Semantic segmentation

Inputs

- RGB image

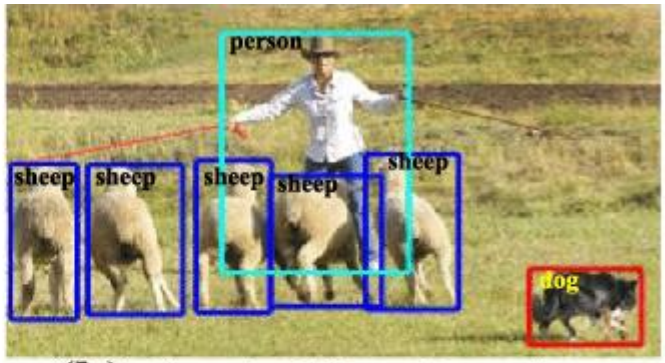
Targets

- Class label for every pixel

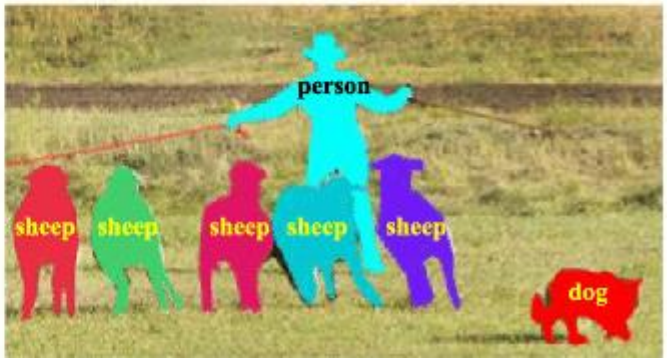

30 FPS

Instance Segmentation

(a) Object Classification



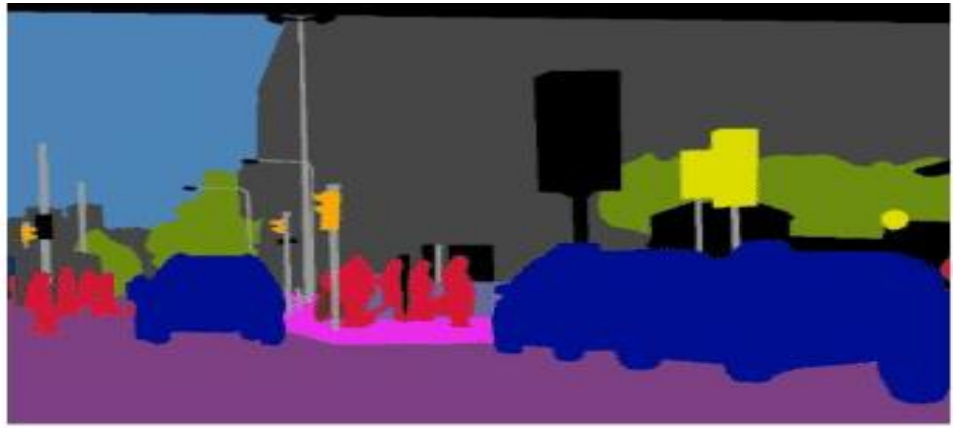
(b) Generic Object Detection (Bounding Box)



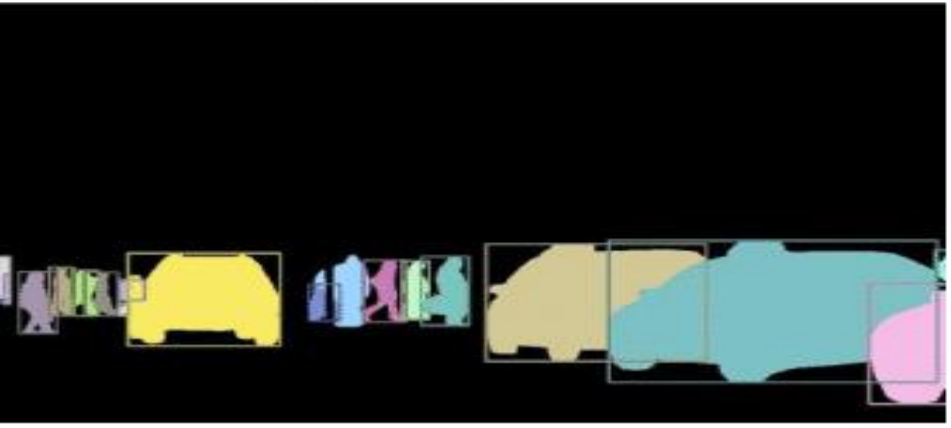
Panoptic Segmentation


person
car
motorcycle

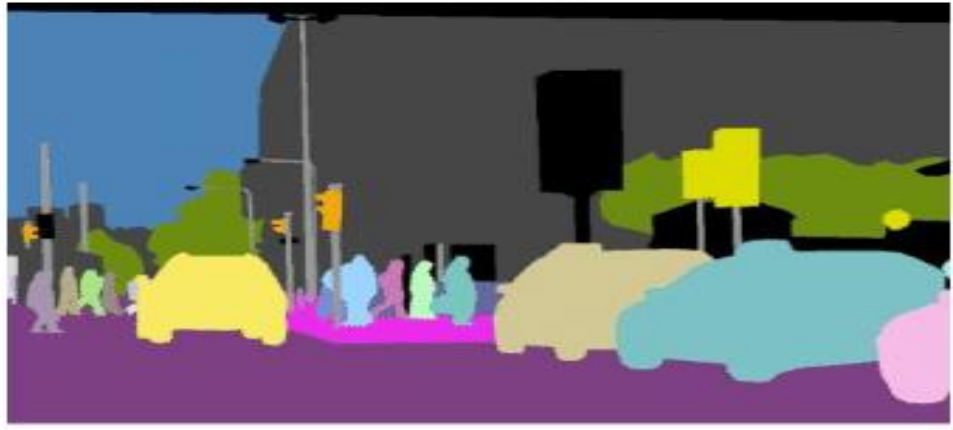
(a) Image



(b) Semantic Segmentation



(c) Instance Segmentation



(d) Panoptic Segmentation

Semantic segmentation

как добиться такого предикта? как сгенерировать выход с такого же размера, как и вход?

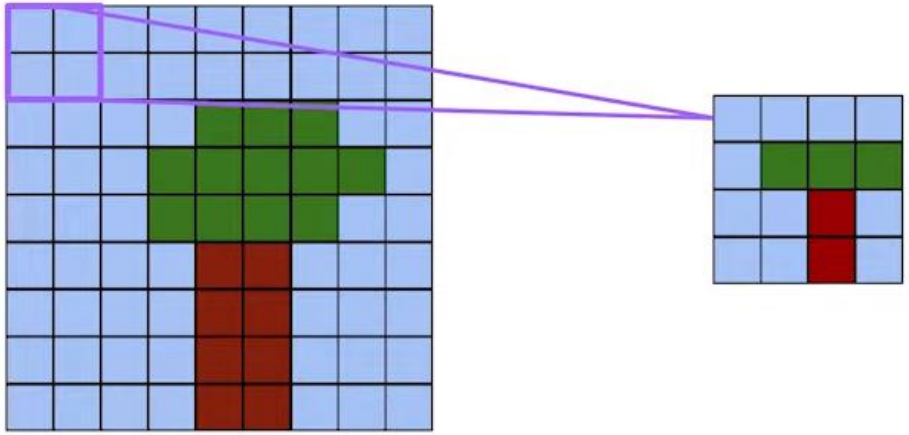
Inputs

- RGB image

Targets

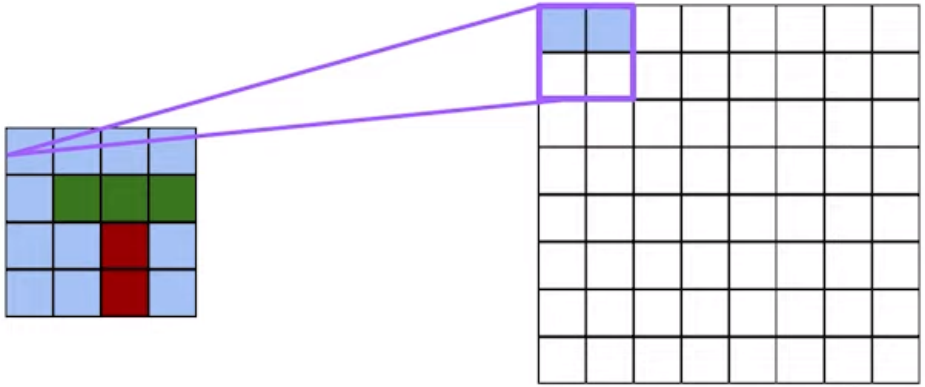
- Class label for every pixel

Recap: Pooling



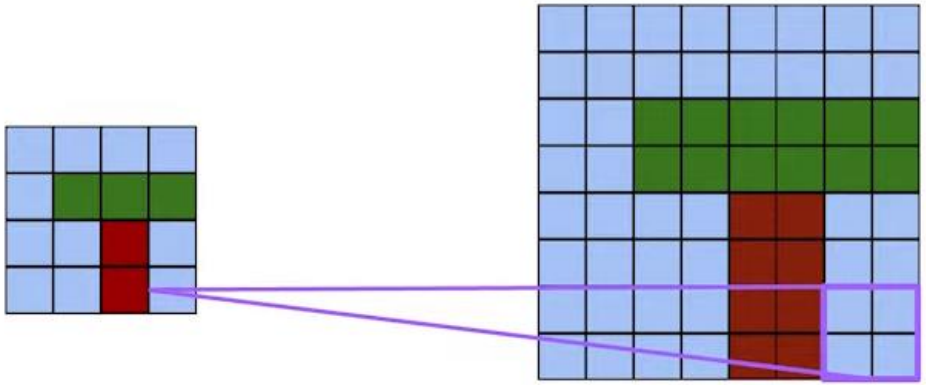
Pooling: compute mean or max over small windows to reduce resolution.

