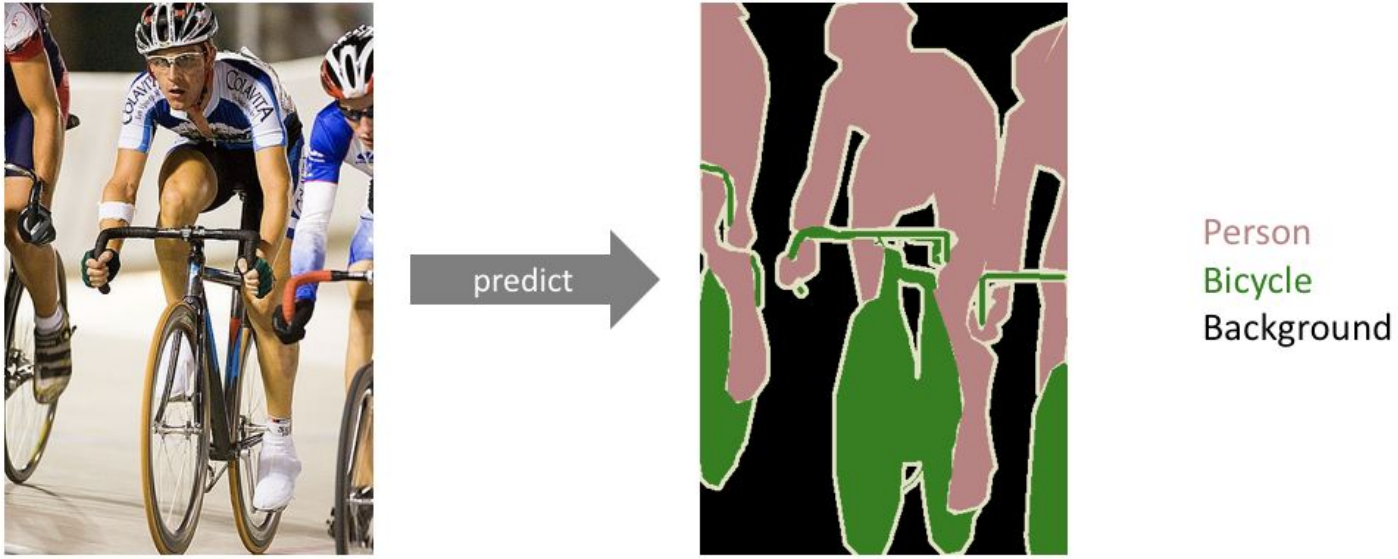
Image Segmentation

Igor Krashenyi, PhD

Outline

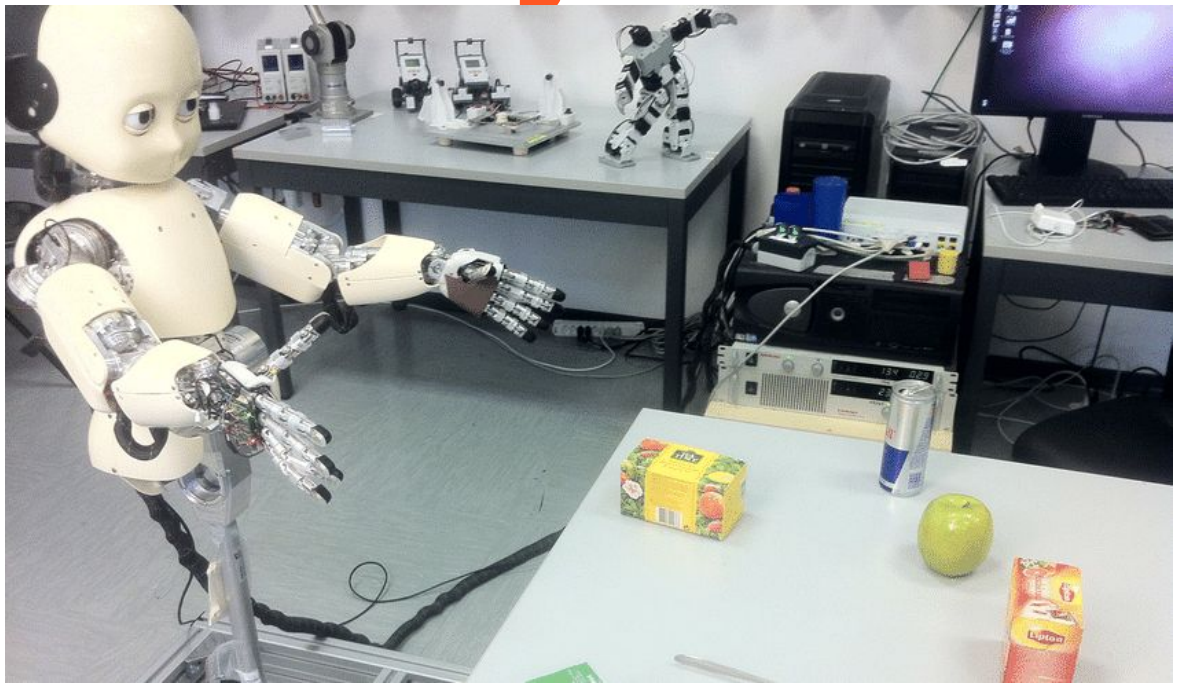
- Semantic segmentation
- Why?
- Benchmark Datasets
- Segmentation before Deep Learning, CRF
- CNN for Semantic Segmentation
- Losses for Semantic Segmentation
- Instance, panoptic and whatever segmentation

