СЕГМЕНТАЦИЯ, VAE

Сергей Николенко

ИТМО — Санкт-Петербург

16 ноября 2026 г.





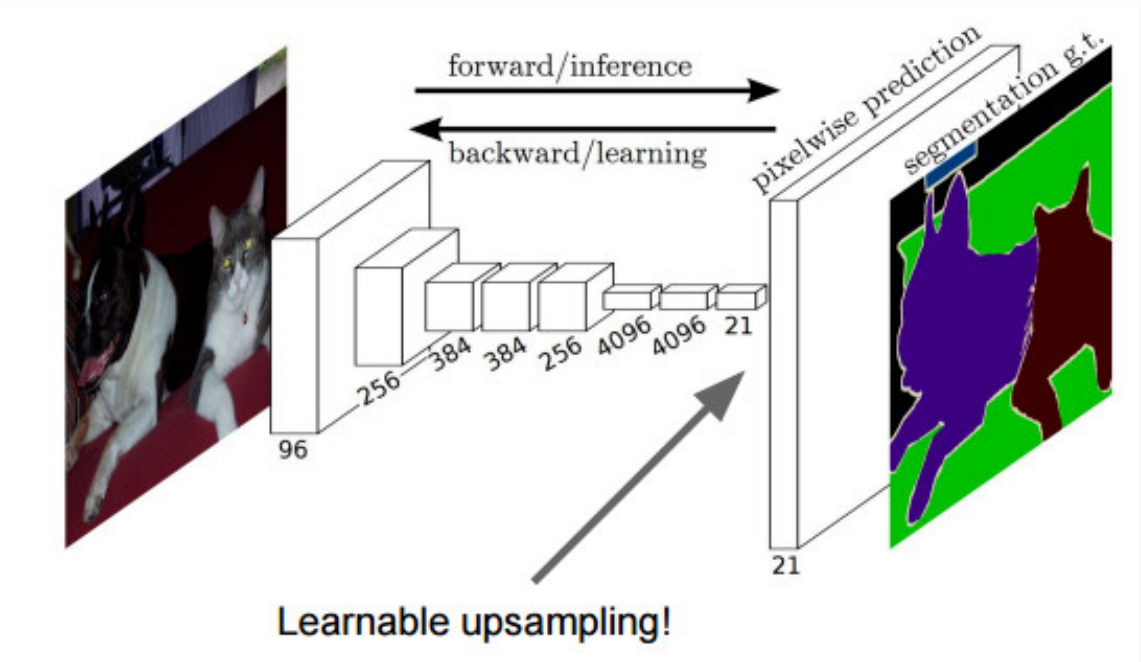
Random facts:

- 16 ноября 1532 г. отряд Франсиско Писарро захватил правителя инков Атауальпу в Кахамарке
- 16 ноября 1904 г. Джон Амброз Флеминг подал заявку на патент двухэлектродной электронной лампы — вакуумного диода
- 16 ноября 1945 г. в Лондоне подписали устав ЮНЕСКО, организации ООН по вопросам образования, науки и культуры
- 16 ноября 1973 г. стартовала экспедиция Skylab 4; её экипаж провёл в космосе 84 дня, установив рекорд длительности полёта
- 16 ноября 1974 г. радиотелескоп Аресибо отправил к скоплению M13 послание из 1679 бит — произведения простых чисел 23 и 73
- 16 ноября 1988 г. Верховный Совет Эстонской ССР принял декларацию о суверенитете республики

Сегментация

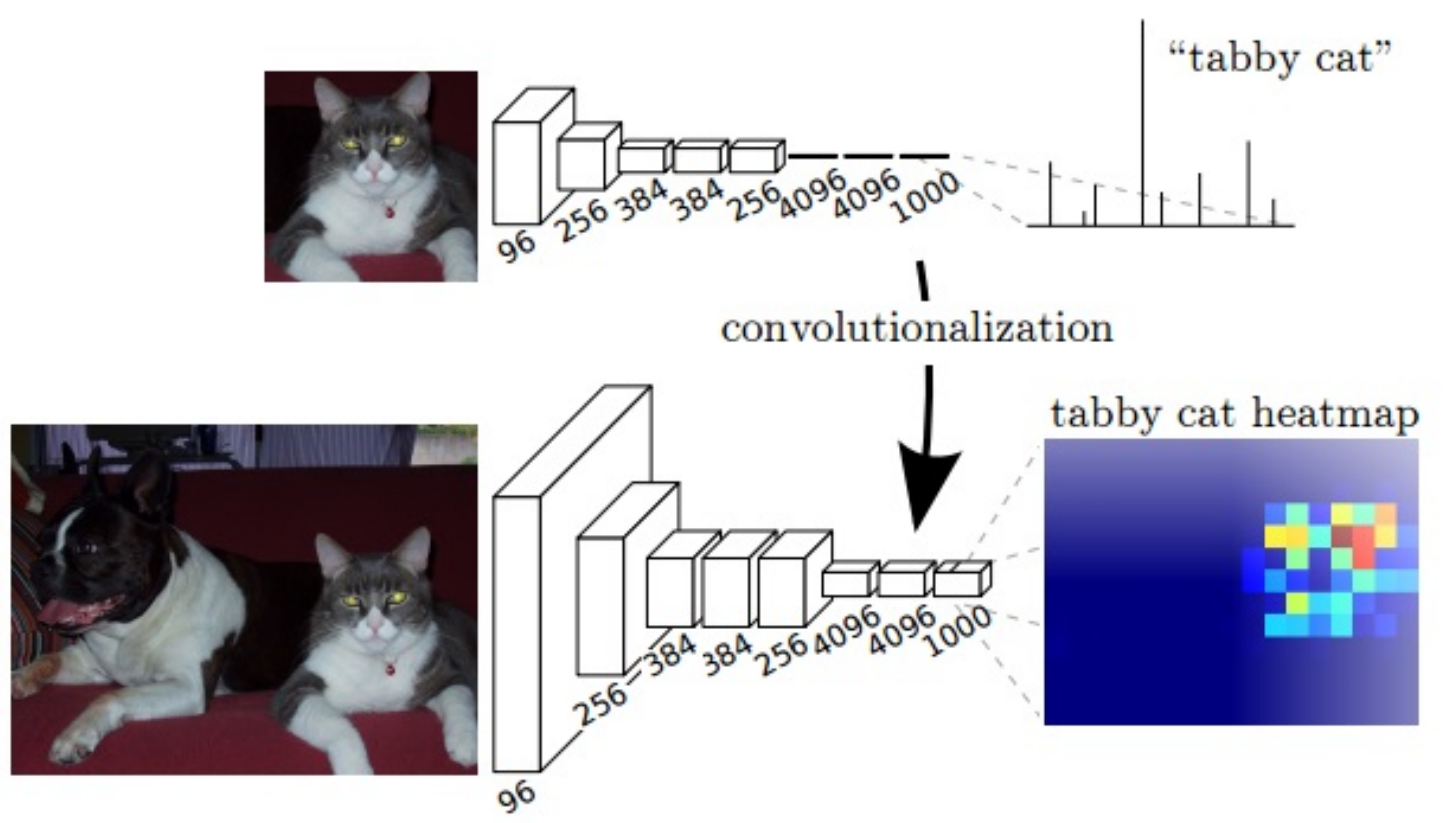
Fully Convolutional Networks

- (Long et al., 2015): Fully Convolutional Networks for Semantic Segmentation
- FCN (fully convolutional network): вместо полносвязных уровней добавляем свёртки с большим receptive field (upsampling).



Fully Convolutional Networks

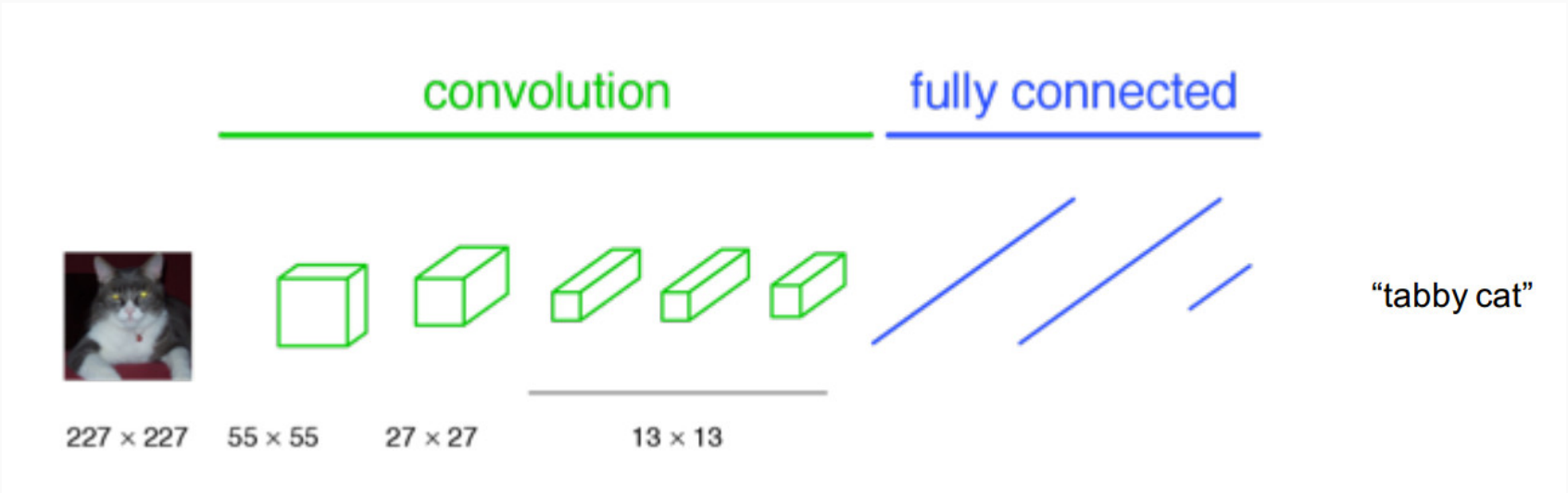
- Эффект локализации получается за счёт того, чтобы посмотреть, где более сильные активации:



- Идея в том, чтобы этот эффект отмасштабировать обратно к исходному размеру картинки.

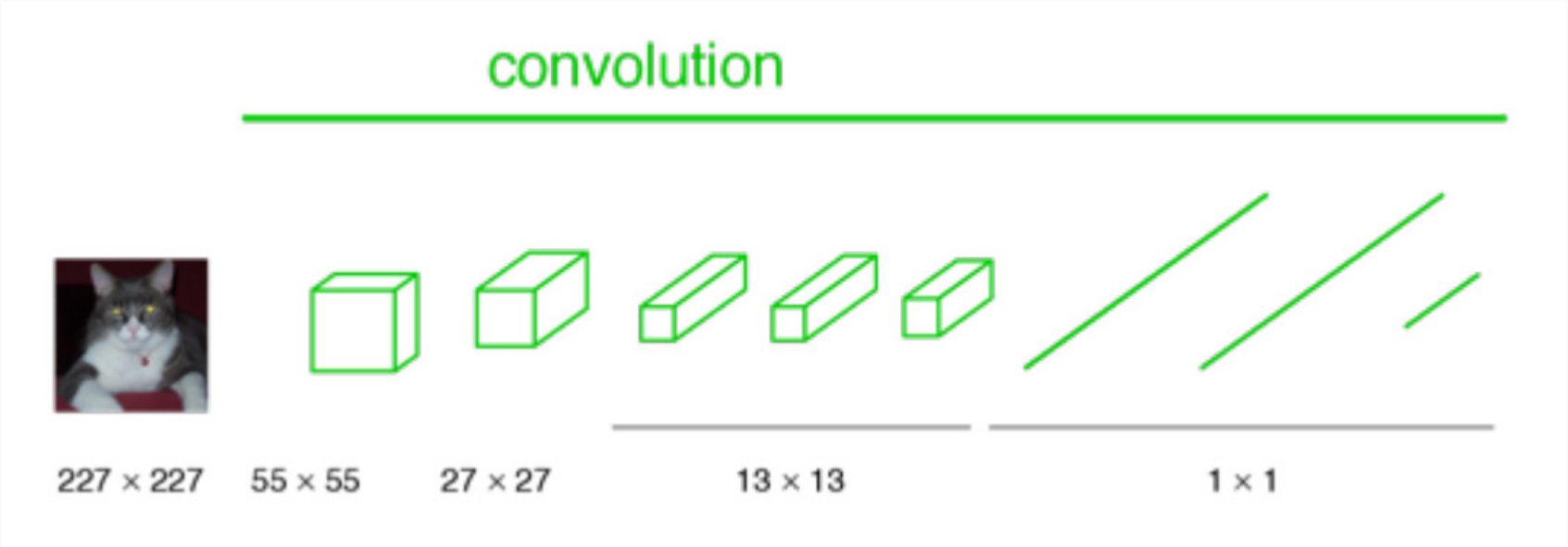
Fully Convolutional Networks

- И можно сконвертировать “обычную” CNN в FCN – начнём с обычной CNN для классификации:



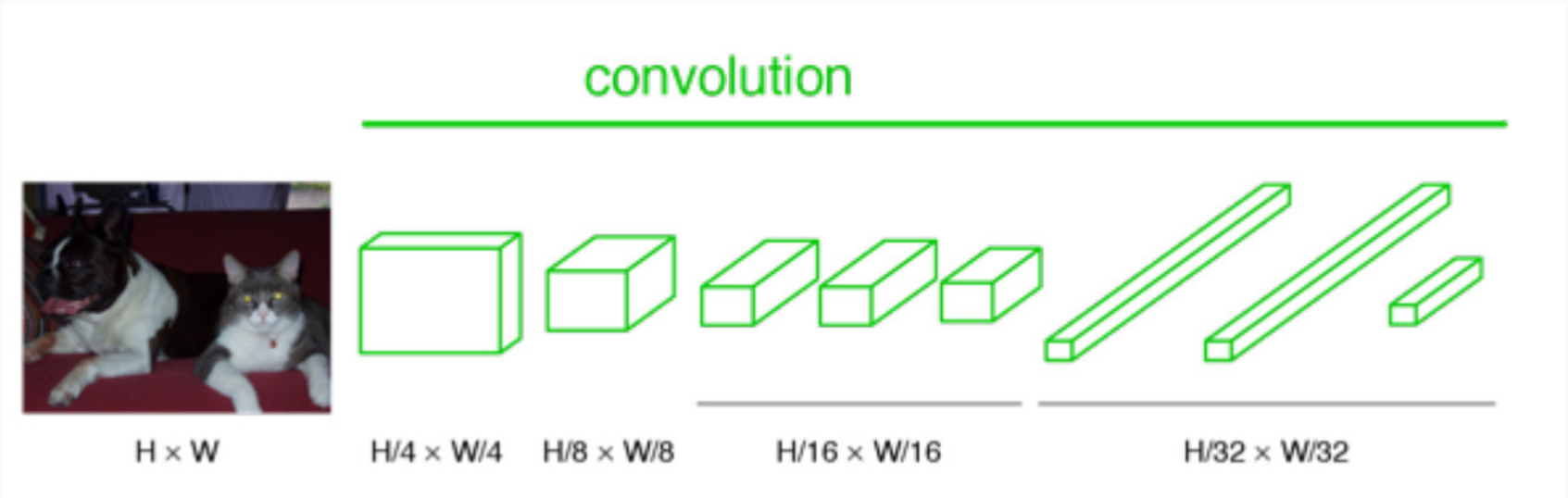
Fully Convolutional Networks

- Конвертируем полносвязные слои в 1×1 свёрточные слои (с теми же весами):



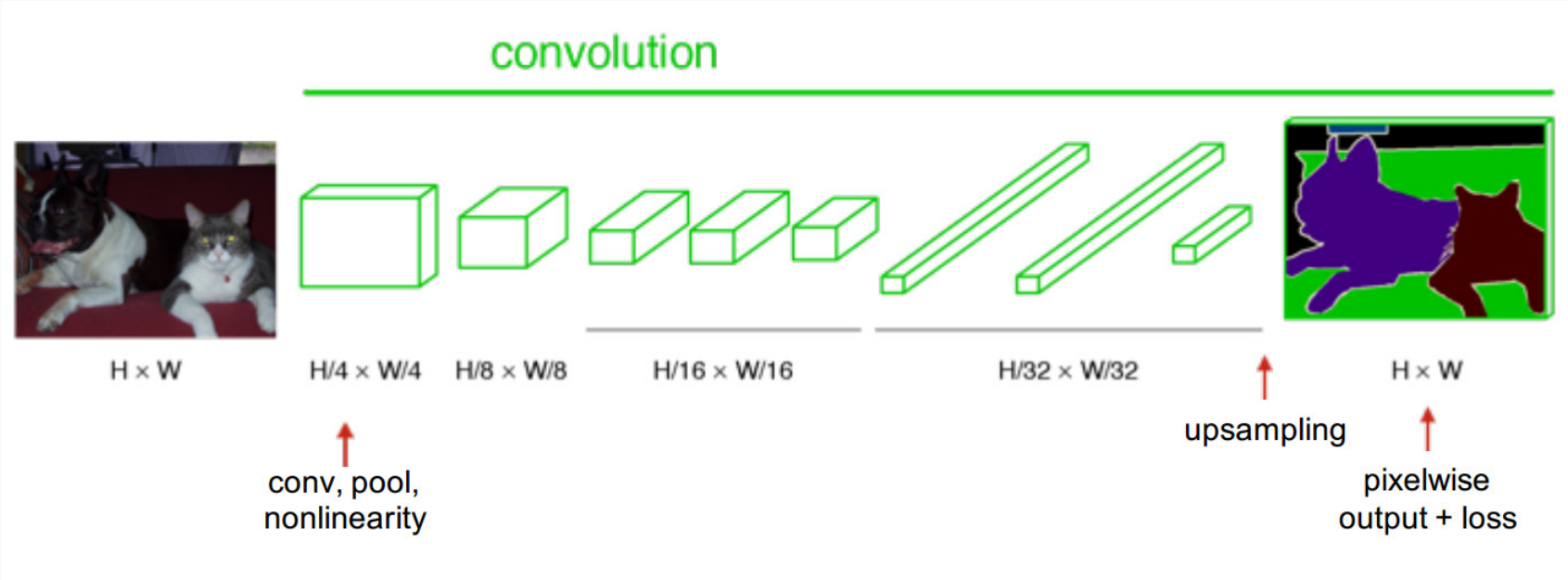
Fully Convolutional Networks

- Теперь вход может быть, кстати говоря, переменного размера:



Fully Convolutional Networks

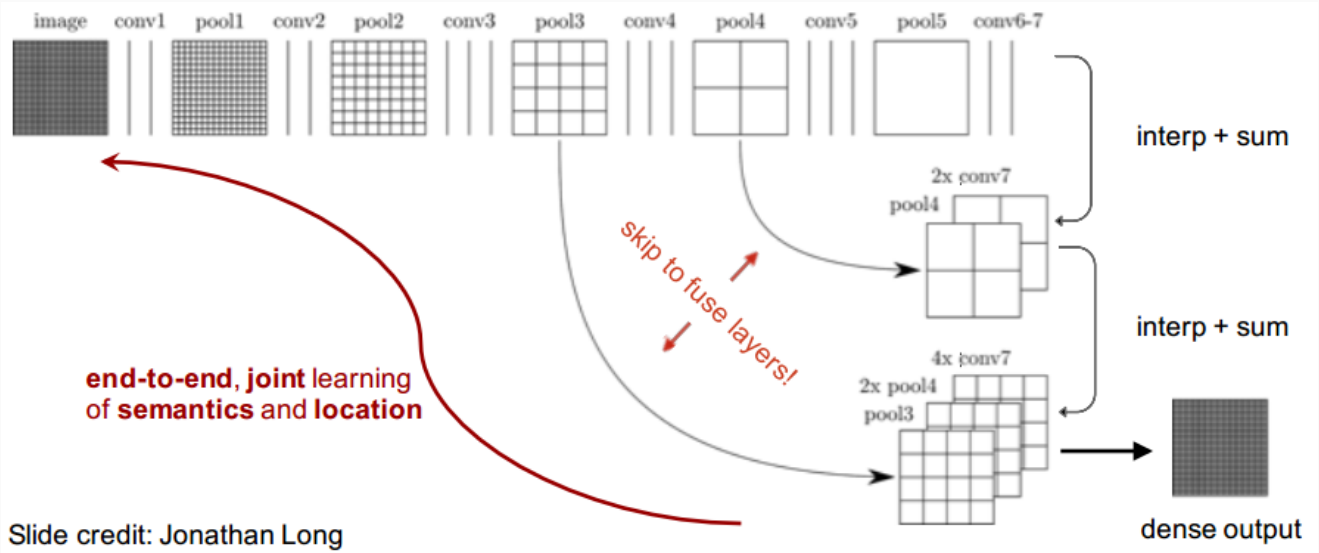
- А для upsampling добавляет деконволюцию (транспонированную свёртку):

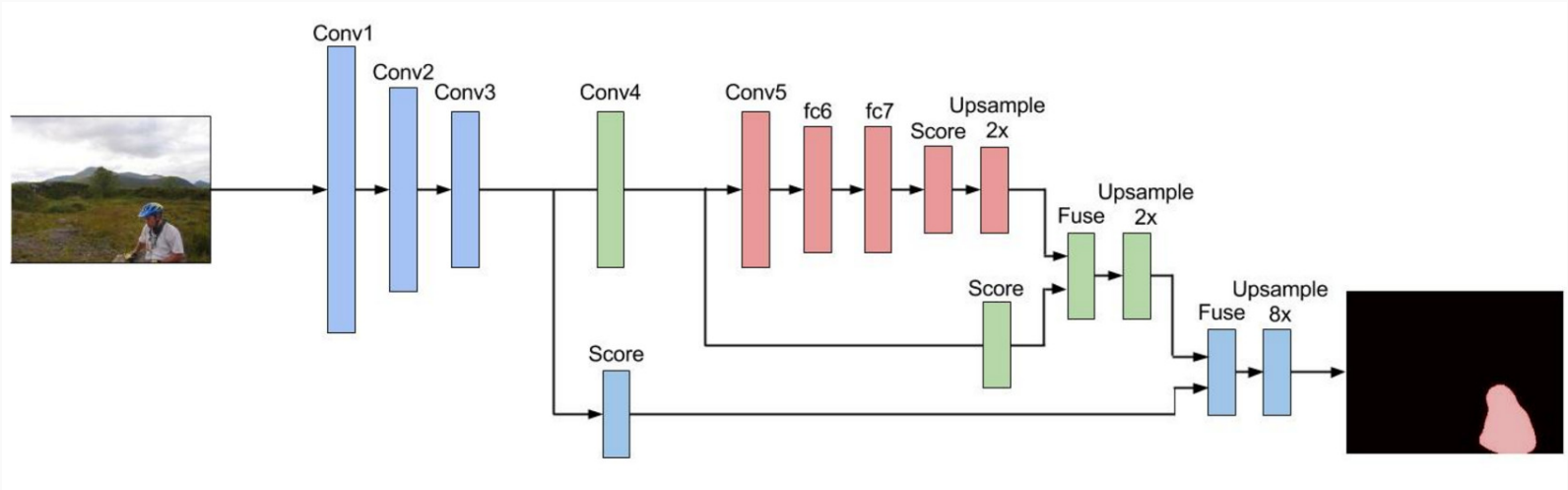


- Проблема здесь в том, что мы, конечно, теряем разрешение, потому что с активациями сначала был downsampling.

Fully Convolutional Networks

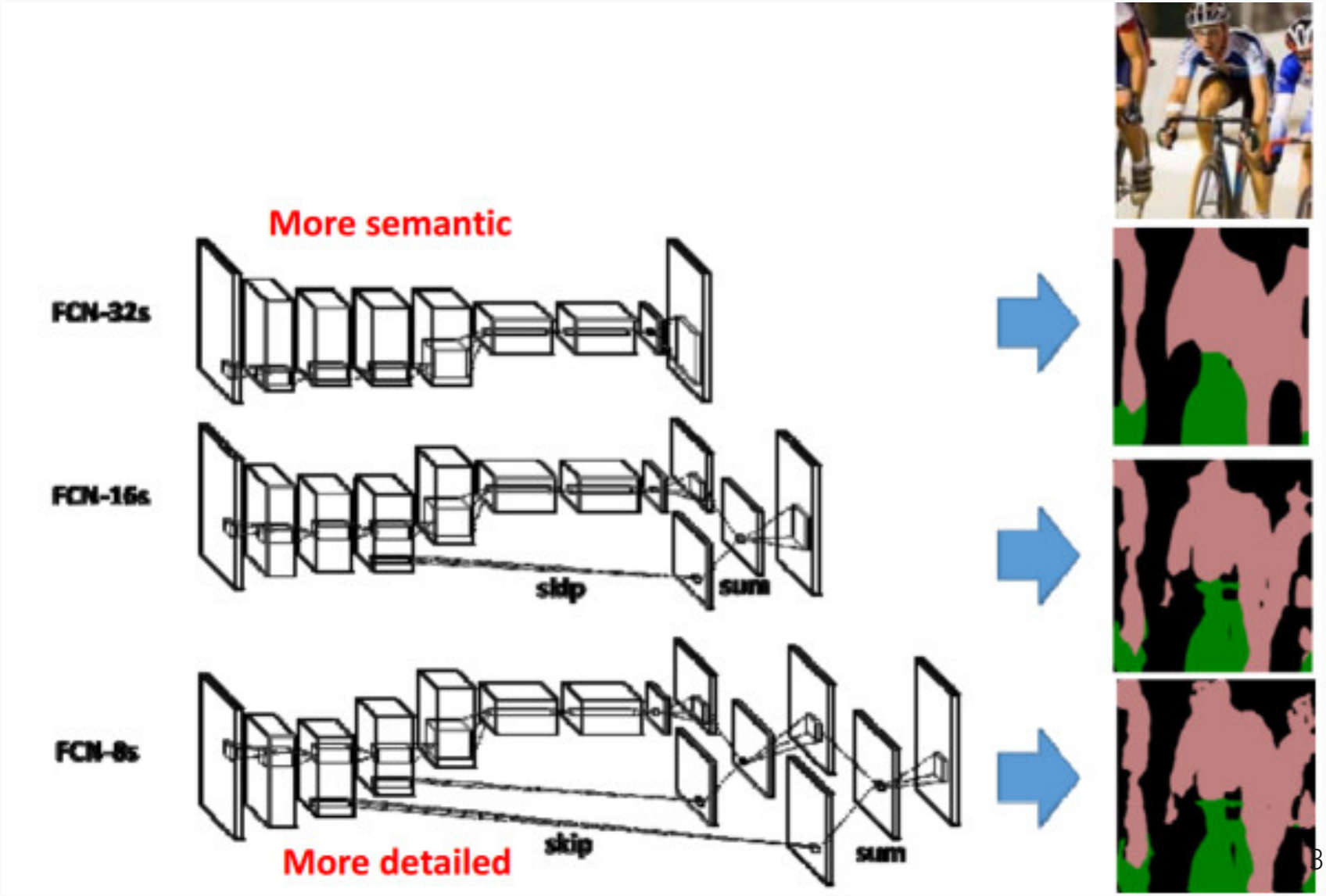
- Решение в том, чтобы добавить skip connections с предыдущих уровней:





Fully Convolutional Networks

- Вот какой эффект дают skip connections:



Fully Convolutional Networks

- Пример результатов FCN:



DeconvNet

- Большое достоинство FCN – берут всю картинку на вход и быстро работают: (Long et al., 2015) был в 286 раз быстрее предшественника (SDS, Simultaneous Detection and Segmentation).
- Проблема FCN – трудно обработать разные масштабы, причём в обе стороны:



(a) Inconsistent labels due to large object size

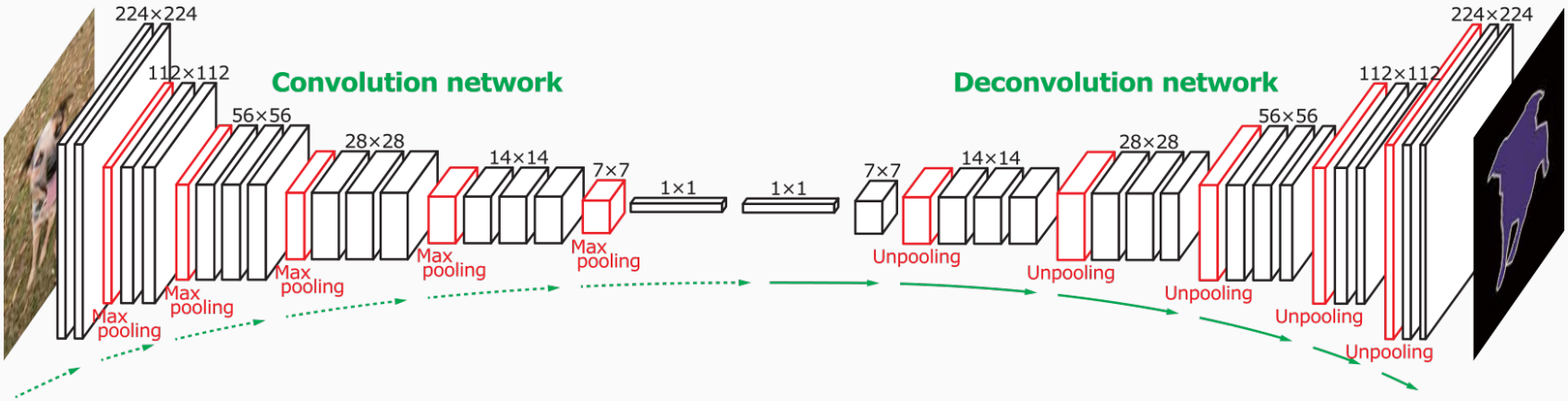


(b) Missing labels due to small object size

- И skip connections хоть и помогают, но не радикально.
- Кроме того, в итоге всё-таки достаточно грубая сегментация получается; в исходной FCN карта сегментации размером 16×16, а потом фактически билинейная интерполяция

DeconvNet

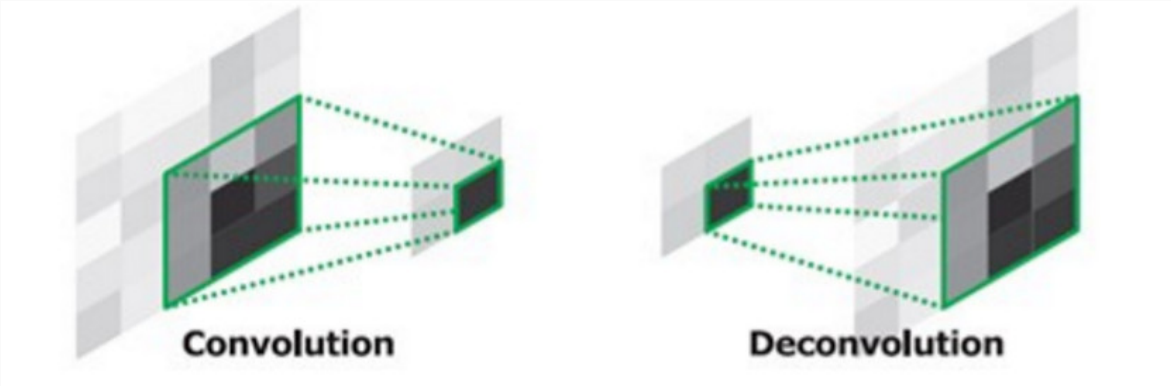
- DeconvNet (Noh et al., 2015) – давайте не делать skip connections, а просто deconvolutions/unpooling всю дорогу, применяя эту сеть к каждому object proposal:

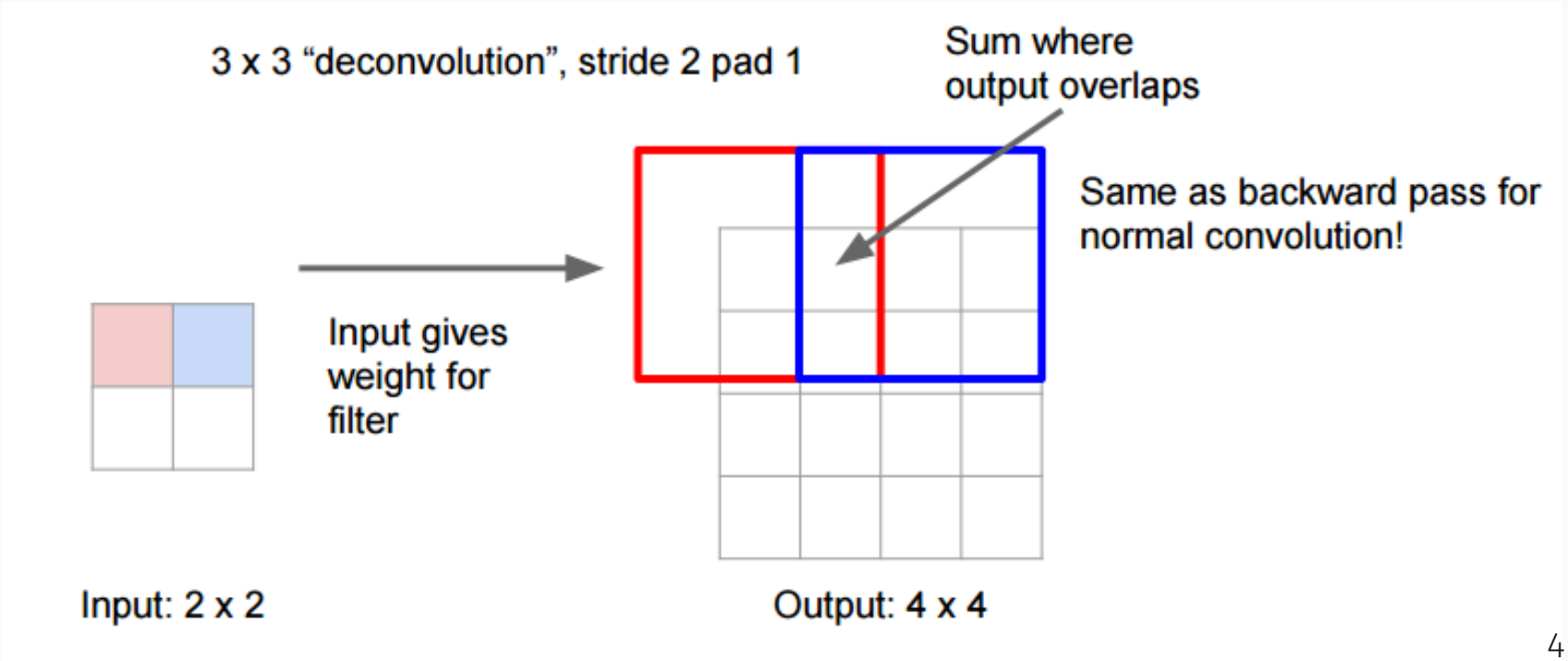


- Деконволюция, unpooling... что это такое вообще?

DeconvNet

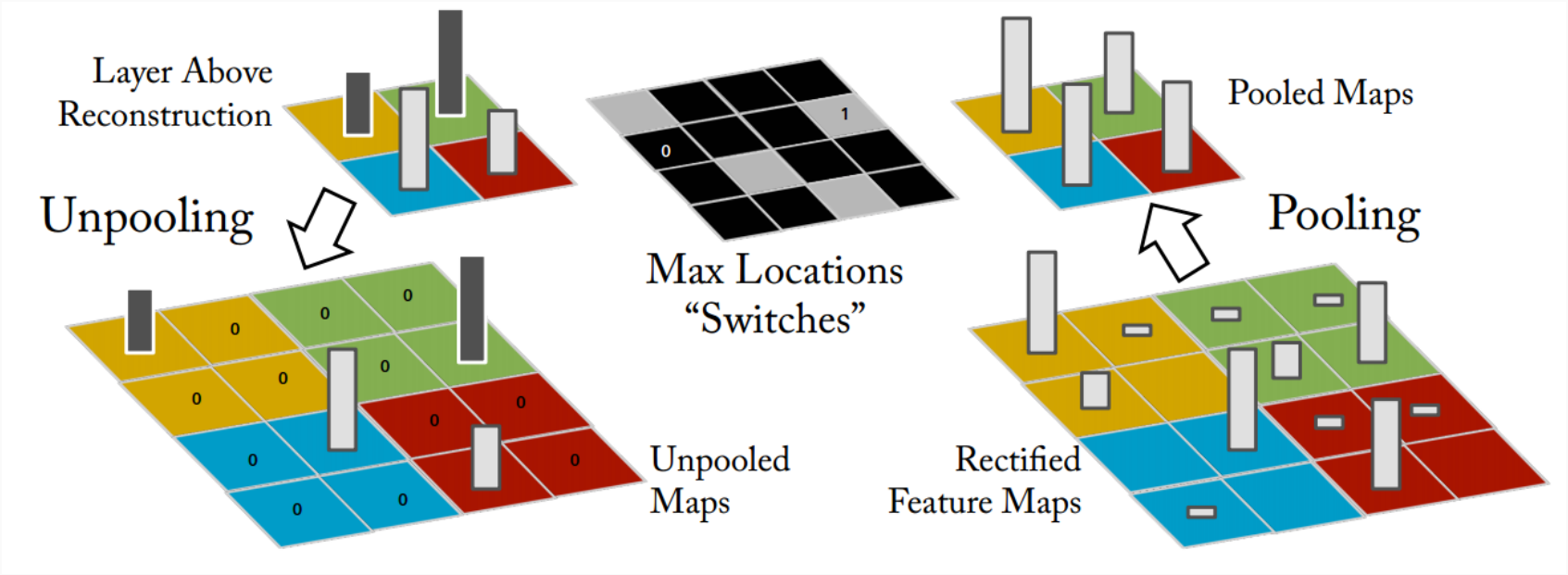
- Деконволюция:

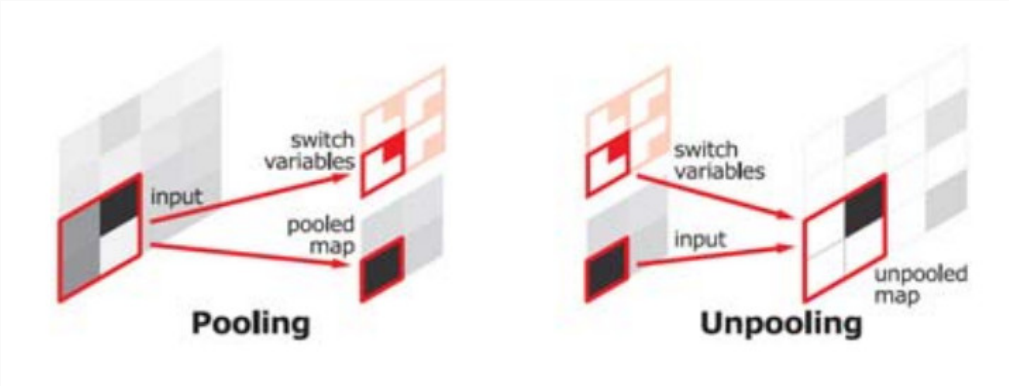




DeconvNet

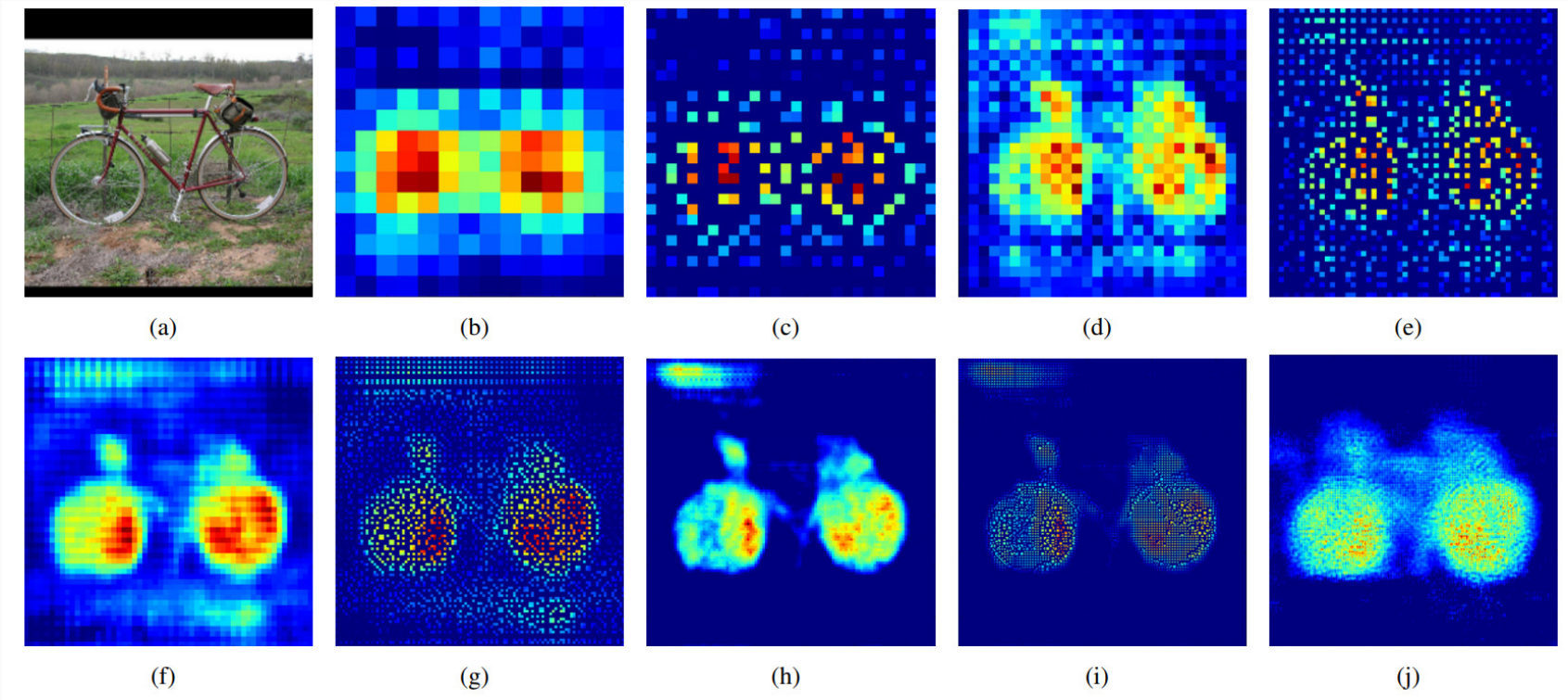
- Unpooling:





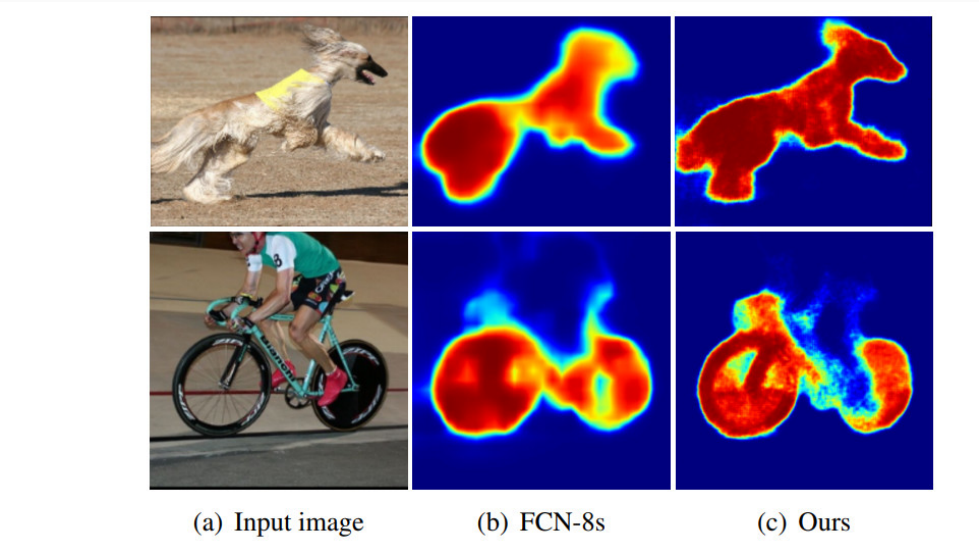
DeconvNet

- Вот так постепенно происходит деконволюция/unpooling:



DeconvNet

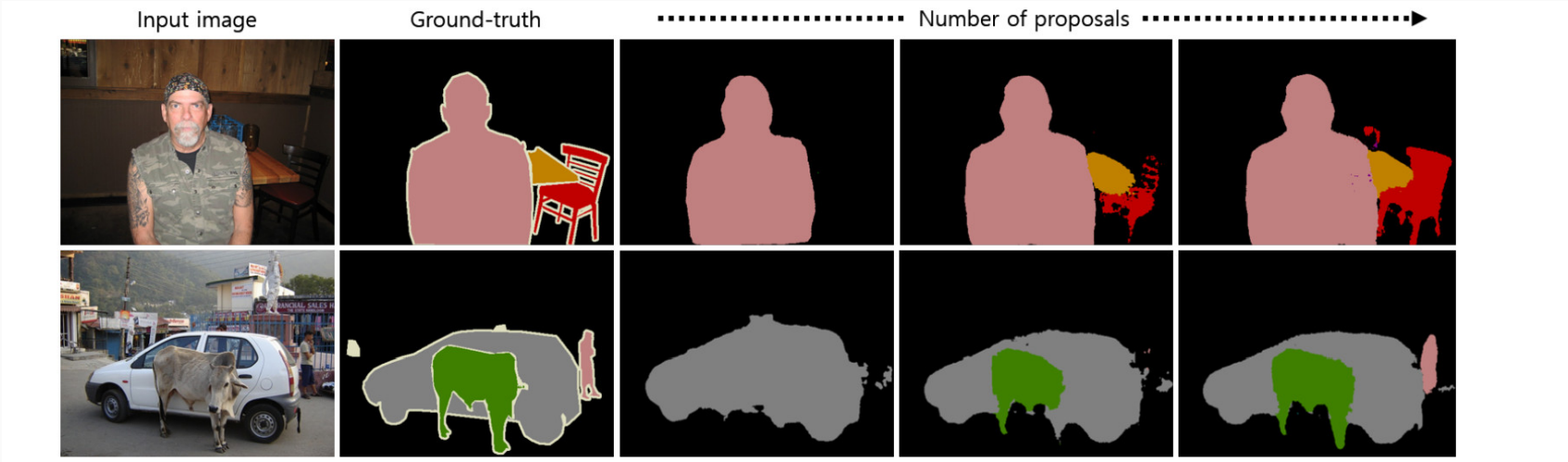
- Дольше обучается, но результаты могут быть лучше, а применение такое же по времени:



(a) Input image (b) FCN-8s (c) Ours

DeconvNet

- Работает отдельно на proposals, их можно упорядочить от большего к меньшему:



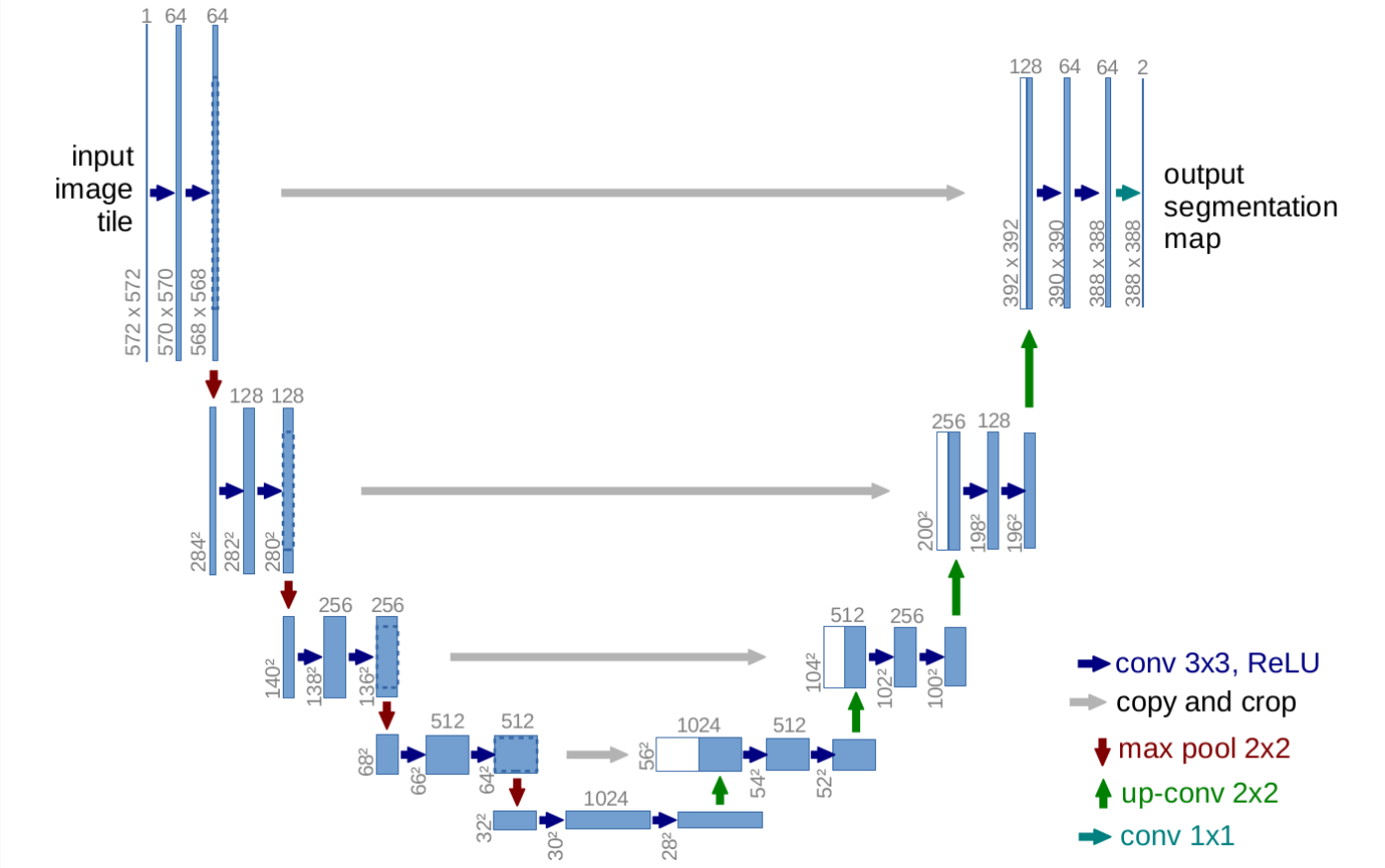
DeconvNet

- А можно сделать ансамбль с FCN:



U-Net

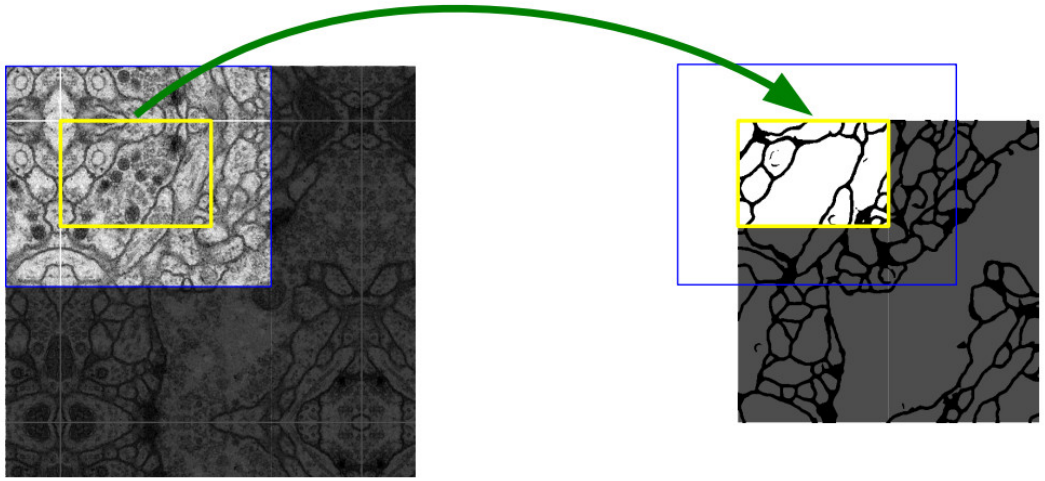
- U-Net (Ronneberger et al., 2015): очень популярная архитектура, особенно в medical imaging.



- Тоже skip connections; теперь upsampling часть по сути симметрична downsampling части, много признаков переходят.

U-Net

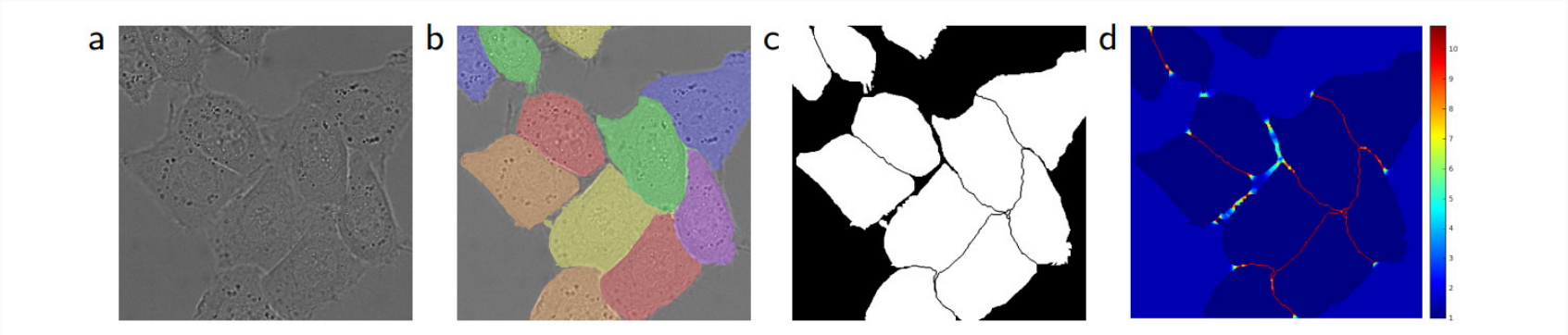
- Может работать с маленькой обучающей выборкой.
- Кроме того, поскольку это fully convolutional сеть, она может работать на картинках любого размера, покрывая их тайлами:



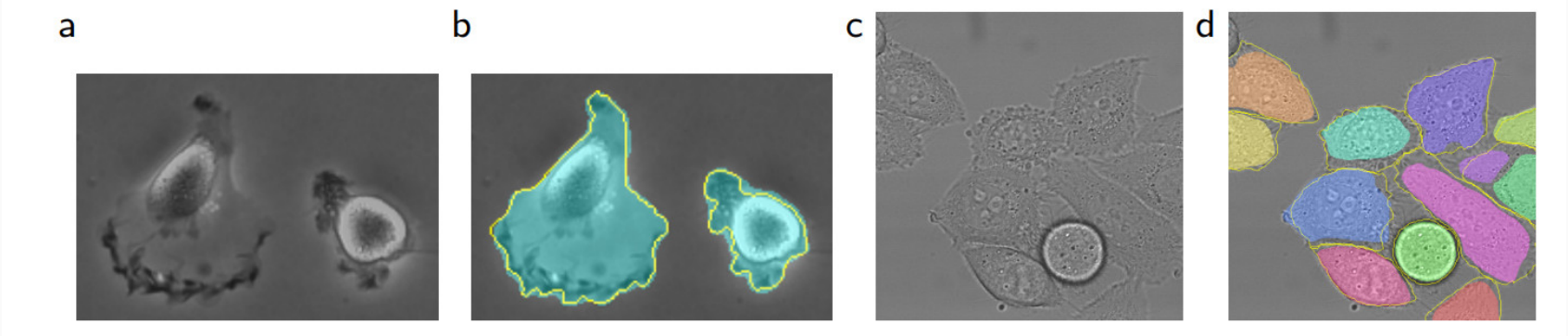
- Пиксели на границе предсказываем, например, отражая вход.

U-Net

- В исходной статье клетки сегментировали:

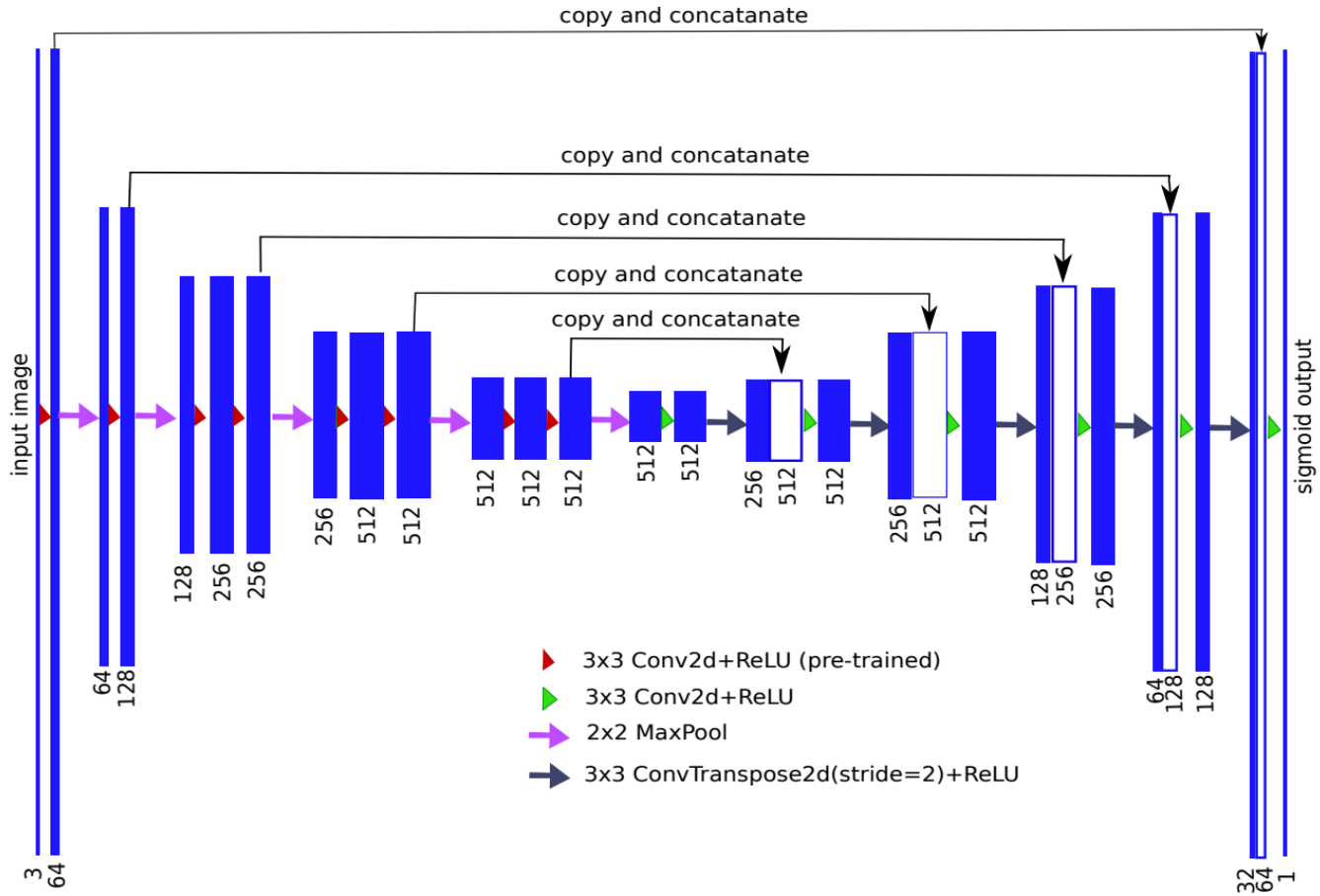


- Цвета – полученная маска, граница – ground truth:



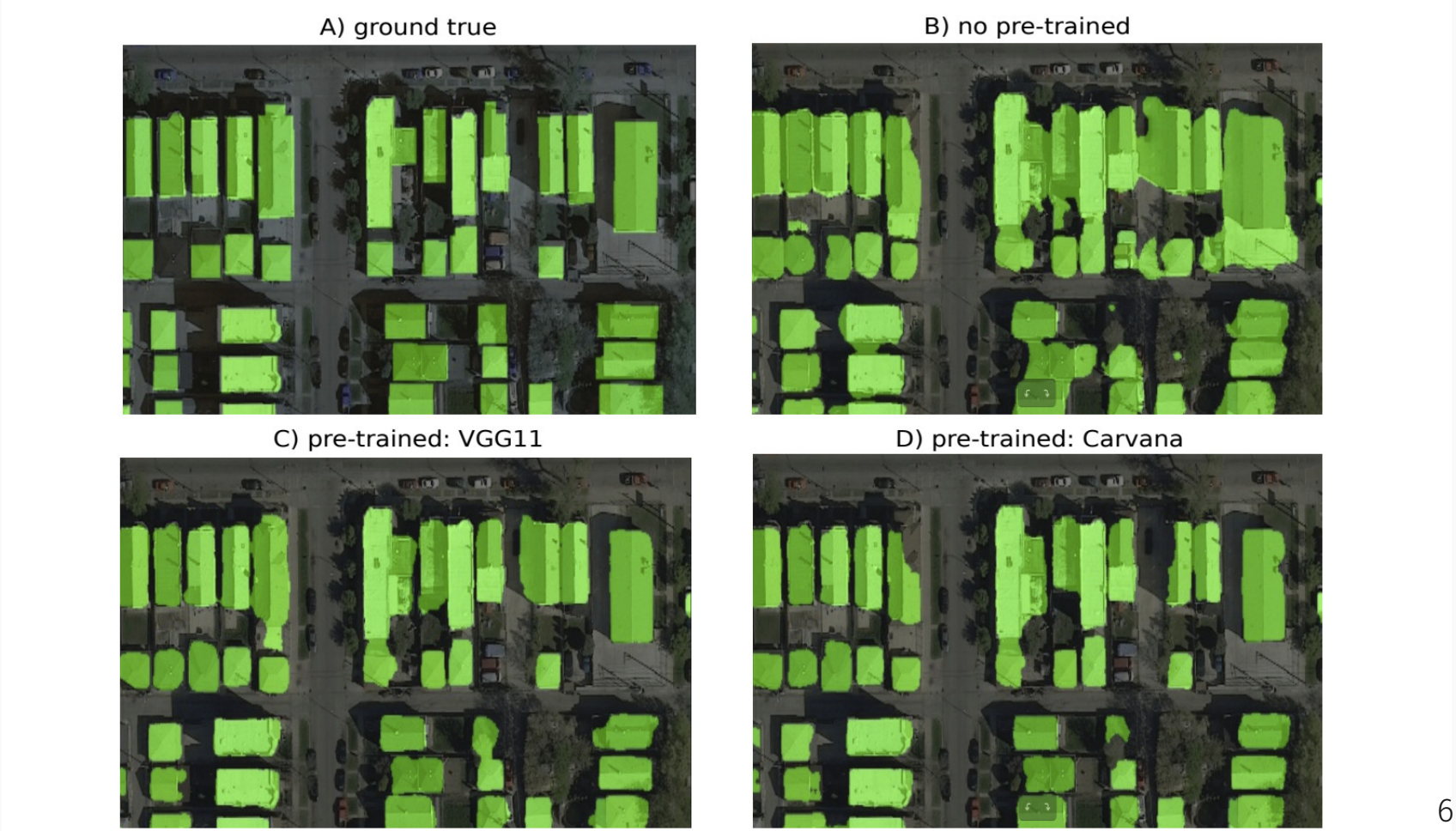
TernausNet и спутниковые снимки

- U-Net в последнее время стал основой для многих лучших моделей в сегментации с ограниченными датасетами.
- Вариант – TernausNet (Iglovikov, Shvets, 2018); основан на VGG11 без полносвязных слоёв:



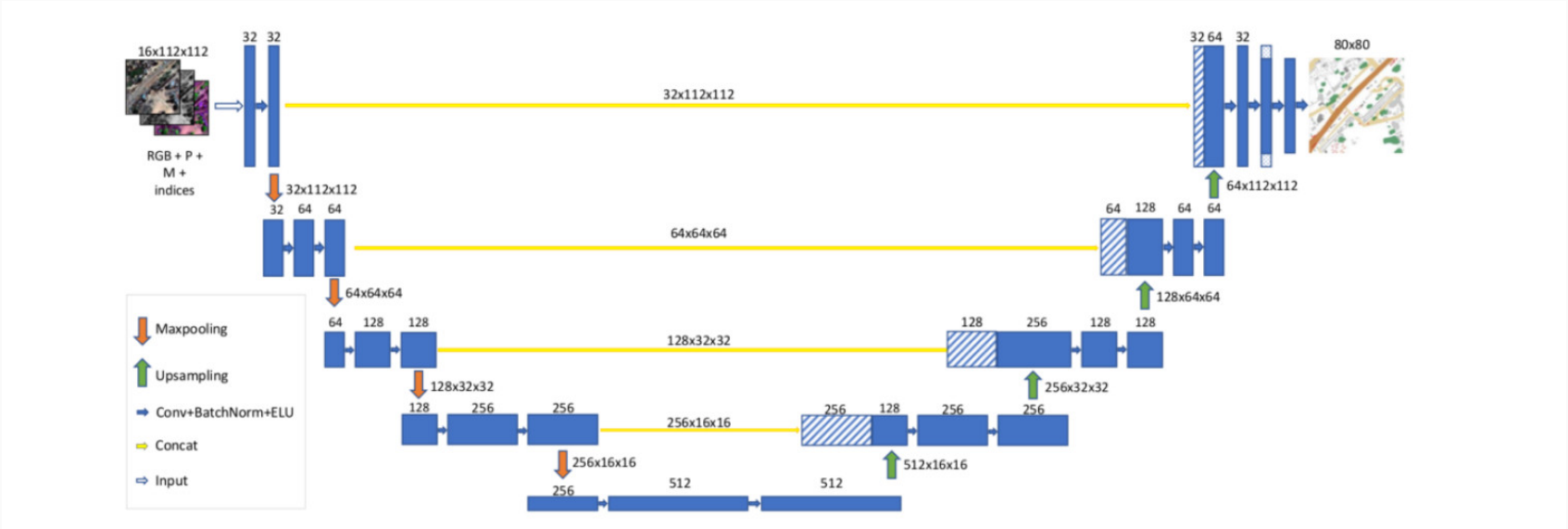
TernausNet и спутниковые снимки

- TernausNet выиграл Carvana Image Masking Challenge
- В (Iglovikov, Shvets, 2018) применён к сегментации спутниковых снимков:



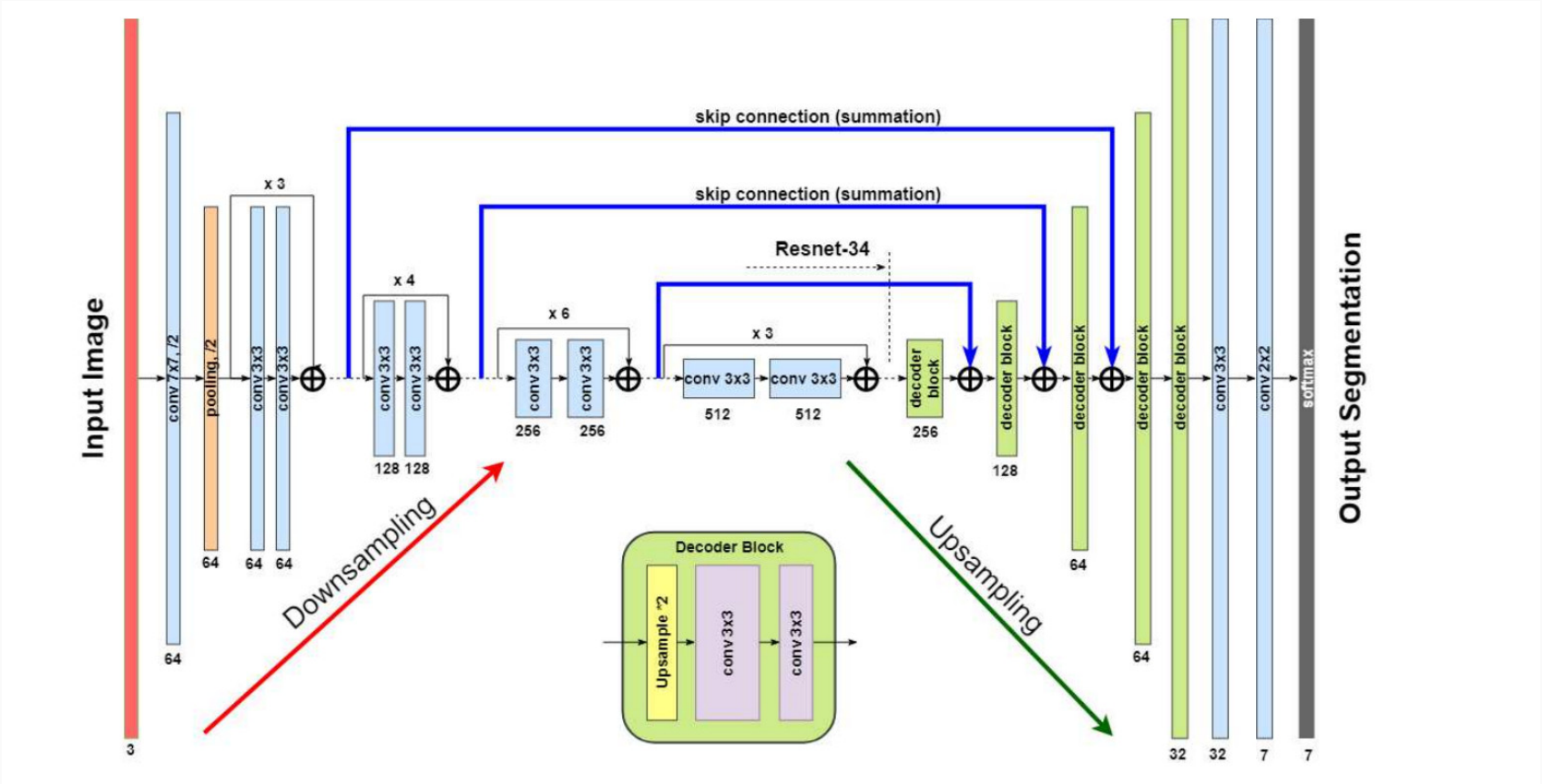
TernausNet и спутниковые снимки

- (Iglovikov et al., 2017): multispectral U-Net для Kaggle competition на спутниковых снимках:



TernausNet и спутниковые снимки

- (Rakhlin, Davydow, Nikolenko, 2018): U-Net для спутниковых снимков с ResNet-34



- Важный момент: какую функцию мы оптимизируем?

TernausNet и спутниковые снимки

- Обычно перекрёстную энтропию
- Но это не то, что нам нужно, нужно оптимизировать mIoU, т.е. средний коэффициент Жаккара:

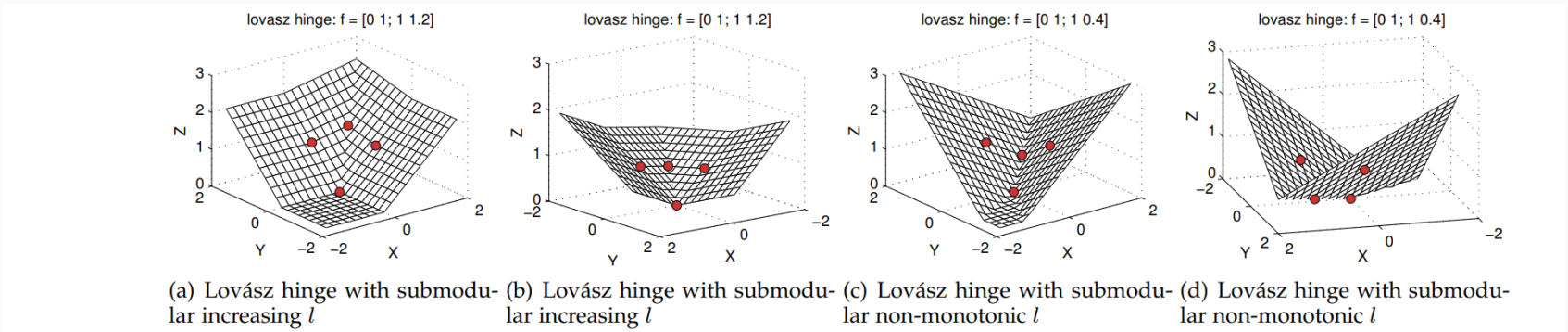
$$J_c(\hat{\mathbf{y}}, \mathbf{y}) = \frac{|\{\hat{\mathbf{y}} = c\} \cap \{\mathbf{y} = c\}|}{|\{\hat{\mathbf{y}} = c\} \cup \{\mathbf{y} = c\}|}.$$

- Но это, конечно, не продифференцируешь.
- Можно сделать мягкий вариант Жаккара:

$$J = \frac{1}{CN} \sum_c \sum_n \frac{y_n^c \hat{y}_n^c}{y_n^c + \hat{y}_n^c - y_n^c \hat{y}_n^c}.$$

TernausNet и спутниковые снимки

- Другая идея: давайте найдём выпуклую нижнюю оценку для Жаккара и её будем оптимизировать.
- (Yu, Blaschko, 2017) – Lovász hinge, нижняя оценка для субмодулярных функций:

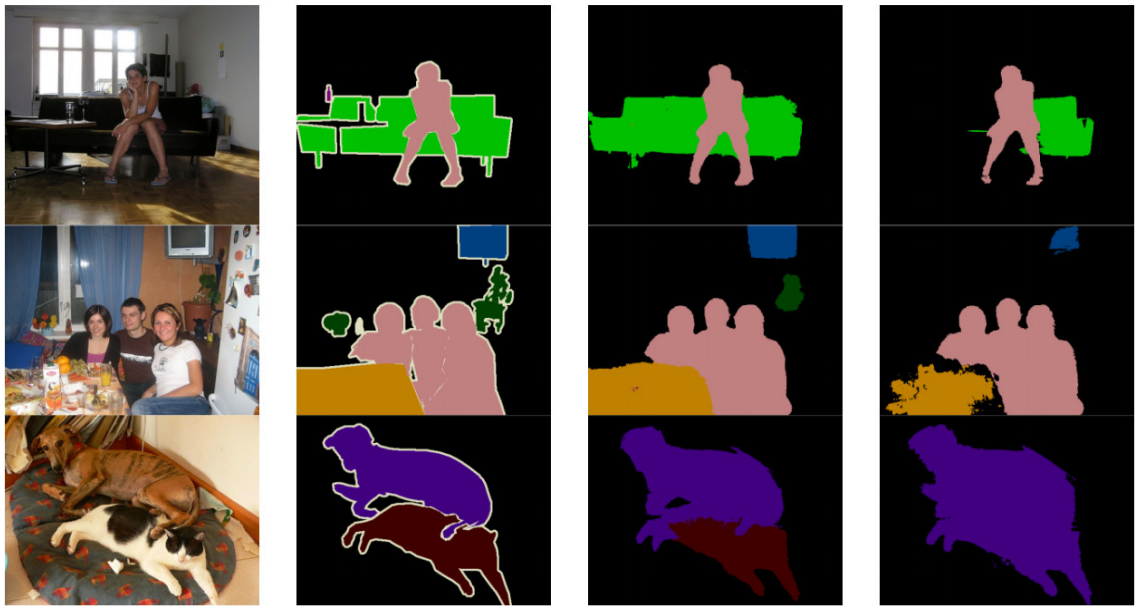


(a) Lovász hinge with submodular increasing l (b) Lovász hinge with submodular increasing l (c) Lovász hinge with submodular non-monotonic l (d) Lovász hinge with submodular non-monotonic l

- (Berman et al., 2018) – применили Lovász hinge для Жаккара с двумя классами и определили Lovász-Softmax loss, обобщение на несколько классов

TernausNet и спутниковые снимки

- (Berman et al., 2018) применили к сегментации:



(a) Input images (b) Ground truth masks (c) Lovász-Softmax + CRF (d) Cross-entropy + CRF

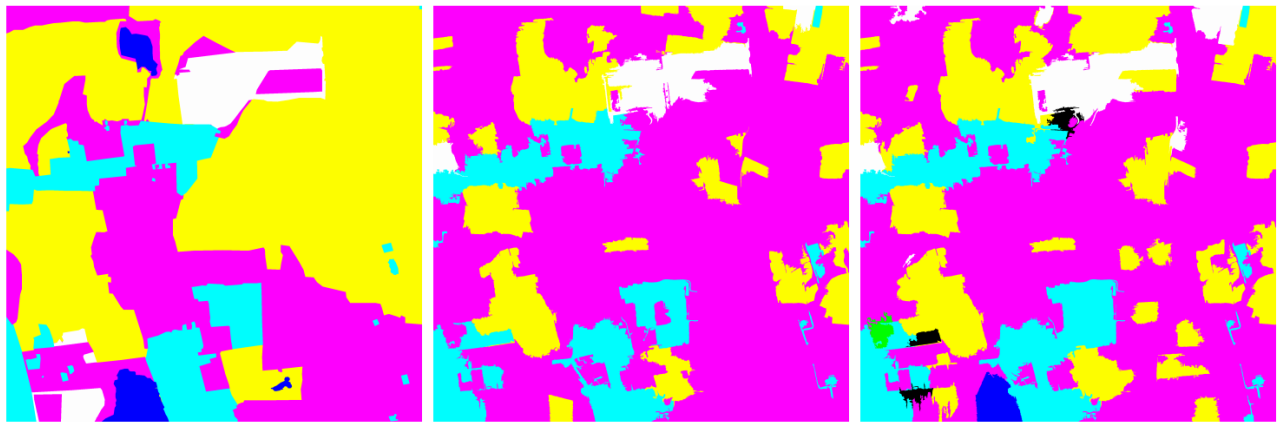
TernausNet и спутниковые снимки

- (Rakhlin, Davydow, Nikolenko, 2018): добавили Lovász-Softmax loss к U-Net, тоже хорошо стало



TernausNet и спутниковые снимки

- (Davydow, Nikolenko, 2018) – другая идея для оптимизации Жаккара: давайте сделаем postprocessing
- После сегментации мы знаем $p_x(c)$ для каждого суперпикселя x и класса c; можем оценить true positives и т.д.; давайте попробуем увеличивать ожидаемый Jaccard index, меняя значения суперпикселей по одному (суперпиксели ещё важны, если данные не слишком точные)



(a) (b) (c)

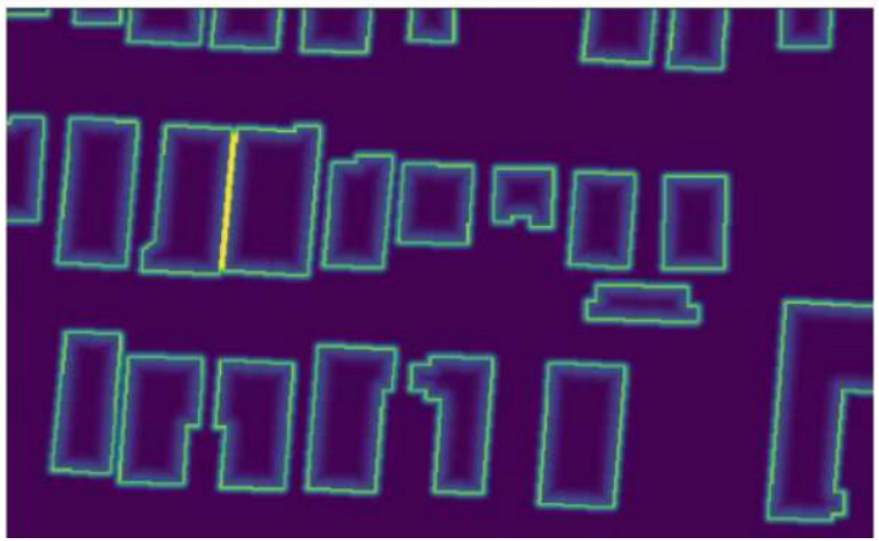
Figure 3. Sample segmentation results. Left to right: (a) ground truth mask; (b) model result before postprocessing, average Jaccard index 0.317, accuracy 0.572; (c) model result after postprocessing, average Jaccard index 0.393, accuracy 0.584.

TernausNet и спутниковые снимки

- (Golovanov et al., 2018) – как сегментировать здания на карте? Проблема в том, что границы трудно провести.
- Идея: давайте добавим веса в кросс-энтропию, которые будут зависеть от расстояния до ближайшей границы:

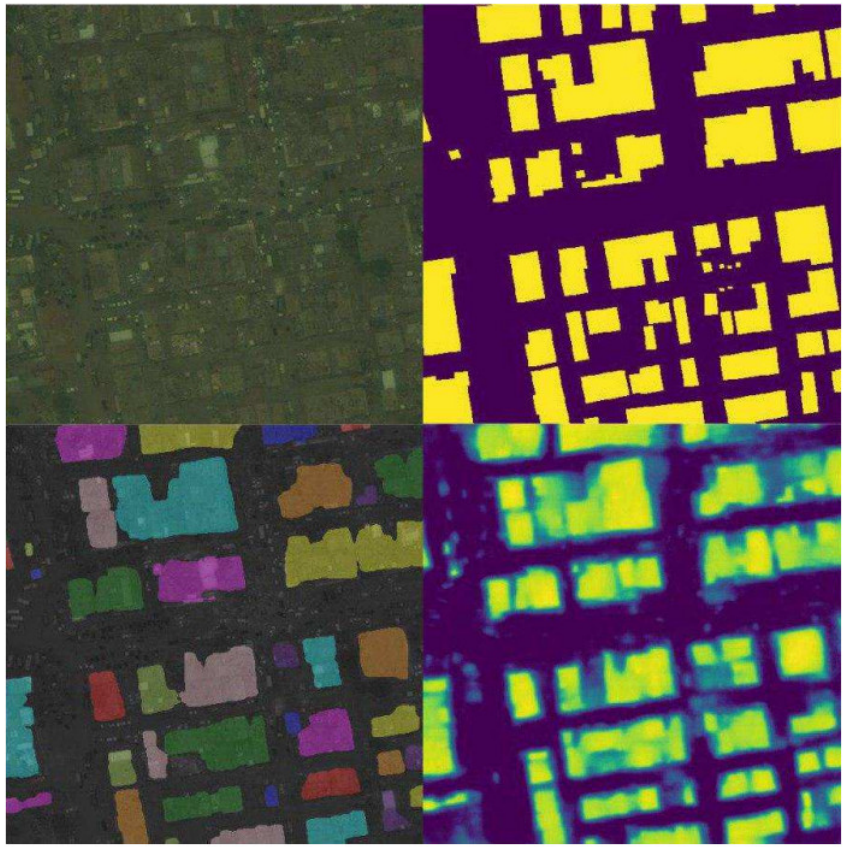
$$w(x) = w_c(x) + w_0 e^{-\frac{1}{2\sigma^2}(d_1(x)+d_2(x))^2} + w_0 \frac{d_3}{2r_d},$$

где d_1 – расстояние до ближайшего здания, d_2 – второго ближайшего, d_3 – до ближайшей узкой границы.



