

Сегментация

Tatiana Gaintseva

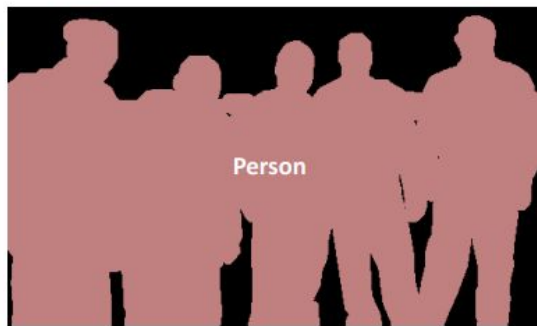
План

- Задача сегментации
- Идеи решения
- Архитектуры и трюки

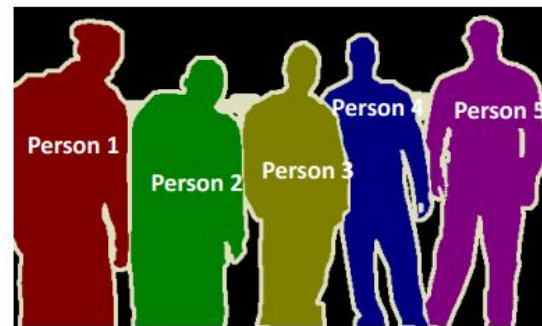
Задача сегментации



Object Detection



Semantic Segmentation

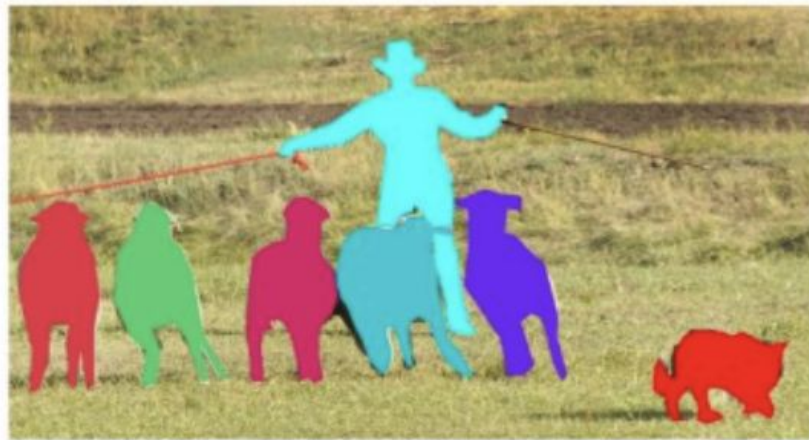


Instance Segmentation

Задача сегментации

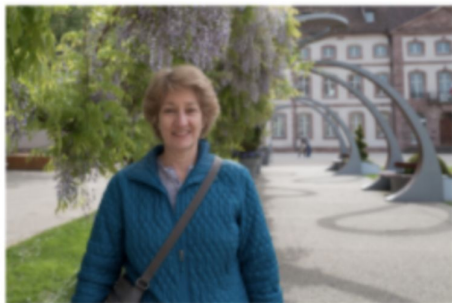


Сегментируется все



... или только определенные объекты

Задача сегментации



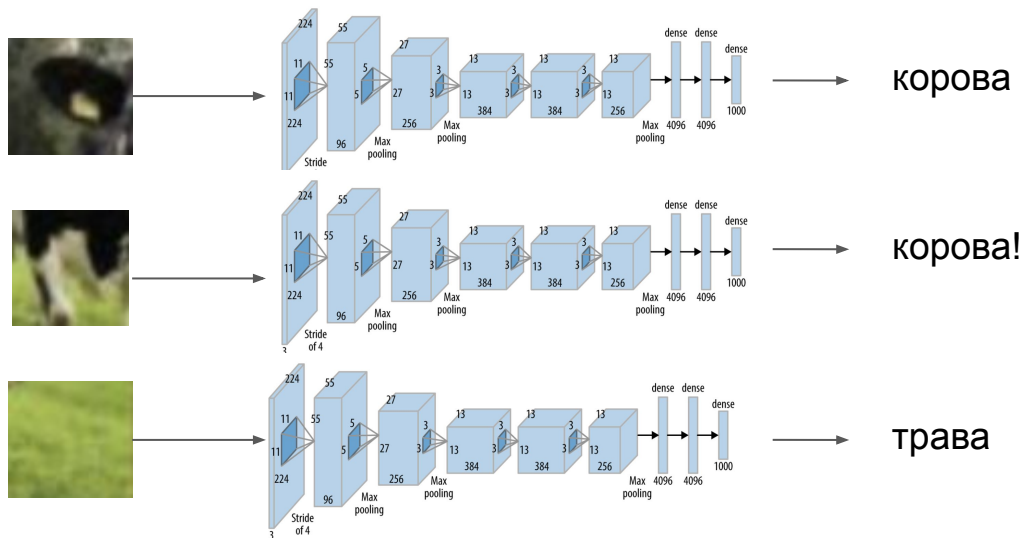
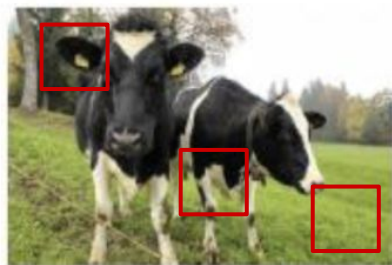
- 0. Unknown
- 1. Person
- 2. Bag
- 3. Grass
- 4. Road
- 5. Building

[0, 00, 0.01, 0.003, 0.67, 0.31, 0.097]



Loss: Pixel-wise Softmax

Идея решения: Sliding Window

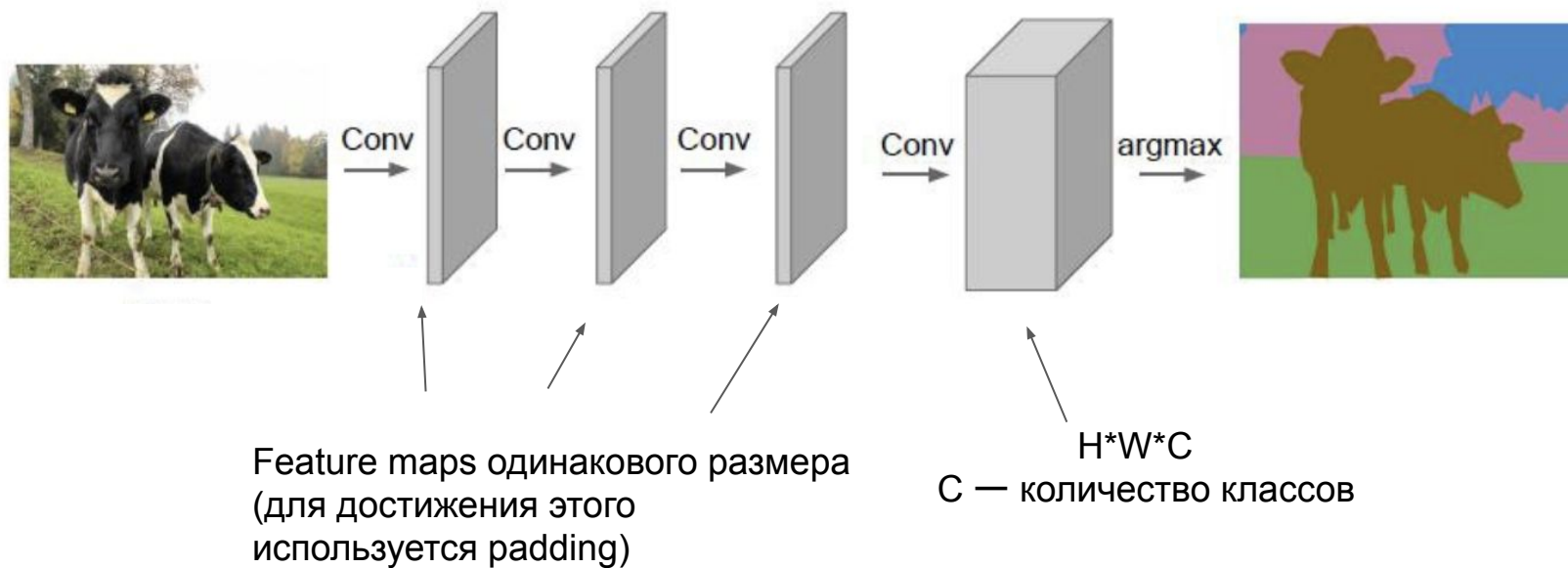


Идея решения: Sliding Window

Недостатки:

- Computationally expensive: нужно совершить несколько (много!) вызовов сети для классификации
- Sliding windows не шарят между собой информацию о частях картинки

Идея решения: Fully-conv network



Идея решения: Fully-conv network

Недостатки:

- Computationally expensive: огромное количество параметров

Идея решения: Fully-conv network

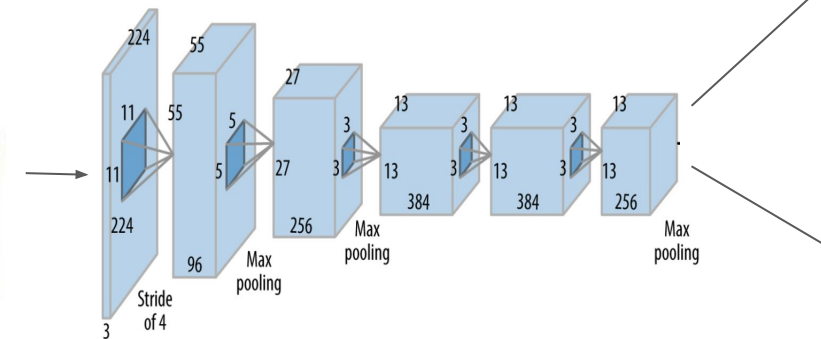
Недостатки:

- Computationally expensive: огромное количество параметров

А значит, нам нужно сжимать информацию о картинке перед построением карты сегментации.

