Лекция 6. Задачи компьютерного зрения и сверточные нейронные сети

Бабич Павел Васильевич

ИММиКН

2026

Что именно должна предсказывать модель?

Пусть $X \in \mathbb{R}^{H \times W \times C}$, а $\Omega = \{1, \dots, H\} \times \{1, \dots, W\}$.

Задача	Пространство выходов
Классификация	Один класс $c \in \{1, \dots, K\}$ для изображения
Детекция	Множество рамок и классов $\{(b_n, c_n)\}_{n=1}^{N}$
Семантическая	Карта $Y : \Omega \to \{1, \dots, K\}$
Экземплярная	Множество масок и классов $\{(M_n, c_n)\}_{n=1}^{N}$
Паноптическая	Карта пар $Y(u) = (c(u), \iota(u))$

K — число классов, N — число объектов, ι — идентификатор экземпляра.

Семантическая сегментация: распределение класса

Сеть выдаёт логиты $Z = f_\theta(X) \in \mathbb{R}^{H \times W \times K}$:

$$p_{u,k} = \frac{\exp Z_{u,k}}{\sum_{j=1}^{K} \exp Z_{u,j}}, \qquad \widehat{Y}(u) = \arg\max_k p_{u,k}.$$

Для размеченных пикселей Ω_{lab}:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{|\Omega_{\text{lab}}|} \sum_{u \in \Omega_{\text{lab}}} \log p_{u,Y(u)}.$$

- Один пиксель имеет один класс, включая фон, если он предусмотрен.
- Два объекта одного класса получают одинаковую метку.
- Сумма потерь по пикселям не означает независимость их признаков.

Экземплярная сегментация: множество объектов

$$\mathcal{Y} = \{(M_n, c_n)\}_{n=1}^{N}, \qquad M_n \in \{0,1\}^{H \times W}.$$

Число объектов N меняется от изображения к изображению.

$$\{(M_1, c_1), (M_2, c_2)\} = \{(M_2, c_2), (M_1, c_1)\}.$$

- Порядок экземпляров не несёт смысла: нужна процедура сопоставления.
- Маски предсказаний могут перекрываться, правила зависят от задачи.
- Связные компоненты семантической маски не всегда дают экземпляры: соприкасающиеся объекты могут образовывать одну компоненту.

Паноптическая сегментация: класс и экземпляр

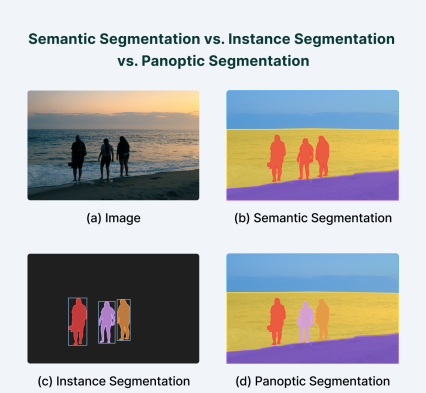
Разделим классы: $\mathcal{C} = \mathcal{C}_{\text{thing}} \cup \mathcal{C}_{\text{stuff}}$.

$$Y(u) = (c(u), \iota(u)), \qquad \begin{cases} \iota(u) \in \mathbb{N}_{>0}, & c(u) \in \mathcal{C}_{\text{thing}}, \\ \iota(u) = 0, & c(u) \in \mathcal{C}_{\text{stuff}}. \end{cases}$$

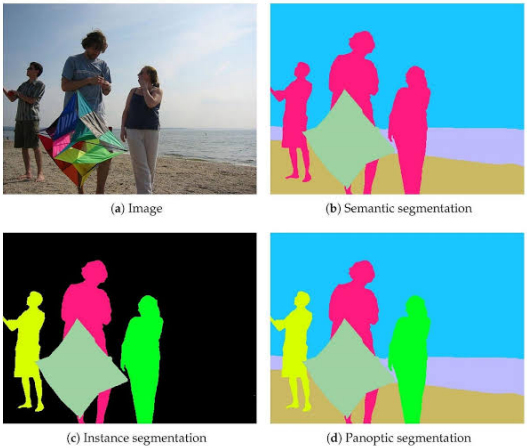
- *Things*: счётные объекты, например люди и автомобили.
- *Stuff*: области, например дорога, небо, трава.
- Каждый валидный пиксель относится ровно к одному сегменту.
- Разметка может содержать исключённые области *void/ignore*.

Идентификатор ι имеет смысл внутри изображения, а не между снимками.

Семантическая, экземплярная и паноптическая маски



Семантическая, экземплярная и паноптическая маски



(a) Image (b) Semantic segmentation

(c) Instance segmentation (d) Panoptic segmentation

Тип маски определяется вопросом исследования

Пример	Требуемый результат	Постановка
Площадь поражения	Маска поражённой ткани	Бинарная семантическая
Подсчёт клеток	Отдельная маска каждой клетки	Экземплярная
Органы на КТ	Метка органа у каждого вокселя	Многоклассовая семантическая
Полное описание улицы	Области и отдельные объекты	Паноптическая

Бинарность / многоклассовость и разделение на экземпляры — разные оси постановки.