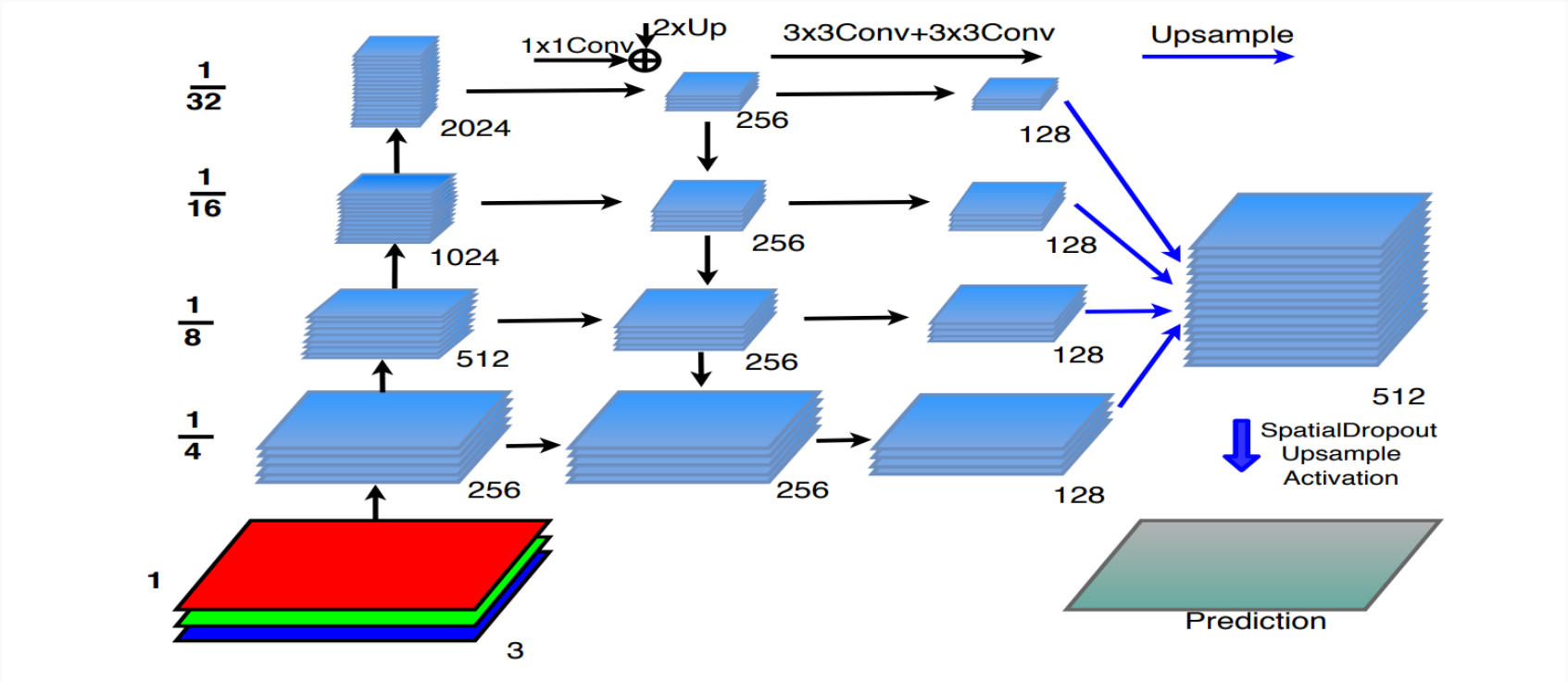
TernausNet и спутниковые снимки

- Помогает:



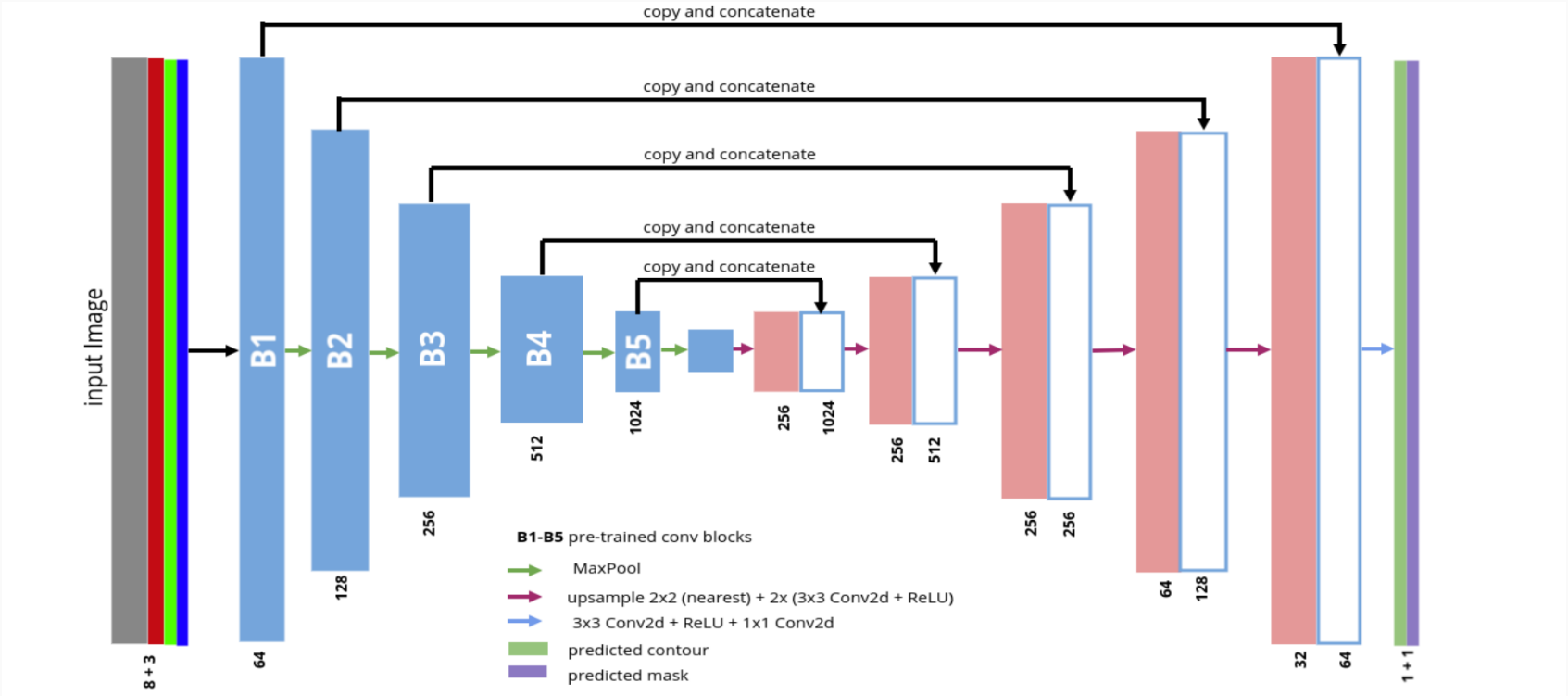
TernausNet и спутниковые снимки

- Кстати, в том конкурсе о зданиях всё тот же Игловиков адаптировал feature pyramid networks (Seferbekov et al., 2018):



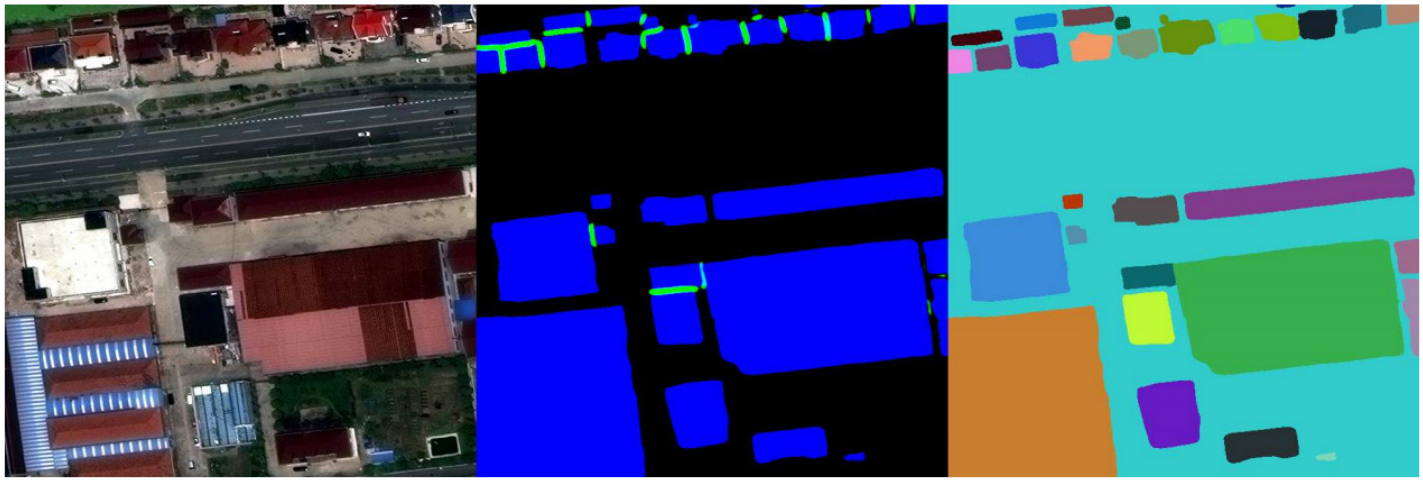
TernausNet и спутниковые снимки

- (Iglovikov et al., 2018): TernausNetV2; WideResNet-38 вместо VGG



TernausNet и спутниковые снимки

- Кроме того, RGB дополнено 11 multispectral channels, а на выходе есть отдельный канал, который предсказывает места, где объекты очень близки или касаются; это позволяет разделить здания и сделать instance segmentation:



TernausNet и спутниковые снимки

- Сейчас TernausNet и подобные архитектуры активно применяют в medical imaging.
- (Rakhlin et al., 2018): pediatric bone age assessment

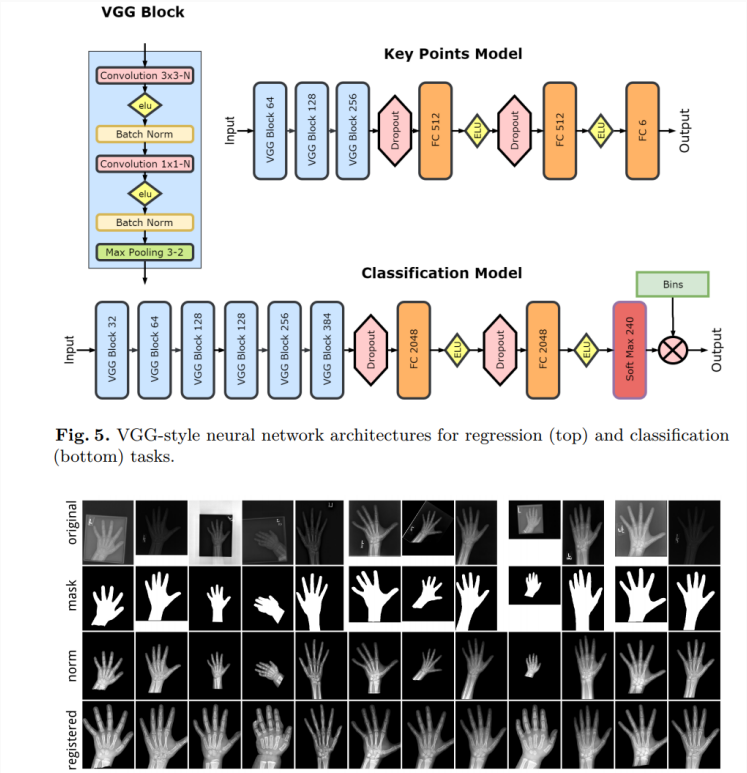
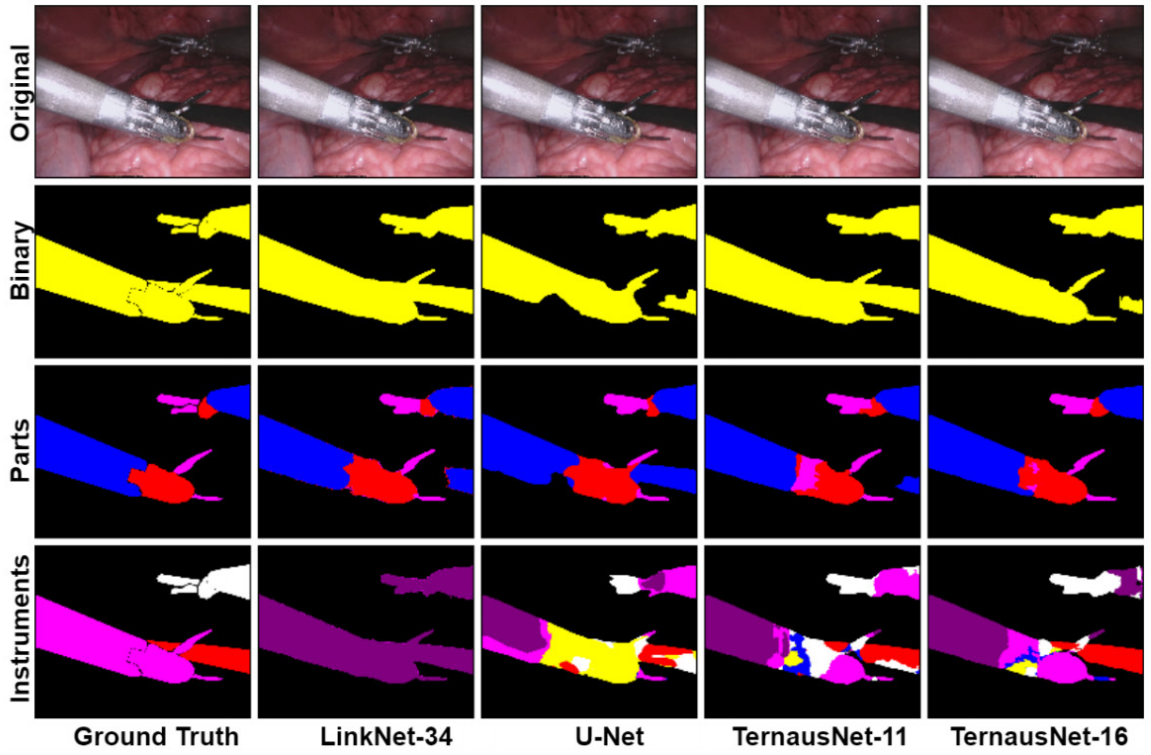


Fig. 5. VGG-style neural network architectures for regression (top) and classification (bottom) tasks.

TernausNet и спутниковые снимки

- (Shvets et al., 2018): сегментация инструментов для роботов-хирургов



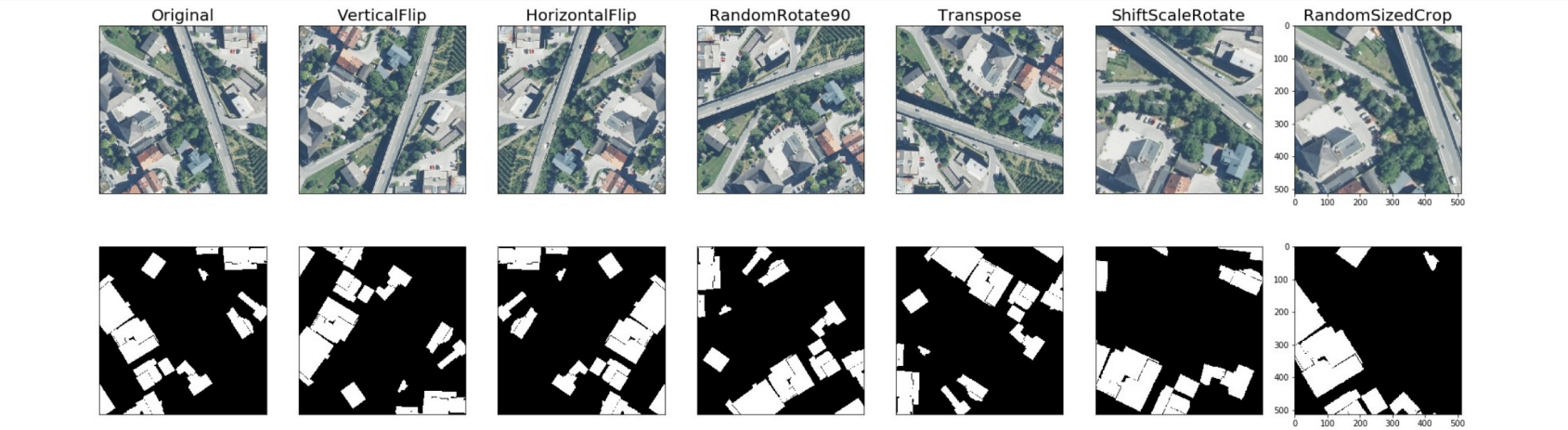
Albumentations

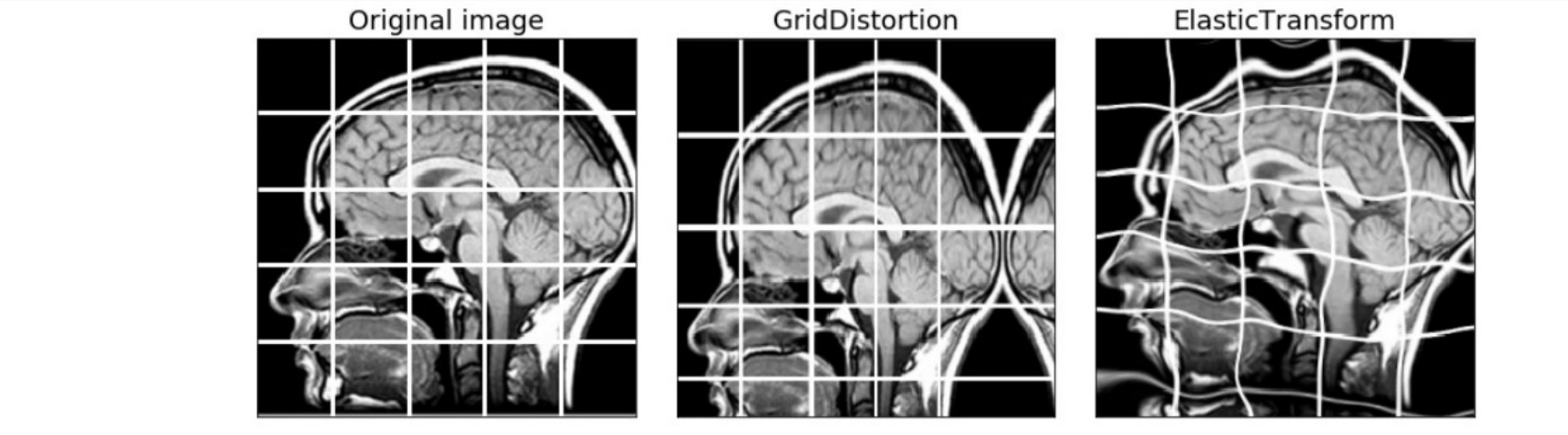
- Ну и раз мы об Игловикове, небольшое отступление
- Albumentations (Buslaev et al., 2018) – наверное, лучшая библиотека для data augmentation:



Albumentations

- Геометрические преобразования:





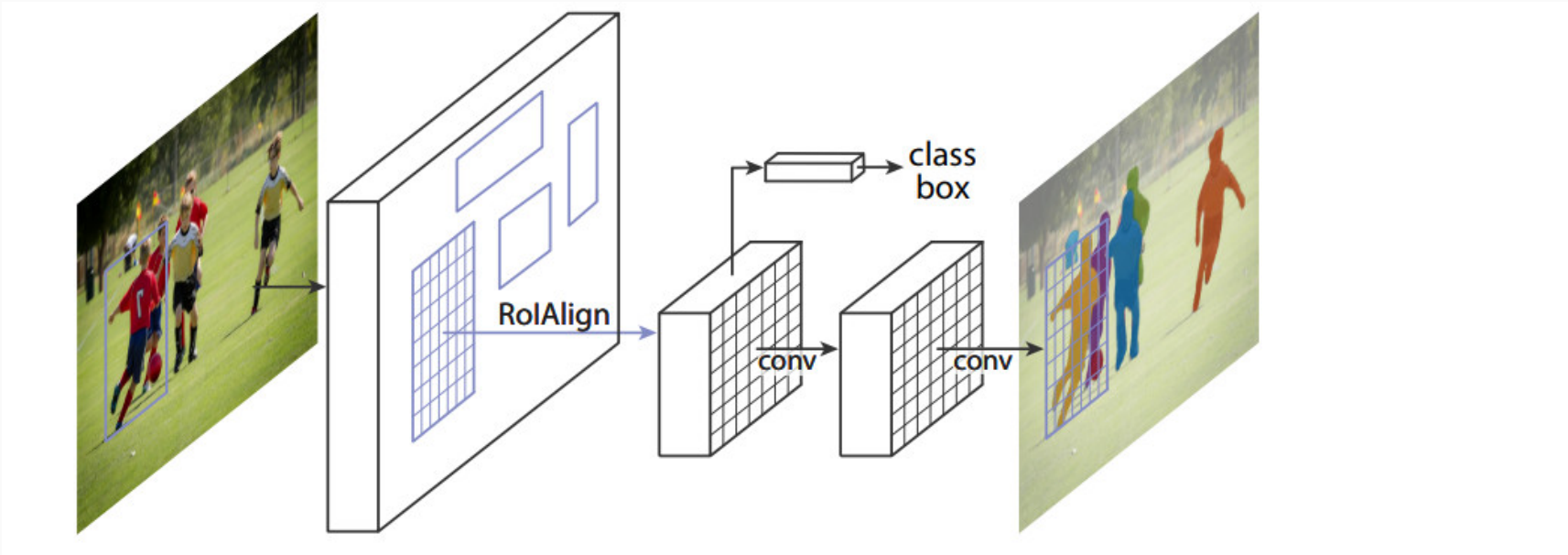
Albumentations

- Всё можно применить и к bbox'ам, и к маскам:



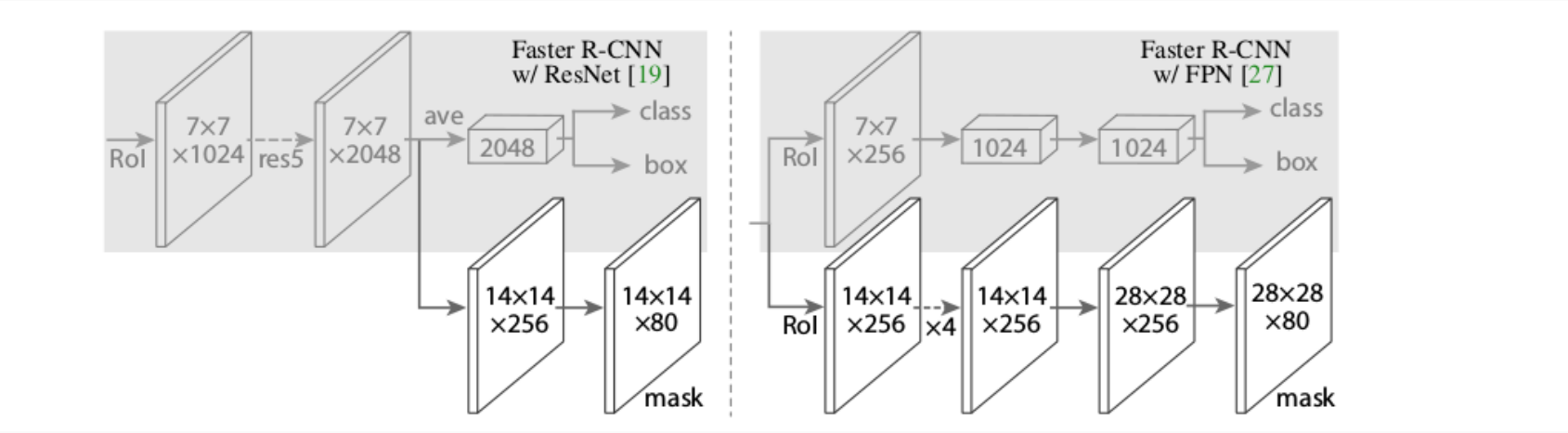
Mask R-CNN

- В прошлый раз мы много говорили о Faster R-CNN; оказывается, его можно адаптировать для сегментации:
  - у Faster R-CNN два выхода: метка класса и сдвиг bounding box;
  - а теперь давайте добавим третий выход, который считает прямо маску объекта, попиксельно.
- Получается *Mask R-CNN* (He et al., 2017).



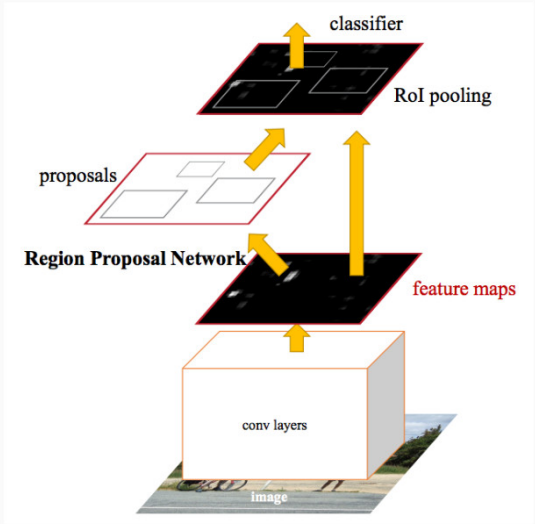
Mask R-CNN

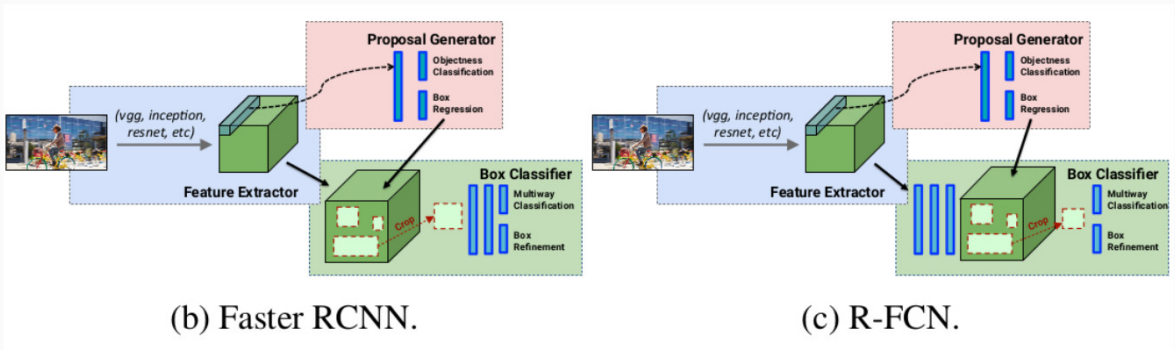
- Добавляем ошибку за маску (средняя кросс-энтропия по пикселям).
- Это можно добавить к разным базовым вариантам Faster R-CNN.



Mask R-CNN

- В принципе это очень прямолинейное расширение. Но есть тонкость: Faster R-CNN не приспособлен для точной локализации.





(b) Faster RCNN. (c) R-FCN.

Mask R-CNN

- RoIPool необратим и сильно уменьшает размер сетки
- В Mask R-CNN добавляется слой RoIAlign, который не делает квантование, а делает билинейную интерполяцию:

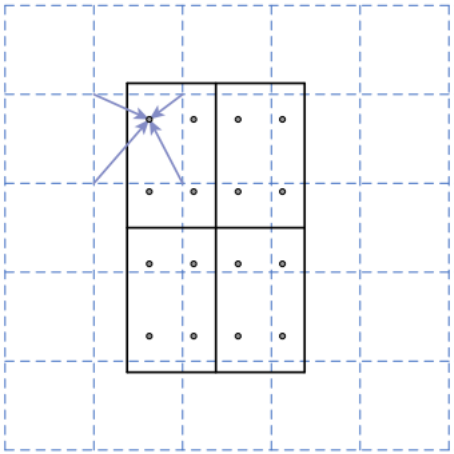
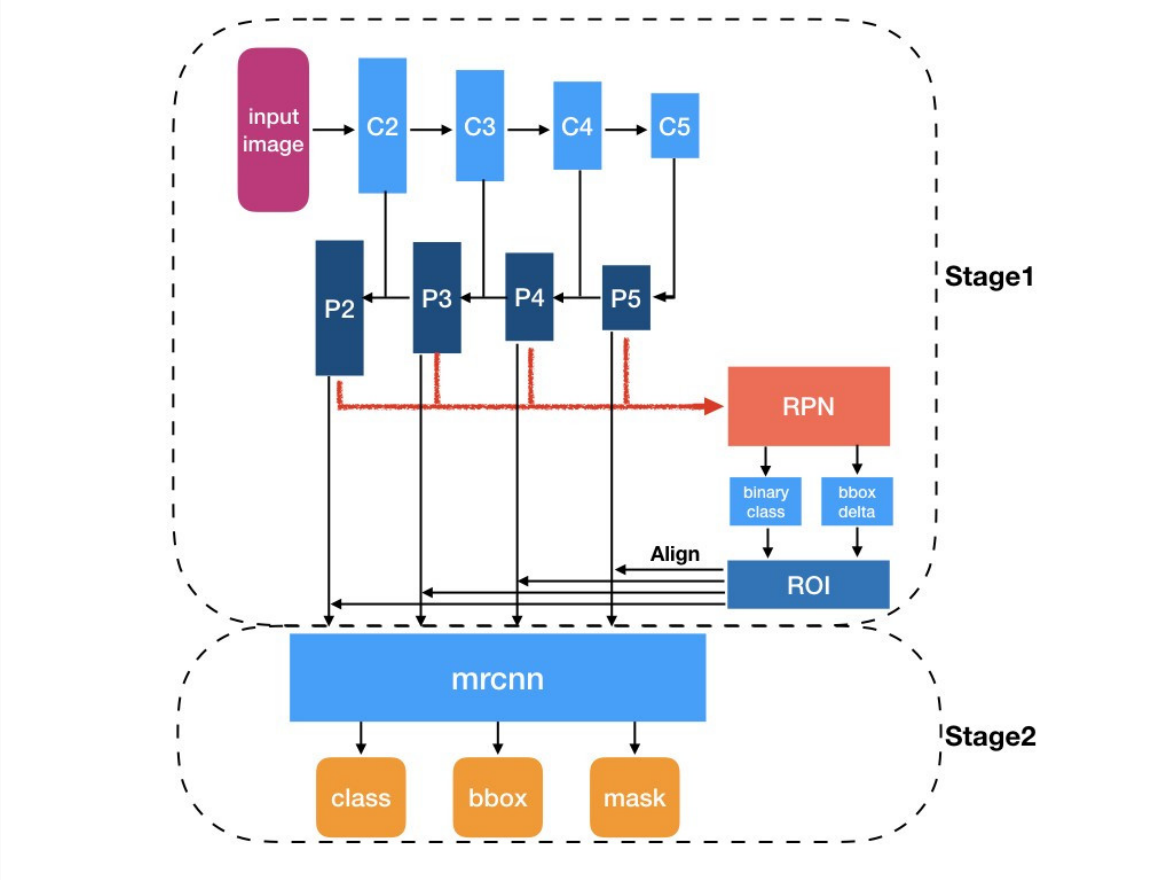


Figure 3. **RoIAlign:** The dashed grid represents a feature map, the solid lines an RoI (with 2×2 bins in this example), and the dots the 4 sampling points in each bin. RoIAlign computes the value of each sampling point by bilinear interpolation from the nearby grid points on the feature map. No quantization is performed on any coordinates involved in the RoI, its bins, or the sampling points.

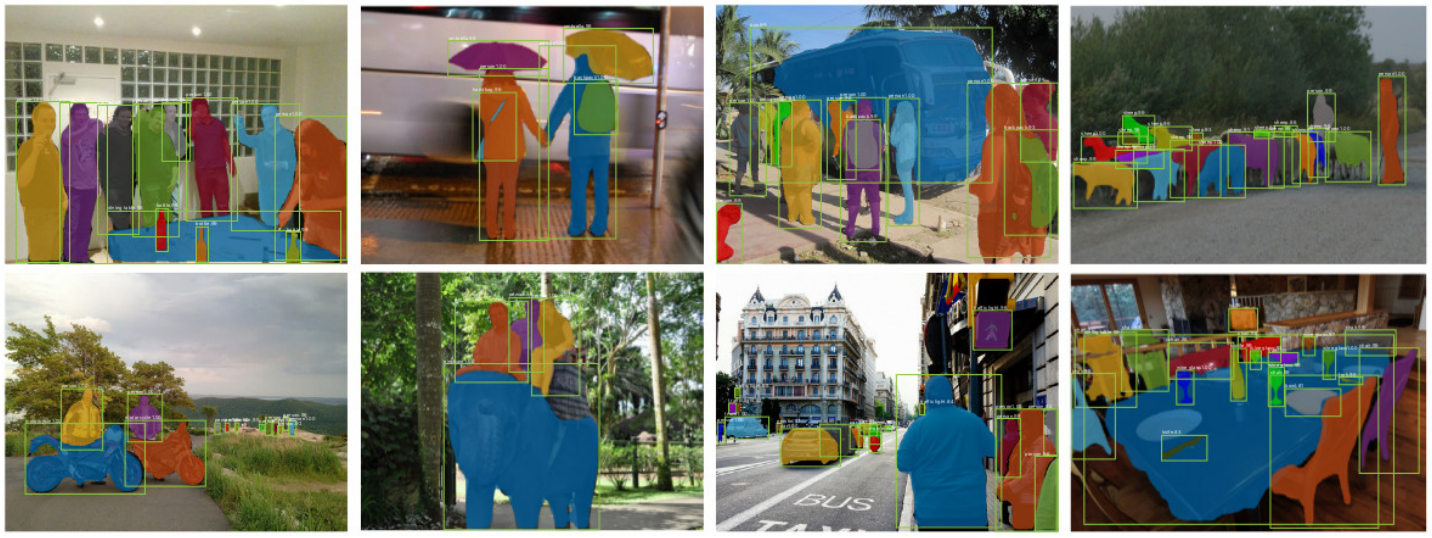
Mask R-CNN

- И его теперь можно вернуть на разные уровни признаков, чтобы распознавать объекты разных размеров:



Mask R-CNN

- Получается очень хорошо.





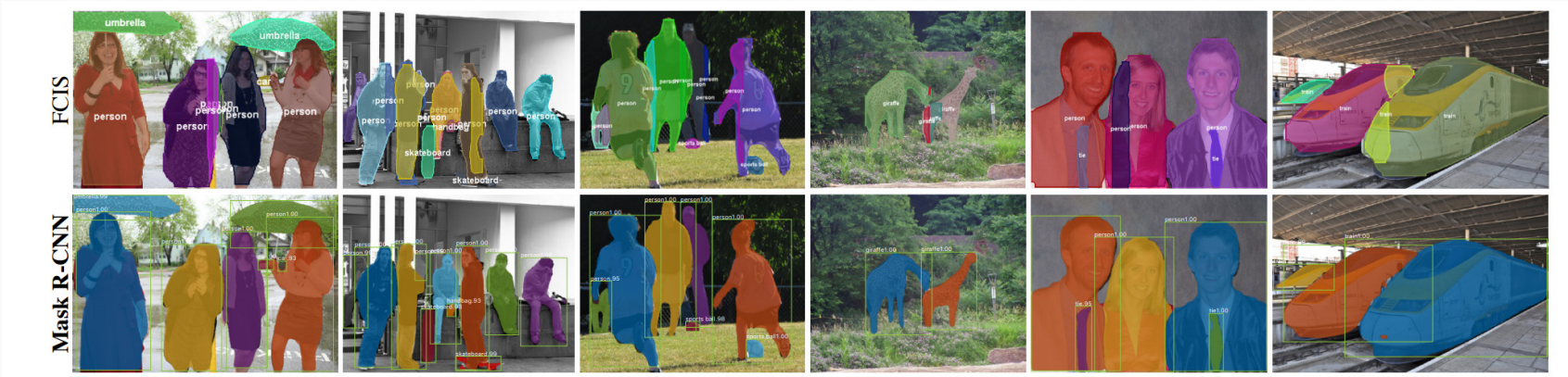
Mask R-CNN

- Ещё примеры из COCO:



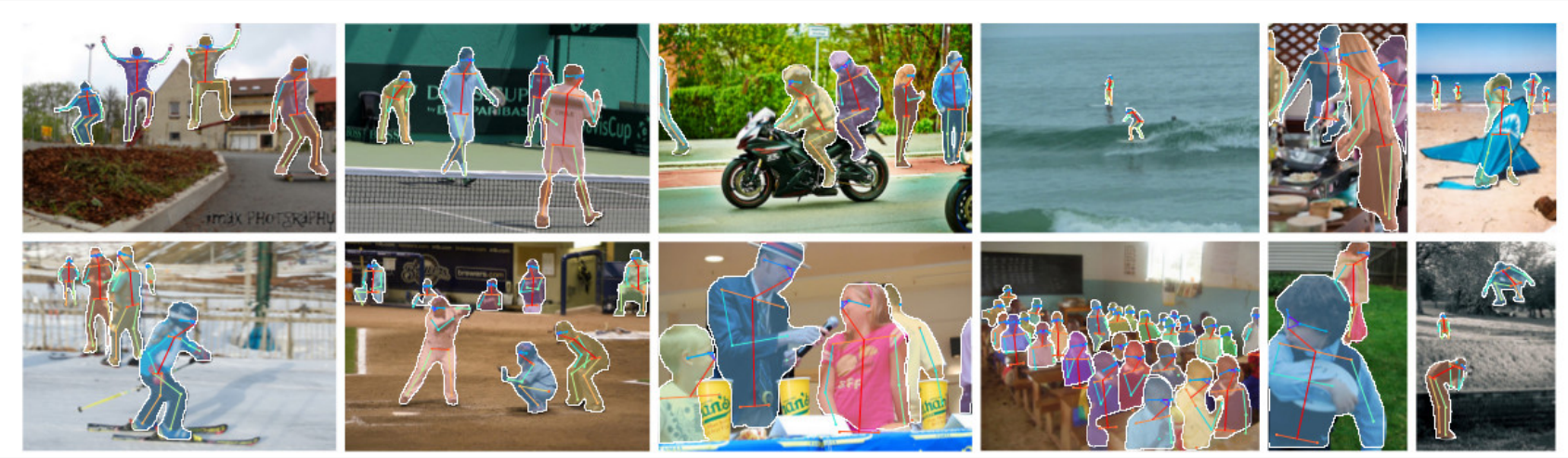
Mask R-CNN

- Причём очень хорошо работает именно instance segmentation, у конкурентов пересекающиеся объекты плохо обрабатываются:



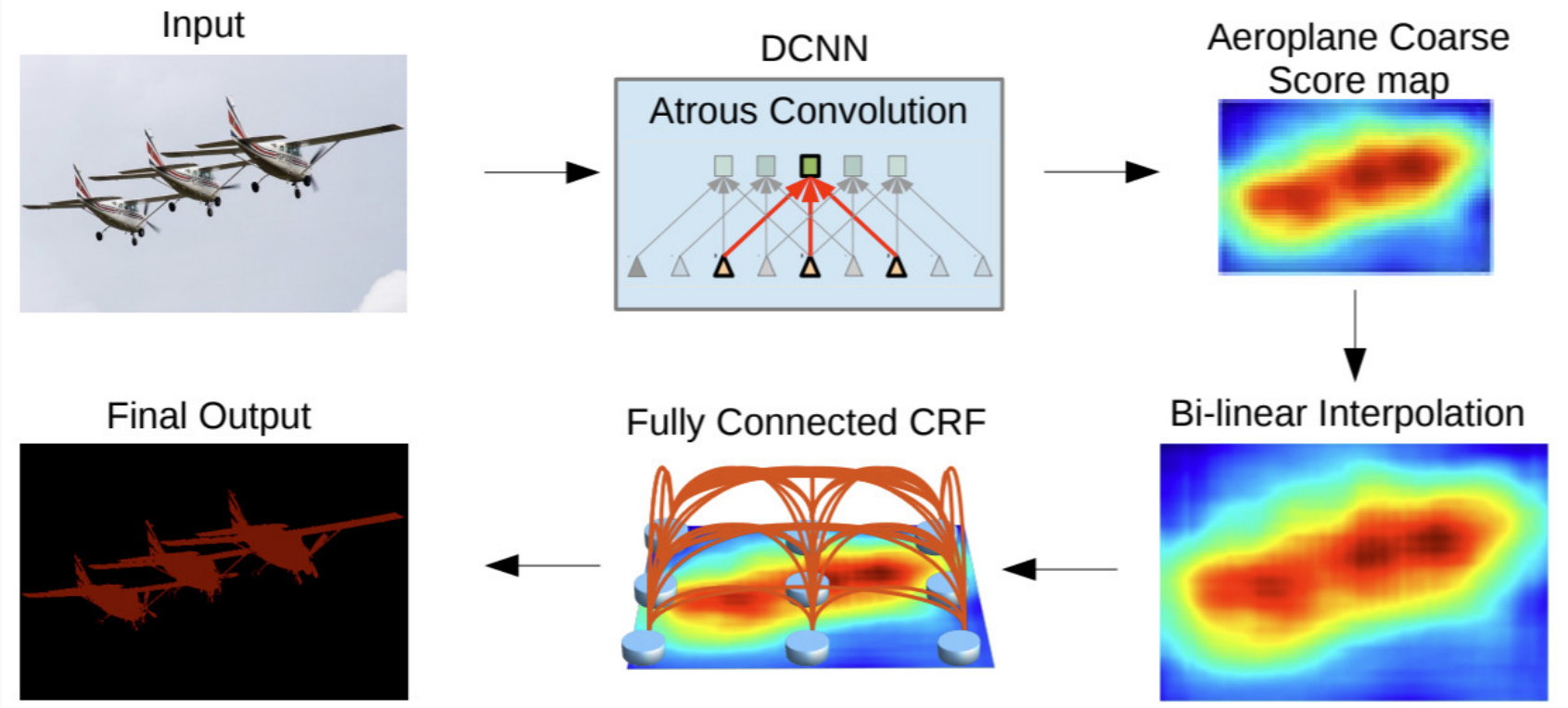
Mask R-CNN

- А можно, например, вместо маски распознавать ключевые точки позы человека.
- Гибкая архитектура, можно любые нужные пиксели пытаться вставить и начать распознавать.



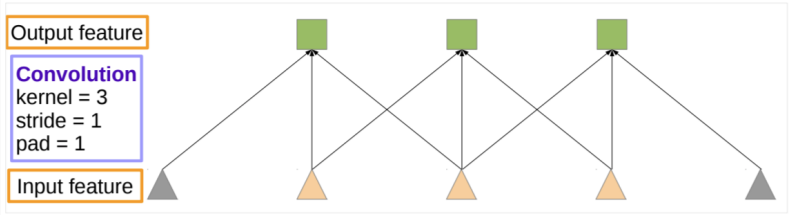
DeepLab

- DeepLab (Chen et al., 2016): разреженные свёртки + CRF

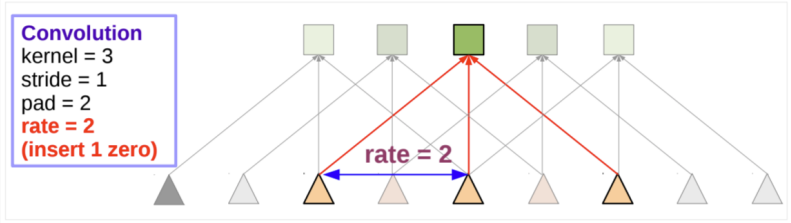


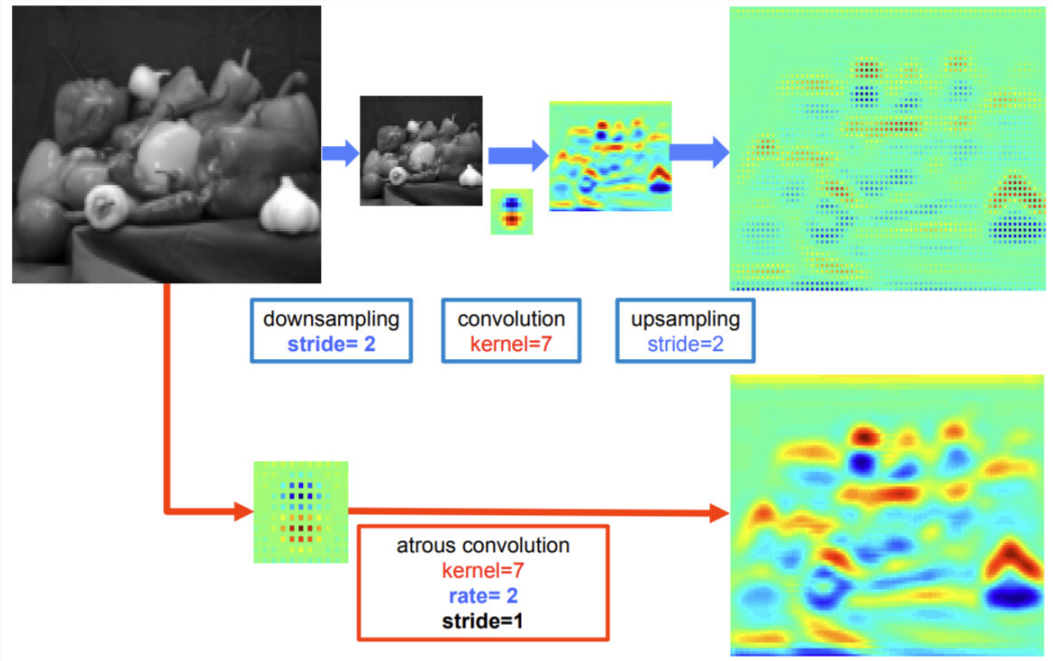
- CRF до сих пор остаются важной моделью (позже поговорим о них подробнее)

- Разреженные свёртки (atrous, or dilated convolutions):

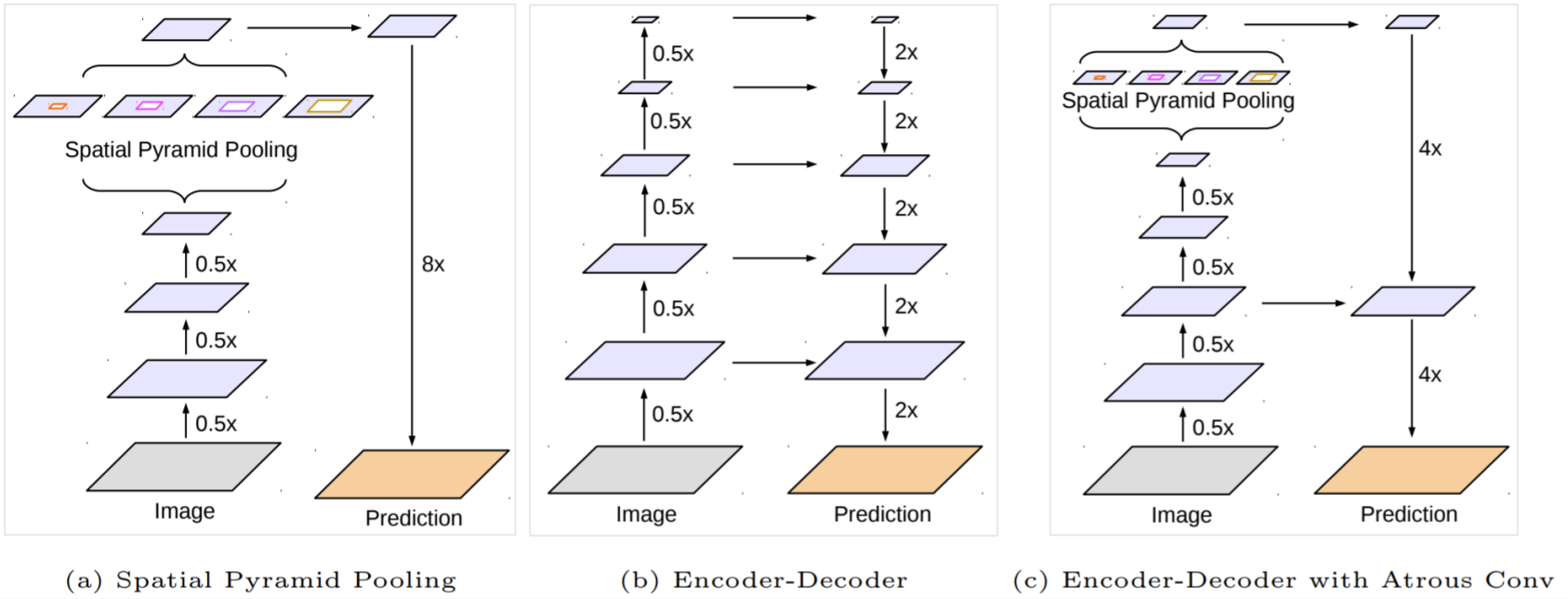


(a) Sparse feature extraction





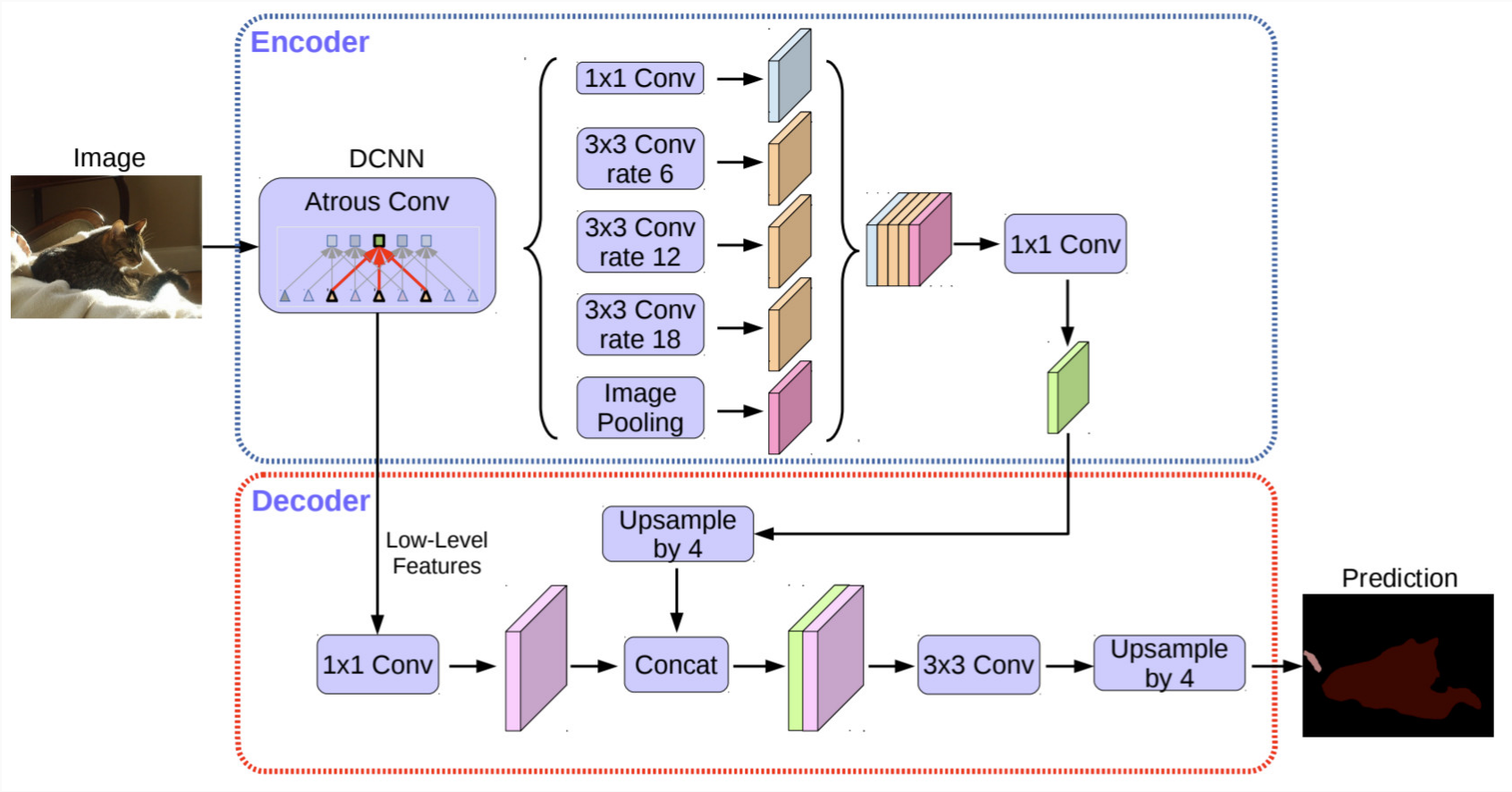
- DeepLab v3+ (Chen et al., 2018): от пирамидального пулинга к encoder-decoder