Идея решения: CNN

VGG, ResNet, Inception... whatever



upsampling



Nearest Neighbor

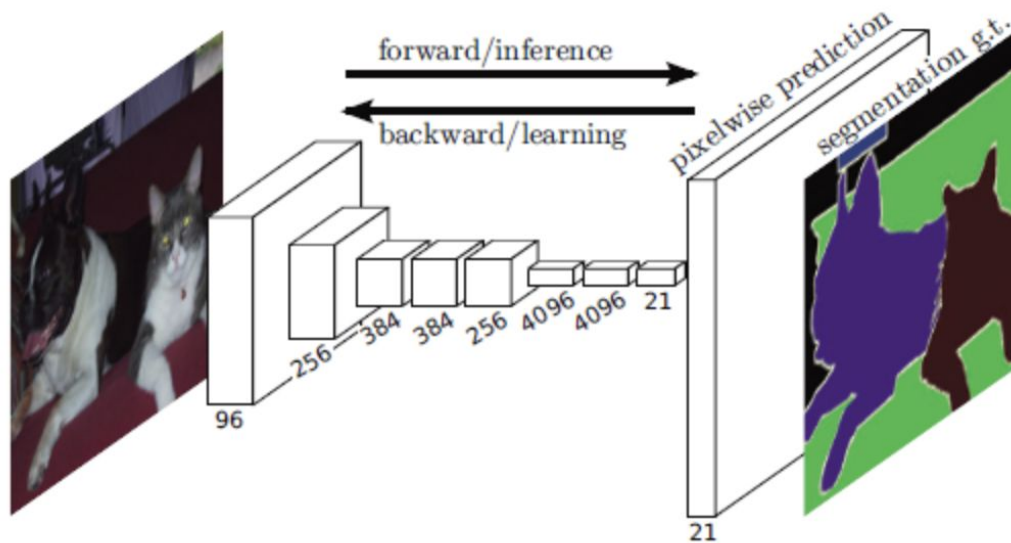
1	2
3	4

Input: 2 x 2

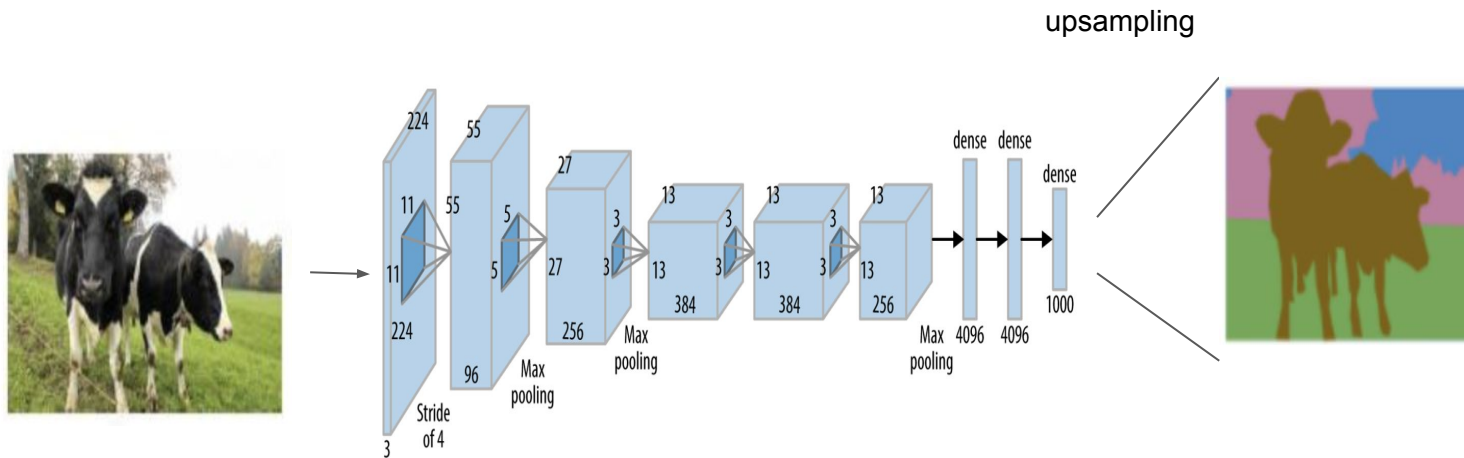
1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

Output: 4 x 4

Идея решения: CNN



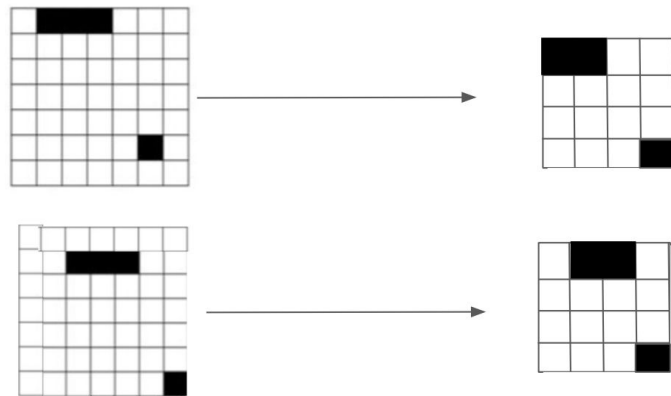
Идея решения: CNN



FCN: недостатки

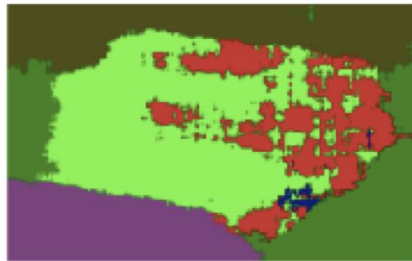
Ну, кажется, сегментация будет не очень...

- Upsampling плохо восстанавливает информацию
- Downsampling и большой stride разрушают пространственную информацию
- Scale Variability



Идея решения: CNN

Сегментация сверточной сетью

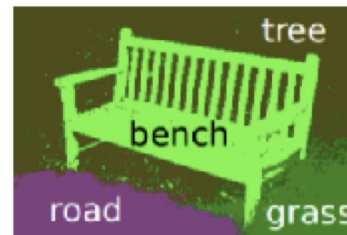


Идея решения: CNN

Сегментация сверточной сетью



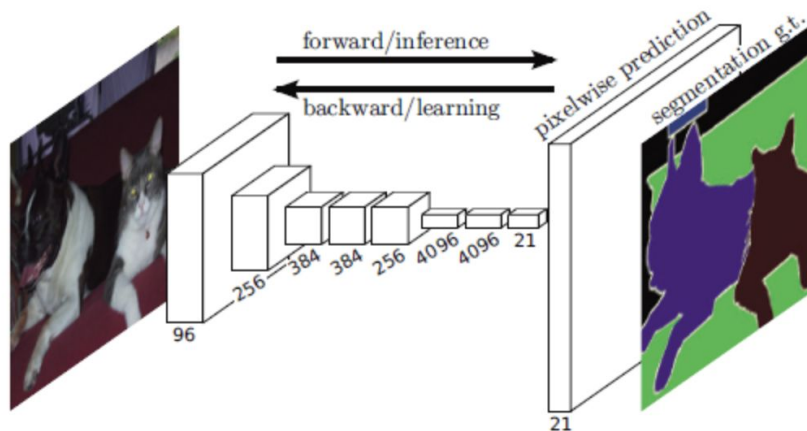
Postprocessing (CRF)



Segmentation Approaches

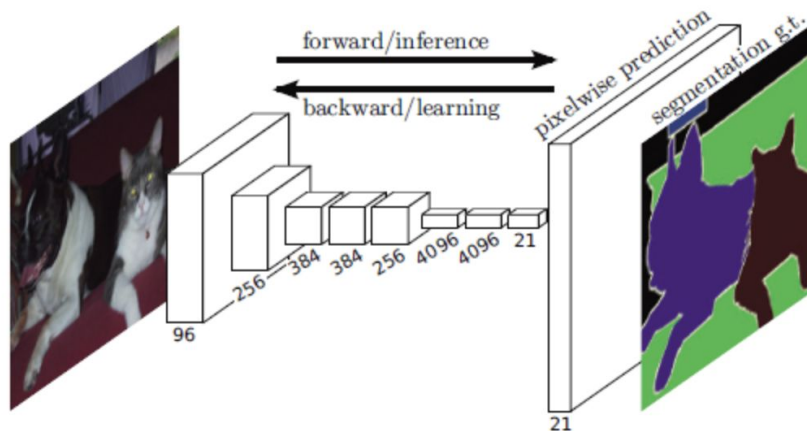
FCN: недостатки

- Upsampling плохо восстанавливает информацию
- Downsampling и большой stride разрушают пространственную информацию
- Scale Variability