Метрики семантической сегментации

Для класса k:

$$\mathrm{IoU}_k = \frac{TP_k}{TP_k + FP_k + FN_k}, \qquad \mathrm{Dice}_k = \frac{2TP_k}{2TP_k + FP_k + FN_k}.$$

$$\mathrm{Dice}_k = \frac{2\mathrm{IoU}_k}{1 + \mathrm{IoU}_k}, \qquad \mathrm{mIoU} = \frac{1}{|\mathcal{K}_{\mathrm{eval}}|} \sum_{k \in \mathcal{K}_{\mathrm{eval}}} \mathrm{IoU}_k.$$

- Связь Dice и IoU верна для одной пары масок, не для средних метрик.
- Нужно определить обработку отсутствующих классов и ignore-пикселей.
- При 99% фона предсказание «везде фон» даёт 99% pixel accuracy, но нулевой IoU объекта.

Функции потерь при дисбалансе

Для бинарной маски $y_u \in \{0,1\}$ и $p_u \in [0,1]$:

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_u p_u y_u + \varepsilon}{\sum_u p_u + \sum_u y_u + \varepsilon}.$$

$$p_{t,u} = \begin{cases} p_u, & y_u = 1, \\ 1 - p_u, & y_u = 0, \end{cases} \qquad \mathcal{L}_{\text{focal}} = -\frac{1}{|\Omega|}\sum_u \alpha_{t,u}(1 - p_{t,u})^\gamma \log p_{t,u}.$$

$$\mathcal{L} = \lambda_{\text{CE}}\mathcal{L}_{\text{CE}} + \lambda_{\text{Dice}}\mathcal{L}_{\text{Dice}}.$$

Soft Dice связывает пиксели через сумму. Нужно явно выбрать усреднение по изображениям / batch / классам. Эти потери не привязаны к одной архитектуре.

Panoptic quality

Сопоставление сегментов одного класса при $\mathrm{IoU} > 0.5$:

$$\mathrm{PQ} = \frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p,g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} = \underbrace{\frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p,g)}{|TP|}}_{\mathrm{SQ}} \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\mathrm{RQ}}.$$

- SQ оценивает маски сопоставленных сегментов.
- RQ учитывает пропущенные и лишние сегменты.
- Формула приведена для одного класса; итог обычно усредняется по классам.
- Пустые классы и void-области обрабатываются по протоколу датасета.

http://arxiv.org/pdf/1801.00868

FCN: слияние предсказаний разных масштабов

Пусть F_s — признаки с шагом s, $S_s = \mathrm{Conv}_{1\times1}(F_s)$ — логиты классов.

$$
\begin{aligned}
Z_{32} &= \mathrm{Up}_{32}(S_{32}),\\
Z_{16} &= \mathrm{Up}_{16}\big(S_{16} + \mathrm{Up}_2(S_{32})\big),\\
Z_8 &= \mathrm{Up}_8\big(S_8 + \mathrm{Up}_2(S_{16} + \mathrm{Up}_2(S_{32}))\big).
\end{aligned}
$$

FCN объединяет грубую семантику и более точную локализацию через суммы карт классов. Пространственное выравнивание подразумевается. В U-Net далее будут объединяться *признаки*, обычно конкатенацией.

https://arxiv.org/pdf/1411.4038

FCN

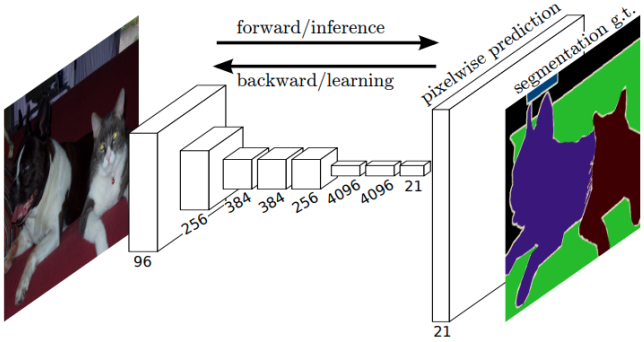


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

https://arxiv.org/pdf/1411.4038

FCN

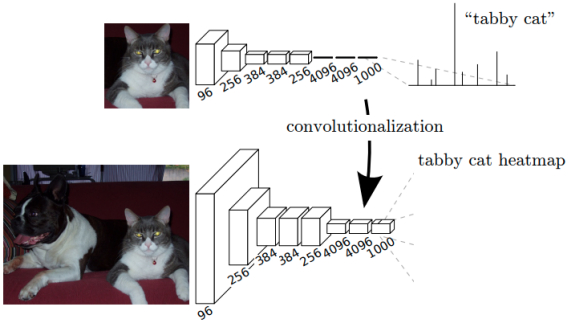


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

https://arxiv.org/pdf/1411.4038

FCN

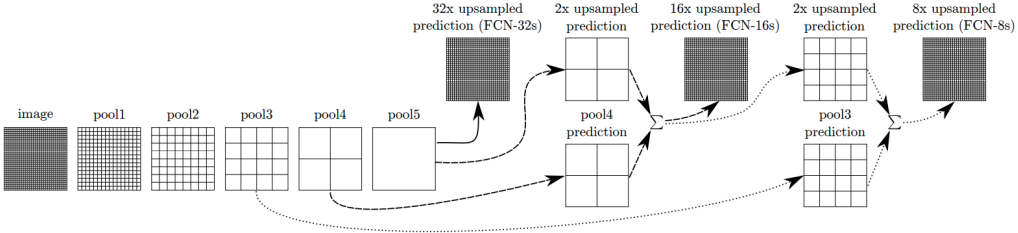


Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Layers are shown as grids that reveal relative spatial coarseness. Only pooling and prediction layers are shown; intermediate convolution layers (including our converted fully connected layers) are omitted. Solid line (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Dashed line (FCN-16s): Combining predictions from both the final layer and the `pool4` layer, at stride 16, lets our net predict finer details, while retaining high-level semantic information. Dotted line (FCN-8s): Additional predictions from `pool3`, at stride 8, provide further precision.

