментации?



Каким условиям должны удовлетворять сегменты?

- Границы сегментов должны соответствовать границам объектов
- Сегмент должен целиком содержаться внутри объекта
- Небольшие объекты не должны быть частью сегмента, а описываться своим сегментом
- Сегмент должен быть однородным по визуальным характеристикам
- Сегменты должны быть достаточно большими, чтобы быть «информативными»
- Компактные, примерно одного размера
- Равномерно распределенные по изображению
- Алгоритм должен работать быстро

Суперпиксели

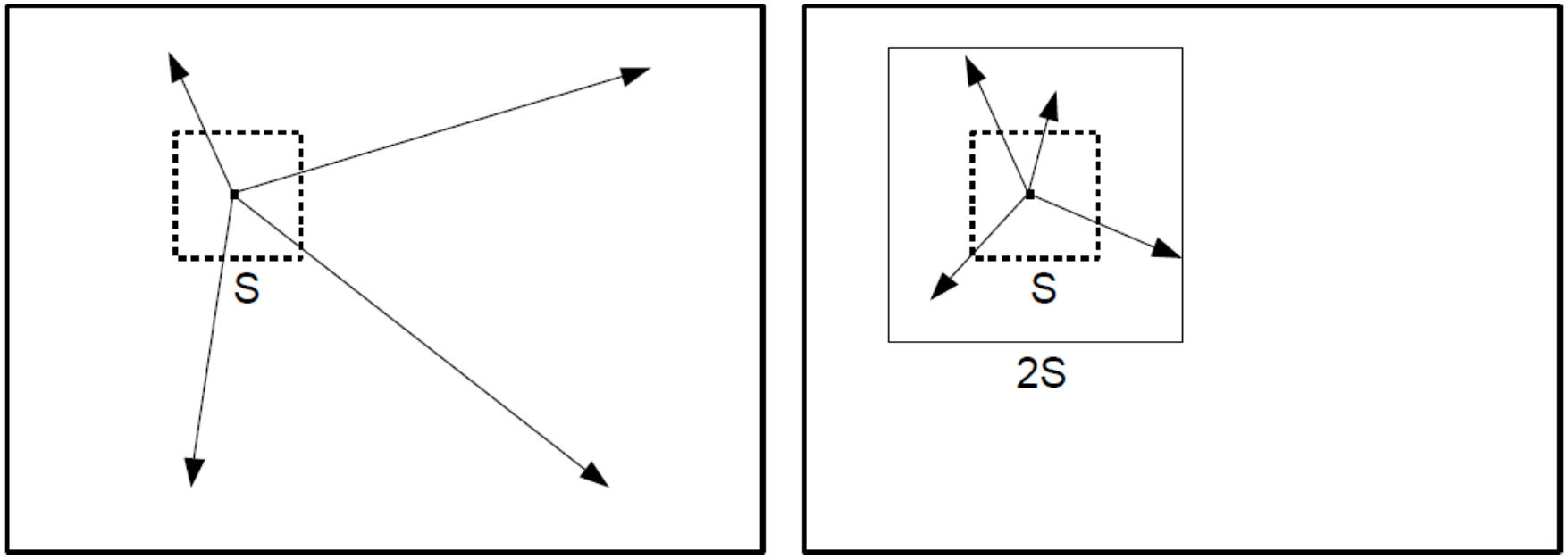


- Сегменты, «удовлетворяющие» вышеуказанным требованиям называют «суперпиксели»
- «Суперпиксельная сегментация»
- Ещё называют «пересегментацией» (oversegmentation)

Пересегментация через кластеризацию

- Представим множество пикселов как выборку (множество вектор-признаков)
- Применим какой-нибудь метод кластеризации к данным в пространстве признаков
- Каждый кластер будет соответствовать одному сегменту
- Как обеспечить «локальность» кластеров в изображении?
  - (x,y) можно включить в вектор-признак
  - Жесткие ограничения на расстояния между пикселами

Simple linear iterative clustering (SLIC)



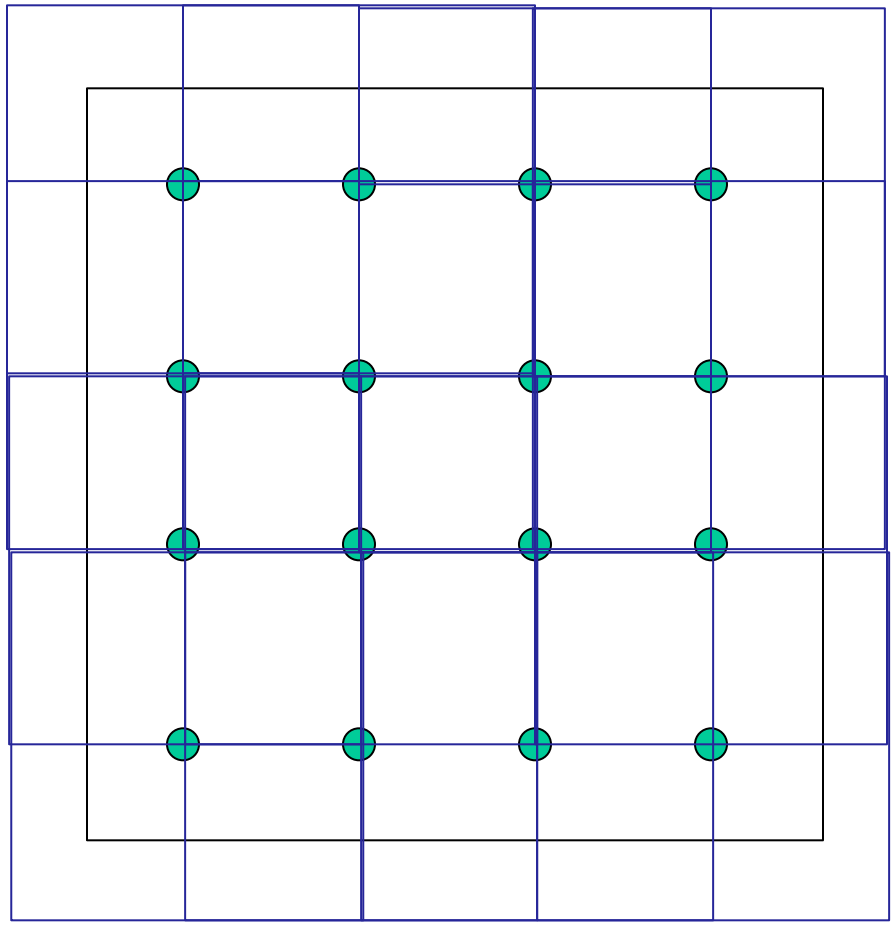
(a) standard k-means searches the entire image

(b) SLIC searches a limited region

- Зачем сопоставлять пиксель со *всеми* остальными?
- Будем сравнивать с «центром кластера» только пиксели, которые *могут принадлежать* этому сегменту (на расстоянии < s)
- Инициализируем кластеры по сетке на расстоянии s

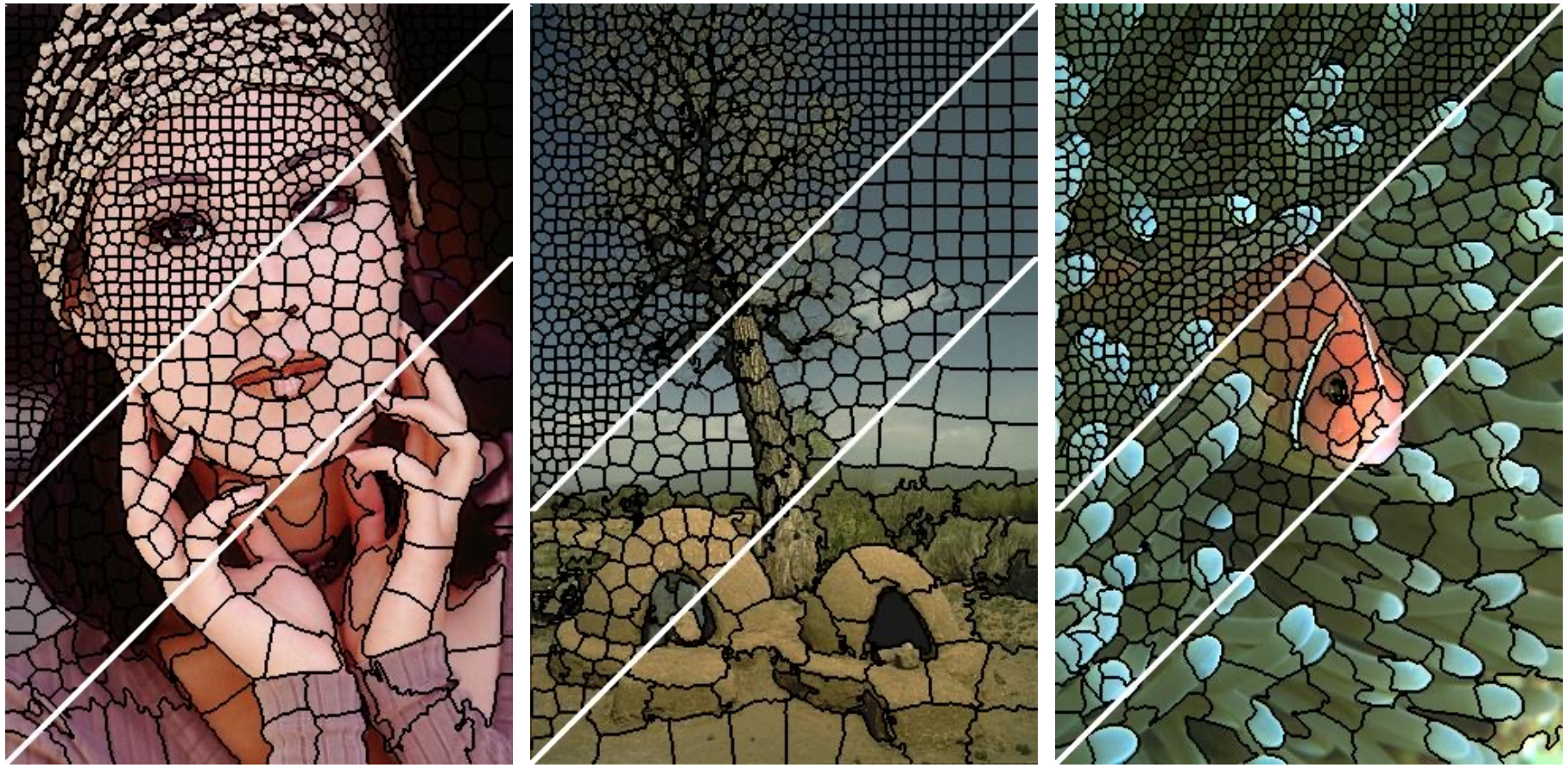
Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk, SLIC Superpixels Compared to State-of-the-art Superpixel Methods, IEEE PAMI 2012.

Алгоритм SLIC



- Инициализируем центры кластеров C_k по сетке с шагом S
- Инициализируем в каждом пикселе метку L(i)=-1 и расстояние D(i) до ближайшего кластера -∞
- Проходим по кластерам C_k
  - Для каждого пикселя в области 2Sx2S считаем расстояние до C_k
  - Расстояние считаем в CIE LAB + (x,y)
  - Если расстояние меньше D(i), тогда ставим метку L(i)= k, и записываем в D(i) новое значение
- Повторяем до тех пор, пока суммарное изменение кластеров не будет меньше порога

Примеры работы

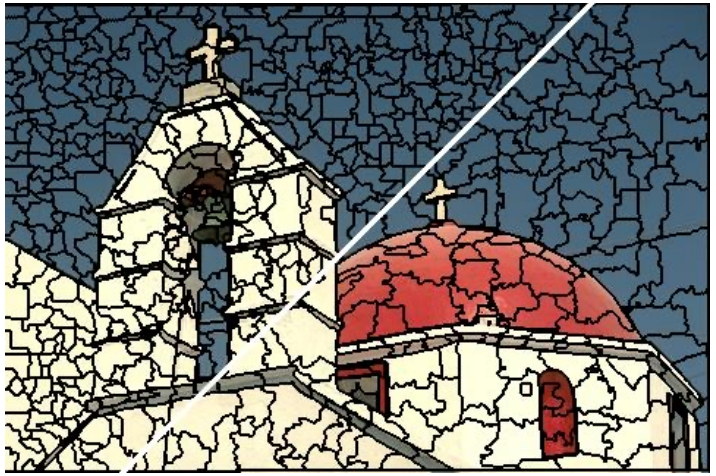


Сложность - $\left(\frac{n}{s}\right)^2 S^2 t = n^2 t = Nt$

Визуальное сравнение

Efficient Graph-Based

TurboPixel



QuickShift

SLIC

План лекции

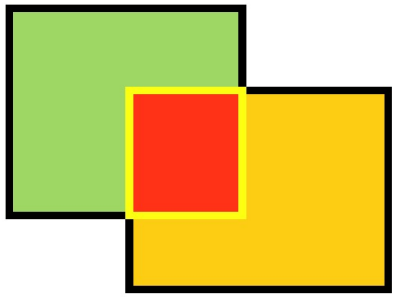
1. Такая разная сегментация
2. Пересегментация
3. Семантическая сегментация
4. Интерактивная
5. Сегментация экземляров
6. Паноптическая сегментация
7. Оценка позы человека

Семантическая сегментация

Метрики качества

Source: https://arxiv.org/abs/1611.09326

- Intersection over Union (IoU) – по классам



$$\text{segmentation accuracy} = \frac{\text{true positives}}{\text{true positives} + \text{false positives} + \text{false negatives}}$$

- mIoU (mean IoU) – средняя IoU по всем классам
- Per-class pixel accuracy:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

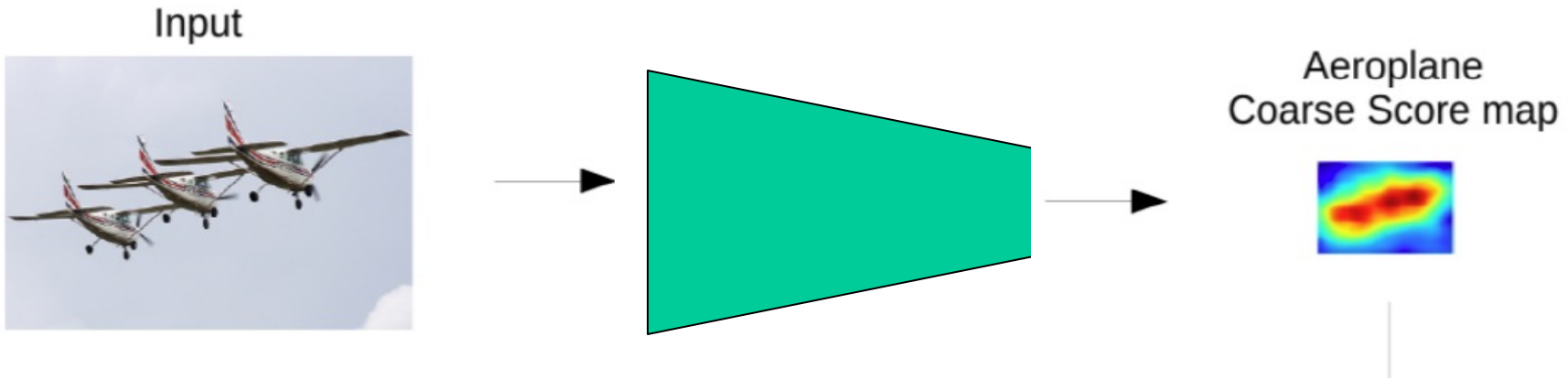
https://www.jeremyjordan.me/evaluating-image-segmentation-models/

Cityscapes как пример датасета (CVPR 2016)

- Изображения с камеры автомобиля
- 30 классов объектов
- 5000 хорошо размеченных и 20000 грубо размеченных изображений
- 1.5 часа на разметку 1 изображения

https://www.cityscapes-dataset.com/

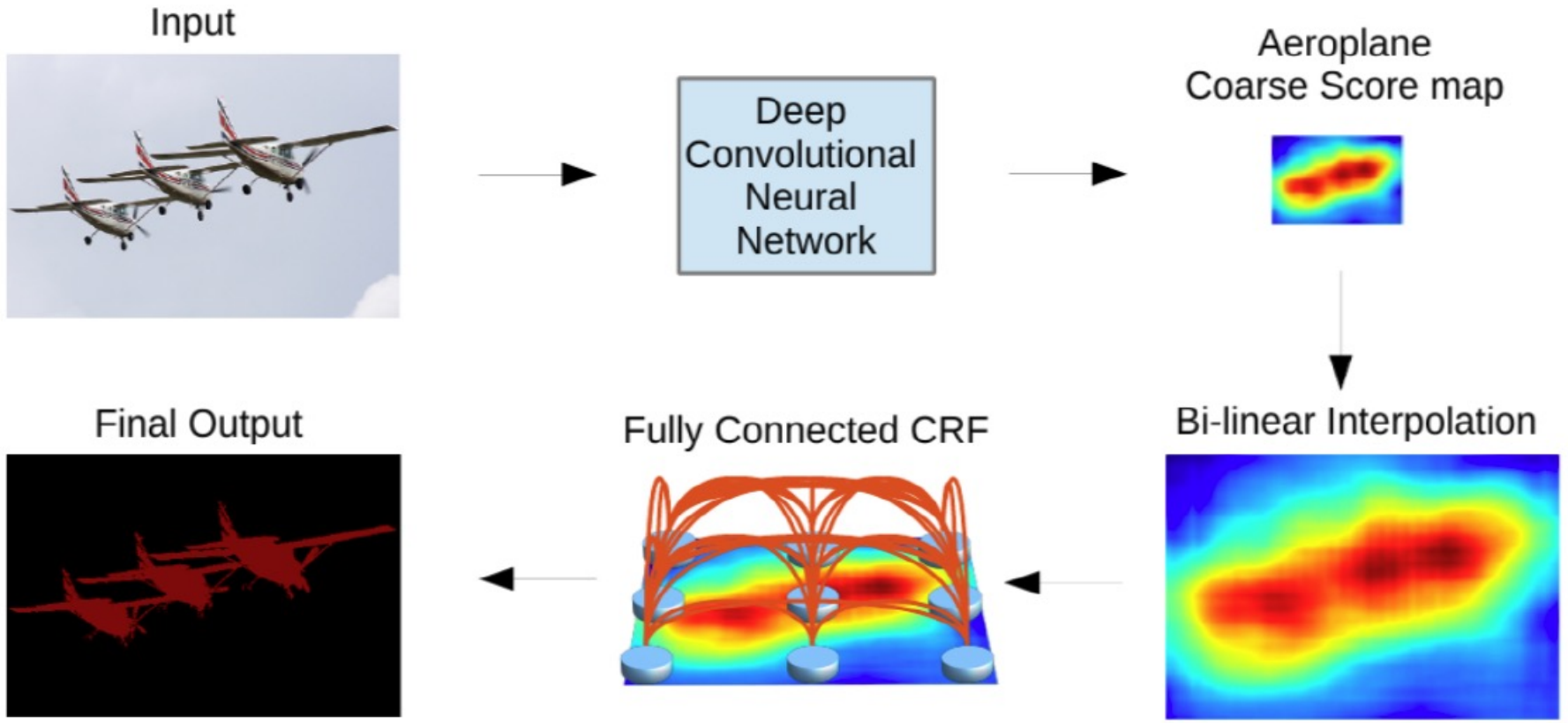
Сегментация по свёрточным признакам



- Сегментацию можно представить как попиксельную классификацию
- Применим свёрточную архитектуру для классификации к изображению, пр. VGG16
- Получим матрицу признаков с разрешением в 32 раз меньше
- Для каждого пикселя признаков применим классификатор для оценки классов
  - Примерно как RPN в Faster R-CNN
- Получим карту разметки низкого разрешения
- Нужно повысить разрешение до исходного!