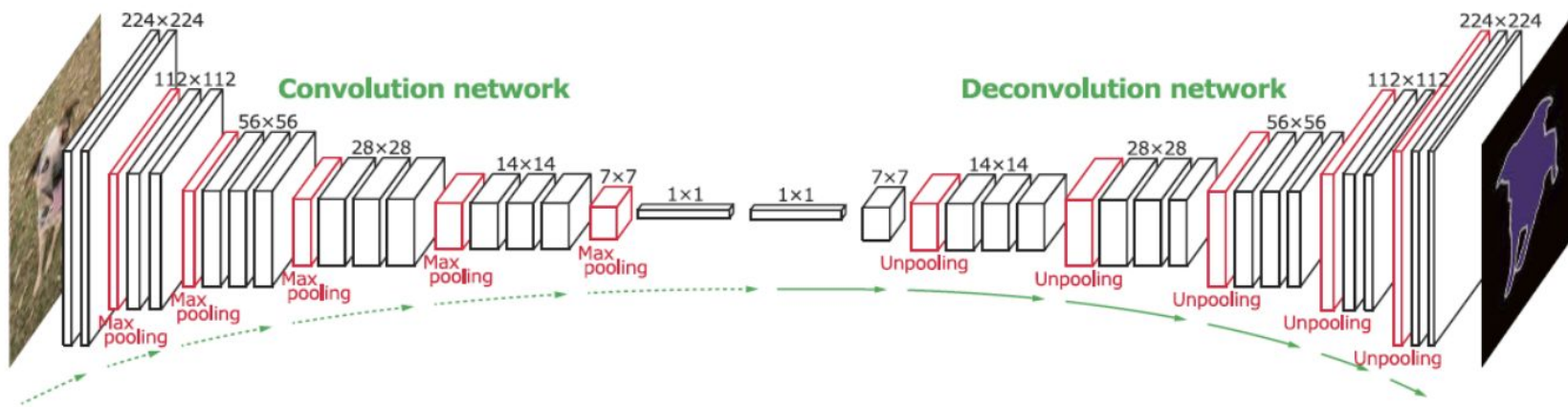


FCN: недостатки

- **Upsampling плохо восстанавливает информацию**
- Downsampling и большой stride разрушают пространственную информацию
- Scale Variability

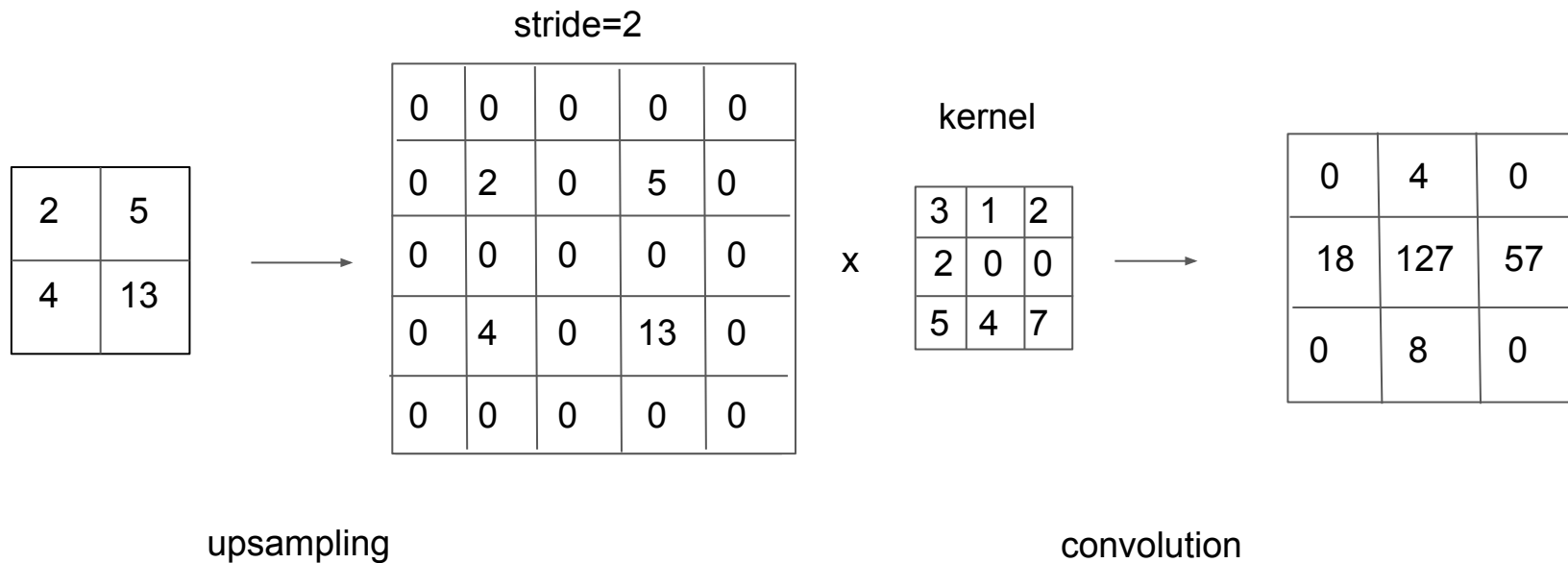


DeConvNet

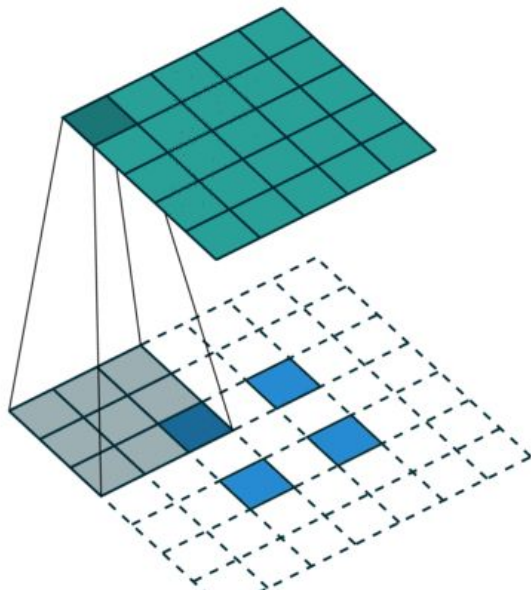


Transposed Convolution*

*также известный как deconvolution в литературе



Transposed Convolution*



CLASS `torch.nn.ConvTranspose2d(in_channels: int, out_channels: int, kernel_size: Union[T, Tuple[T, T]], stride: Union[T, Tuple[T, T]] = 1, padding: Union[T, Tuple[T, T]] = 0, output_padding: Union[T, Tuple[T, T]] = 0, groups: int = 1, bias: bool = True, dilation: int = 1, padding_mode: str = 'zeros')`

[SOURCE]

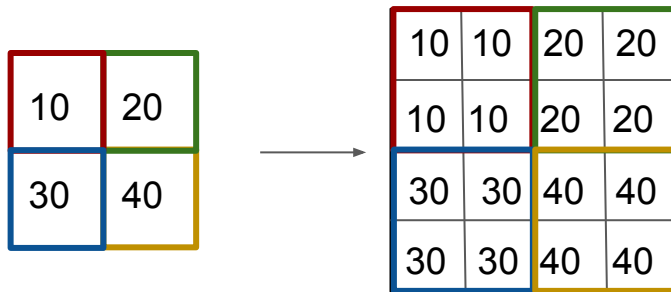
Transposed Convolution

```
CLASS torch.nn.ConvTranspose2d(in_channels: int, out_channels: int, kernel_size:  
Union[T, Tuple[T, T]], stride: Union[T, Tuple[T, T]] = 1, padding: Union[T,  
Tuple[T, T]] = 0, output_padding: Union[T, Tuple[T, T]] = 0, groups: int =  
1, bias: bool = True, dilation: int = 1, padding_mode: str = 'zeros')
```

[SOURCE]

