Segmentation

План лекции

- Classification recap
- Problem statement
- Main architecture
- Metrics
- Loss functions
- Extra

Классификация. Кросс-энтропия

Правдоподобие позволяет понять, насколько вероятно получить данные значения таргета y при данных X и весах w. Оно имеет вид

$$p(y \mid X, w) = \prod_i p(y_i \mid x_i, w)$$

и для распределения Бернулли его можно выписать следующим образом:

$$p(y \mid X, w) = \prod_i p_i^{y_i} (1 - p_i)^{1 - y_i}$$

где p_i – это вероятность, посчитанная из ответов модели. Оптимизировать произведение неудобно, хочется иметь дело с суммой, так что мы перейдём к логарифмическому правдоподобию и подставим формулу для вероятности, которую мы получили выше:

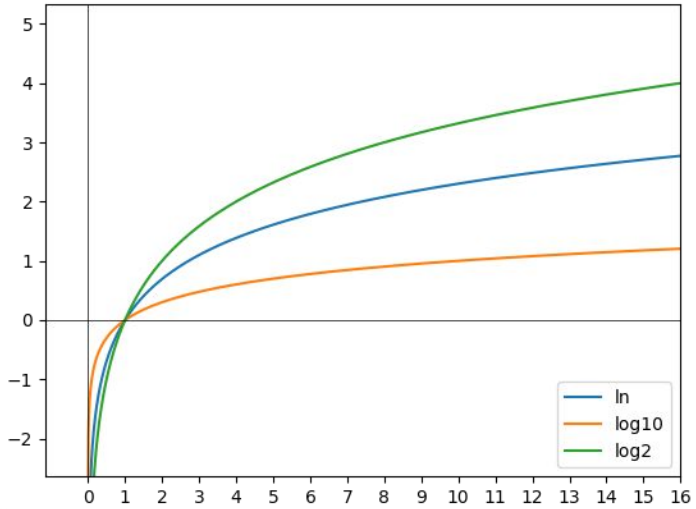
$$\ell(w, X, y) = \sum_i \big(y_i \log(p_i) + (1 - y_i) \log(1 - p_i)\big) =$$

$$= \sum_i \big(y_i \log(\sigma(\langle w, x_i \rangle)) + (1 - y_i) \log(1 - \sigma(\langle w, x_i \rangle))\big)$$

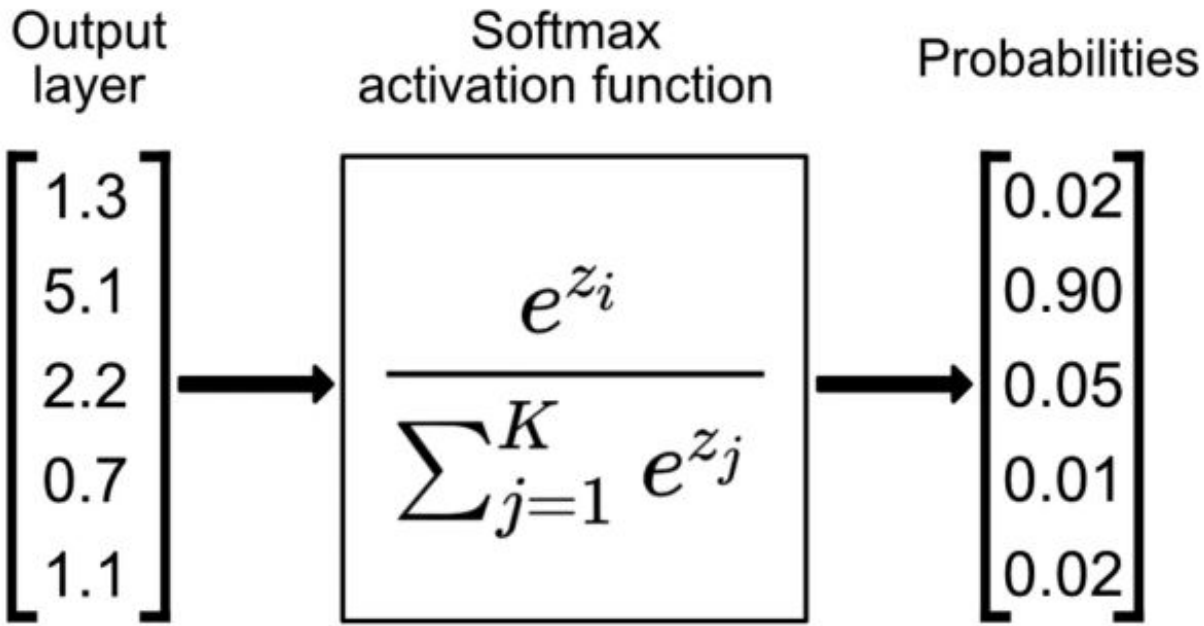
link

Классификация. Кросс-энтропия

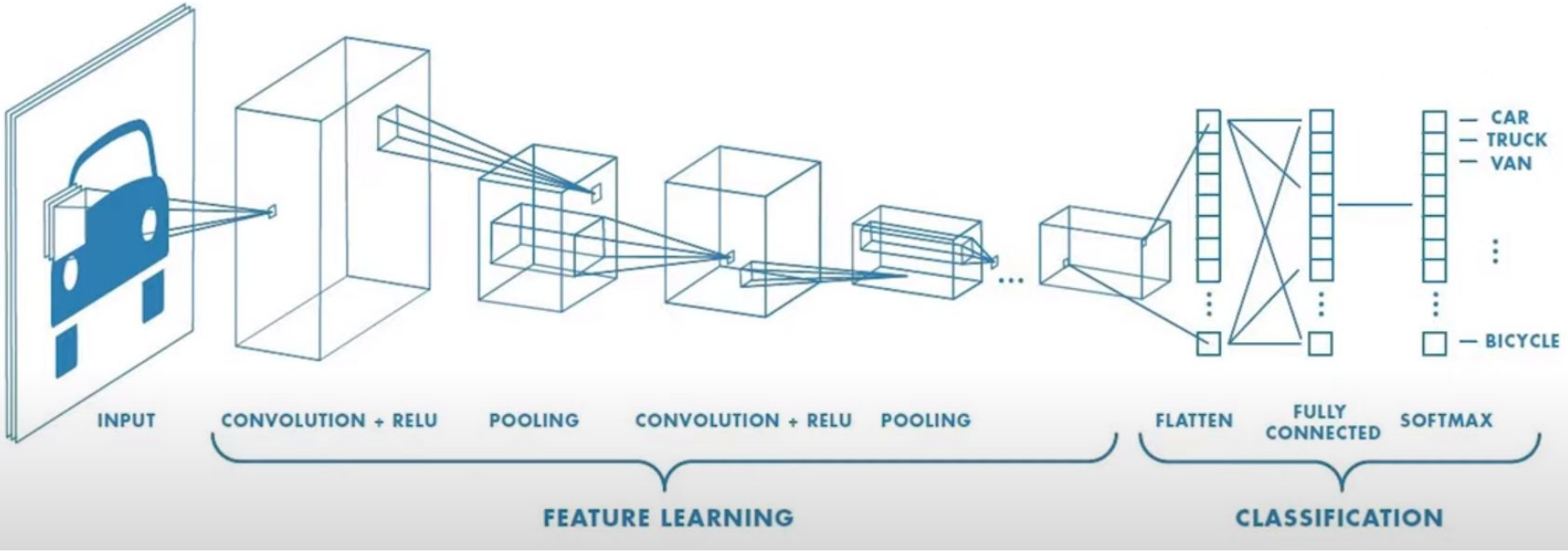
$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \qquad CE = -\sum_i^C t_i log(f(s)_i)$$



Классификация. Много классов

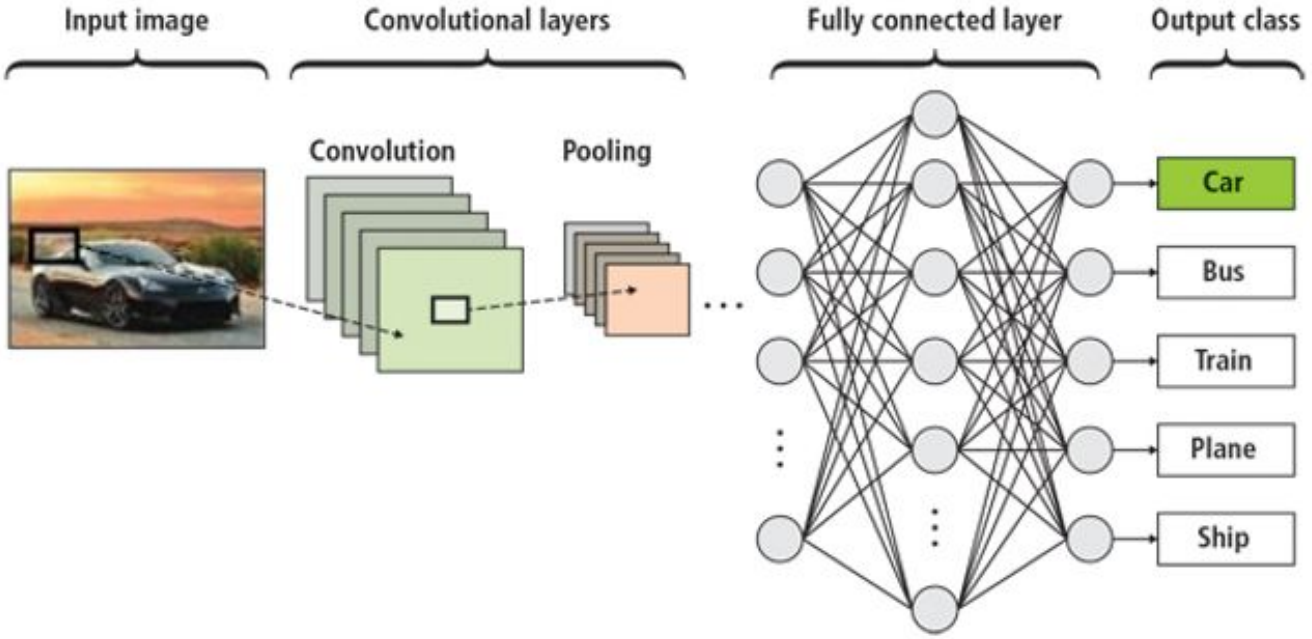


Классификация



Какие из слоев незнакомы?

Классификация



link

Что тут изображено?

link

Что тут изображено?

Дорога

Машины

Дома

Знаки

link

Где это на картинке?

Дорога

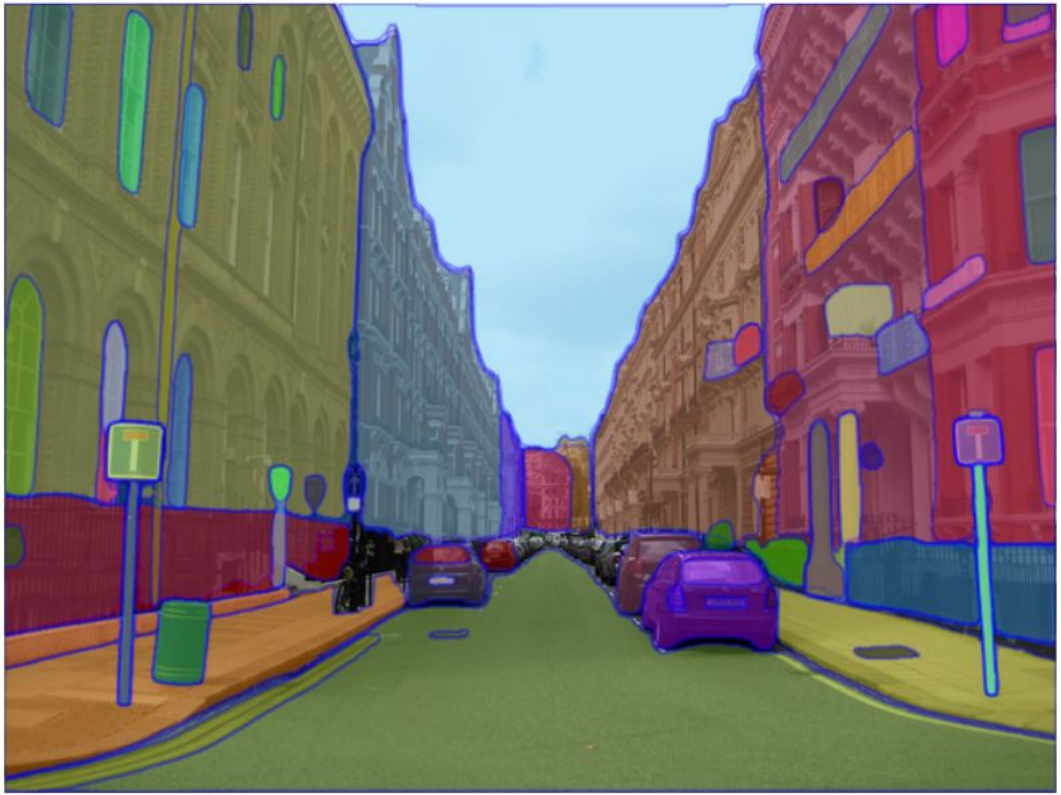
Машины

Дома

Знаки

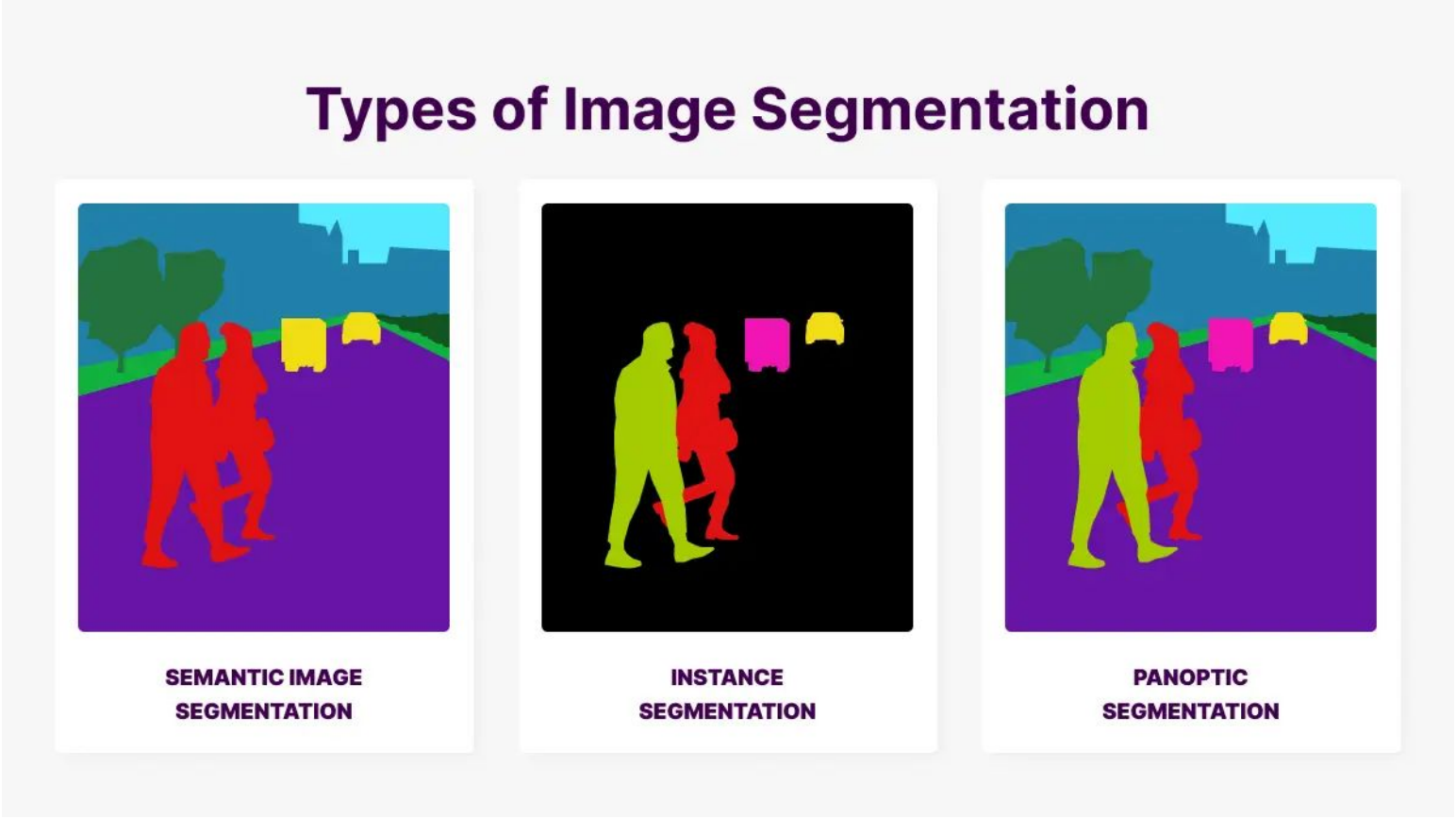
link

Что нам на самом деле нужно?



link

Сегментация



link

Семантическая сегментация. Первая идея?

link

Семантическая сегментация. Первая идея?

Попиксельная классификация. Потенциальная проблема?