(a) Spatial Pyramid Pooling (b) Encoder-Decoder (c) Encoder-Decoder with Atrous Conv

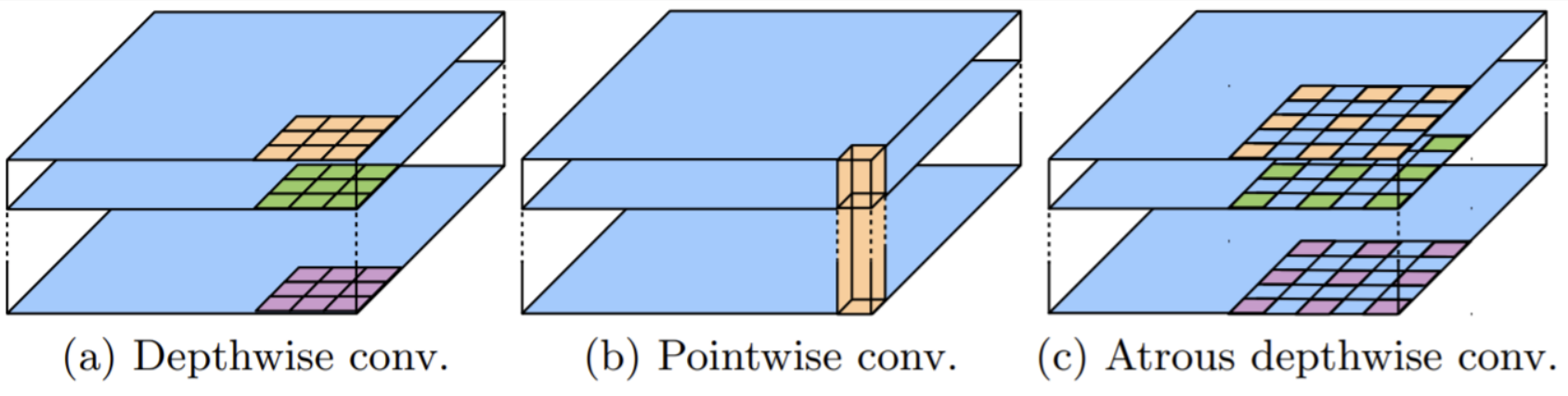
DeepLab

- DeepLab v3+ (Chen et al., 2018): архитектура



DeepLab

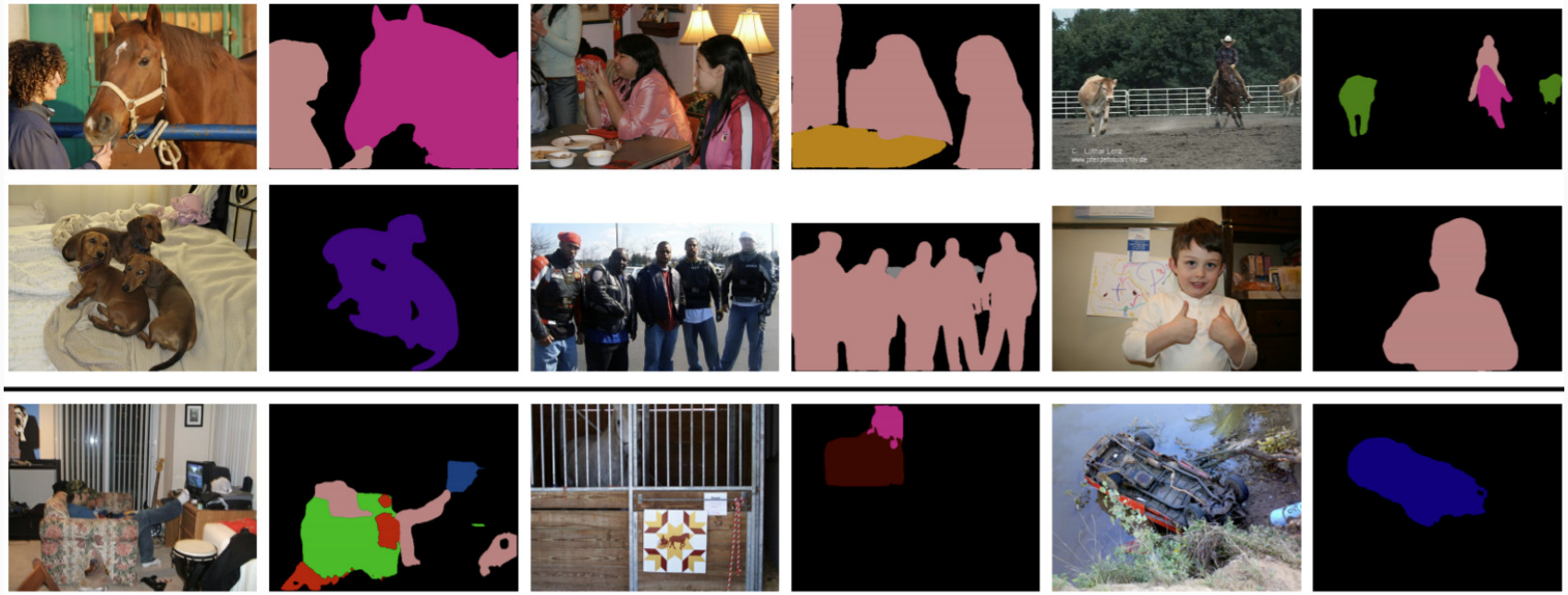
- DeepLab v3+ (Chen et al., 2018): depthwise separable convolutions



- Плюс модификации модуля Xception, плюс ещё куча маленьких улучшений...

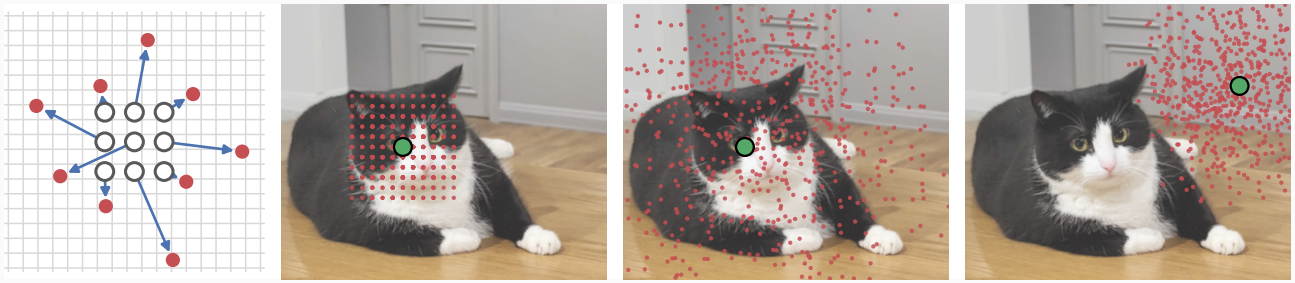
DeepLab

- DeepLab v3+ (Chen et al., 2018): хорошо работает и до сих пор составляет разумный baseline



Деформируемые свёртки

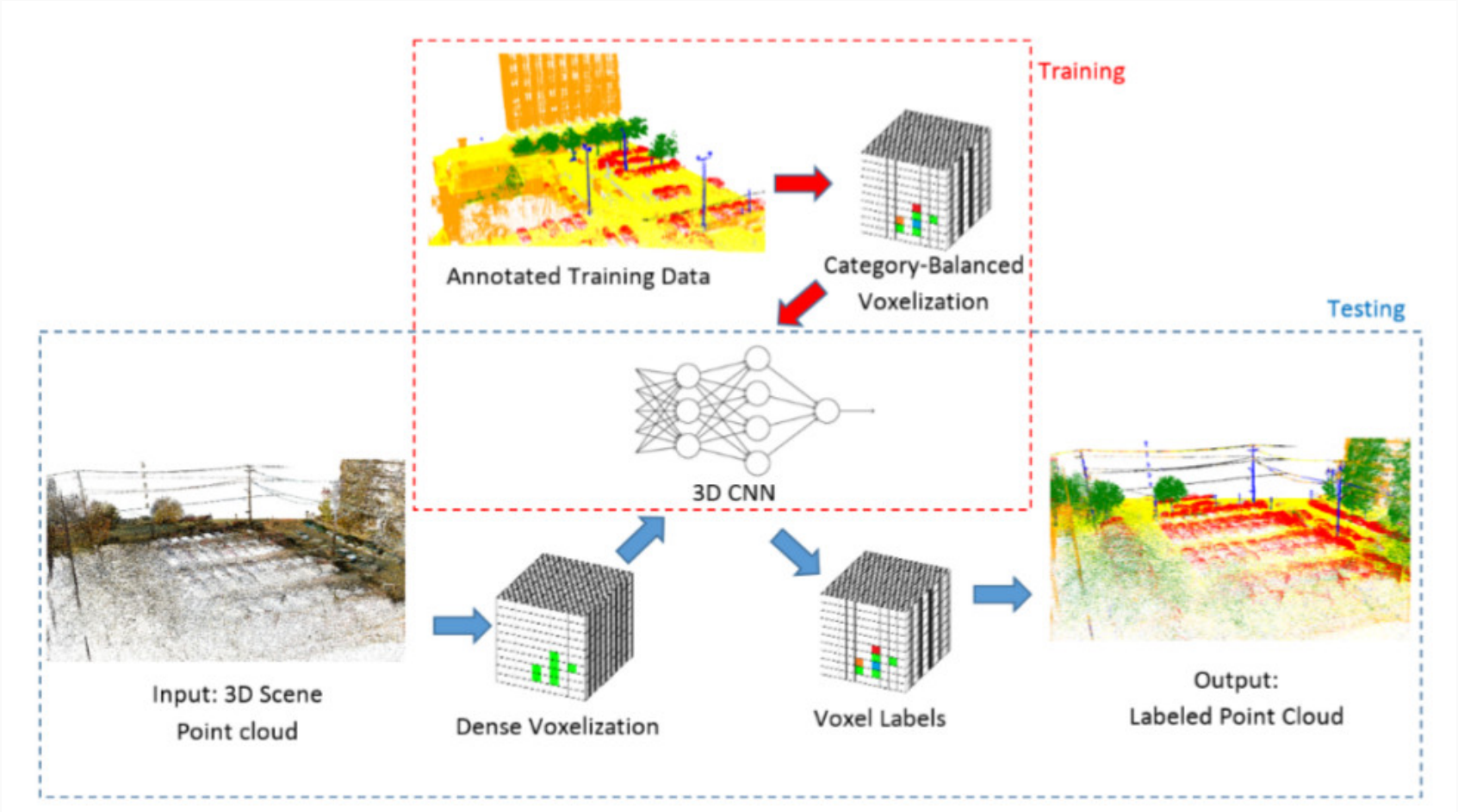
- Разреженные свёртки фиксируют сетку отсчётов заранее; следующий шаг — её *выучить*.
- Deformable convolutions (Dai et al., 2017): к каждой из k^2 точек ядра добавляется выученный сдвиг; его выдаёт ещё одна свёртка, а значение берётся билинейной интерполяцией, так что всё дифференцируемо.
- Эксперимент: ResNet-50 с деформируемыми свёртками дообучили на Oxford-IIIT Pet размечать пиксели породой питомца; смотрим на моего кота:



- Сдвиги одного фильтра; откуда берёт данные нейрон последней стадии без сдвигов, на коте и на стене.
- Порода верна у 91% пикселей питомцев против 83% у обычных свёрток.

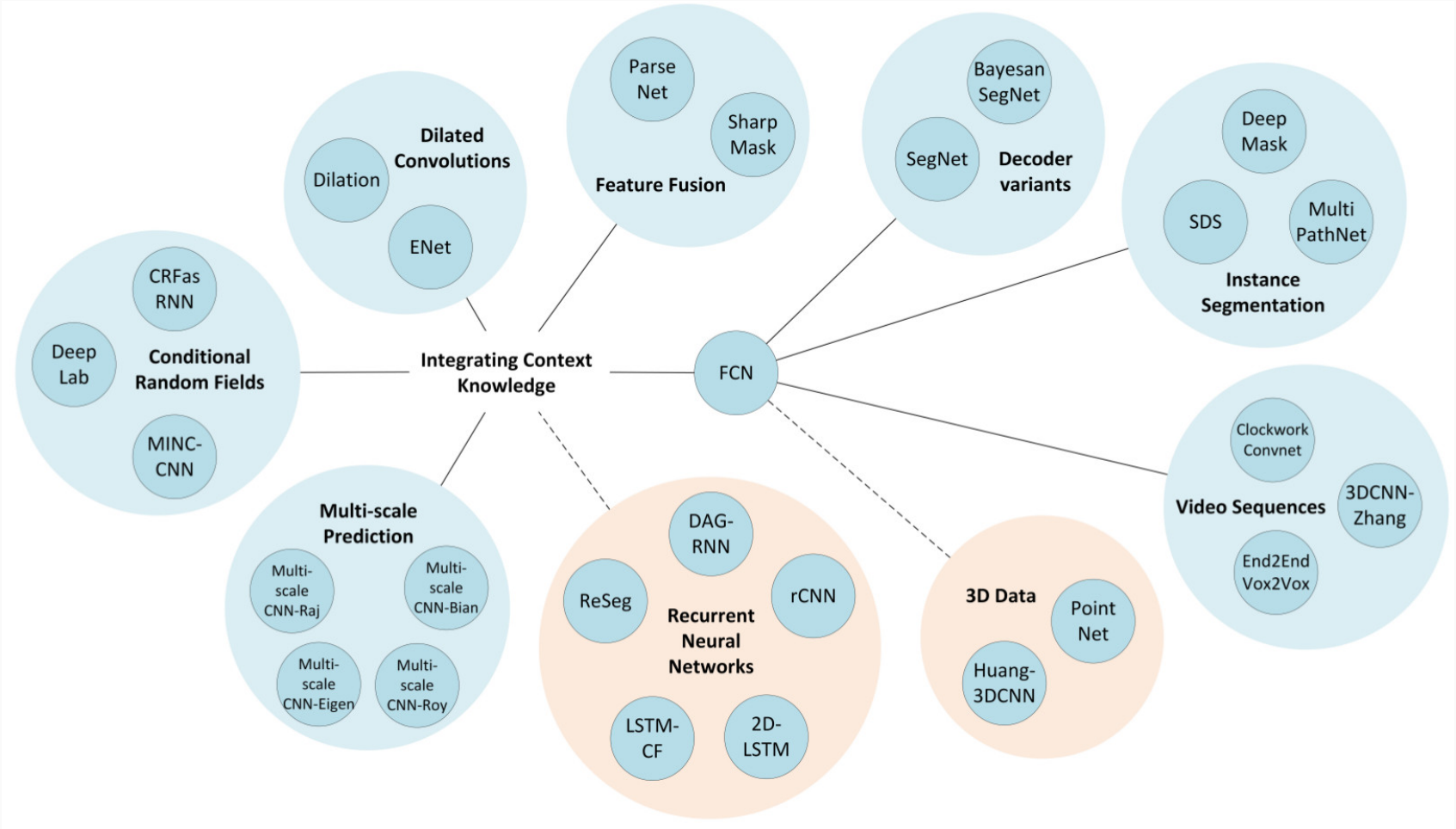
Сегментация в 3D

- А мы ещё совсем не коснулись сегментации в 3D:



Обзор

- И всё это всего лишь небольшая часть ландшафта сегментации, всего лишь примеры...

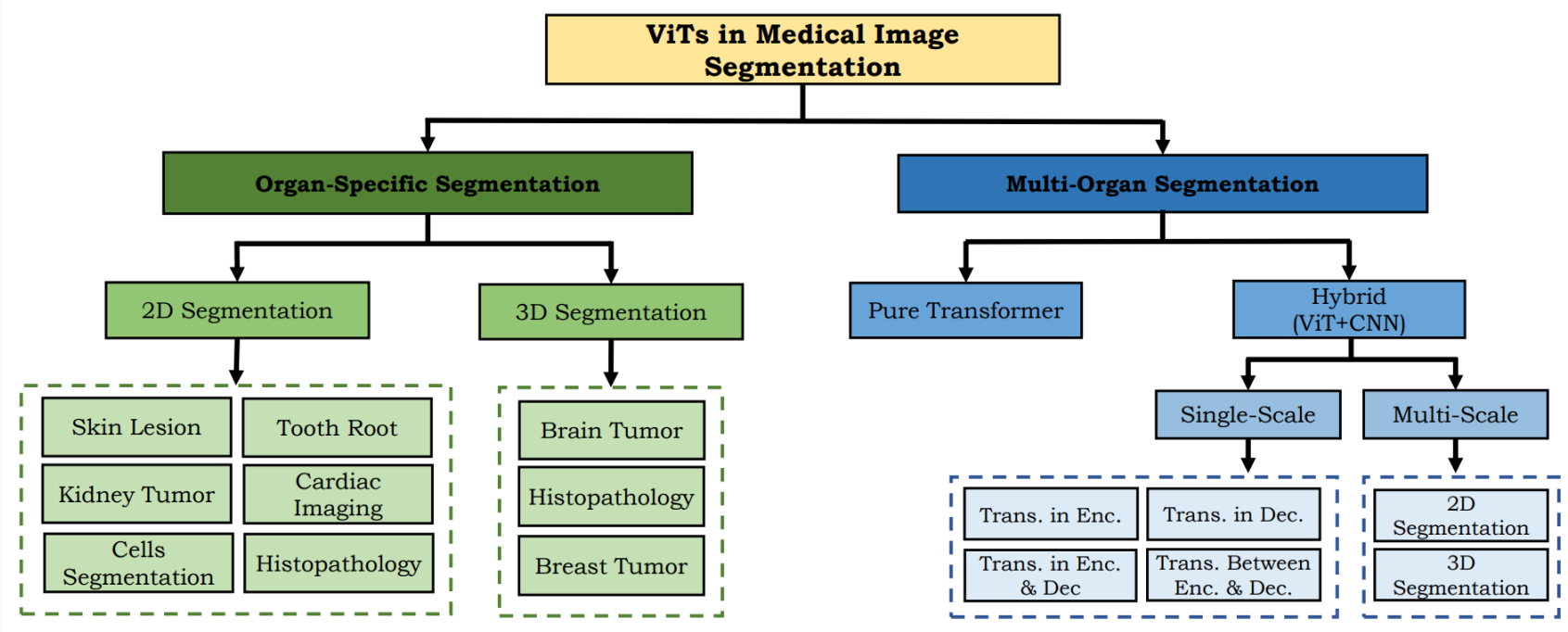


- А сейчас всё опять изменилось из-за трансформеров

Case study: сегментация медицинских изображений

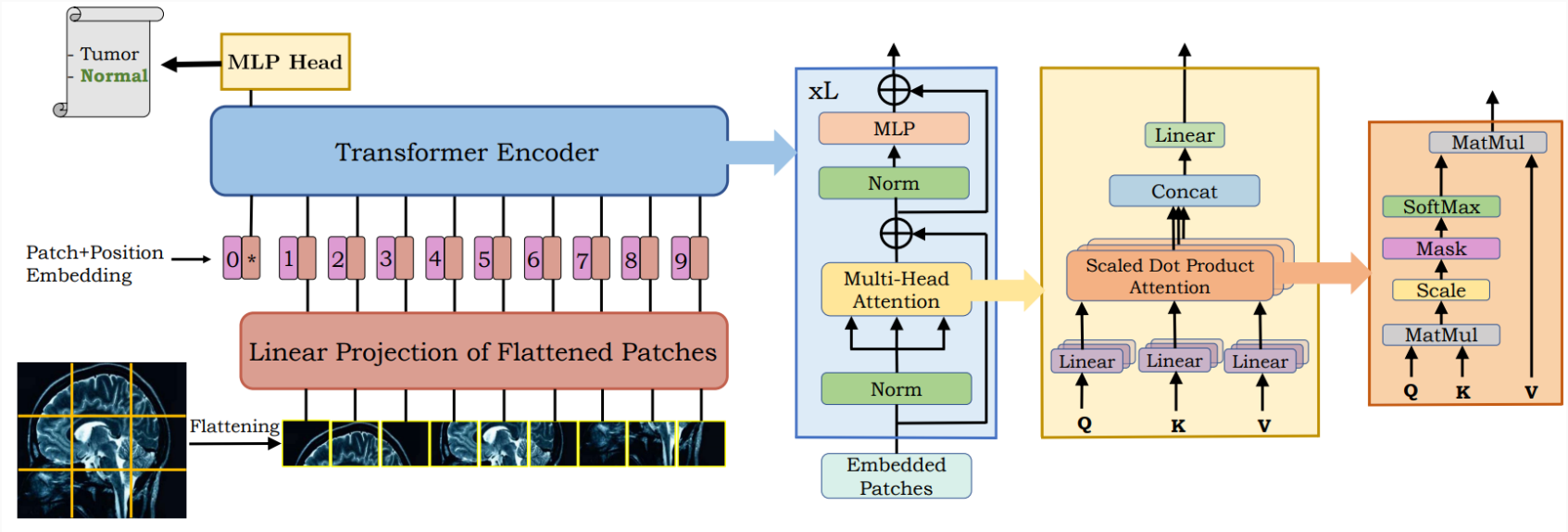
Сегментация медицинских изображений

- Сегментация звучит как более сложная задача для трансформеров
- Но вот таксономия только для ViT, только в сегментации медицинских снимков, причём три года назад (Shamshad et al., 2022):



Сегментация медицинских изображений

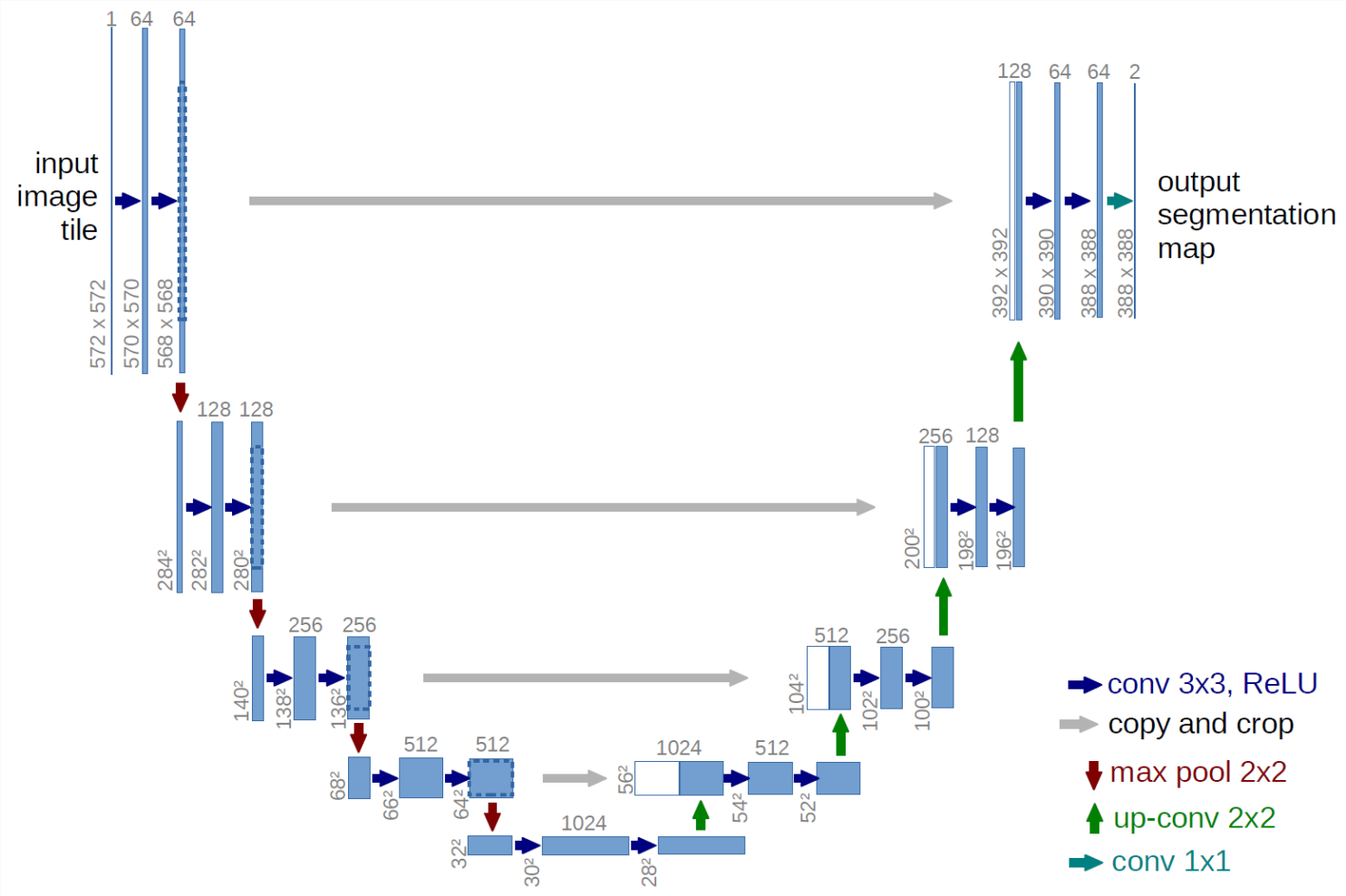
- В медицинских задачах *маленькие датасеты*, поэтому трудно ожидать ViT, обученный с нуля на медицинском датасете



- Иногда это возможно, но часто приходится модифицировать архитектуру, чтобы добавить априорную информацию о медицинских особенностях

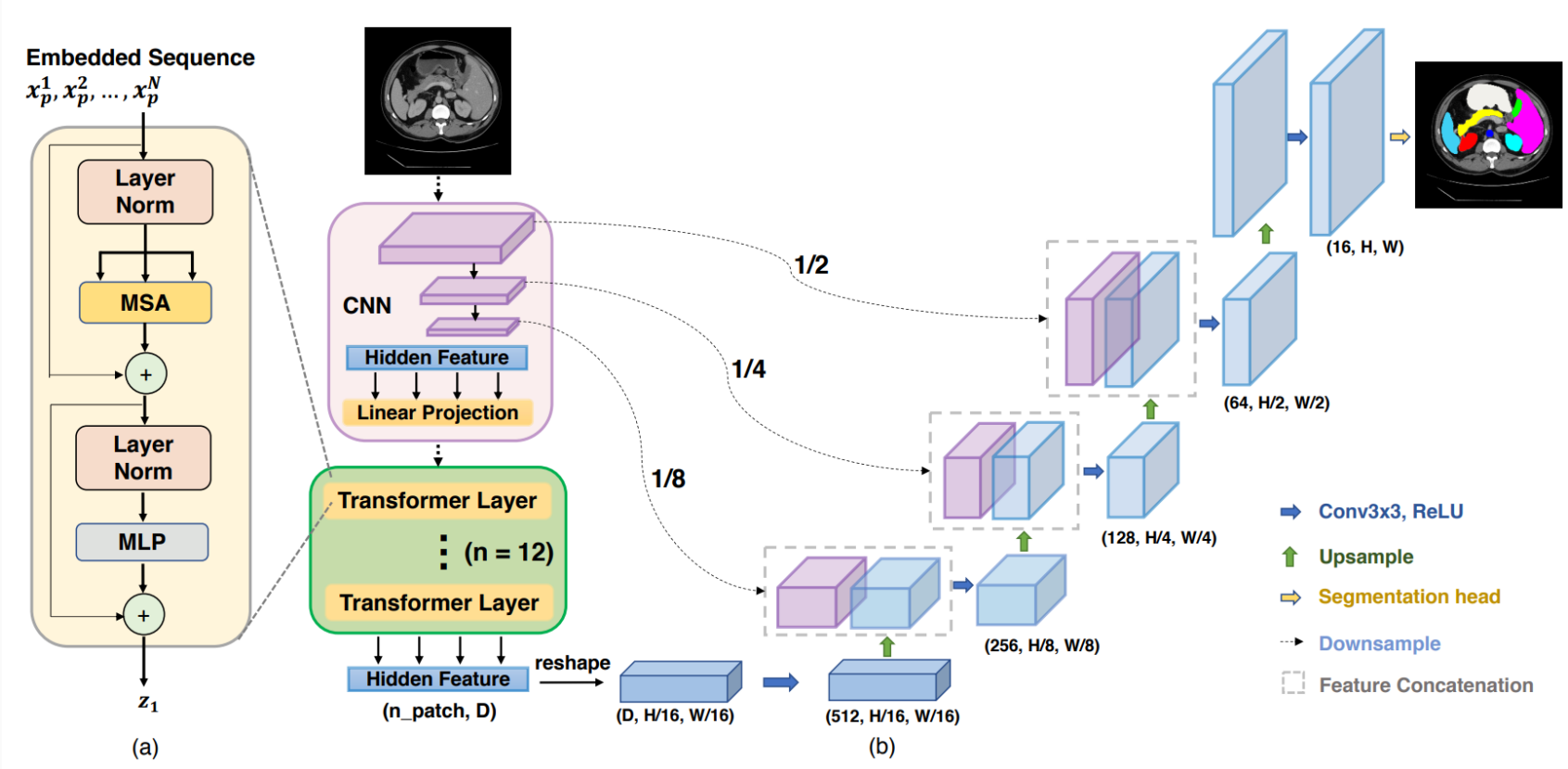
Сегментация медицинских изображений

- В сегментации U-Net долгое время была самой важной архитектурой для medical imaging:



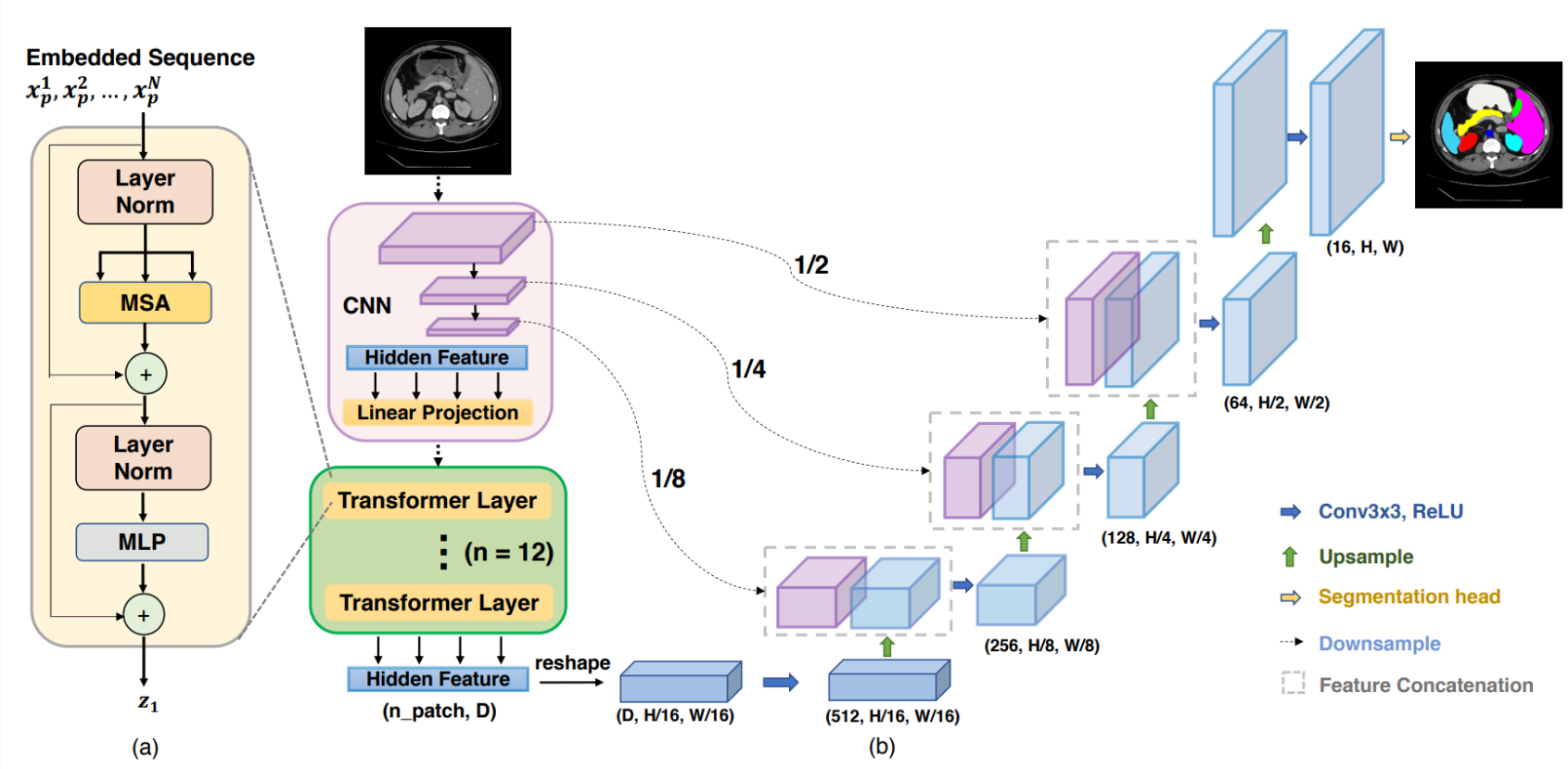
Сегментация медицинских изображений

- TransUNet (Chen et al., 2021): давайте используем трансформеры в кодировщике U-Net



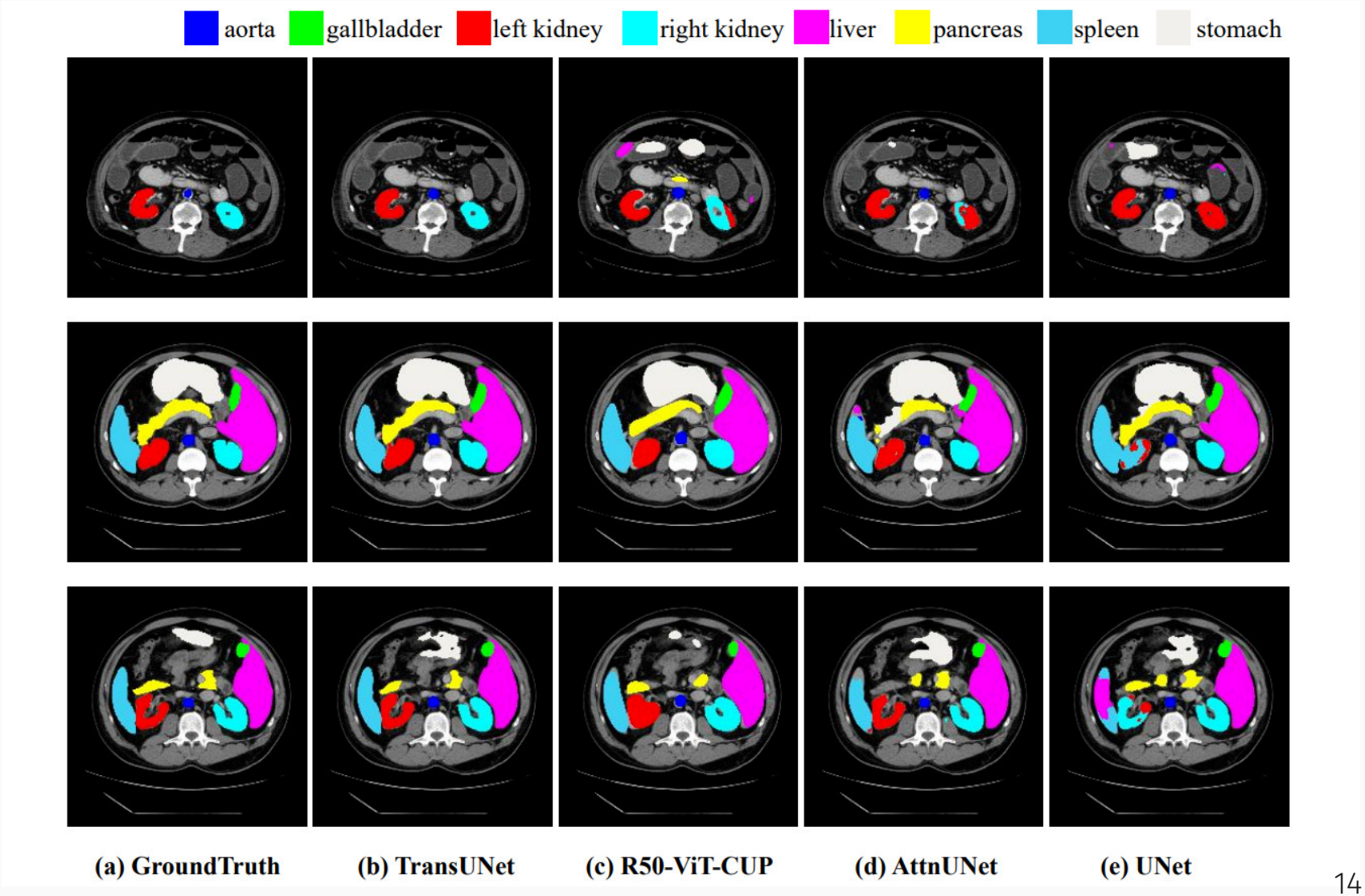
Сегментация медицинских изображений

- ViT получает вложения патчей из разных слоев CNN, а повышение разрешения аналогично U-Net:



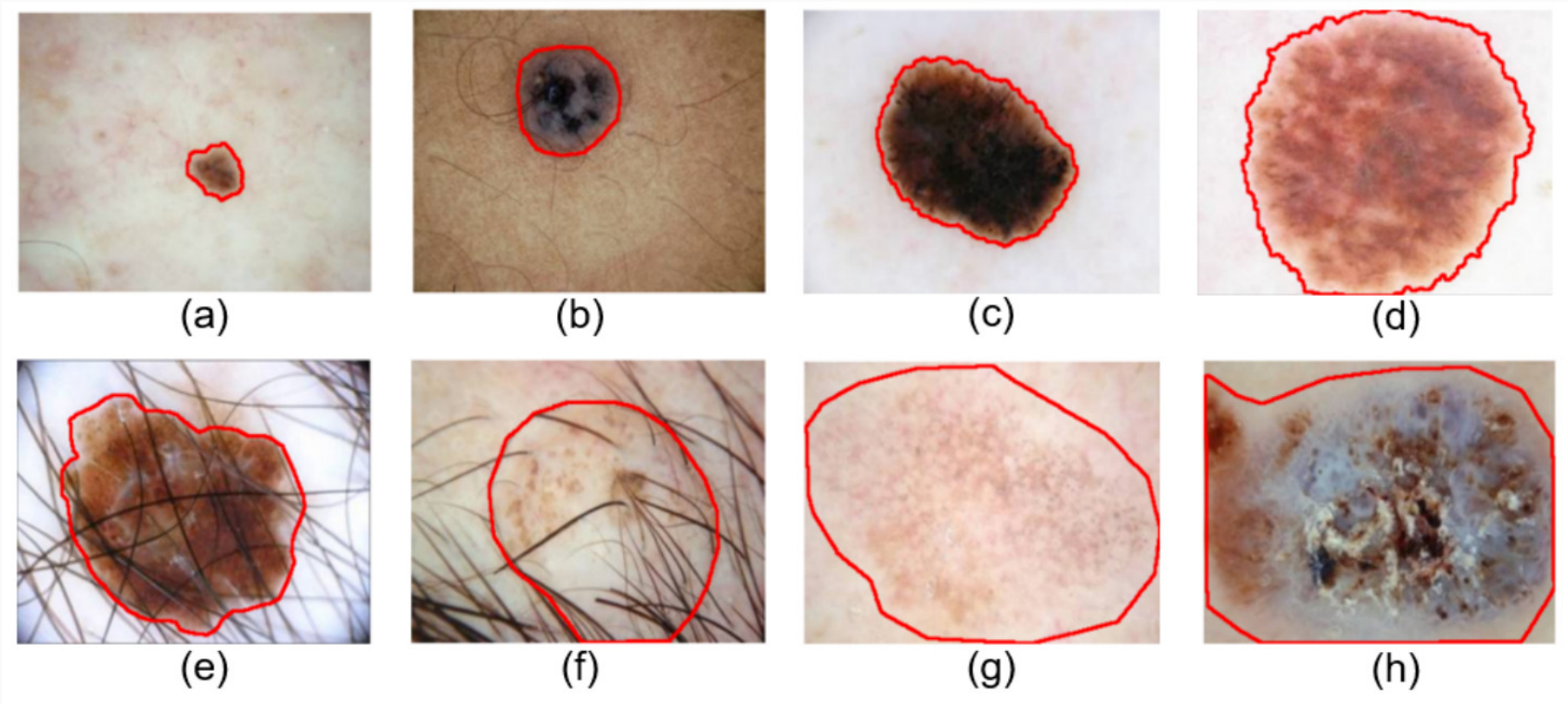
Сегментация медицинских изображений

- TransUNet сохраняет больше деталей:



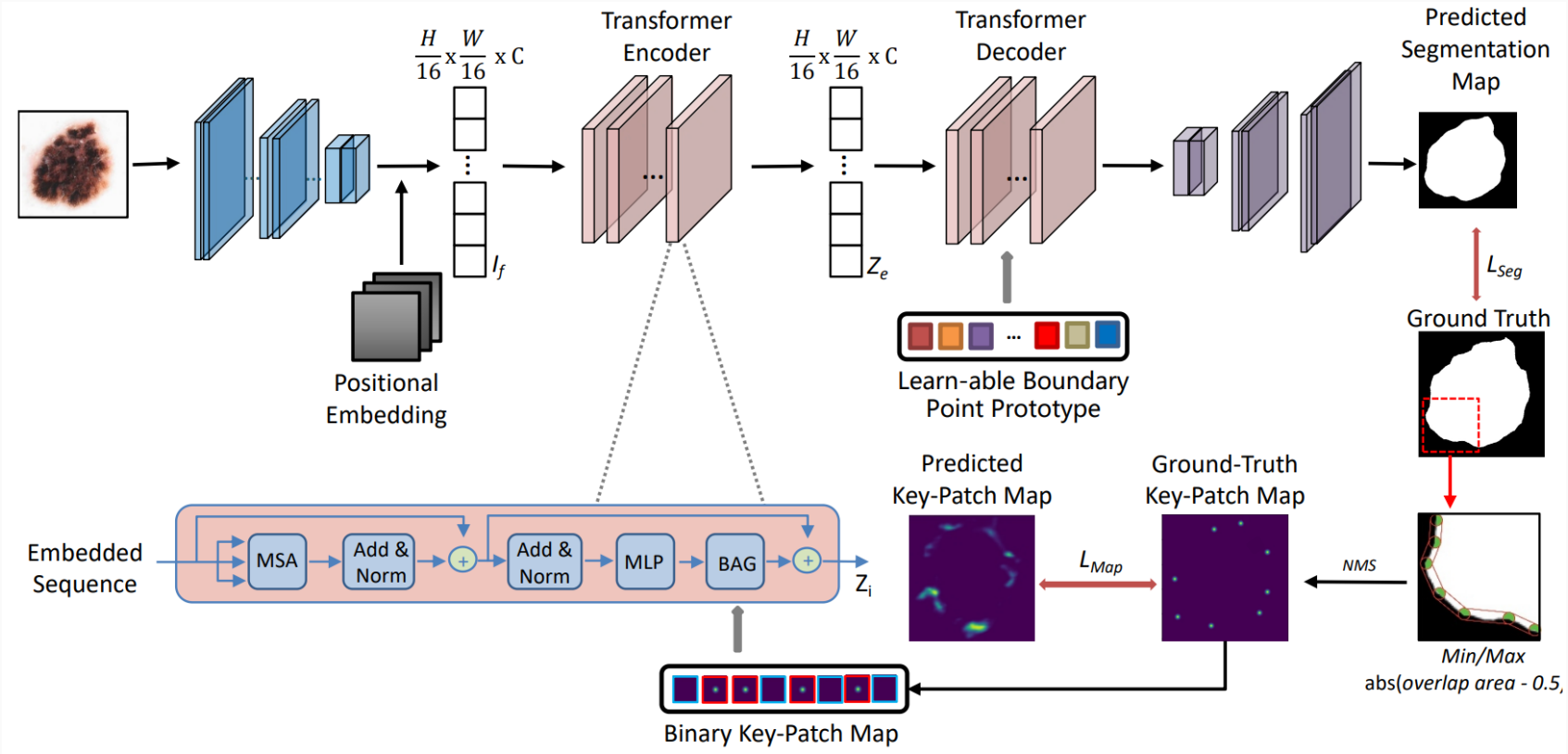
Сегментация медицинских изображений

- Boundary-aware Transformers (BAT; Wang et al., 2021) для сегментации поражений кожи, типичная задача визуализации, где нужно добавить какие-то дополнительные априорные распределения



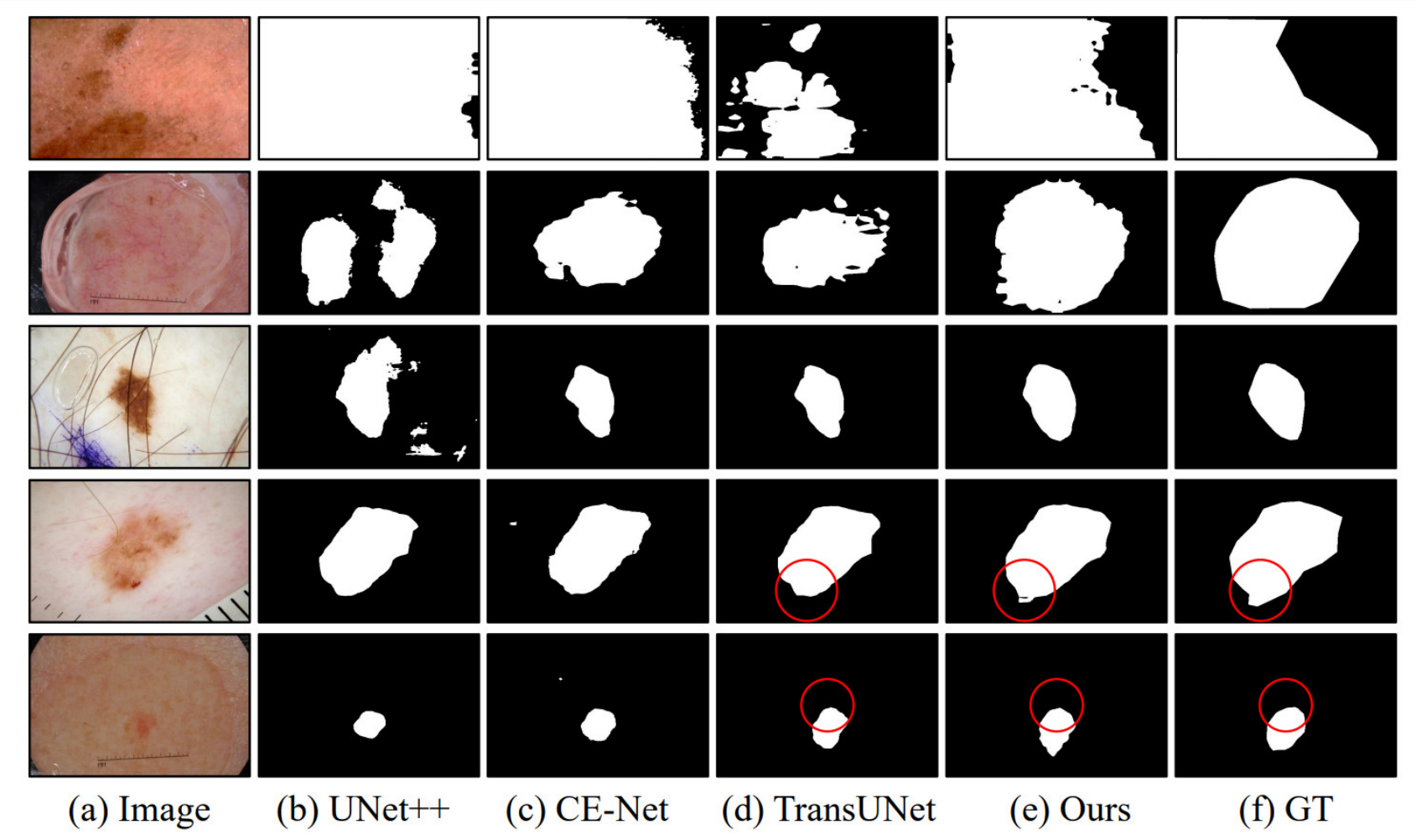
Сегментация медицинских изображений

- BAT похож на ViT, но добавляет гейт граничного внимания (BAG), который принимает признаки на вход и выдает бинарную карту внимания на уровне патчей, где 1 означает, что патч лежит на неоднозначной границе:



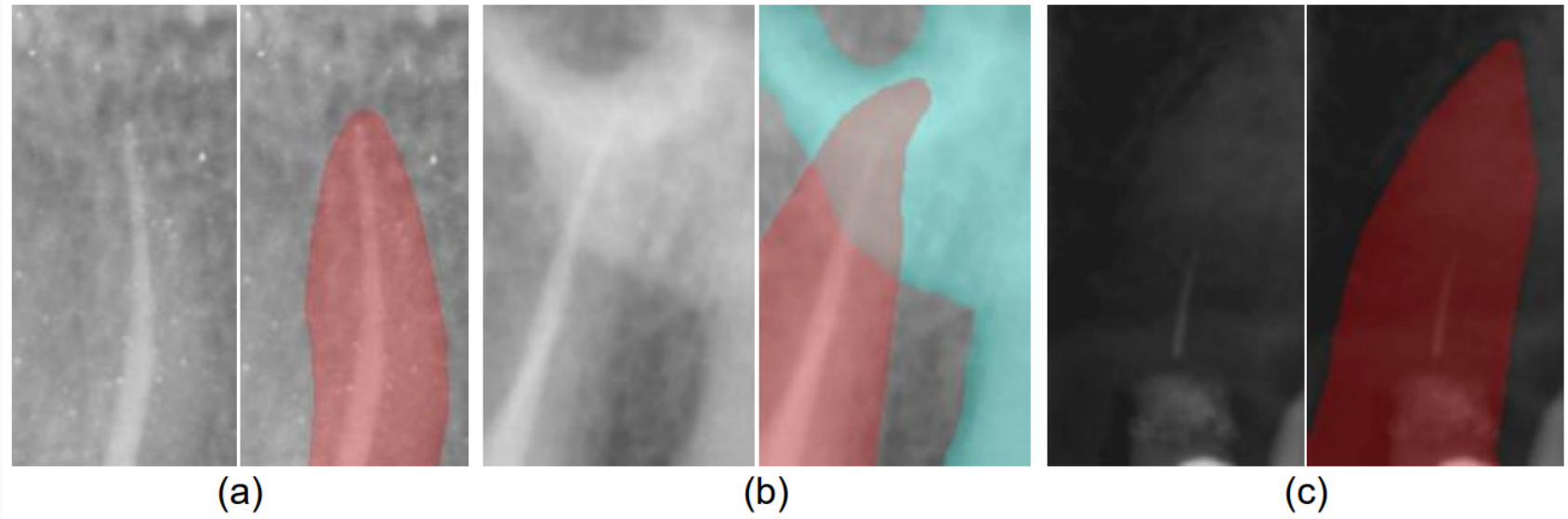
Сегментация медицинских изображений

- BAT дает результаты лучше, чем, например, TransUNet:



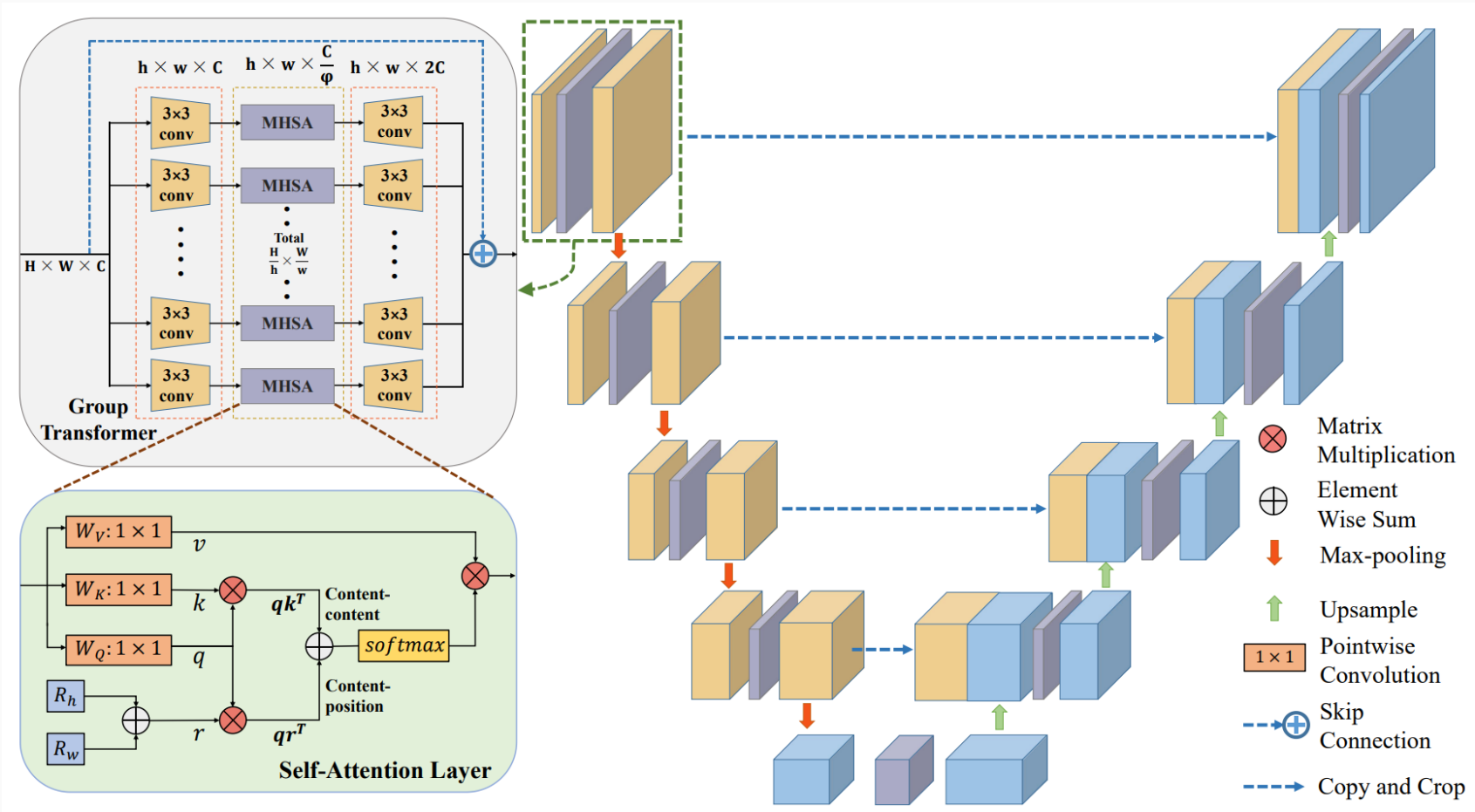
Сегментация медицинских изображений

- Group Transformer Network (GT U-Net; Li et al., 2021): сегментация рентгеновских снимков зубов для эндодонтического лечения
- Рентгеновские снимки часто плохого качества (недодержанные или передержанные), границы размыты, есть наложения...



Сегментация медицинских изображений

- GT U-Net сохраняет структуру U-Net, но добавляет Group Transformer:



Сегментация медицинских изображений

- Интересная часть – *shape-sensitive Fourier descriptor*: если (x_m, y_m) — координата на границе корня зуба с N пикселями, мы можем сформировать форму границы как комплексное число $z(m) = x_m + jy_m$ и определить дескриптор Фурье как

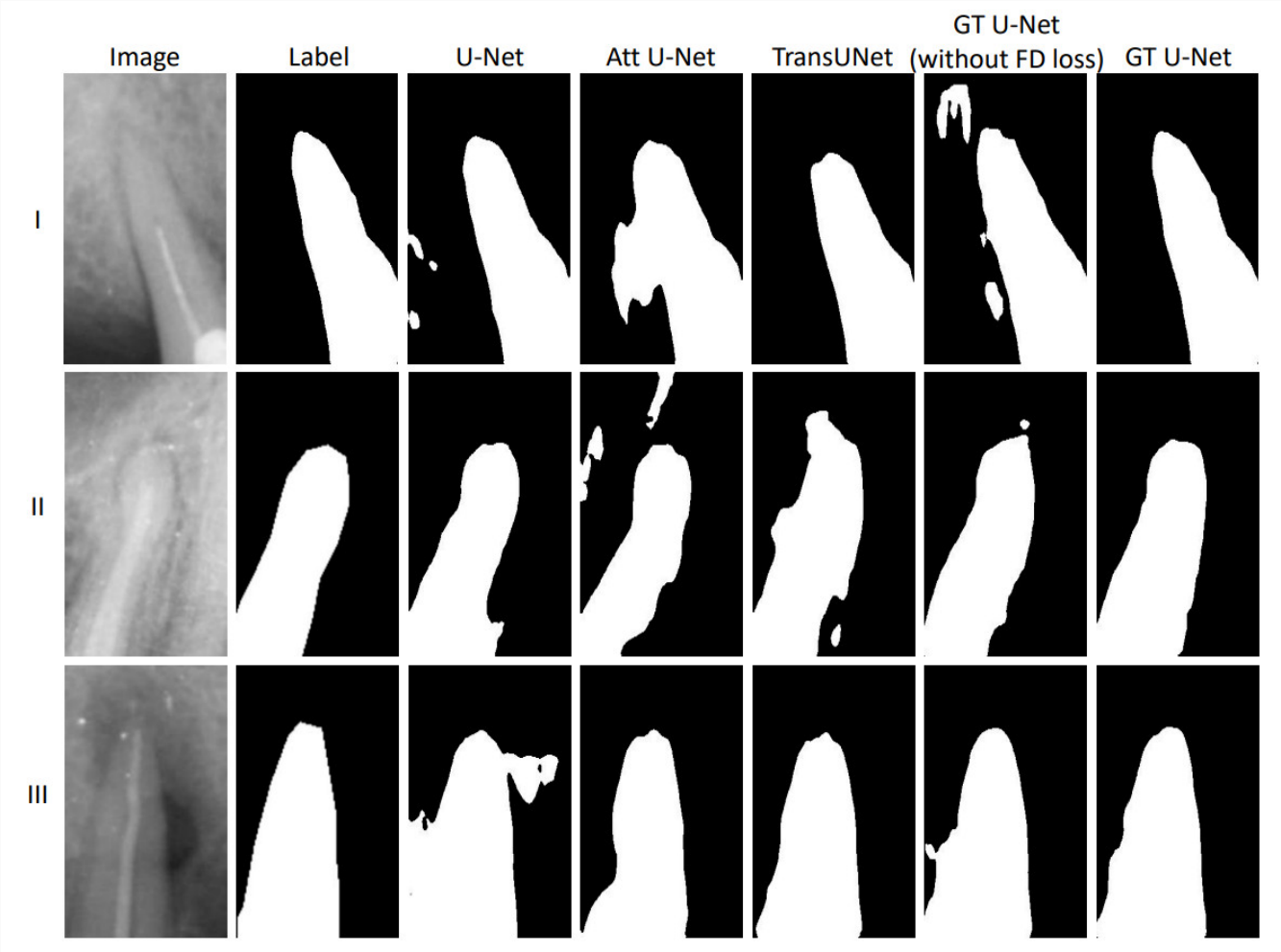
$$Z(k) = \mathrm{DFT}(z(m)) = \frac{1}{N} \sum_{m=0}^{N-1} z(m) e^{-j\frac{2\pi mk}{N}}.$$

- Это количественное представление замкнутых контуров, независимое от их начальной точки, масштаба, местоположения и вращения
- Поэтому в дополнение к бинарной кросс-энтропии мы можем вычислить разность между дескрипторами Фурье $\Delta Z(k) = \left| Z(k) - \hat{Z}(k) \right|$ и определить функцию потерь FD как

$$L_{\mathrm{FD}} = \mathrm{BCE}(\mathbf{x}, \hat{x}) \cdot \frac{1}{1 + e^{-\beta \cdot \Delta Z(k)}}.$$

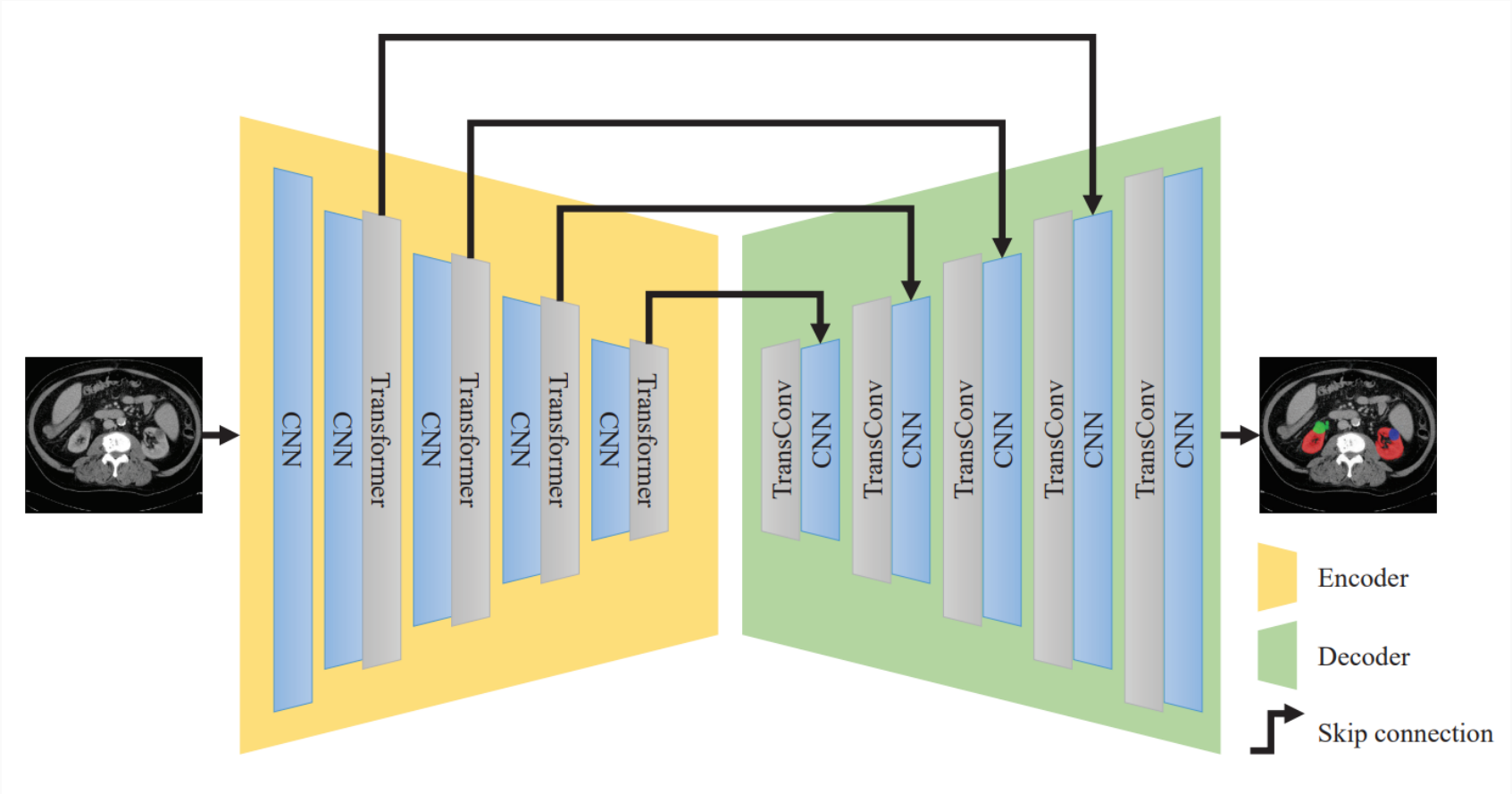
Сегментация медицинских изображений

- В результате GT U-Net улучшает результаты сегментации, особенно с функцией потерь FD:



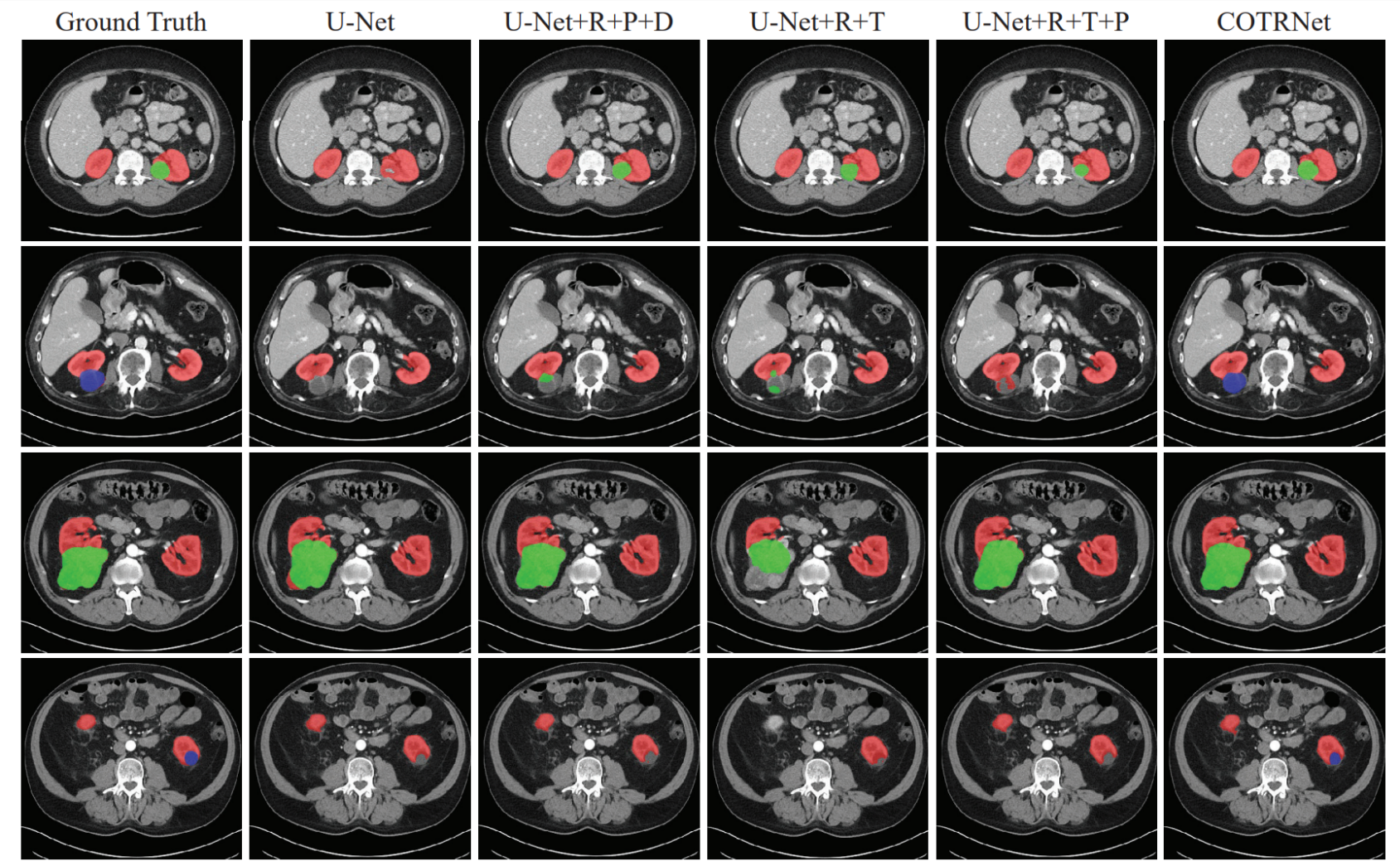
Сегментация медицинских изображений

- Сегментация опухолей почек с COTR-Net (Shen et al., 2021): добавим трансформеры внутрь U-Net



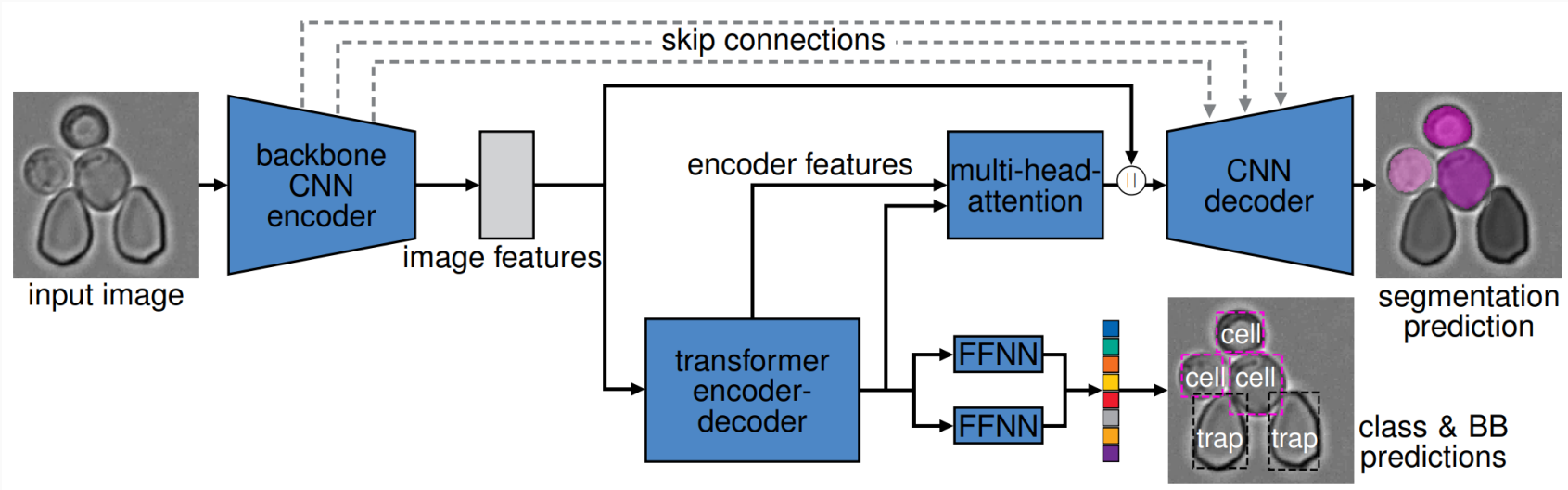
Сегментация медицинских изображений

- COTR-Net улучшает сегментацию:

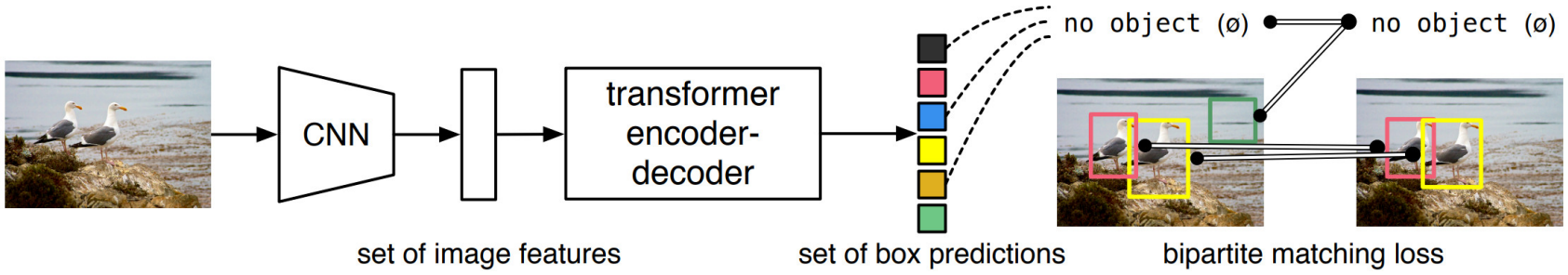


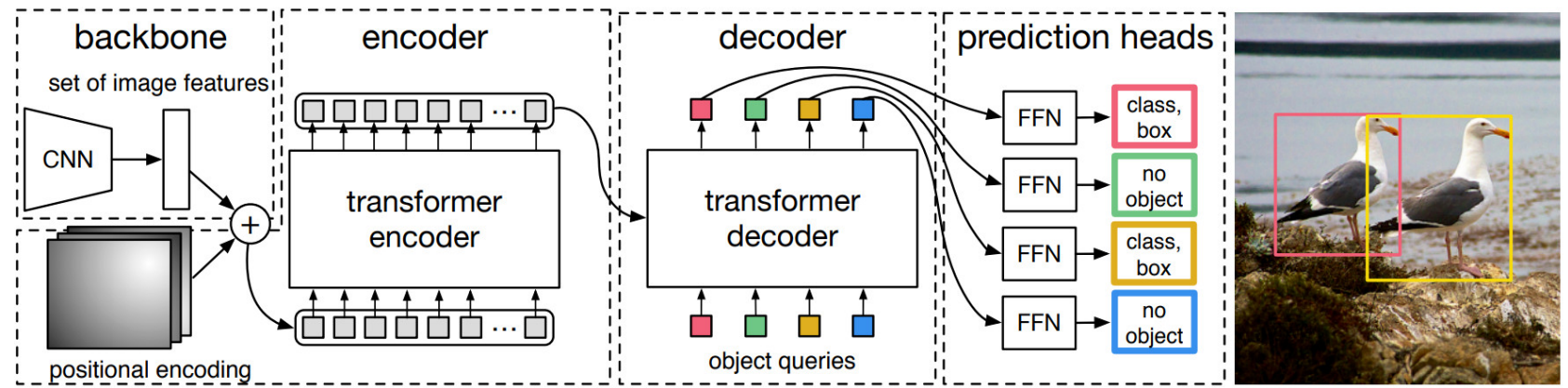
Сегментация медицинских изображений

- Сегментация клеток с Cell-DETR (Prangemeier et al., 2020) на основе Detection Transformer (DETR):



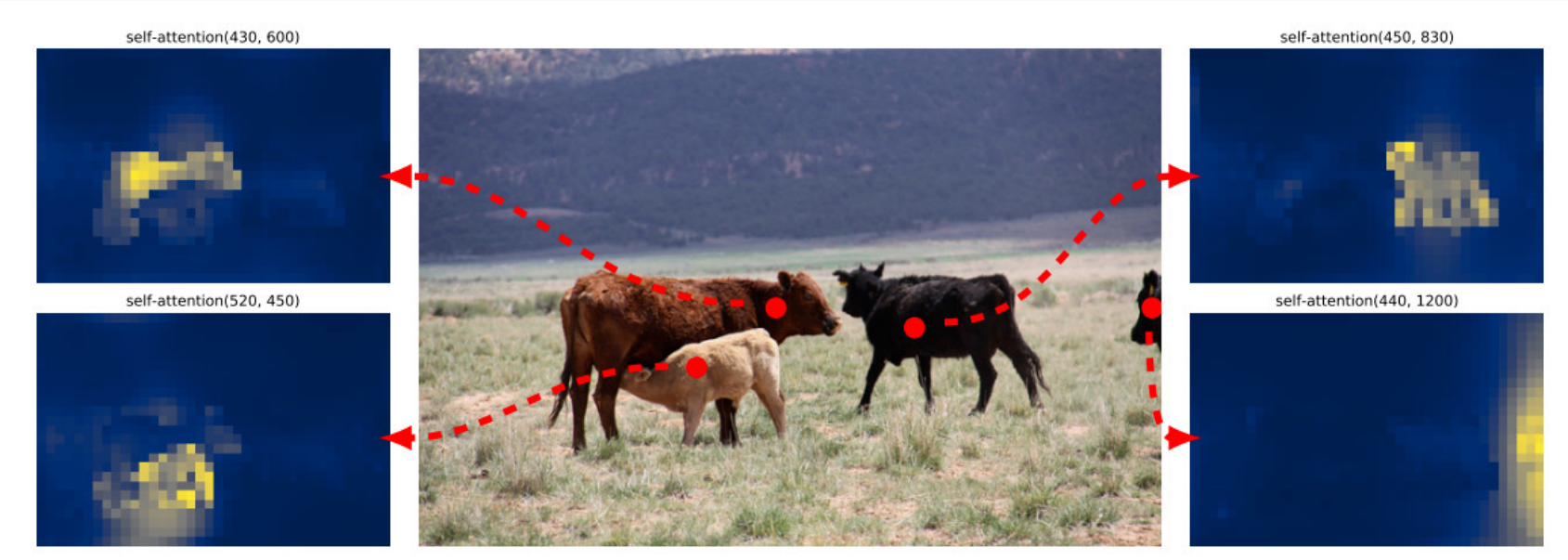
- Сам DETR (Carion et al., 2020) мы уже обсуждали:





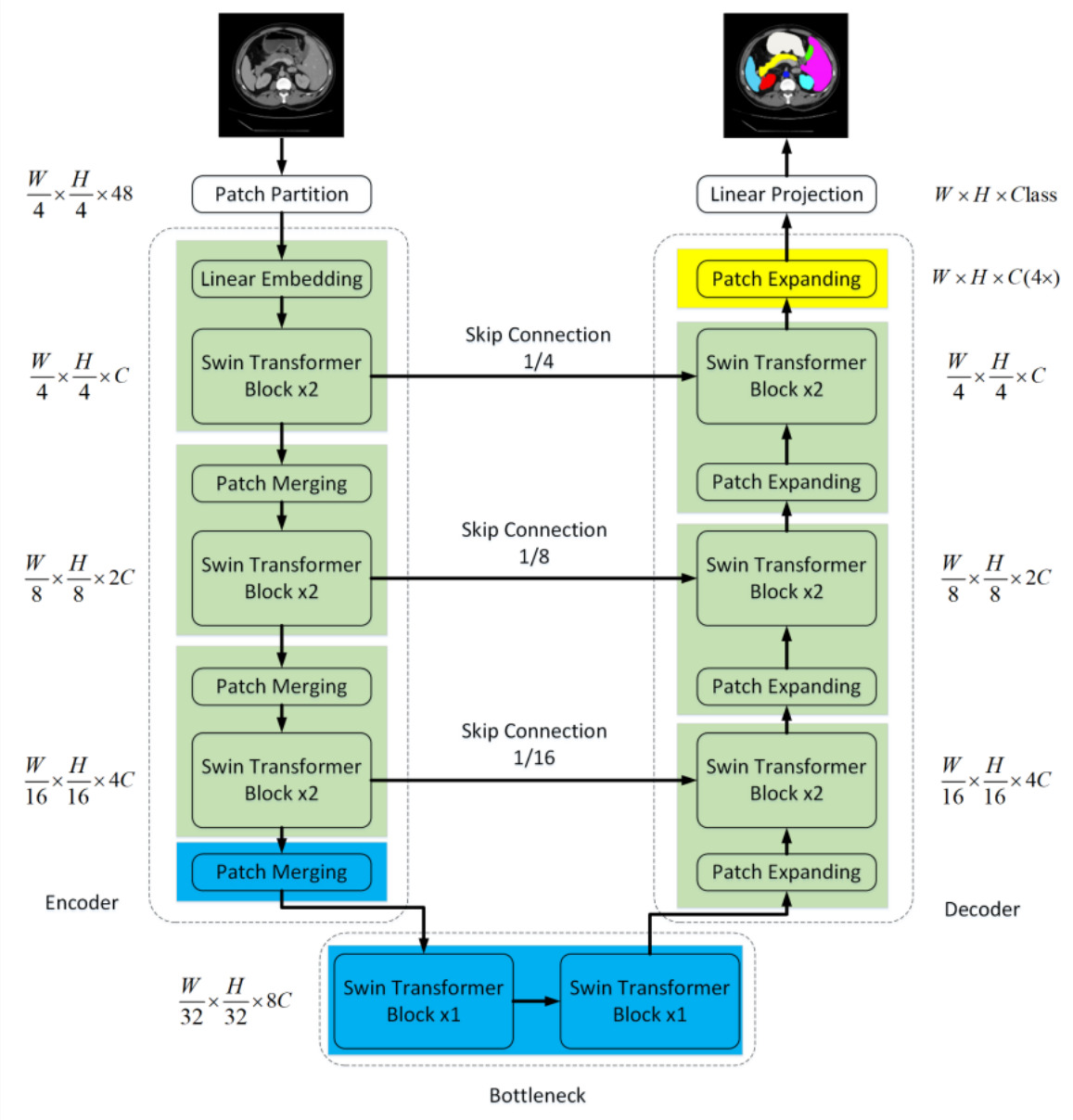
Сегментация медицинских изображений

- Карты внимания DETR хорошо разделяют экземпляры:



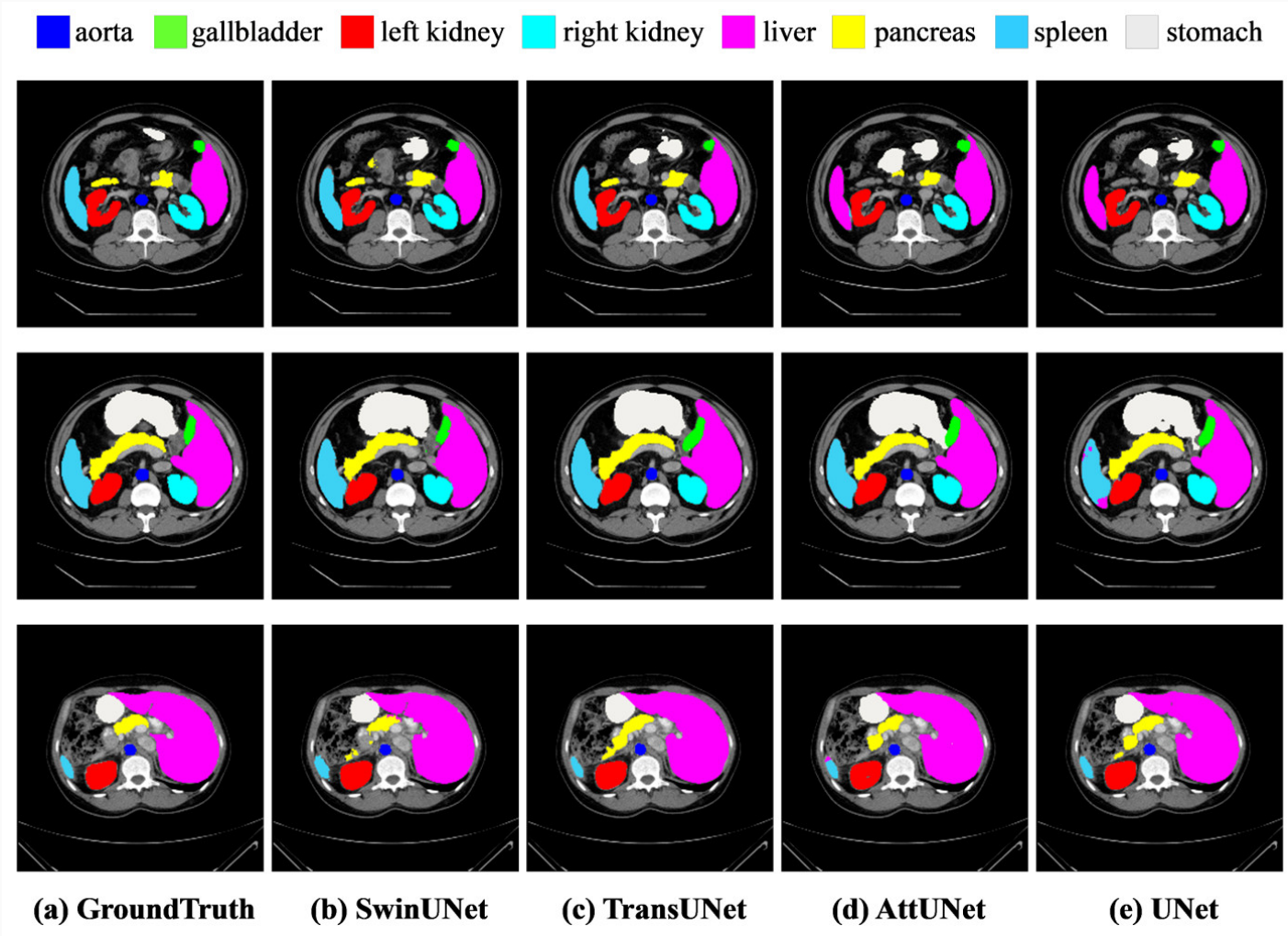
Сегментация медицинских изображений

- Swin-UNet (Cao et al., 2021):