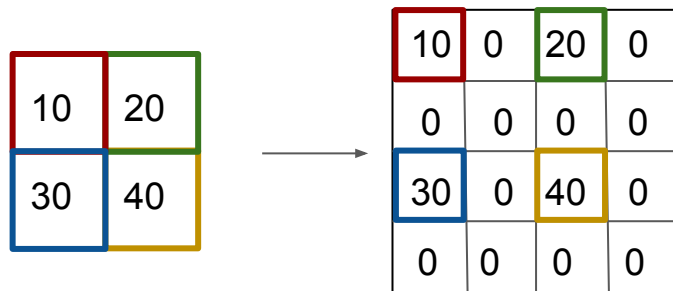
Upsampling

Nearest Neighbours



Upsampling

Bed of Nails

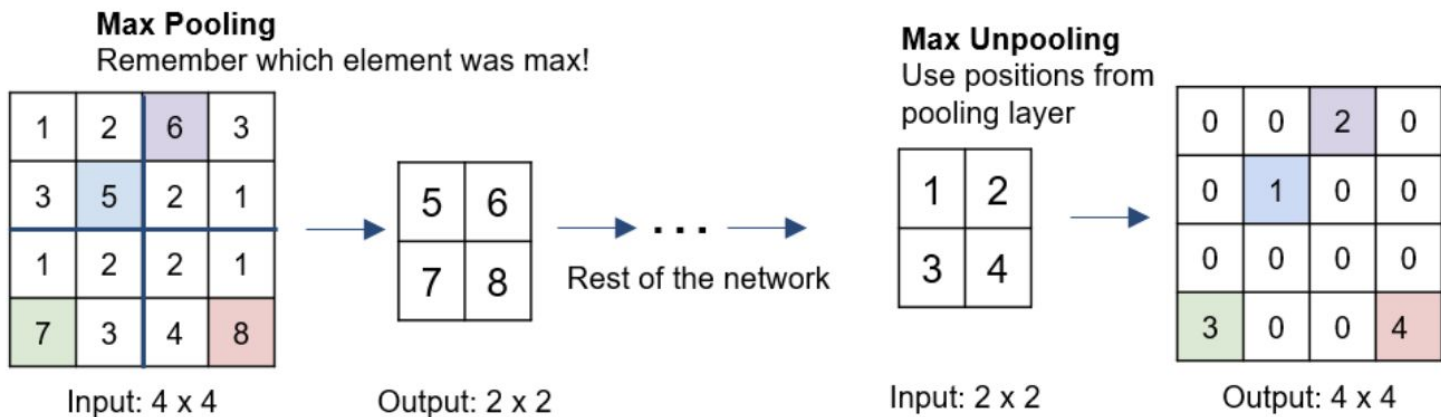


Upsampling

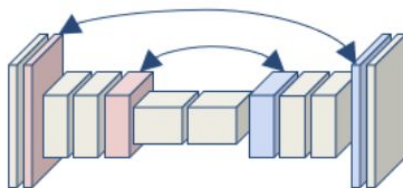
Bilinear

10	20	→	10	12	17	20
30	40		15	17	22	25
			25	27	32	35
			30	32	37	40

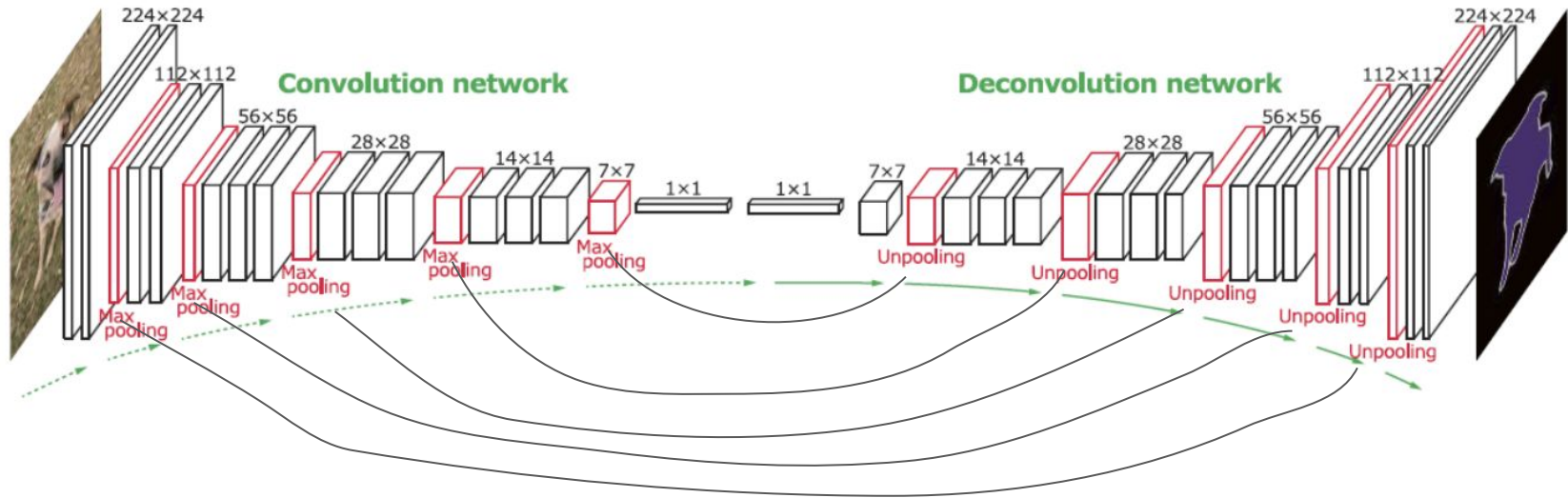
Max-Unpooling



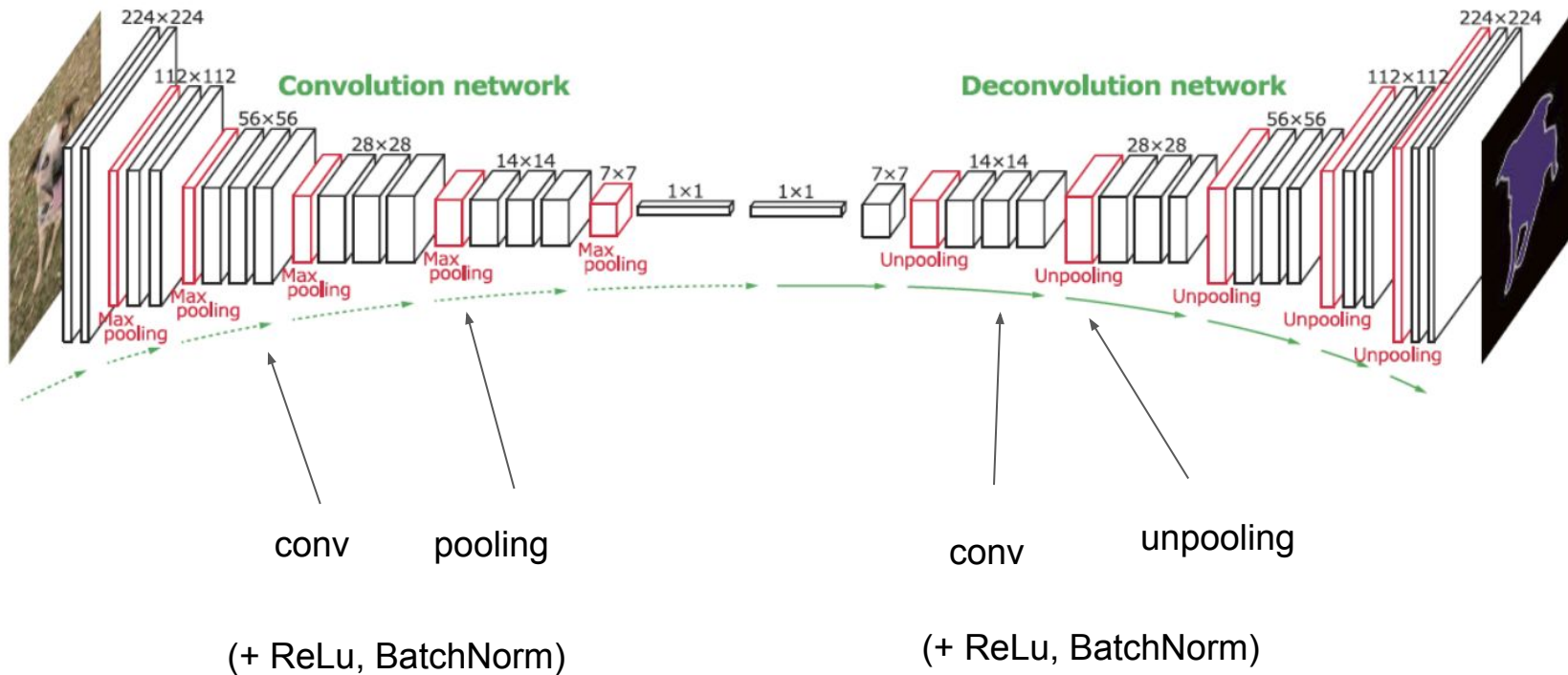
Corresponding pairs of
downsampling and
upsampling layers



Max-Unpooling



DeConvNet

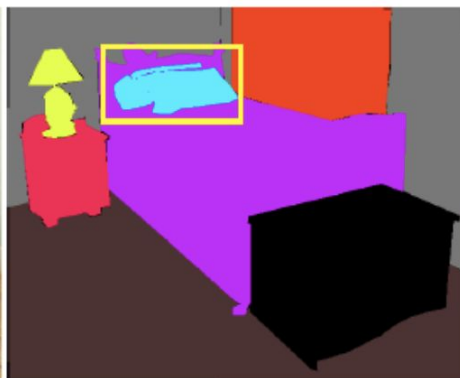


FCN: недостатки

- Upsampling плохо восстанавливает информацию
- **Downsampling** и большой **stride** разрушают пространственную информацию
- Scale Variability



(a) Image



(b) Ground Truth

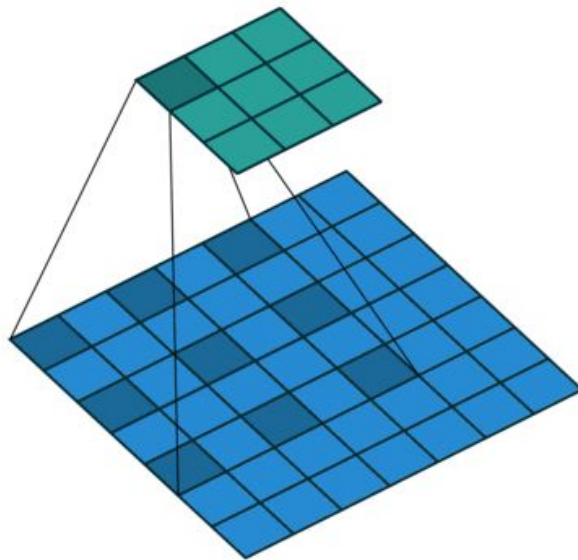


(c) FCN

Dilated Convolutions

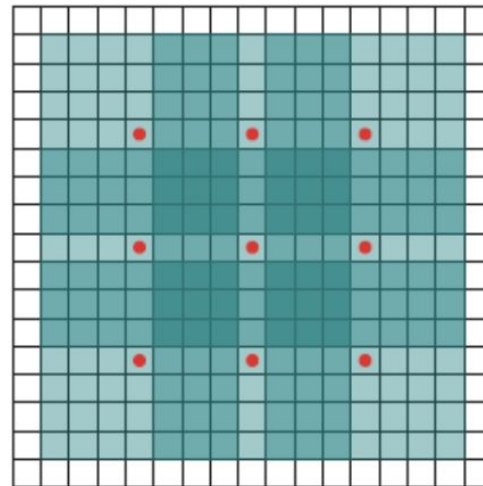
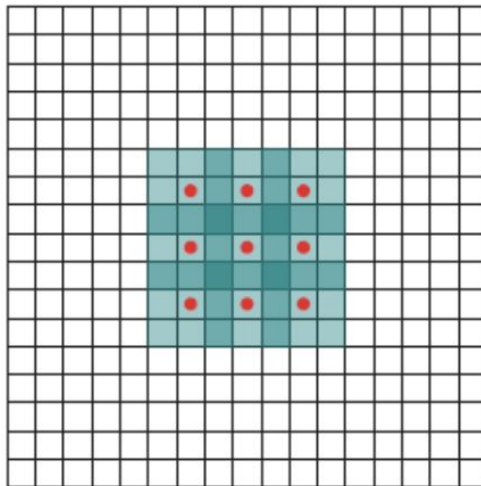
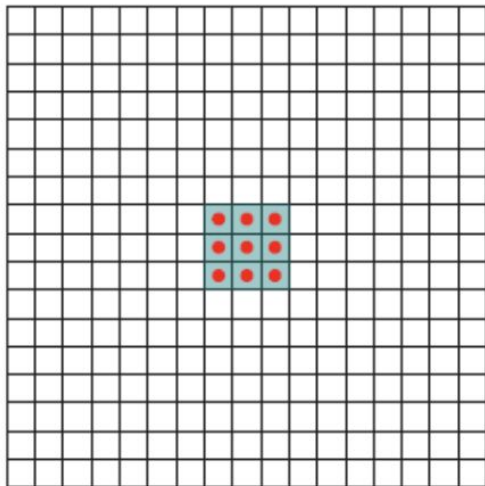
(Atrous convolution) (algorithme à trous)