Semantic Segmentation



- Label each pixel in the image with a category label
- Don’t differentiate instances, only care about pixels

Why Semantic Segmentation?



To let robots segment objects so that they can grasp them

Why Semantic Segmentation?

Road scene understanding and autonomous navigation of cars and drones

Why Semantic Segmentation?

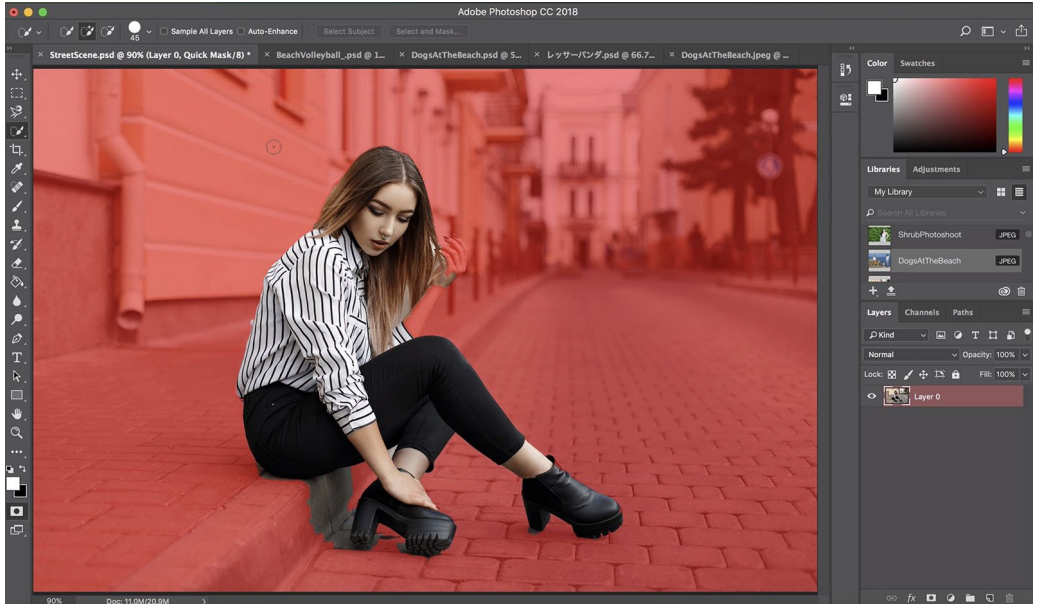
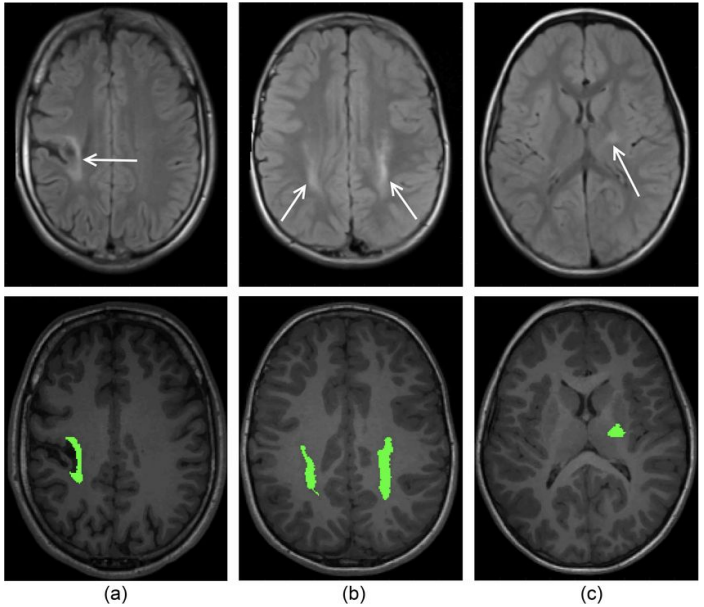
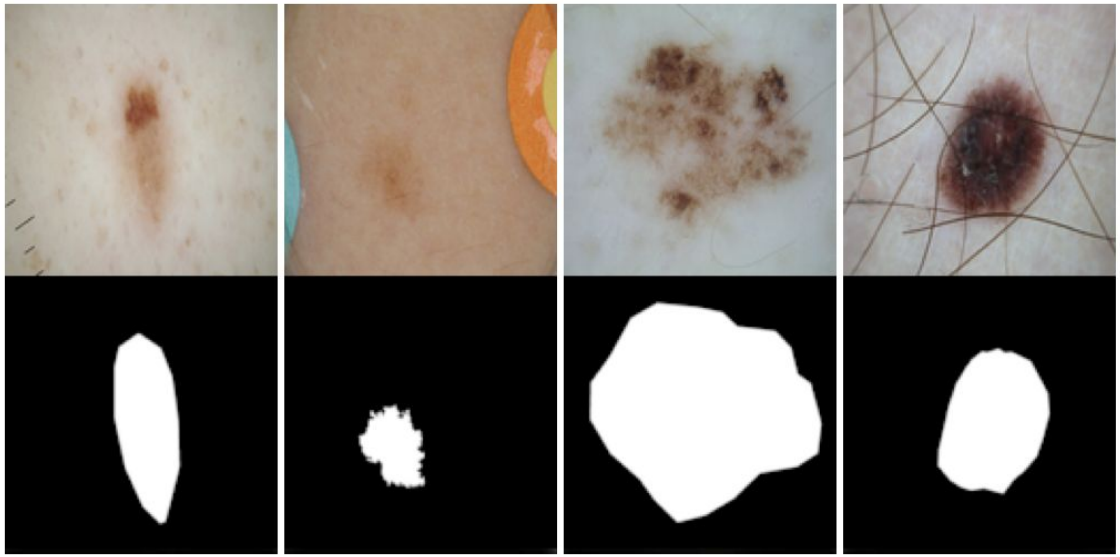


Photo and video editing

Why Semantic Segmentation?