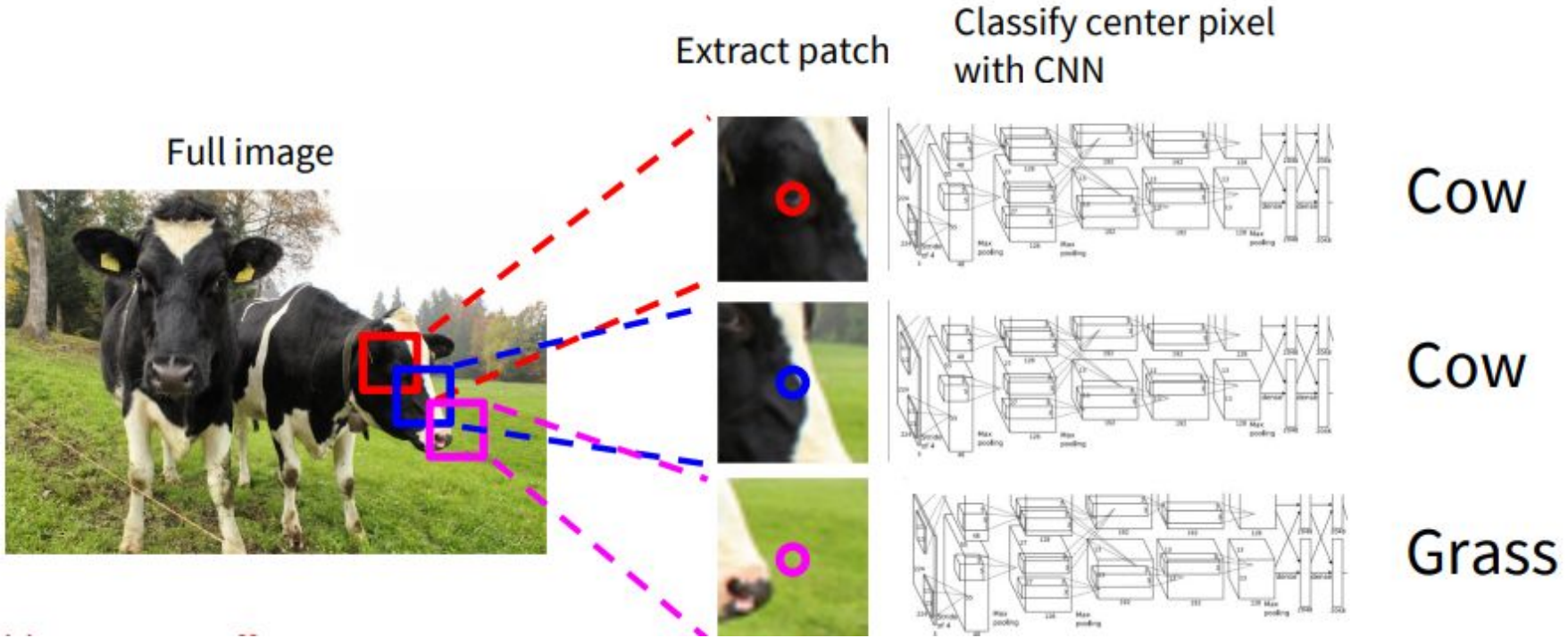
link

Семантическая сегментация. Первая идея?

Не хватает контекста, чтобы ответить на вопрос, что изображено в этом пикселе

link

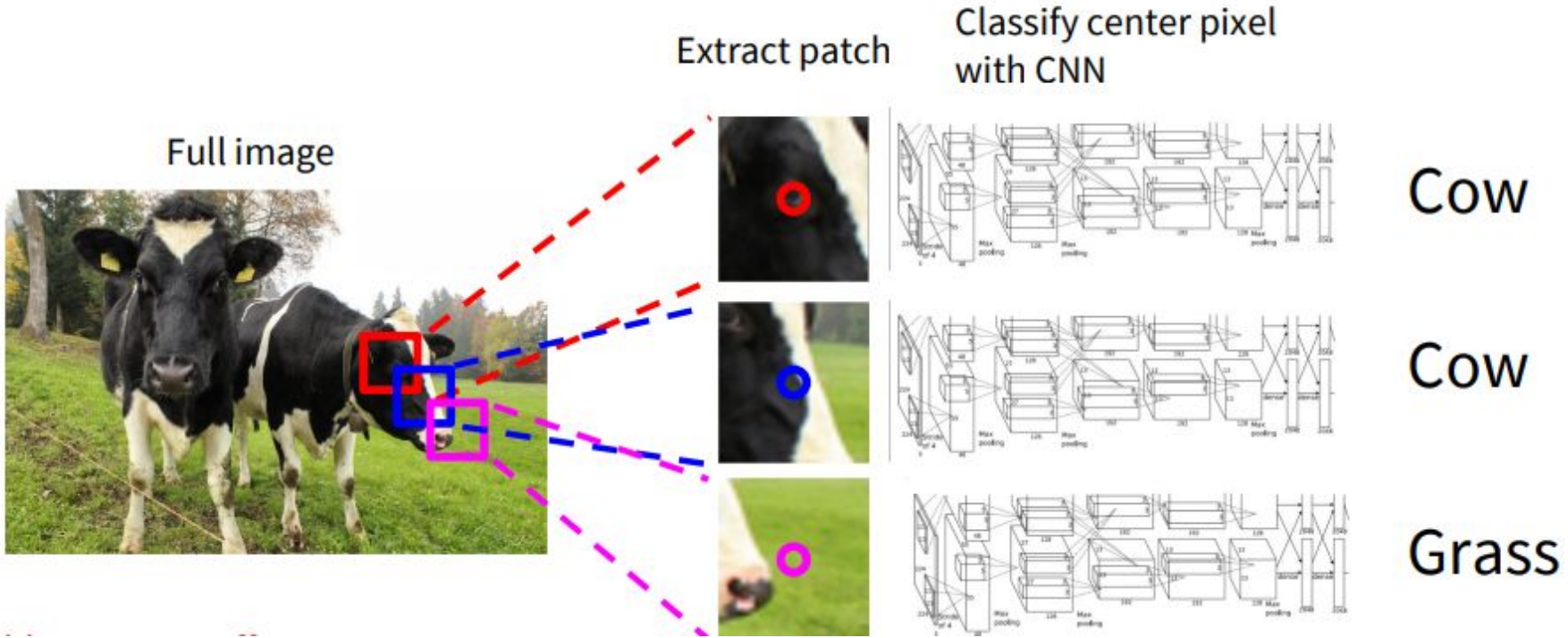
Семантическая сегментация. Первая идея?



Проблема?

link

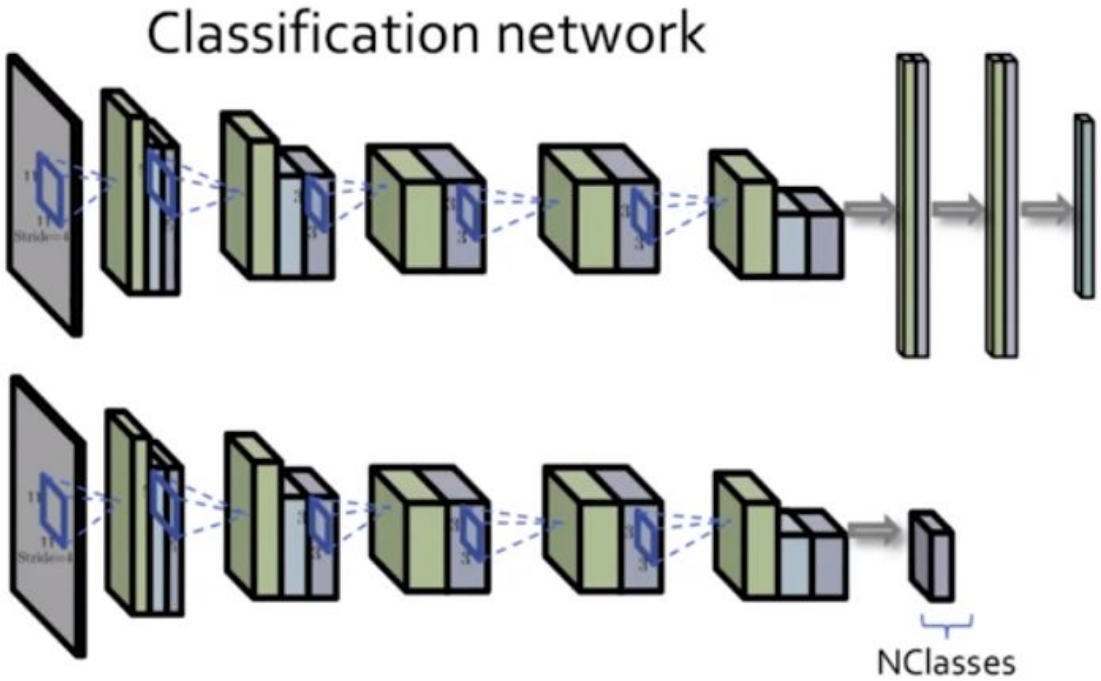
Семантическая сегментация. Первая идея?



Крайне неэффективно.

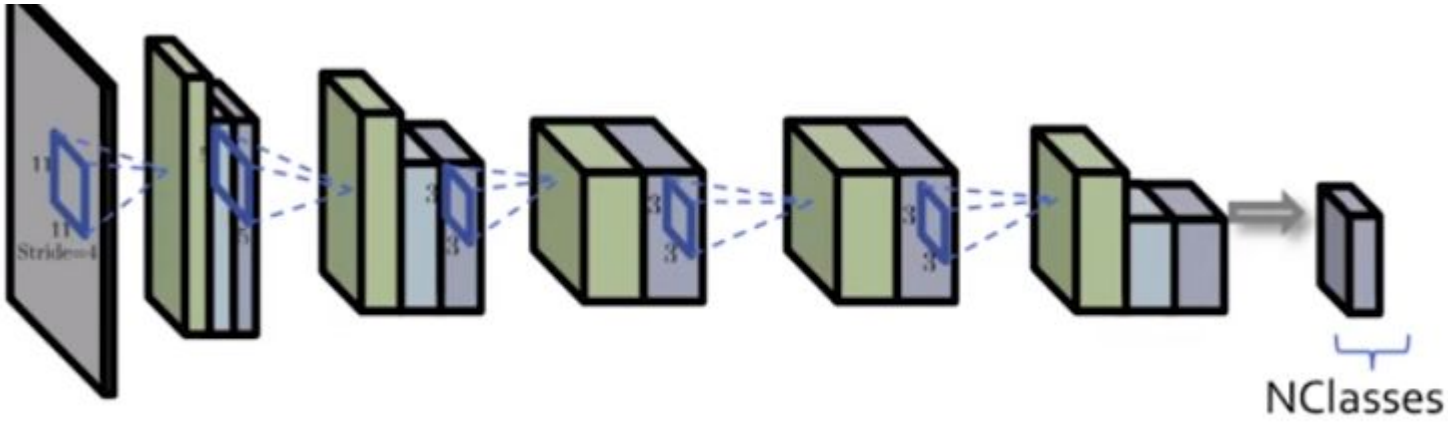
link

Другая идея:



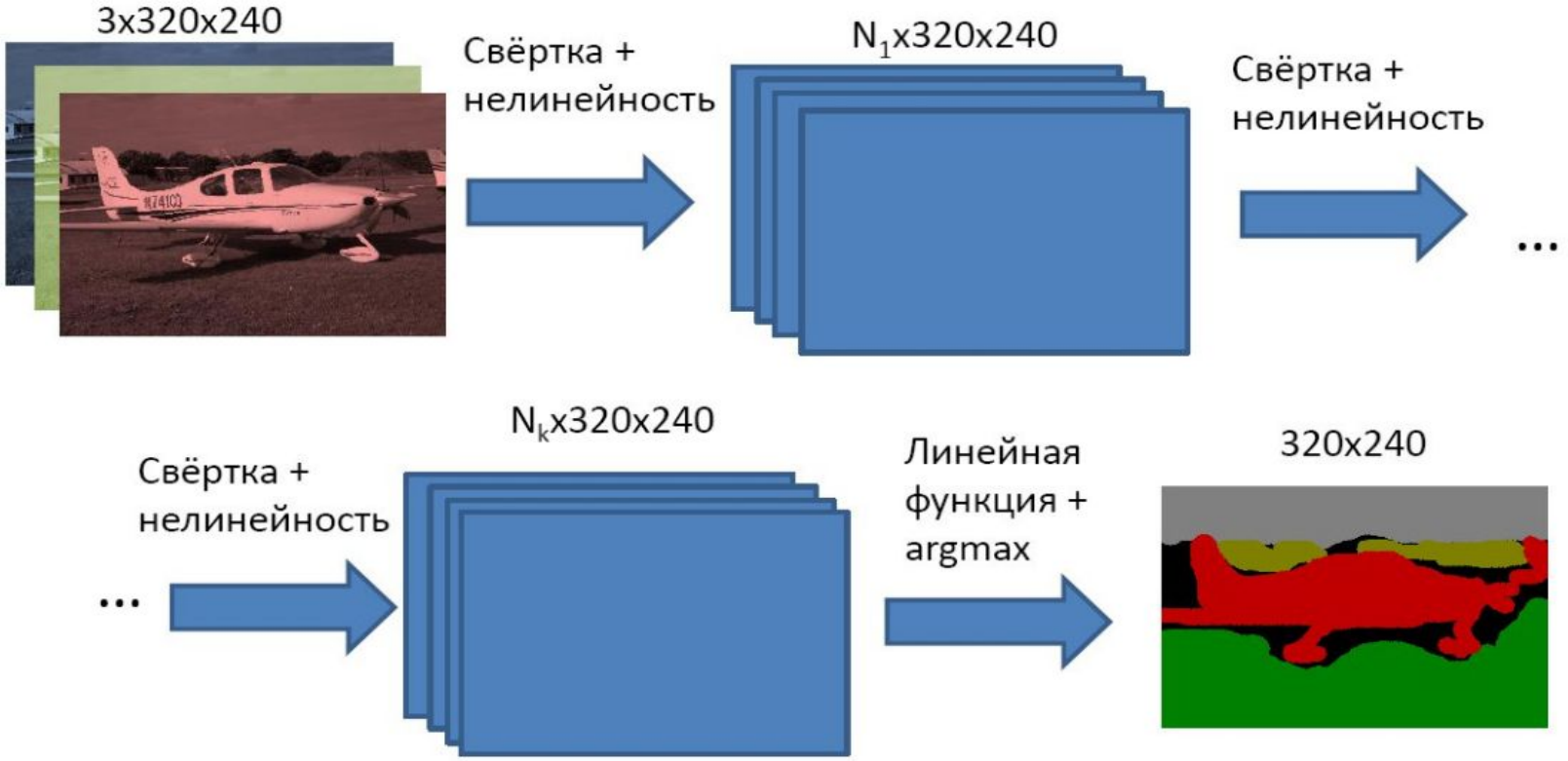
Чем плох такой подход?

Другая идея:

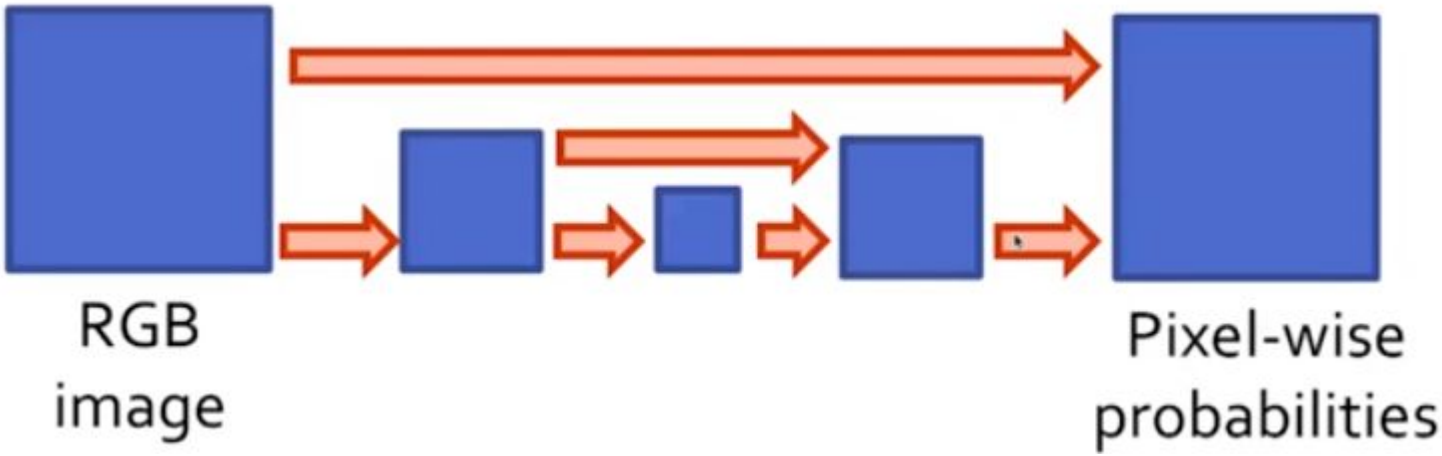


Нужно, чтобы выход был размера входа.
Если убрать страйды, мы уменьшим receptive field.
Огромные по размеру свертки будут неэффективны.

Deep convolutional networks for scene parsing. 2009



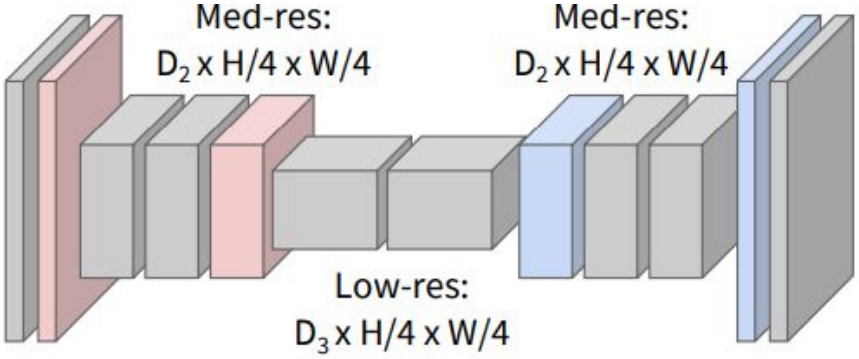
Главная идея:



Понижая пространственную размерность, мы можем учить больше фичей
Благодаря skip connection’ам при декодировании учитываются детали.