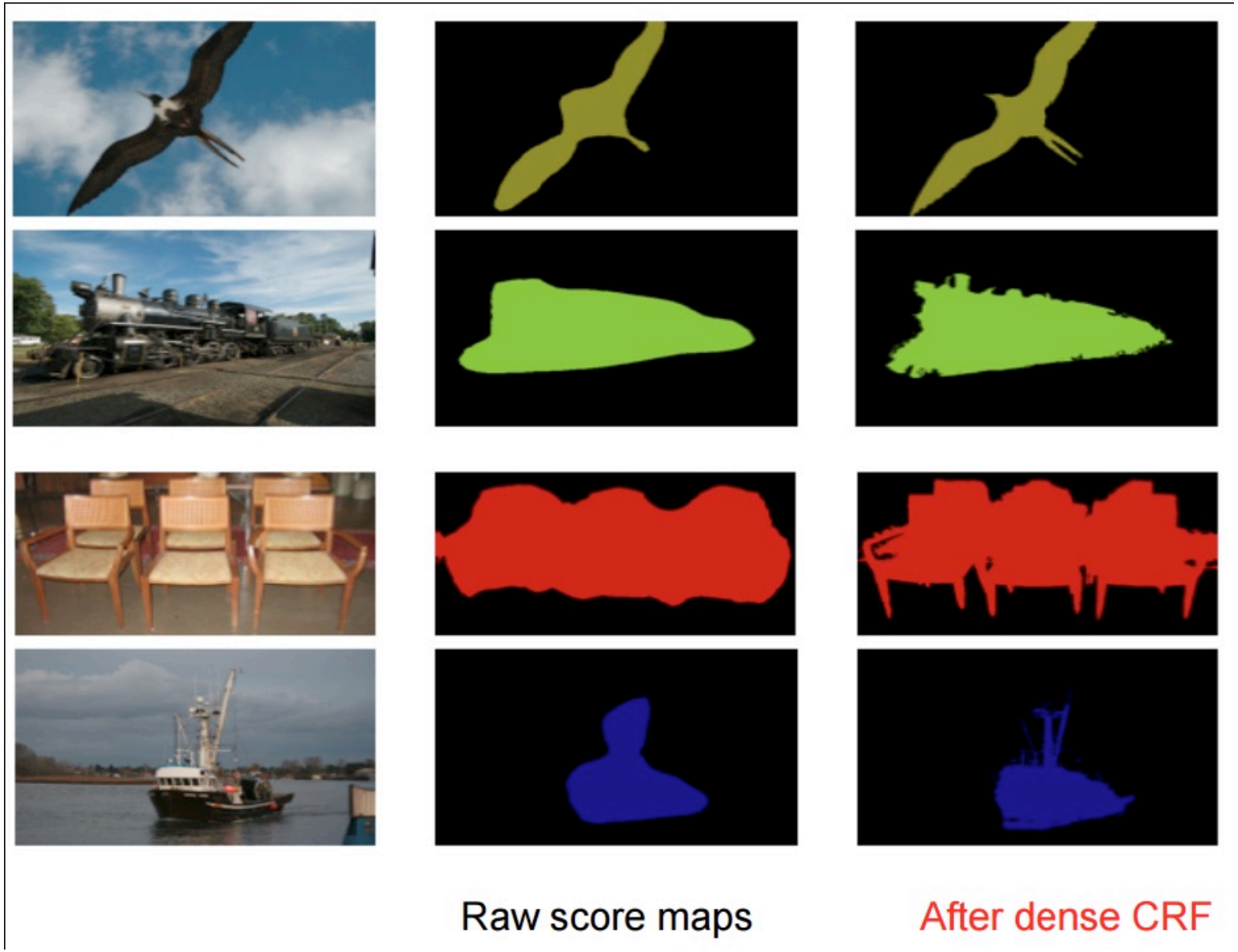
DeepLab

Воспользуемся билинейной интерполяцией и Dense CRF для повышения разрешения

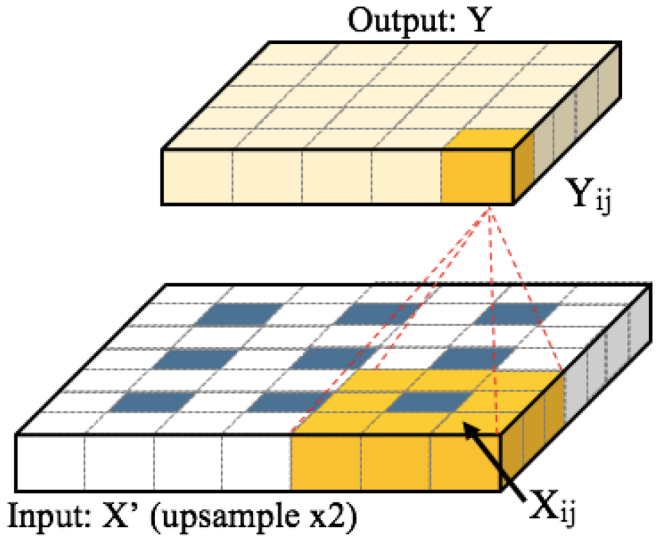


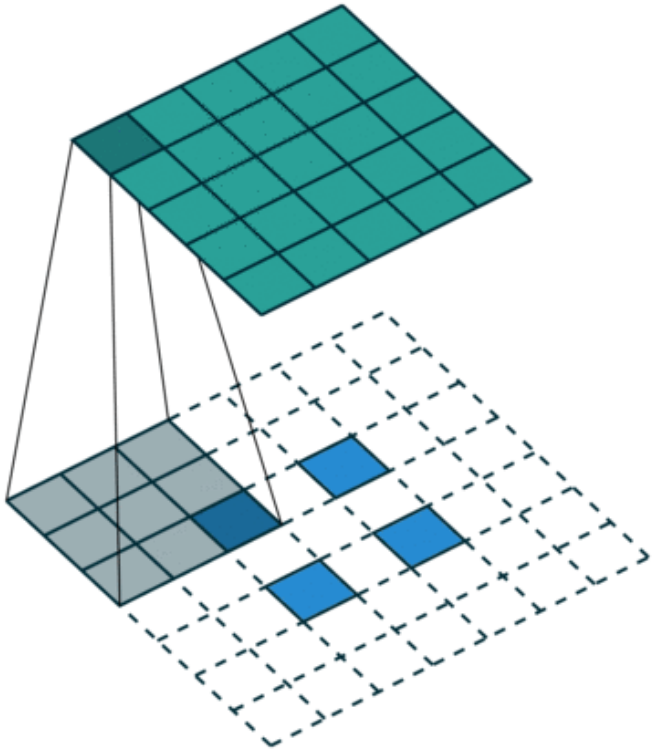
Chen et al. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. TPAMI 2016

Результаты



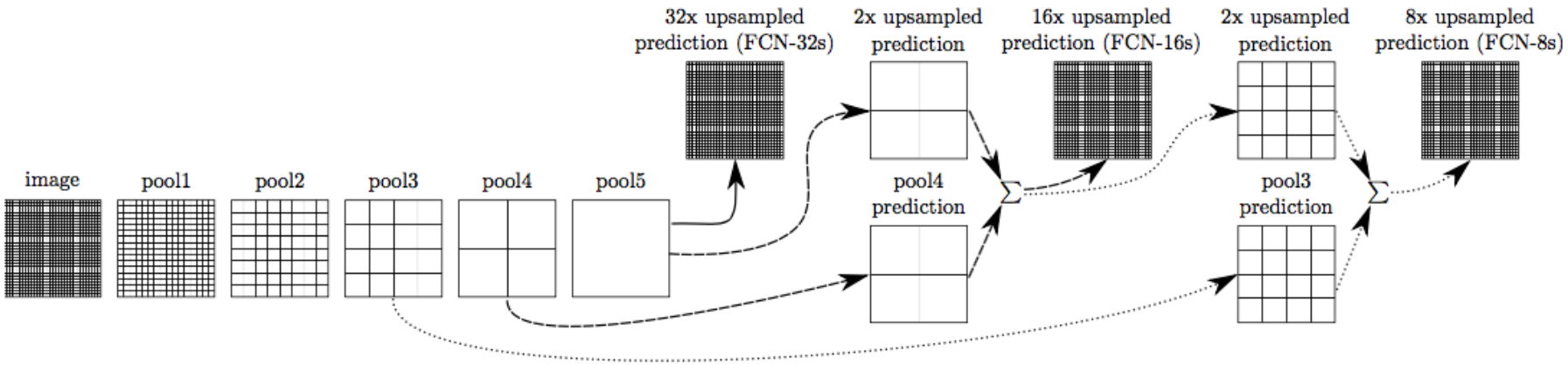
Transposed convolution

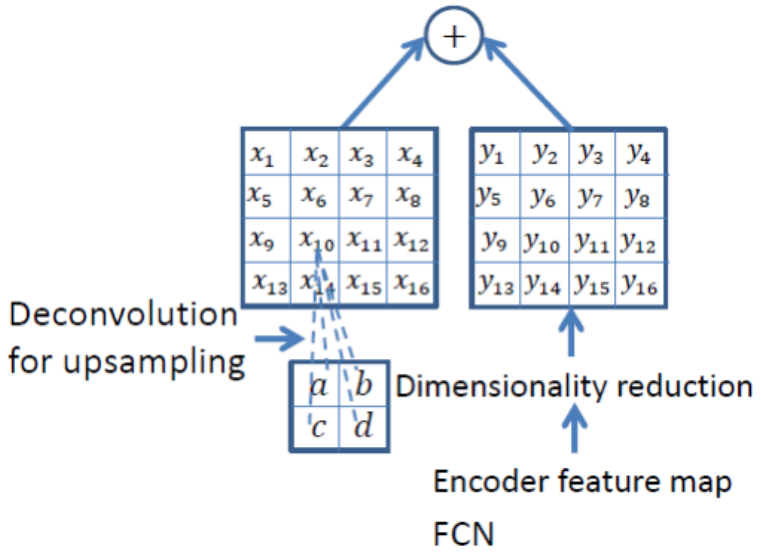




Можем увеличить разрешение картинки, расставив пиксели редко, и затем интерполируя значения в промежутках с помощью свёртки

Fully convolutional networks

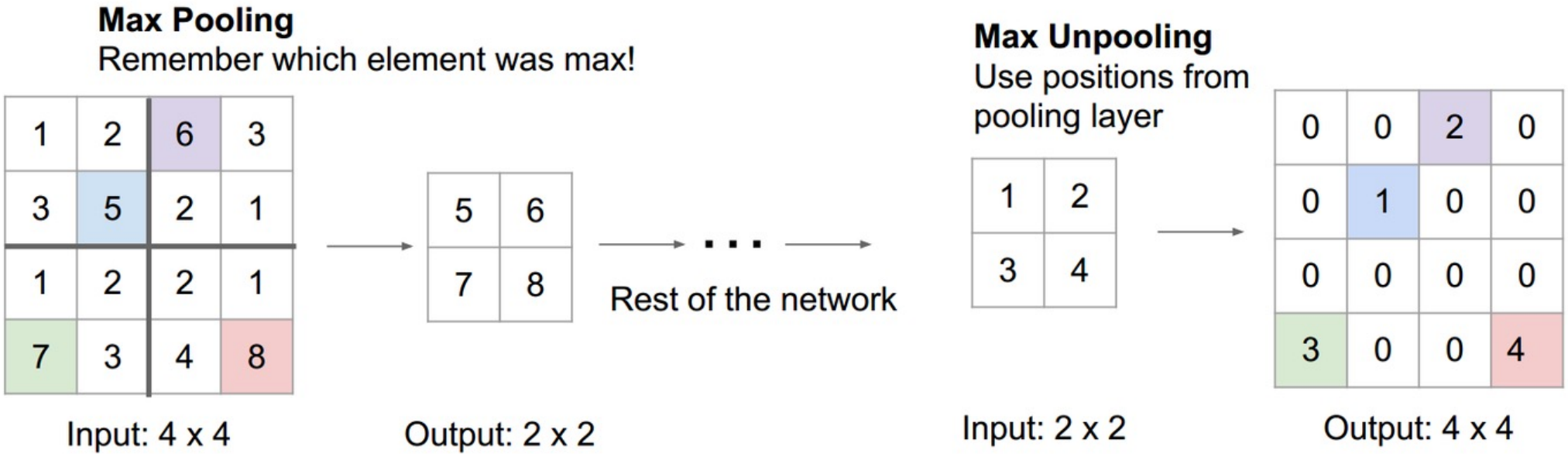




- CNN признаки низкого разрешения на последнем уровне (32x уменьшение)
- Несколько этапов повышения разрешения через “deconvolutional filters”
- Выход уровня повышения разрешения складывается с признаками соответствующего уровня кодирования

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. CoRR, abs/1411.4038, 2014.

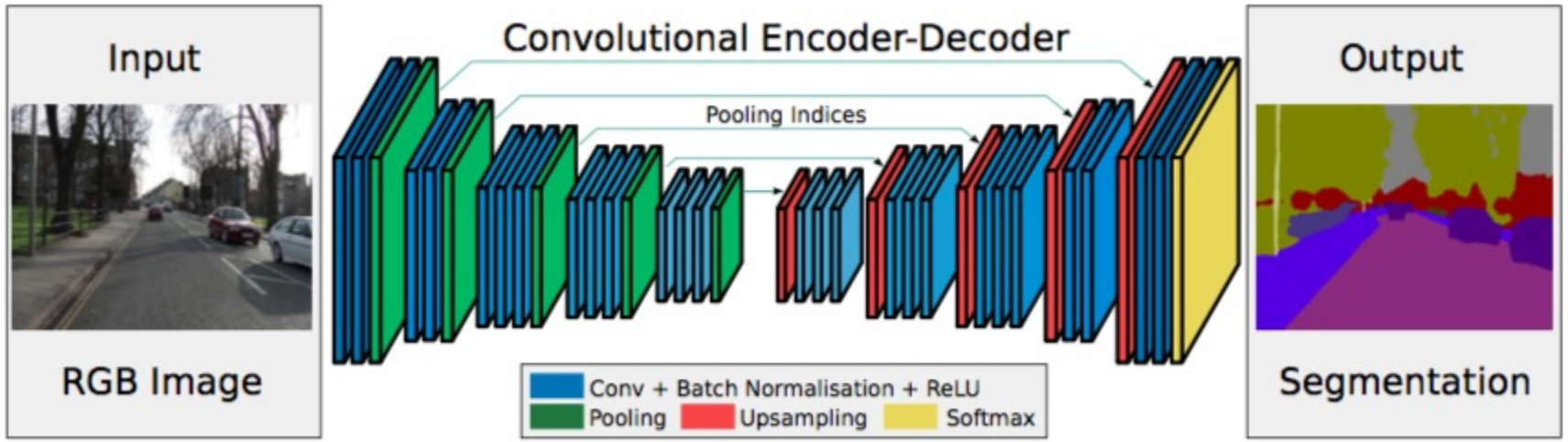
Max Unpooling

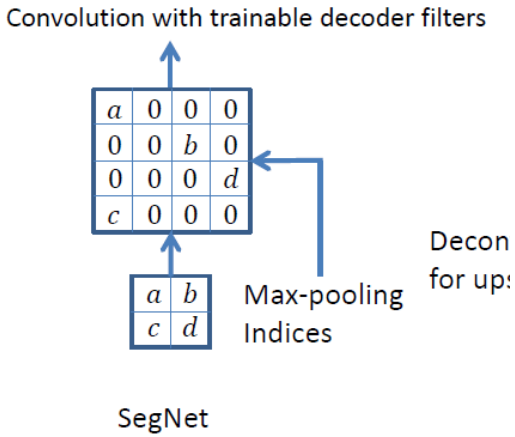


- Сохраняем индексы каждого max-pooling слоя
- Про повышении разрешения делаем так:
  - Копируем значения из выхода max-pooling слоя с учётом запомненный индексов
  - Применяем обученные свёртки для сглаживания

Source: https://tariq-hasan.github.io/concepts/computer-vision-semantic-segmentation/

Encoder-Decoder with Max Unpooling

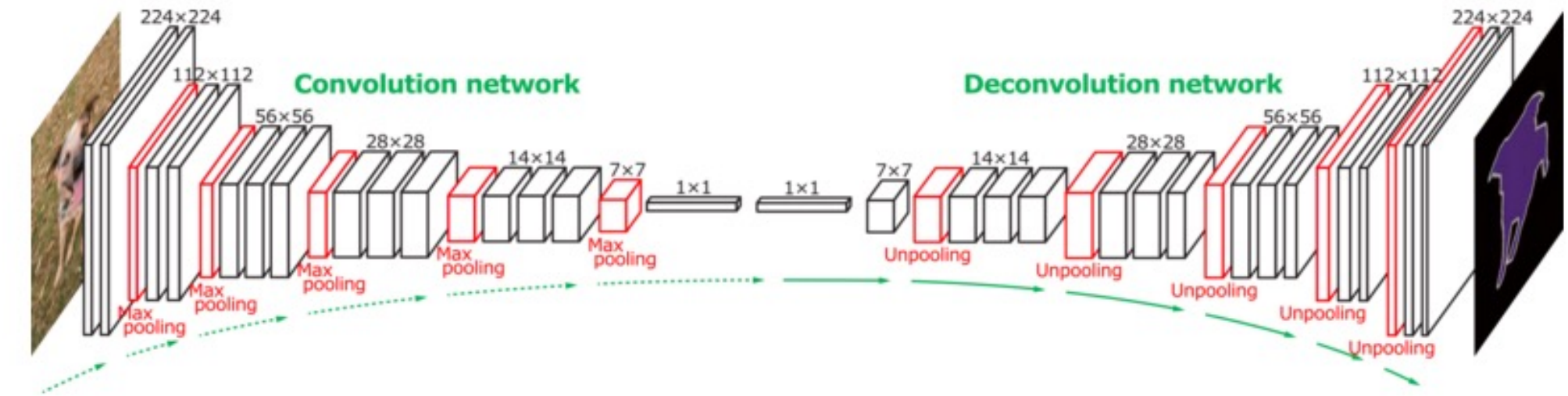




- Меняем схему «повышения разрешения»
- Сохраняем индексы каждого max-pooling слоя
- Про повышении разрешения делаем так:
  - Копируем значения из выхода max-pooling слоя с учётом запомненный индексов
  - Применяем обученные свёртки для сглаживания
- Делаем в несколько этапов до исходного разрешения

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. CoRR, abs/1511.00561, 2015.