Unpooling



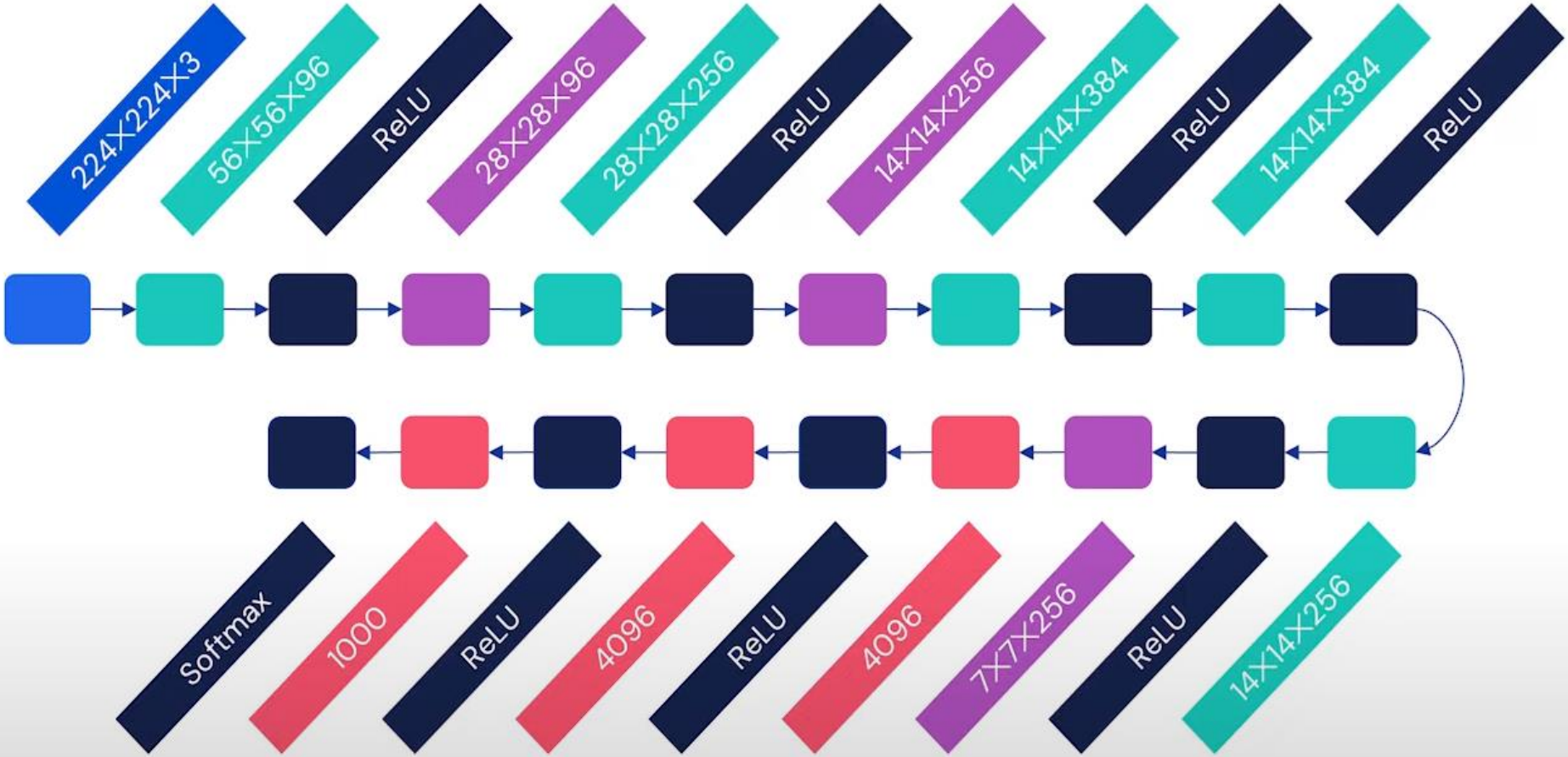
Unpooling: upsample to increase resolution; here 2x2 kernel.

Unpooling



Unpooling: upsample to increase resolution; here 2x2 kernel.

AlexNet (2012)



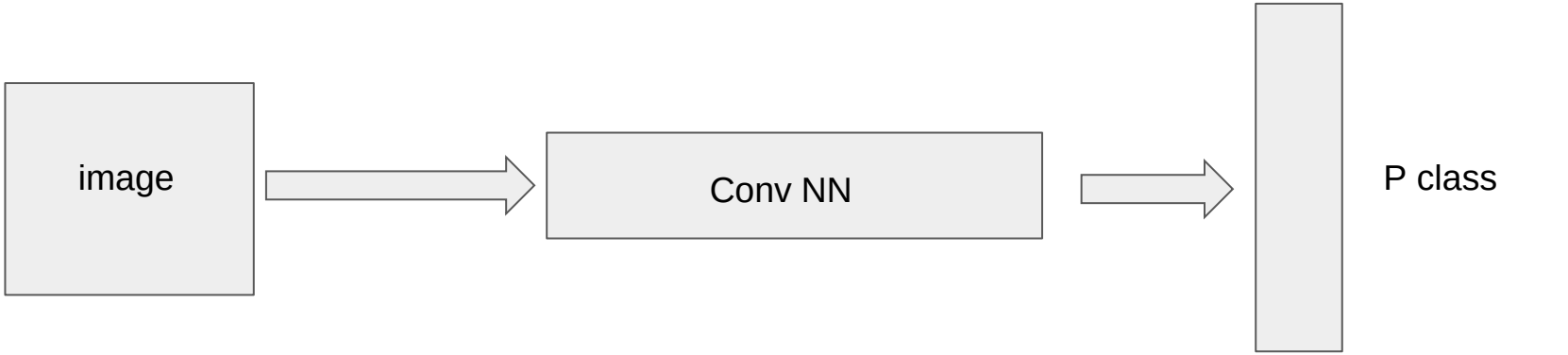


image
Conv NN
P class

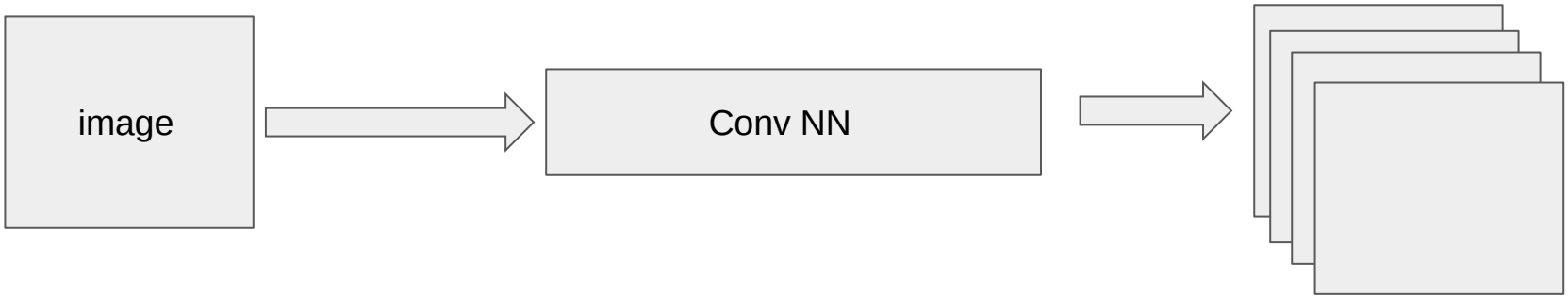
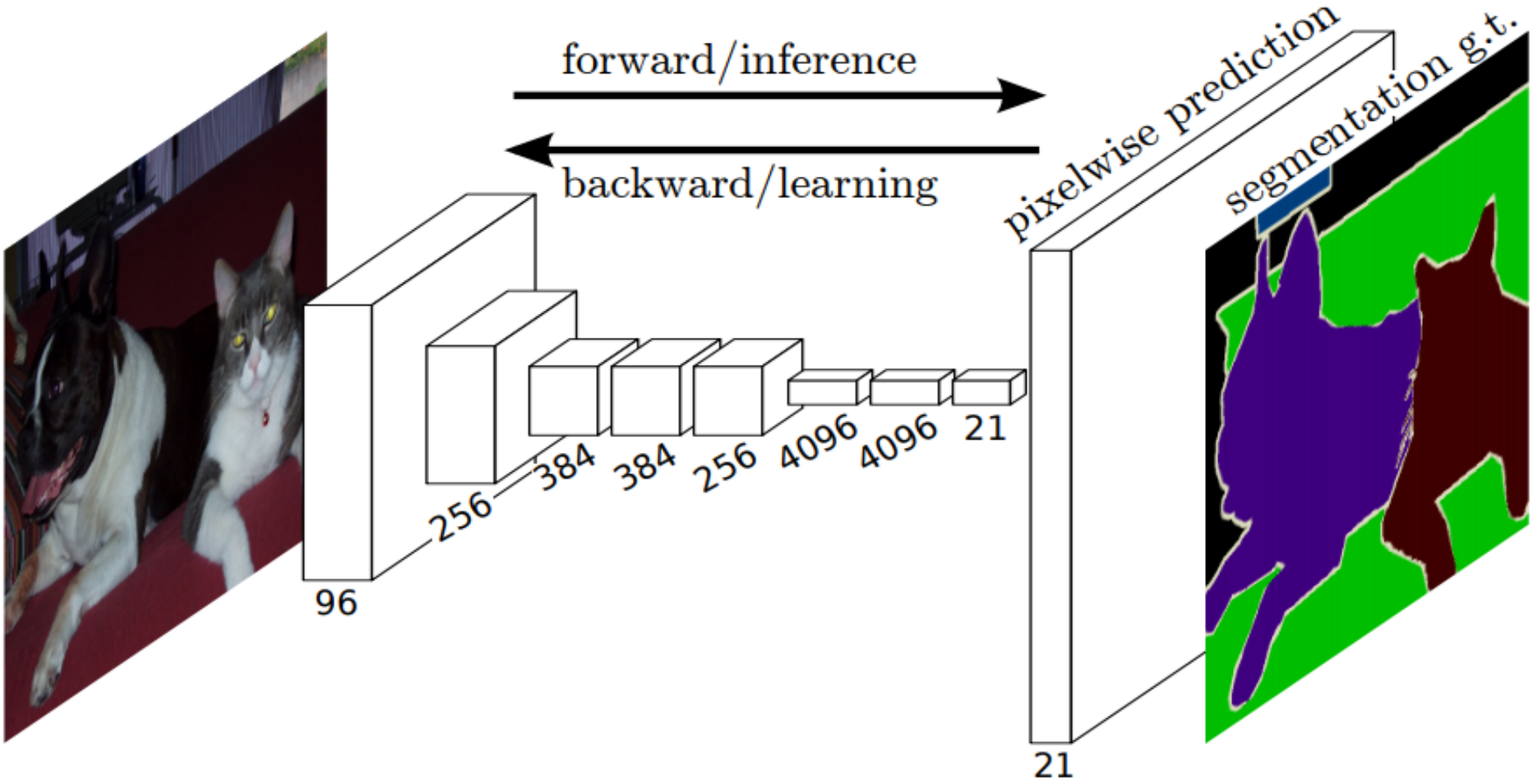
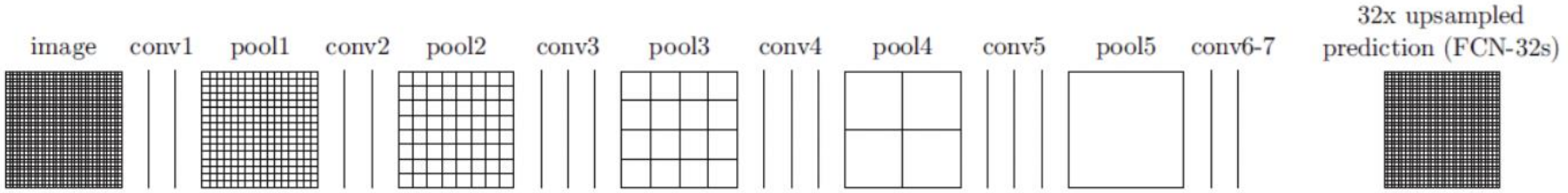