https://arxiv.org/pdf/1411.4038

U-Net: декодирование с доступом к признакам

Для уровня ℓ обозначим признаки encoder через E_ℓ, decoder через D_ℓ:

$$E_{\ell+1} = \mathcal{E}_\ell(\mathrm{Pool}(E_\ell)), \qquad D_L = E_L,$$

$$D_\ell = \mathcal{D}_\ell(\mathrm{Concat}[E_\ell, \mathrm{Up}_2(D_{\ell+1})]), \qquad Z = \mathrm{Conv}_{1\times1}(D_0).$$

- Encoder задаёт контекст, skip-путь сохраняет локальные признаки.
- Декодер обучается совместно использовать обе группы признаков.
- В оригинальной U-Net: valid-свёртки и crop skip-карт. В вариантах с padding размеры могут сохраняться.

https://arxiv.org/abs/1505.04597

U-Net

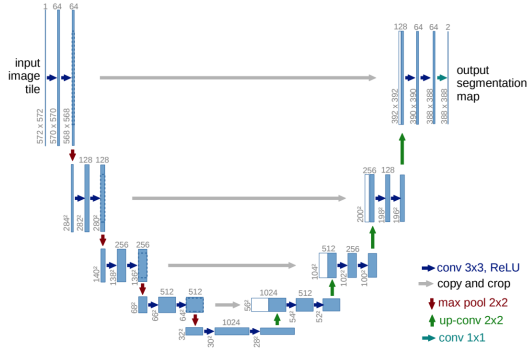


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

https://arxiv.org/abs/1505.04597

U-Net

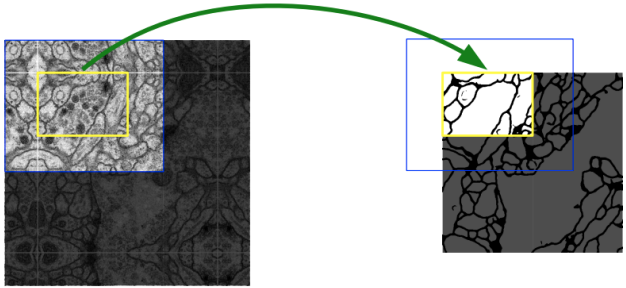


Fig. 2. Overlap-tile strategy for seamless segmentation of arbitrary large images (here segmentation of neuronal structures in EM stacks). Prediction of the segmentation in the yellow area, requires image data within the blue area as input. Missing input data is extrapolated by mirroring

https://arxiv.org/abs/1505.04597

U-Net

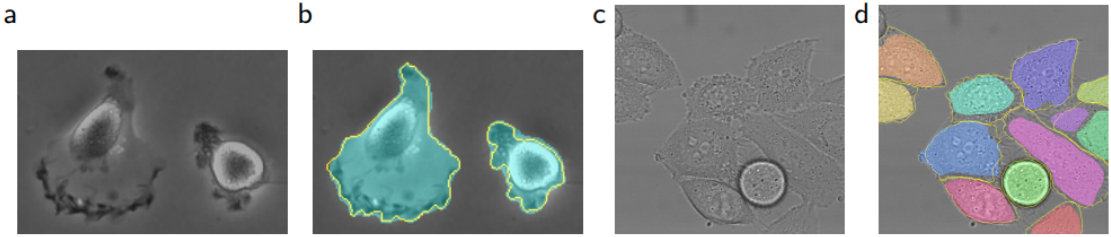


Fig. 4. Result on the ISBI cell tracking challenge. (**a**) part of an input image of the "PhC-U373" data set. (**b**) Segmentation result (cyan mask) with manual ground truth (yellow border) (**c**) input image of the "DIC-HeLa" data set. (**d**) Segmentation result (random colored masks) with manual ground truth (yellow border).

https://arxiv.org/abs/1505.04597

SegNet: передача индексов вместо skip-карт

Для каждого pooling-окна $\mathcal{W}_i$ и канала:

$$v_i = \max_{u \in \mathcal{W}_i} x_u, \qquad q_i = \arg\max_{u \in \mathcal{W}_i} x_u.$$

В декодере значения d_i размещаются по сохранённым индексам:

$$\widetilde{d}_u = \begin{cases} d_i, & u = q_i, \\ 0, & \text{иначе}, \end{cases} \qquad D_\ell = \mathcal{D}_\ell(\widetilde{d}).$$

- Unpooling создаёт разреженную карту, свёртки делают её плотной.
- Для skip-информации хранятся позиции максимумов, а не все активации.
- Отброшенные значения не восстанавливаются: pooling необратим.