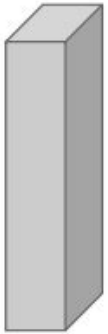
Главная идея:

Input:
3 x H x W



High-res:
D_1 x H/2 x W/2

High-res:
D_1 x H/2 x W/2

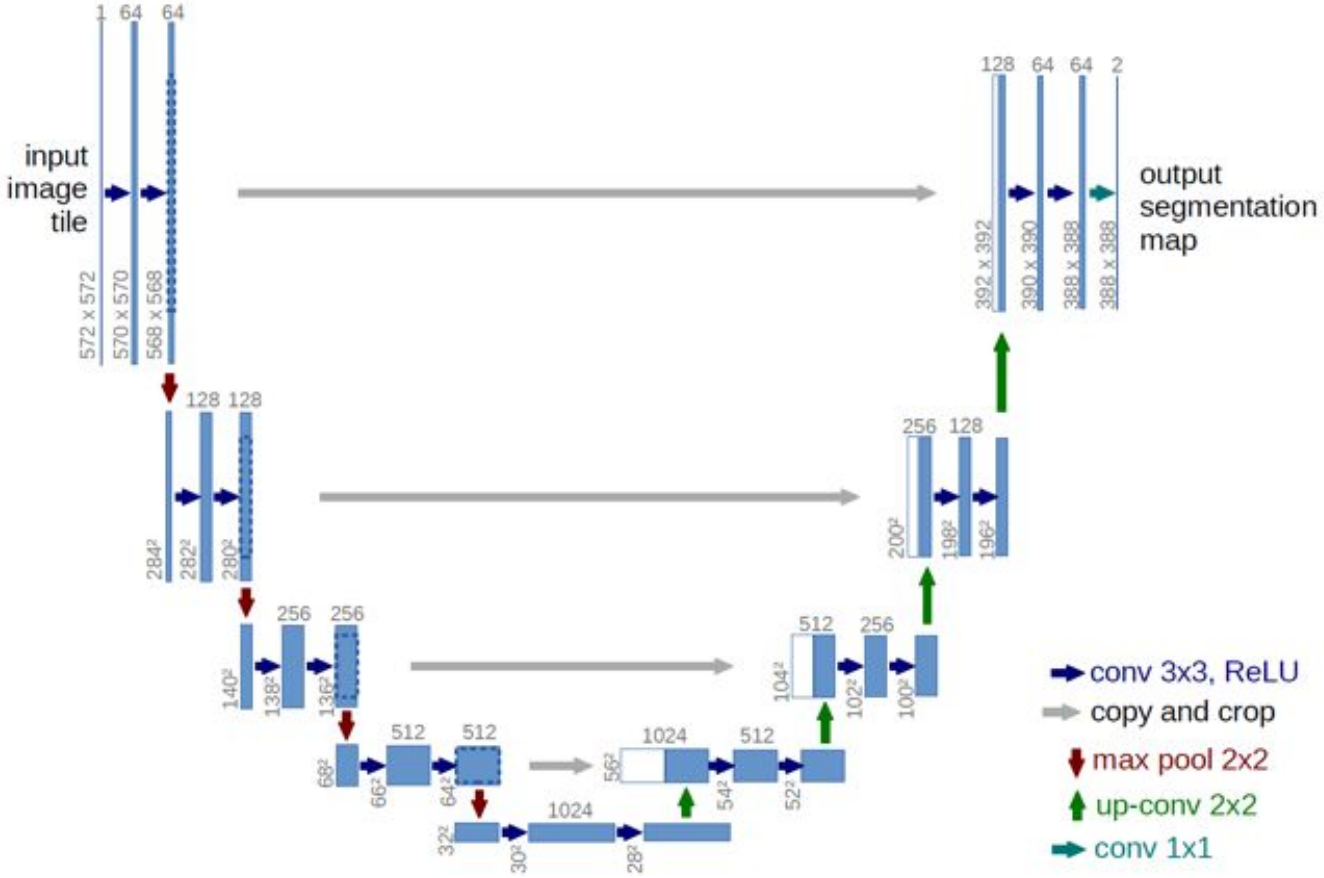


C x H x W

Predictions:
H x W

link

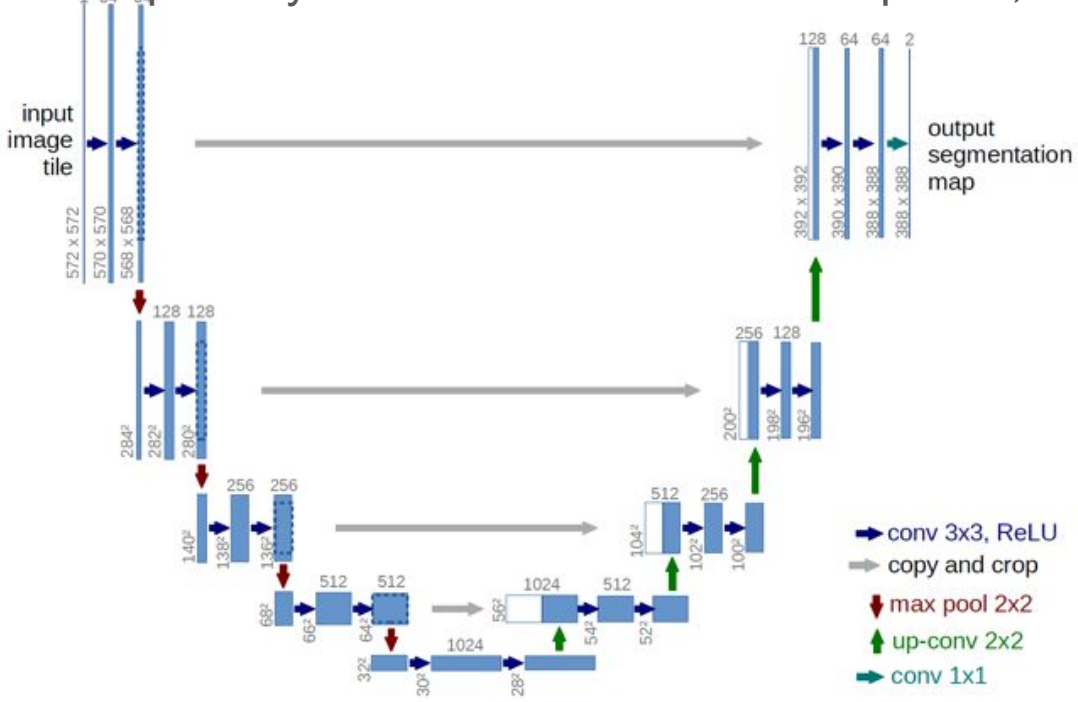
Основная архитектура. Unet, 2015:



link

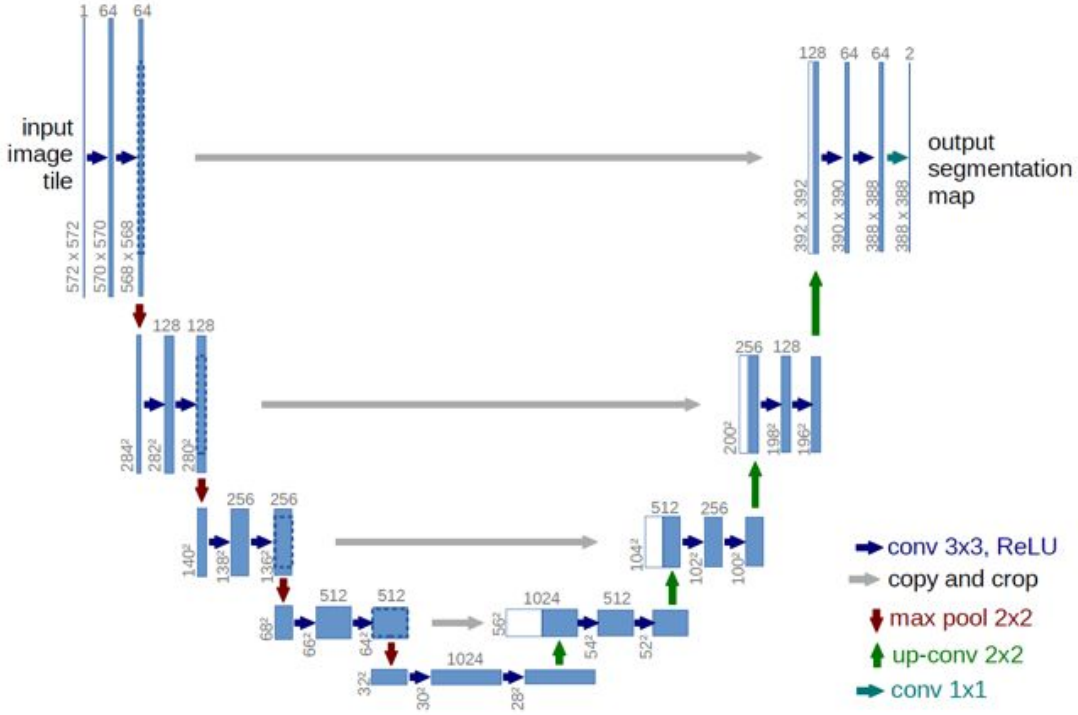
Основная архитектура. Unet, 2015:

Зачем нам понижать пространственную размерность? Наверняка будет работать лучше, если просто увеличивать число слоев/фичей, не теряя в разрешении.



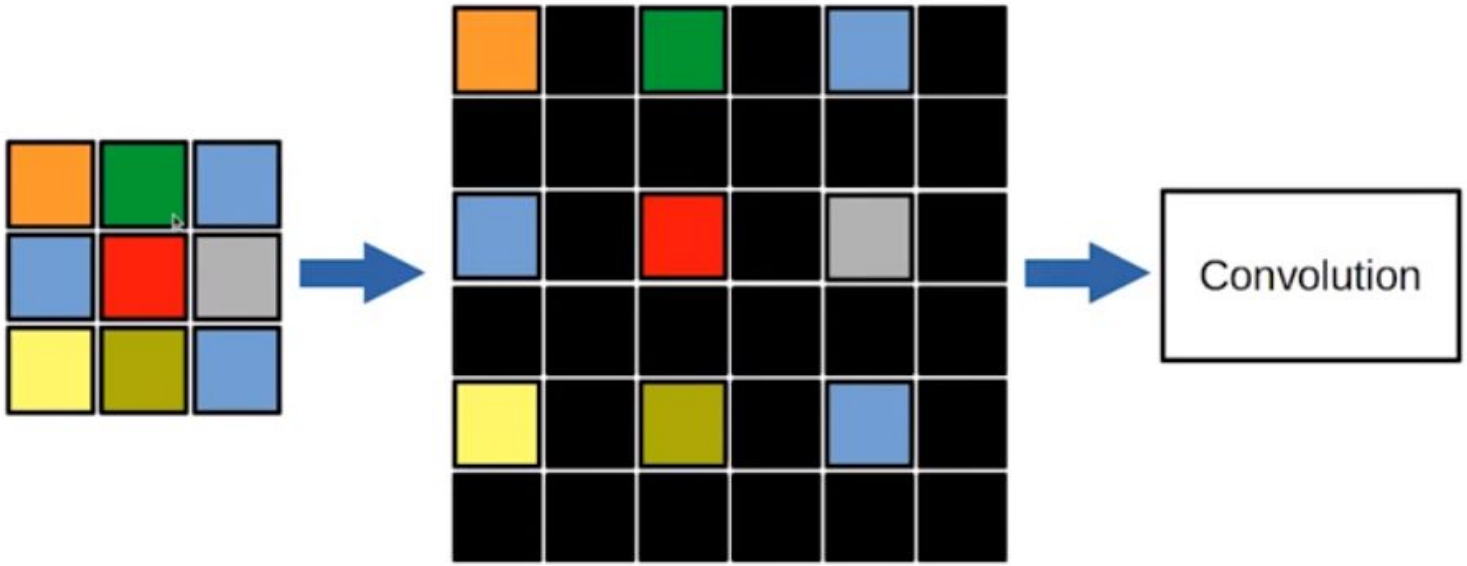
Основная архитектура. Unet, 2015:

Это очень вычислительно затратно + повышаем шанс переобучиться



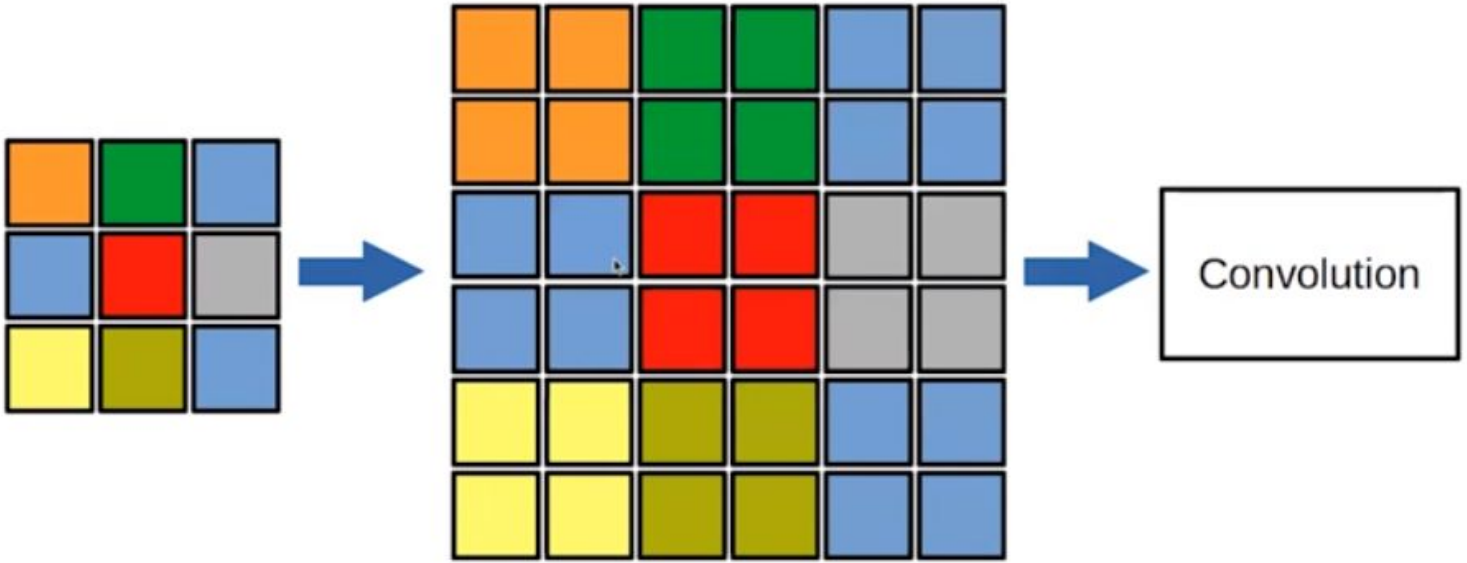
Upsampling

- “Bed of nails” upsampling

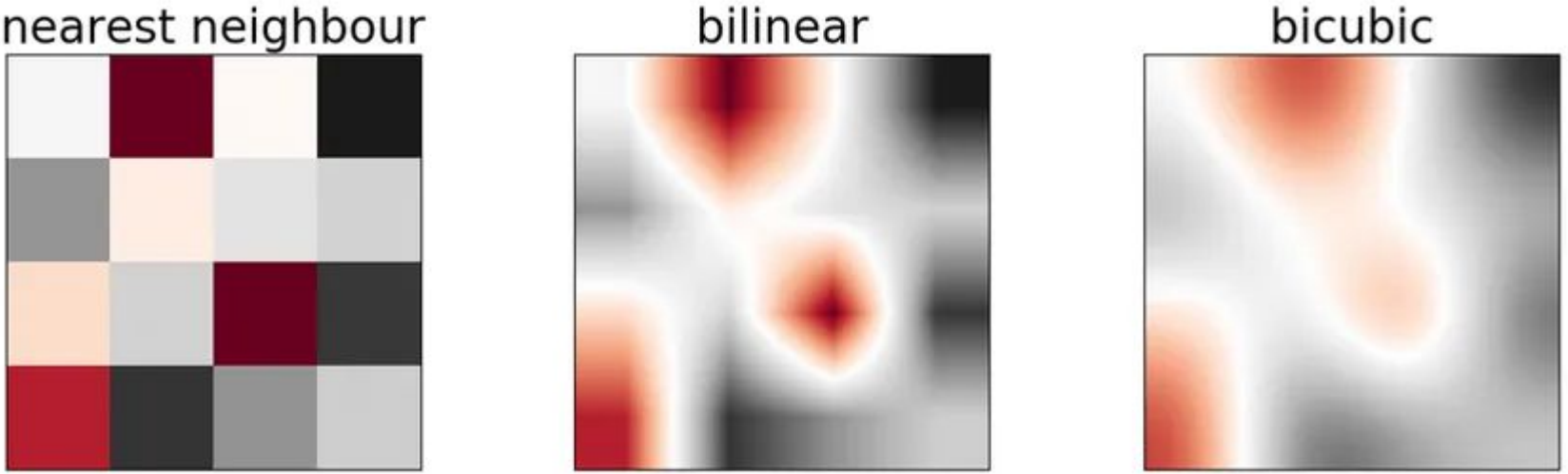


Upsampling

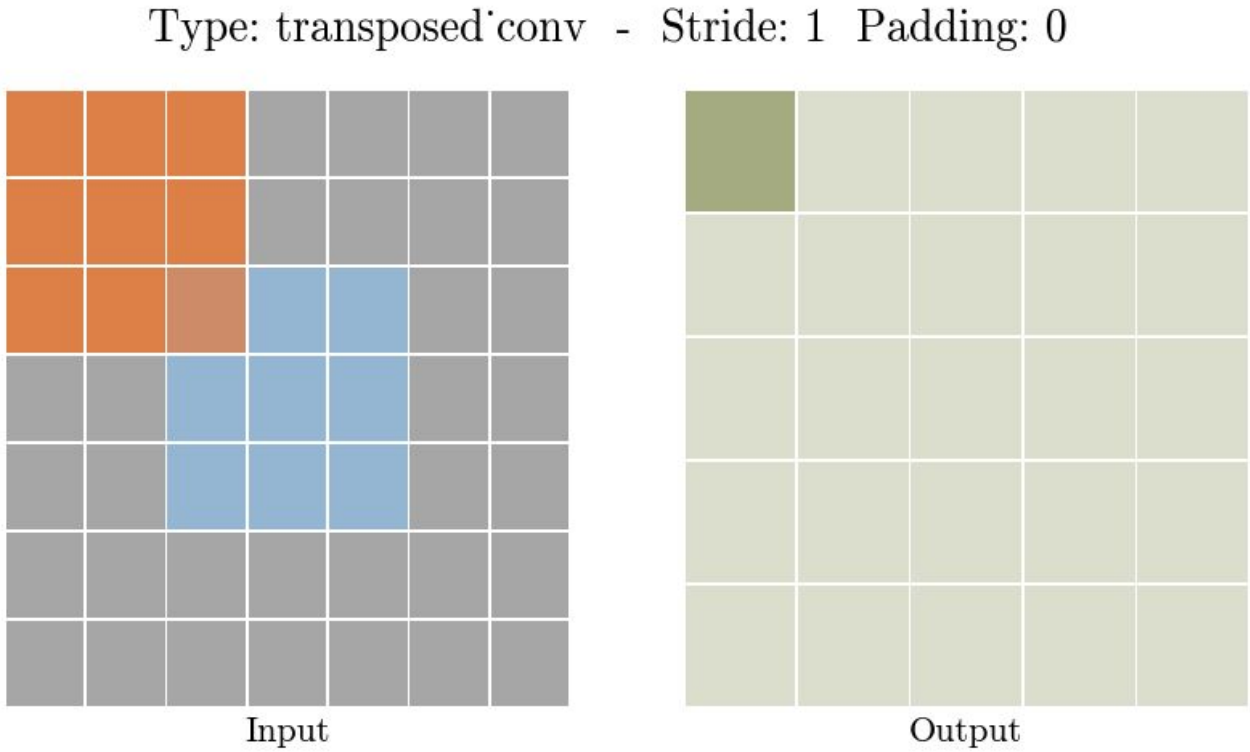
- Nearest neighbour upsampling



Bilinear

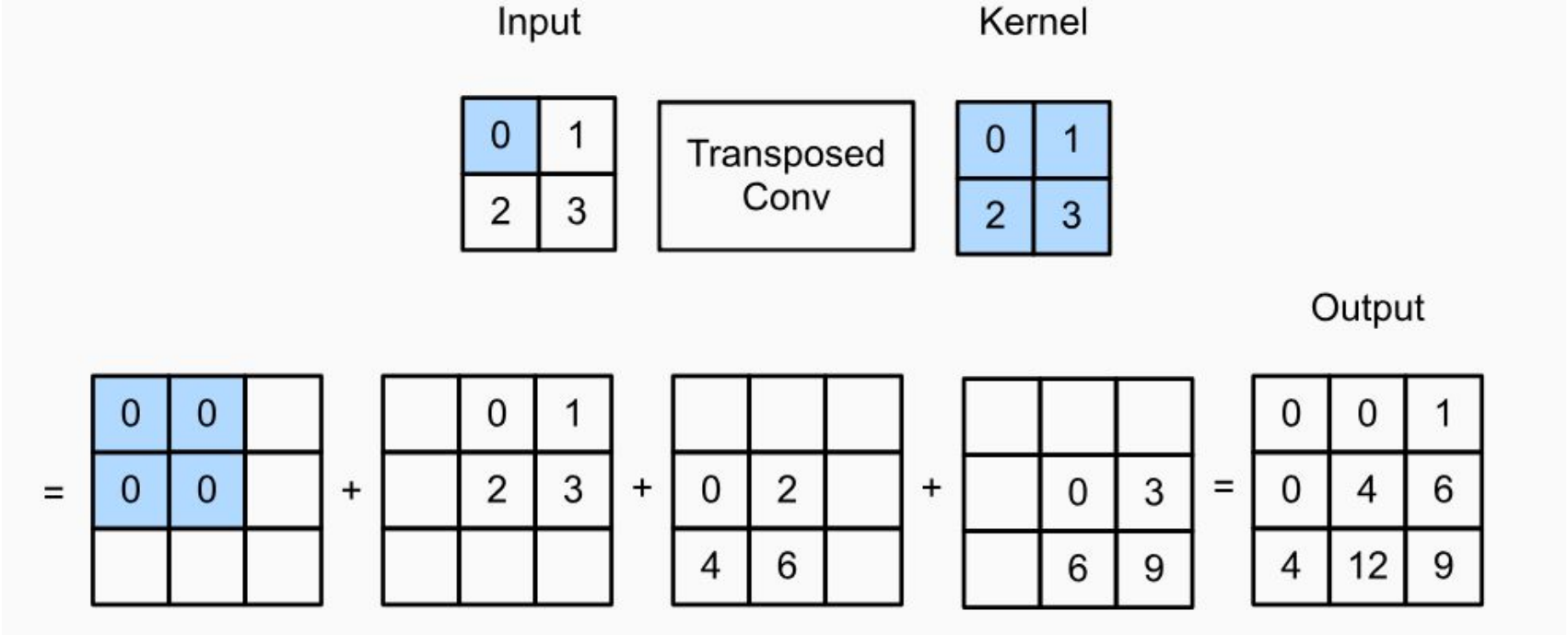


Transposed convolution



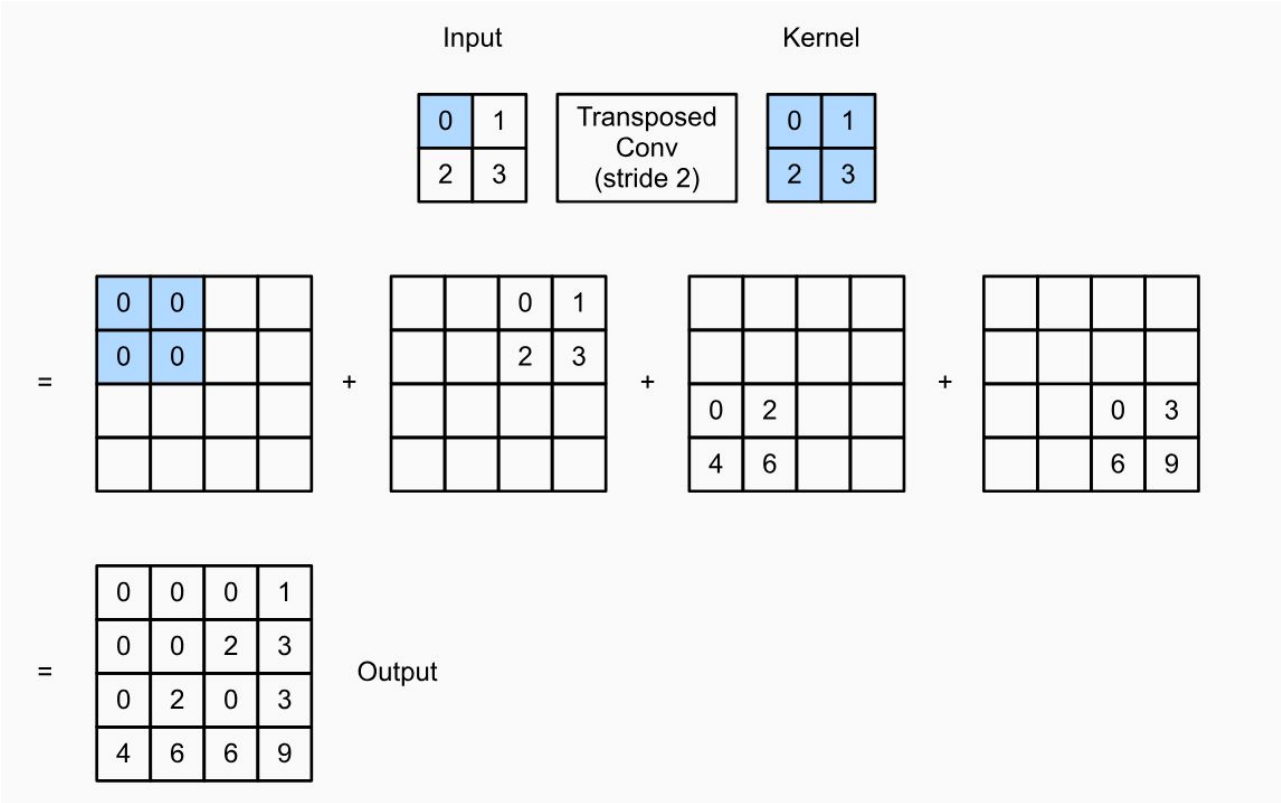
link

Transposed convolution



link

Transposed convolution

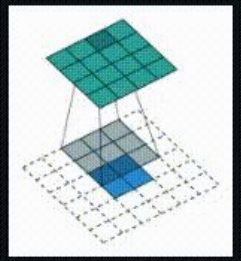


link

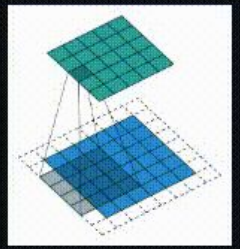
Transposed convolution

Transposed convolution animations

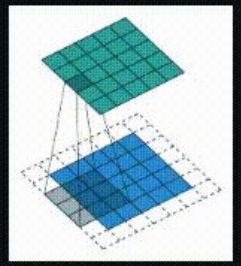
N.B.: Blue maps are inputs, and cyan maps are outputs.



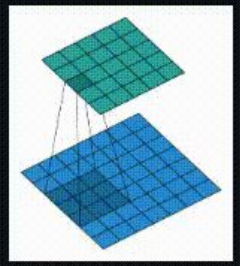
No padding, no strides, transposed



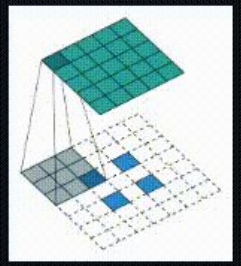
Arbitrary padding, no strides, transposed



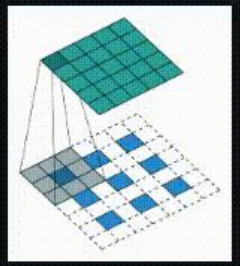
Half padding, no strides, transposed



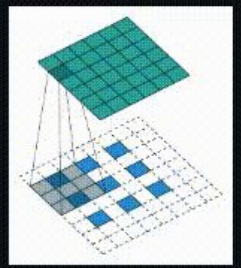
Full padding, no strides, transposed



No padding, strides, transposed



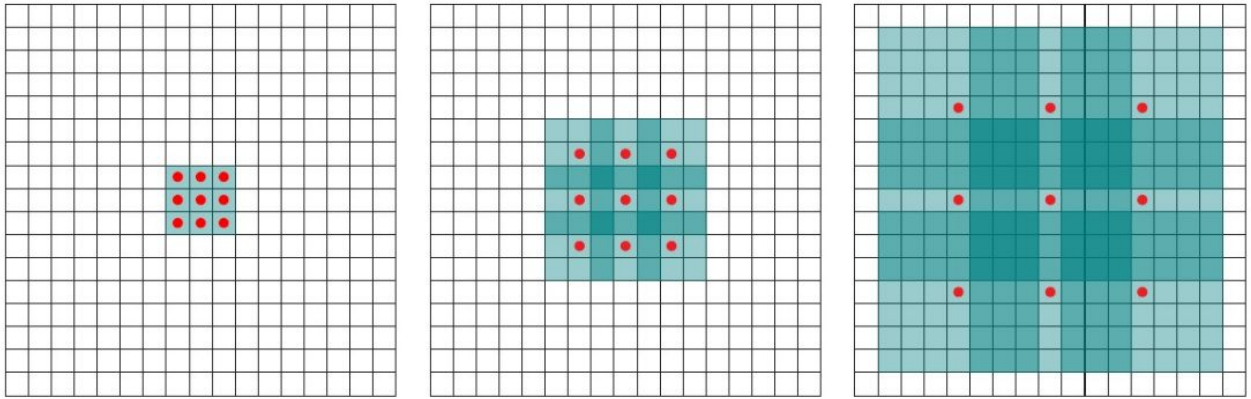
Padding, strides, transposed



Padding, strides, transposed (odd)

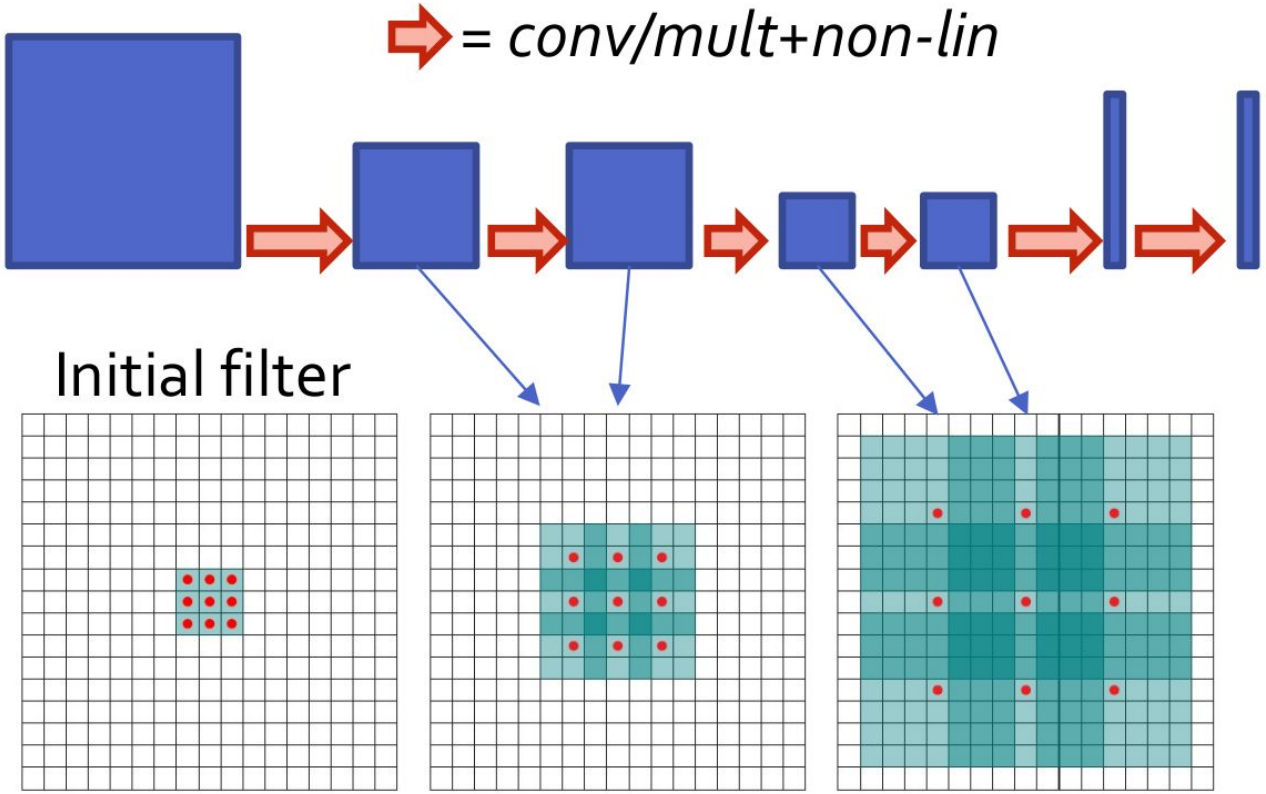
link

Dilated convolutions, 2016

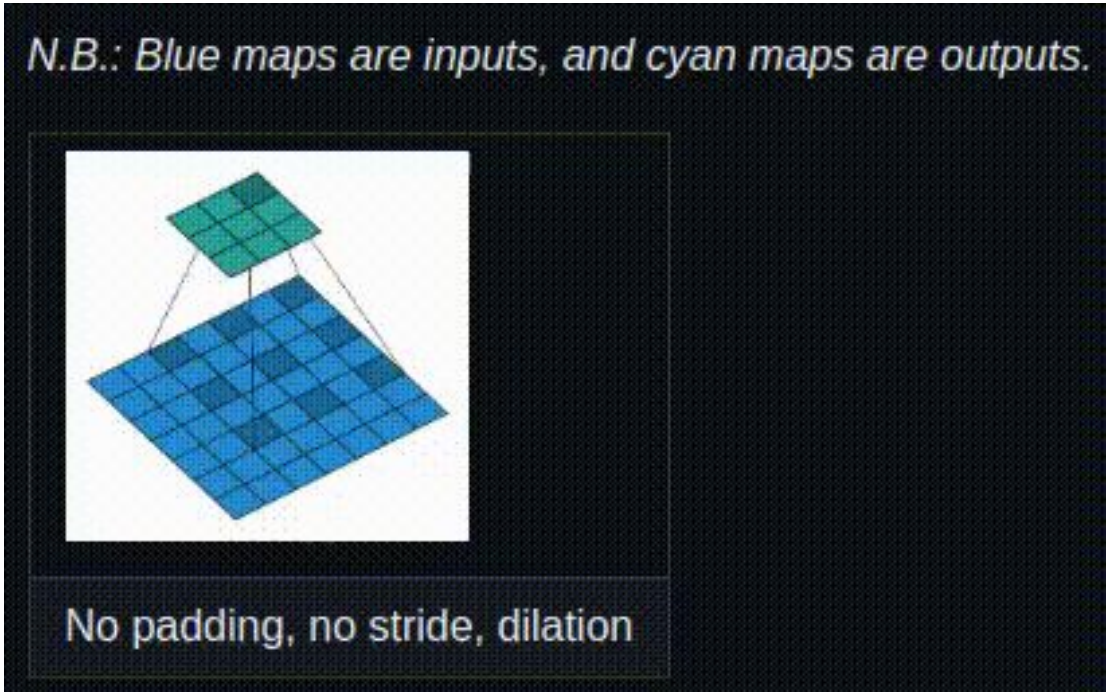


$$V(x,y,t) = \sum_{i=x-\delta}^{x+\delta} \sum_{j=y-\delta}^{y+\delta} \sum_{s=1}^{S} K(i-x+\delta, j-y+\delta, s, t) \cdot U(x+(i-x)\,d,\ y+(j-y)\,d, s)$$