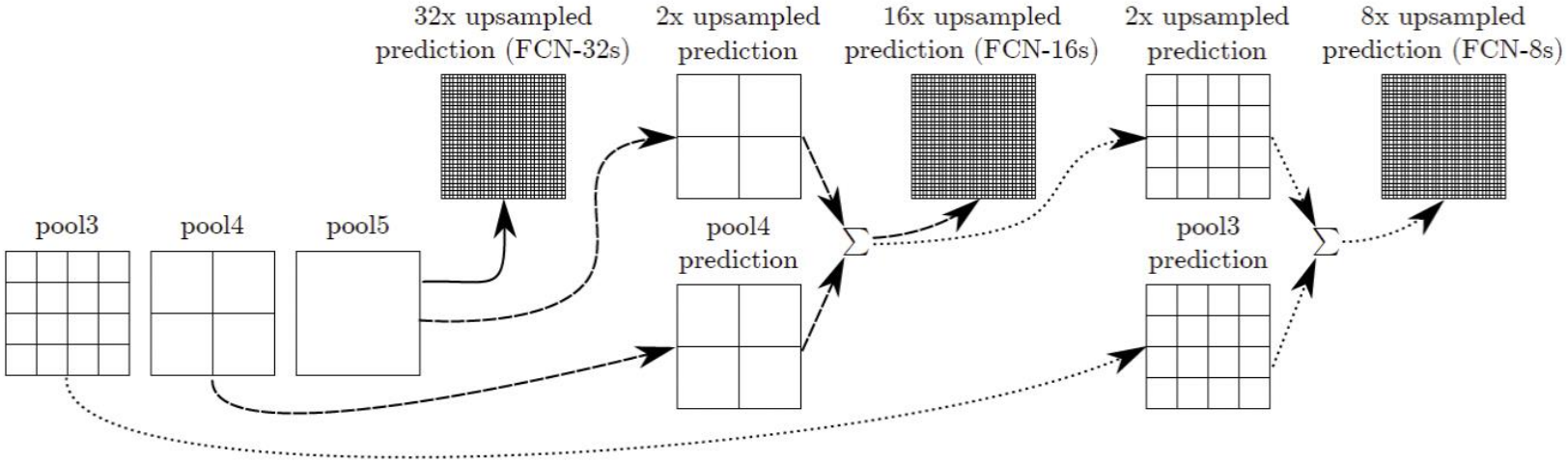
image
Conv NN

Fully Convolutional Networks for Semantic Segmentation

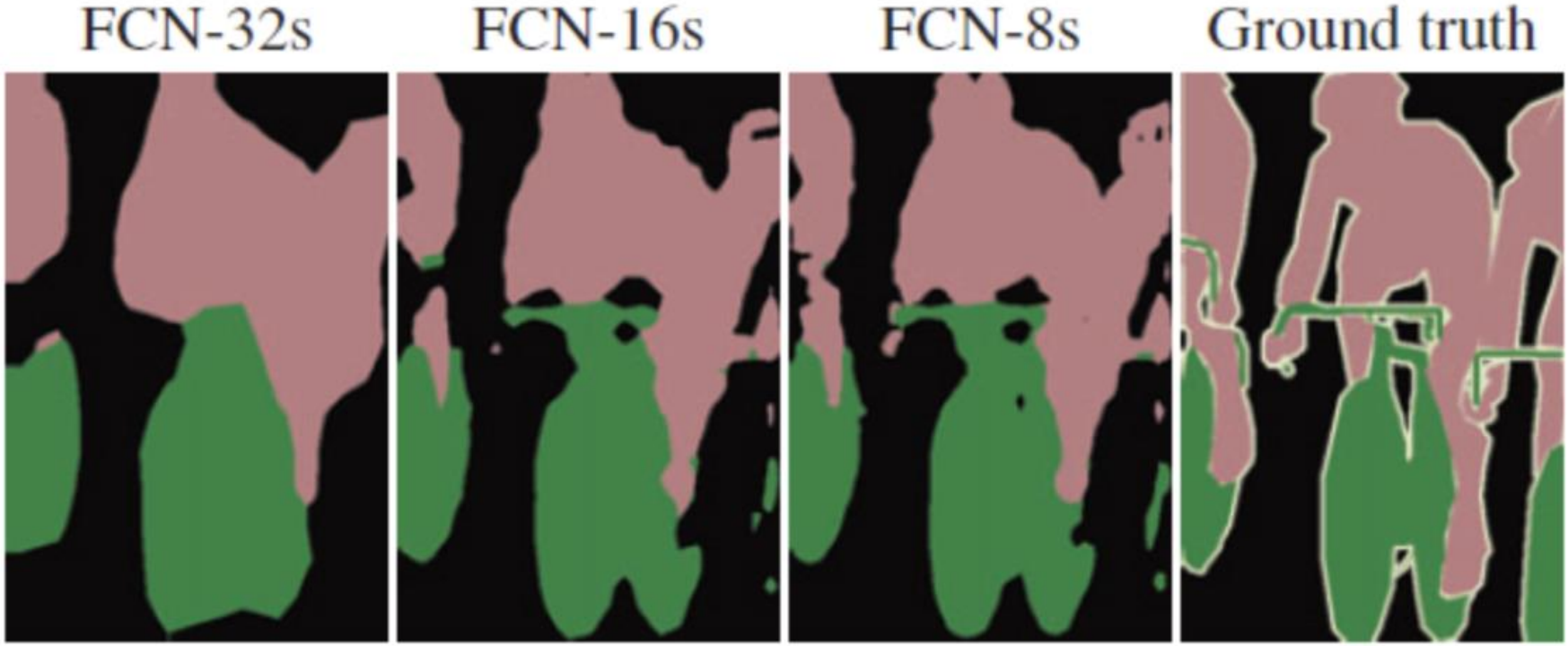




FCN-32s

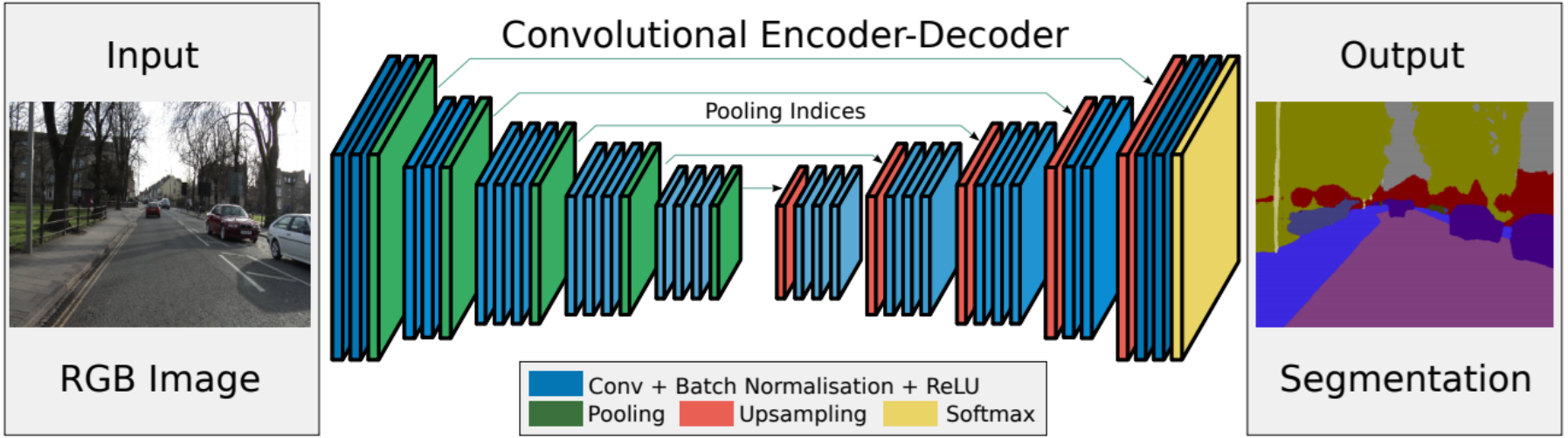


Fusing for FCN-16s and FCN-8s

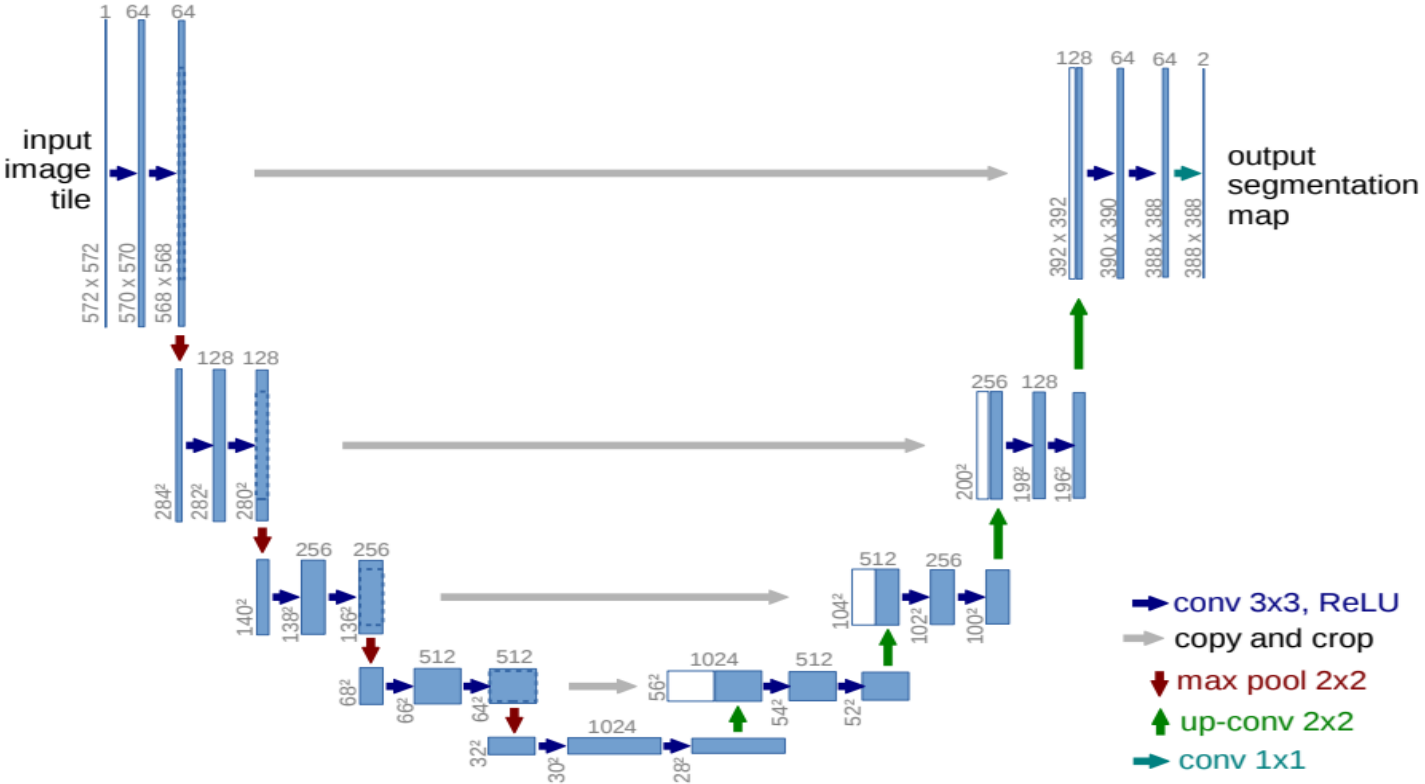


Comparison with different FCNs

SegNet


Input
RGB Image
Convolutional Encoder-Decoder
Pooling Indices
Conv + Batch Normalisation + ReLU
Pooling
Upsampling
Softmax
Output