https://arxiv.org/pdf/1511.00561

SegNet

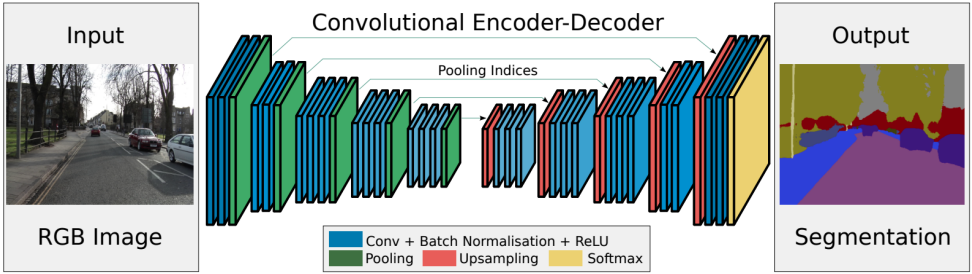
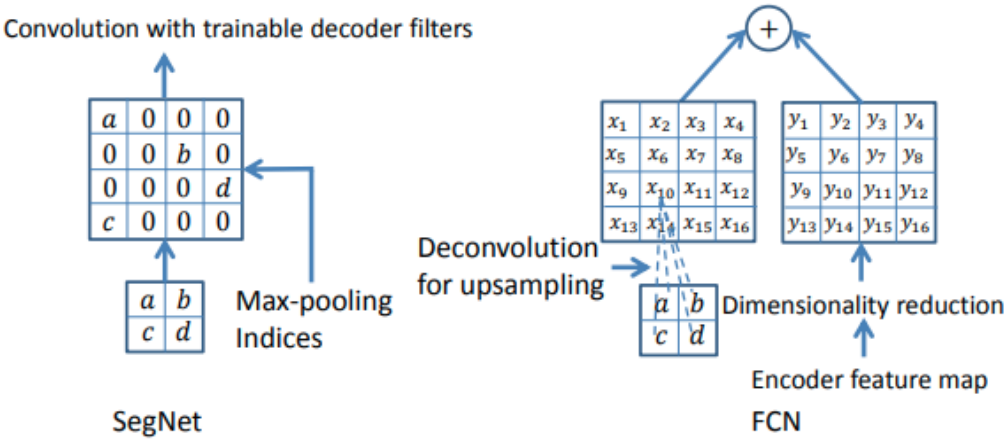


Fig. 2. An illustration of the SegNet architecture. There are no fully connected layers and hence it is only convolutional. A decoder upsamples its input using the transferred pool indices from its encoder to produce a sparse feature map(s). It then performs convolution with a trainable filter bank to densify the feature map. The final decoder output feature maps are fed to a soft-max classifier for pixel-wise classification.

https://arxiv.org/pdf/1511.00561

SegNet



https://arxiv.org/pdf/1511.00561

UNet++: вложенные skip-пути

Пусть $X^{i,j}$ — узел глубины i и стадии уточнения j:

$$X^{i,j} = \begin{cases} \mathcal{H}_i(X^{i-1,0}), & j = 0,\ i > 0, \\ \mathcal{H}_{i,j}\big(\mathrm{Concat}\big[X^{i,0}, \ldots, X^{i,j-1}, \mathrm{Up}_2(X^{i+1,j-1})\big]\big), & j > 0. \end{cases}$$

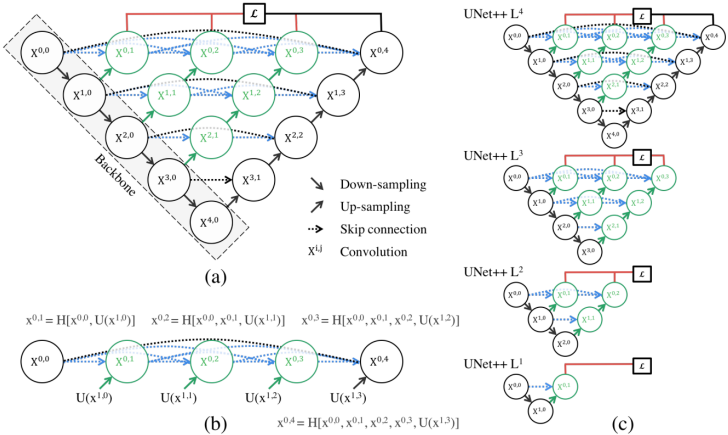
$X^{0,0}$ — начальные признаки; $\mathcal{H}_i$ включает downsampling.

$$\mathcal{L}_{\mathrm{deep}} = \sum_{j=1}^{J} \lambda_j \mathcal{L}\big(h_j(X^{0,j}), Y\big).$$

Промежуточные преобразования сближают признаки encoder и decoder; глубокое наблюдение обучает несколько выходов.

https://arxiv.org/pdf/1807.10165

UNet++

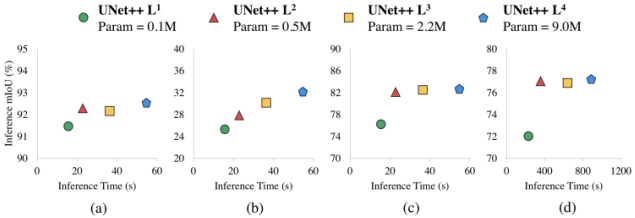


https://arxiv.org/pdf/1807.10165

UNet++

Table 3: Segmentation results (IoU: %) for U-Net, wide U-Net and our suggested architecture UNet++ with and without deep supervision (DS).