$$y[i] = \sum_{k=1}^K x[i + r \cdot k] w[k].$$

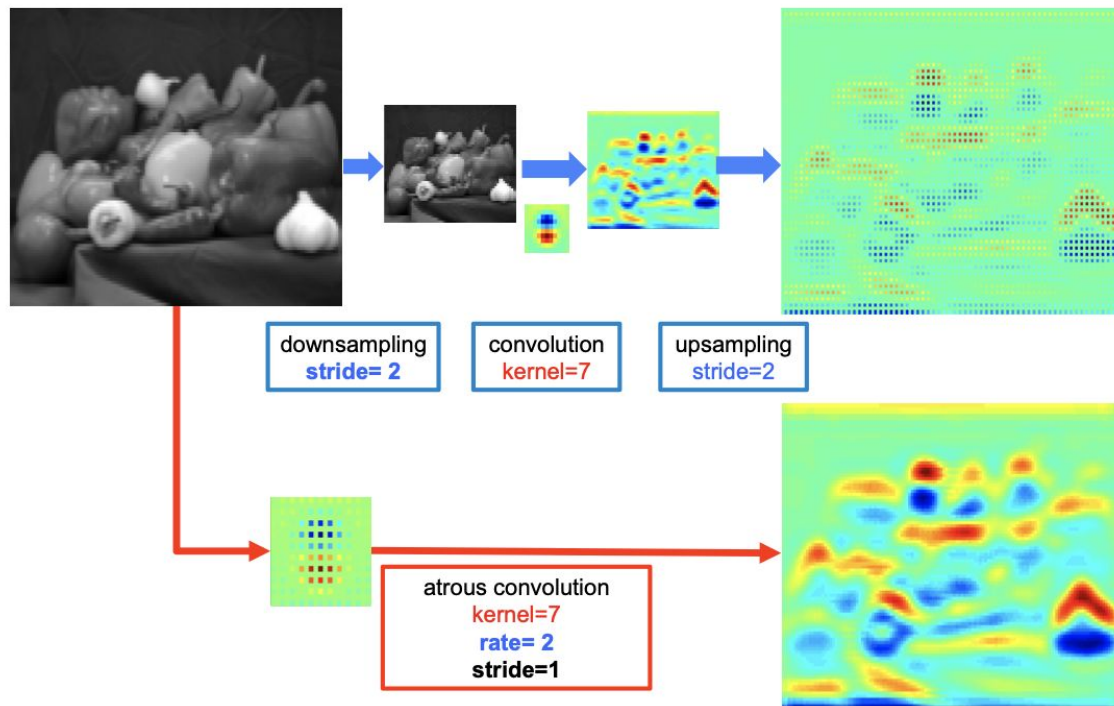


Dilated Convolutions

У Dilated convolution больший receptive field



Dilated Convolutions

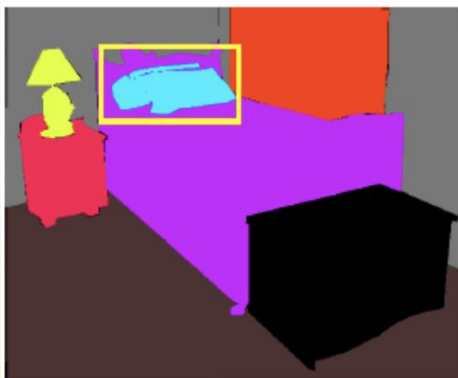


FCN: недостатки

- Upsampling плохо восстанавливает информацию
- Downsampling и большой stride разрушают пространственную информацию
- **Scale Variability**



(a) Image

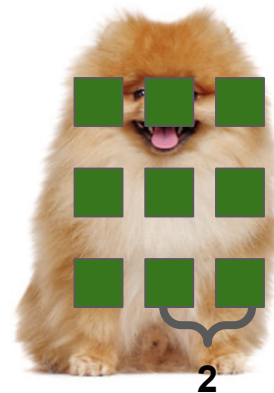
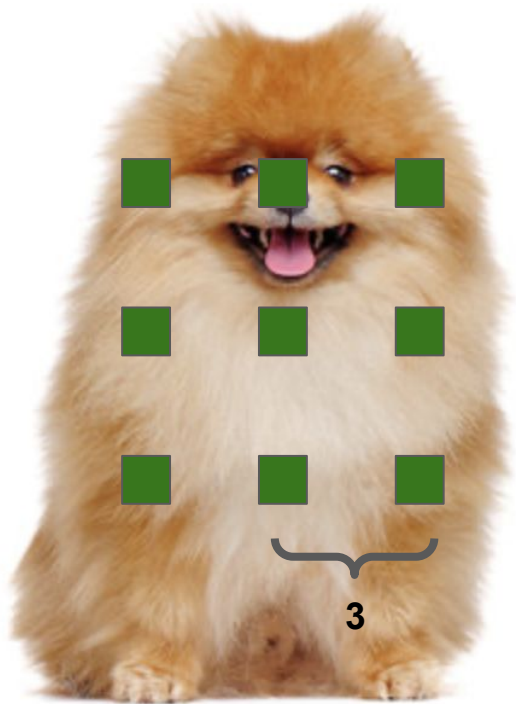


(b) Ground Truth



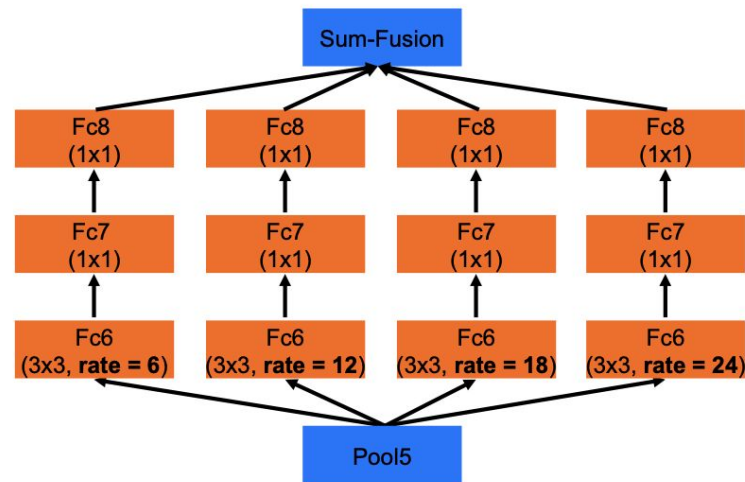
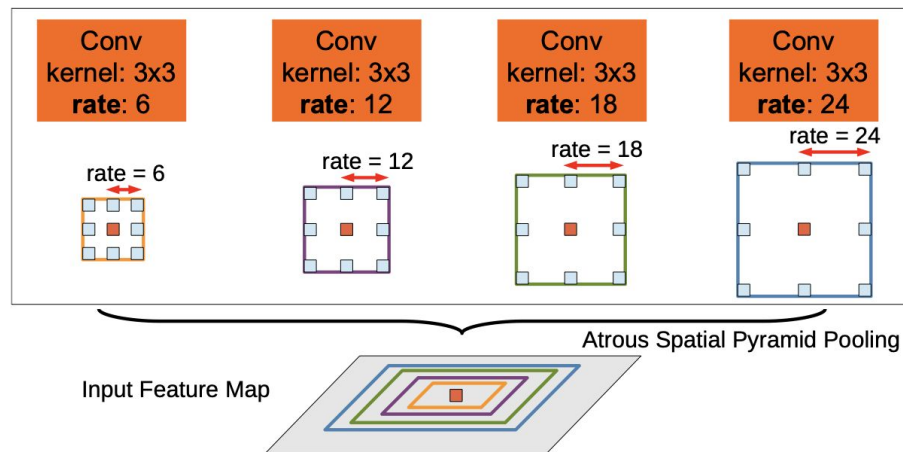
(c) FCN