Segmentation

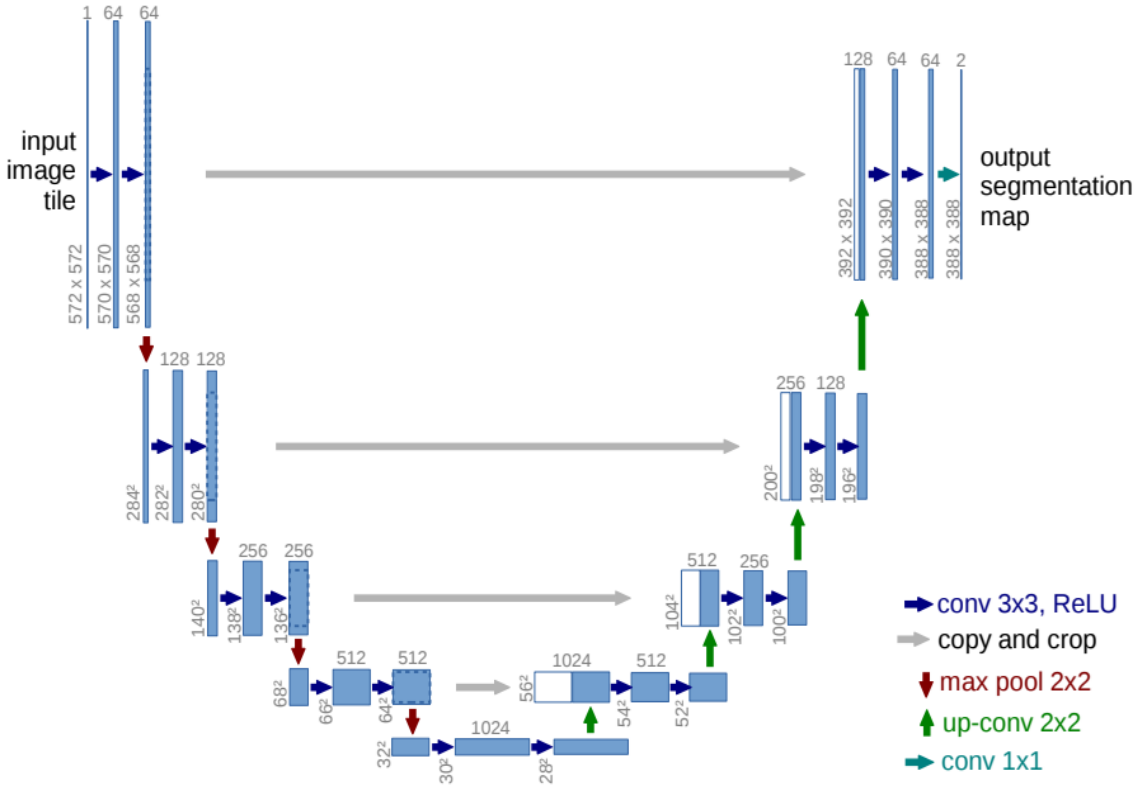
U-net

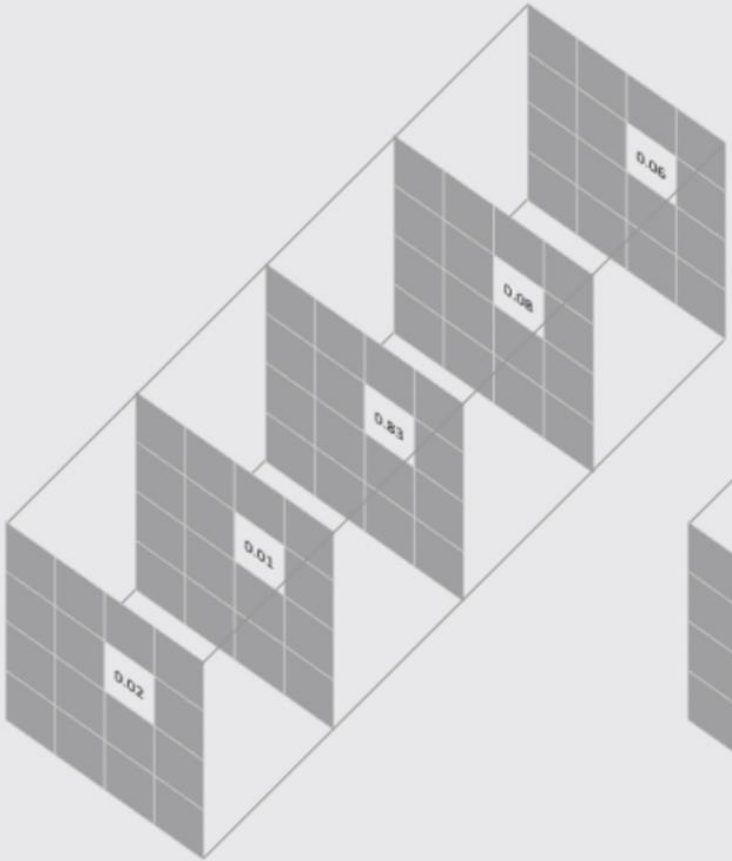

input image tile
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

- Output $H \times W \times N_{\text{classes}}$

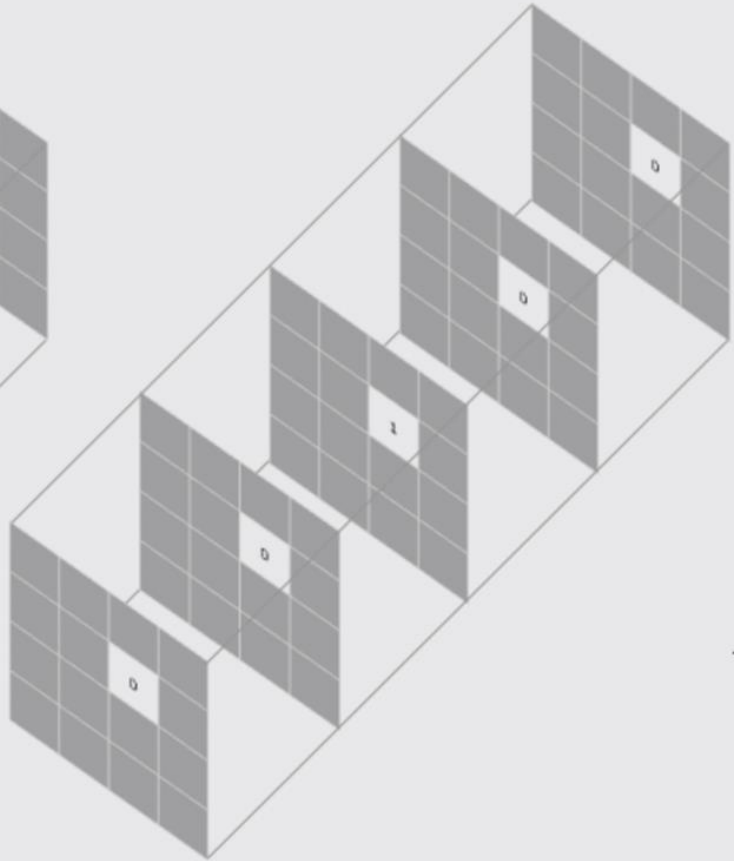
- Loss: pixel-wise cross entropy

$$\ell_{\text{CE}}(\mathbf{p}, \mathbf{t}) = -\frac{1}{HW} \sum_{i=1}^{HW} \sum_{j=1}^{N_{\text{classes}}} \mathbf{t}_{ij} \log \mathbf{p}_{ij}$$