Architecture	Params	Dataset			
		cell nuclei	colon polyp	liver	lung nodule
U-Net [9]	7.76M	90.77	30.08	76.62	71.47
Wide U-Net	9.13M	90.92	30.14	76.58	73.38
UNet++ w/o DS	9.04M	**92.63**	**33.45**	79.70	76.44
UNet++ w/ DS	9.04M	92.52	32.12	**82.90**	**77.21**



https://arxiv.org/pdf/1807.10165

Attention U-Net: управление skip-признаками

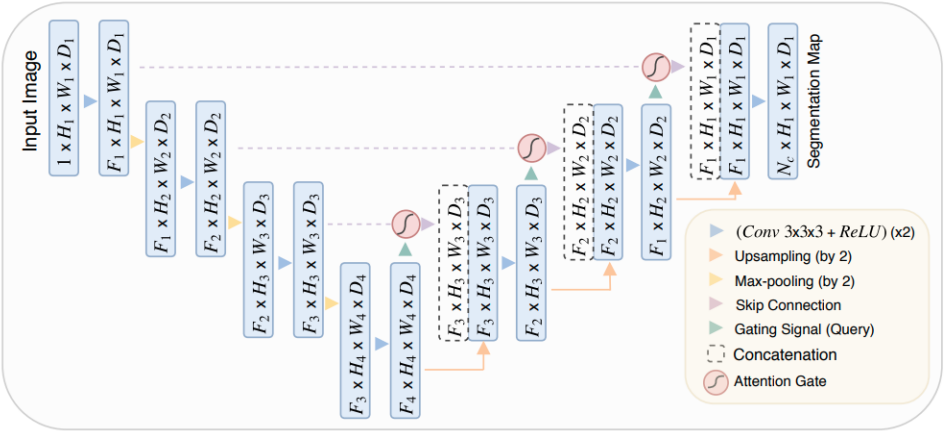
Для позиции u: x_u — skip-признак, g_u — выровненный сигнал из decoder.

$$e_u = \psi^\top \mathrm{ReLU}(W_x x_u + W_g g_u + b) + b_\psi,$$
$$\alpha_u = \mathrm{sigmoid}(e_u), \qquad \widetilde{x}_u = \alpha_u x_u.$$

- Коэффициент $\alpha_u \in (0, 1)$ подавляет нерелевантные области.
- Gating зависит от локальных признаков и контекста decoder.
- Это аддитивный attention gate, без попарной матрицы внимания всех пикселей.

https://arxiv.org/pdf/1804.03999

Attention U-Net



https://arxiv.org/pdf/1804.03999

Attention U-Net

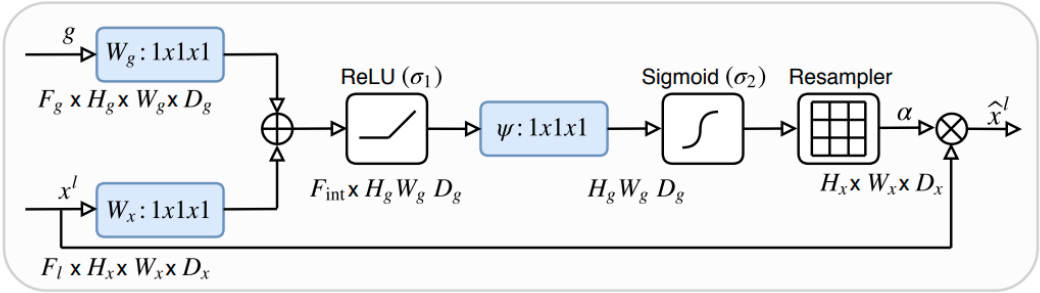


Figure 2: Schematic of the proposed additive attention gate (AG). Input features (x^l) are scaled with attention coefficients (α) computed in AG. Spatial regions are selected by analysing both the activations and contextual information provided by the gating signal (g) which is collected from a coarser scale. Grid resampling of attention coefficients is done using trilinear interpolation.

https://arxiv.org/pdf/1804.03999

Attention U-Net

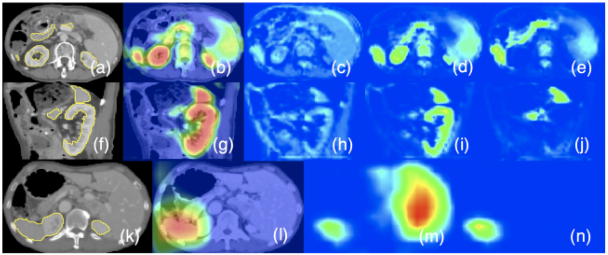


Figure 3(a): From left to right (a-e, f-j): Axial and sagittal views of a 3D abdominal CT scan, attention coefficients, feature activations of a skip connection before and after gating. Similarly, (k-n) visualise the gating on a coarse scale skip connection. The filtered feature activations (d-e, i-j) are collected from multiple AGs, where a subset of organs is selected by each gate. Activations shown in (d-e, i-j) consistently correspond to specific structures across different scans.

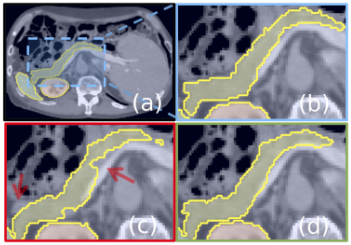


Figure 3(b): The ground-truth pancreas segmentation (a) is highlighted in blue (b). Similarly, U-Net model prediction (c) and the predictions obtained with Attention U-Net (d) are shown. The missed dense predictions by U-Net are highlighted with red arrows.