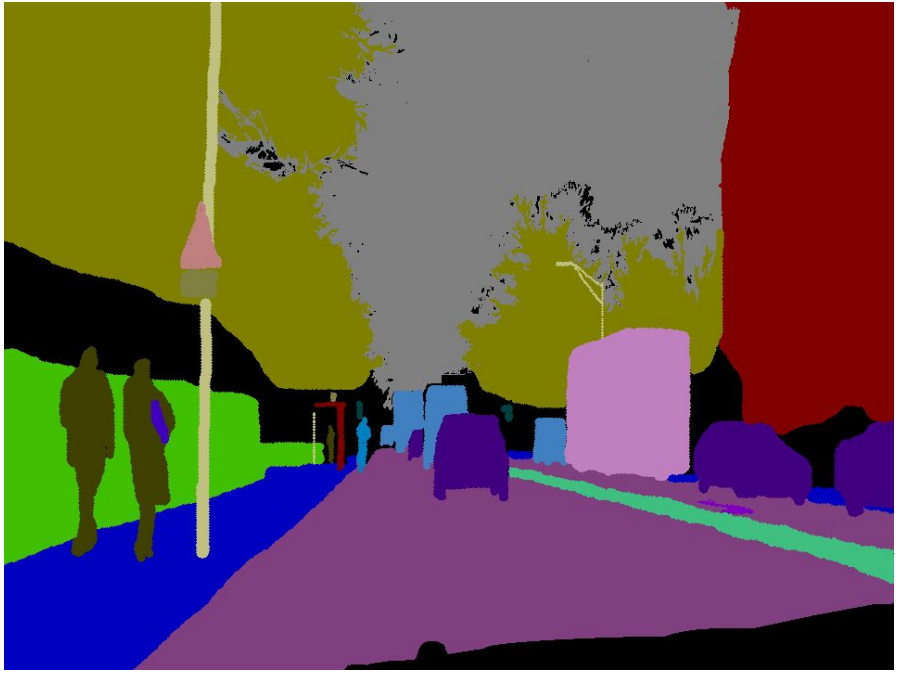




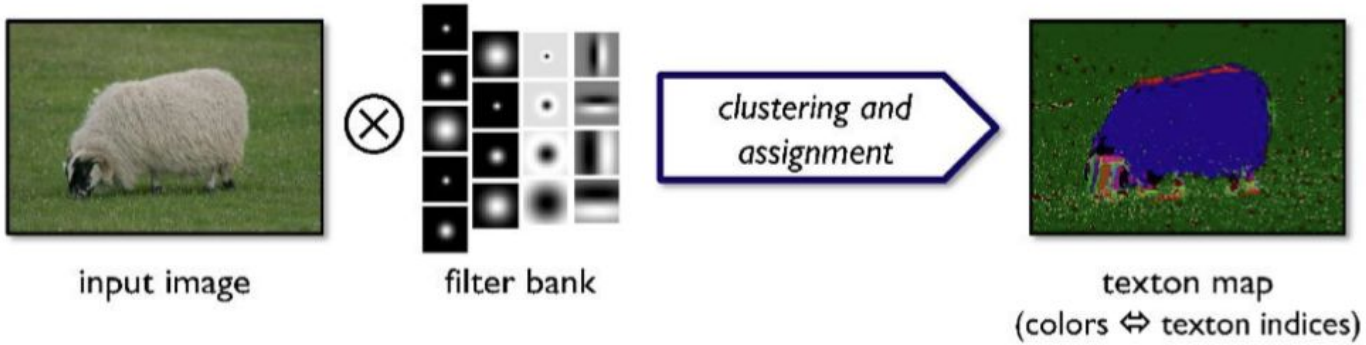
To find pathologies in medical images and to segment tissues.

Semantic Segmentation Datasets

- Cityscapes
- ADE20k
- Mapillary Vistats
- Pascal VOC 2012
- CamVid
- KITTI Semantic Segmentation
- other



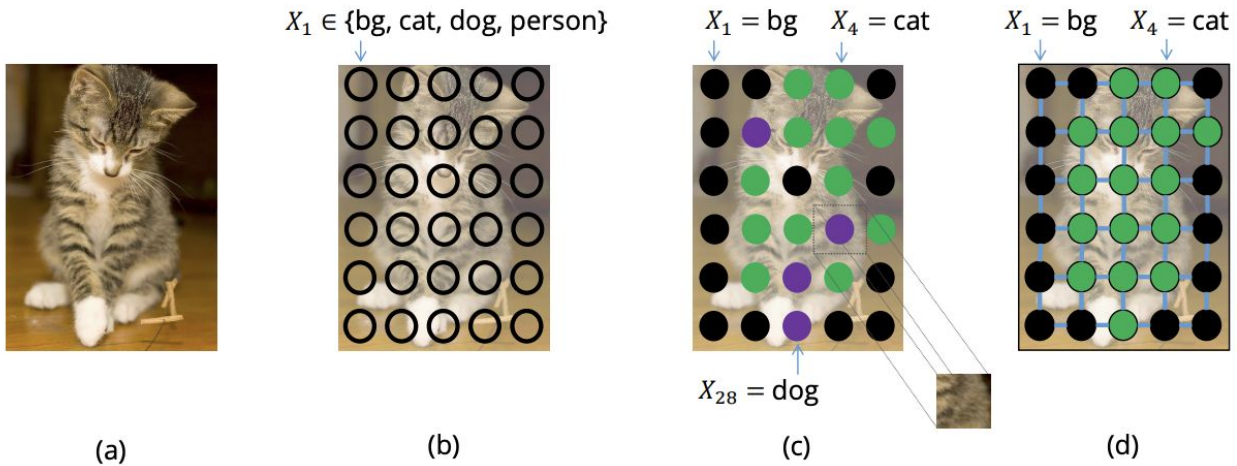
Semantic Segmentation before Deep Learning