Prediction for a selected pixel



Target for the corresponding pixel

Pixel-wise loss is calculated as the log loss, summed over all possible classes

$$-\sum_{classes} y_{true} \log\left(y_{pred}\right)$$

This scoring is repeated over all **pixels** and averaged

Classification

- **Accuracy**: percentage of correct predictions

Top-1: top prediction is the correct class

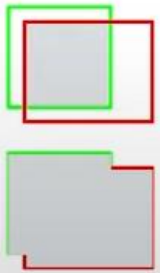
Top-5: correct class is in top-5 predictions

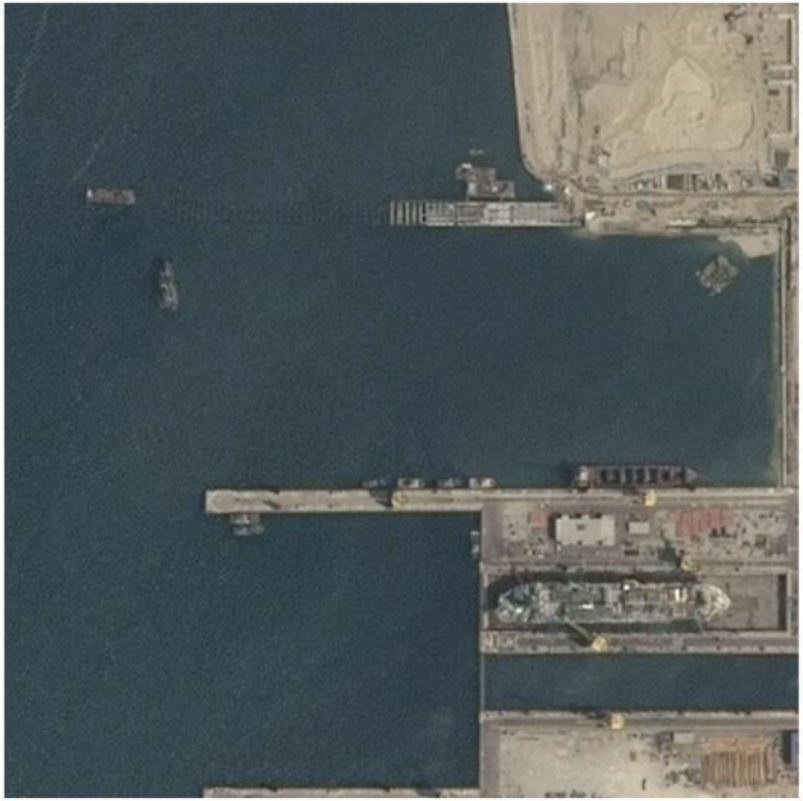
Object detection and segmentation

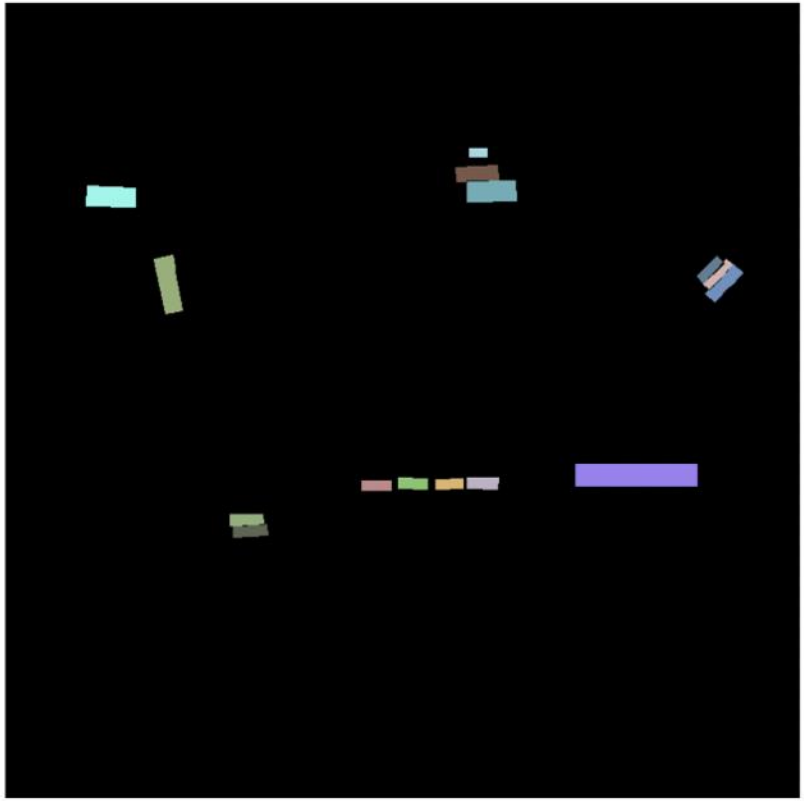
- intersection-over-union (**IoU**)

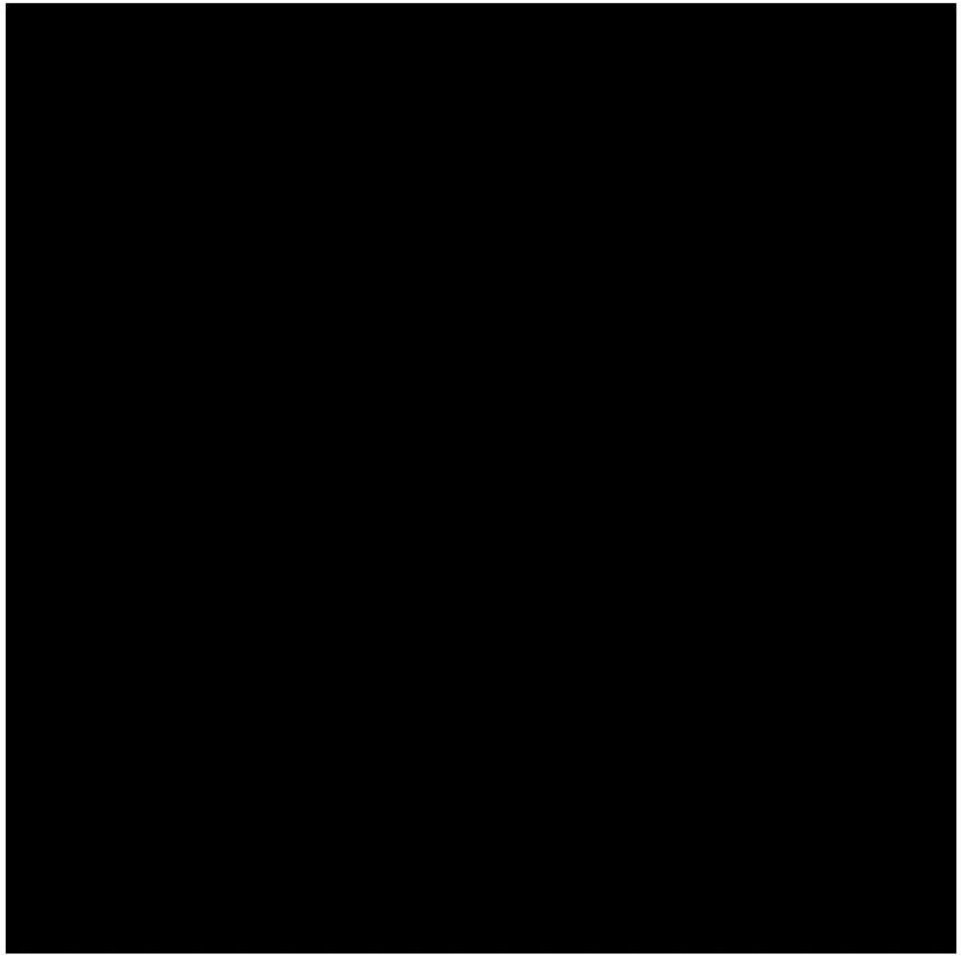
  non-differentiable: used only for evaluation

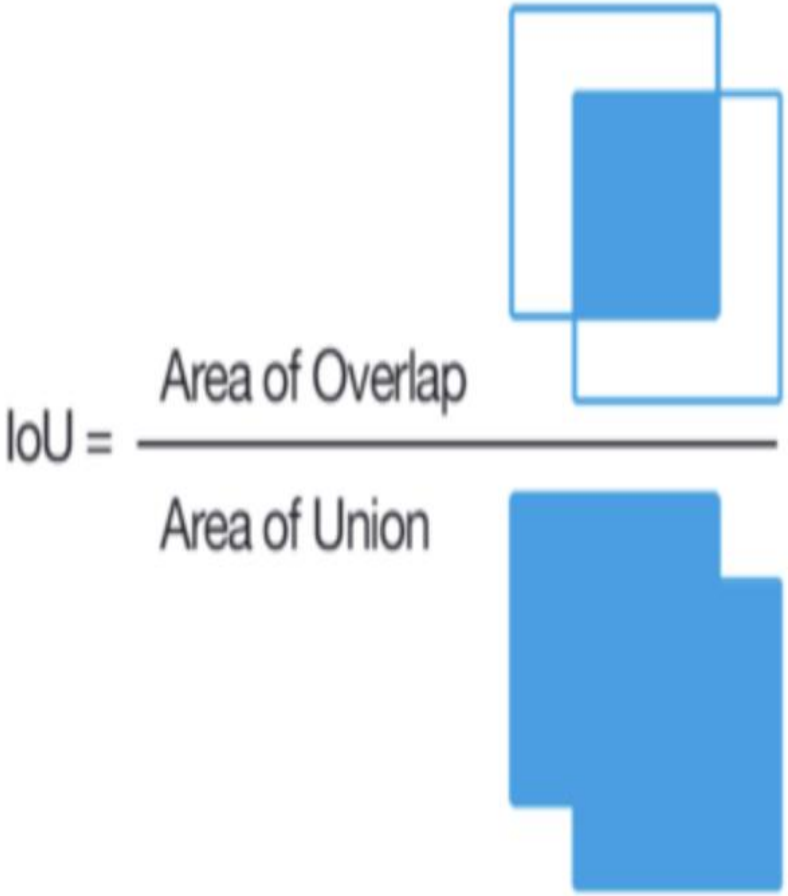
$$\mathcal{J}(\mathbf{P}, \mathbf{T}) = \frac{\mathbf{P} \bigcap \mathbf{T}}{\mathbf{P} \bigcup \mathbf{T}}$$