https://arxiv.org/pdf/1804.03999

DeepLab: развитие идеи контекста

Версия	Основное изменение	Какую проблему решает
v1 (2014/15)	Atrous convolution и dense CRF	Контекст и согласование границ
v2 (2016/17)	ASPP с разными dilation rates	Объекты разных масштабов
v3 (2017)	Развитие ASPP, image pooling, BN	Многомасштабный и глобальный контекст
v3+ (2018)	Decoder, atrous separable convolution	Уточнение границ и стоимость

https://arxiv.org/pdf/1802.02611

Atrous convolution и output stride

Для двумерных индексов u, δ и dilation r:

$$(F *_r W)(u) = \sum_{\delta \in \mathcal{K}} W(\delta) F(u + r\delta), \qquad k_{\text{eff}} = k + (k-1)(r-1).$$

$$\text{OS} = H/H_F = W/W_F, \qquad \text{MACs} \propto \frac{HW}{\text{OS}^2} k^2 C_{\text{in}} C_{\text{out}}.$$

- Убирая stride и повышая dilation, сохраняем более плотные признаки.
- Переход OS 16 $\rightarrow$ 8 даёт примерно 4$\times$ spatial-cost затронутых слоёв.
- Большой dilation разрежает выборку и увеличивает влияние padding.

https://arxiv.org/pdf/1606.00915

DeepLabv1

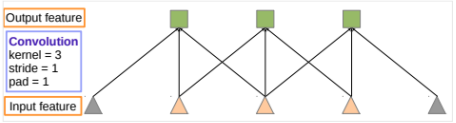


Fig. 2: Illustration of atrous convolution in 1-D. (a) Sparse feature extraction with standard convolution on a low resolution input feature map. (b) Dense feature extraction with atrous convolution with rate $r = 2$, applied on a high resolution input feature map.

https://arxiv.org/pdf/1606.00915

DeepLabv1

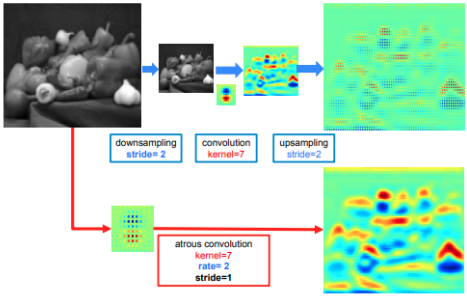


Fig. 3: Illustration of atrous convolution in 2-D. Top row: sparse feature extraction with standard convolution on a low resolution input feature map. Bottom row: Dense feature extraction with atrous convolution with rate $r = 2$, applied on a high resolution input feature map.

https://arxiv.org/pdf/1606.00915

DeepLabv1

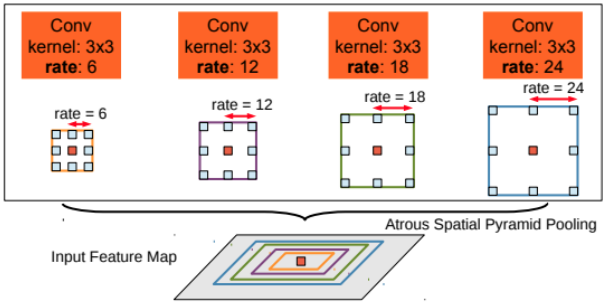


Fig. 4: Atrous Spatial Pyramid Pooling (ASPP). To classify the center pixel (orange), ASPP exploits multi-scale features by employing multiple parallel filters with different rates. The effective Field-Of-Views are shown in different colors.

https://arxiv.org/pdf/1606.00915

DeepLab v1: согласование маски через CRF

$$E(y) = \sum_i \underbrace{-\log p_{i,y_i}}_{\text{unary}} + \sum_{i<j} \underbrace{\mathbf{1}[y_i \neq y_j] \sum_m w_m k_m(f_i, f_j)}_{\text{pairwise}}, \qquad \hat{y} \approx \arg\min_y E(y).$$

Пример bilateral-ядра по координатам s_i и цвету I_i:

$$k(f_i, f_j) = \exp\left(-\frac{\|s_i - s_j\|^2}{2\sigma_s^2} - \frac{\|I_i - I_j\|^2}{2\sigma_I^2} \right).$$

- CNN задаёт предпочтения классов, CRF штрафует несогласованные метки.
- Цветовые границы помогают, но не всегда совпадают с границами объектов.
- В раннем DeepLab CRF — отдельная постобработка с затратами на inference.

https://arxiv.org/pdf/1606.00915

DeepLabv1

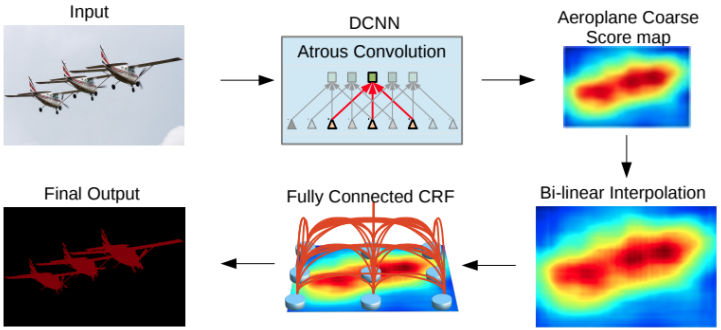


Fig. 1: Model Illustration. A Deep Convolutional Neural Network such as VGG-16 or ResNet-101 is employed in a fully convolutional fashion, using atrous convolution to reduce the degree of signal downsampling (from 32x down 8x). A bilinear interpolation stage enlarges the feature maps to the original image resolution. A fully connected CRF is then applied to refine the segmentation result and better capture the object boundaries.

https://arxiv.org/pdf/1606.00915

DeepLabv3+: лёгкий decoder

Пусть F_{low} имеет шаг 4, а F_{high} — шаг 16:

$$L = \text{Conv}^{48}_{1\times1}(F_{\text{low}}), \qquad A = \text{ASPP}(F_{\text{high}}),$$

$$D = \text{Conv}_{3\times3}(\text{Conv}_{3\times3}(\text{Concat}[L, \text{Up}_4(A)])), \quad Z = \text{Up}_4\Big(\text{Conv}^{K}_{1\times1}(D)\Big).$$

- Проекция low-level признаков до 48 каналов ограничивает их доминирование.
- Decoder уточняет локализацию после агрегации контекста в ASPP.
- Depthwise separable convolution уменьшает вычислительную стоимость.