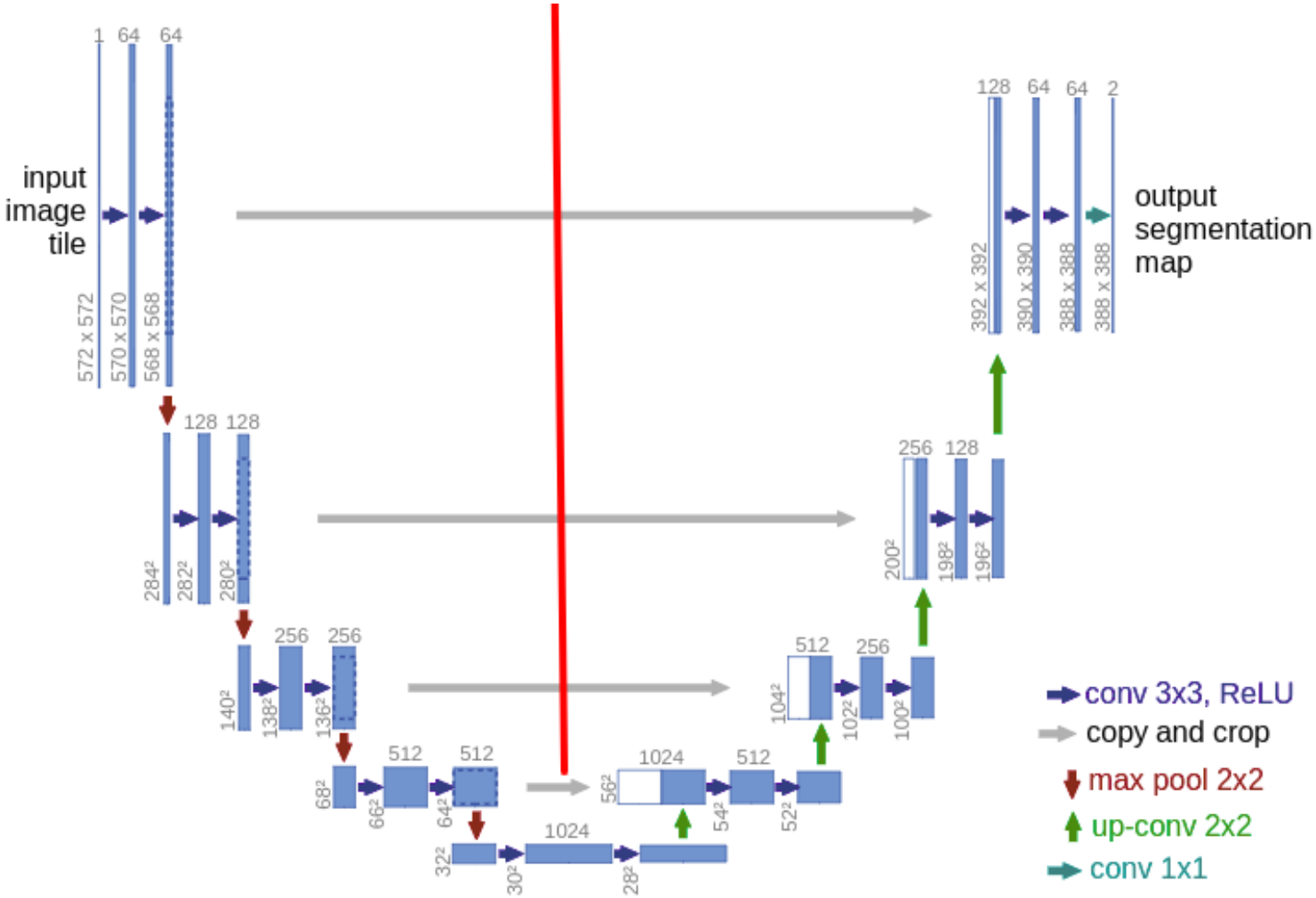
Deconvolutional network



- По всему изображению строим вектор признаков 1x1
- Используя сохранённые max-pooling индексы повышаем разрешение до исходного, предсказывая карту разметки

Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. ICCV 2015

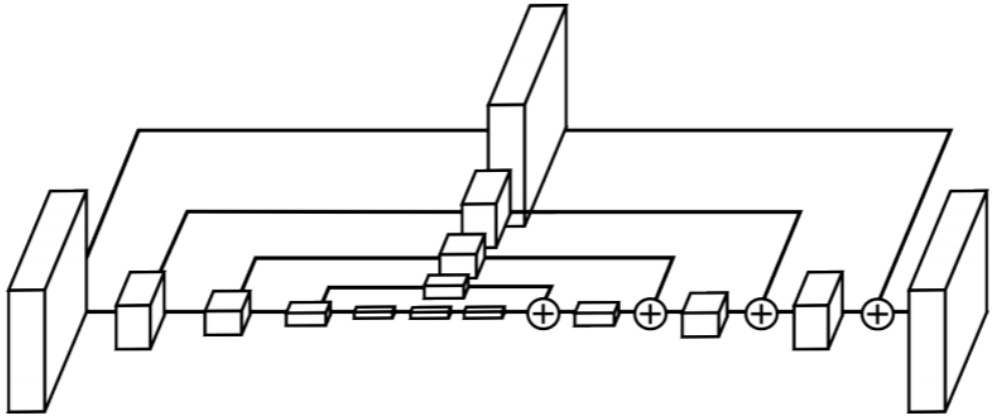
U-Net



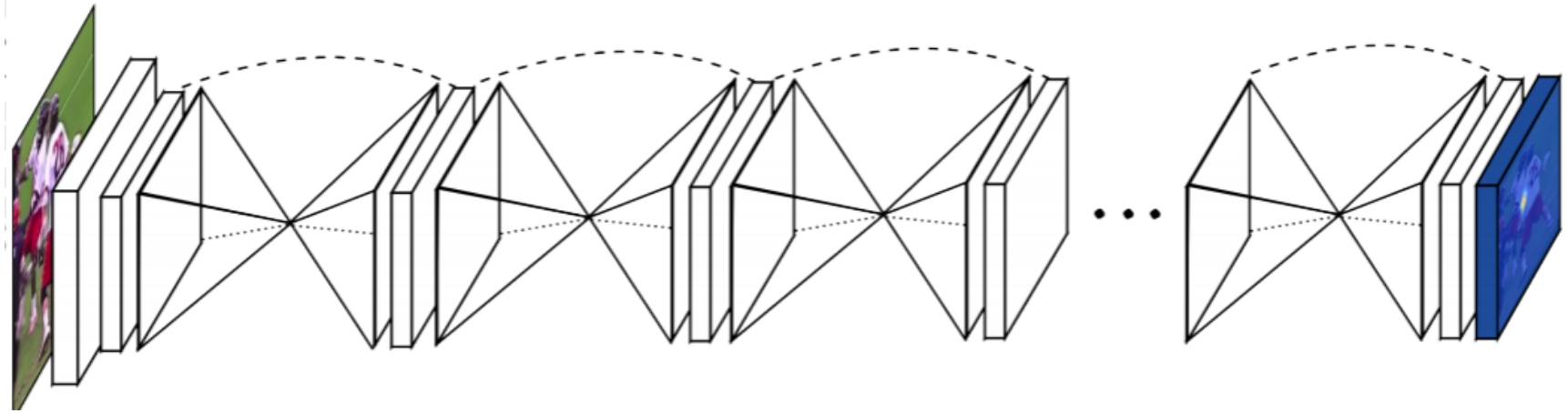
Добавим обходные пути (skip connections)

Ronneberger et al. U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI 2015

Hourglass & Stacked Hourglass



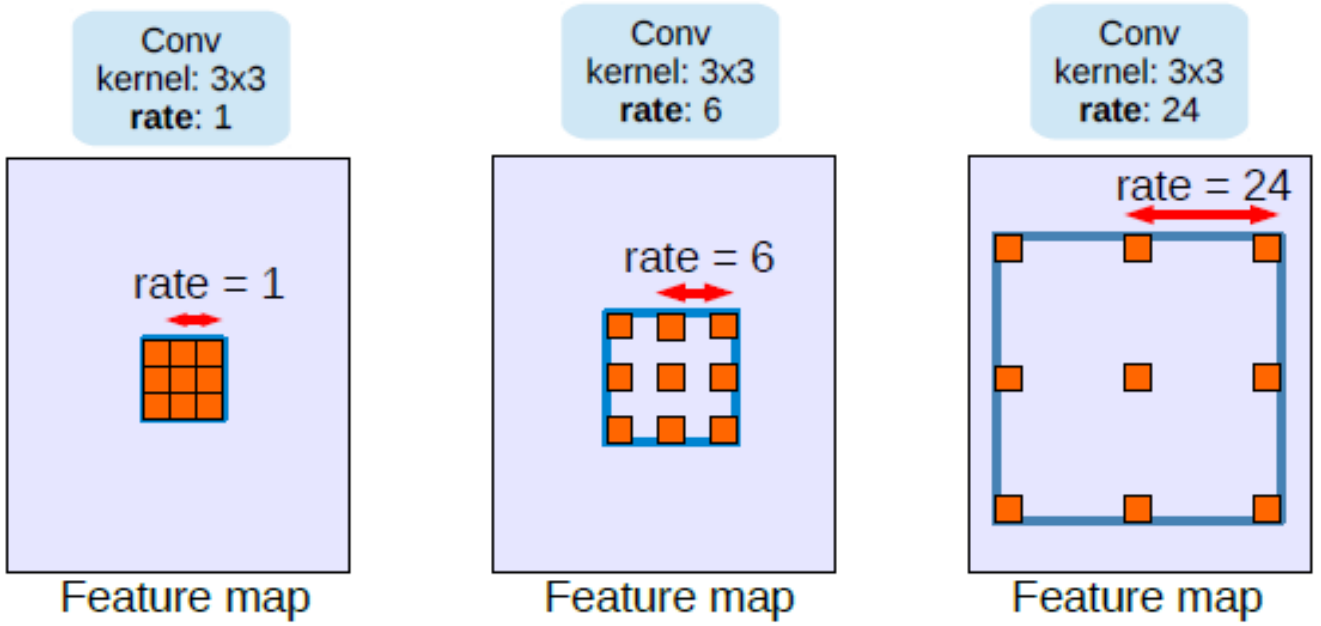
Навесим на обходные пути дополнительные блоки для обработки информации



Объединим модули в цепочку (каскад)

Newell et al. [Stacked Hourglass Networks for Human Pose Estimation](). ECCV 2016

Atrous Convolution

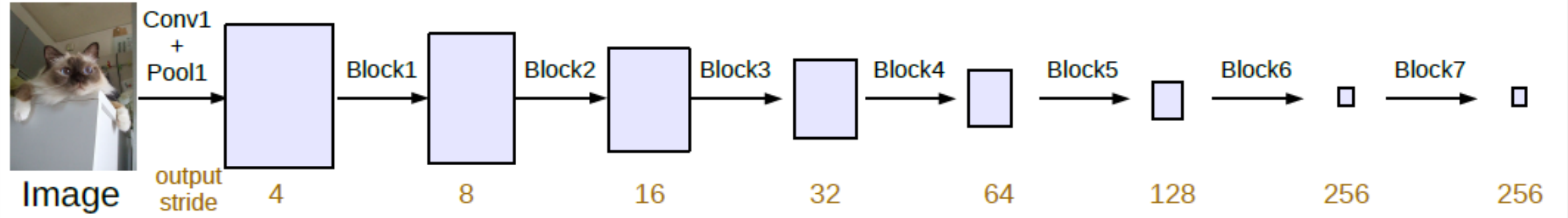


$$y[i] = \sum_{k} x[i + r \cdot k] w[k]$$

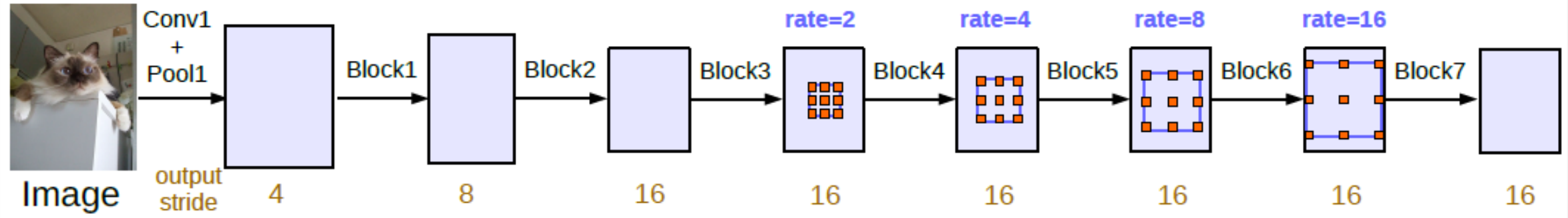
Мы применяем 3x3 свёртку, но отсчёты берём с шагом r

Chen et al. Rethinking Atrous Convolution for Semantic Image Segmentation. 2017

Going Deeper with atrous convolutions



(a) Going deeper without atrous convolution.

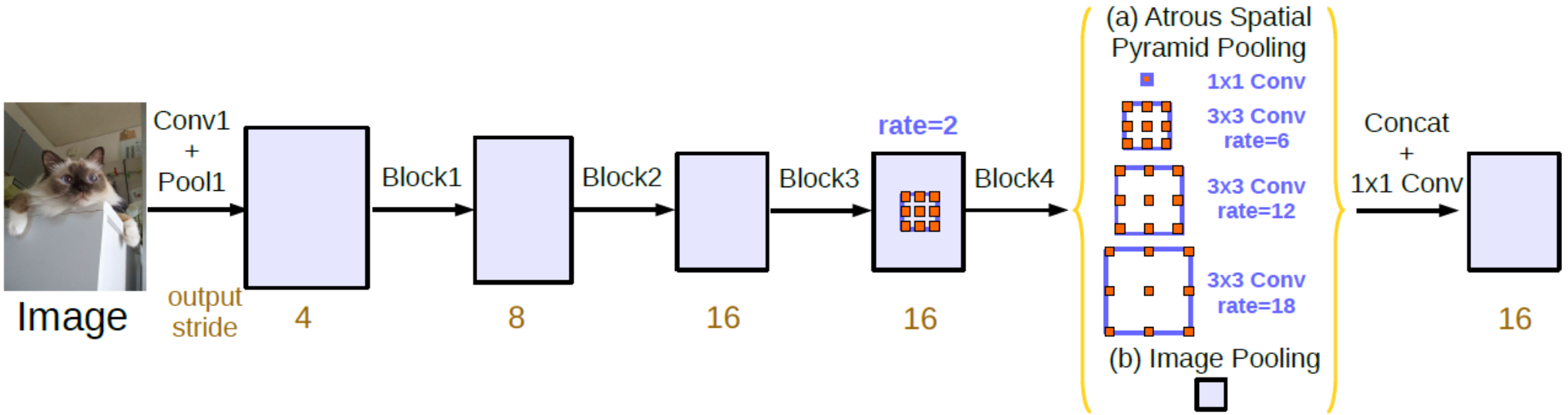


(b) Going deeper with atrous convolution. Atrous convolution with $rate > 1$ is applied after block3 when $output\_stride = 16$.