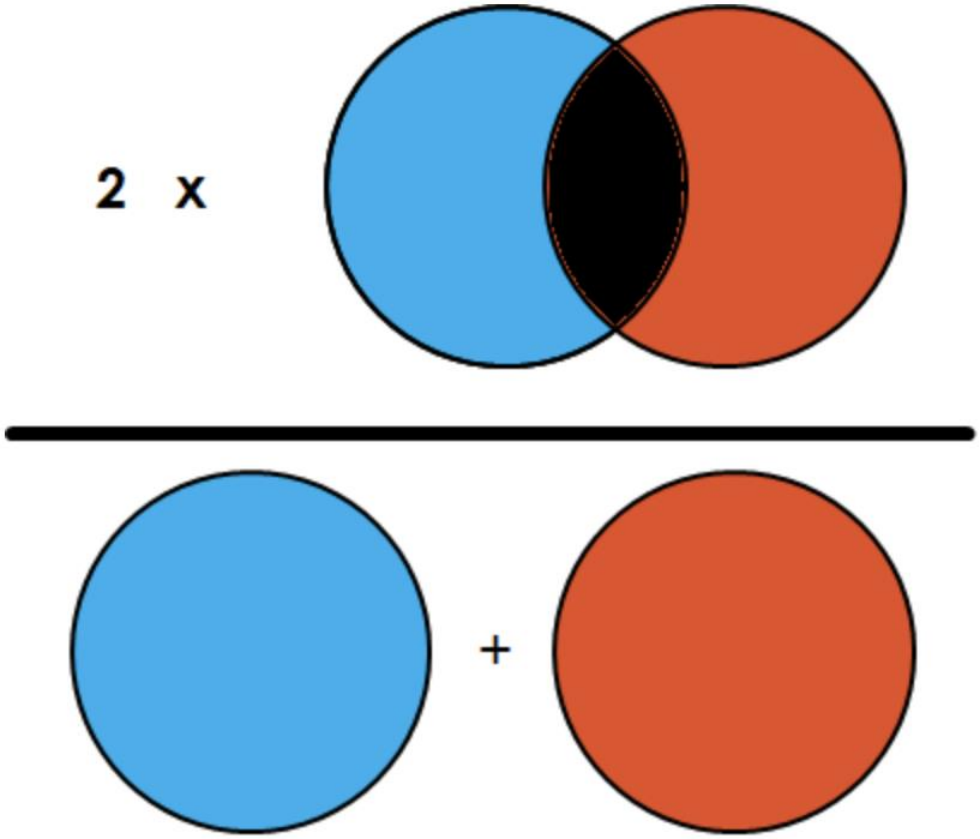








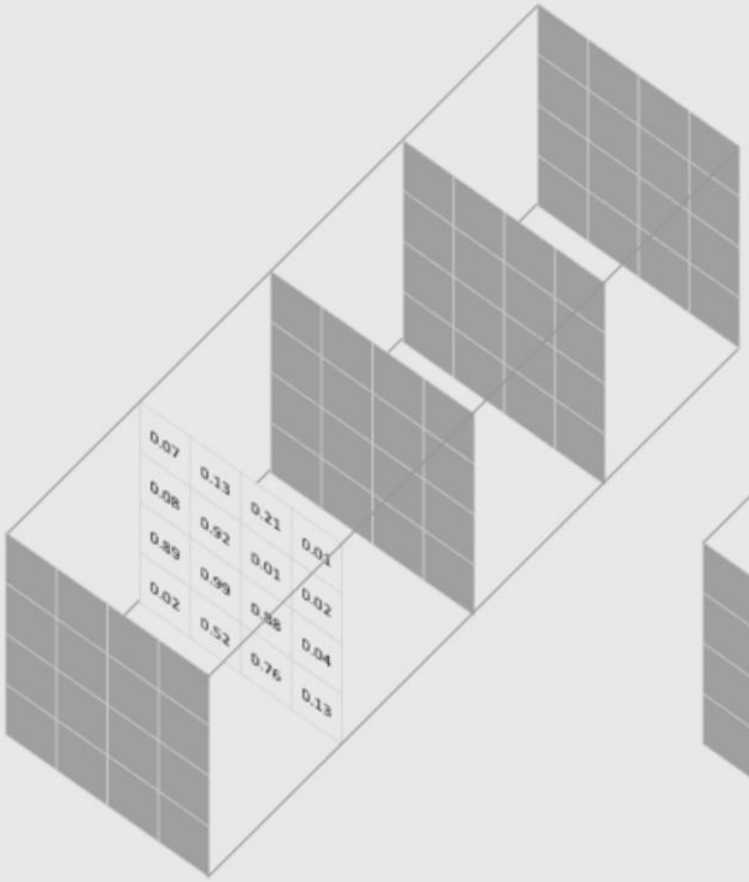
$$\frac{TP}{TP + FP + FN}$$



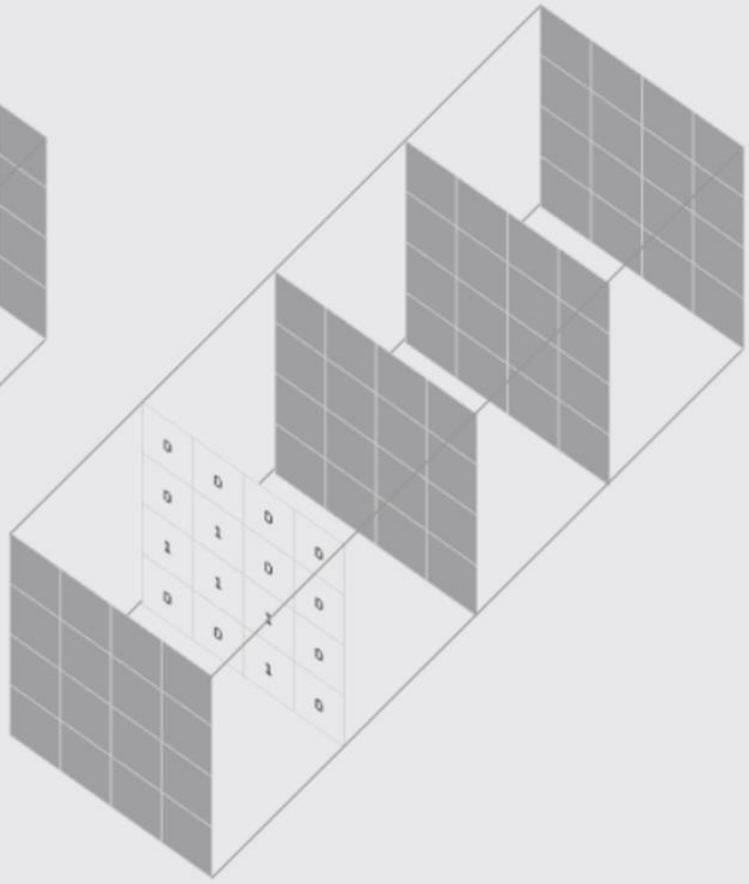
$$\frac{2TP}{2TP + FP + FN}$$

Table 1. The three similarity coefficients

Similarity Coefficient (X,Y)	**Actual Formula**
Dice Coefficient	$2\frac{\lvert X \cap Y \rvert}{\lvert X \rvert + \lvert Y \rvert}$
Cosine Coefficient	$\frac{\lvert X \cap Y \rvert}{\lvert X \rvert^{1/2} . \lvert Y \rvert^{1/2}}$
Jaccard Coefficient	$\frac{\lvert X \cap Y \rvert}{\lvert X \rvert + \lvert Y \rvert - \lvert X \cap Y \rvert}$



Prediction for a selected class

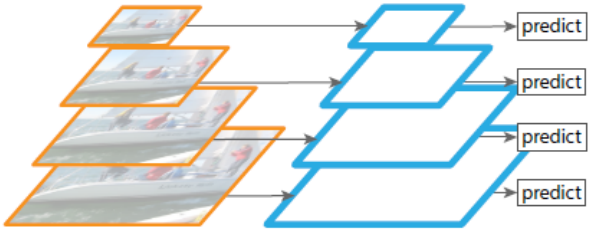


Target for the corresponding class

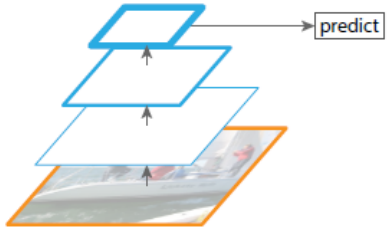
Soft Dice coefficient is calculated for each class mask

$$1 - \frac{2\sum_{pixels} y_{true} y_{pred}}{\sum_{pixels} y_{true}^2 + \sum_{pixels} y_{pred}^2}$$

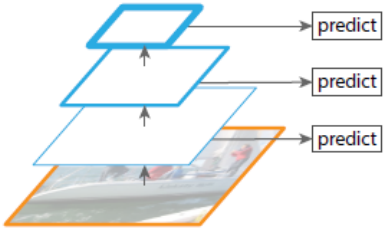
This scoring is repeated over all **classes** and averaged