Dilated convolutions, 2016



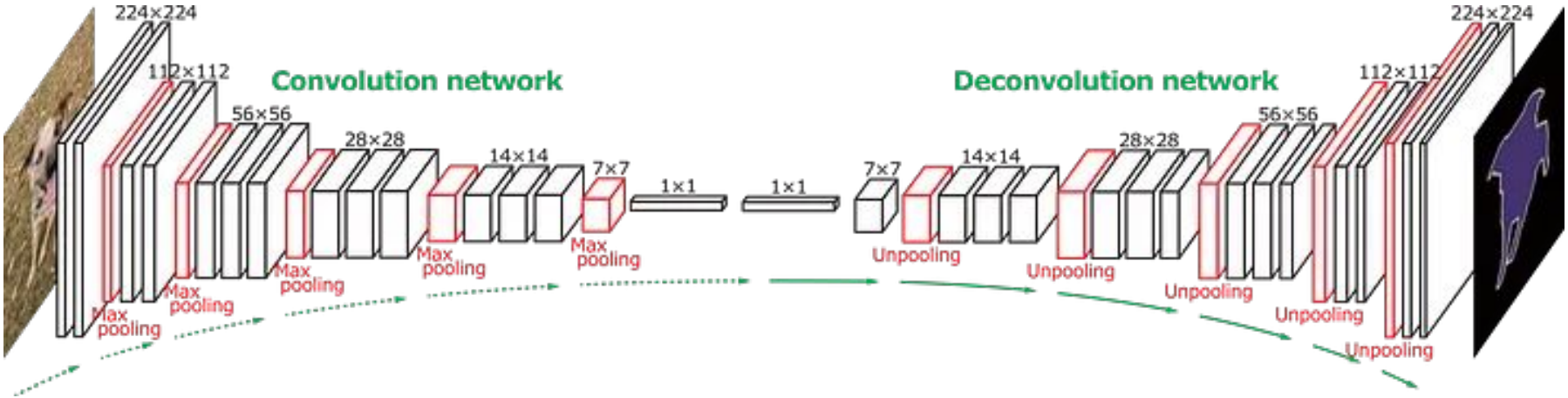
Dilated conv layer



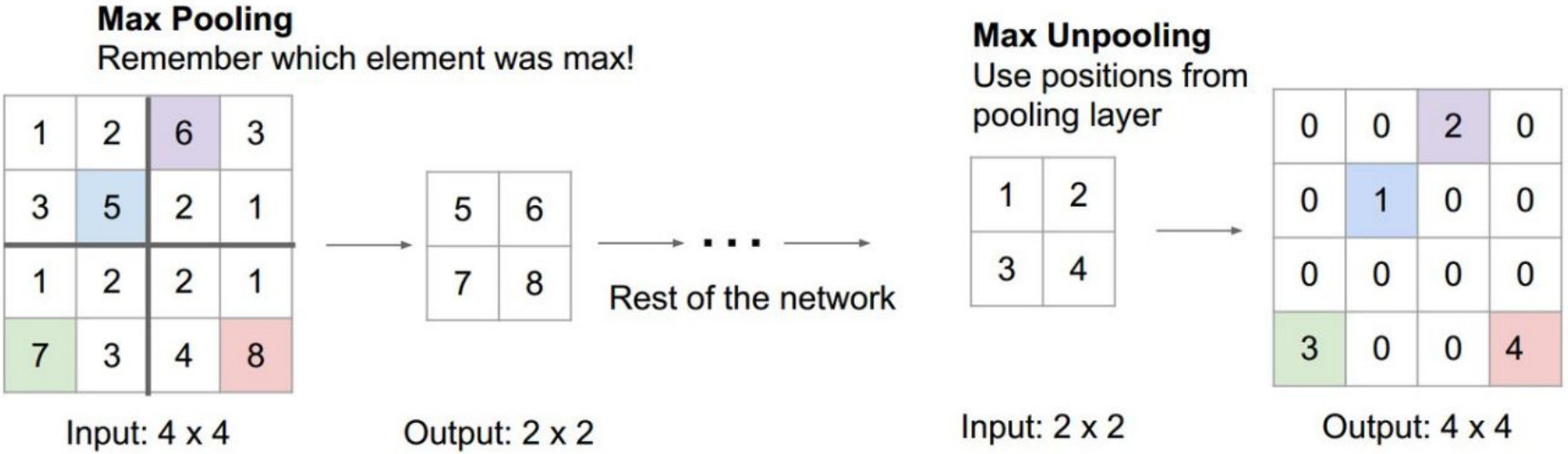
link

Fully convolutional nets

Какой должен быть размер картинки?



Max Unpooling



Сохраняем индексы каждого max-pooling слоя

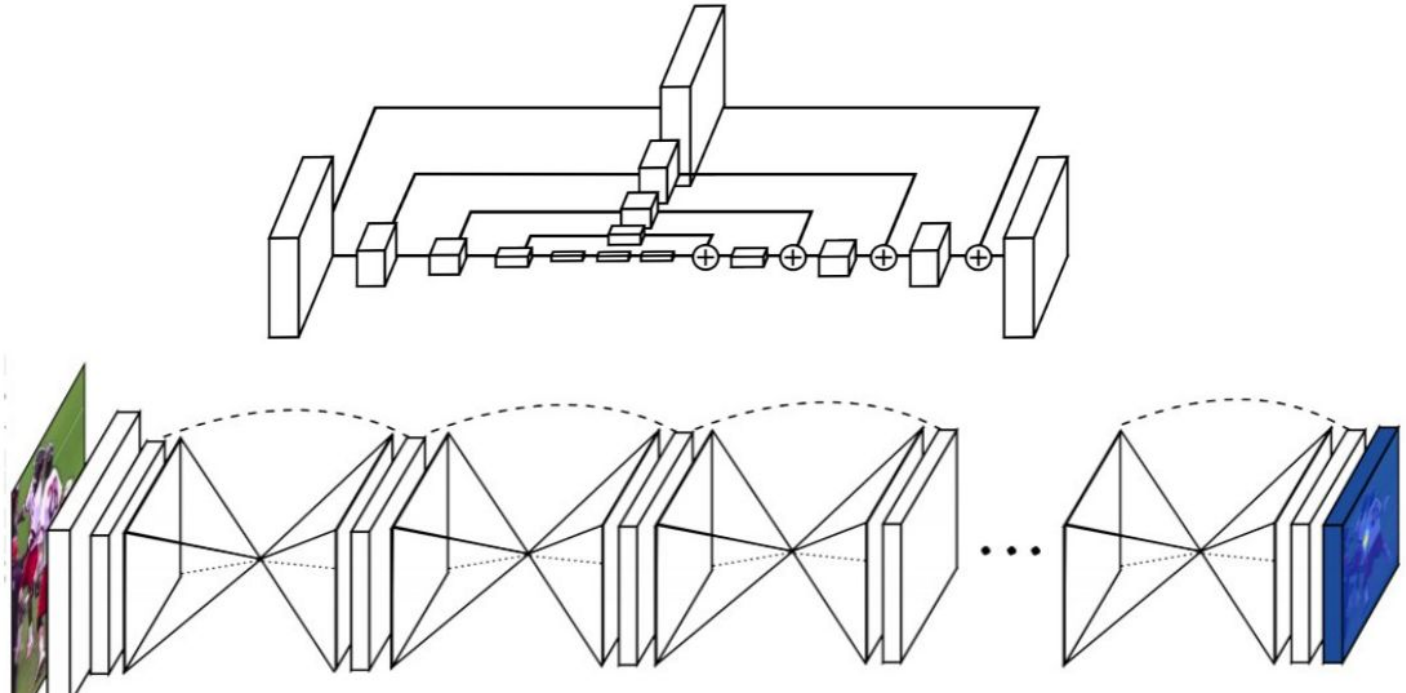
При повышении разрешения делаем так:

• Копируем значения из выхода max-pooling слоя с учётом запомненный индексов

• Применяем обученные свёртки для сглаживания

Stacked hourglass

Объединим модули в цепочку (каскад)



HRNet, 2019

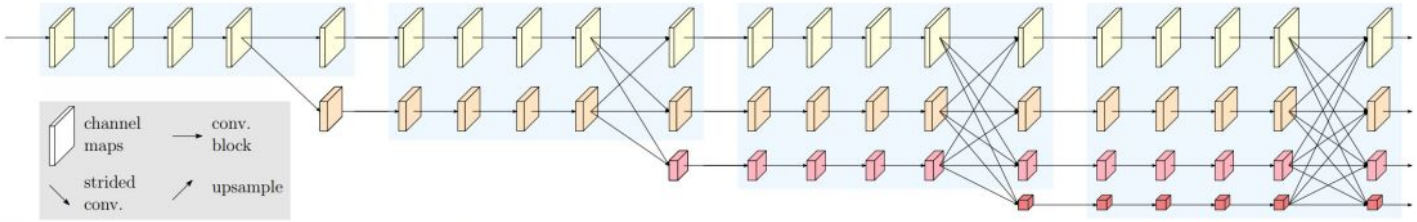


Figure 1. A simple example of a high-resolution network. There are four stages. The 1st stage consists of high-resolution convolutions. The 2nd (3rd, 4th) stage repeats two-resolution (three-resolution, four-resolution) blocks. The detail is given in Section 3.

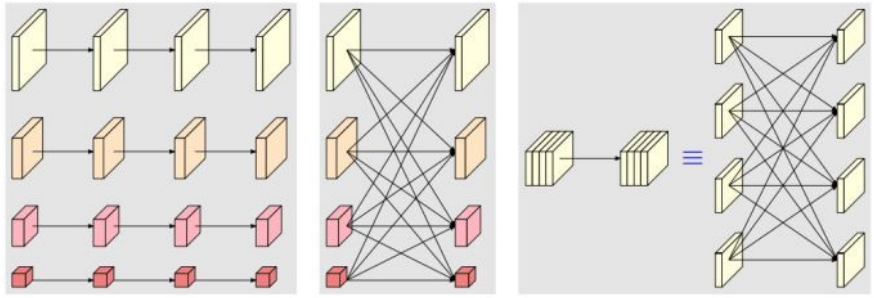
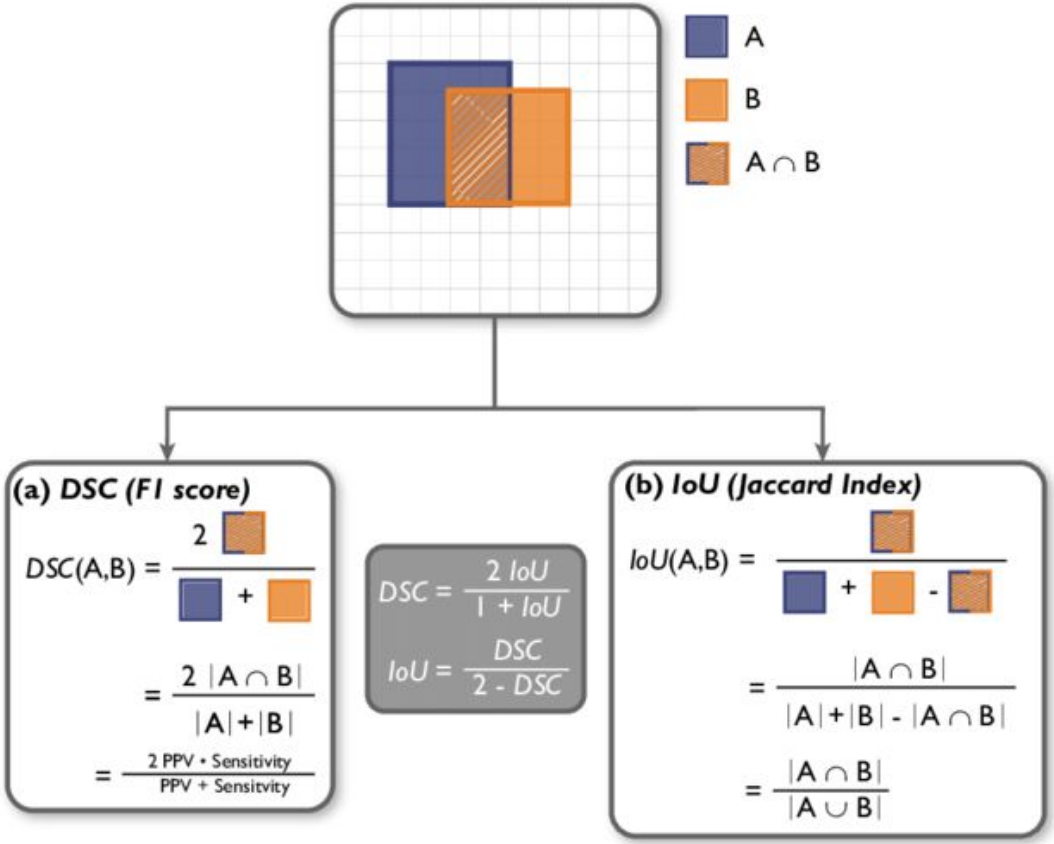


Figure 2. Multi-resolution block: (a) multi-resolution group convolution and (b) multi-resolution convolution. (c) A normal convolution (left) is equivalent to fully-connected multi-branch convolutions (right).

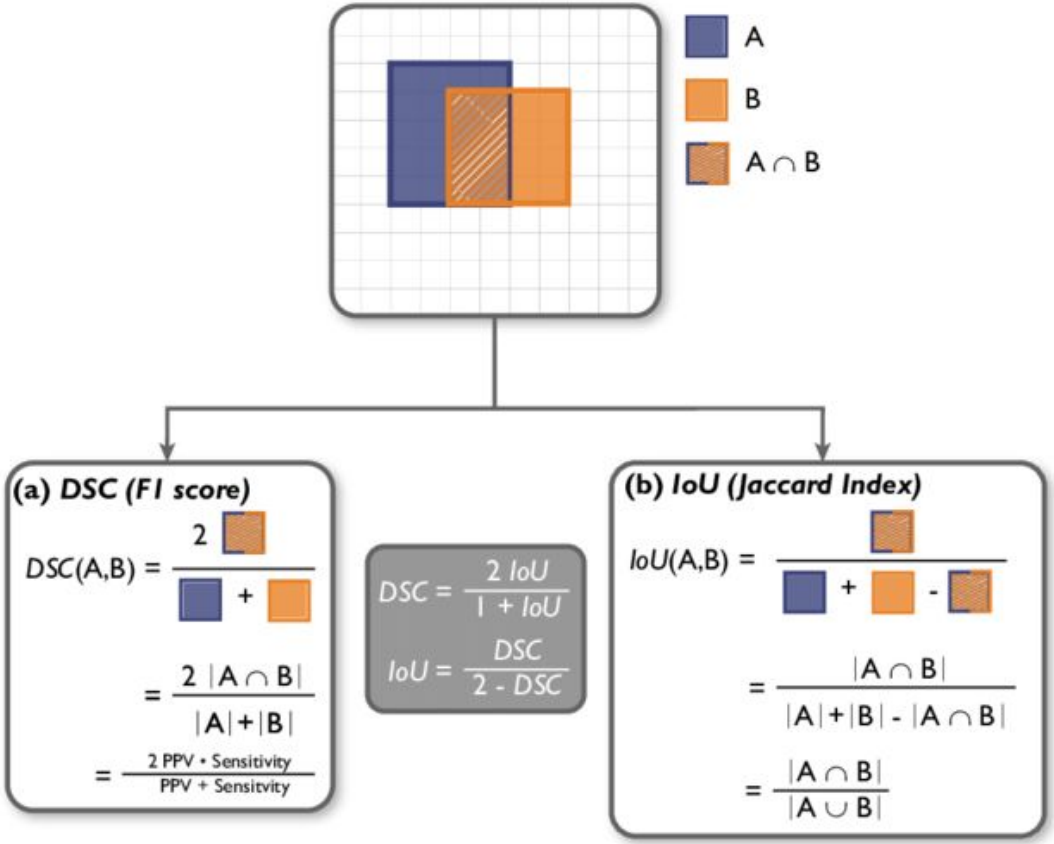
Recap

- Dilated convolutions
- Upsampling layers/upconvolution layers (aka transposed convolution/deconvolution)
- Skip connections (to retain fine-details)
- We can mix and match all of the above

Segmentation. Metrics?



Segmentation. Metrics?

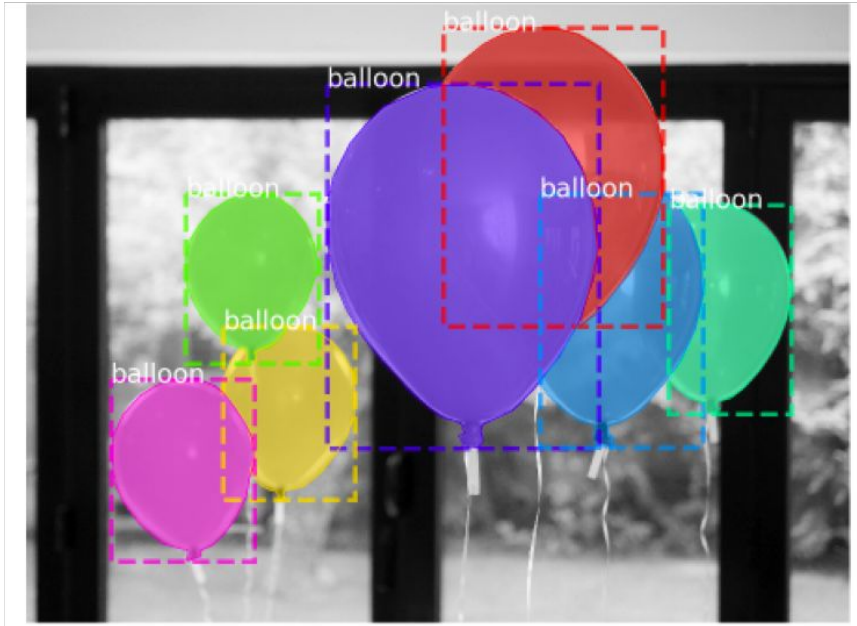


Иногда еще считают попиксельную точность/accuracy

Instance Segmentation. Metrics?

Semantic Segmentation

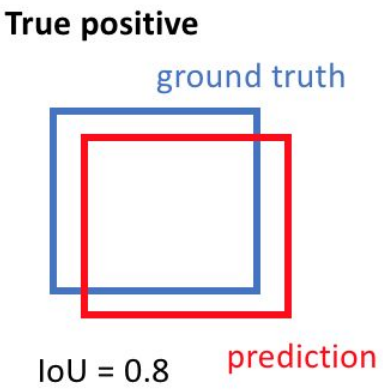
Instance Segmentation

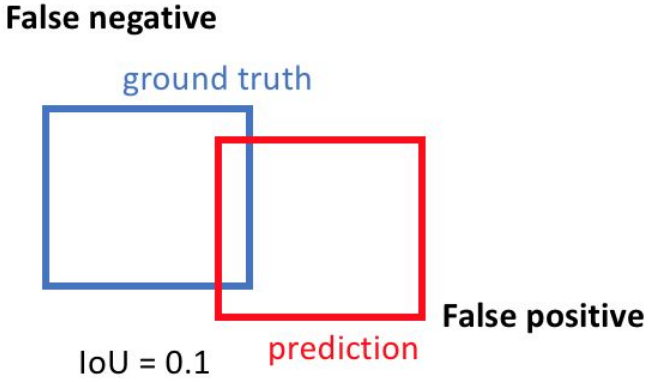


link

Instance Segmentation. Metrics?