Chen et al. Rethinking Atrous Convolution for Semantic Image Segmentation. 2017

ASPP

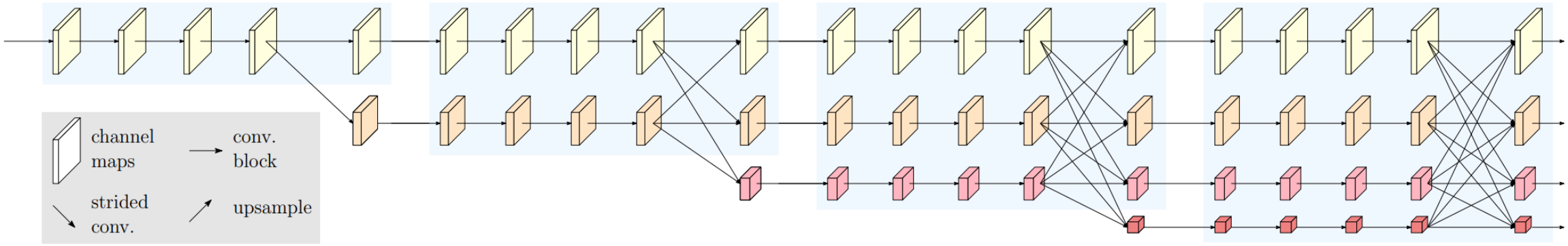
- Ключевой элемент DeepLab, начиная с v2 (сейчас версия 3+)



- Строим «пирамиду», собирая признаки с разных масштабов.

Chen et al. Rethinking Atrous Convolution for Semantic Image Segmentation. 2017

HRNet



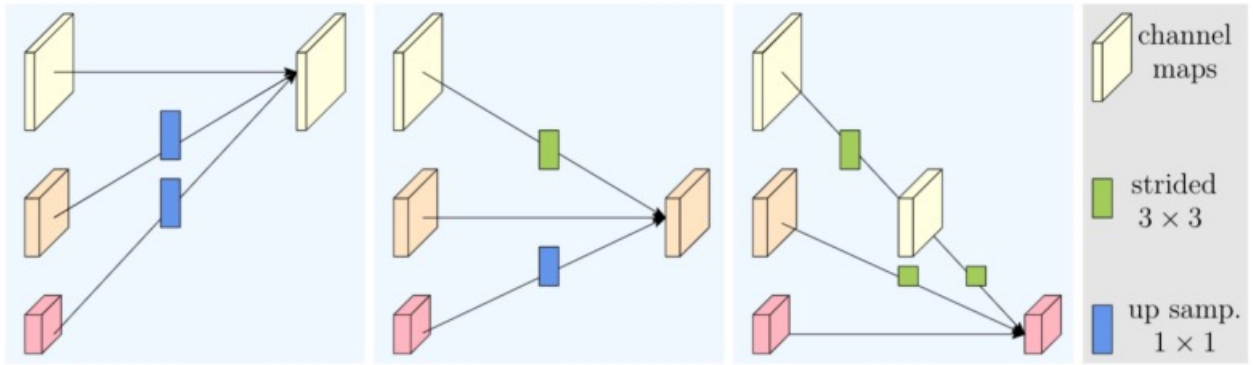


Fig. 3. Illustrating how the fusion module aggregates the information for high, medium and low resolutions from left to right, respectively. Right legend: strided 3×3 = stride-2 3×3 convolution, up samp. 1×1 = bilinear upsampling followed by a 1×1 convolution.

Wang et al. Deep High-Resolution Representation Learning for Visual Recognition. TPAMI 2020

Использование HRNet

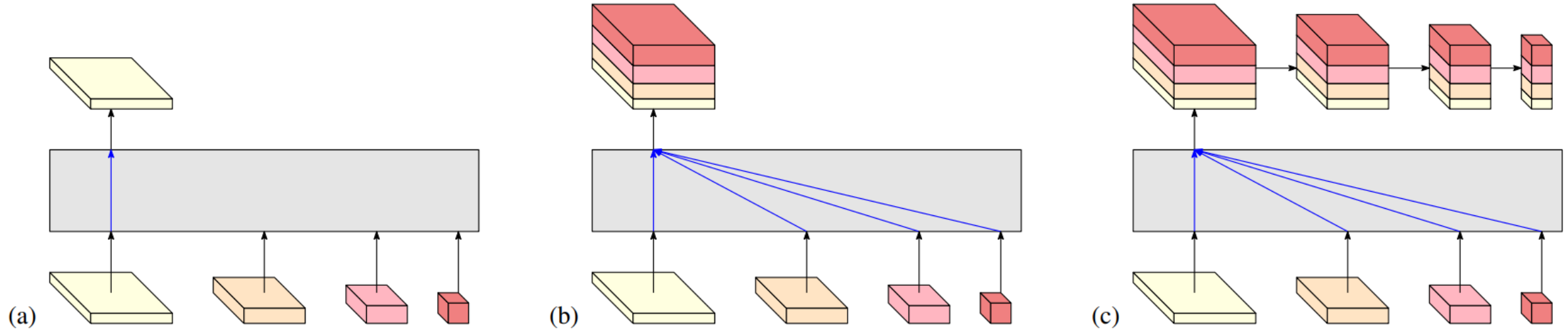
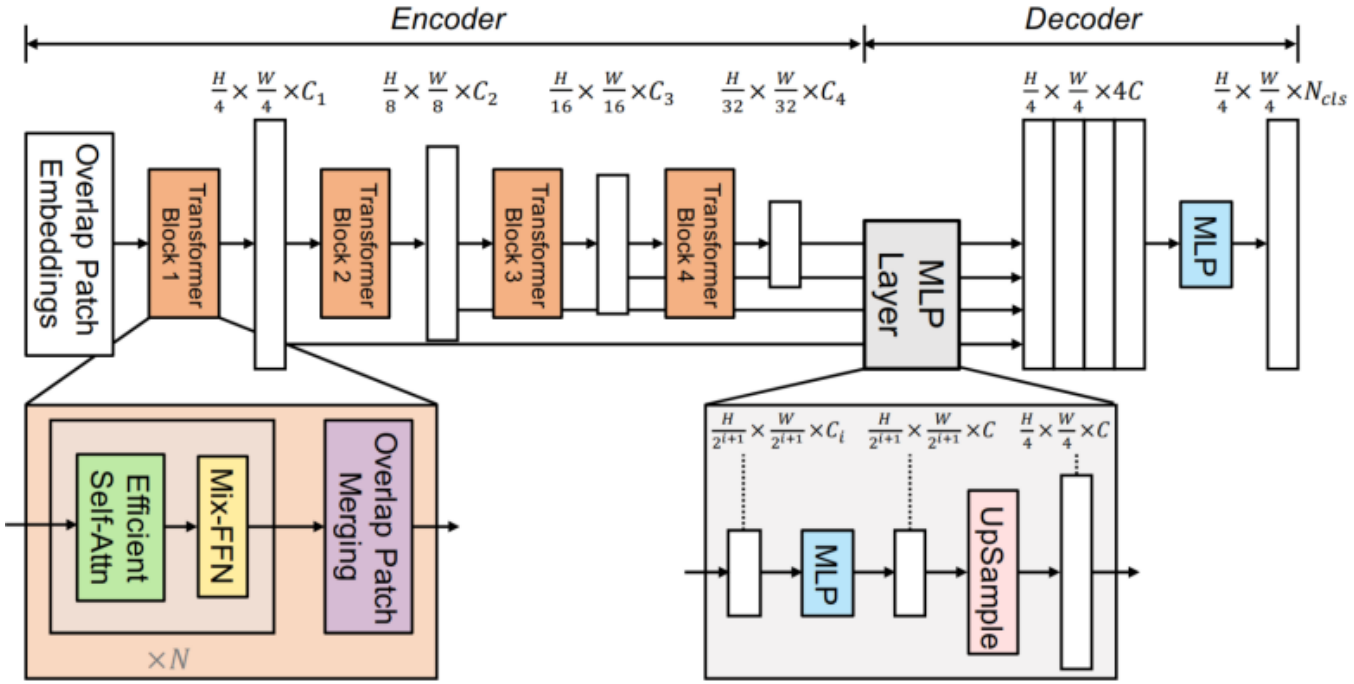


Figure 3. (a) The high-resolution representation proposed in [91] (HRNetV1); (b) Concatenating the (upsampled) representations that are from all the resolutions for semantic segmentation and facial landmark detection (HRNetV2); (c) A feature pyramid formed over (b) for object detection (HRNetV2p). The four-resolution representations at the bottom in each sub-figure are outputted from the network in Figure 1, and the gray box indicates how the output representation is obtained from the input four-resolution representations.

Wang et al. Deep High-Resolution Representation Learning for Visual Recognition. TPAMI 2020

SegFormer



Efficient SA:

$$SA = softmax(qk^T/\sqrt{D_h})v$$

$$k = Reshape(\frac{N}{R}, C \cdot R)(k)$$
$$k = Linear(C \cdot R, C)(k)$$

Mix-FFN:

$$MLP(Conv_{3\times3}(MLP(x))) + x$$

- Не используем positional encoding
- Комбинируем self-attention и свёртки 3x3 для учёта пространственной информации

Xie et al. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. NeurIPS 2021

SegFormer

	Output Size	Layer Name	Mix Transformer					
			B0	B1	B2	B3	B4	B5
Stage 1	$\frac{H}{4} \times \frac{W}{4}$	Overlapping Patch Embedding	$K_1 = 7;\ S_1 = 4;\ P_1 = 3$					
			$C_1 = 32$	$C_1 = 64$				
		Transformer Encoder	$R_1 = 8$ $N_1 = 1$ $E_1 = 8$ $L_1 = 2$	$R_1 = 8$ $N_1 = 1$ $E_1 = 8$ $L_1 = 2$	$R_1 = 8$ $N_1 = 1$ $E_1 = 8$ $L_1 = 3$	$R_1 = 8$ $N_1 = 1$ $E_1 = 8$ $L_1 = 3$	$R_1 = 8$ $N_1 = 1$ $E_1 = 8$ $L_1 = 3$	$R_1 = 8$ $N_1 = 1$ $E_1 = 4$ $L_1 = 3$
Stage 2	$\frac{H}{8} \times \frac{W}{8}$	Overlapping Patch Embedding	$K_2 = 3;\ S_2 = 2;\ P_2 = 1$					
			$C_2 = 64$	$C_2 = 128$				
		Transformer Encoder	$R_2 = 4$ $N_2 = 2$ $E_2 = 8$ $L_2 = 2$	$R_2 = 4$ $N_2 = 2$ $E_2 = 8$ $L_2 = 2$	$R_2 = 4$ $N_2 = 2$ $E_2 = 8$ $L_2 = 3$	$R_2 = 4$ $N_2 = 2$ $E_2 = 8$ $L_2 = 3$	$R_2 = 4$ $N_2 = 2$ $E_2 = 8$ $L_2 = 8$	$R_2 = 4$ $N_2 = 2$ $E_2 = 4$ $L_2 = 6$
Stage 3	$\frac{H}{16} \times \frac{W}{16}$	Overlapping Patch Embedding	$K_3 = 3;\ S_3 = 2;\ P_3 = 1$					
			$C_3 = 160$	$C_3 = 320$				
		Transformer Encoder	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 2$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 2$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 6$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 18$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 27$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 40$
Stage 4	$\frac{H}{32} \times \frac{W}{32}$	Overlapping Patch Embedding	$K_4 = 3;\ S_4 = 2;\ P_4 = 1$					
			$C_4 = 256$	$C_4 = 512$				
		Transformer Encoder	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 2$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 2$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 3$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 3$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 3$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 3$

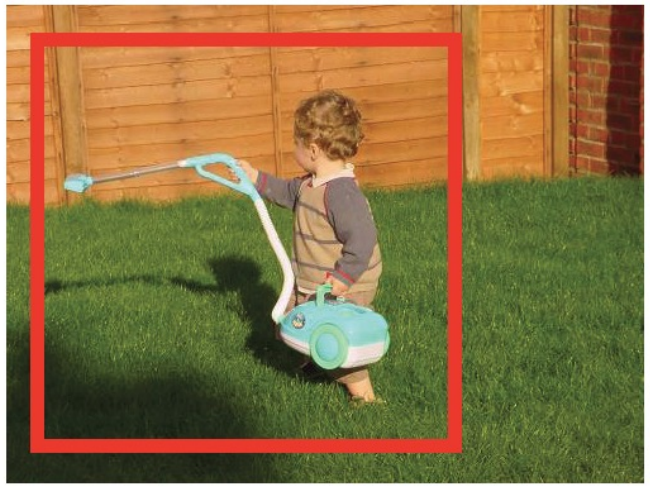
Резюме сегментационных моделей

- Основной подход для построения сегментационных сетей – Encoder-Decoder
- В качестве Encoder мы берём любую классификационную сеть
- Decoder состоит из блоков, между которыми повышается разрешение
- Есть несколько способов повышения разрешения
- Обходные пути позволяют учитывать признаки высокого разрешения из Encoder
- Трансформерные архитектуры обеспечивают контекст за счёт self-attention, и мы можем декодер упростить

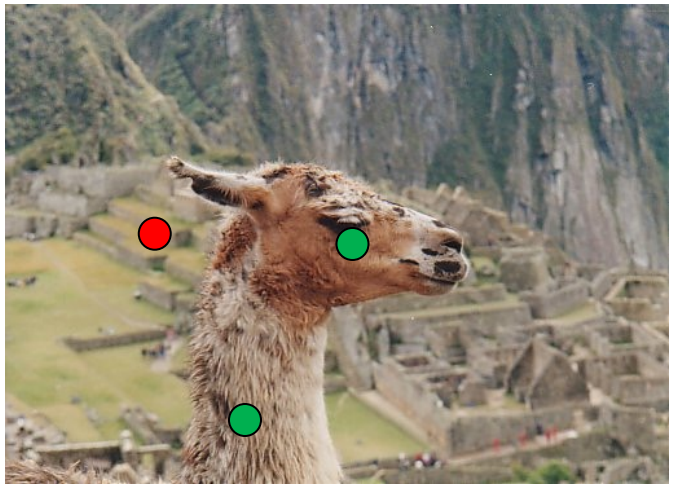
План лекции

1. Такая разная сегментация
2. Пересегментация
3. Семантическая сегментация
4. Интерактивная сегментация
5. Сегментация экземпляров
6. Паноптическая сегментация
7. Оценка позы человека

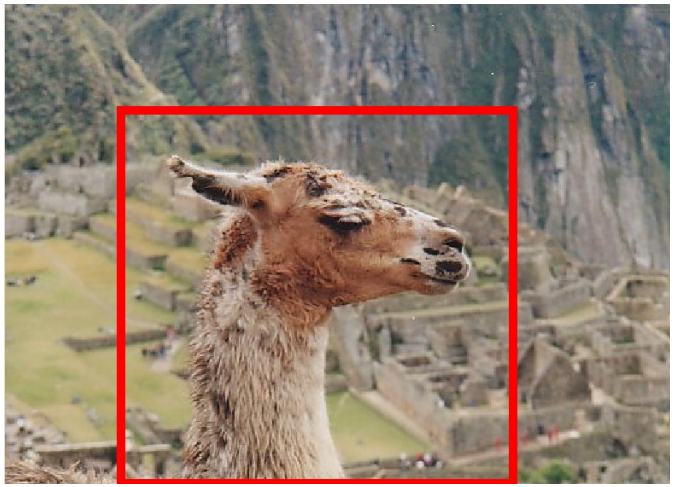
Интерактивная сегментация



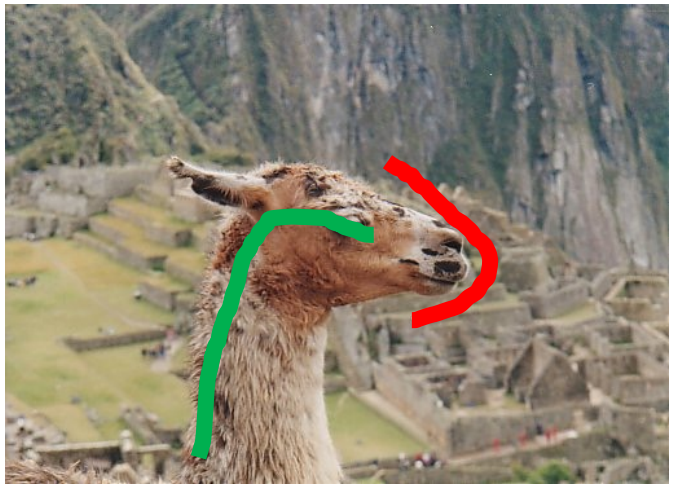
Пользовательский ввод



Клики



Bounding box



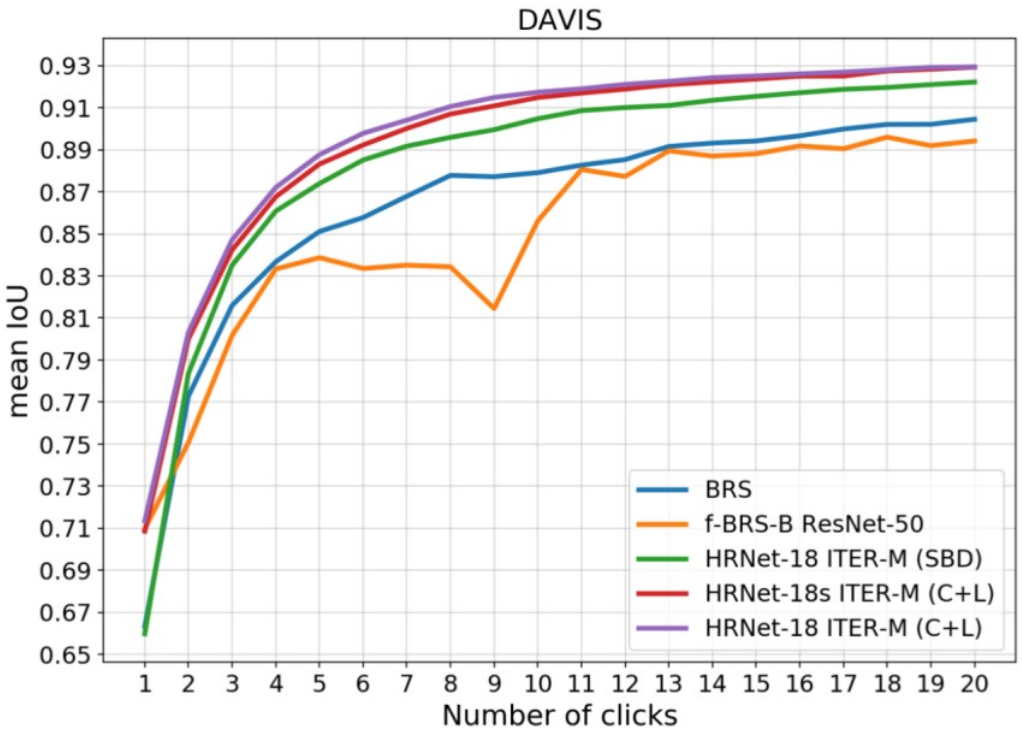
Мазки

Контур

Интерактивный режим!