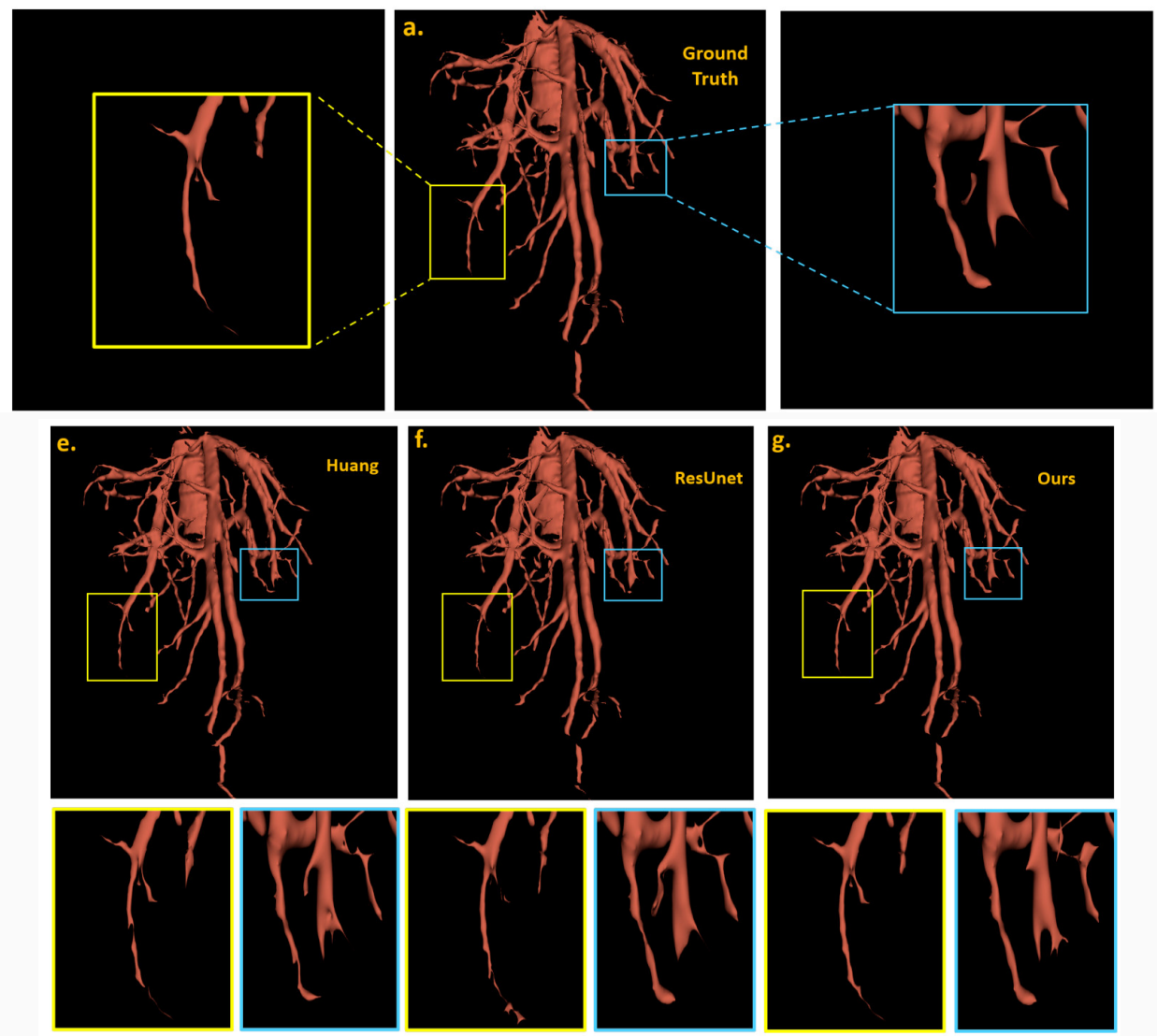
Сегментация медицинских изображений

- Дополнительно улучшает сегментацию компьютерных томограмм:



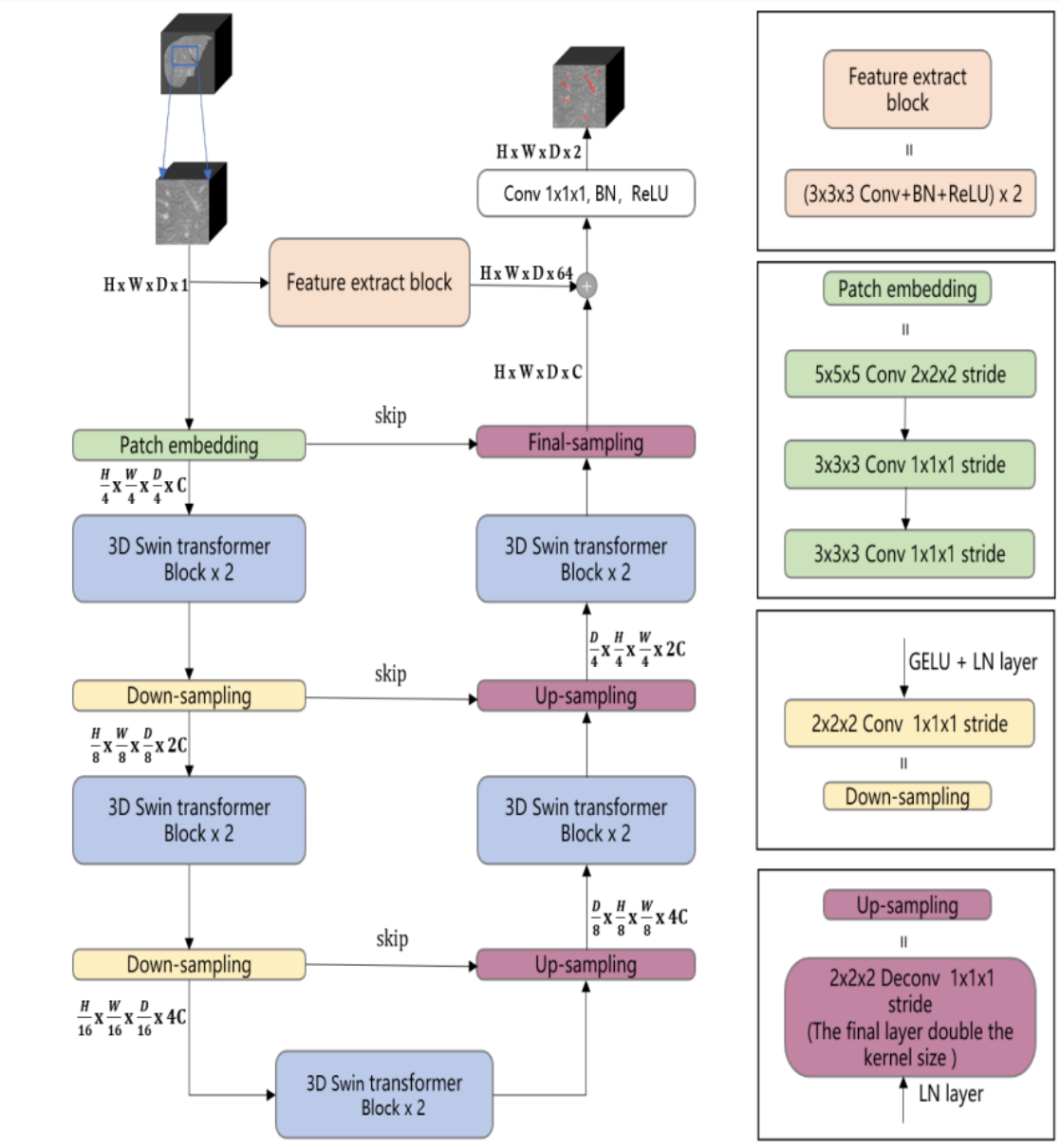
Сегментация медицинских изображений

- В 3D: сегментация печеночных сосудов на компьютерных томограммах (Wu et al., 2021)



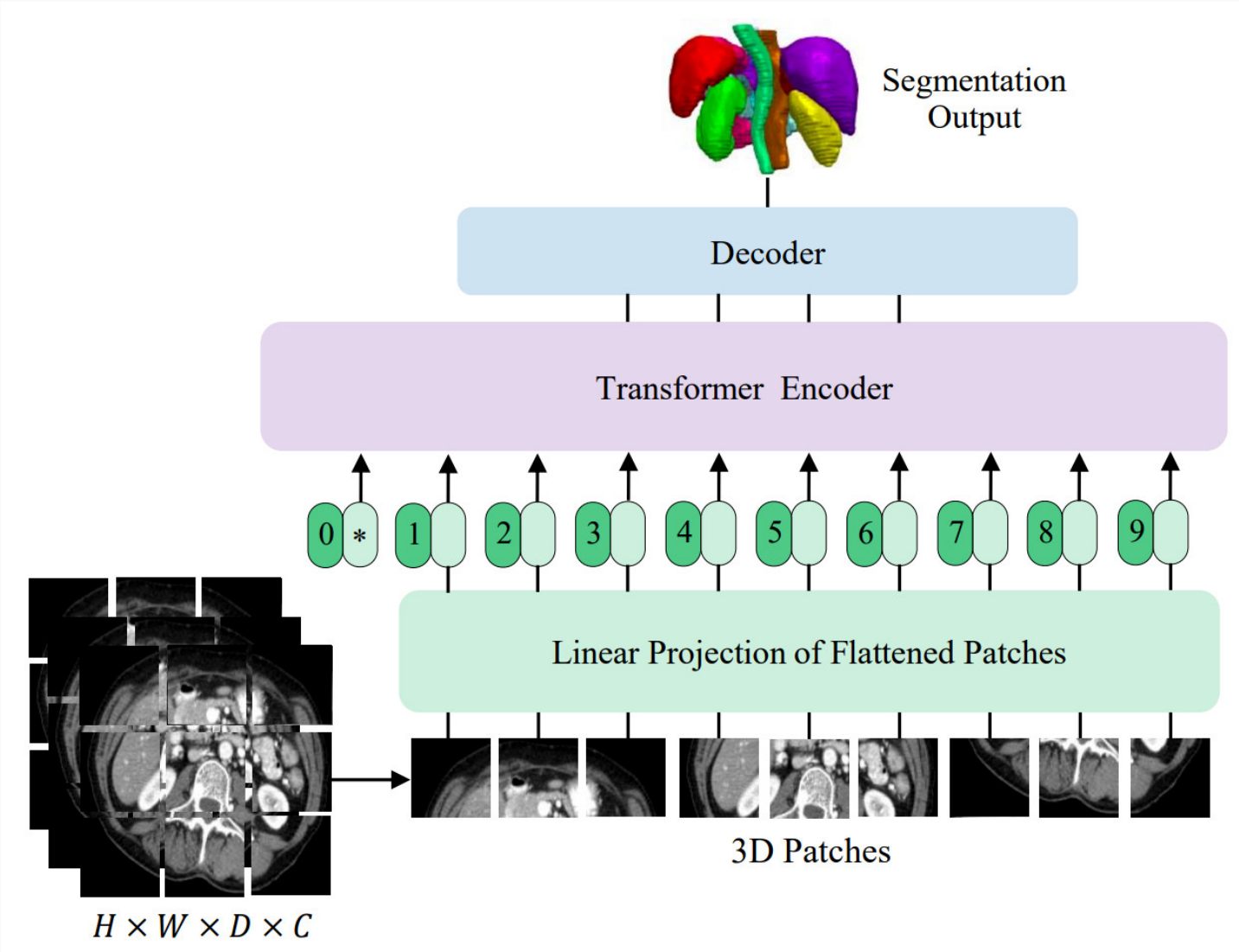
Сегментация медицинских изображений

- Архитектура основана на 3D Swin Transformer:



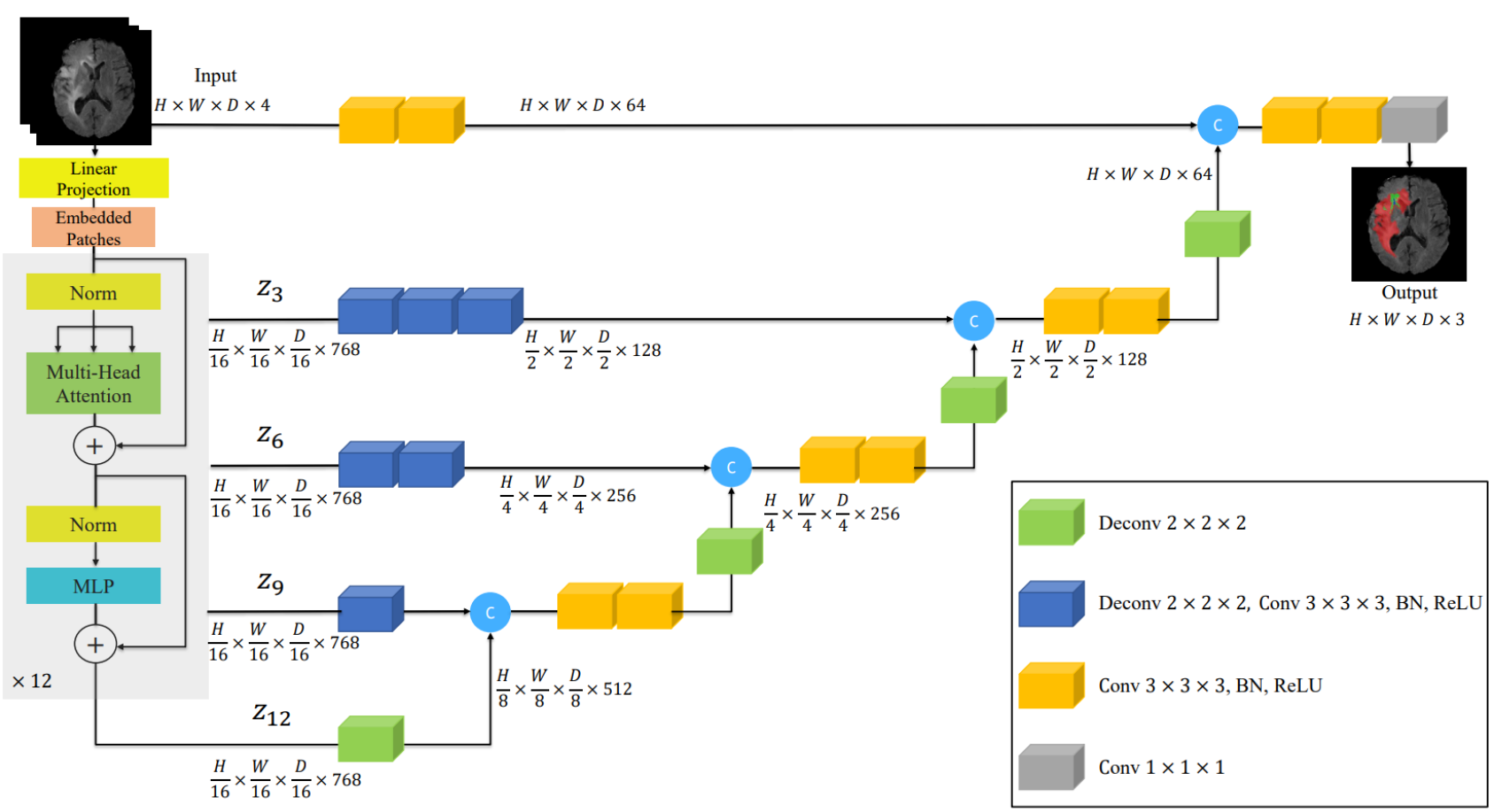
Сегментация медицинских изображений

- UNETR (Hatamizadeh et al., 2021): еще одна структура ViT+UNet для 3D сегментации



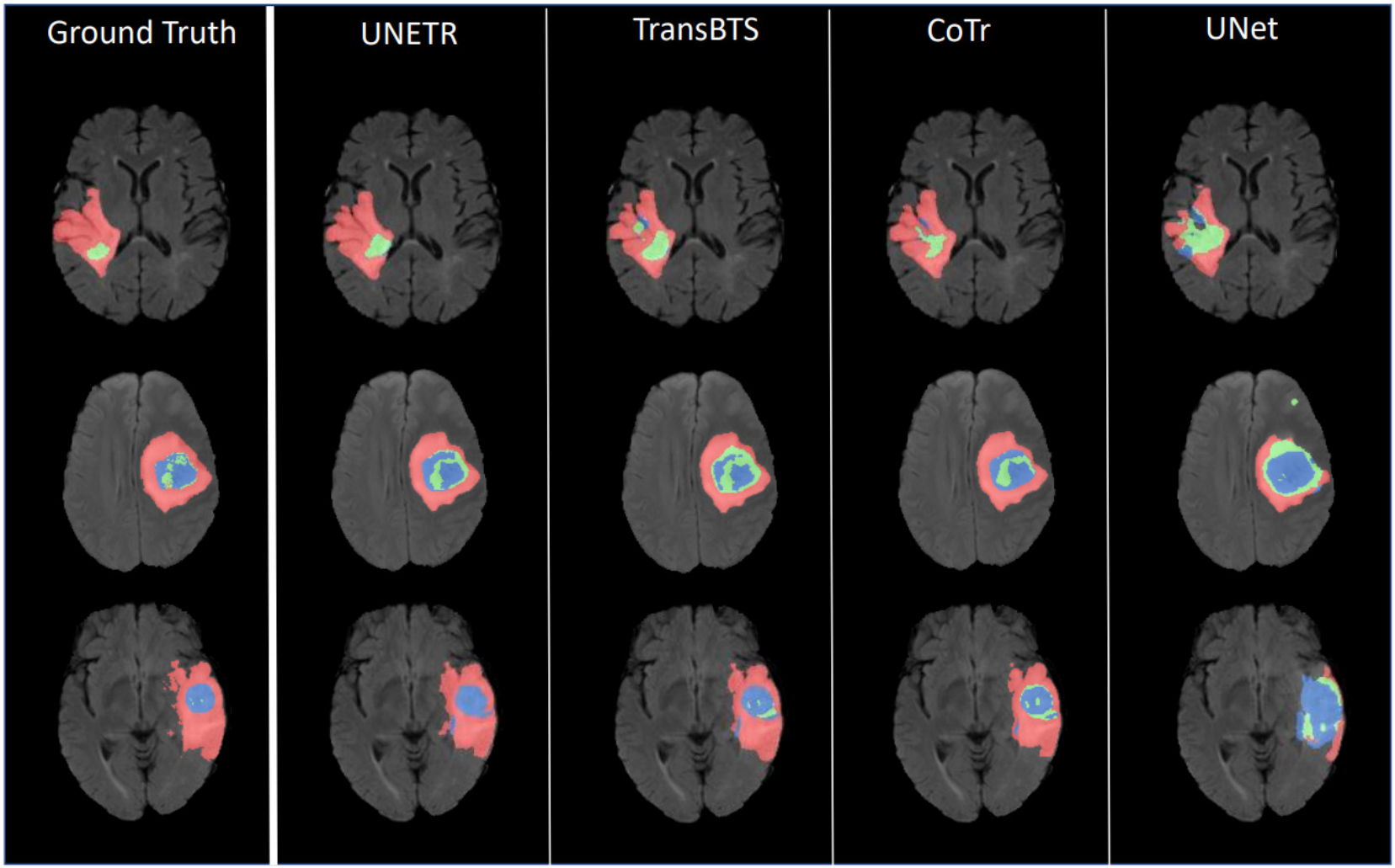
Сегментация медицинских изображений

- Представления, произведенные трансформером, объединяются в декодере для обеспечения мелких деталей, в стиле U-Net:



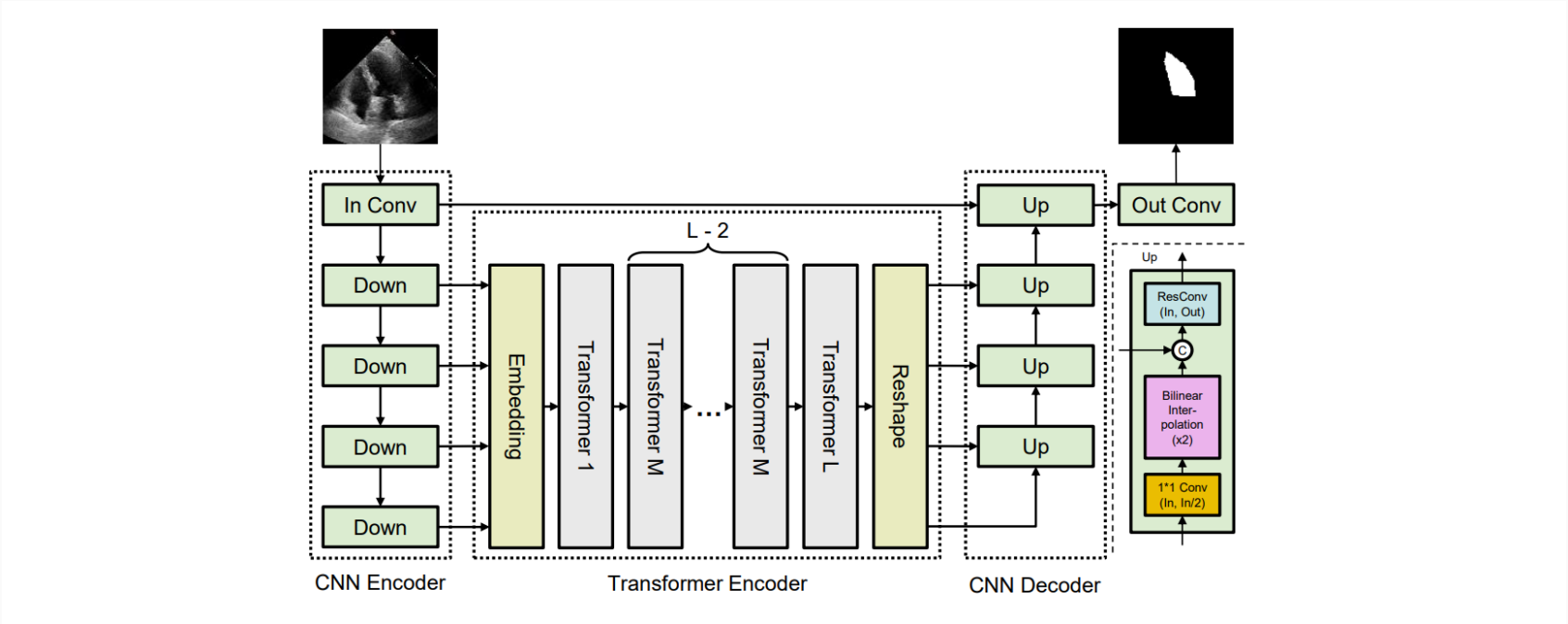
Сегментация медицинских изображений

- Результаты улучшаются:



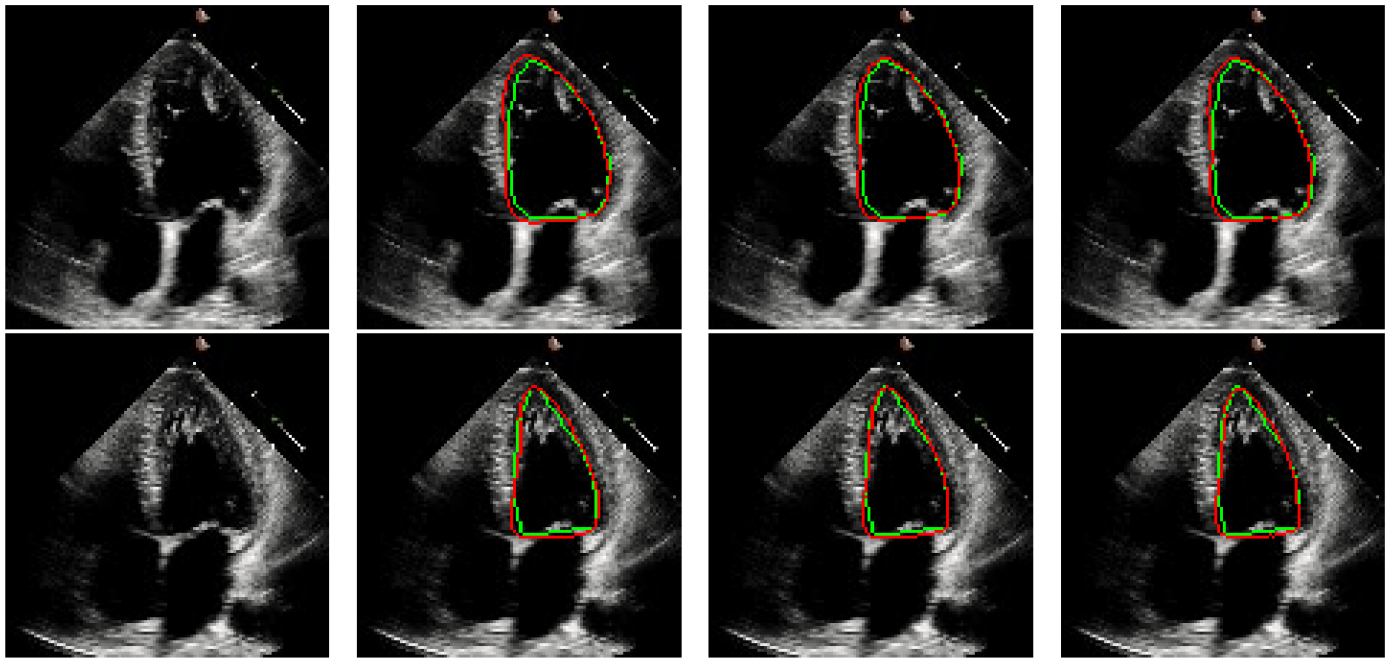
Сегментация медицинских изображений

- Transbridge (Deng et al., 2021): сегментация в электрокардиографии
- Трансформер работает как мост между свёрточным кодировщиком и свёрточным декодером



Сегментация медицинских изображений

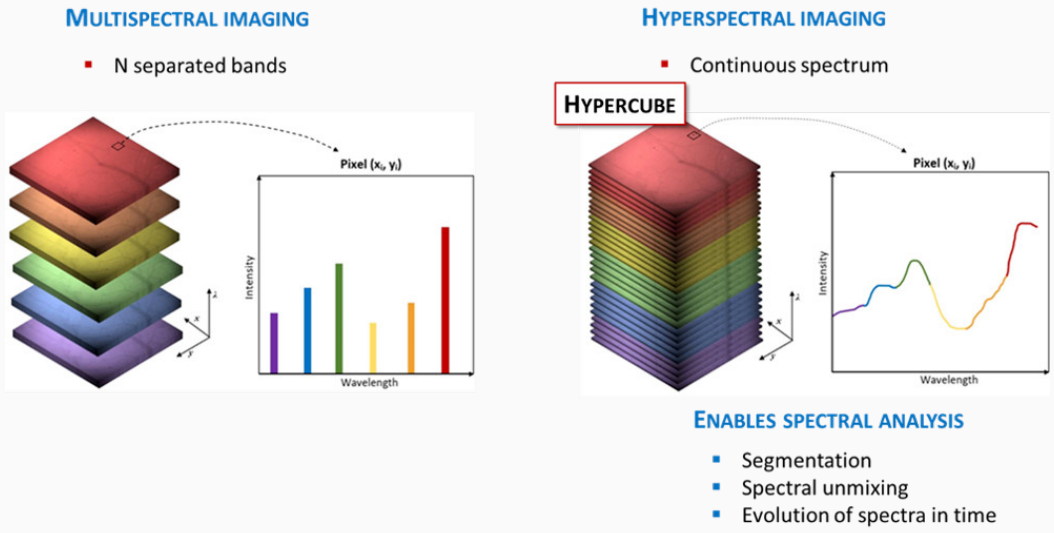
- Улучшает границы сегментации:



(a) Original image (b) CoTr (c) TransBridge-B (d) TransBridge-L

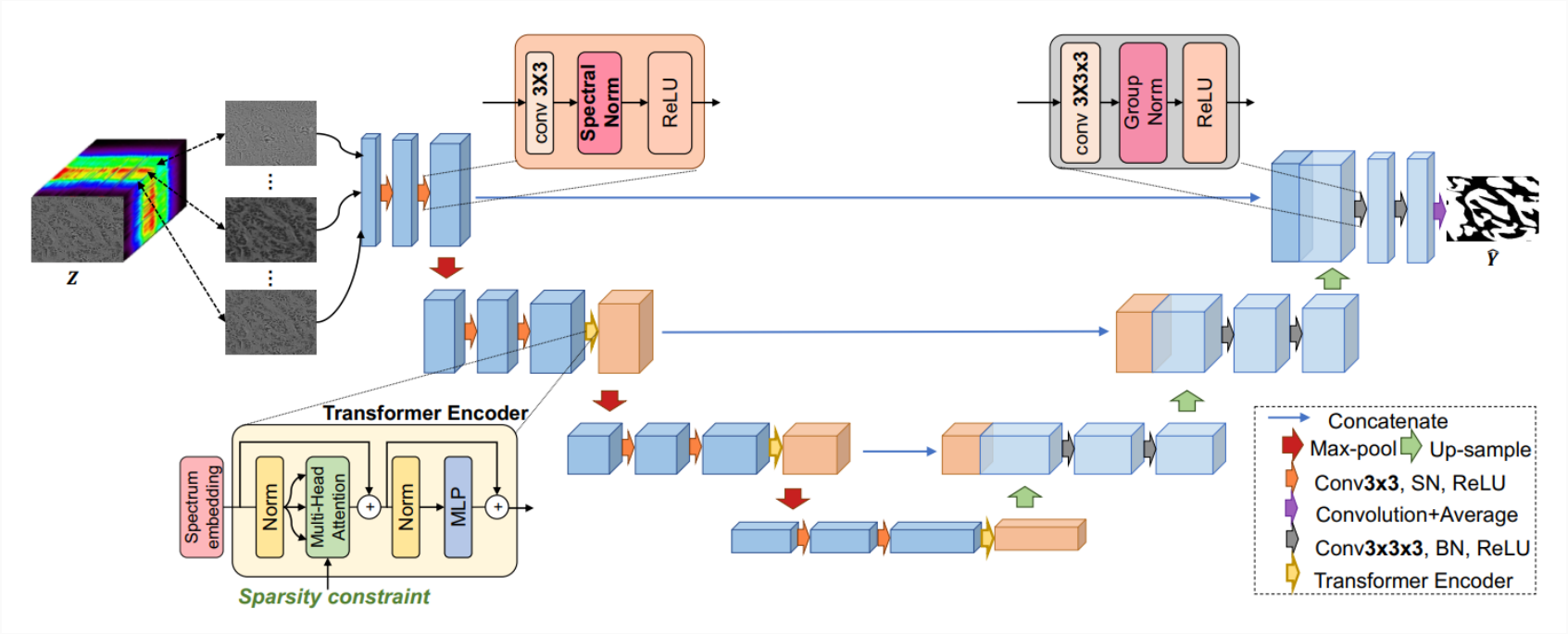
Сегментация медицинских изображений

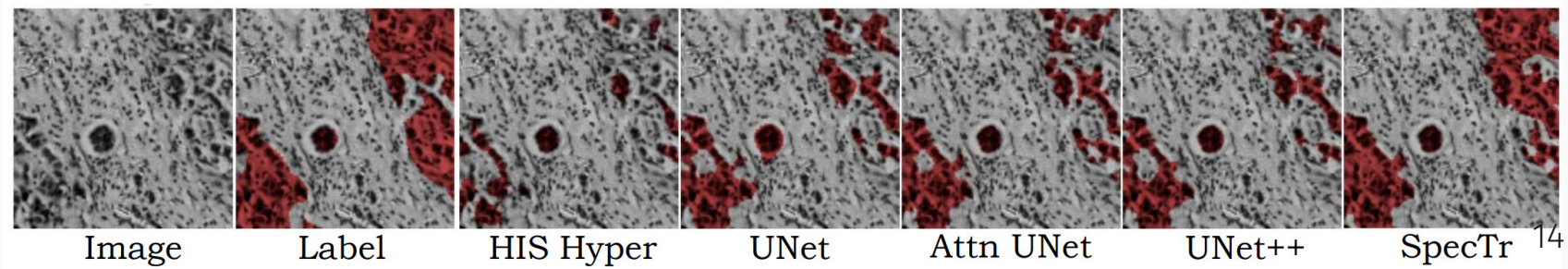
- Более того, можно не ограничиваться видимыми изображениями
- Гиперспектральная съёмка (hyperspectral imaging, HSI) изучает, как широкий спектр света взаимодействует с наблюдаемыми материалами; спектральная информация представлена сотнями узких смежных спектральных полос, гораздо больше каналов, чем просто три основных цвета
- Вход теперь – гиперкуб со множеством спектральных полос:



Сегментация медицинских изображений

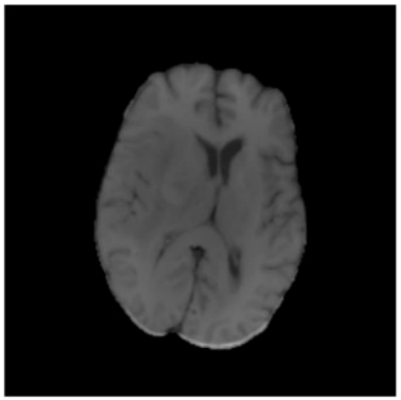
- SpecTr (Yun et al., 2021): U-Net + Transformer для гиперспектральных снимков



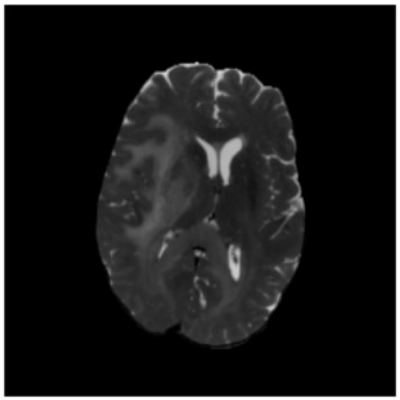


Сегментация медицинских изображений

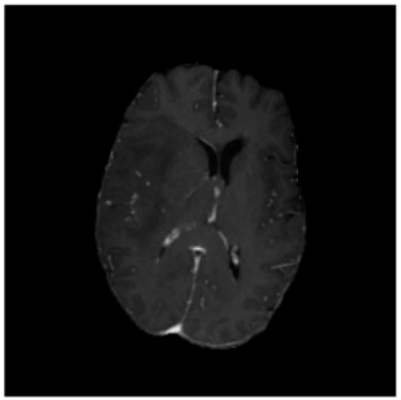
- Другое направление – 3D сегментация
- BiTr-Unet (Jia, Shu, 2021): CNN + Transformer для сегментации опухолей мозга на МРТ



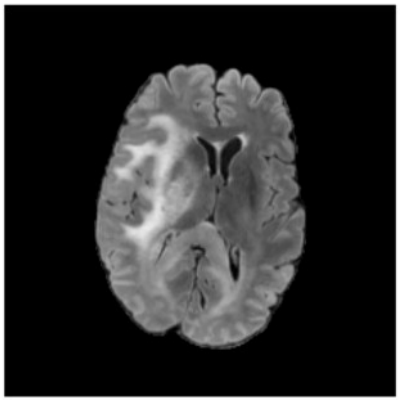
(a)T1



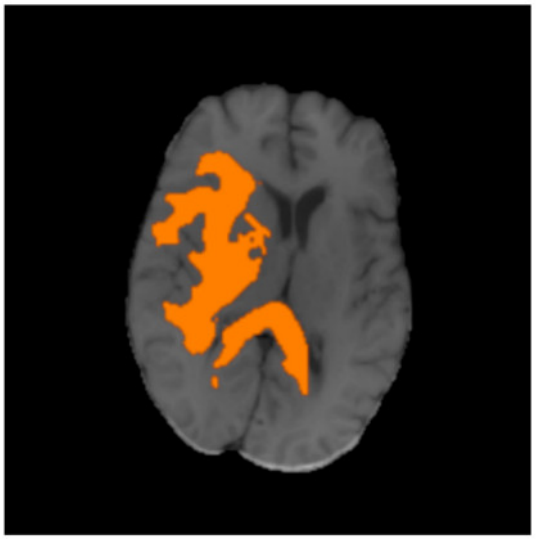
(b)T2



(c)T1c



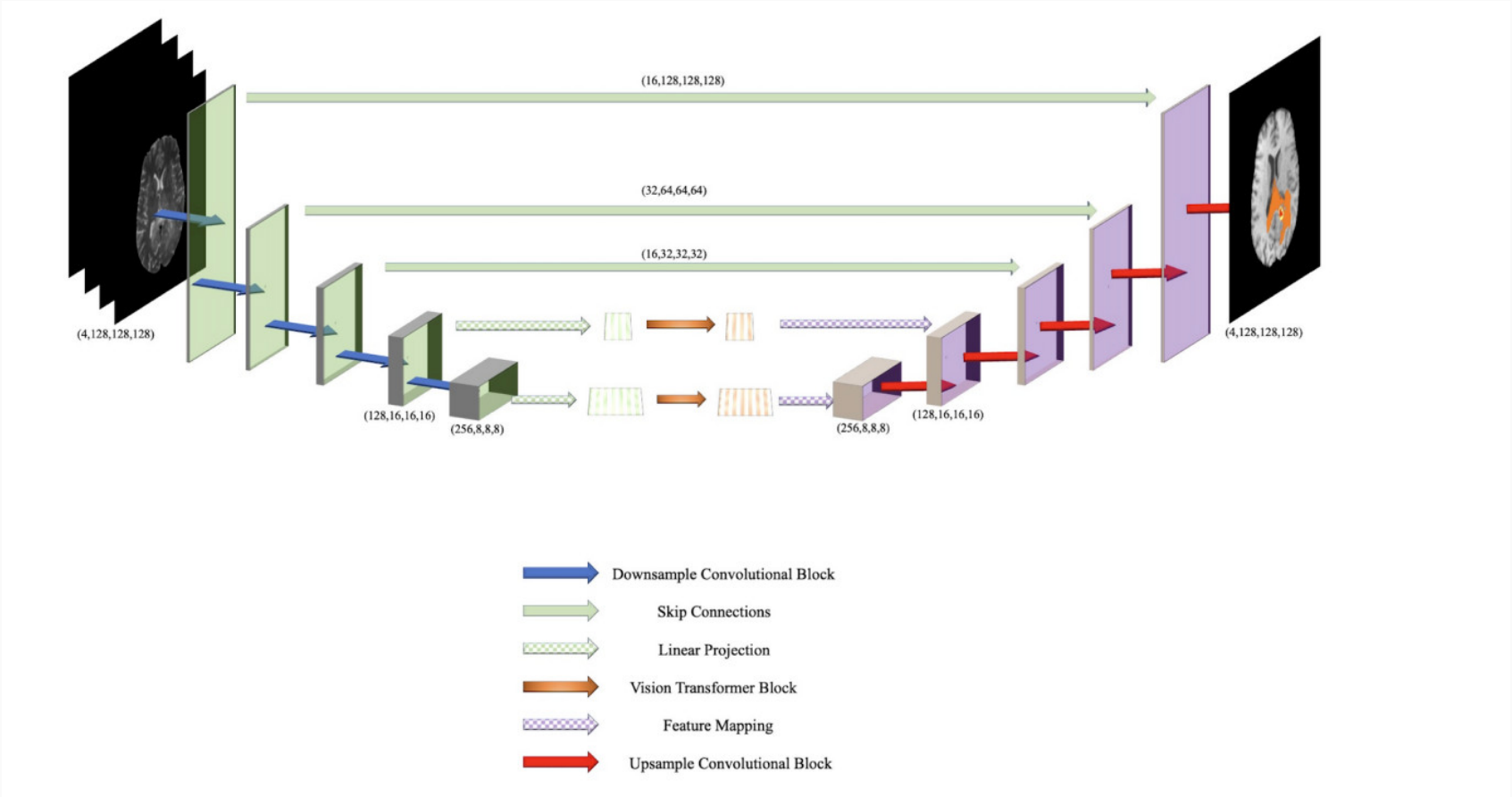
(d)T2-FLAIR



(e)T1 with segmentaton

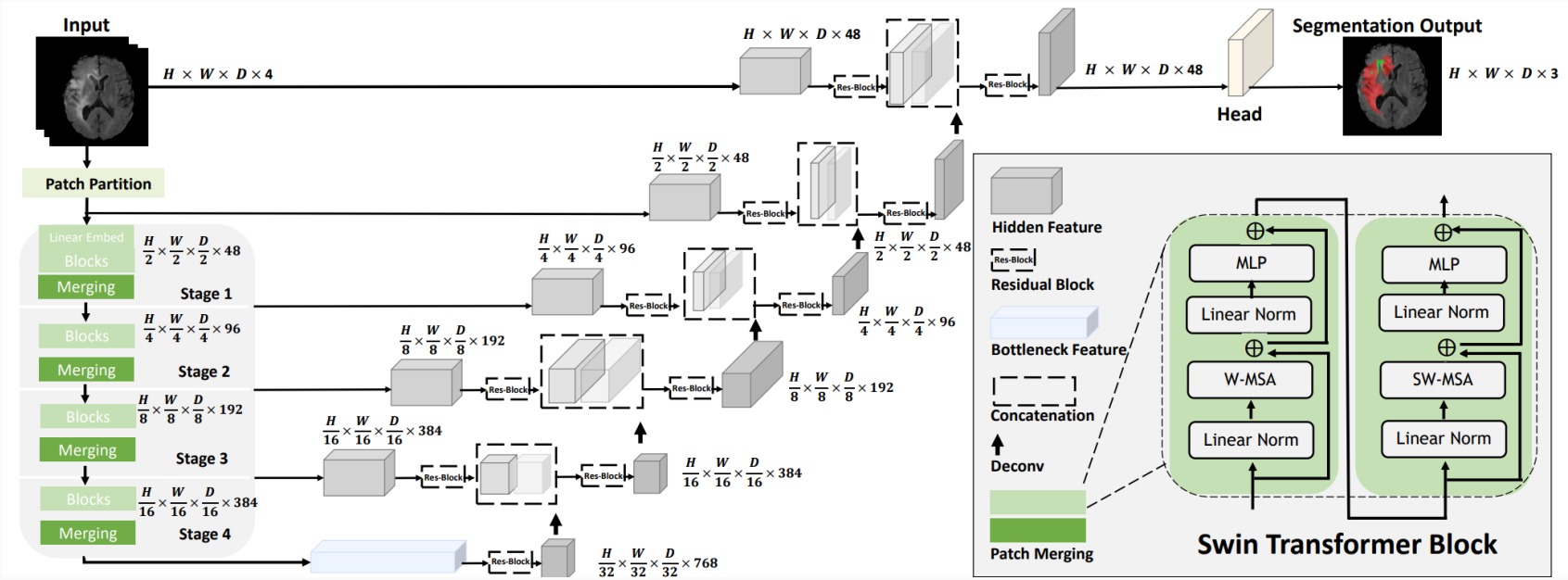
Сегментация медицинских изображений

- Снова U-Net-подобная архитектура с блоками ViT:



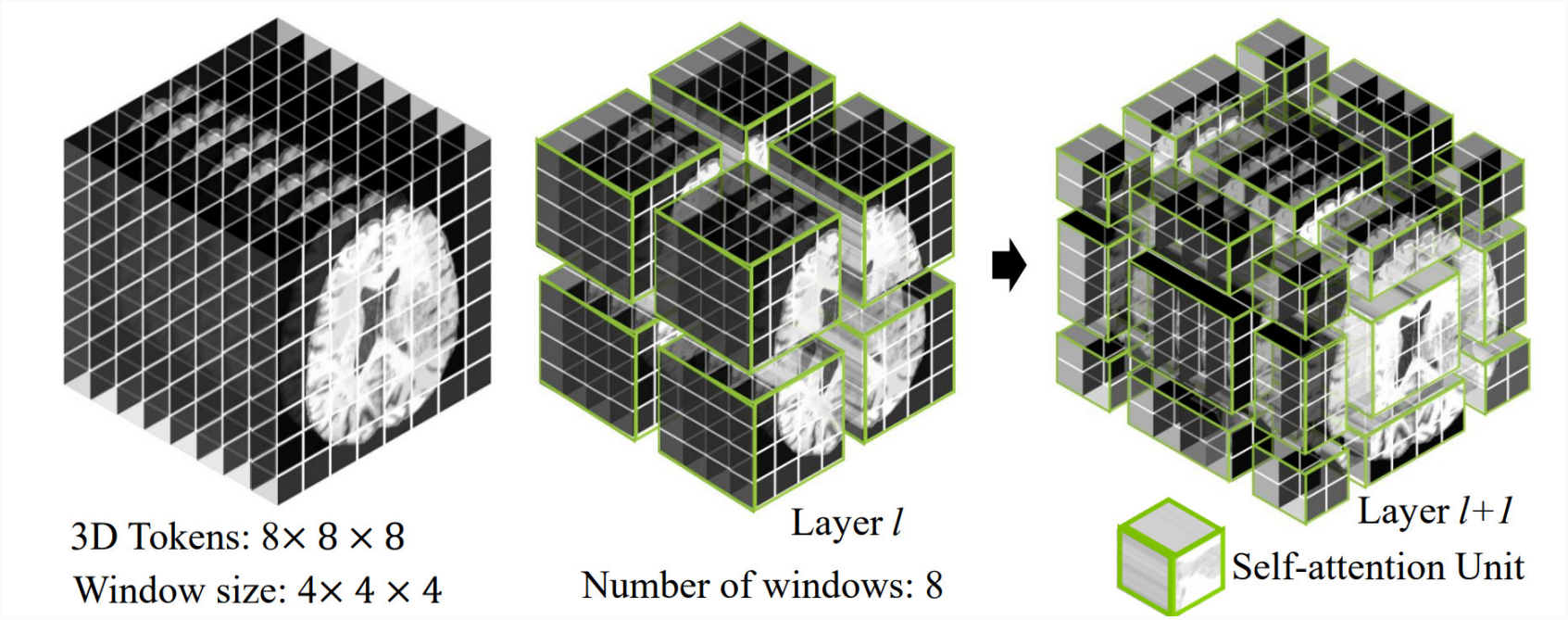
Сегментация медицинских изображений

- Swin UNETR: Swin Transformers для семантической сегментации опухолей мозга на МРТ-изображениях