(a) Featurized image pyramid

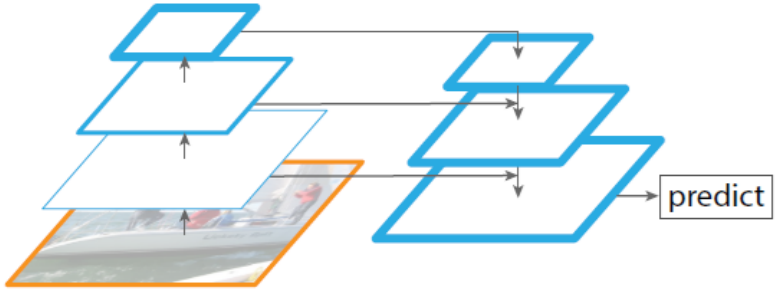


(b) Single feature map

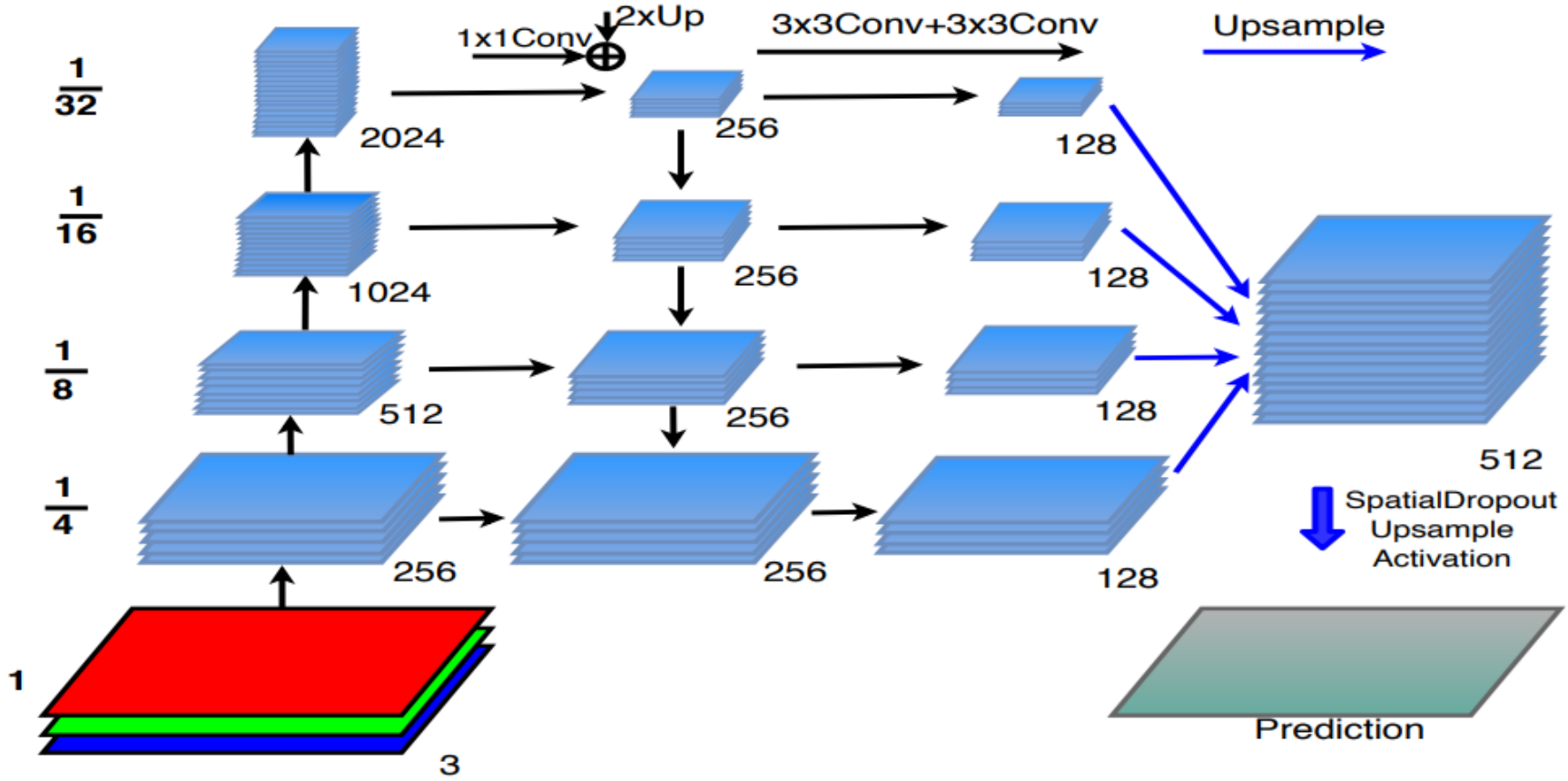


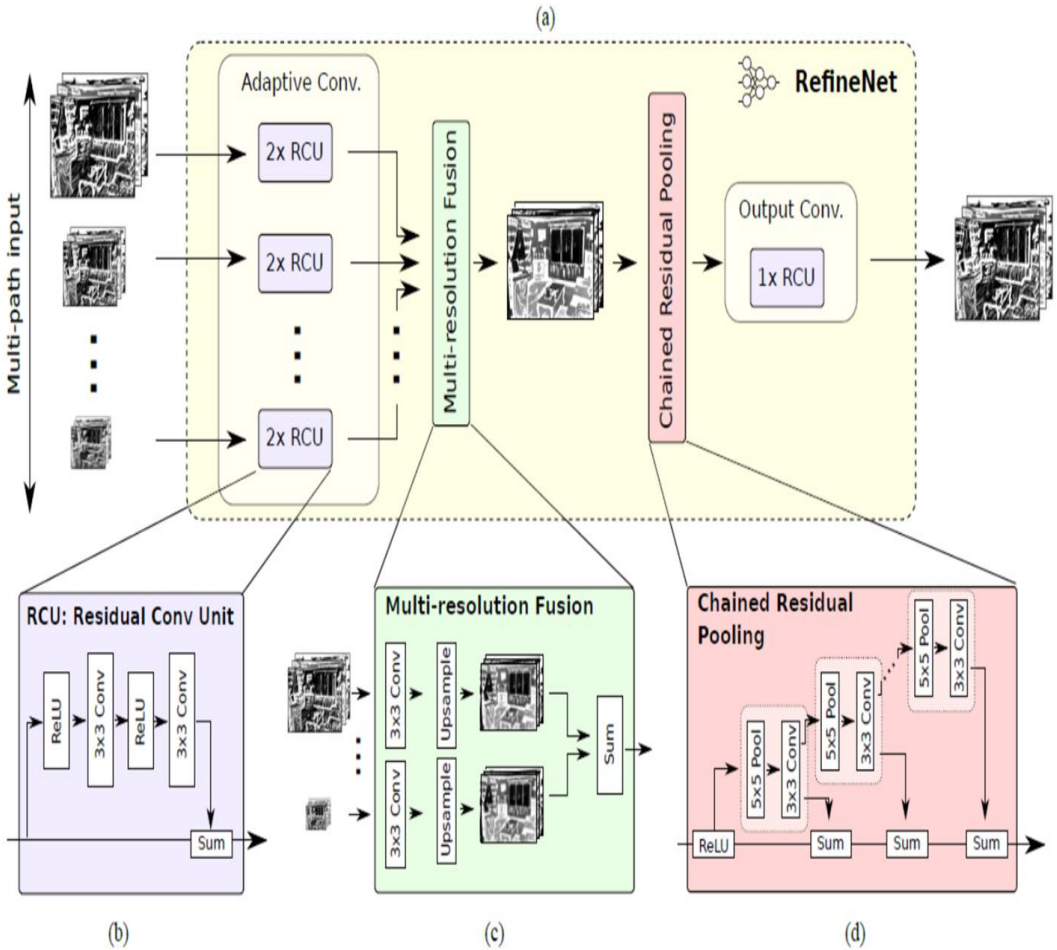
(c) Pyramidal feature hierarchy

(d) Feature Pyramid Network



(e) Similar Structure with (d)

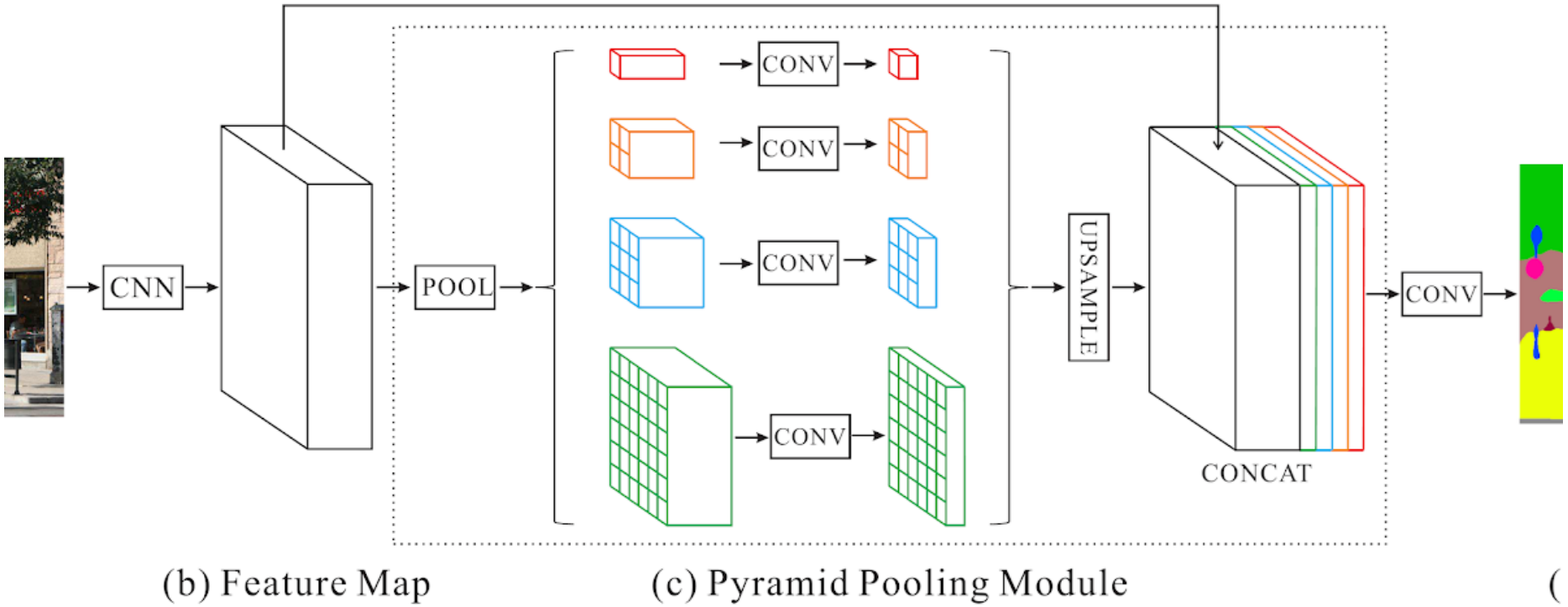

1x1Conv
2xUp
3x3Conv+3x3Conv
Upsample
1/32
1/16
1/8
1/4
2024
1024
512
256
256
256
256
256
128
128
128
128
512
SpatialDropout
Upsample
Activation
Prediction
1
3