- Relying on CRF
- Operating with pixels or superpixels
- Incorporate local evidence in unary potentials
- Interactions between label assignments

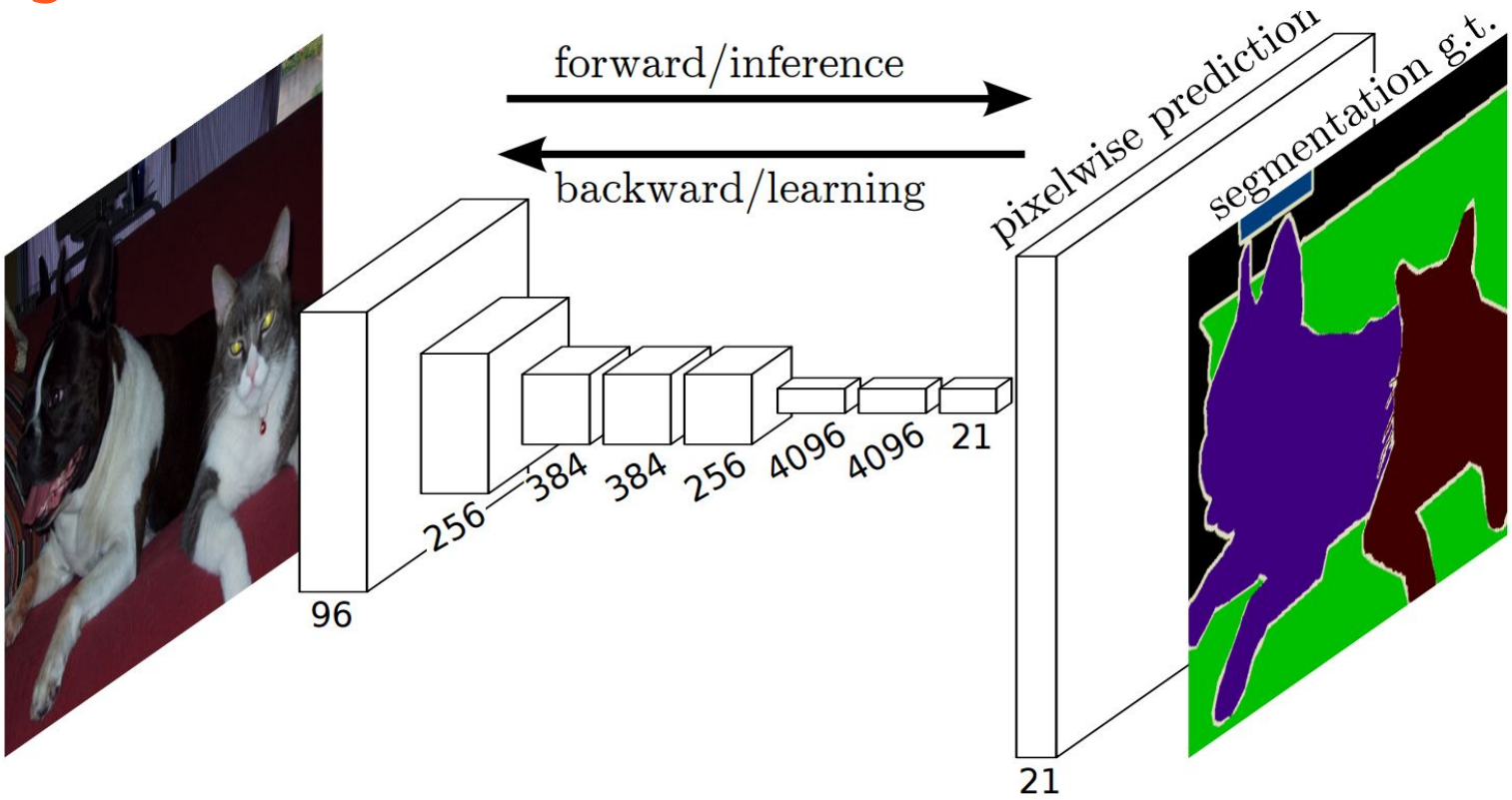
Conditional Random Fields

- tries to mode relationship between pixels
- nearby pixels more likely to have same label
- pixels with similar color more likely to have same label
- the pixels above the pixels "chair" more likely to be "person" instead of "plane"
- refine results by iterations