Датасеты и метрики

Berkeley — 50 images
GrabCut — 100 images
DAVIS — 345 images
SBD — 2857 images, 6671 masks



- NoC@0.9 – Сколько кликов нужно сделать, чтобы добиться уровня IoU 0.9
- Обычно ограничивают максимум 20

Deep GrabCut

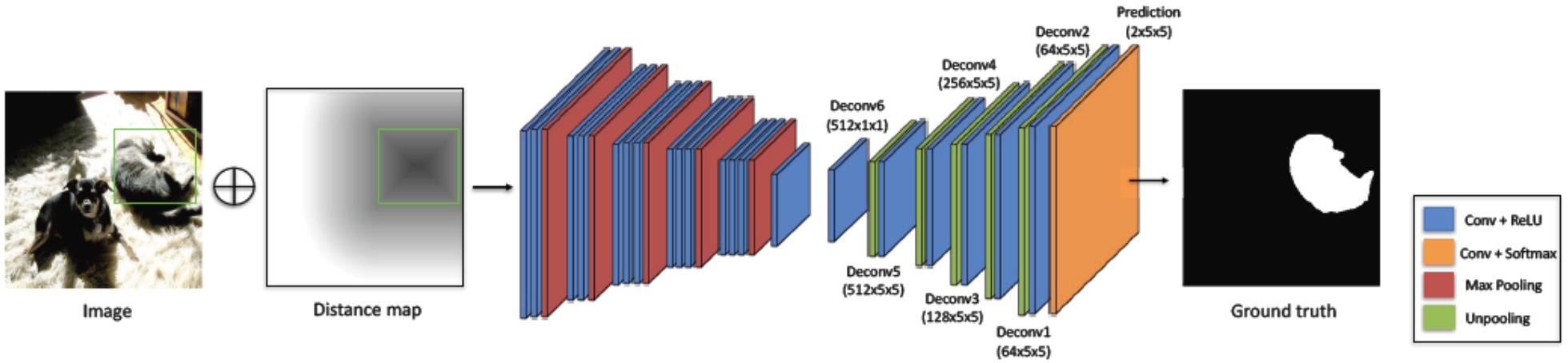
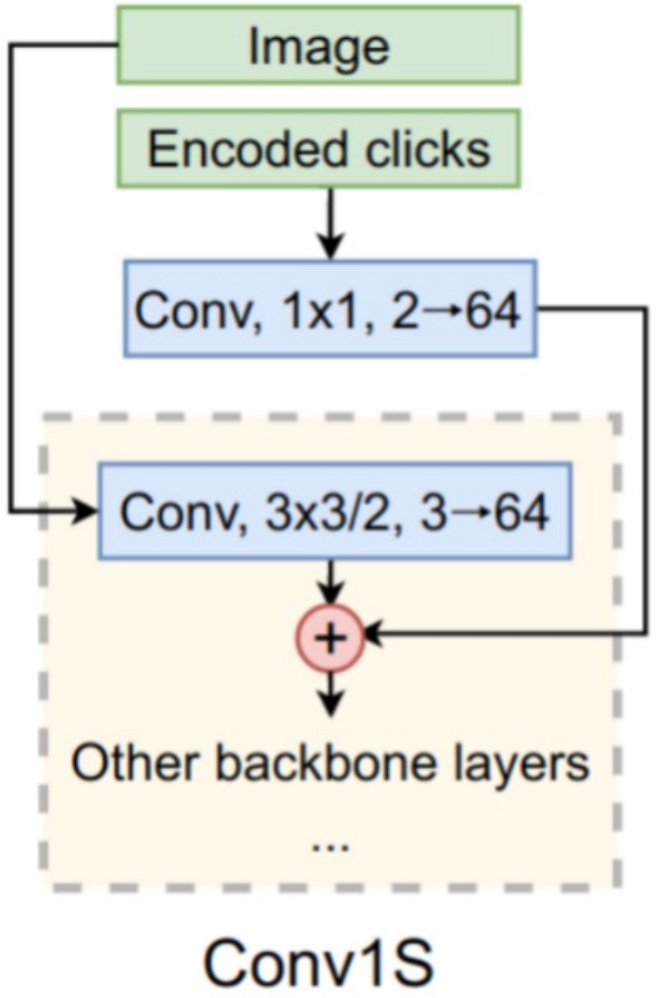


Figure 2: The framework of our segmentation model. The rectangle is indicated in green in the “Image" and “Distance map". The symbol ⊕ denotes the concatenation operation.

Насколько это интерактивный метод?

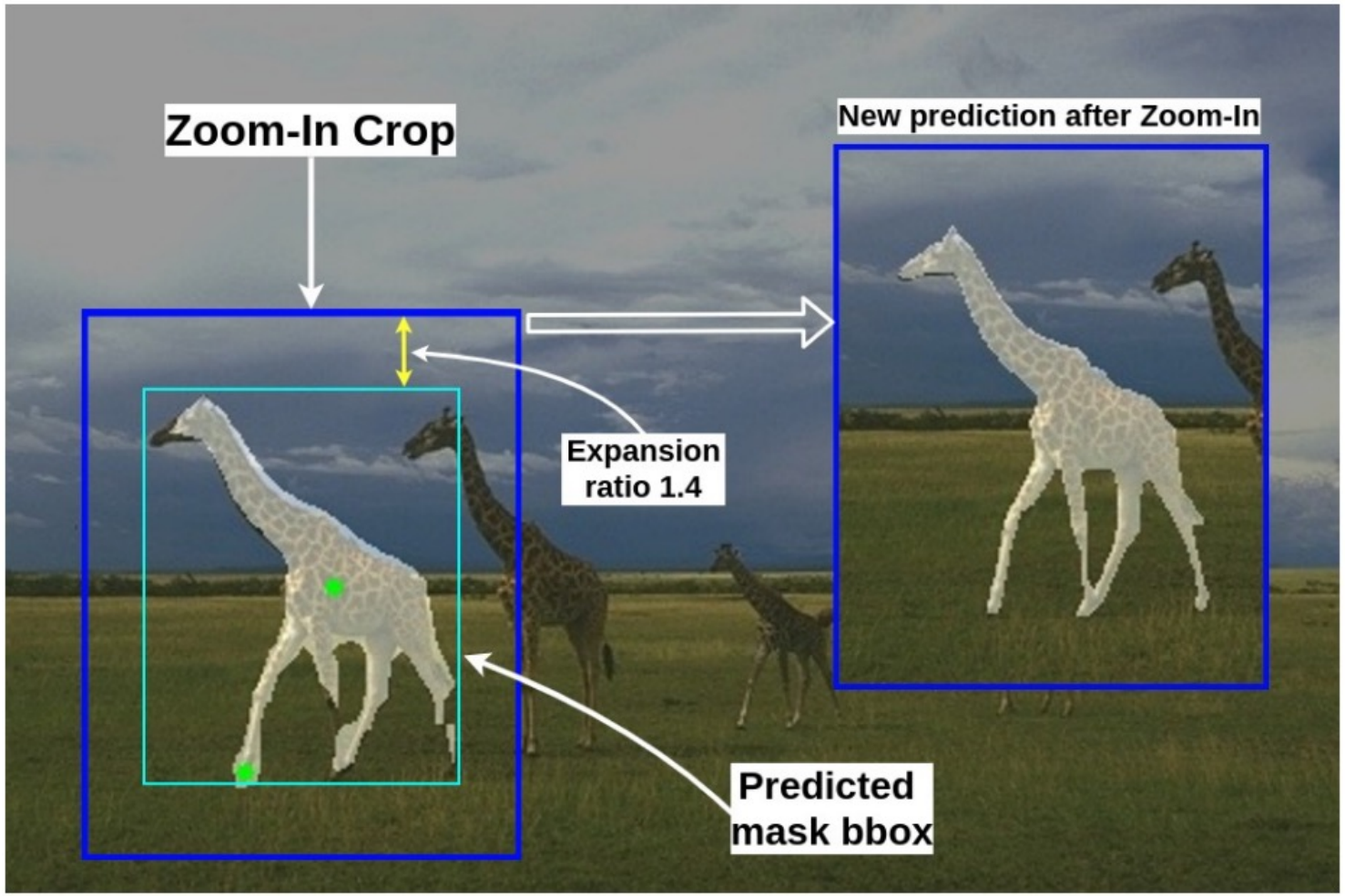
RITM



- Кодирование кликов
- Смешение с признаками изображения
- Итеративное обучение
- Использование маски с прошлого шага
- Использование современных датасетов (COCO+LVIS) для обучения

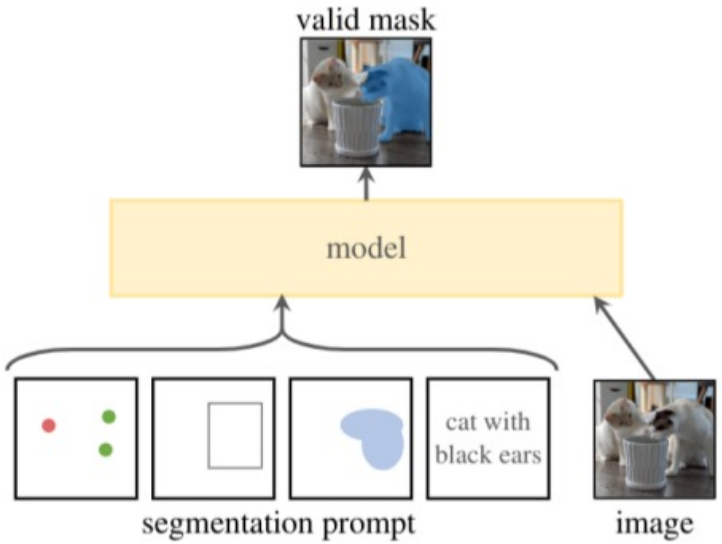
K.Sofiiuk, I.Petrov, A.Konushin Reviving Iterative Training with Mask Guidance for Interactive Segmentation. ICIP 2022

Zoom-In

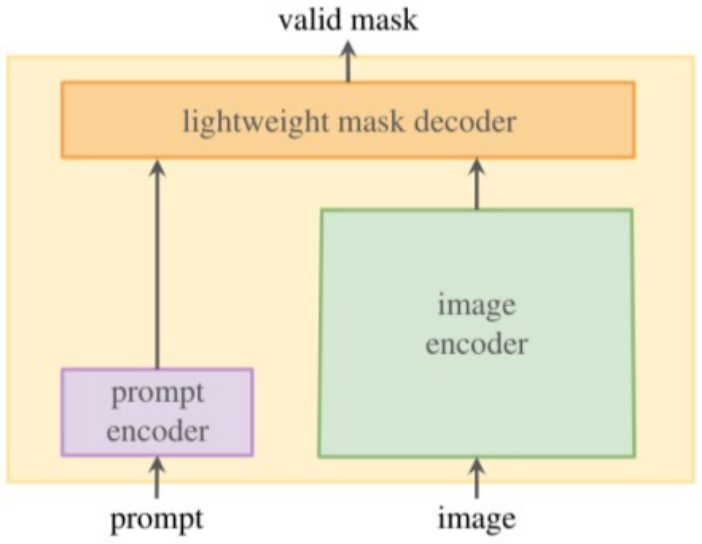


Sofiiuk et al. f-BRS: Rethinking Backpropagating Refinement for Interactive Segmentation. CVPR 2020

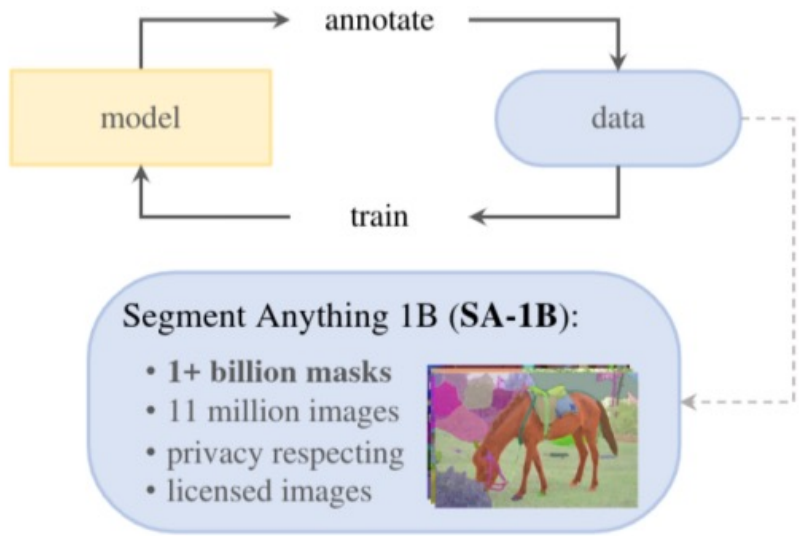
Segment Anything Model (SAM)



(a) **Task**: promptable segmentation



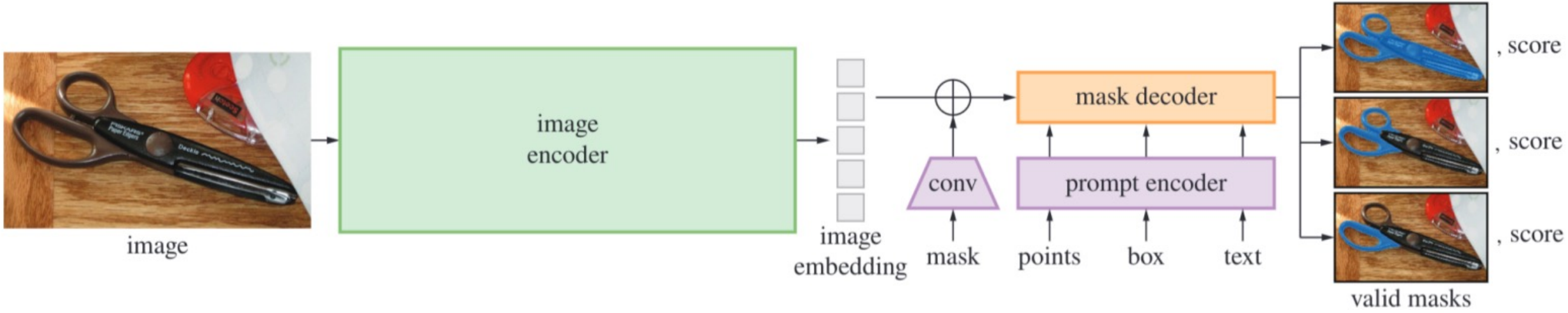
(b) **Model**: Segment Anything Model (**SAM**)



(c) **Data**: data engine (top) & dataset (bottom)

Kirillov et al. Segment Anything. arXiv:2304.02643

SegmentAnything



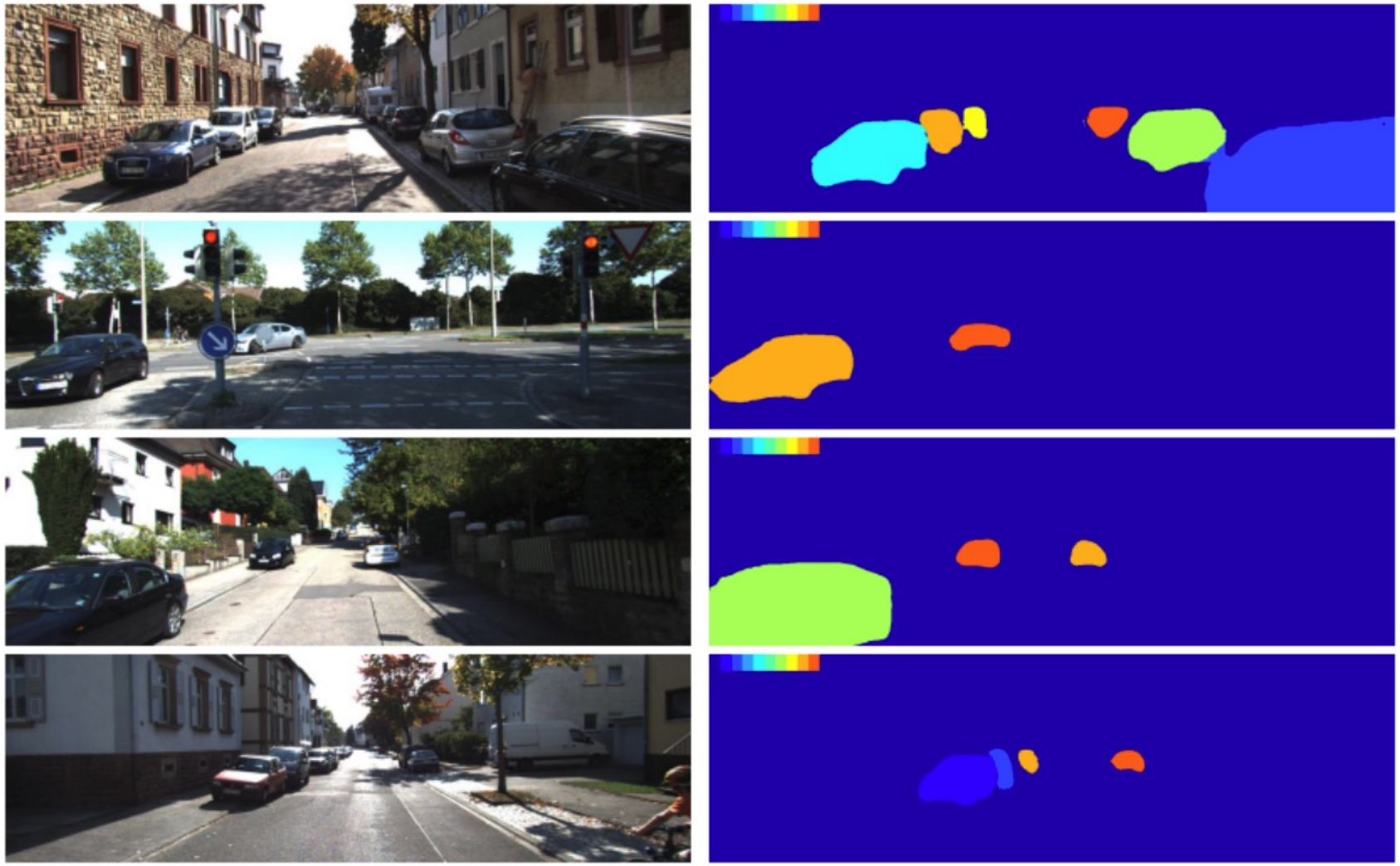
- Используем ViT для кодирования изображения
- “Amortized inference” – признаки изображения считаем через encoder один раз, а интерактивную информацию и маску подмешиваем только перед легковесным декодером