Example
Threshold: 0.5





link

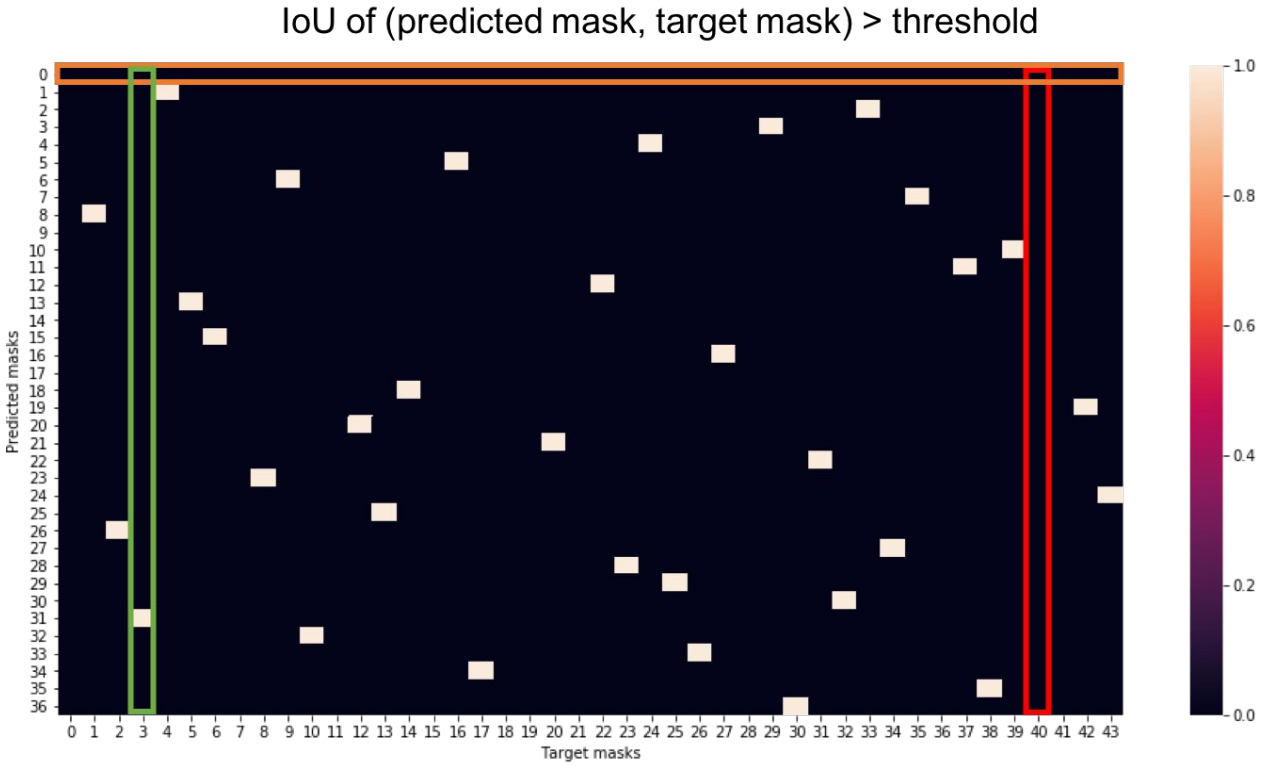
Instance Segmentation. Metrics?

False positive

Predicted mask has no corresponding ground truth label. We detected an object that didn't exist.

True positive

Ground truth mask has a corresponding predicted mask which has an IoU that exceeds the threshold value.

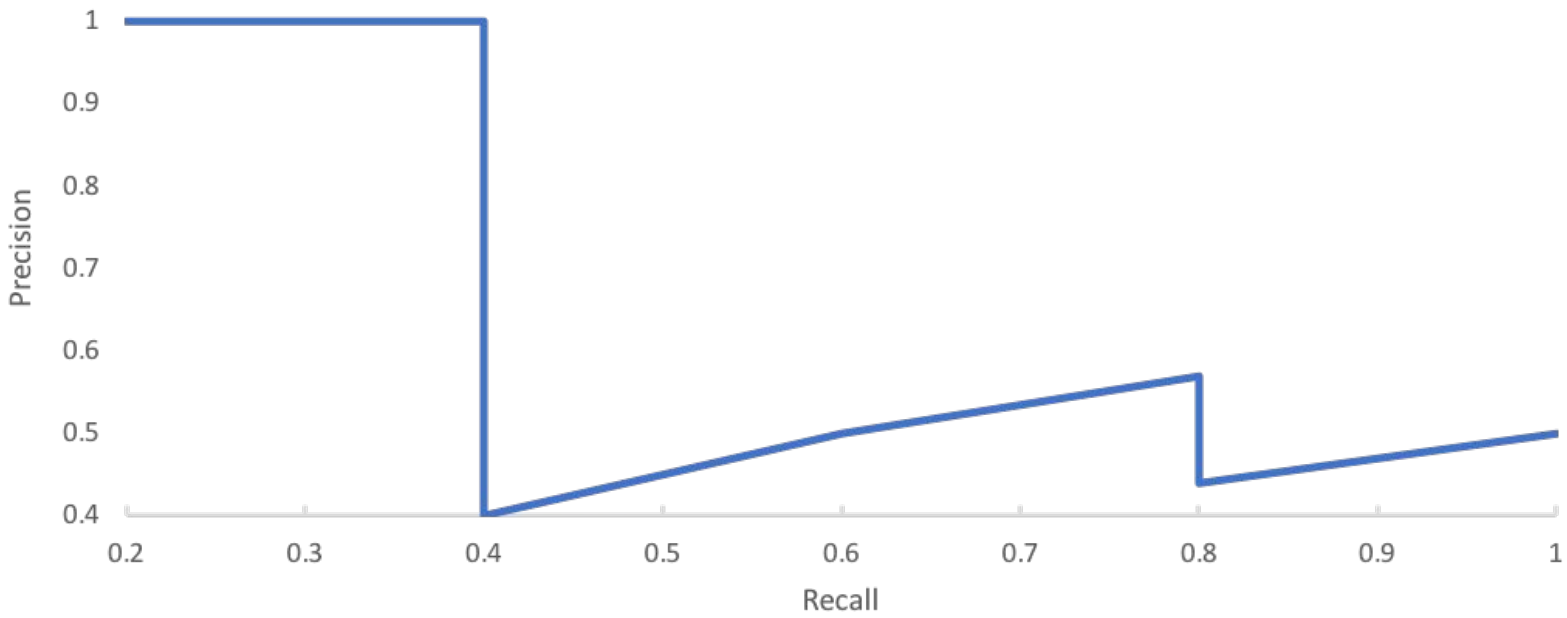


False negative

Ground truth mask has no corresponding predicted mask. We failed to identify this object.

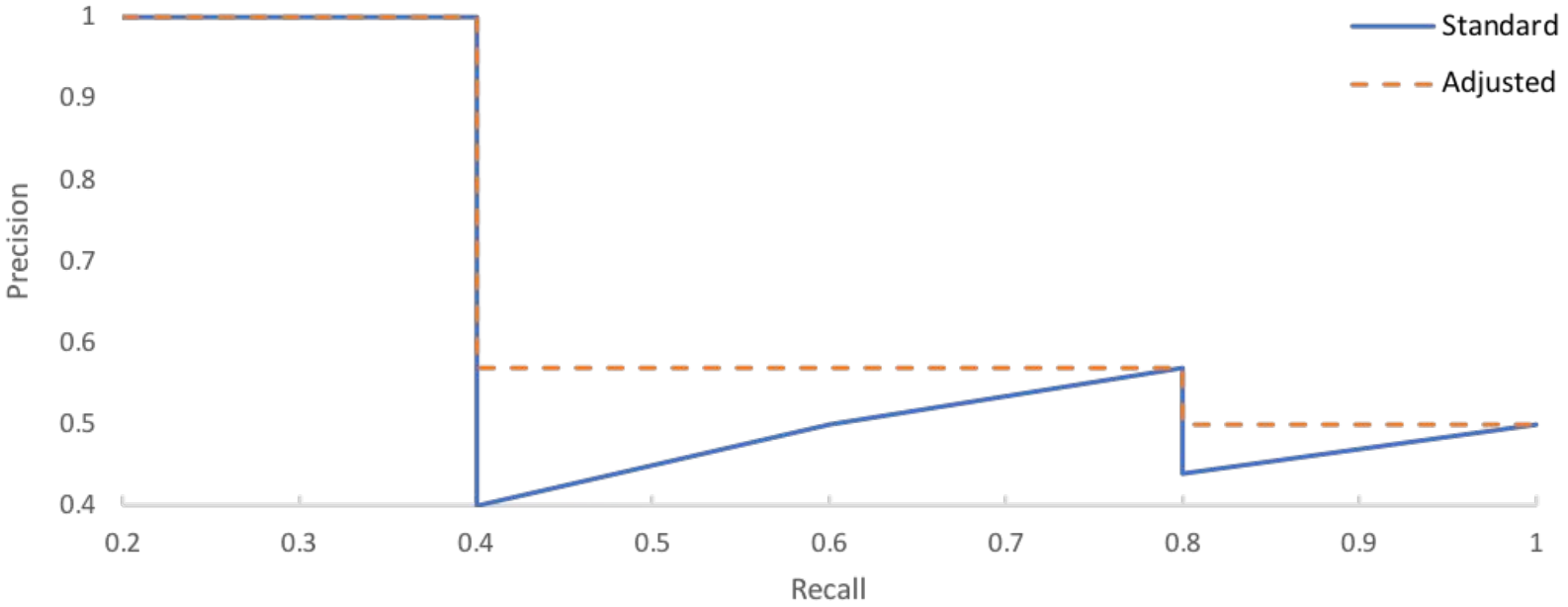
link

Instance Segmentation. Metrics?



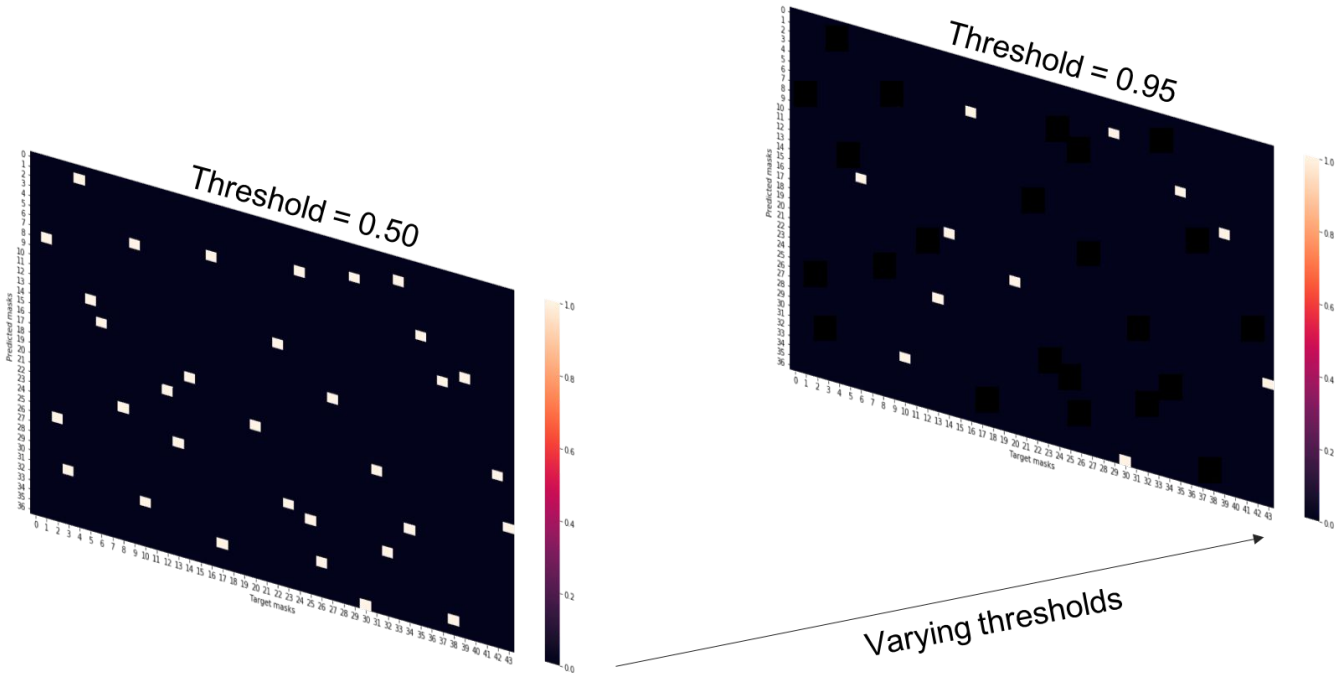
link

Instance Segmentation. Metrics?



link

Instance Segmentation. Metrics?



Less objects are recognized as “true positives” when we use a stricter IoU threshold

link

Segmentation. Loss functions?

$$\text{Loss} = -\sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i$$

$$L_{dice} = 1 - \frac{1}{C}\sum_{c=0}^{C-1} \frac{2\sum_{n=1}^{N} t_n^c y_n^c}{\sum_{n=1}^{N} (t_n^c + y_n^c)}$$

Segmentation. Loss functions?

Что делать, если классы сильно не сбалансированы?

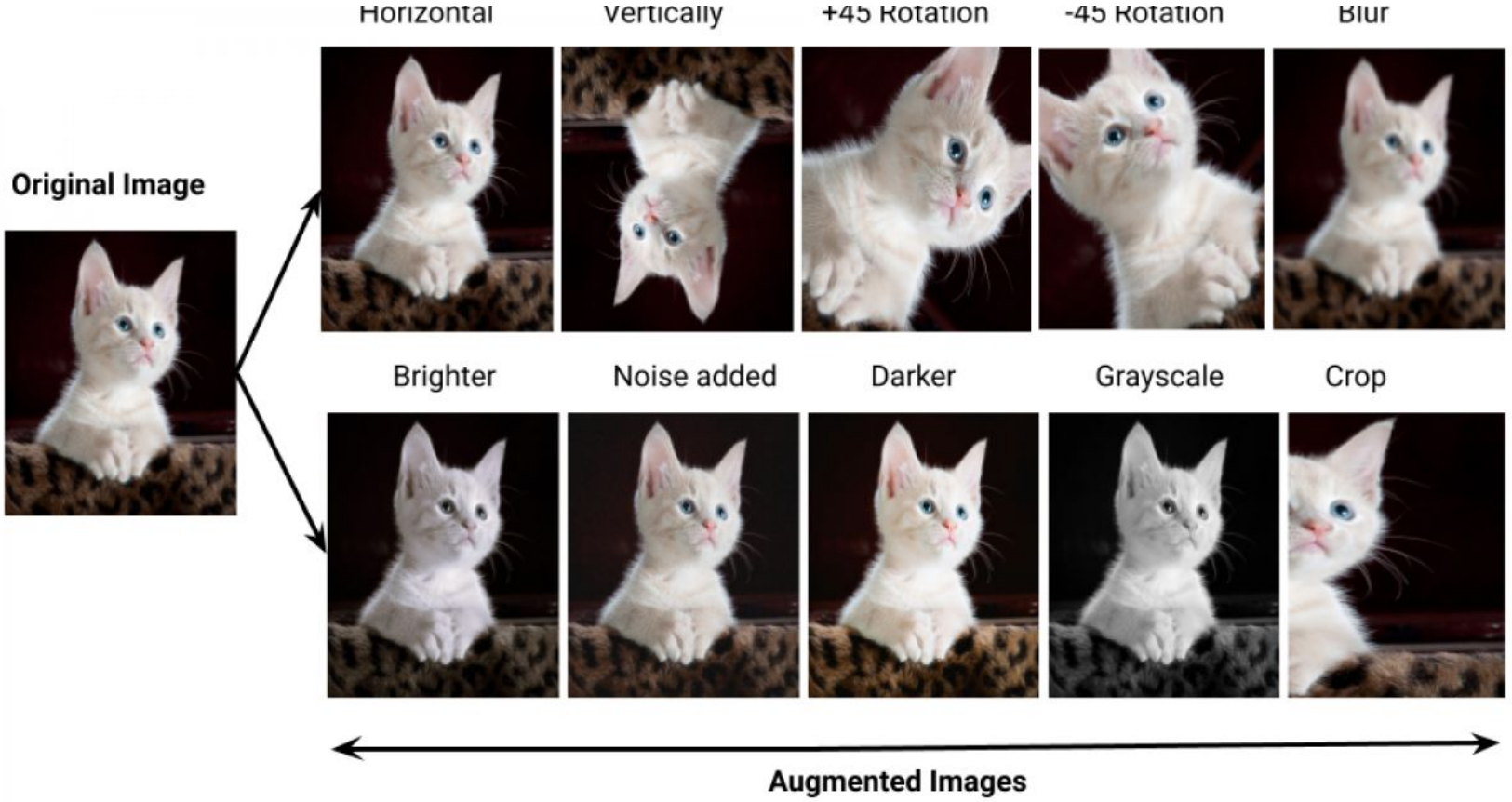
Segmentation. Loss functions?

Что делать, если классы сильно не сбалансированы?
Weighted Cross Entropy

$$l_n = -w_{y_n} \log \left(\frac{\exp(x_{n,y_n})}{\sum_{c=1}^{C} \exp(x_{n,c})} \right)$$

Что делать, если данных мало?