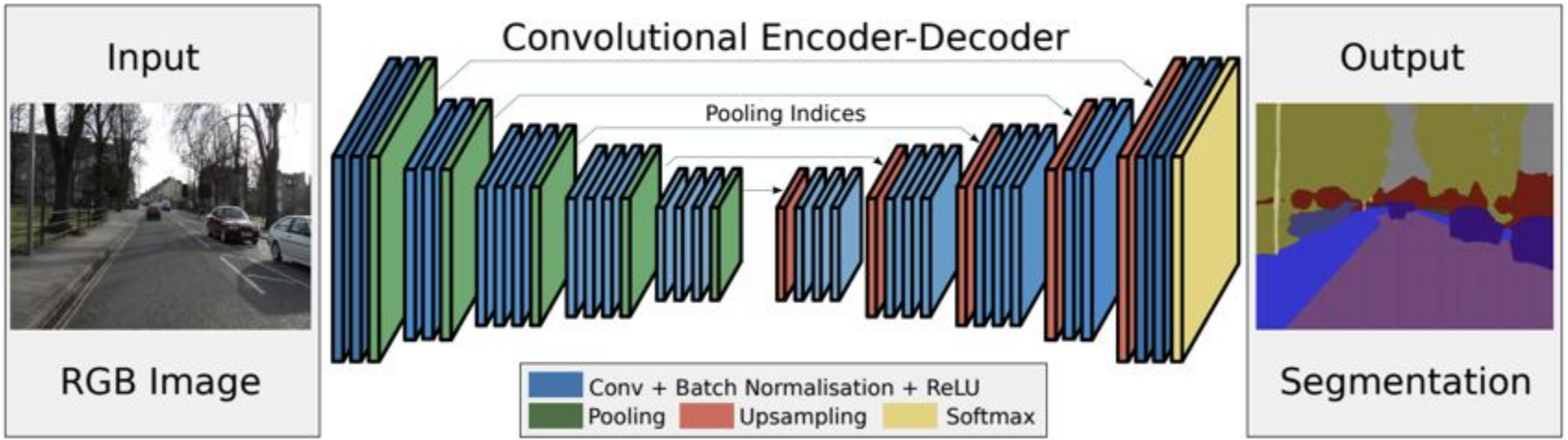
Arnab et al. 2018

Fully Convolutional Network



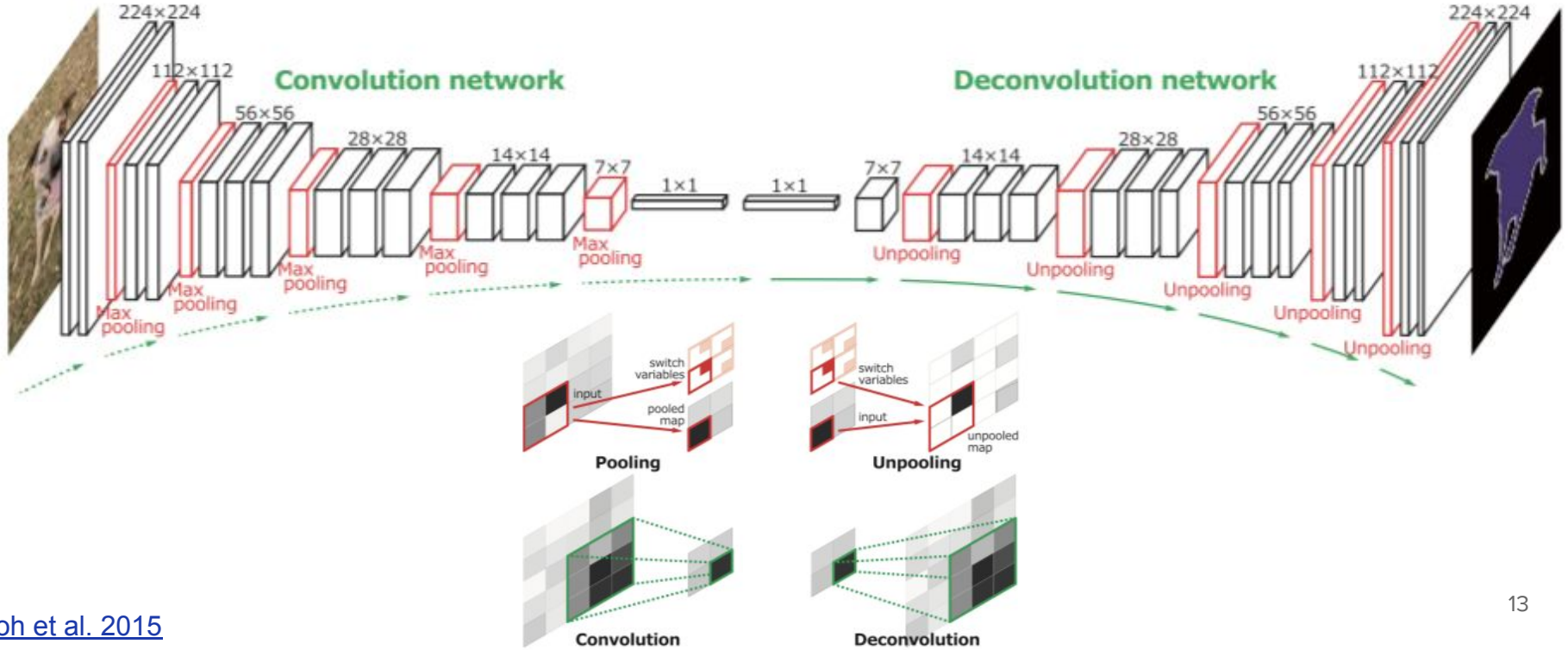
Long et al. 2014

SegNet