Kirillov et al. Segment Anything. arXiv:2304.02643

План лекции

1. Такая разная сегментация
2. Пересегментация
3. Семантическая сегментация
4. Интерактивная
5. Сегментация экземпляров
6. Паноптическая сегментация
7. Оценка позы человека

Instance Segmentation



Mask R-CNN для Instance Segmentation

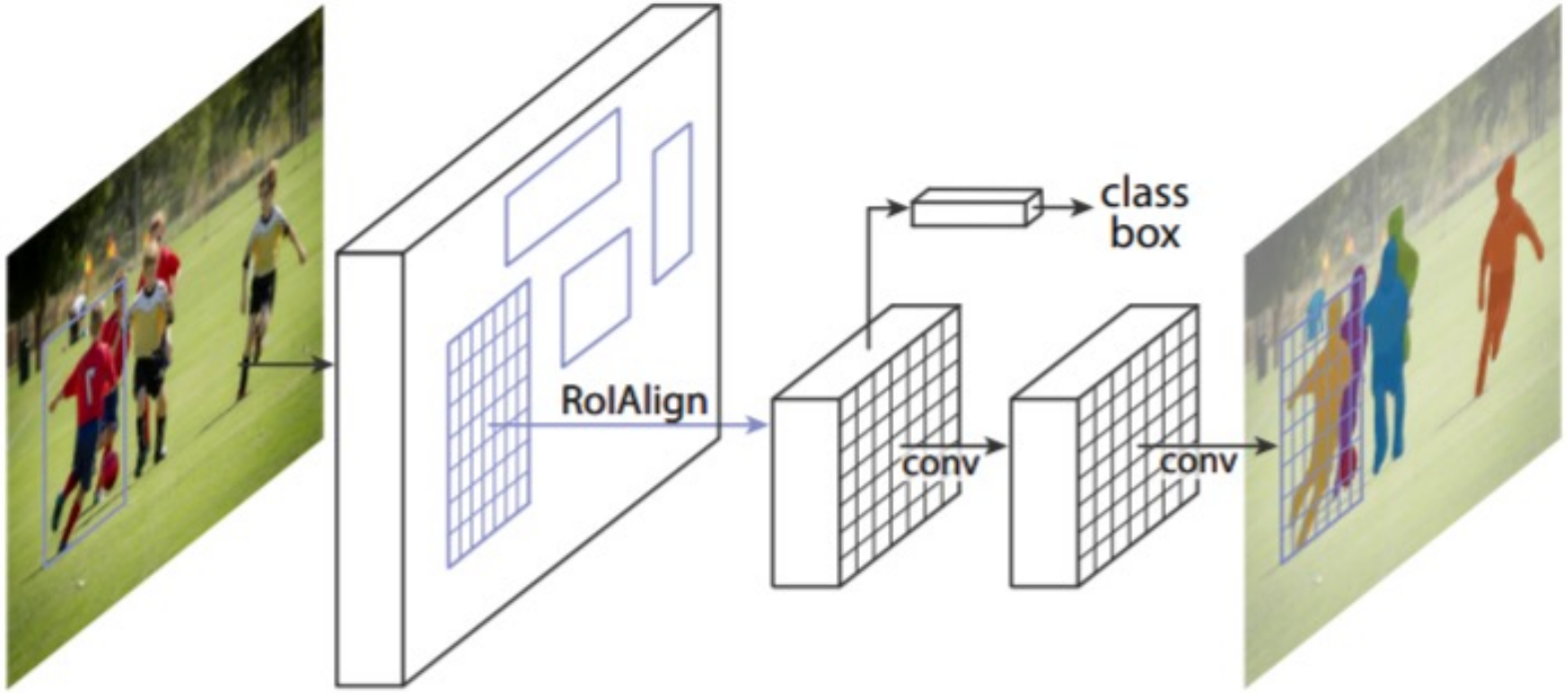


Figure 1. The **Mask R-CNN** framework for instance segmentation.

He et al. Mask R-CNN. ICCV 2017

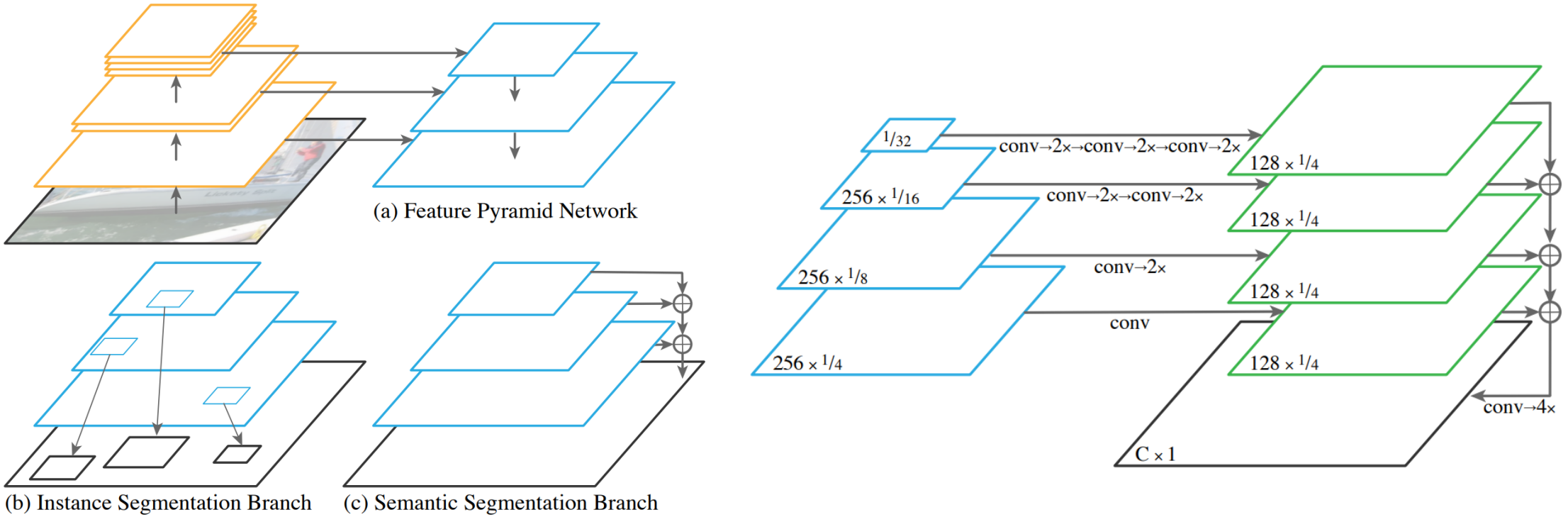
Результаты

Figure 4. More results of **Mask R-CNN** on COCO test images, using ResNet-101-FPN and running at 5 fps, with 35.7 mask AP (Table 1).

План лекции

1. Такая разная сегментация
2. Пересегментация
3. Семантическая сегментация
4. Интерактивная
5. Сегментация экземляров
6. Паноптическая сегментация
7. Оценка позы человека

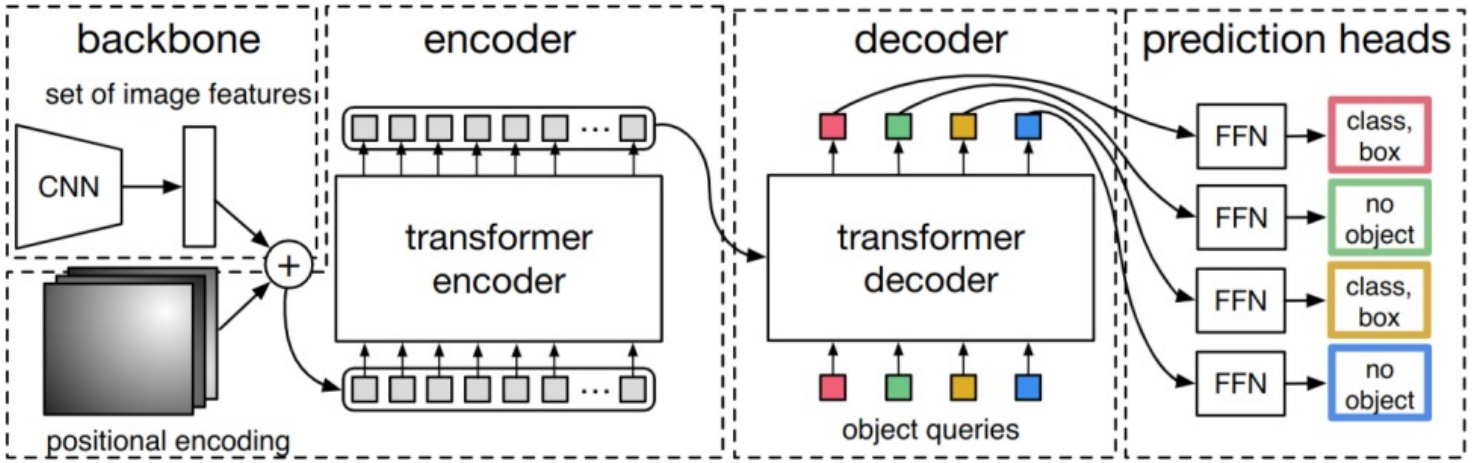
Panoptic Feature Pyramid Networks

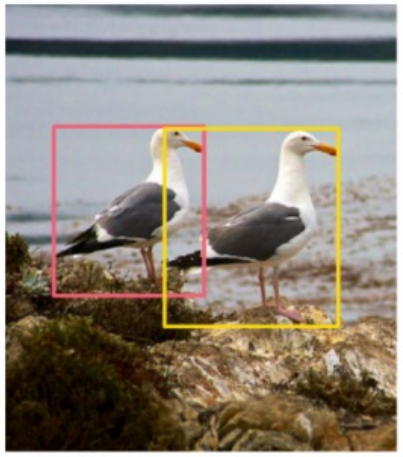


Возьмём идею Mask R-CNN и дополним её веткой семантической сегментации по тем же признакам

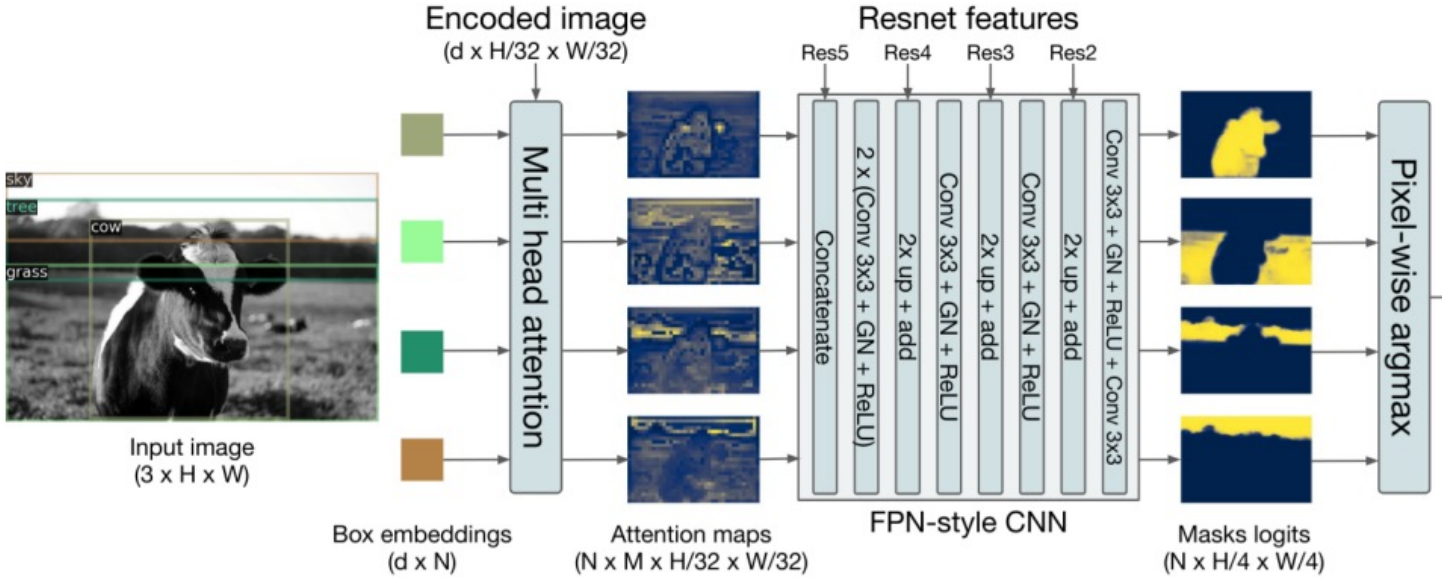
Kirillov et. al. Panoptic Feature Pyramid Networks. CVPR 2019