Аугментации



Пример датасета. Cityscapes

Изображения с камеры автомобиля

• 30 классов объектов

• 5000 хорошо размеченных и 20000 грубо размеченных

изображений

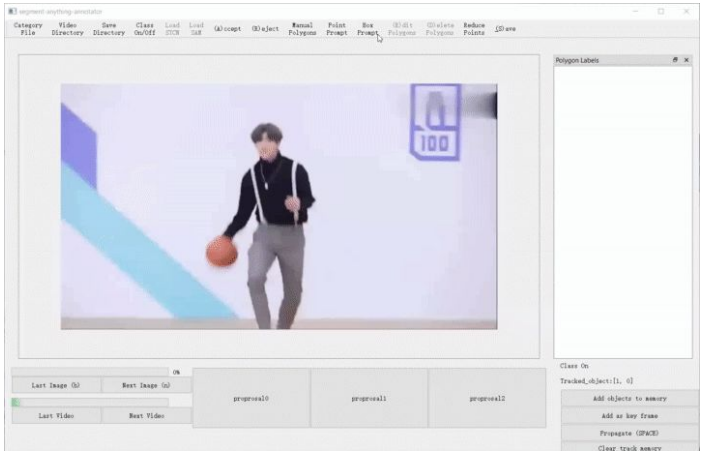
https://www.cityscapes-dataset.com/

Разметка

https://github.com/haochenheheda/segment-anything-annotator

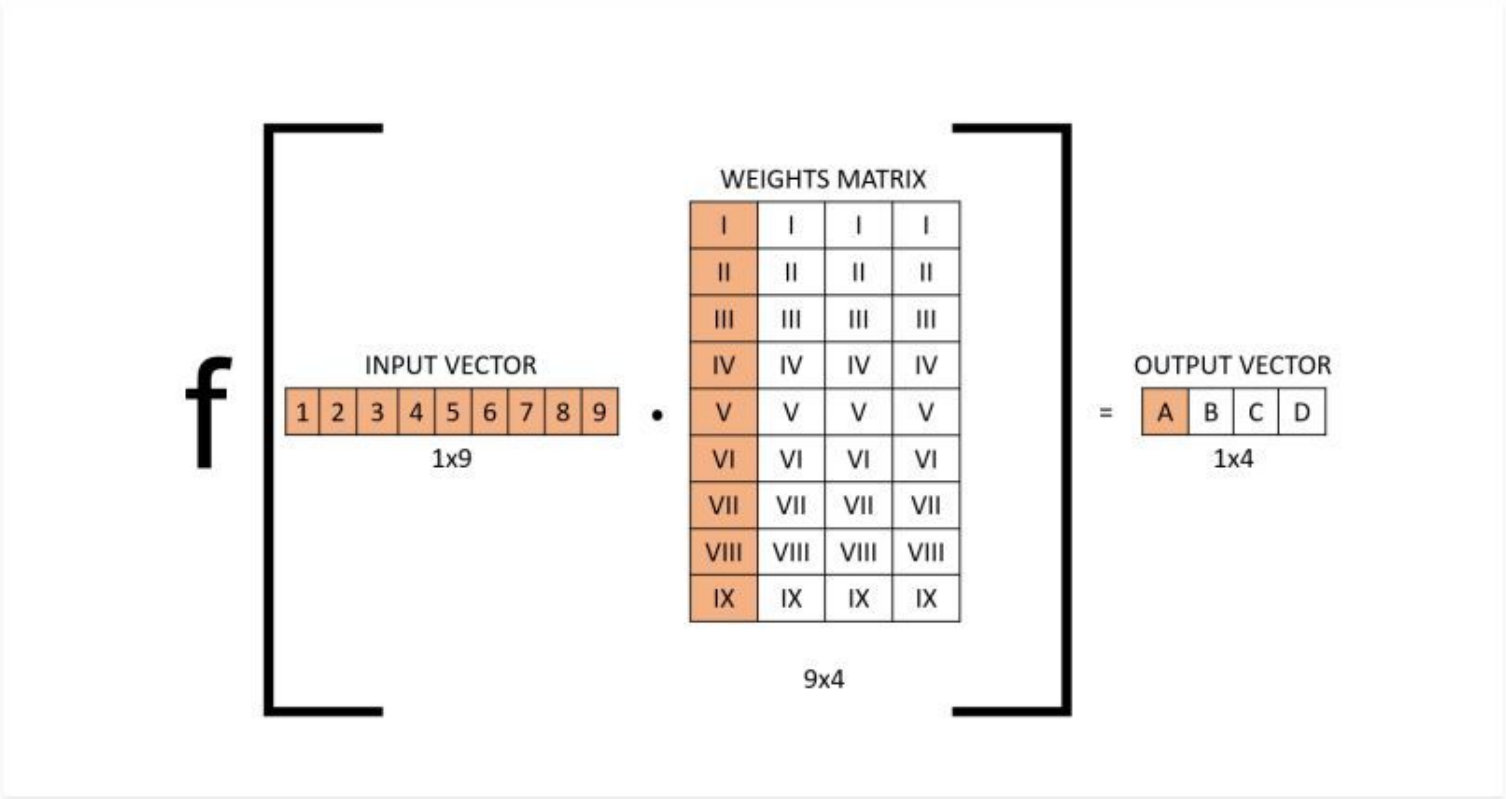
https://www.reddit.com/r/computervision/comments/179kyg3/are_there_any_tools_that_use_sam_for_segmentation/

https://humansintheloop.org/10-of-the-best-open-source-annotation-tools-for-computer-vision/

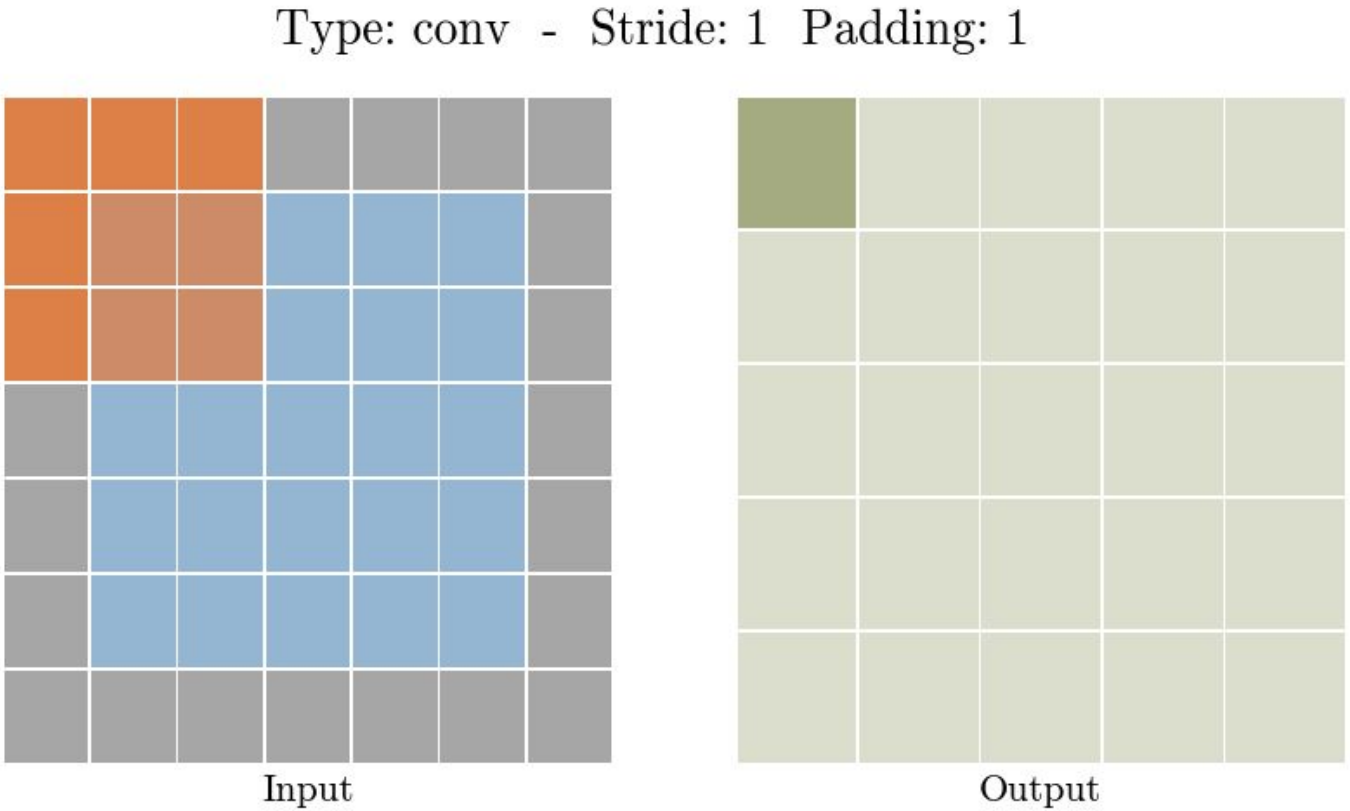


Recap conv

Linear layer

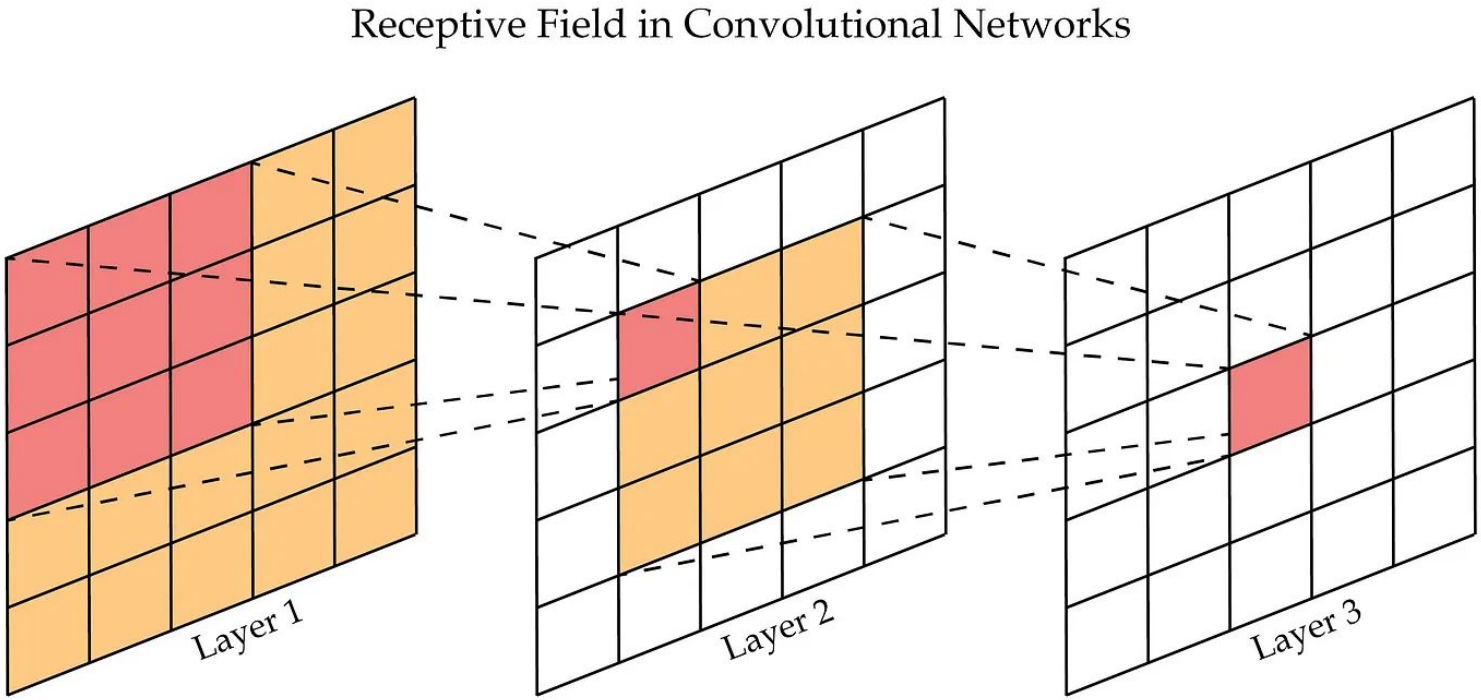


Conv layers

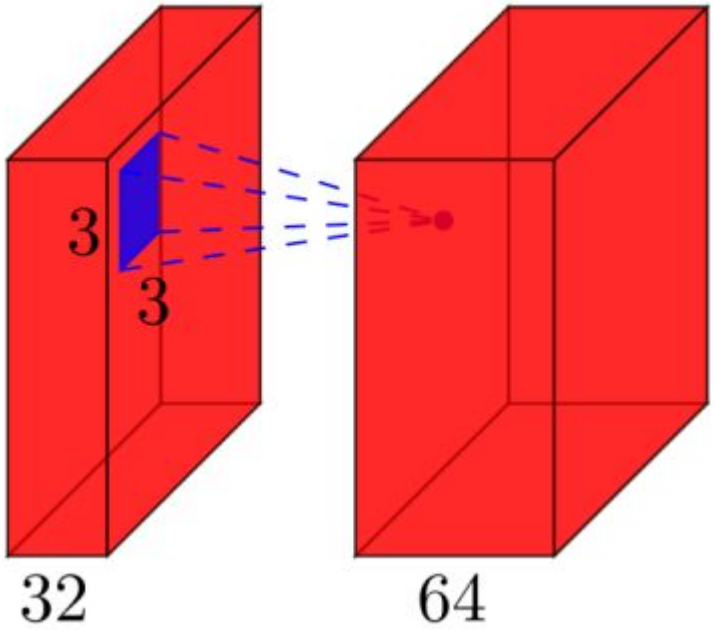


link

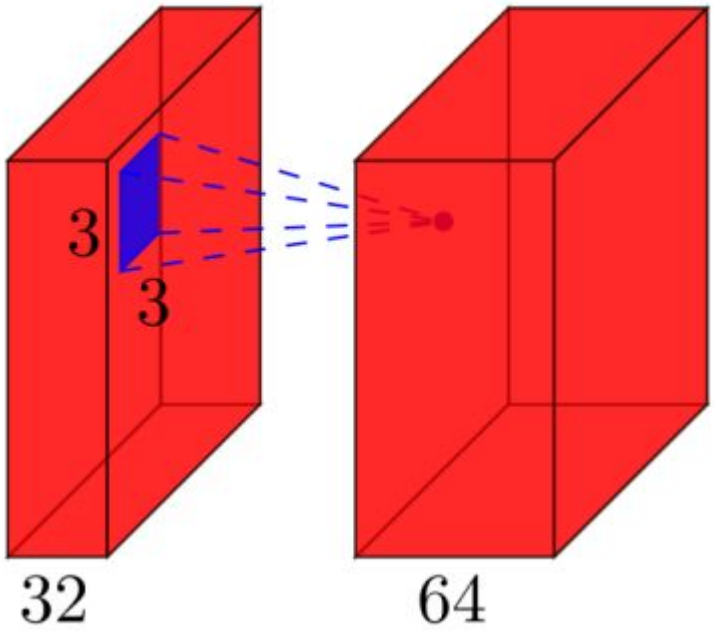
Receptive field



Num of params?



Num of params?



(3 * 3 * 32 +1) * 64

Где

3 * 3 - kernel_size

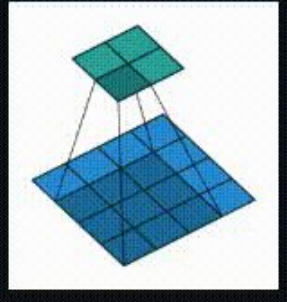
32 - in_channels

1 - bias

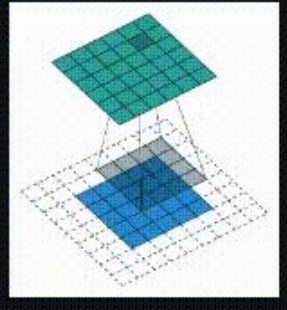
64 - out_channels

Conv layers

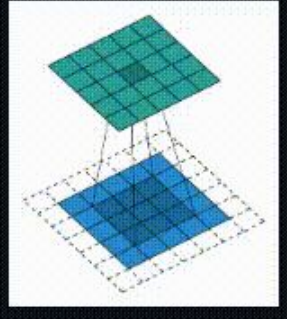
N.B.: Blue maps are inputs, and cyan maps are outputs.



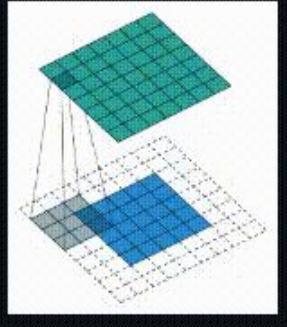
No padding, no strides



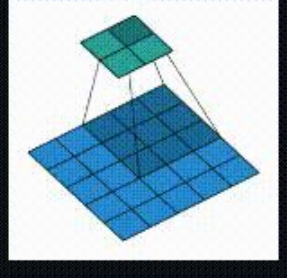
Arbitrary padding, no strides



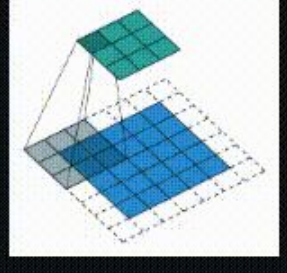
Half padding, no strides



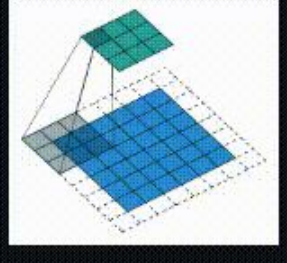
Full padding, no strides



No padding, strides



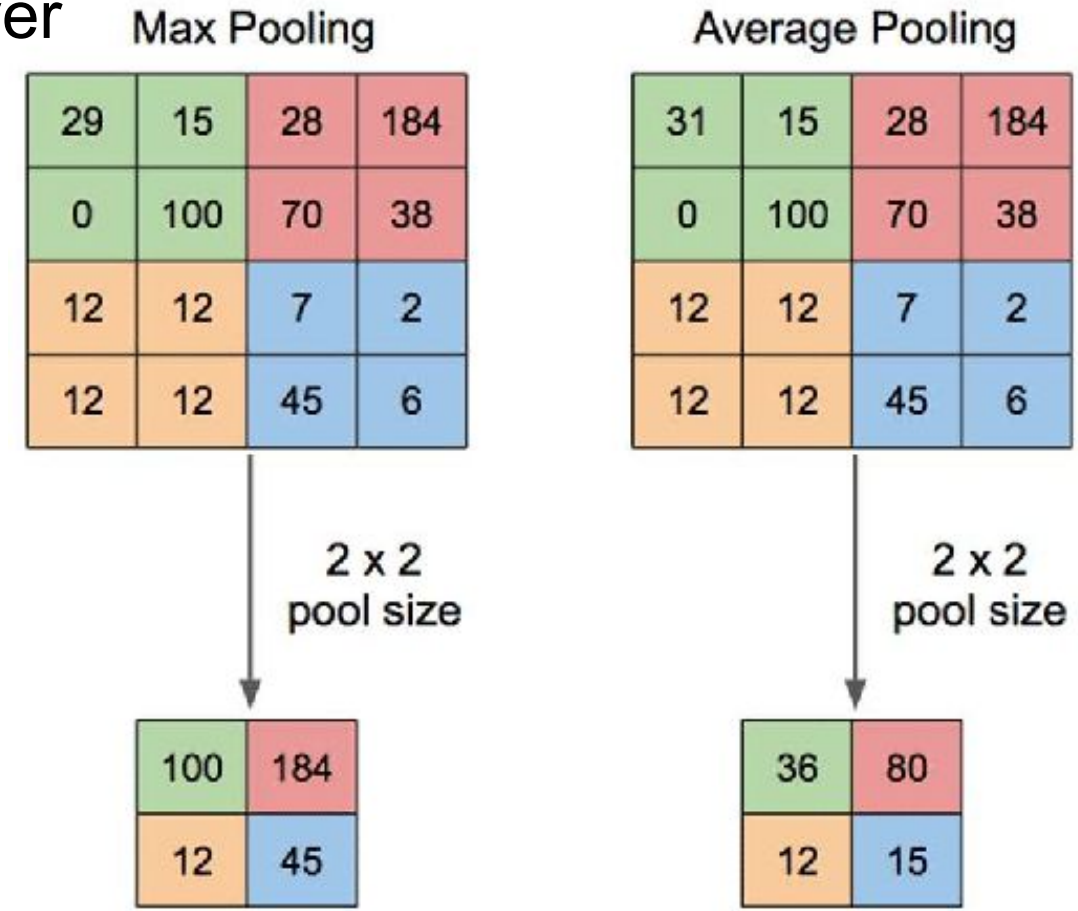
Padding, strides



Padding, strides (odd)