Badrinarayanan et al. 2015

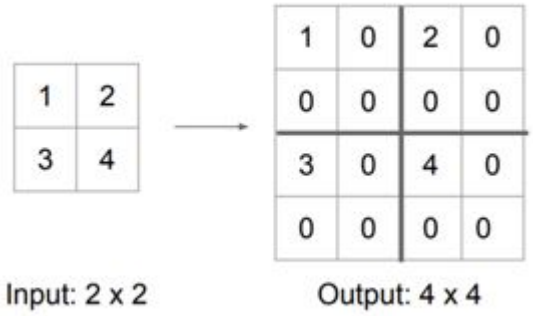
Learning Deconvolution Network



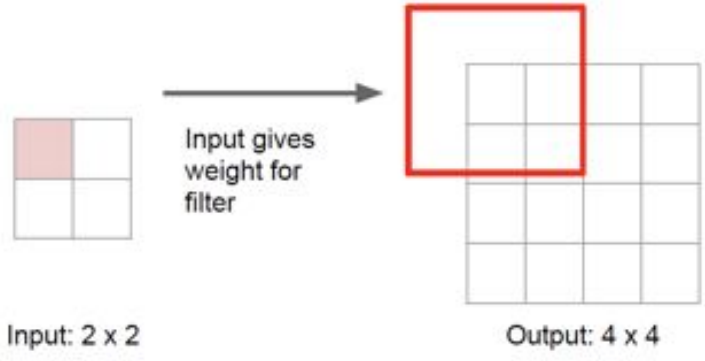
Noh et al. 2015

How Does Upsampling Work?