Multi-scale Context Aggregator



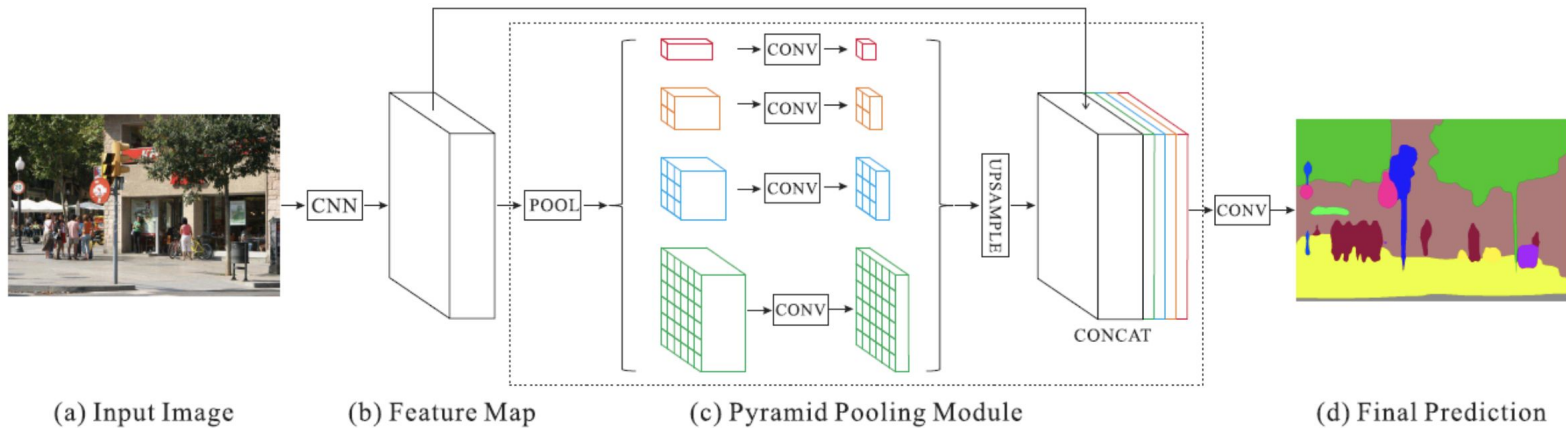
Dilated Convolutions разных размеров
могут выделять информацию об
объектах разных размеров

Multi-scale Context Aggregator



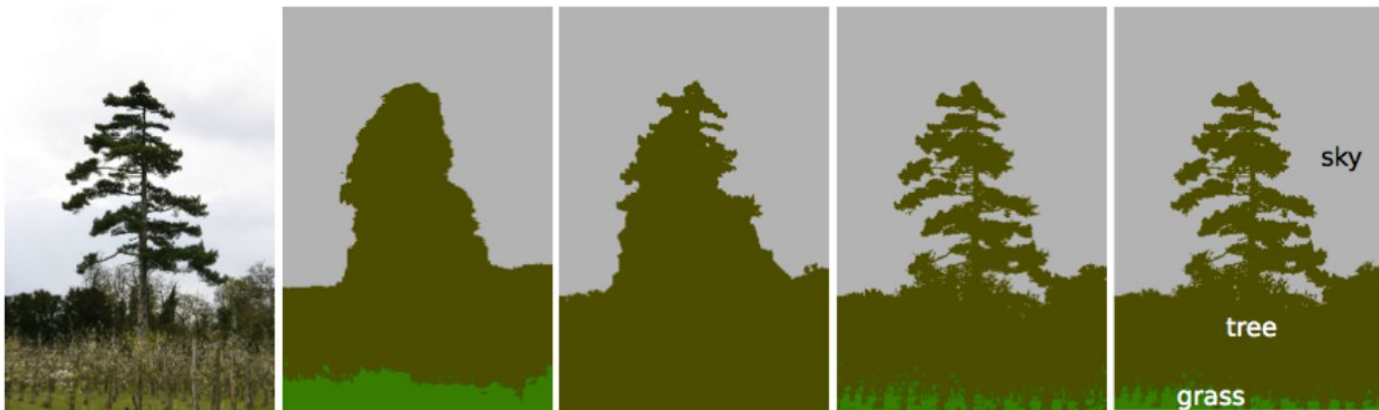
(b) DeepLab-ASPP

Pyramid Pooling Network



Segmentation tricks

1. Deconvolution
2. Dilated Convolutions
3. Multi-scale Context Aggregator
4. Pyramid Pooling Network
5. CRF (Conditional Random Field, postprocessing)



Segmentation tricks

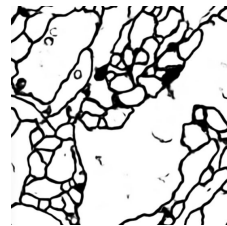
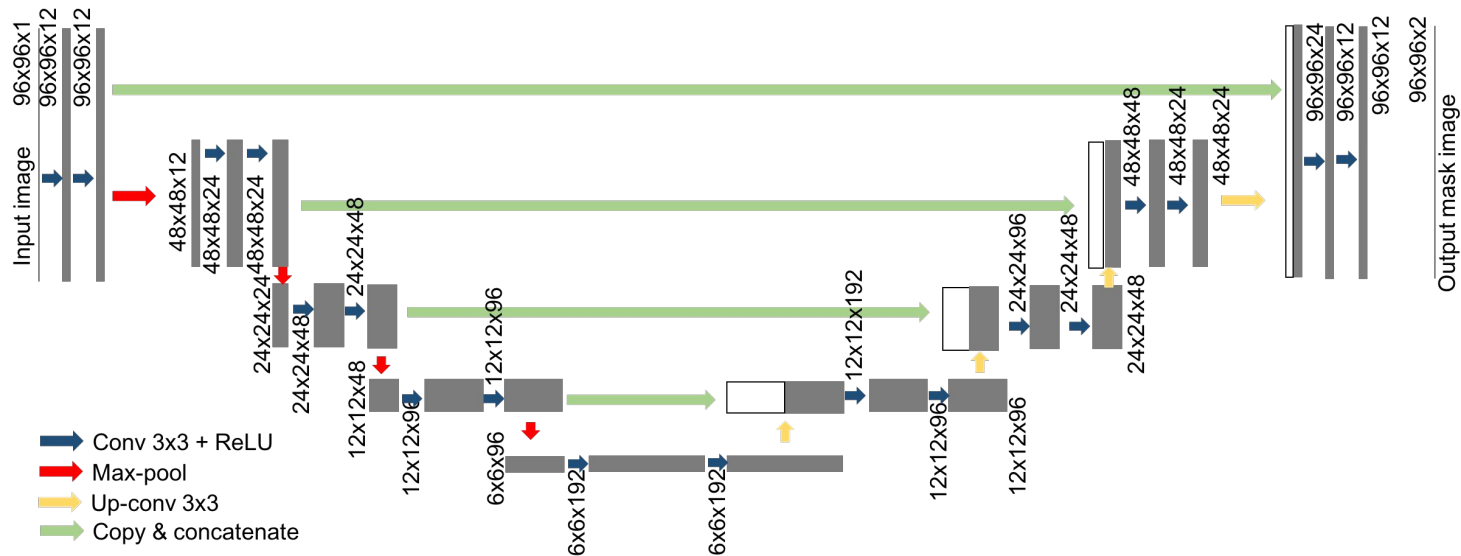
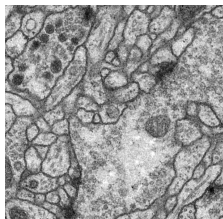
1. Deconvolution
2. Dilated Convolutions
3. Multi-scale Context Aggregator
4. Pyramid Pooling Network
5. CRF (Conditional Random Field, postprocessing)

Segmentation Architectures

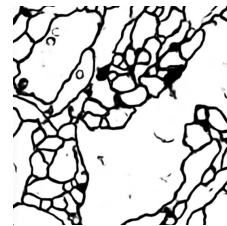
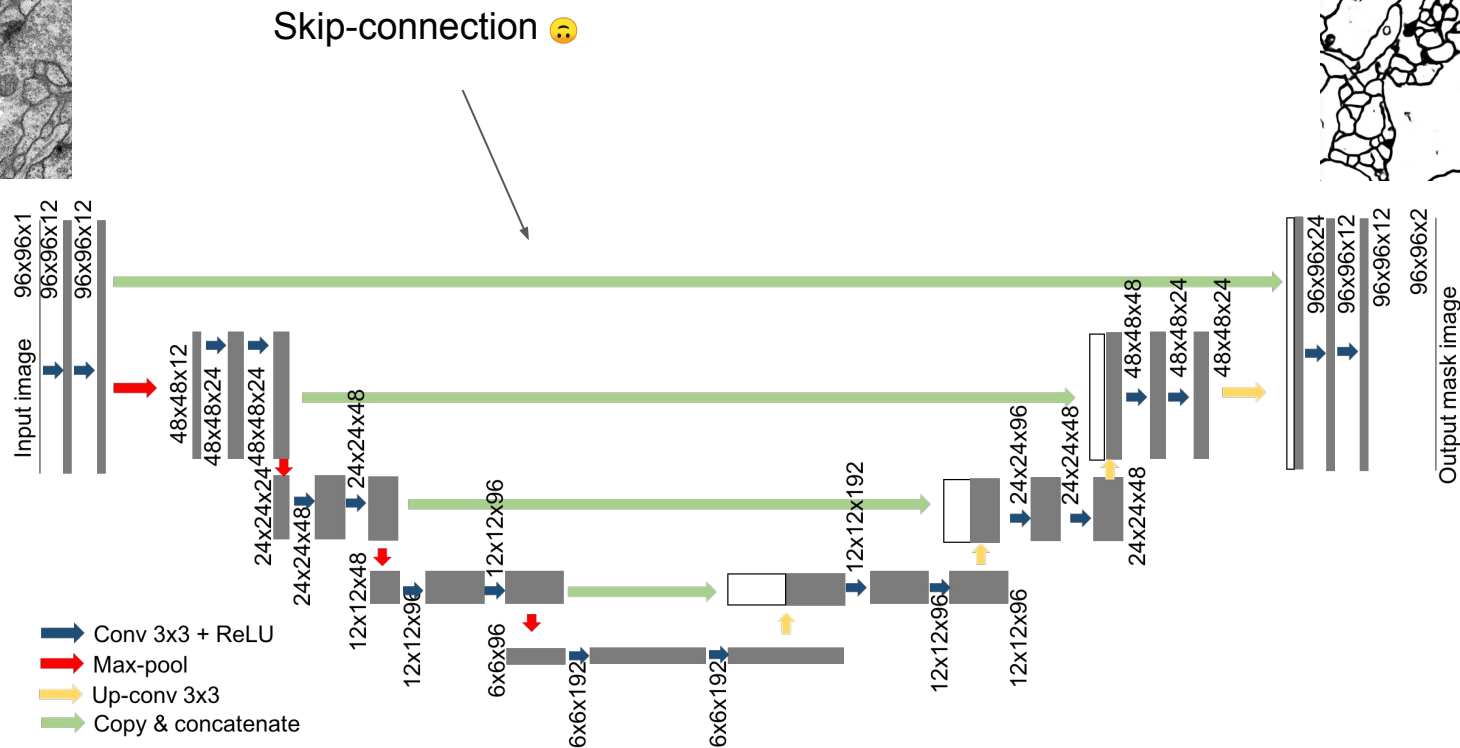
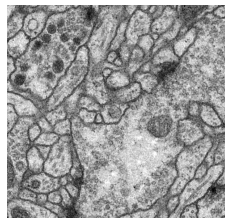
- SegNet (1)
- DeepLab v1 (2, 5)
- DeepLab v2 (2, 3, 5)
- DeepLab v3 (1, 2, 3, 5)
- DeepLab v4 (1, 2, 3, 5)
- PSPNet (1, 2, 4)