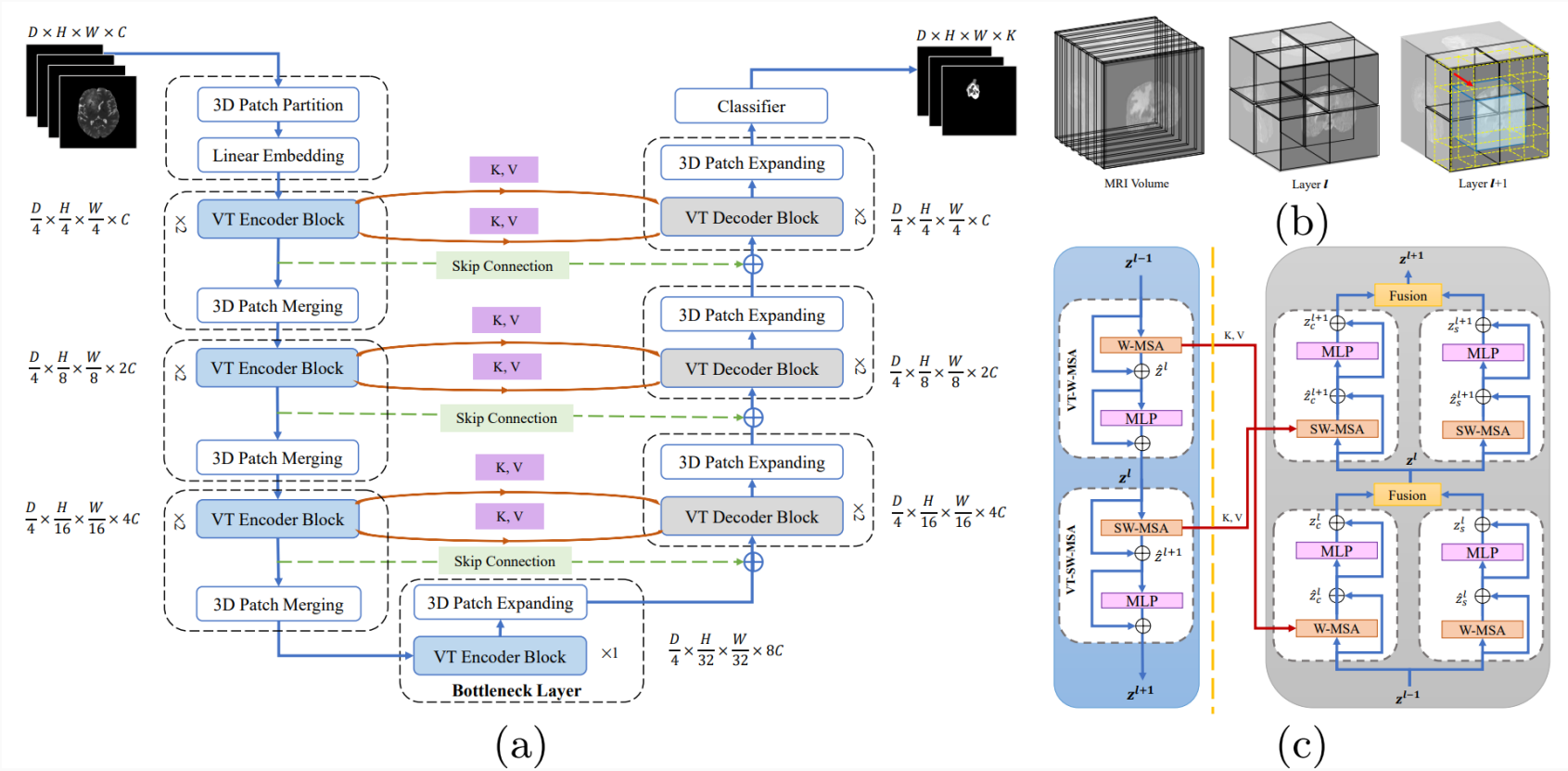
Сегментация медицинских изображений

- Swin UNETR адаптирует метод сдвинутых окон к 3D объёмным окнам:



Сегментация медицинских изображений

- VT-UNET: Transformer для объёмной сегментации (Peiris et al., 2021)



Сегментация медицинских изображений

- VT-UNET: Transformer для объёмной сегментации (Peiris et al., 2021)

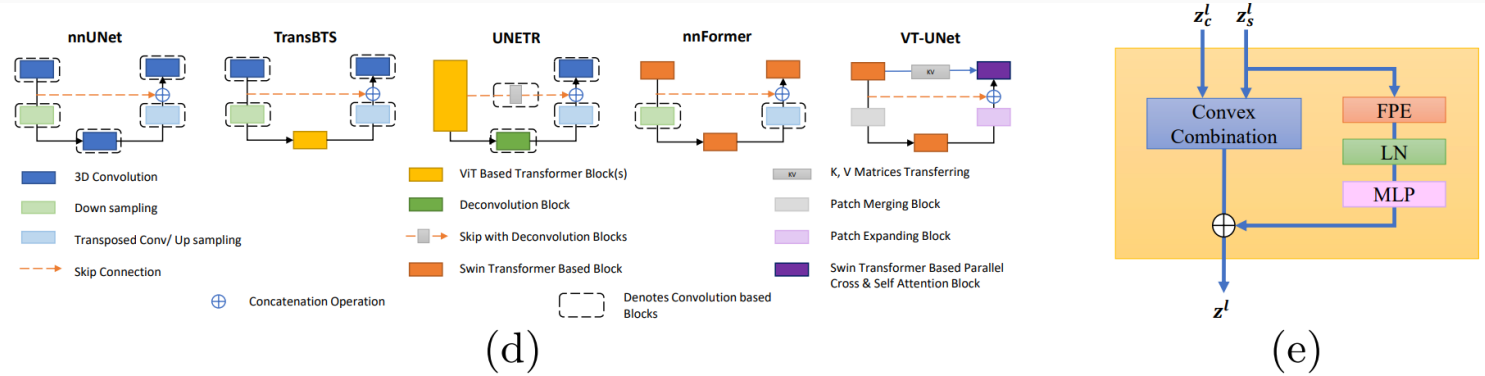


Fig. 2: (a) Illustrates VT-UNet Architecture. Here, k denotes the number of classes. (b) shows visualization of Volumetric Shifted Windows. Consider an MRI volume of size $D \times H \times W$ with $D = H = W = 8$ for the sake of illustration. Further, let the window size for partitioning the volume be $P \times M \times M$ with $P = M = 4$. Here, layer l adopts the regular window partition in the first step of Volumetric Transformer(VT) block which results in $2 \times 2 \times 2 = 8$ windows. Inside layer $l + 1$, volumetric windows are shifted by $(\frac{P}{2}, \frac{M}{2}, \frac{M}{2})$ =(2, 2, 2) tokens. This results in $3 \times 3 \times 3 = 27$ windows. (c) shows VT Encoder-Decoder Structure. (d)Encoder-Decoder structural comparison with other SOTA methods. The proposed VT-UNet architecture has no convolution modules and is purely based on Transformer blocks. (e) Illustrates the structure of the Fusion Module.

Сегментация медицинских изображений

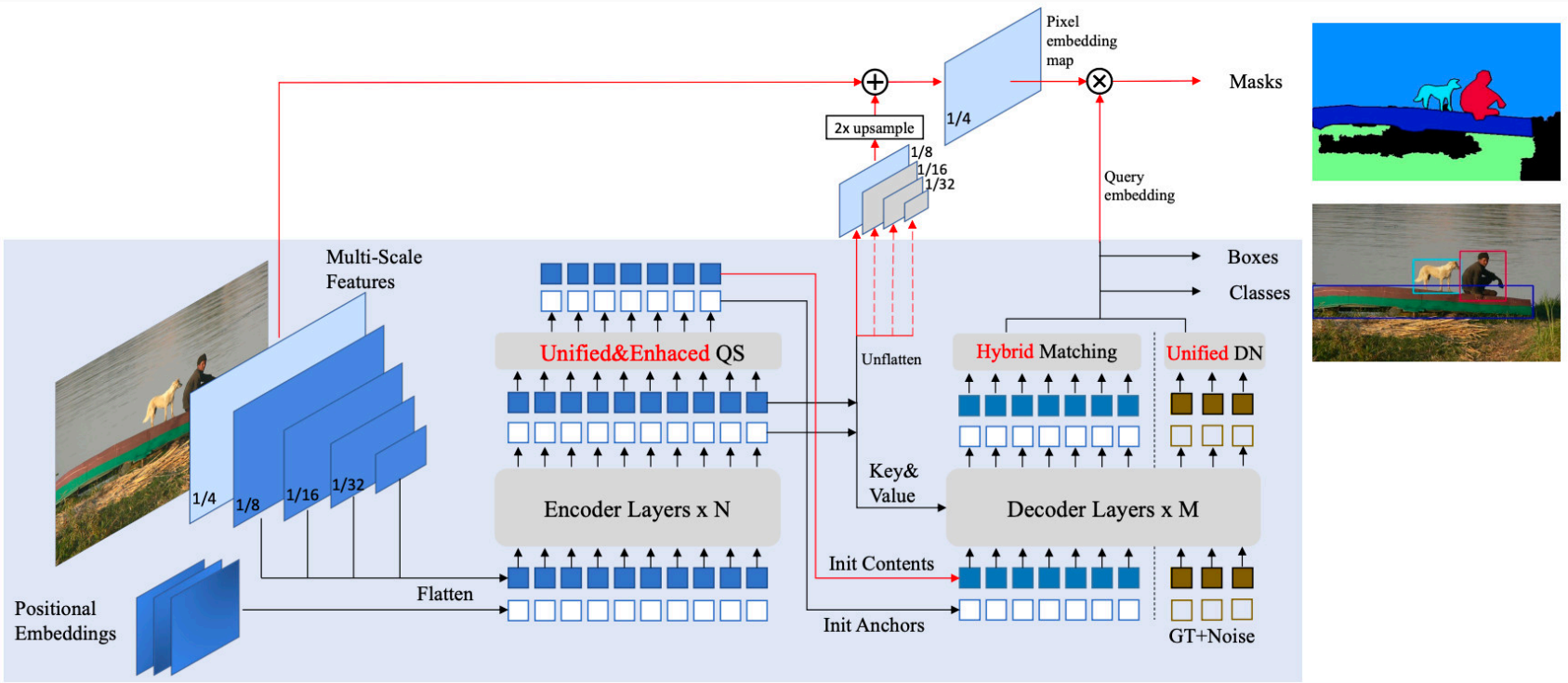
- Есть общие датасеты для сравнения; например, 3D датасет опухолей мозга BraTS 2021 (Baid et al., 2021)

Method	#params	Flops	Dice Score (Avg.)
TransBTS [138]	33 M	333 G	84.99
BiTr-UNet [139]	-	-	86.20
UNETR [35]	102.5 M	193.5 G	84.51
nnFormer [144]	39.7 M	110.7 G	86.56
Swin UNETR [145]	61.98 M	394.84 G	**88.97**
VT-UNET-T [143]	**5.4 M**	**52 G**	86.82
VT-UNET-S [143]	11.8 M	100.8 G	87.00
VT-UNET-B [143]	20.8 M	165 G	88.07



Mask DINO

- Mask DINO (Li et al., 2023): идеи Mask R-CNN можно применить и к детекторам на основе трансформеров



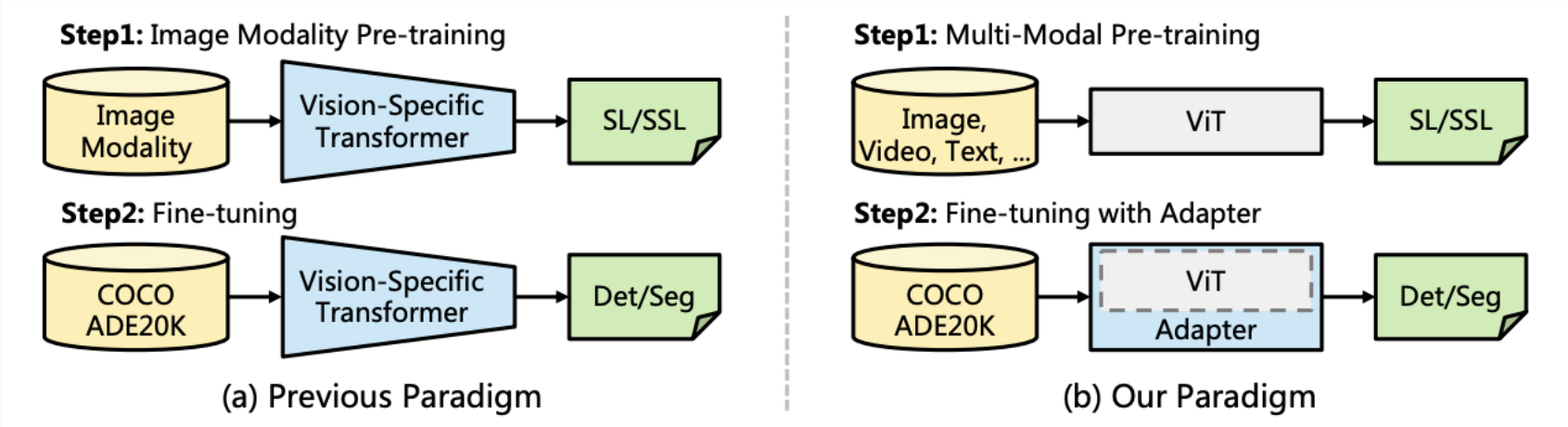
Mask DINO

- Улучшаются результаты на COCO и ADE20K

Method	Params	Backbone	Backbone Pre-training Dataset	Detection Pre-training Dataset	val	
					w/o TTA	w/ TTA
Instance segmentation on COCO					AP	
Mask2Former [3]	216M	SwinL	IN-22K-14M	–	50.1	–
Soft Teacher [36]	284M	SwinL	IN-22K-14M	O365	51.9	52.5
SwinV2-G-HTC++ [23]	3.0B	SwinV2-G	IN-22K-ext-70M [23]	O365	53.4	53.7
MasK DINO(Ours)	**223**M	SwinL	IN-22K-14M	–	**52.6**	–
MasK DINO(Ours)	**223**M	SwinL	IN-22K-14M	O365	**54.5**(+1.1)	–
Panoptic segmentation on COCO					PQ	
Panoptic SegFormer [19]	–M	SwinL	IN-22K-14M	–	55.8	–
Mask2Former [3]	216M	SwinL	IN-22K-14M	–	57.8	–
MasK DINO (ours)	223M	SwinL	IN-22K-14M		**58.4**(+0.6)	–
MasK DINO (ours)	223M	SwinL	IN-22K-14M	O365	**59.4**(+1.6)	–
Semantic segmentation on ADE20K					mIoU	
Mask2Former [3]	215M	SwinL	IN-22K-14M	–	56.1	57.3
SeMask-L MSFaPN-Mask2Former [14]	–M	SwinL-FaPN	IN-22K-14M	–	–	58.2
SwinV2-G-UperNet [23]	3.0B	SwinV2-G	IN-22K-ext-70M [23]	–	59.3	59.9
MasK DINO (ours)	**223M**	SwinL	IN-22K-14M	–	**56.6**	–
MasK DINO (ours)	**223M**	SwinL	IN-22K-14M	O365	**59.5**	**60.8**(+0.9)

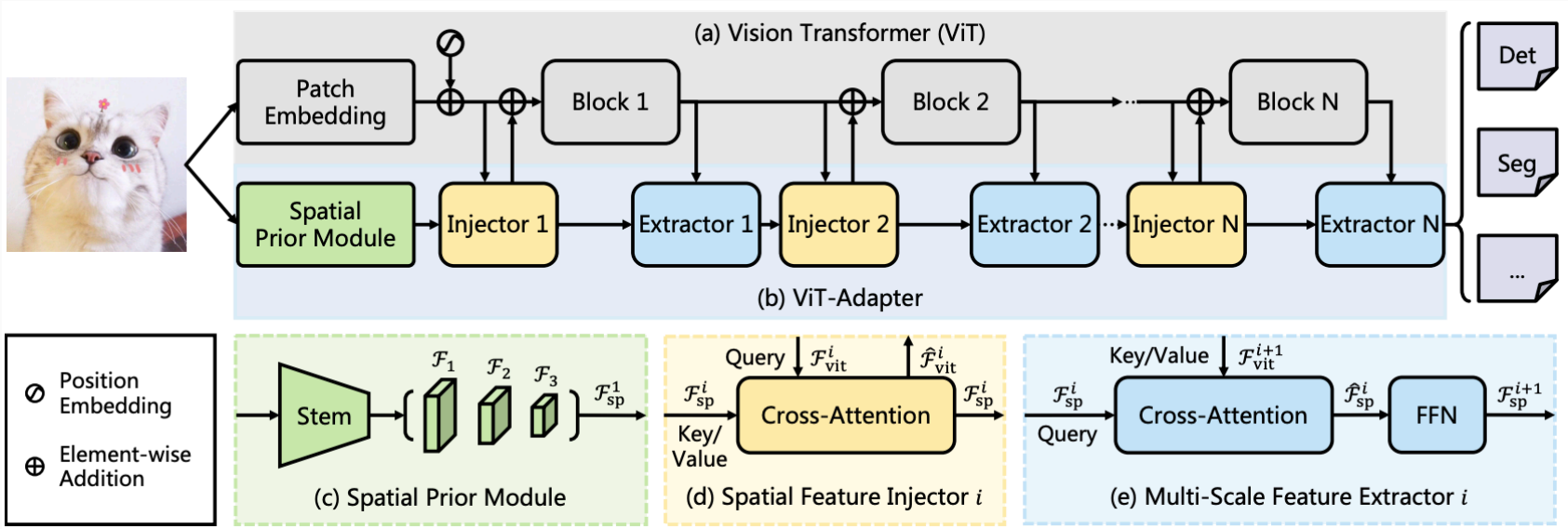
ViT-Adapter

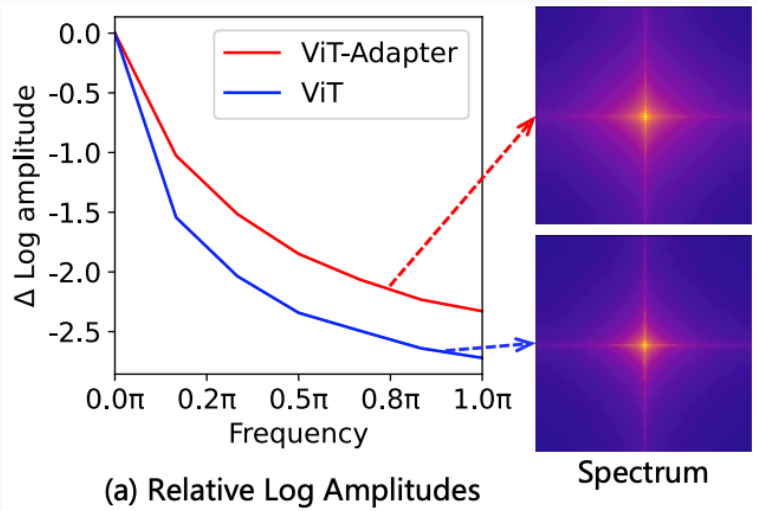
- ViT-Adapter (Chen et al., 2023): адаптер без предобучения, чтобы приблизить ViT к Swin для плотного предсказания

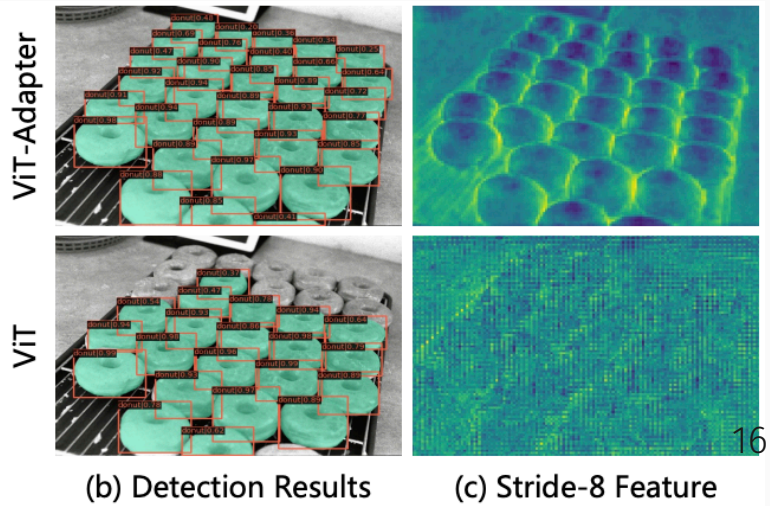


ViT-Adapter

- Адаптер позволяет ViT работать с сегментацией

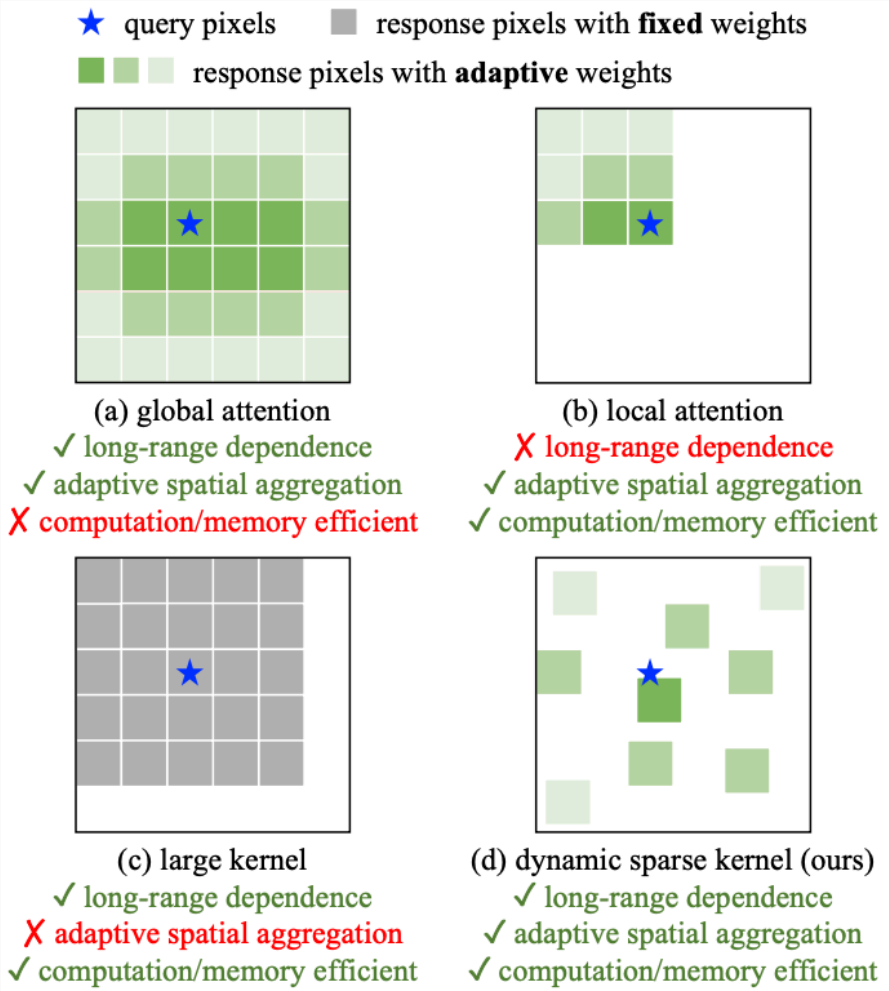






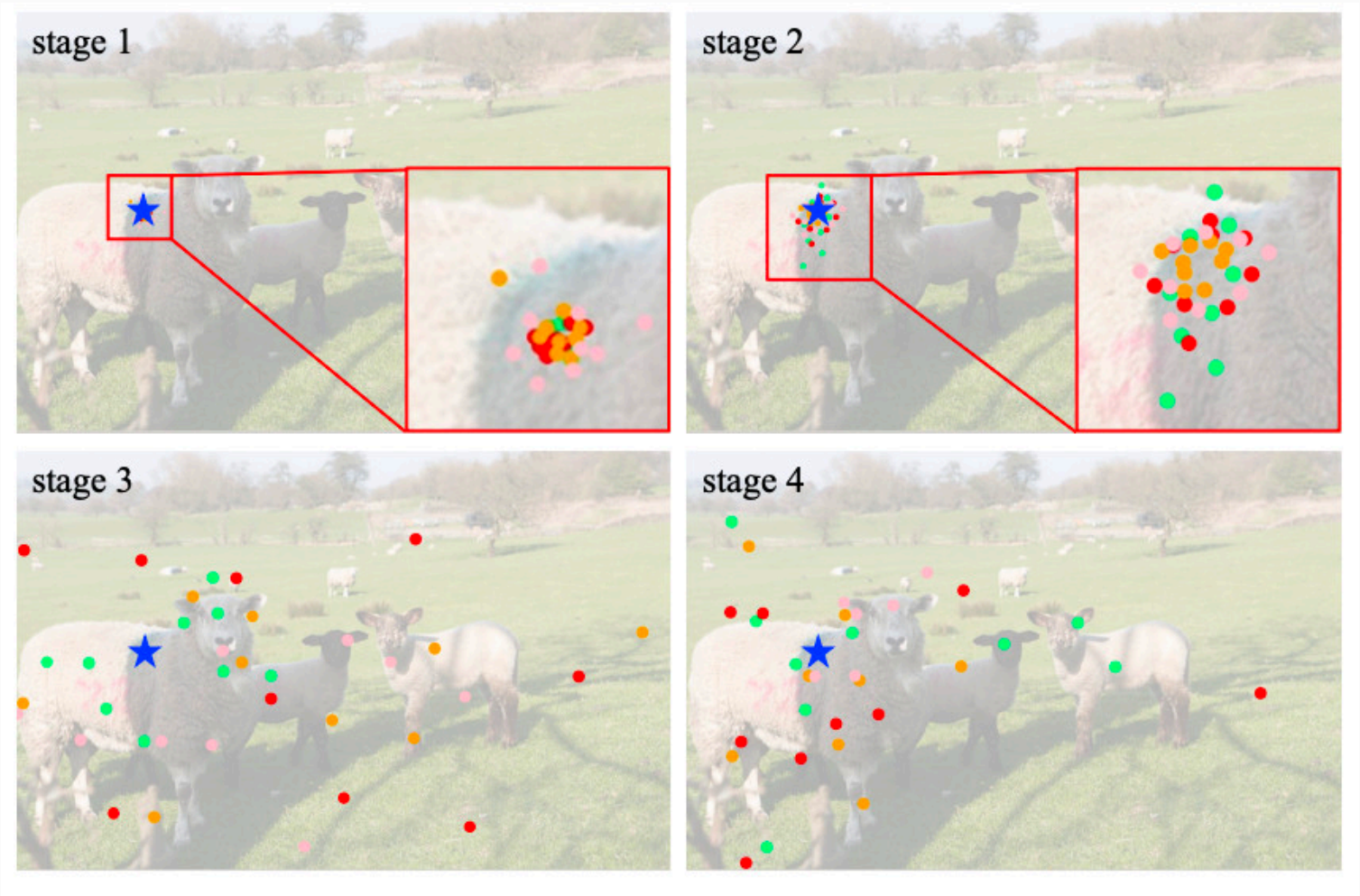
InternImage

- CNN не умерли! InternImage (Wang et al., 2023) – новая базовая модель на основе CNN, использующая деформируемые свертки

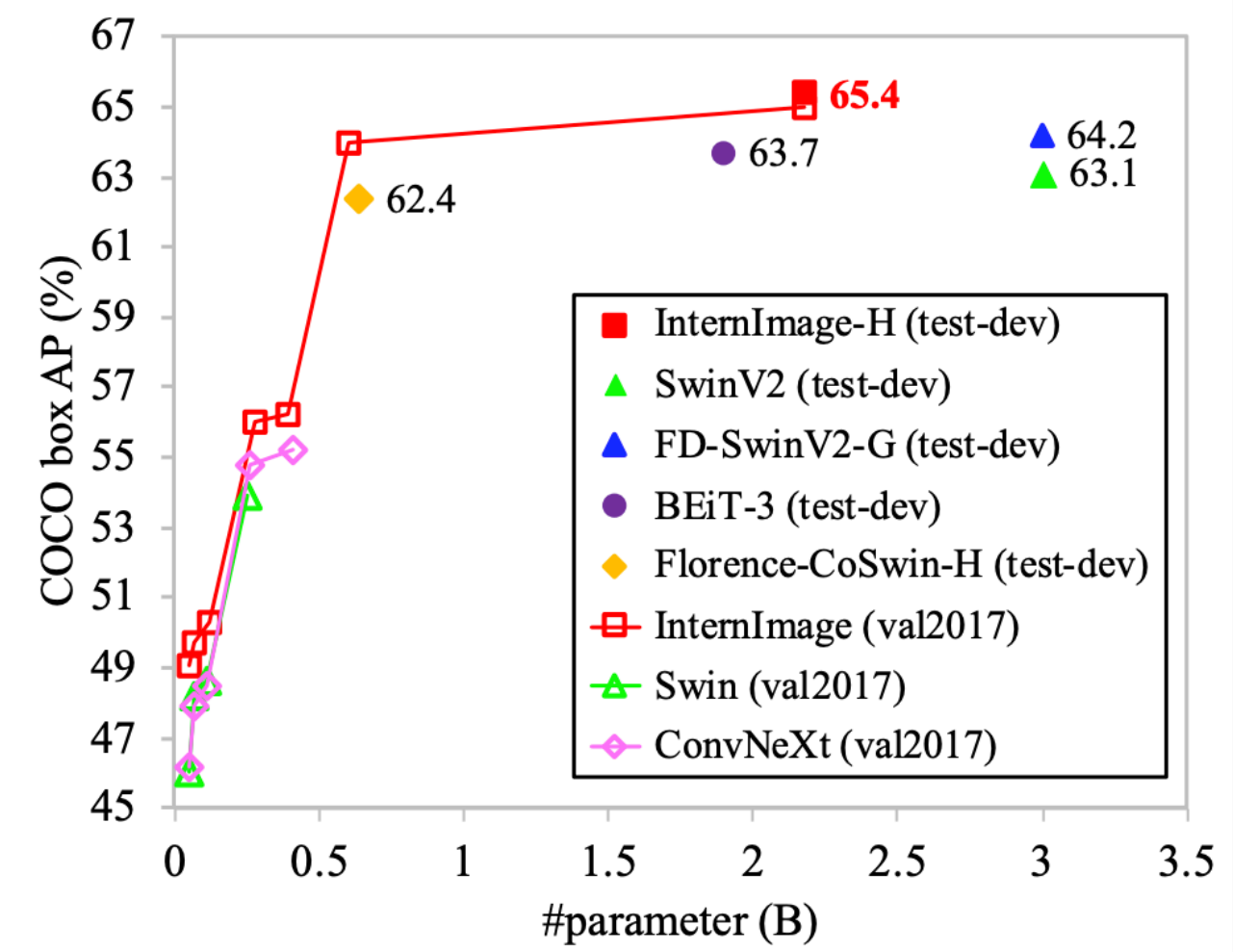


InternImage

- Новая идея – иметь многогрупповую структуру, как разные головы внимания: разные группы имеют разные смещения в свертках и действительно получают разную семантику

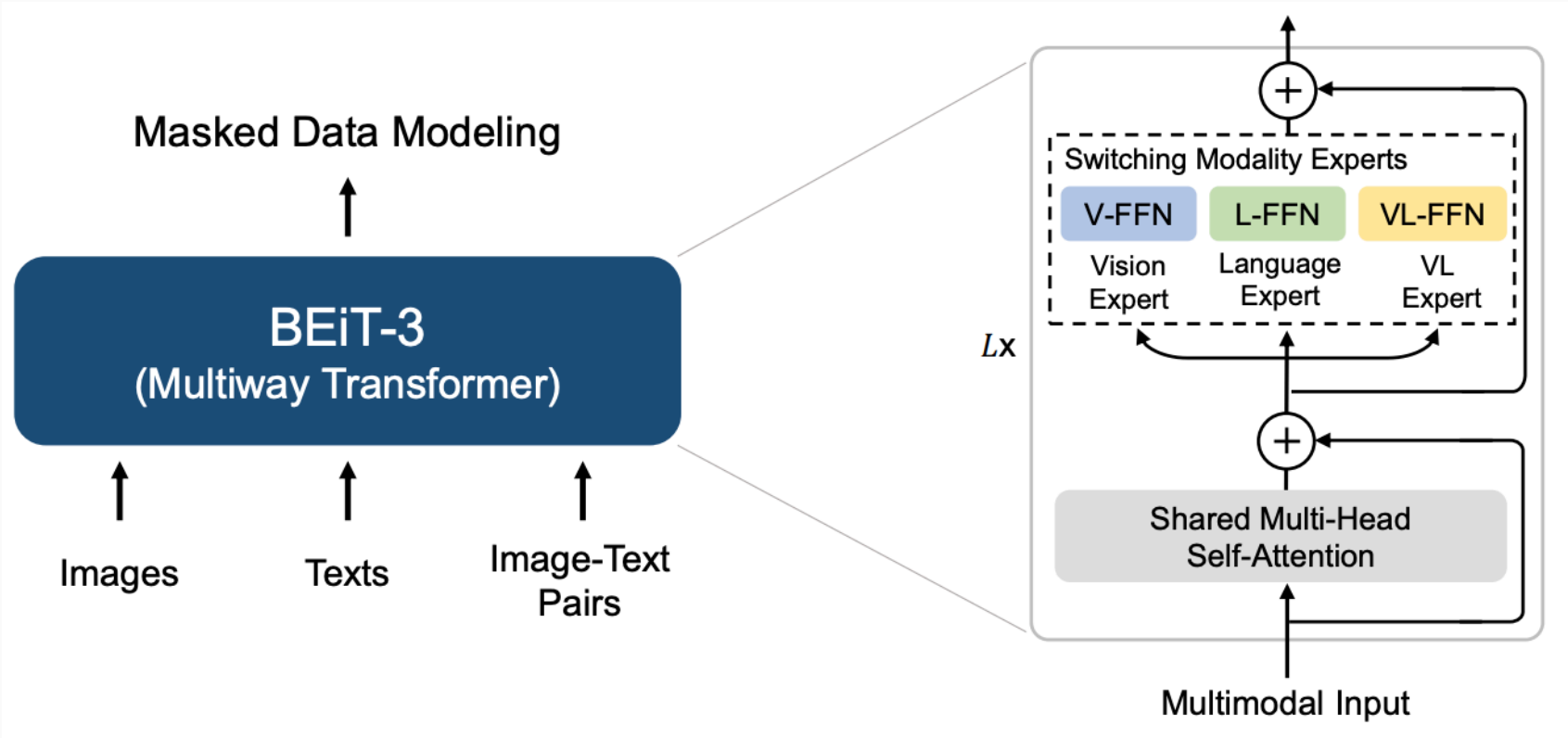


- Хорошо и для распознавания объектов:



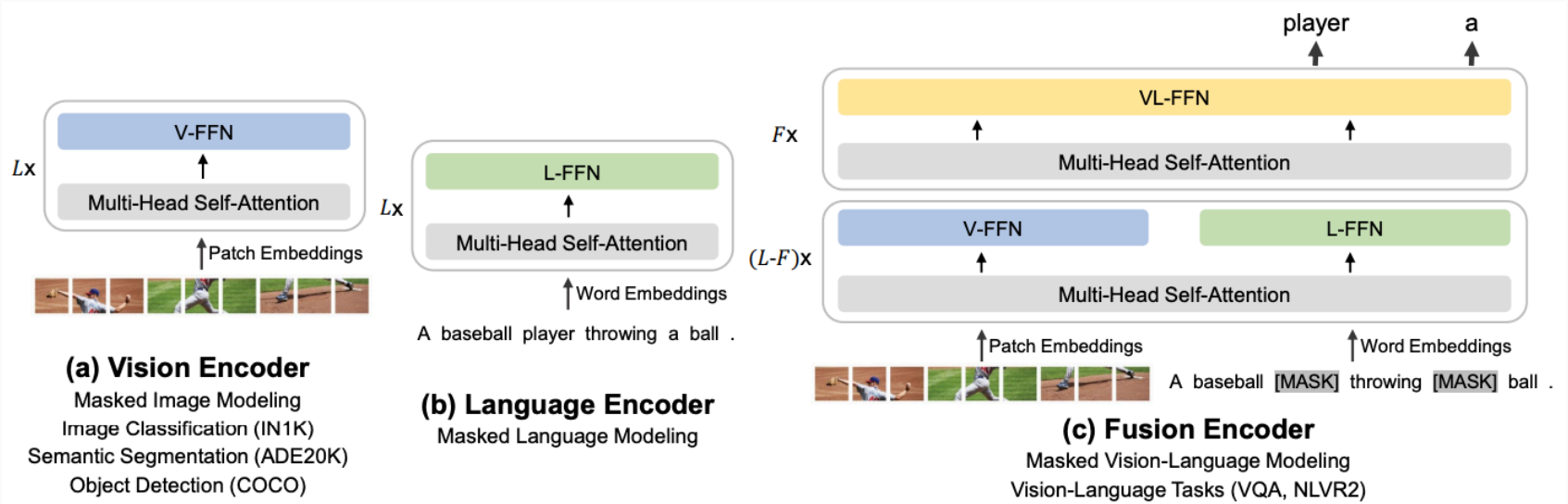
BEiT

- Еще одна большая область – мультимодальные модели; они могут быть очень эффективны для сегментации
- BEiT-3 (Wang et al., 2022): сходятся вместе предобучение языка, визуальные данныы и мультимодальность

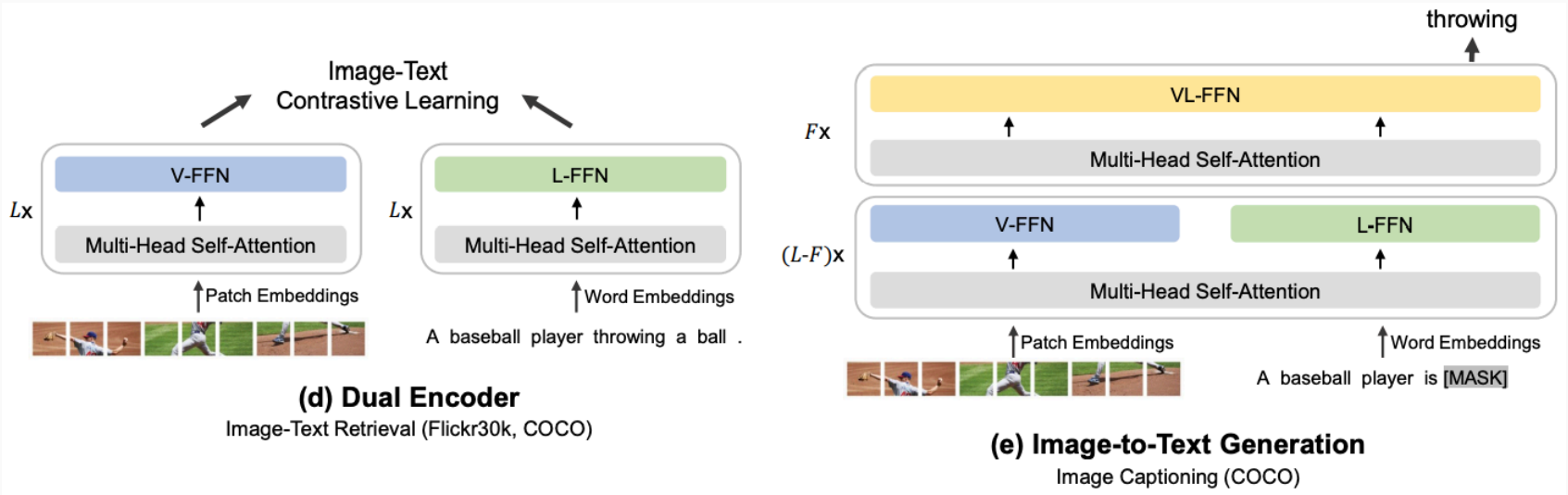


BEiT

- BEiT выполняет маскированное визуально-языковое моделирование во многих модальностях:

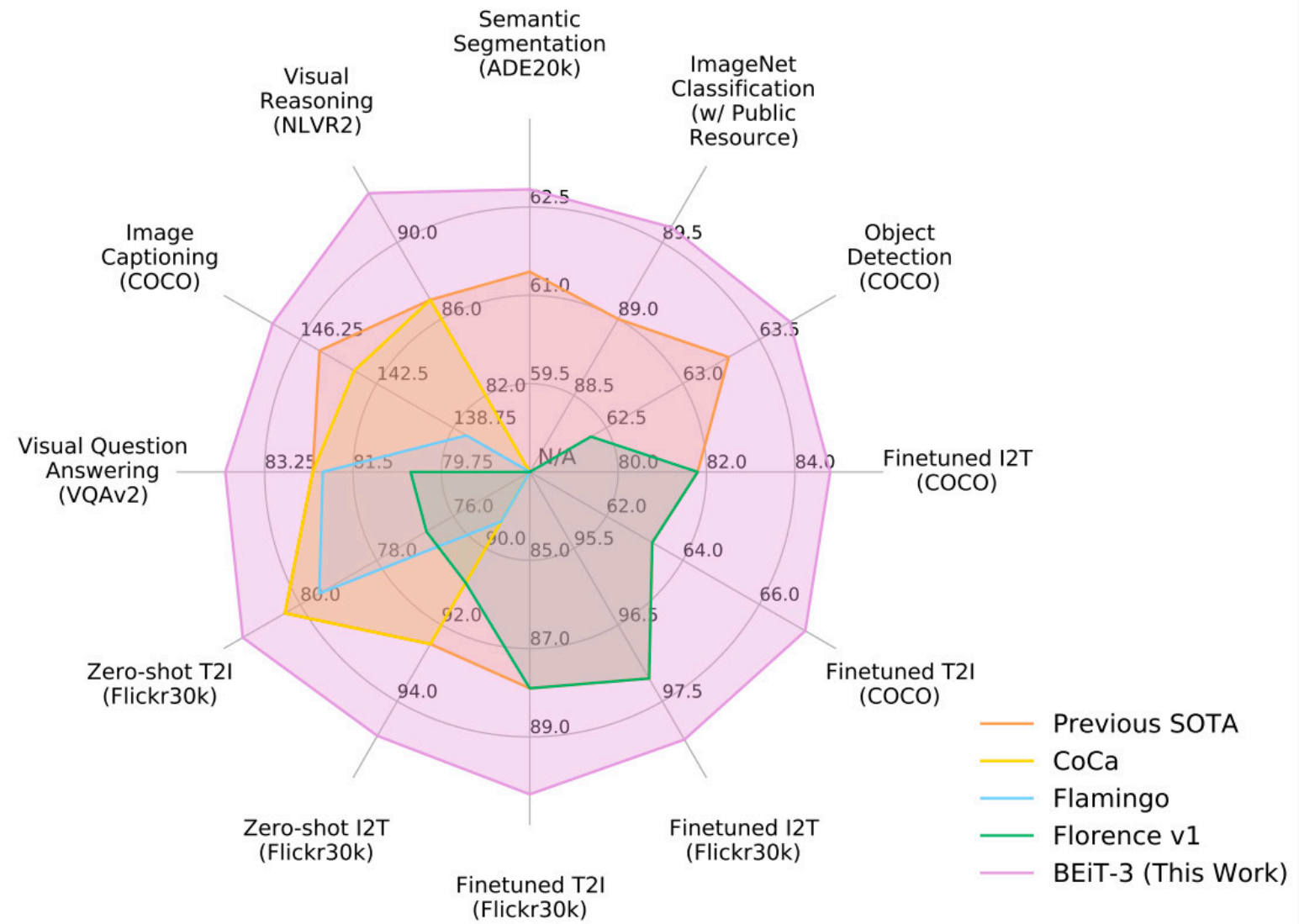


- А также поиск и порождение:



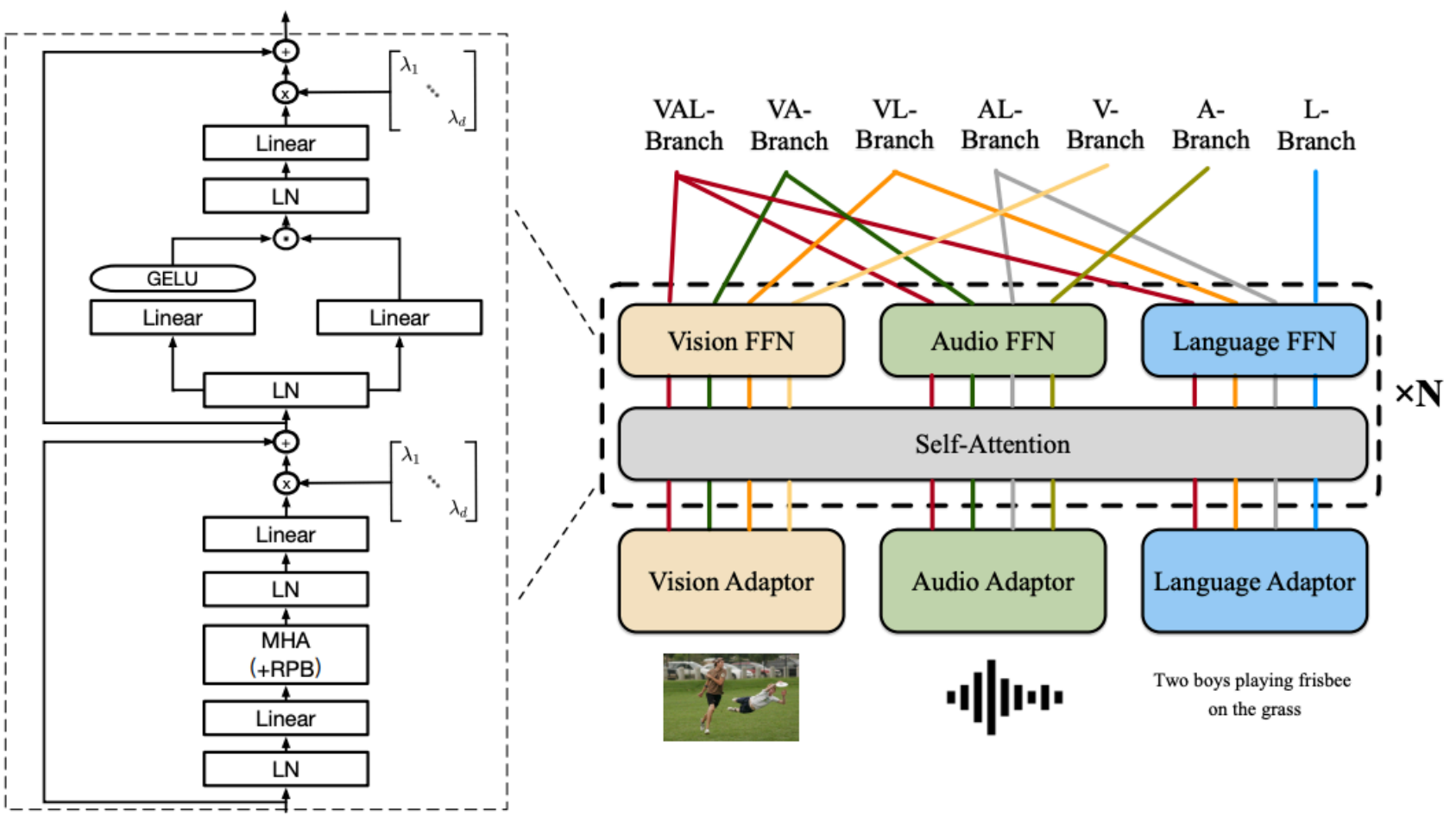
BEiT

- Улучшил многие результаты (в 2022 году):



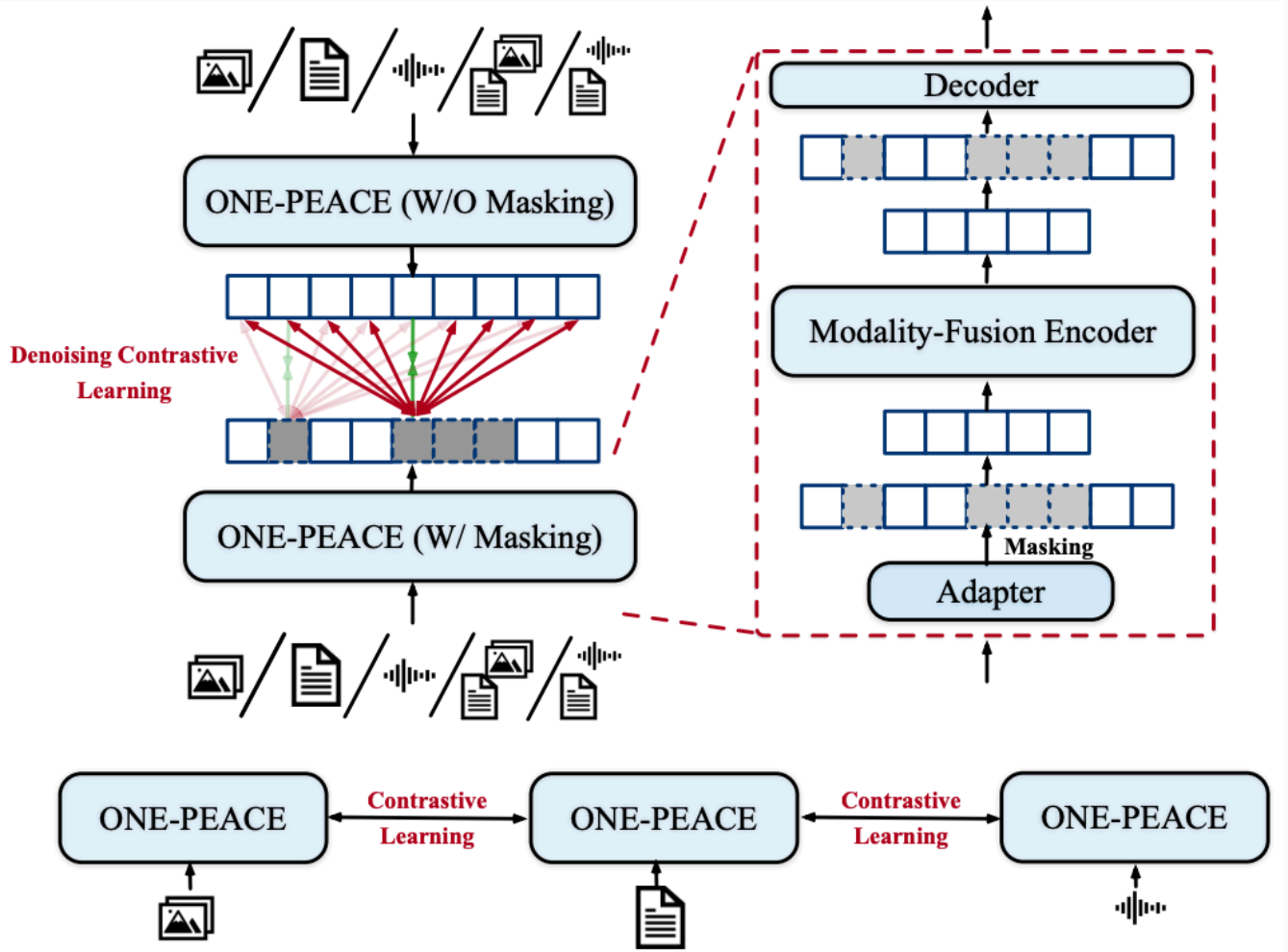
ONE-PEACE

- ONE-PEACE (Wang et al., 2023)



ONE-PEACE

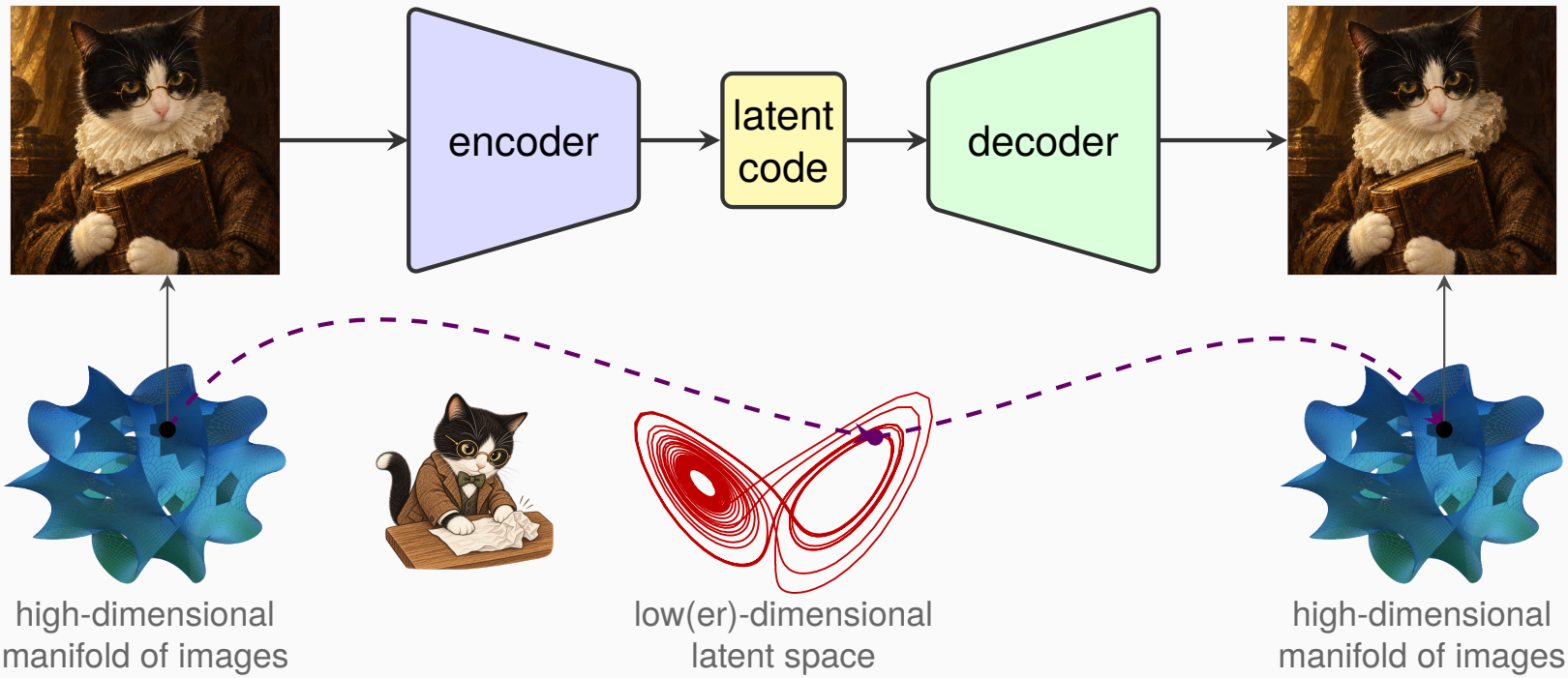
- Задача здесь обычно в том, чтобы изобрести множество задач с self-supervision во всех модальностях; и это хороший момент, чтобы перейти к обсуждению мультимодальных представлений



Вариационные автокодировщики

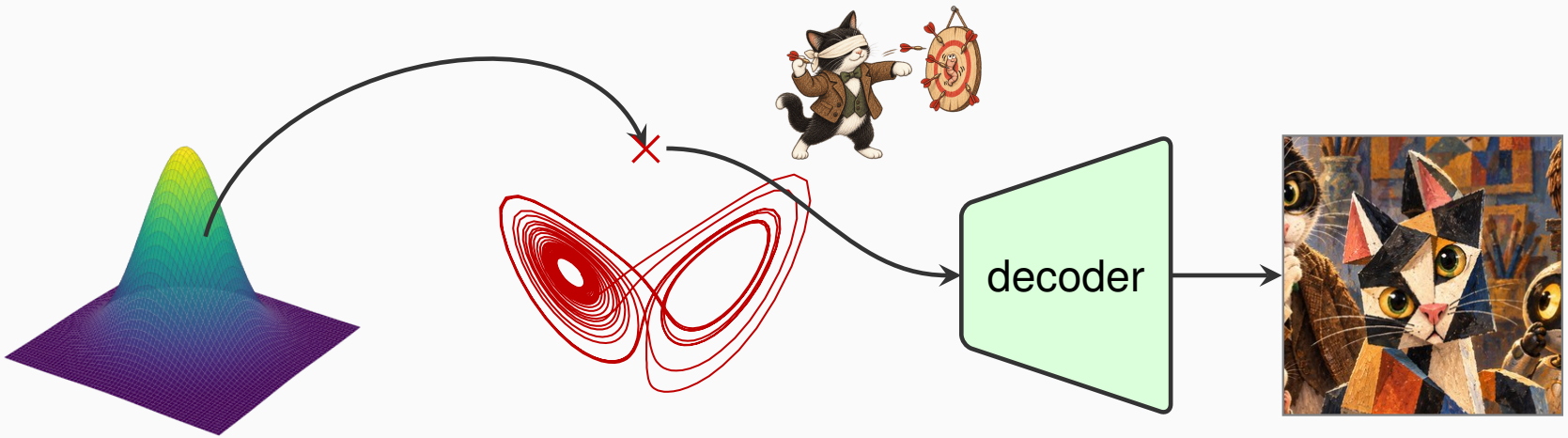
Идея и мотивация

- Главная задача — как всегда: как сделать из обычного автокодировщика порождающую модель?



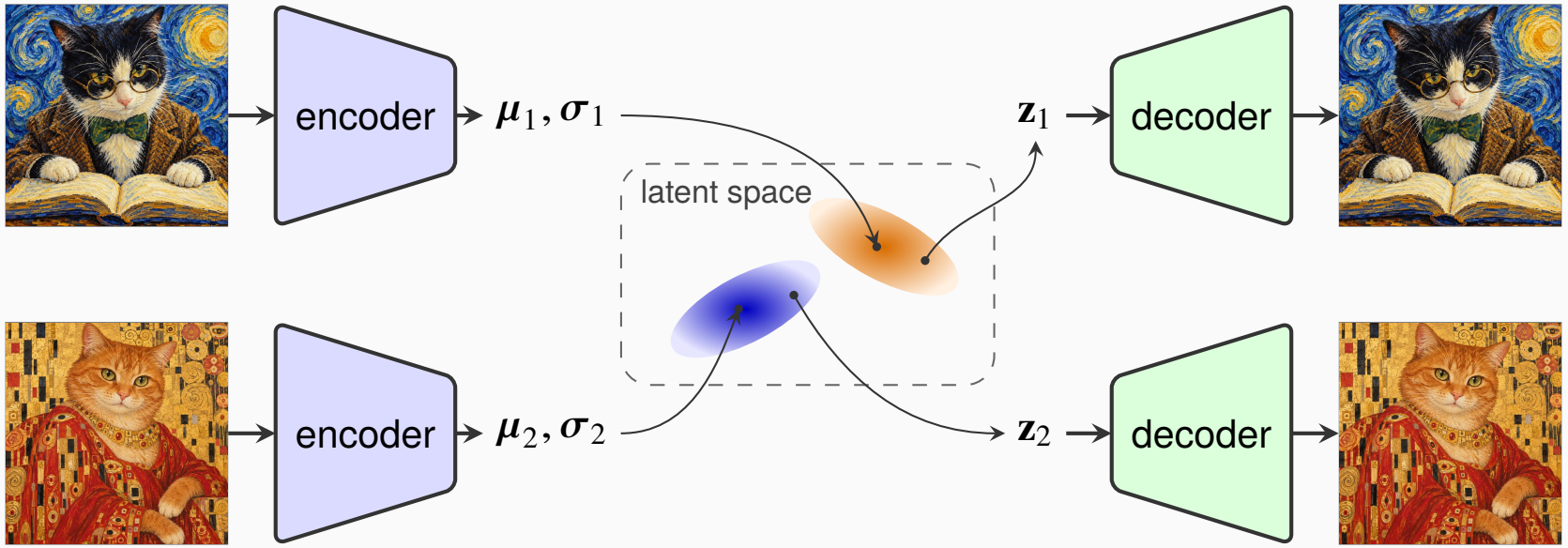
Идея и мотивация

- Даже в пространстве низкой размерности нелегко выбрать хорошую точку:



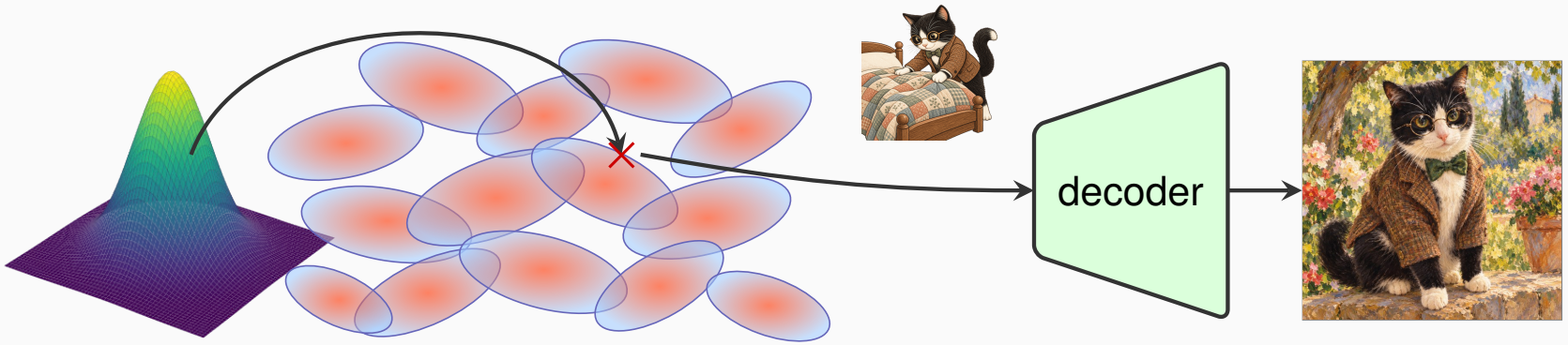
Идея и мотивация

- Главная идея вариационного автокодировщика (Kingma et al., 2013) — пусть из каждой точки $\mathbf{x}$ получается не точка $\mathbf{z}$, а целое распределение $p(\mathbf{z}|\mathbf{x})$:



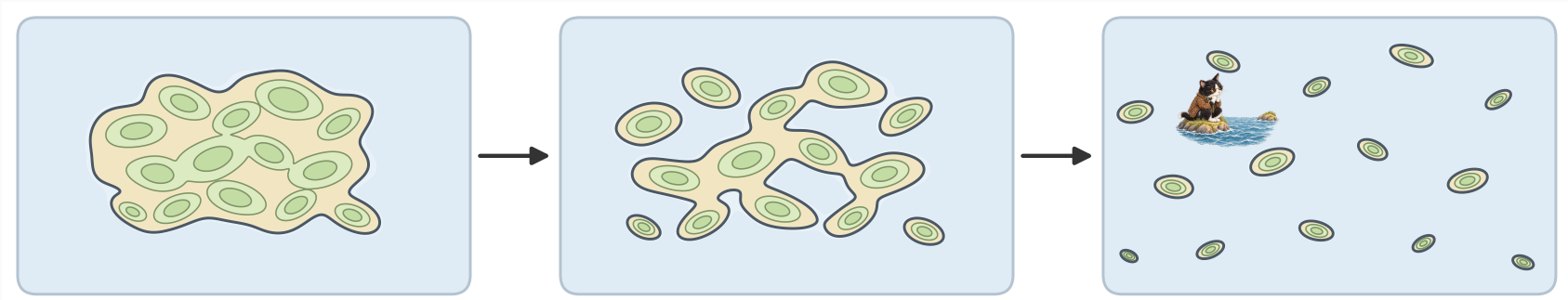
Идея и мотивация

- Интуиция в том, что декодеру приходится быть устойчивым к небольшим изменениям в $\mathbf{z}$; некий более идейный аналог шумоподавляющего автокодировщика
- В идеале мы покроем пространство латентных кодов достаточно «широкими» распределениями:



Идея и мотивация

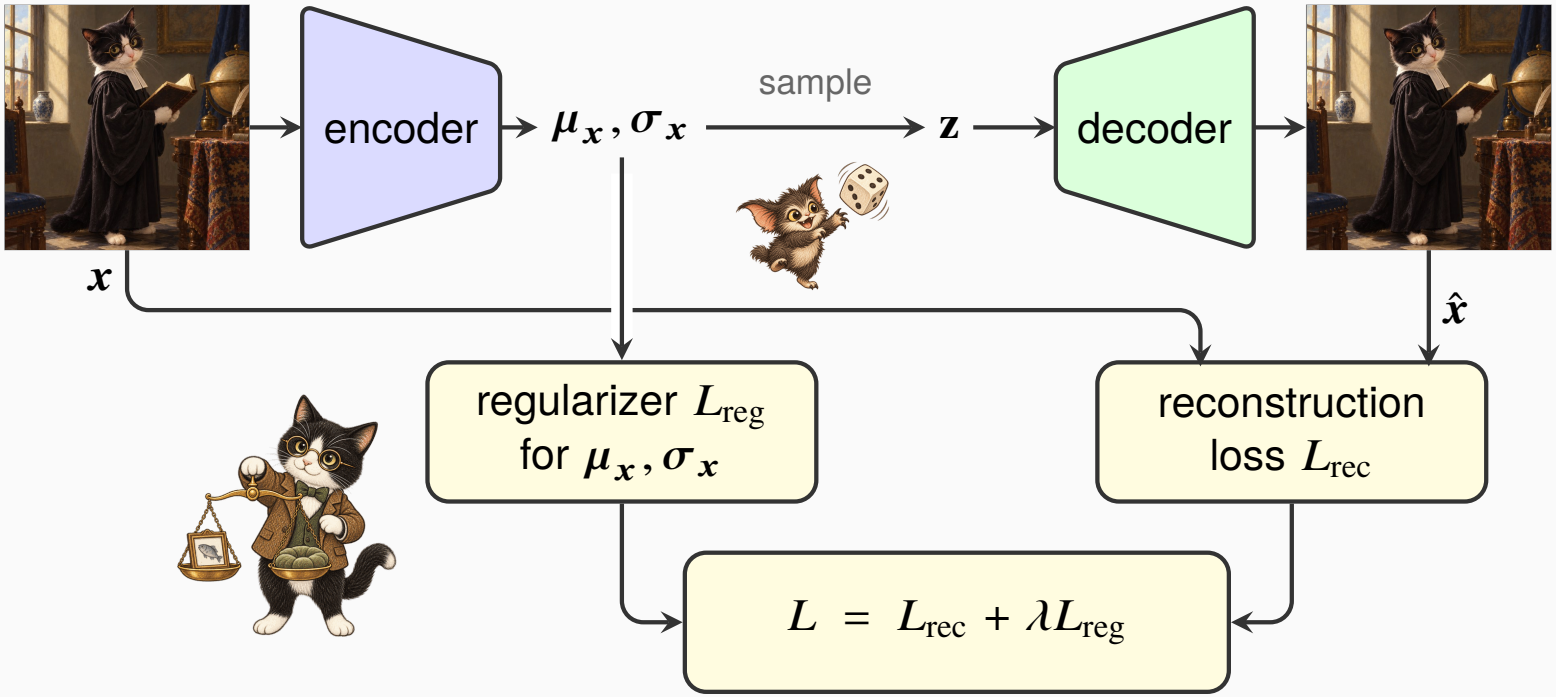
- Однако есть проблема: если просто обучать автокодировщик, ему будет легче декодировать «узкие» распределения, и мы в итоге придём к тем же точкам с нулевой дисперсией



- Как решить эту проблему?

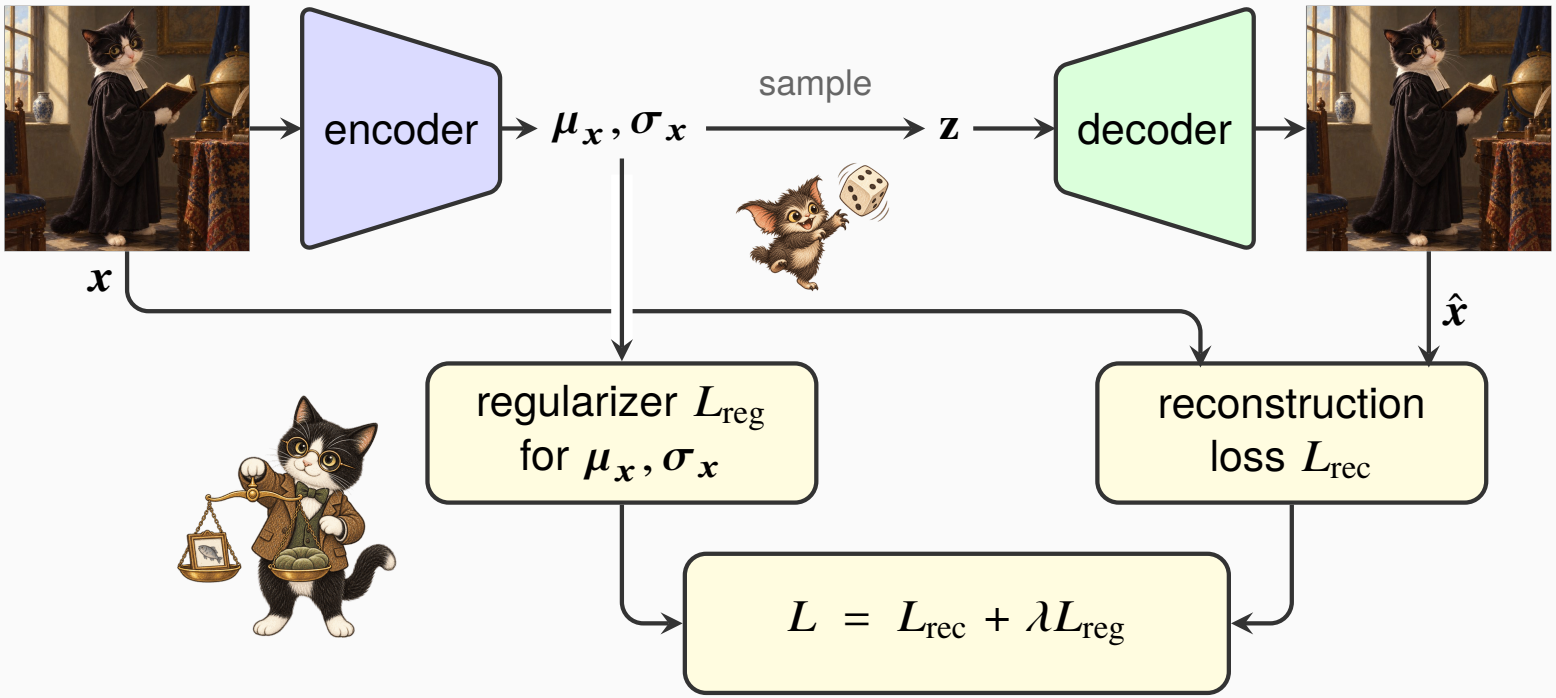
Идея и мотивация

- Нужно добавить некое ограничение на $p(\mathbf{z}|\mathbf{x})$, то есть регуляризовать его:



Идея и мотивация

- Теперь остались два вопроса:
  - какие выбрать функцию ошибки и регуляризатор?
  - как протаскивать градиенты через сэмплирование?
- Давайте на оба этих вопроса ответим...



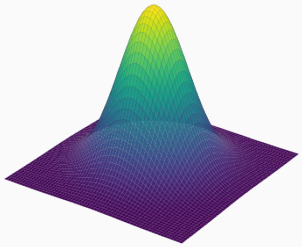
Вариационные приближения в VAE

- Давайте начнём с начала — мы хотим обучить кодировщик делать латентные коды $\mathbf{z}$ из картинок $\mathbf{x}$ и декодировщик, который $\mathbf{z}$ превращает обратно в $\mathbf{x}$.
- Совместное распределение можно разложить по-разному:

$$p(\mathbf{z})\,p(\mathbf{x}|\mathbf{z}) = p(\mathbf{x},\mathbf{z}) = p(\mathbf{x})\,p(\mathbf{z}|\mathbf{x})$$

$$p(\mathbf{z})\;\; p(\boldsymbol{x}|\mathbf{z}) \;\; = \;\; p(\boldsymbol{x},\mathbf{z}) \;\; = \;\; p(\boldsymbol{x})\;\; p(\mathbf{z}|\boldsymbol{x})$$

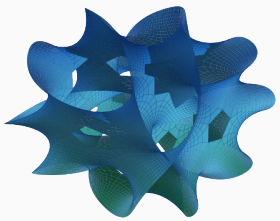
distribution we want to sample $\mathbf{z}$ from



decoder distribution



high-dimensional manifold of real images



encoder distribution

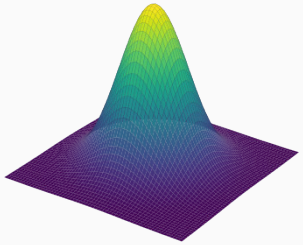


Вариационные приближения в VAE

- Здесь:
  - $p(\mathbf{x})$ — это распределение изображений,
  - $p(\mathbf{z})$ — это распределение латентных кодов, т.е. простое распределение, из которого можно сэмплировать;
  - а остальные два нам нужно найти — это распределение кодировщика $p(\mathbf{z}|\mathbf{x})$ и декодировщика $p(\mathbf{x}|\mathbf{z})$.

$$p(\mathbf{z})\ p(\boldsymbol{x}|\mathbf{z}) = p(\boldsymbol{x}, \mathbf{z}) = p(\boldsymbol{x})\ p(\mathbf{z}|\boldsymbol{x})$$

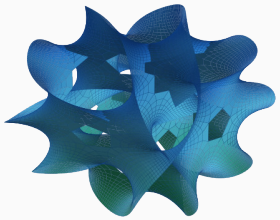
distribution we want to sample $\mathbf{z}$ from



decoder distribution



high-dimensional manifold of real images



encoder distribution



Вариационные приближения в VAE

- Для порождающей модели надо обучить и $p(\mathbf{x}|\mathbf{z})$, и $p(\mathbf{z}|\mathbf{x})$.
- Но у нас $p(\mathbf{z})$ — это простое распределение, а $p(\mathbf{x})$ — очень сложное!
- Значит, придётся выбрать одно из двух:
  - или предположим, что $p(\mathbf{x}|\mathbf{z})$ простое, а затем будем искать сложное $p(\mathbf{z}|\mathbf{x})$;
  - или наоборот, предположим, что $p(\mathbf{z}|\mathbf{x})$ простое, а затем будем искать сложное $p(\mathbf{x}|\mathbf{z})$.

Вариационные приближения в VAE

- В VAE удобнее использовать первый вариант: предположим, что

$$p(\mathbf{x}|\mathbf{z}) = N(\mathbf{x}|f(\mathbf{z}), c\mathbf{I}), \quad \text{где} \quad f(\mathbf{z}) = \text{Decoder}(\mathbf{z})$$

- Слева получается $p(\mathbf{x}|\mathbf{z})$ умножить на $p(\mathbf{z}) = N(\mathbf{z}|0, \mathbf{I})$, то есть совместное распределение $p(\mathbf{x}, \mathbf{z})$.

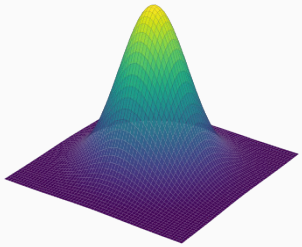
$$p(\mathbf{z})\ p(\boldsymbol{x}|\mathbf{z}) = p(\boldsymbol{x}, \mathbf{z}) = p(\boldsymbol{x})\ p(\mathbf{z}|\boldsymbol{x})$$

distribution we want to sample $\mathbf{z}$ from

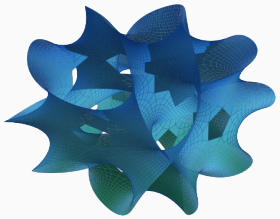
decoder distribution

high-dimensional manifold of real images

encoder distribution









Вариационные приближения в VAE

- А справа будем делать приближение $q(\mathbf{z}) \approx p(\mathbf{z}|\mathbf{x})$:

$$p(\mathbf{z})\ p(\boldsymbol{x}|\mathbf{z}) = p(\boldsymbol{x}, \mathbf{z}) = p(\boldsymbol{x})\ p(\mathbf{z}|\boldsymbol{x}) \approx q(\mathbf{z})$$

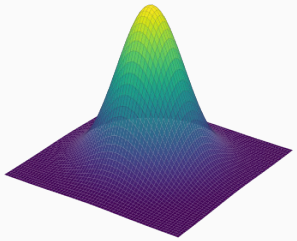


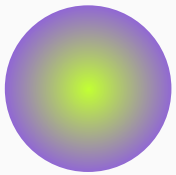
distribution we want to sample $\mathbf{z}$ from

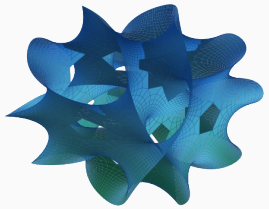
decoder distribution $\mathcal{N}(\boldsymbol{x}|f(\mathbf{z}), c\mathbf{I})$

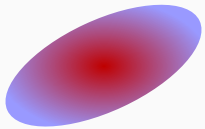
high-dimensional manifold of real images

encoder distribution $\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\sigma}^2_{\boldsymbol{x}})$









- А именно вариационное приближение...

Вариационные приближения в VAE

- Вспомним идею вариационных приближений:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x})\, p(\mathbf{z}|\mathbf{x}),$$

$$\log p(\mathbf{x}) = \log p(\mathbf{x}, \mathbf{z}) - \log p(\mathbf{z}|\mathbf{x}),$$

возьмём ожидание по $q(\mathbf{z})$:

$$\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x})] = \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}, \mathbf{z})] - \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{z}|\mathbf{x})],$$

а затем сделаем стандартные преобразования:

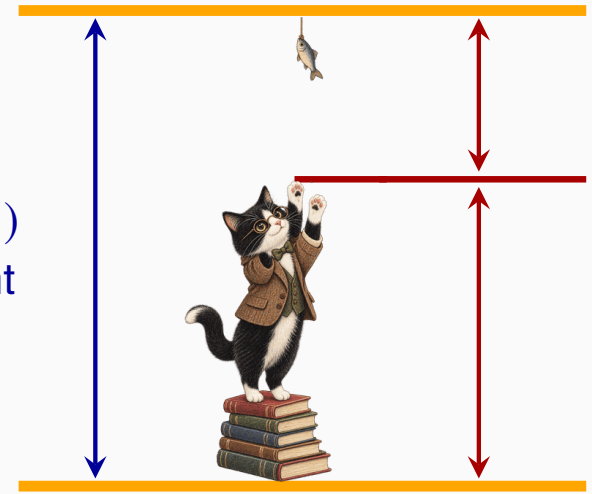
$$\begin{aligned}\log p(\mathbf{x}) =& \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}, \mathbf{z})] - \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})] + \\ &+ \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z})] - \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{z}|\mathbf{x})], \\ \log p(\mathbf{x}) =& \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \mathrm{d}\mathbf{z} + \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})} \mathrm{d}\mathbf{z}.\end{aligned}$$

Вариационные приближения в VAE

- Итого получается, что

$$\log p\left(\mathbf{x}\right) = L(q) + \mathrm{KL}\left(q\left(z\right) \| p\left(\mathbf{z}|\mathbf{x}\right)\right), \quad \text{где}$$

$$L(q) = \int q\left(\mathbf{z}\right) \log \frac{p\left(\mathbf{x}, \mathbf{z}\right)}{q\left(\mathbf{z}\right)} \mathrm{d}\mathbf{z}.$$



Kullback–Leibler divergence

$$\mathrm{KL}\left(q\left(\mathbf{z}\right) \| p\left(\mathbf{z}|\boldsymbol{x}\right)\right) = \int q\left(\mathbf{z}\right) \log \frac{q\left(\mathbf{z}\right)}{p\left(\mathbf{z}|\boldsymbol{x}\right)} \mathrm{d}\mathbf{z}$$

evidence lower bound (ELBO)

$$\mathcal{L}(q) = \int q\left(\mathbf{z}\right) \log \frac{p\left(\boldsymbol{x}, \mathbf{z}\right)}{q\left(\mathbf{z}\right)} \mathrm{d}\mathbf{z}$$

Вариационные приближения в VAE

- Значит, можно аппроксимировать $p(\mathbf{z}|\mathbf{x})$ через $q(\mathbf{z})$, максимизируя $L(q)$. В случае VAE его можно найти:

$$
\begin{aligned}
L(q) &= \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \mathrm{d}\mathbf{z} = \int q(\mathbf{z}) \log \frac{p(\mathbf{z})\, p(\mathbf{x}|\mathbf{z})}{q(\mathbf{z})} \mathrm{d}\mathbf{z} \\
&= \int q(\mathbf{z}) \log p(\mathbf{x}|\mathbf{z}) \mathrm{d}\mathbf{z} + \int q(\mathbf{z}) \log \frac{p(\mathbf{z})}{q(\mathbf{z})} \mathrm{d}\mathbf{z} \\
&= \int q(\mathbf{z}) \log N(\mathbf{x}|f(\mathbf{z}), c\mathbf{I}) \mathrm{d}\mathbf{z} - \mathrm{KL}(q(\mathbf{z}) \| p(\mathbf{z})) \\
&= -\frac{1}{2c} \mathbb{E}_{q(\mathbf{z})} \left[\|\mathbf{x} - f(\mathbf{z})\|^2 \right] - \mathrm{KL}(q(\mathbf{z}) \| p(\mathbf{z})) .
\end{aligned}
$$

Вариационные приближения в VAE

- И вот получилось как раз то, чего мы ожидали интуитивно:

$$-\mathcal{L}(q) = \frac{1}{2c}\mathbb{E}_{q(\mathbf{z})}\left[\|\boldsymbol{x} - f(\mathbf{z})\|^2\right] + \mathrm{KL}\left(q\left(\mathbf{z}\right)\|p\left(\mathbf{z}\right)\right)$$



reconstruction loss L_{rec}

regularizer L_{reg} for $\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\sigma}_{\boldsymbol{x}}$

$$L = L_{\text{rec}} + \lambda L_{\text{reg}}$$

Вариационные приближения в VAE

- Только теперь мы знаем, как выглядит L_{rec}, да и KL-дивергенцию между двумя гауссианами можно подсчитать:

$$\mathrm{KL}\left(q\left(\mathbf{z}\right)\|p\left(\mathbf{z}\right)\right) = \frac{1}{2}\sum_{j=1}^{d}\left(\sigma_{\mathbf{x},j}^2 + \mu_{\mathbf{x},j}^2 - \log\sigma_{\mathbf{x},j}^2 - 1\right).$$

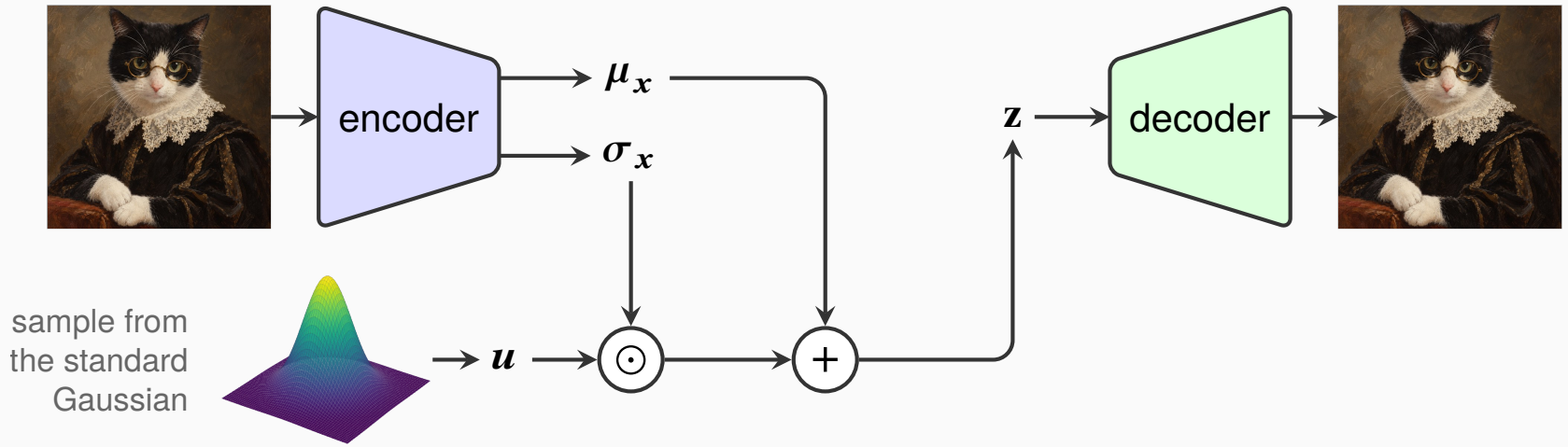
- Получается окончательная функция ошибки для вариационного автокодировщика:

$$L(q) = -\frac{1}{2c}\mathbb{E}_{q(\mathbf{z})}\left[\|\mathbf{x} - f(\mathbf{z})\|^2\right] - \frac{1}{2}\sum_{j=1}^{d}\left(\sigma_{\mathbf{x},j}^2 + \mu_{\mathbf{x},j}^2 - \log\sigma_{\mathbf{x},j}^2 - 1\right).$$

- Осталась только проблема с сэмплированием...

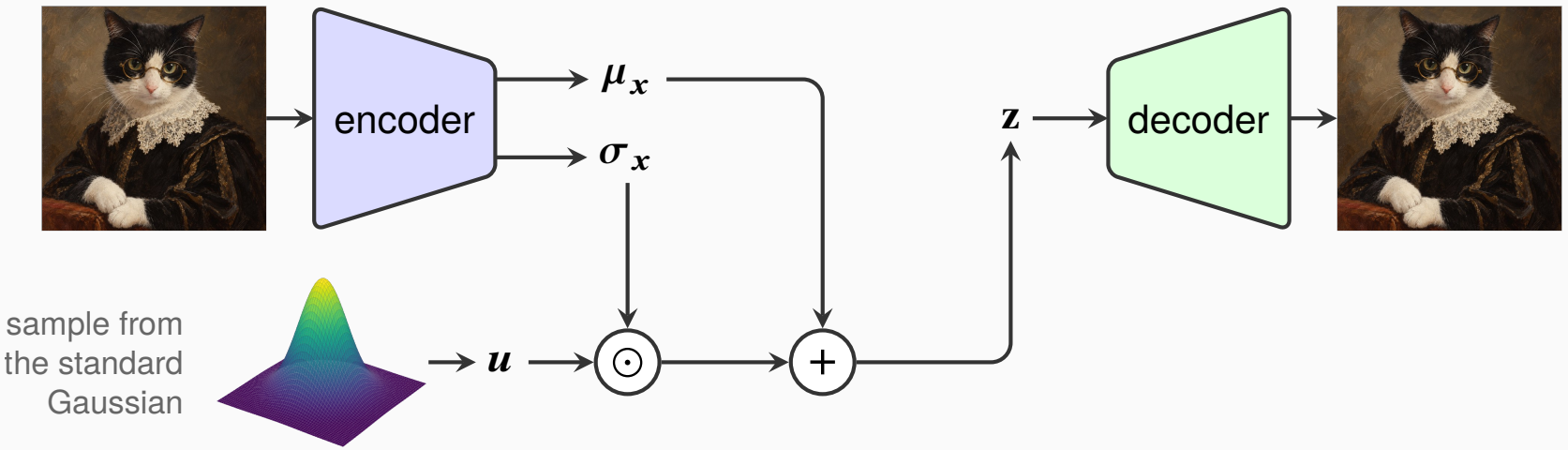
Reparametrization trick

- Чтобы обучить модель, нужно пропустить градиент через $\mathbf{z}$ обратно к кодировщику, но там сэмплирование
- Reparametrization trick: давайте сначала сэмплировать, а потом преобразовывать результат!



Reparametrization trick

- И теперь уже точно все компоненты на месте: сэмплируем мини-батч $\mathbf{u}$ для мини-батча $\mathbf{x}$ и используем их для обучения



Спасибо за внимание!