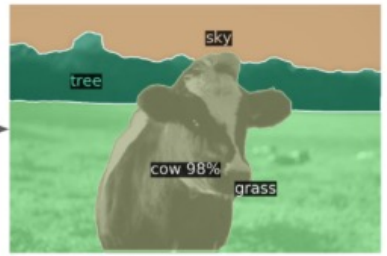
DETR





Дополним список классов stuff, и будем предсказывать bbox этих классов

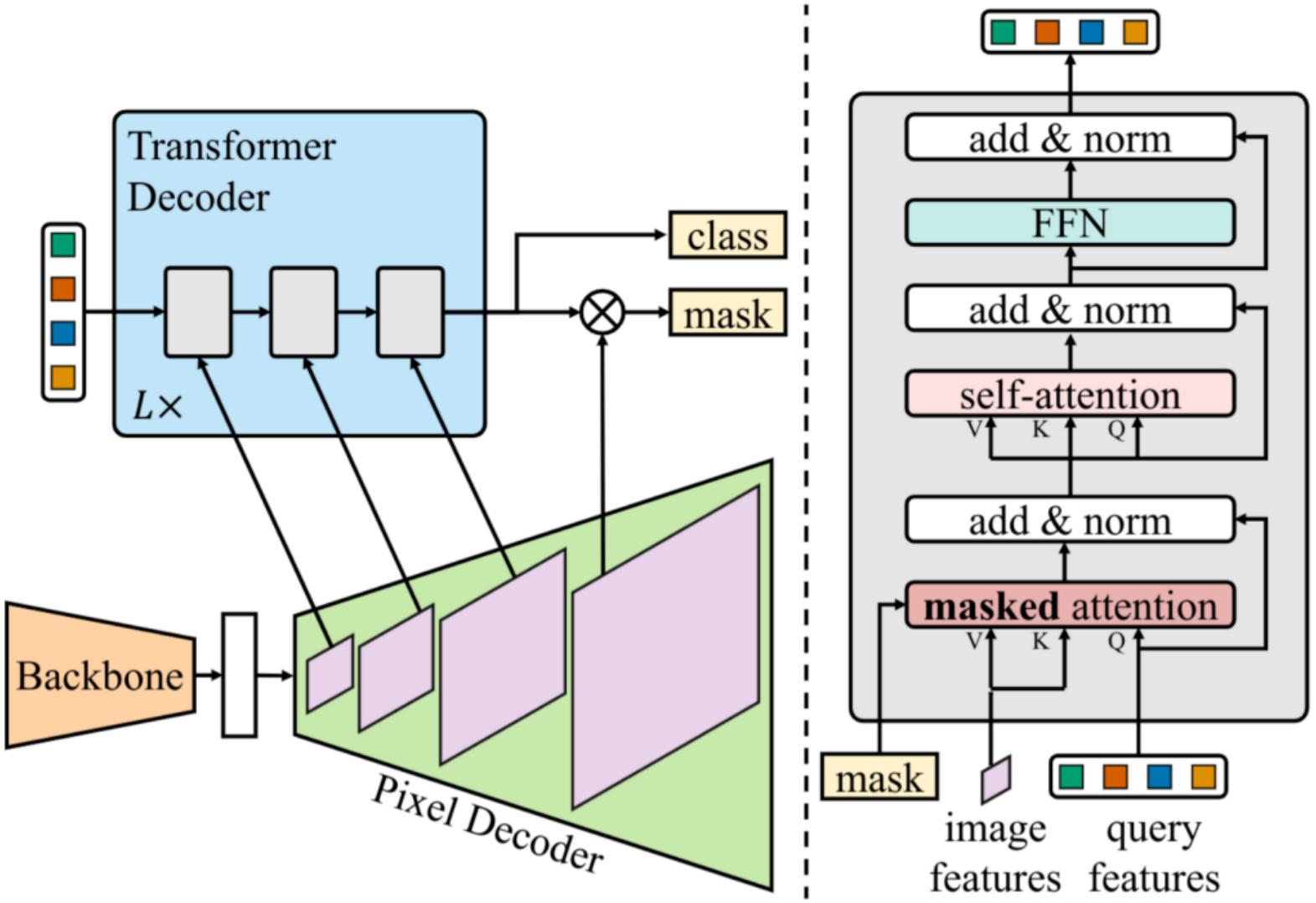




Добавим голову для предсказания масок по bbox предсказаниям stuff классов

Carion et al. End-to-End Object Detection with Transformers. ECCV 2020

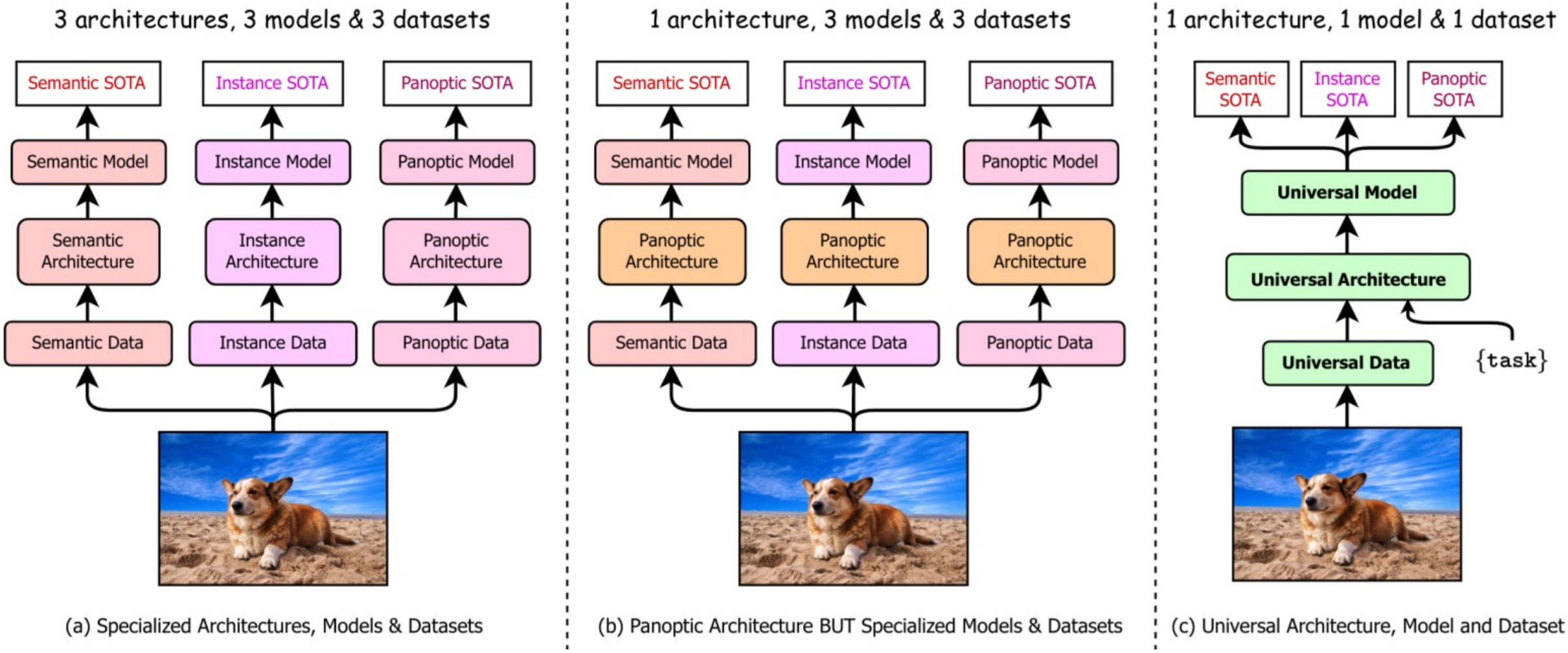
Mask2Former



- Развитие Encoder-Decoder архитектуры + Feature Pyramid
- Backbone + Pixel Decoder + Transformer Decoder
- Можем использовать ResNet или SwinTransformer как backbone
- Простая модель FPN для генерации признаков высокого разрешения для Pixel Decoder
- Добавим голову для предсказания масок по bbox предсказаниям stuff классов
- Masked Attention учитывает только признаки из маски

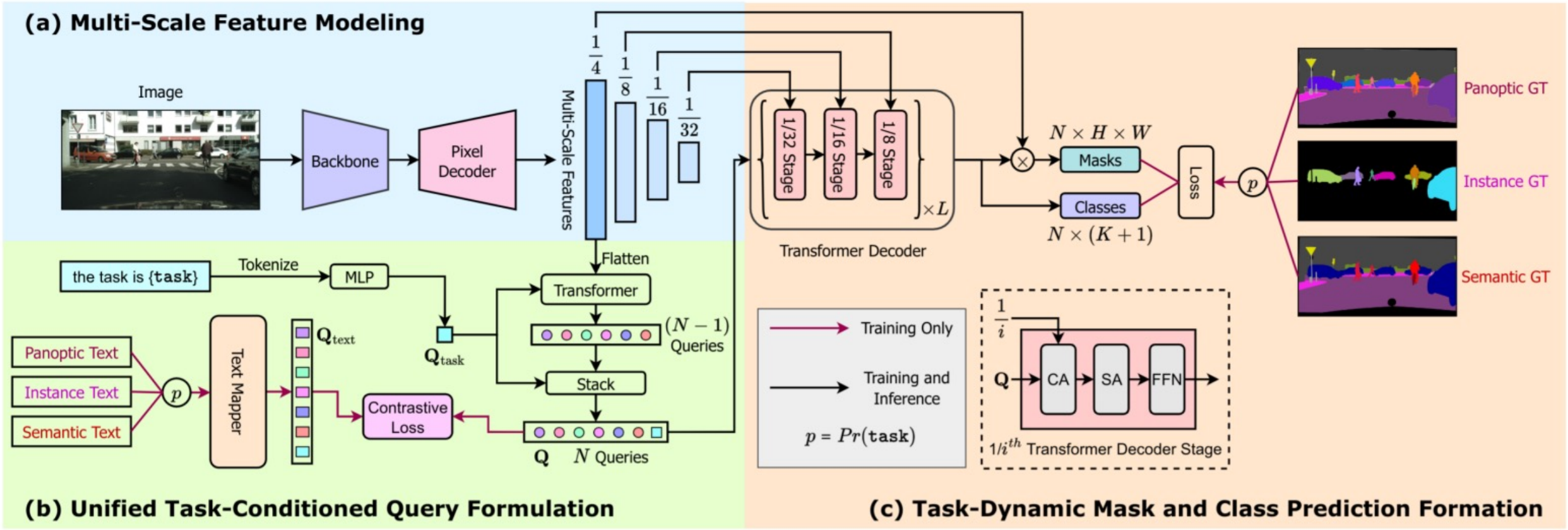
Cheng et al. Masked-attention Mask Transformer for Universal Image Segmentation. CVPR 2022

OneFormer



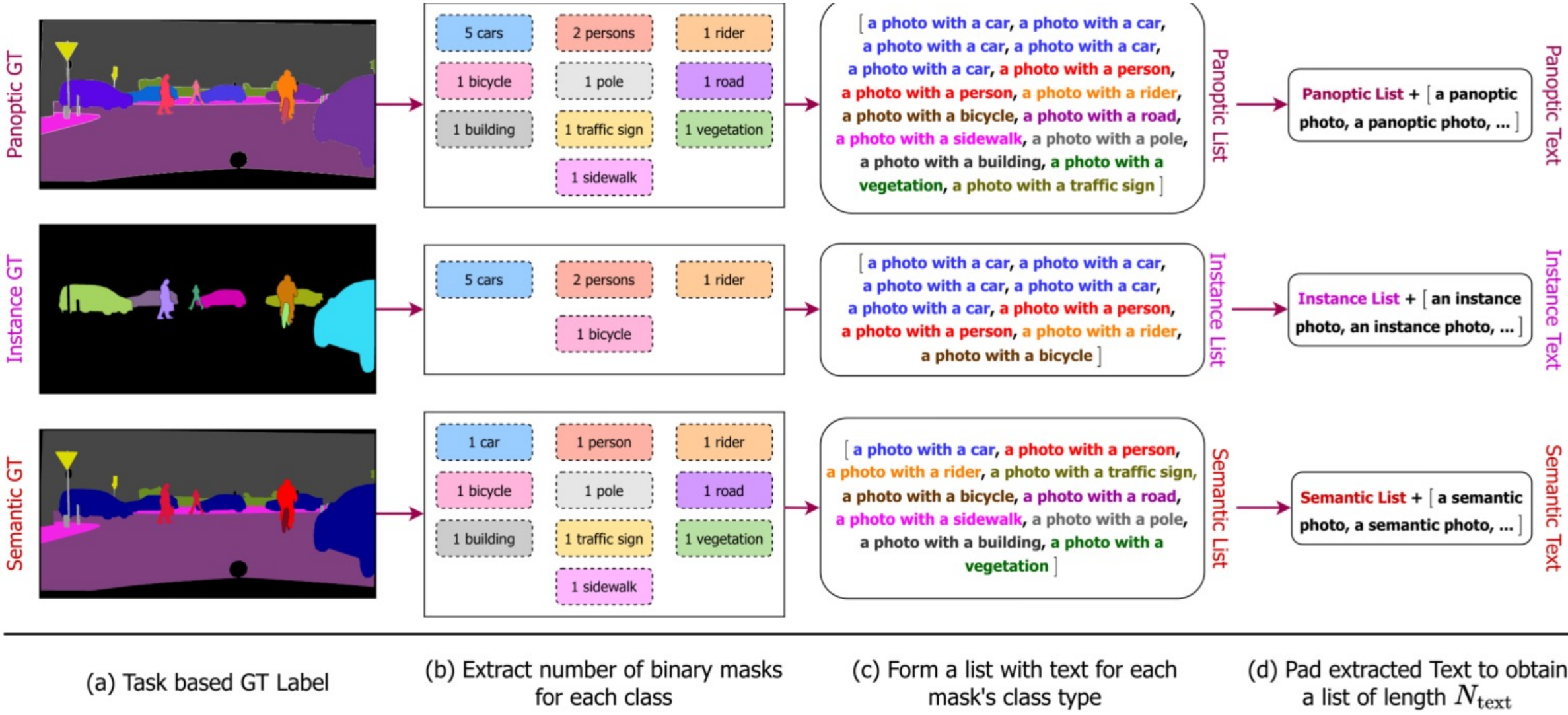
Jain et al. OneFormer: One Transformer to Rule Universal Image Segmentation. CVPR 2023

OneFormer



Jain et al. OneFormer: One Transformer to Rule Universal Image Segmentation. CVPR 2023

OneFormer



Jain et al. OneFormer: One Transformer to Rule Universal Image Segmentation. CVPR 2023

OneFormer

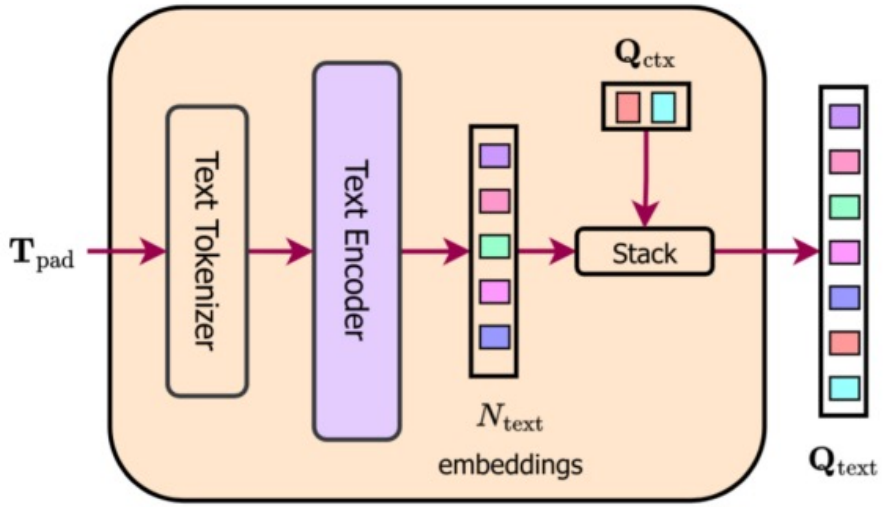


Figure 4. **Text Mapper.** We tokenize and then encode the input text list ($\mathbf{T}_{\text{pad}}$) using a 6-layer transformer text encoder [49, 57] to obtain a set of N_{text} embeddings. We concatenate a set of N_{ctx} learnable embeddings to the encoded representations to obtain the final N text queries ($\mathbf{Q}_{\text{text}}$). The N text queries stand for a text-based representation of the objects present in an image.

$$\mathcal{L}_{\mathbf{Q}\to\mathbf{Q}_{\text{text}}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(q_i^{obj}\odot q_i^{txt}/\tau)}{\sum_{j=1}^{B}\exp(q_i^{obj}\odot q_j^{txt}/\tau)},$$

$$\mathcal{L}_{\mathbf{Q}_{\text{text}}\to\mathbf{Q}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(q_i^{txt}\odot q_i^{obj}/\tau)}{\sum_{j=1}^{B}\exp(q_i^{txt}\odot q_j^{obj}/\tau)}$$

$$\mathcal{L}_{\mathbf{Q}\leftrightarrow\mathbf{Q}_{\text{text}}} = \mathcal{L}_{\mathbf{Q}\to\mathbf{Q}_{\text{text}}} + \mathcal{L}_{\mathbf{Q}_{\text{text}}\to\mathbf{Q}}$$

Jain et al. OneFormer: One Transformer to Rule Universal Image Segmentation. CVPR 2023

OneFormer

Method	Backbone	#Params	#FLOPs	#Queries	Crop Size	Iters	PQ	AP	mIoU (s.s.)	mIoU (m.s.)
Individual Training										
CMT-DeepLab‡ [59]	MaX-S† [50]	—	—	—	1025×2049	60k	64.6	—	81.4	—
Axial-DeepLab-L‡ [51]	Axial ResNet-L† [51]	45M	687G	—	1025×2049	60k	63.9	35.8	81.0	81.5
Axial-DeepLab-XL‡ [51]	Axial ResNet-XL† [51]	173M	2447G	—	1025×2049	60k	64.4	36.7	80.6	81.1
Panoptic-DeepLab‡ [11]	SWideRNet† [8]	536M	10365G	—	1025×2049	60k	66.4	40.1	82.2	82.9
Mask2Former-Panoptic [12]	Swin-L† [38]	216M	514G	200	512×1024	90k	66.6	43.6	82.9	—
Mask2Former-Instance [12]	Swin-L† [38]	216M	507G	200	512×1024	90k	—	43.7	—	—
Mask2Former-Semantic [12]	Swin-L† [38]	215M	494G	100	512×1024	90k	—	—	83.3	84.3
kMaX-DeepLab‡ [60]	ConvNeXt-L† [39]	232M	1673G	256	1025×2049	60k	68.4	44.0	83.5	—
Joint Training										
OneFormer	Swin-L† [38]	219M	543G	250	512×1024	90k	**67.2**	**45.6**	83.0	**84.4**
OneFormer	ConvNeXt-L† [39]	220M	497G	250	512×1024	90k	**68.5**	**46.5**	83.0	84.0
OneFormer	ConvNeXt-XL† [39]	372M	775G	250	512×1024	90k	**68.4**	**46.7**	**83.6**	**84.6**
OneFormer	DiNAT-L† [21]	223M	450G	250	512×1024	90k	**67.6**	**45.6**	83.1	84.0

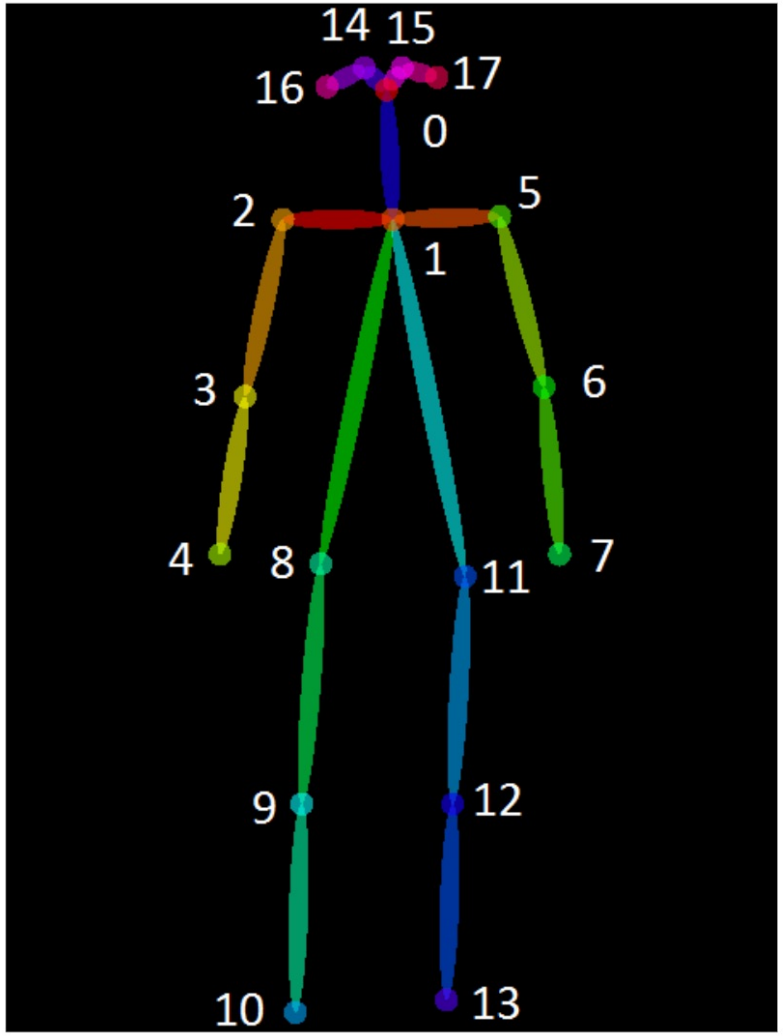
Table 2. **SOTA Comparison on Cityscapes val set.** †: backbones pretrained on ImageNet-22K; ‡: trained with batch size 32, *: hidden dimension 1024. OneFormer outperforms the individually trained Mask2Former [12] models. Mask2Former's performance with 250 queries is not listed, as its performance degrades with 250 queries. We compute FLOPs using the corresponding crop size.