Unet

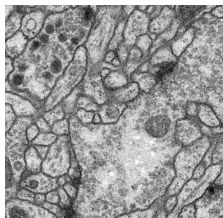
UNet



UNet

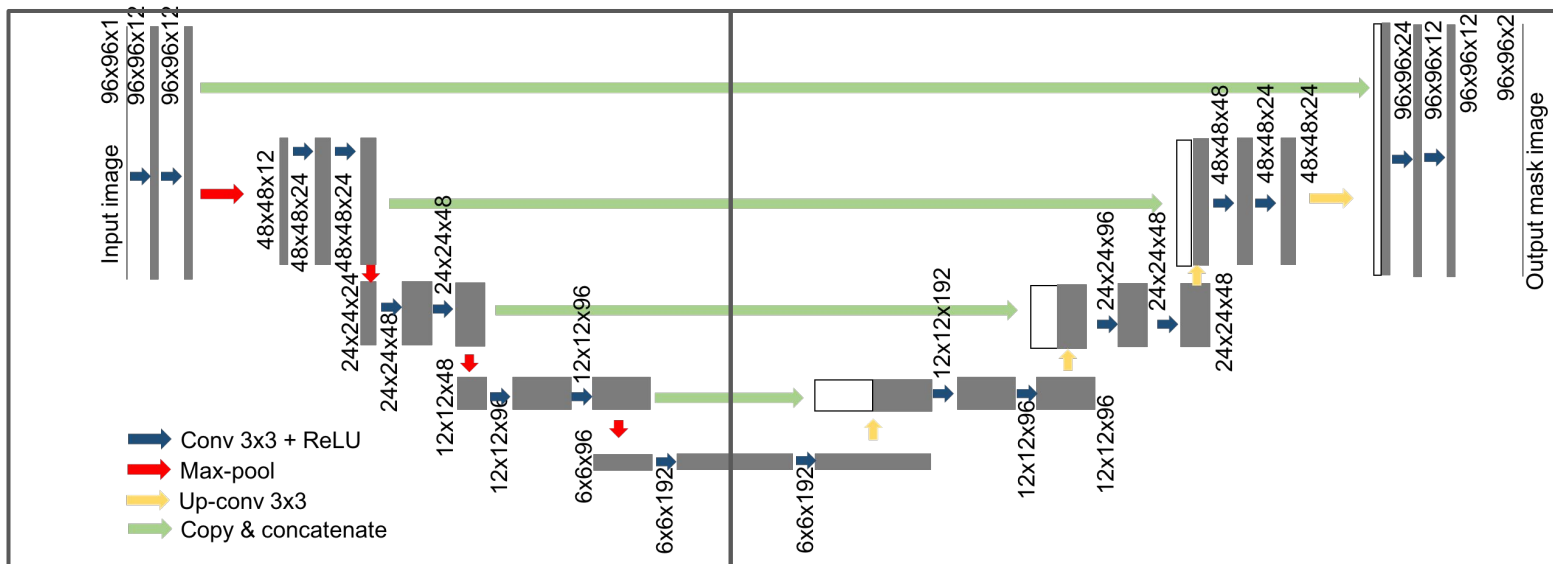
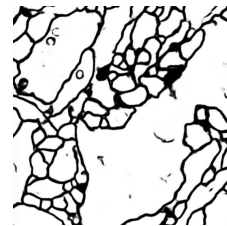


UNet

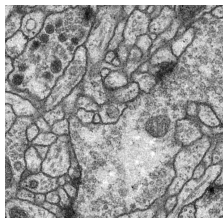


encoder

decoder

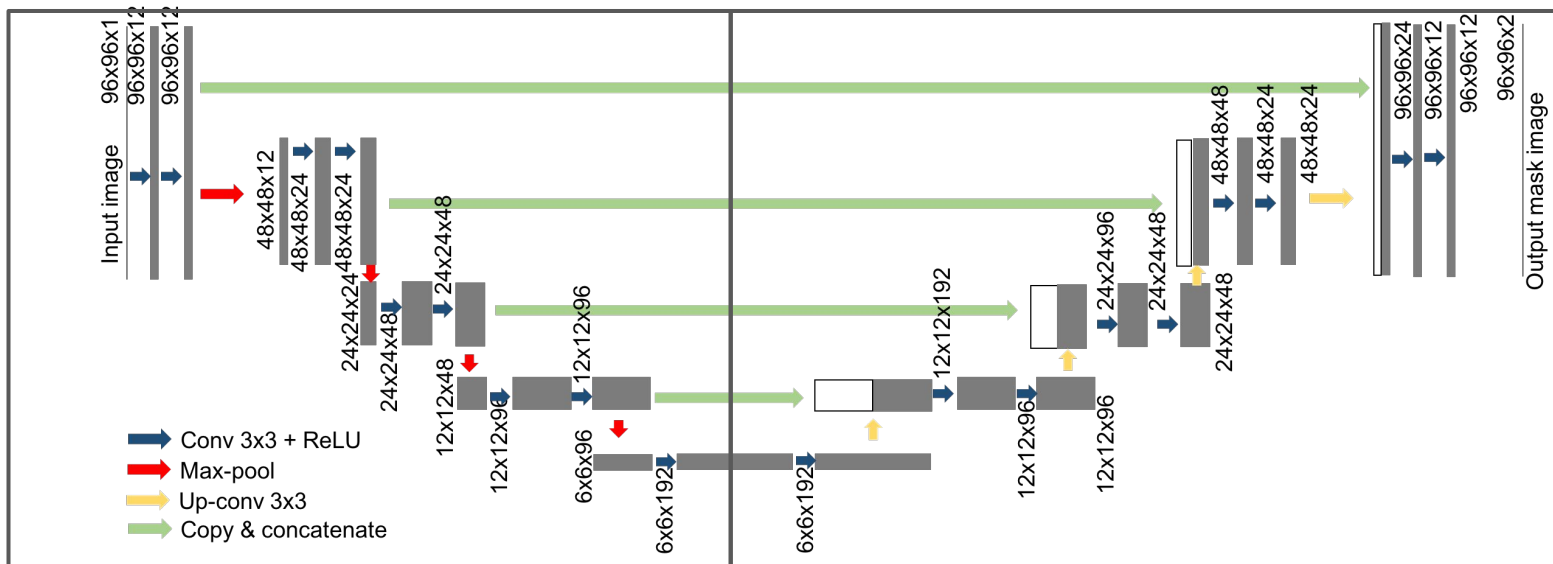
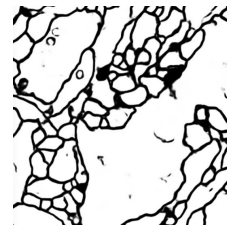