Separable convolution

При фиксированном пространственном размере и без bias:

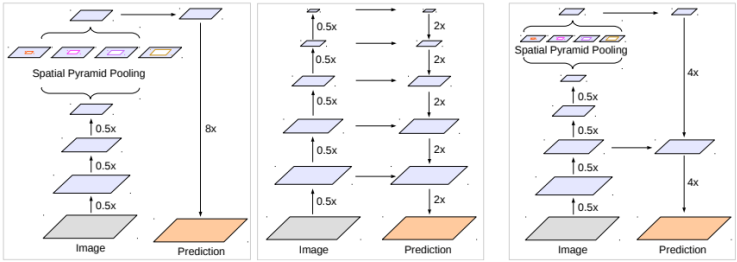
$$P_{\text{standard}} = k^2 C_{\text{in}} C_{\text{out}}, \qquad P_{\text{sep}} = k^2 C_{\text{in}} + C_{\text{in}} C_{\text{out}}.$$

$$\frac{\text{MACs}_{\text{sep}}}{\text{MACs}_{\text{standard}}} = \frac{1}{C_{\text{out}}} + \frac{1}{k^2}.$$

Для $k = 3$, $C_{\text{out}} = 256$: отношение ≈ 0.115.

- Depthwise фильтрует каждый канал отдельно, pointwise смешивает каналы.
- Dilation меняет выборку, но не число весов ядра.
- Уменьшение MACs не гарантирует пропорционального ускорения на GPU.

DeepLabv3



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

Fig. 1. We improve DeepLabv3, which employs the spatial pyramid pooling module (a), with the encoder-decoder structure (b). The proposed model, DeepLabv3+, contains rich semantic information from the encoder module, while the detailed object boundaries are recovered by the simple yet effective decoder module. The encoder module allows us to extract features at an arbitrary resolution by applying atrous convolution.

https://arxiv.org/pdf/1802.02611

DeepLabv3

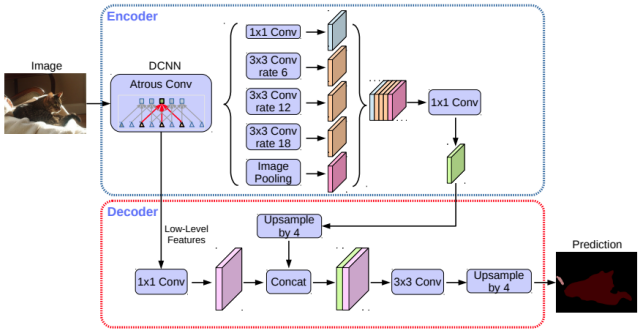


Fig. 2. Our proposed DeepLabv3+ extends DeepLabv3 by employing a encoder-decoder structure. The encoder module encodes multi-scale contextual information by applying atrous convolution at multiple scales, while the simple yet effective decoder module refines the segmentation results along object boundaries.

https://arxiv.org/pdf/1802.02611

HRNet: обмен между разрешениями

Ветвь r хранит $F_r^{(t)} \in \mathbb{R}^{H_r \times W_r \times C_r}$:

$$F_r^{(t+1)} = \mathrm{ReLU}\left(\sum_{s=1}^{R} T_{s \to r}\left(\mathcal{B}_s(F_s^{(t)})\right) \right).$$

$$T_{s \to r} = \begin{cases} \mathrm{Identity}, & s = r, \\ \text{проекция каналов и upsampling}, & H_s < H_r, \\ \text{цепочка stride-2 свёрток}, & H_s > H_r. \end{cases}$$

Высокое разрешение сохраняется на протяжении сети и многократно обогащается контекстом низких разрешений. Высокое разрешение обычно означает шаг 4, а не исходный размер изображения.

HRNetV2: объединение выходов и цена

Для сегментации признаки всех ветвей приводятся к высокому разрешению:

$$F_{\text{head}} = \text{Concat}\left[F_1, \text{Up}(F_2), \ldots, \text{Up}(F_R)\right], \qquad Z = \text{Up}_4\big(h(F_{\text{head}})\big).$$

При $H_r W_r = H_1 W_1 / 4^{r-1}$ и $C_r = 2^{r-1} C_1$:

$$\underbrace{H_r W_r C_r}_{\text{активации}} = \frac{H_1 W_1 C_1}{2^{r-1}}, \quad \underbrace{k^2 H_r W_r C_r^2}_{\text{MACs слоя}} = k^2 H_1 W_1 C_1^2.$$

Падение разрешения компенсирует рост каналов: низкая ветвь не обязательно дешевле. Здесь сравниваются одинаковые dense-свёртки $C_r \to C_r$.