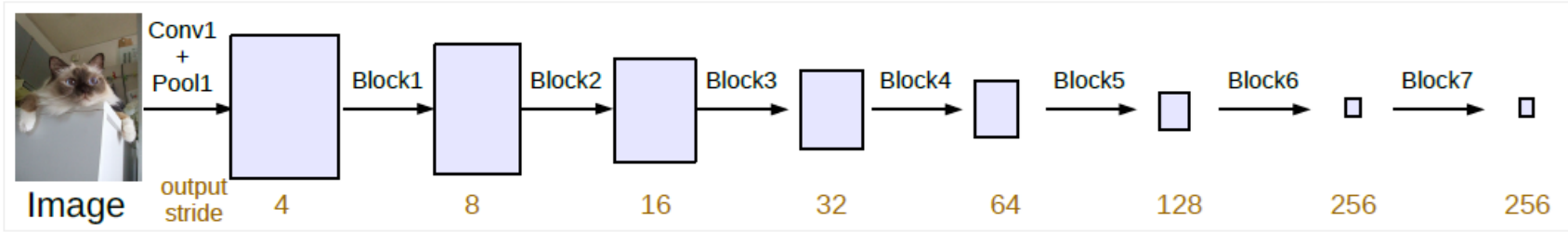


(a) Overall Architecture, (b) RCU, (c) Fusion, (d) Chained Residual Pooling

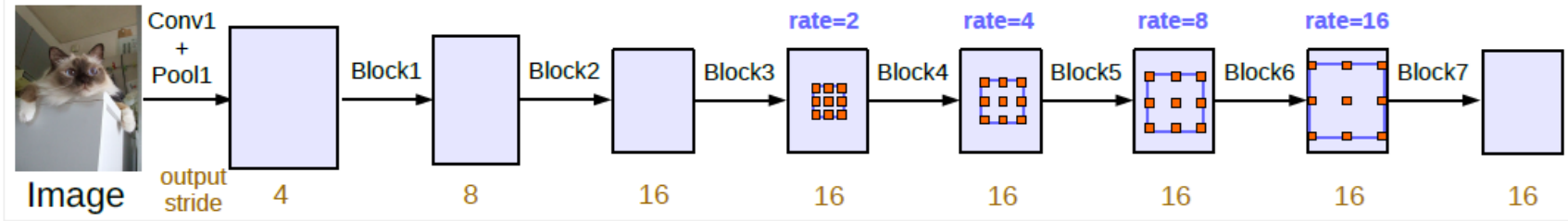
- **(a)**: At the top left of the figure, it is the ResNet backbone. Along the ResNet, different resolutions of feature maps go through Residual Conv Unit (RCU). Pre-Activation ResNet is used.
- **(b) RCU**: Residual block is used but with batch normalization removed.
- **(c) Fusion**: Then multi-resolution fusion is used to merge the feature maps using element-wise summation.
- **(d) Chained Residual Pooling**: The output feature maps of all pooling blocks are fused together with the input feature map through summation of residual connections. It **aims to capture background context from a large image region.**
- **(a) Output Conv**: At the right of the figure, finally, another RCU is placed here, to employ non-linearity operations on the multi-path fused feature maps to generate features for further processing or for final prediction.

PSP Net (Pyramid Special pooling)





(a) Going deeper without atrous convolution.



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

DeepLab v2

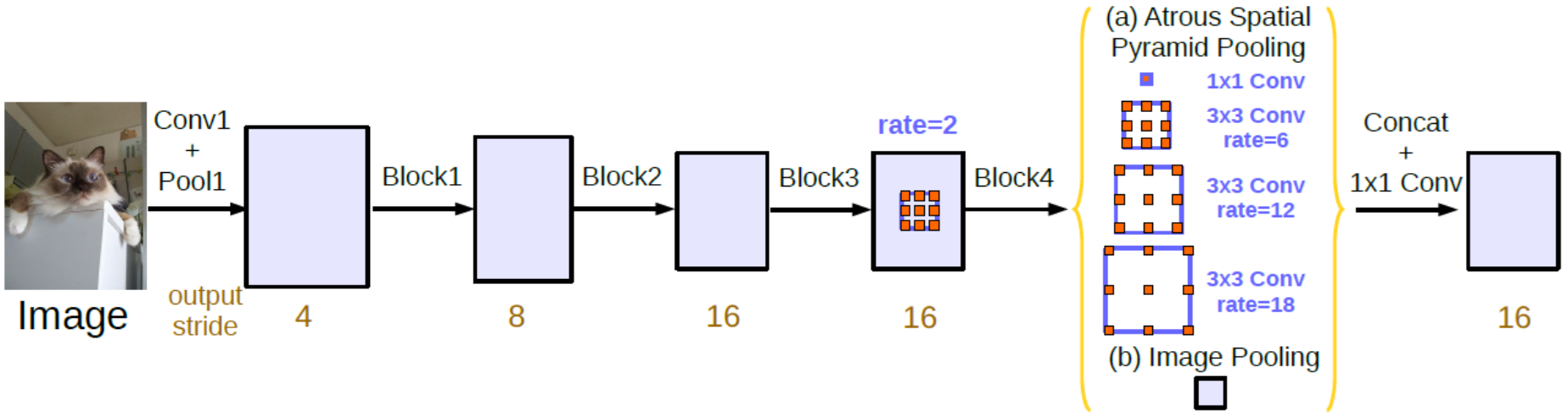
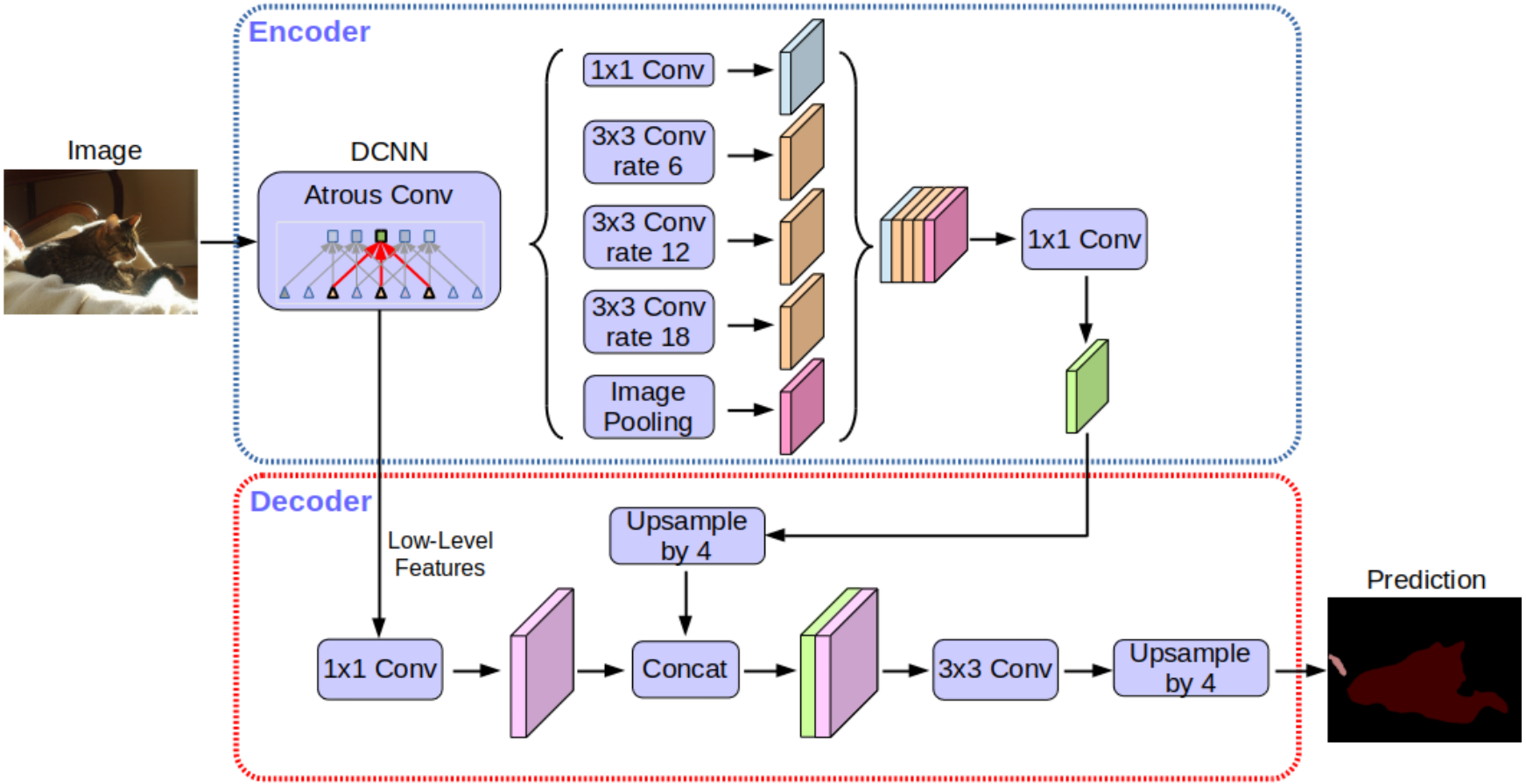


Image
Conv1
+
Pool1
output
stride
4
Block1
8
Block2
16
Block3
16
rate=2
Block4
(a) Atrous Spatial
Pyramid Pooling
1x1 Conv
3x3 Conv
rate=6
3x3 Conv
rate=12
3x3 Conv
rate=18
(b) Image Pooling
Concat
+
1x1 Conv
16

DeepLab V3 plus



HRNet

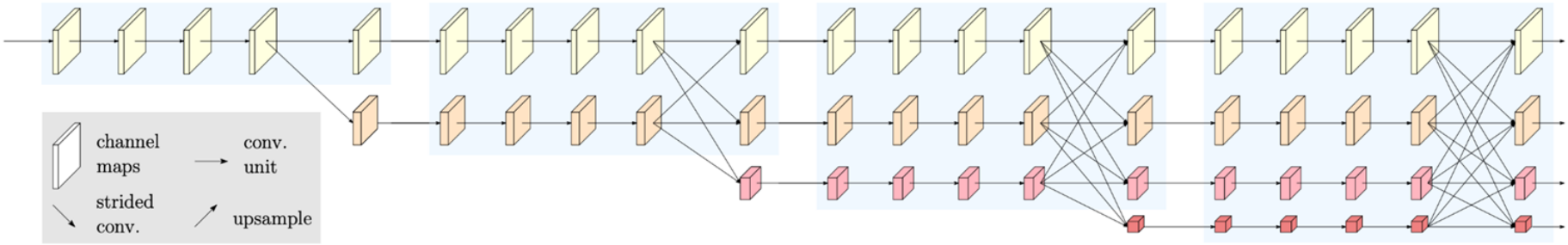


Fig. 2. An example of a high-resolution network. Only the main body is illustrated, and the stem (two stride-2 3×3 convolutions) is not included. There are four stages. The 1st stage consists of high-resolution convolutions. The 2nd (3rd, 4th) stage repeats two-resolution (three-resolution, four-resolution)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