link

Pooling layer



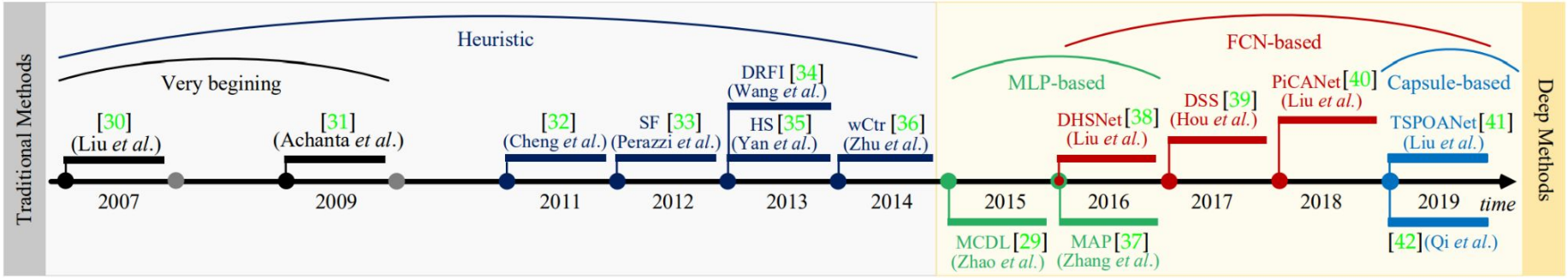
Salient detection

Salient detection

Salient detection

Цель состоит в том, чтобы обнаружить наиболее привлекающие внимание объекты в кадре и затем выделить для них силуэты с точностью до пикселя.

Salient detection. Architectures



Salient detection. Architectures

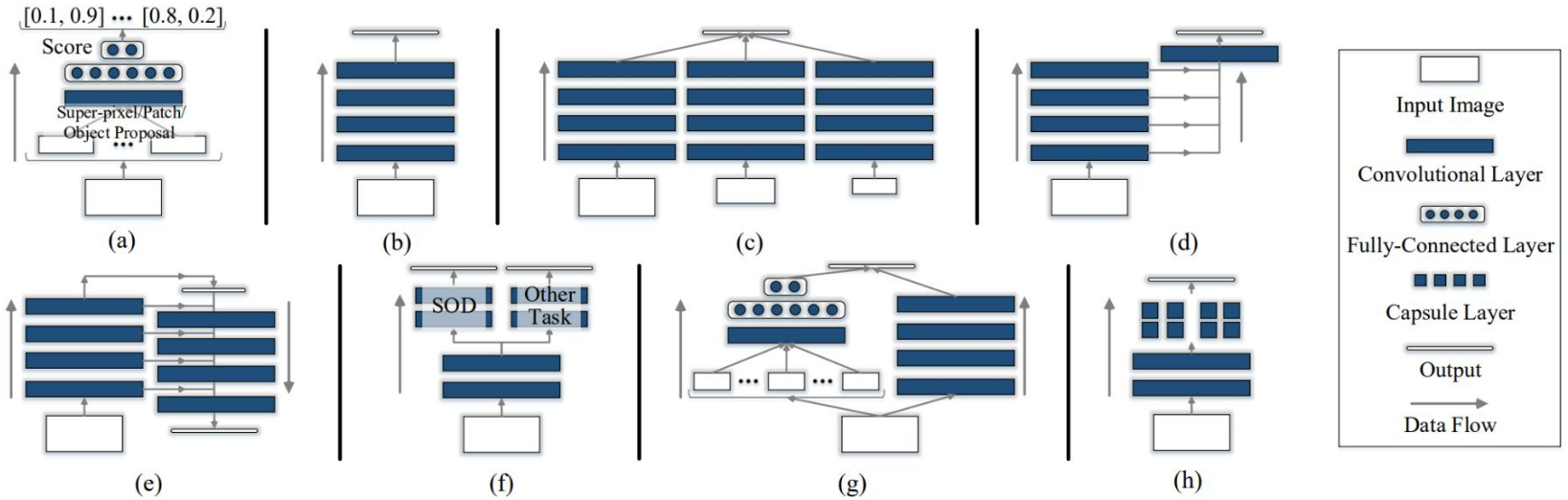
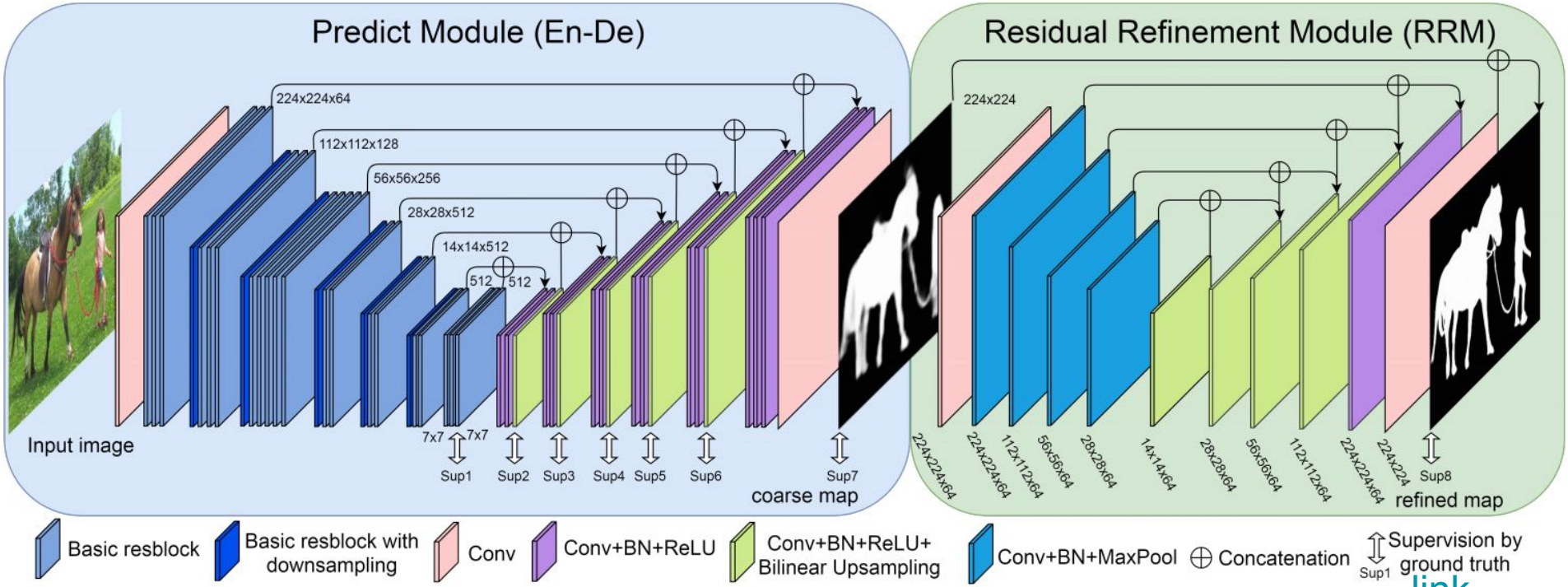


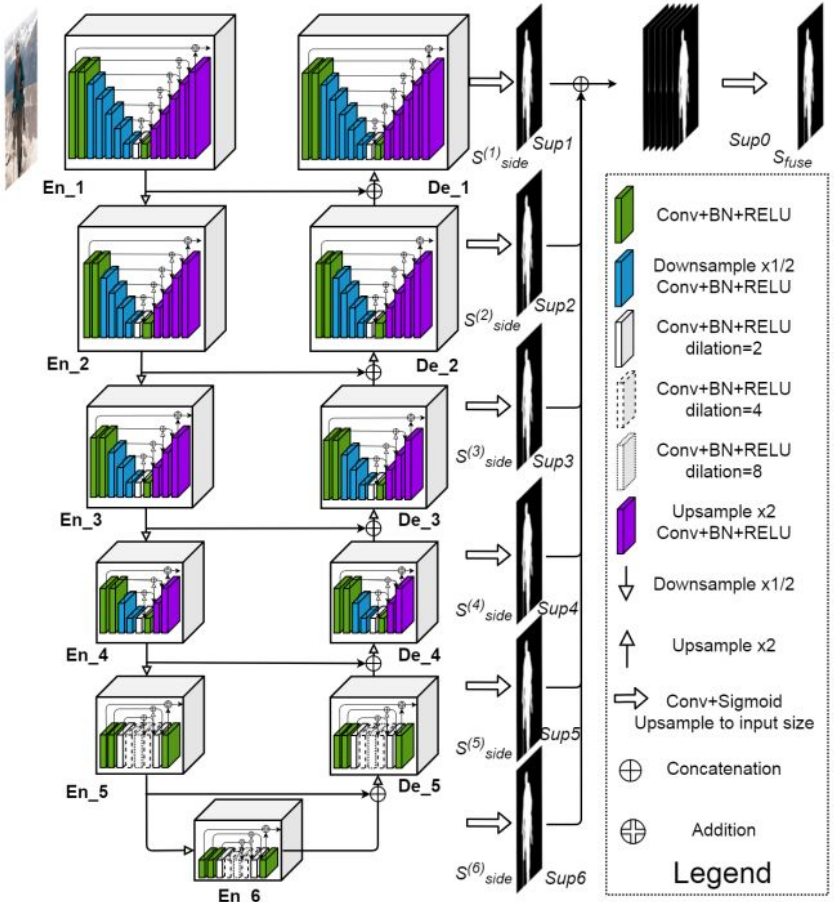
Fig. 2. Categorization of previous deep SOD models according to the adopted network architecture. (a) MLP-based methods. (b)-(f) FCN-based methods, mainly using (b) single-stream network, (c) multi-stream network, (d) side-out fusion network, (e) bottom-up/top-down network, and (f) branch network architectures. (g) Hybrid network-based methods. (h) Capsule-based methods. See §2.1 for more detailed descriptions.

Salient detection. Architectures
BASNet

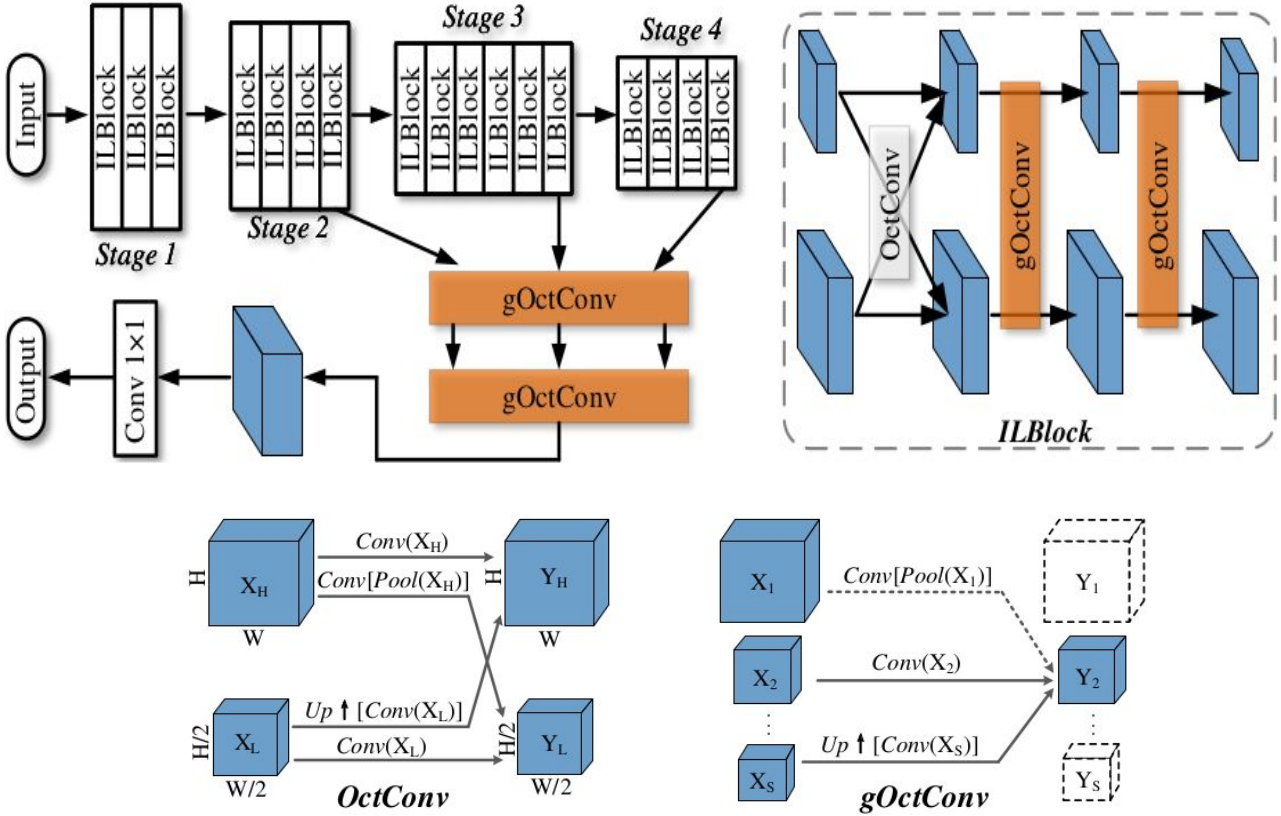


link

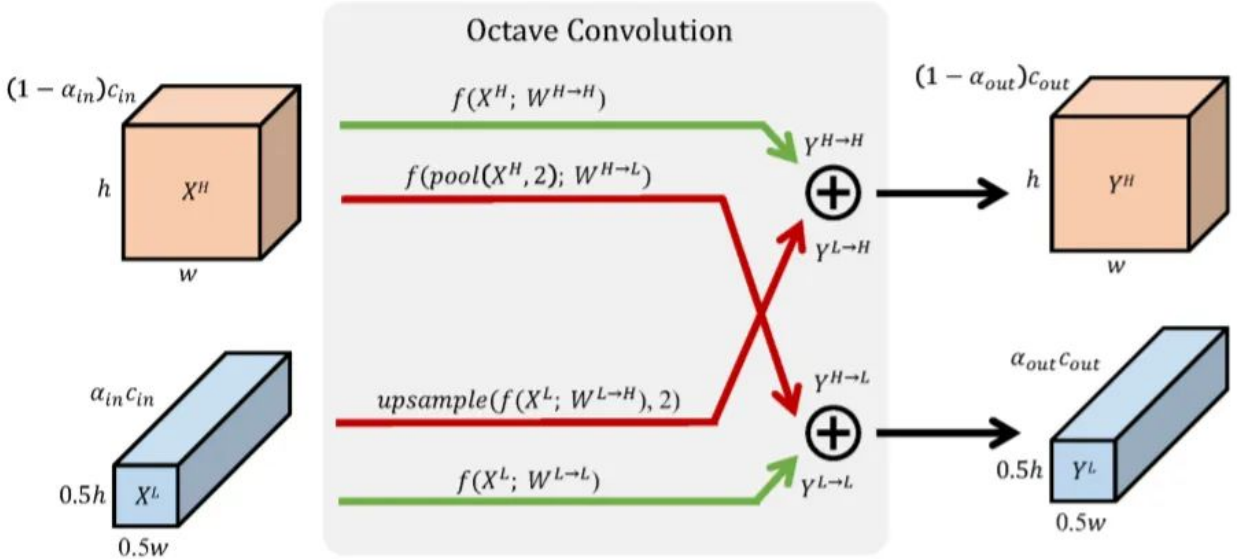
Salient detection. Architectures
U^2Net



Salient detection. Architectures



OctConv



(a) Detailed design of the Octave Convolution. Green arrows correspond to information updates while red arrows facilitate information exchange between the two frequencies.

Salient detection. Metrics

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

$$F_\beta = \frac{(1+\beta^2)\text{Precision} \times \text{Recall}}{\beta^2 \text{Precision} + \text{Recall}}.$$

$$\text{MAE} = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} |\boldsymbol{G}(i,j) - \boldsymbol{S}(i,j)|$$

$$F_\beta^\omega = \frac{(1+\beta^2)\text{Precision}^\omega \times \text{Recall}^\omega}{\beta^2 \text{Precision}^\omega + \text{Recall}^\omega}.$$

- **S-measure** [138] evaluates the structural similarity between the real-valued saliency map and the binary ground-truth. It considers object-aware (S_o) and region-aware (S_r) structure similarities:

$$S = \alpha \times S_o + (1-\alpha) \times S_r, \tag{5}$$

where α is empirically set to 0.5.

- **E-measure** [139] considers global means of the image and local pixel matching simultaneously:

$$Q_{\boldsymbol{S}} = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} \phi_{\boldsymbol{S}}(i,j), \tag{6}$$

where $\phi_{\boldsymbol{S}}$ is the enhanced alignment matrix, reflecting the correlation between $\boldsymbol{S}$ and $\boldsymbol{G}$ after subtracting their global means, respectively.

link