Jain et al. OneFormer: One Transformer to Rule Universal Image Segmentation. CVPR 2023

План лекции

1. Такая разная сегментация
2. Визуальная сегментация
3. Семантическая сегментация
4. Интерактивная сегментация
5. Сегментация экземпляров
6. Паноптическая сегментация
7. Оценка позы человека

Оценка позы человека



Photograph taken from Pexels

Какая связь с сегментацией?

Метрики

Percentage of Correct Parts - PCP: A limb is considered detected (a correct part) if the distance between the two predicted joint locations and the true limb joint locations is less than half of the limb length (Commonly denoted as PCP@0.5).

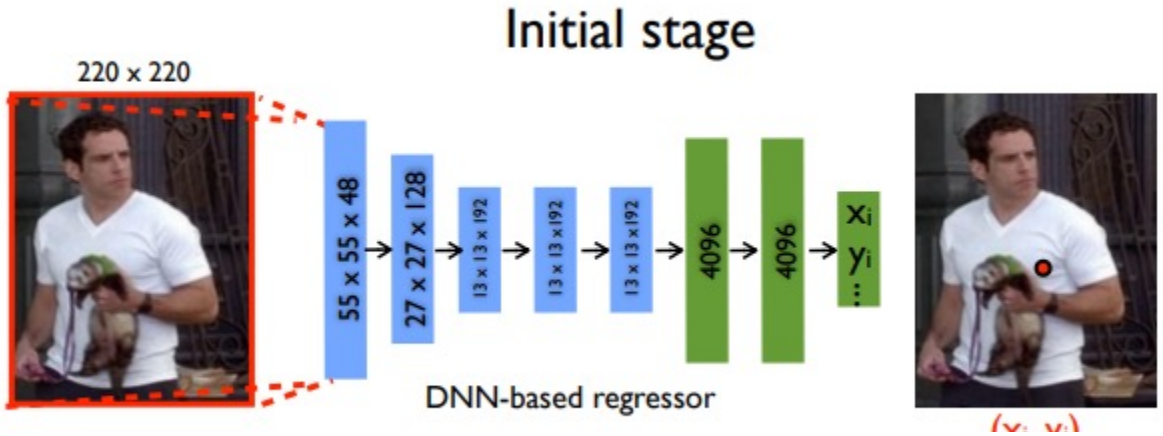
•It measures the detection rate of limbs. The con is that it penalizes shorter limbs more since shorter limbs have smaller thresholds.

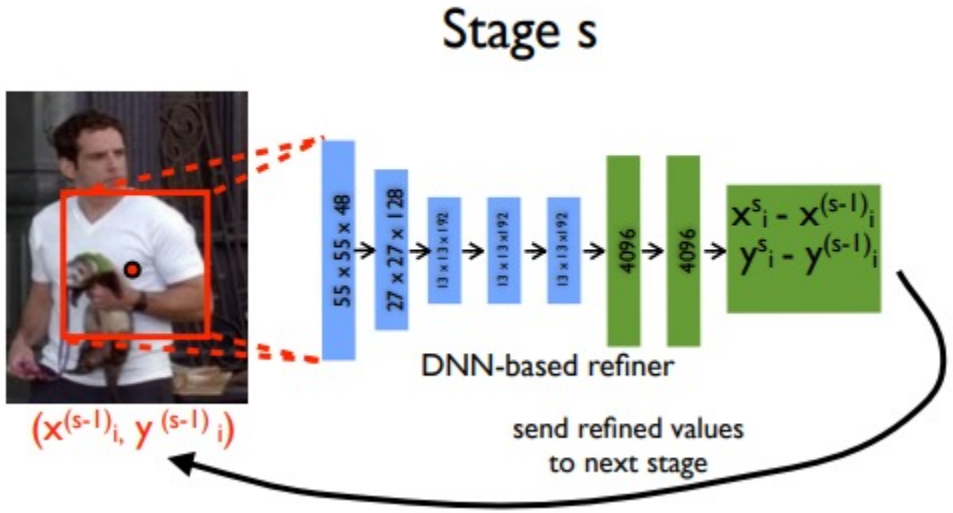
Percentage of Correct Key-points - PCK: A detected joint is considered *correct* if the distance between the predicted and the true joint is within a certain threshold. The threshold can either be:

• PCKh@0.5 is when the threshold = 50% of the head bone link

• PCK@0.2 == Distance between predicted and true joint < 0.2 * torso diameter

Прямая регрессия





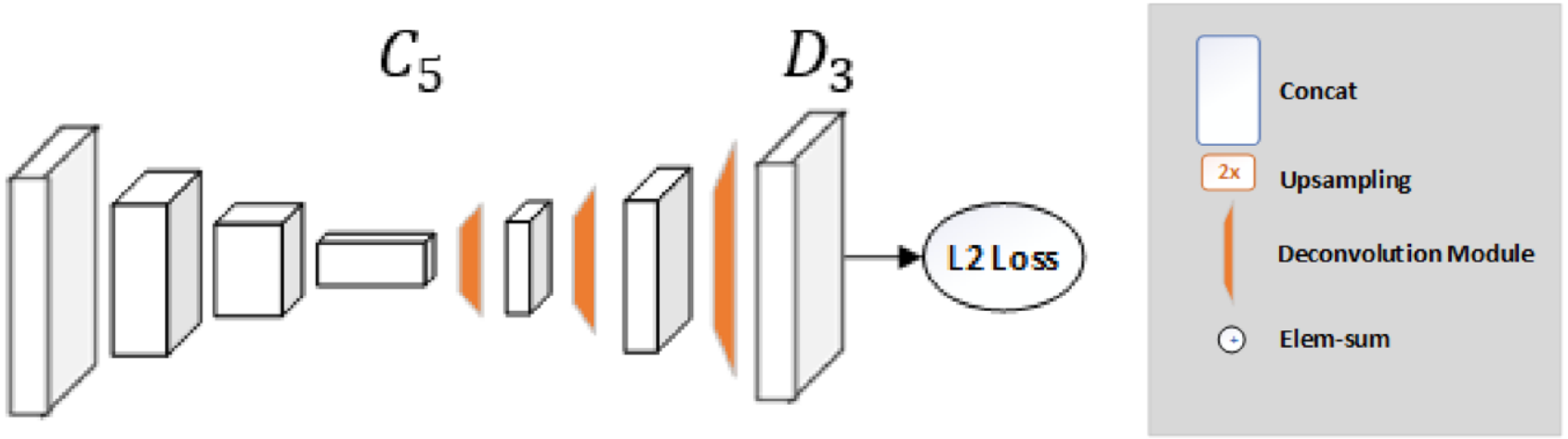
Toshev, Szegedy. DeepPose: Human Pose Estimation via Deep Neural Networks. CVPR 2014

Поза человека через сегментацию

Fig. 2. Example output produced by our network. On the left we see the final pose estimate provided by the max activations across each heatmap. On the right we show sample heatmaps. (From left to right: neck, left elbow, left wrist, right knee, right ankle)

Простой baseline

- ResNet + deconvolution



ECCV2018 paper "Simple Baselines for Human Pose Estimation andTracking» (https://arxiv.org/abs/1804.06208)

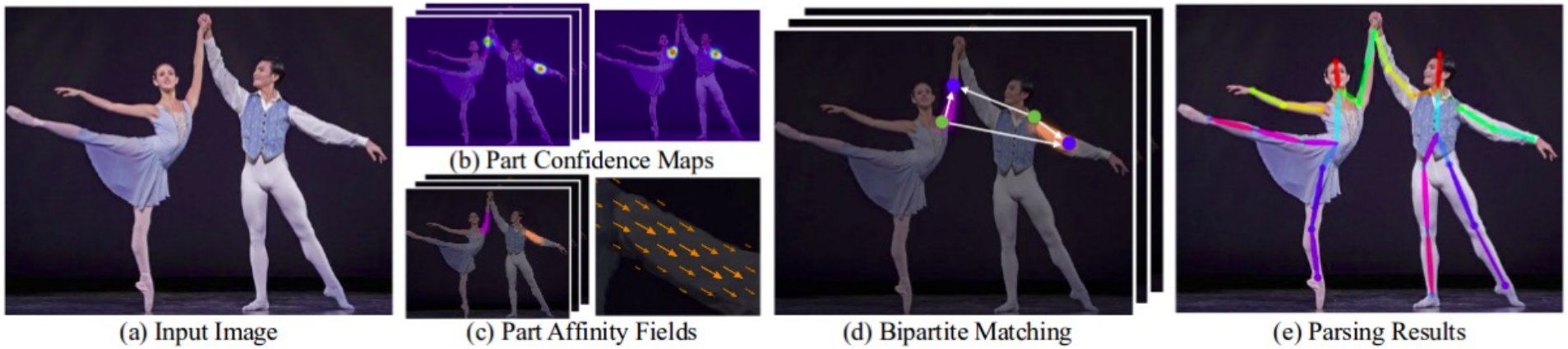
https://github.com/Microsoft/human-pose-estimation.pytorch

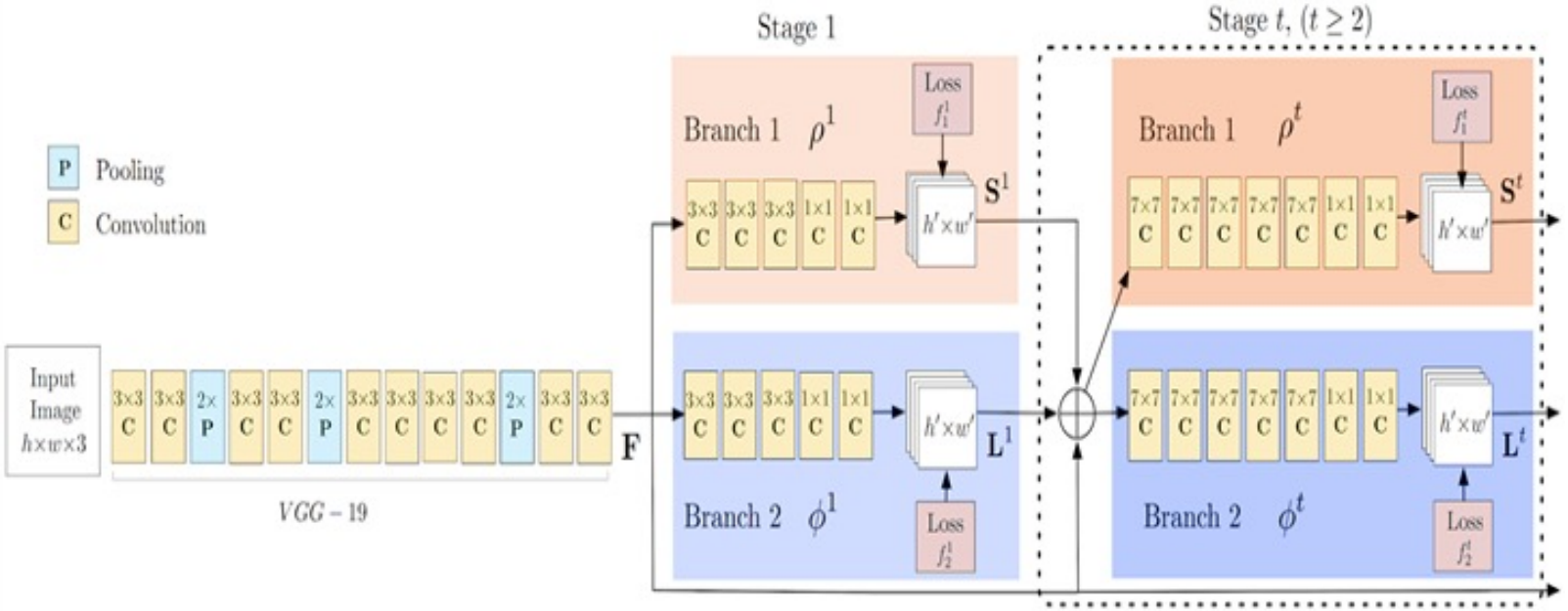
Оценка качества

Table 4. Comparisons on COCO test-dev dataset. **Top**: methods in the literature, trained only on COCO training dataset. **Middle**: results submitted to COCO test-dev leaderboard [9], which have either extra training data (*) or models ensamled (+). **Bottom**: our single model results, trained only on COCO training dataset.

Method	Backbone	Input Size	AP	AP_{50}	AP_{75}	AP_m	AP_l	AR
CMU-Pose [5]	-	-	61.8	84.9	67.5	57.1	68.2	66.5
Mask-RCNN [12]	ResNet-50-FPN	-	63.1	87.3	68.7	57.8	71.4	-
G-RMI [24]	ResNet-101	353 × 257	64.9	85.5	71.3	62.3	70.0	69.7
CPN [6]	ResNet-Inception	384 × 288	72.1	91.4	80.0	68.7	77.2	78.5
FAIR* [9]	ResNeXt-101-FPN	-	69.2	90.4	77.0	64.9	76.3	75.2
G-RMI* [9]	ResNet-152	353 × 257	71.0	87.9	77.7	69.0	75.2	75.8
oks* [9]	-	-	72.0	90.3	79.7	67.6	78.4	77.1
bangbangren*+ [9]	ResNet-101	-	72.8	89.4	79.6	68.6	**80.0**	78.7
CPN+ [6,9]	ResNet-Inception	384 × 288	73.0	**91.7**	80.9	69.5	78.1	**79.0**
Ours	ResNet-152	384 × 288	**73.7**	**91.9**	**81.1**	**70.3**	**80.0**	**79.0**

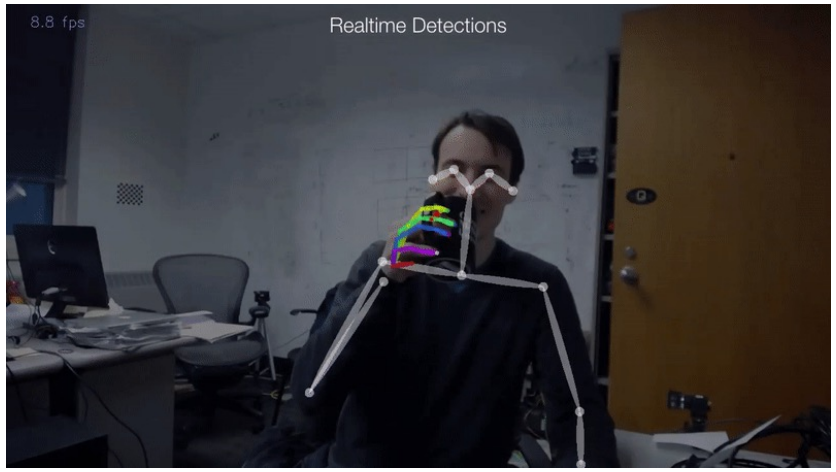
OpenPose

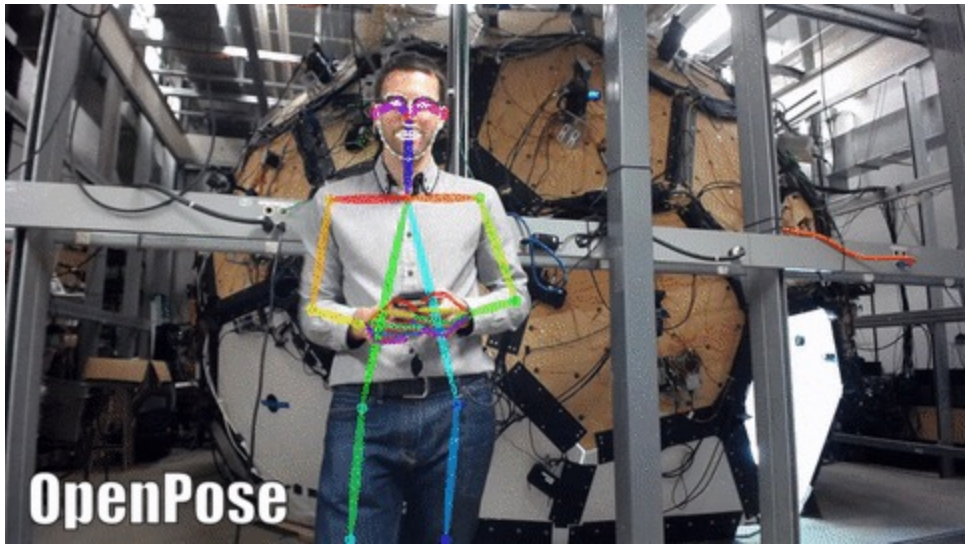




https://github.com/CMU-Perceptual-Computing-Lab/openpose

Примеры работы





Резюме лекции

- Множество разных задач сегментации
  - Извлечение объектов
  - Пересегментация / суперпиксели
  - Семантическая сегментация
  - Instance segmentation
  - Паноптическая сегментация
  - Поза человека через сегментацию
- Текущее развитие
  - Нейросетевые модели = построение признаков + предсказание попиксельное
  - Encoder-decoder vs признаки высокого разрешения
  - Сегментационная голова как дополнение к другим моделям
  - Трансформерные модели