UNet



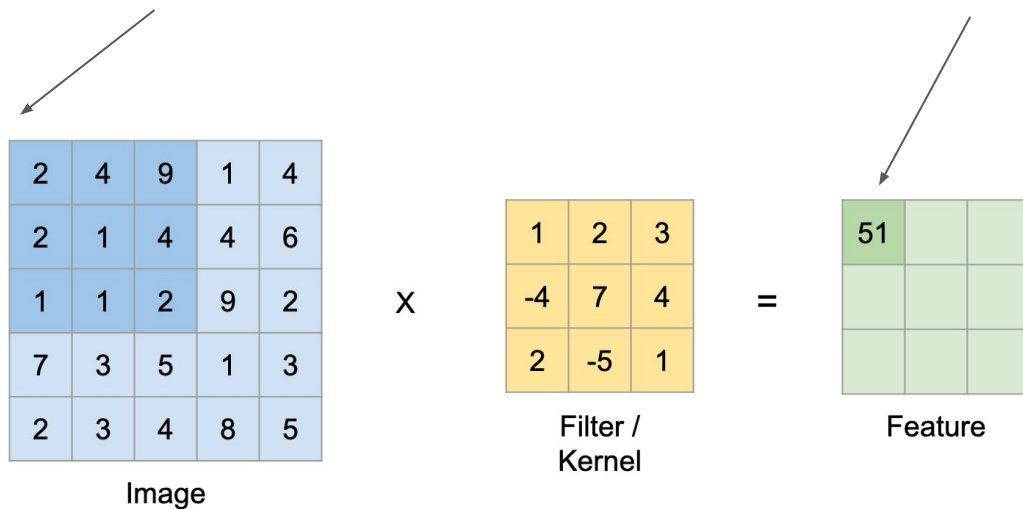
Encoder: extracts
“what” information

Decoder: recovers
“where” information



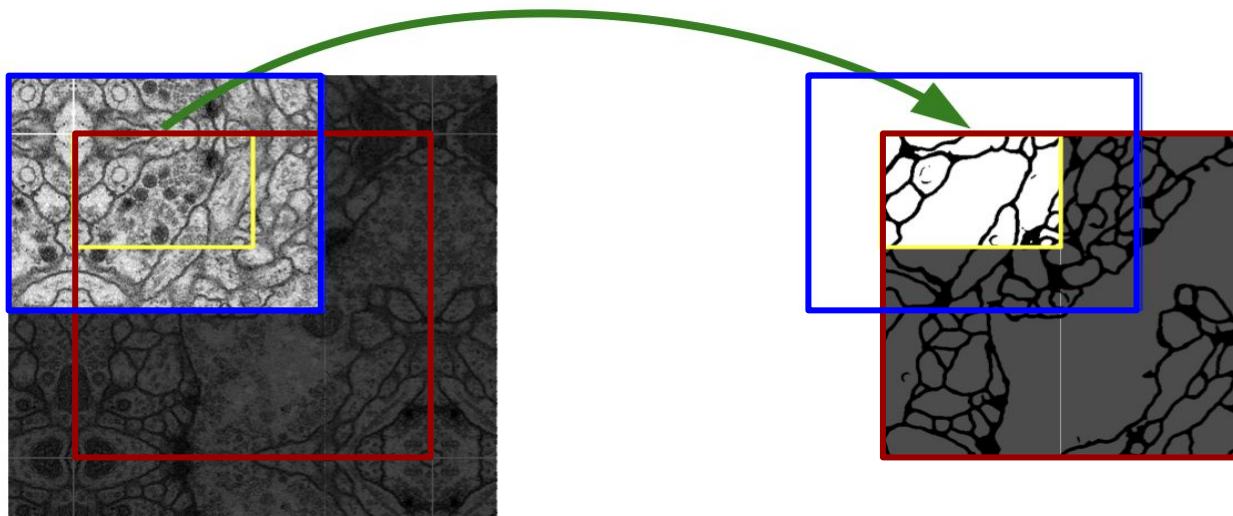
Операция свертки

Информация об этом пикселе содержится только тут



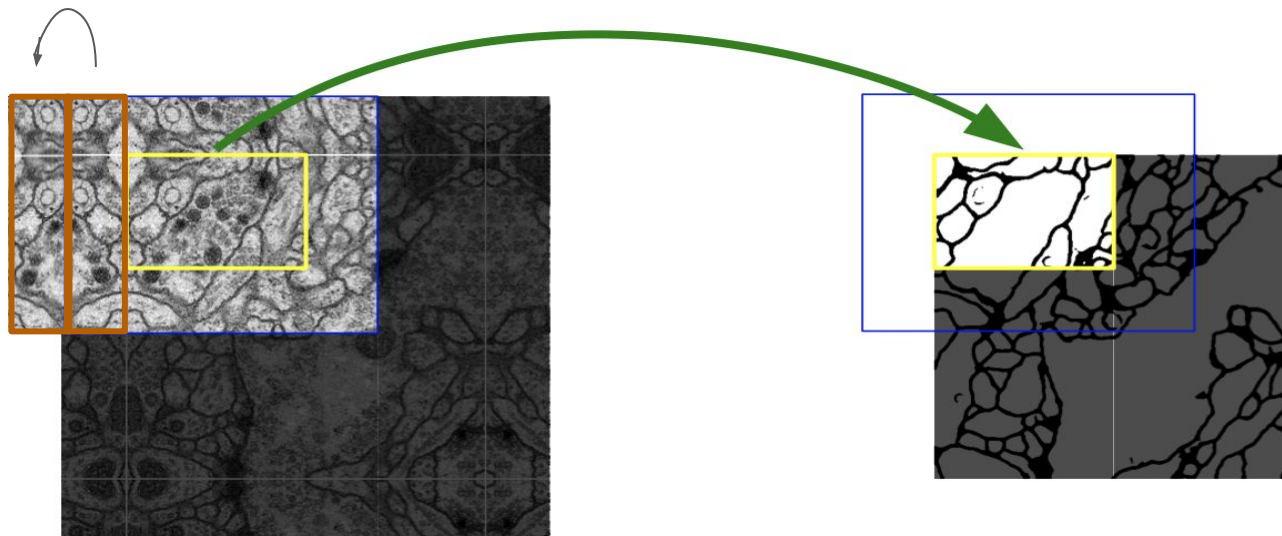
Overlap-tile strategy

Т.к. В UNet используются conv без паддингов, сегментация на выходе может быть получена только для внутренней области изображения



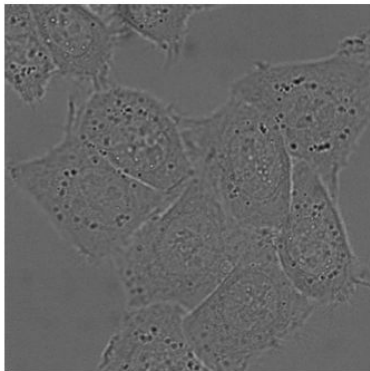
Overlap-tile strategy

Т.к. В UNet используются conv без паддингов, сегментация на выходе может быть получена только для внутренней области изображения

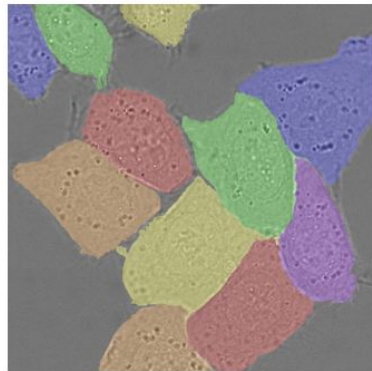


Unet: Loss

Unet uses pixel-wise Softmax loss ...



Input



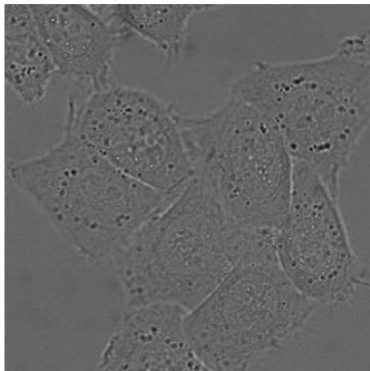
Ground-truth



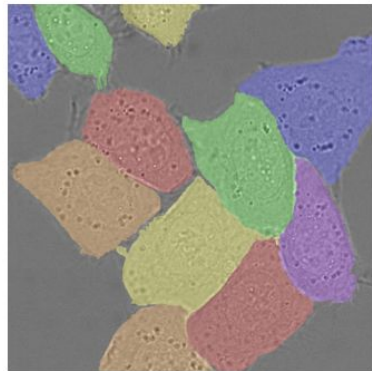
Network output
(what we wanna get)

Unet: Loss

Unet uses pixel-wise Softmax loss ... with weights



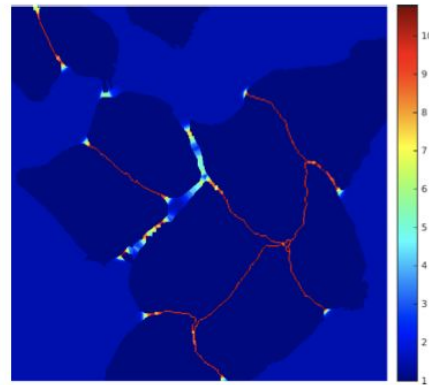
Input



Ground-truth



Network output
(what we wanna get)



Segmentation tricks

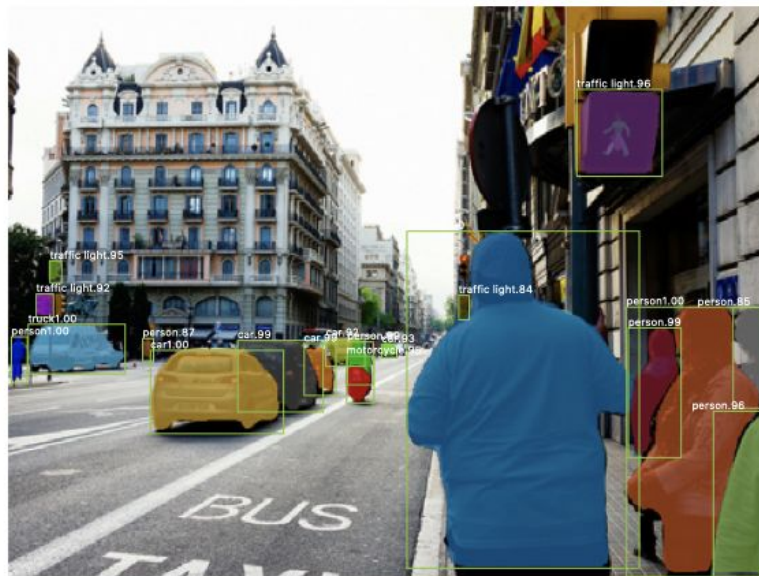
1. Deconvolution
2. Dilated Convolutions
3. Multi-scale Context Aggregator
4. Pyramid Pooling Network
5. CRF (Conditional Random Field, postprocessing)
6. *Skip-connection*

Segmentation Architectures

- SegNet (1)
- DeepLab v1 (2, 5)
- DeepLab v2 (2, 3, 5)
- DeepLab v3 (1, 2, 3, 5)
- DeepLab v4 (1, 2, 3, 5)
- PSPNet (1, 2, 4)
- *UNet (1, 6)*

Other Approaches

Mask R-CNN (<https://arxiv.org/pdf/1703.06870.pdf>)





<https://www.youtube.com/watch?v=ATlcEDSPWXY>



<https://www.youtube.com/watch?v=QYIQbfnS9jA&feature=youtu.be&t=127>