HRNet

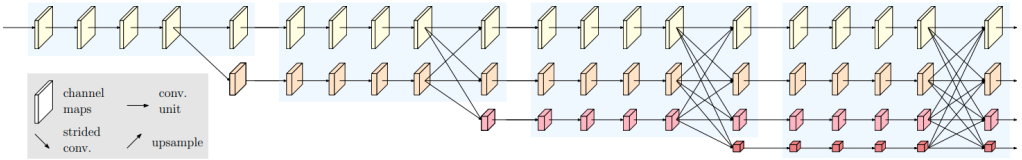


Fig. 2. An example of a high-resolution network. Only the main body is illustrated, and the stem (two stride-2 3×3 convolutions) is not included. There are four stages. The 1st stage consists of high-resolution convolutions. The 2nd (3rd, 4th) stage repeats two-resolution (three-resolution, four-resolution) blocks. The detail is given in Section 3.

https://arxiv.org/pdf/1908.07919

HRNet

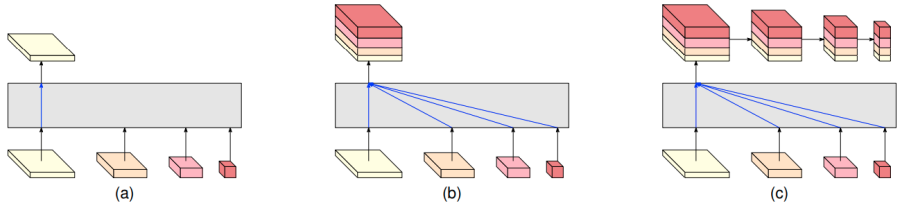


Fig. 4. (a) HRNetV1: only output the representation from the high-resolution convolution stream. (b) HRNetV2: Concatenate the (upsampled) representations that are from all the resolutions (the subsequent 1×1 convolution is not shown for clarity). (c) HRNetV2p: form a feature pyramid from the representation by HRNetV2. The four-resolution representations at the bottom in each sub-figure are outputted from the network in Figure 2, and the gray box indicates how the output representation is obtained from the input four-resolution representations.

https://arxiv.org/pdf/1908.07919

Mask R-CNN: маска внутри найденного объекта

$$F = \mathrm{Backbone}(X), \quad \{b_n\} = \mathrm{RPN}(F), \quad R_n = \mathrm{RoIAlign}(F, b_n).$$

$$R_n \longmapsto \left(p_n^{\mathrm{class}}, \Delta b_n, P_n^{\mathrm{mask}} \in [0,1]^{m \times m \times K}\right).$$

$$\mathcal{L}_{\mathrm{RoI}} = \mathcal{L}_{\mathrm{cls}} + \mathcal{L}_{\mathrm{box}} + \mathcal{L}_{\mathrm{mask}}.$$

- Маски разных классов имеют независимые sigmoid-выходы.
- Для положительного RoI mask-loss берут только по истинному классу.
- Полная функция обучения дополнительно содержит потери RPN.

https://arxiv.org/pdf/1703.06870

RoIAlign и функция потерь маски

Дробная позиция (x, y) интерполируется без округления координат RoI:

$$F(x,y) = \sum_{i \in \{\lfloor x \rfloor, \lfloor x \rfloor + 1\}} \sum_{j \in \{\lfloor y \rfloor, \lfloor y \rfloor + 1\}} F_{ij}(1 - |x - i|)(1 - |y - j|).$$

Для истинного класса c и целевой маски $T \in \{0,1\}^{m \times m}$:

$$\mathcal{L}_{\text{mask}} = -\frac{1}{m^2} \sum_{u} \left[T_u \log P_{u,c} + (1 - T_u) \log(1 - P_{u,c}) \right].$$

Сохранение пространственного выравнивания особенно важно для предсказания маски, а не только класса RoI.

https://arxiv.org/pdf/1703.06870

Mask R-CNN

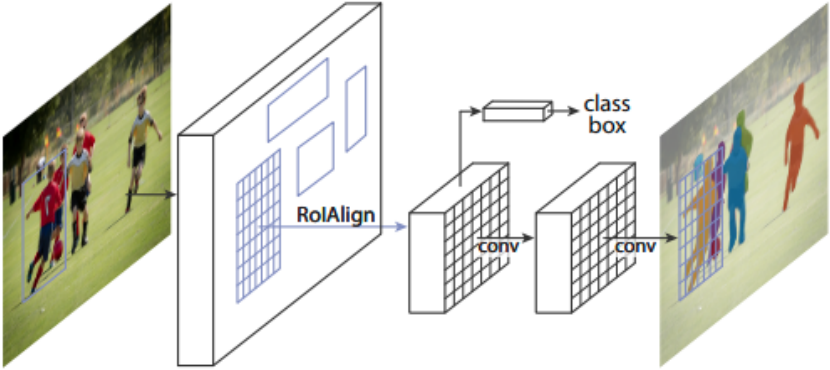


Figure 1. The **Mask R-CNN** framework for instance segmentation.

https://arxiv.org/pdf/1703.06870

Какие идеи изменили архитектуры?

Семейство	Механизм	Основной компромисс
FCN, U-Net	Слияние уровней	Точность / память skip-карт
SegNet	Индексы pooling	Компактность / потеря значений
DeepLab, PSPNet	Многомасштабный контекст	Контекст / детали / стоимость
HRNet	Постоянная high-res ветвь	Локализация / вычисления
ENet, BiSeNet	Экономные блоки, две ветви	Latency / качество
Mask R-CNN	Маска каждого RoI	Экземпляры / зависимость от детекции
SAM	Условная маска по prompt	Гибкость / необходимость подсказки