UnPooling

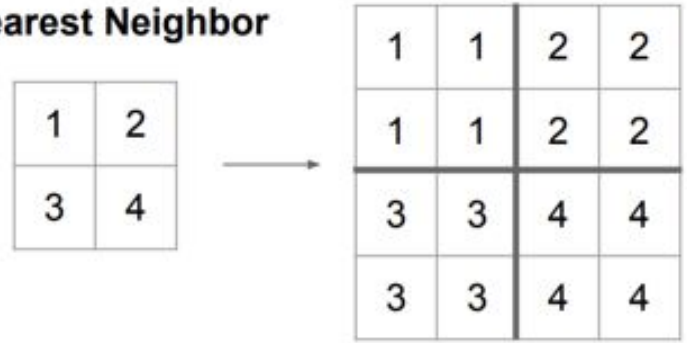


Transpose ~~De~~Convolution

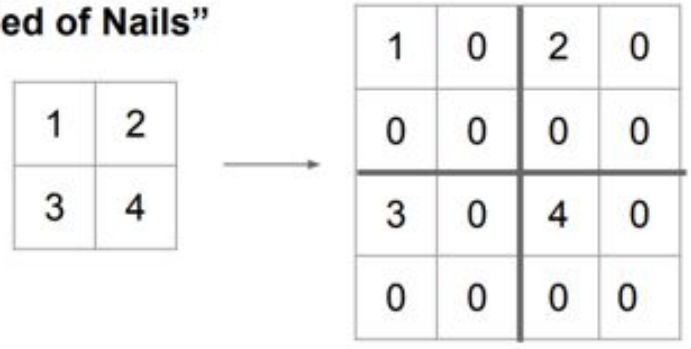


Unpooling Methods

Nearest Neighbor



“Bed of Nails”



Unpooling Methods

Max Pooling
Remember which element was max!

1	2	6	3
3	5	2	1
1	2	2	1
7	3	4	8

Input: 4 x 4

5	6
7	8

Output: 2 x 2

Rest of the network

Max Unpooling
Use positions from pooling layer

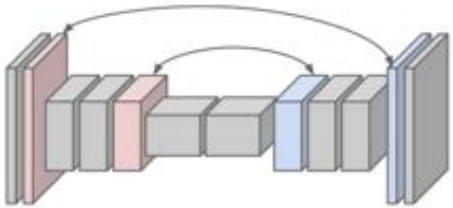
1	2
3	4

Input: 2 x 2

0	0	2	0
0	1	0	0
0	0	0	0
3	0	0	4

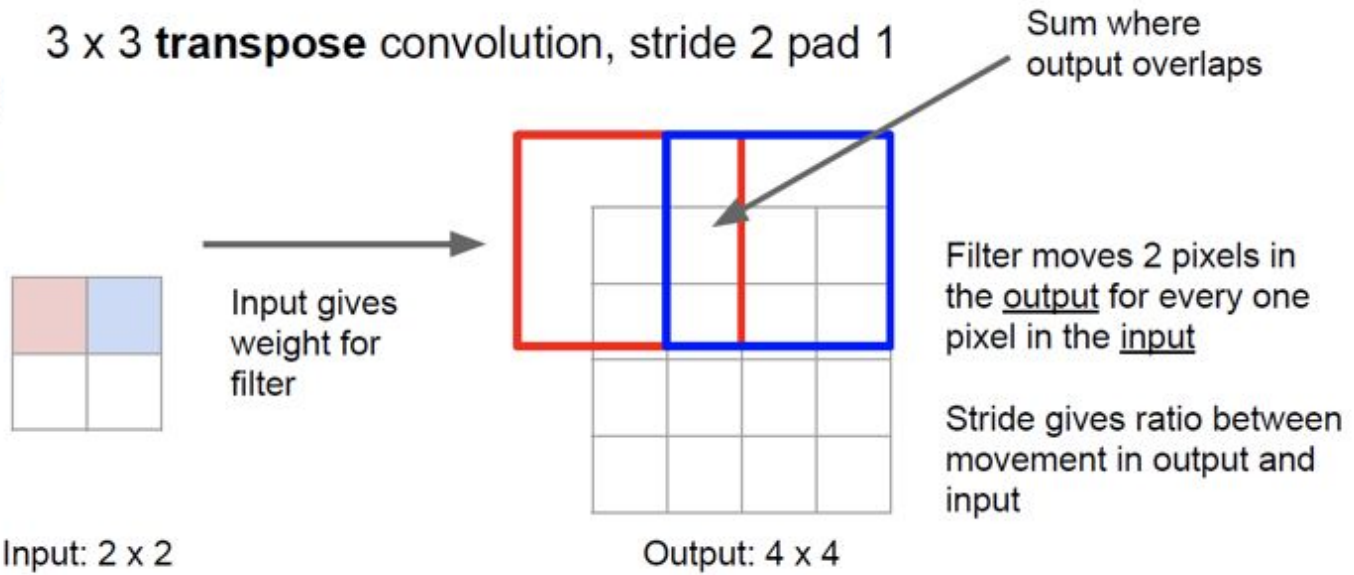
Output: 4 x 4

Corresponding pairs of downsampling and upsampling layers



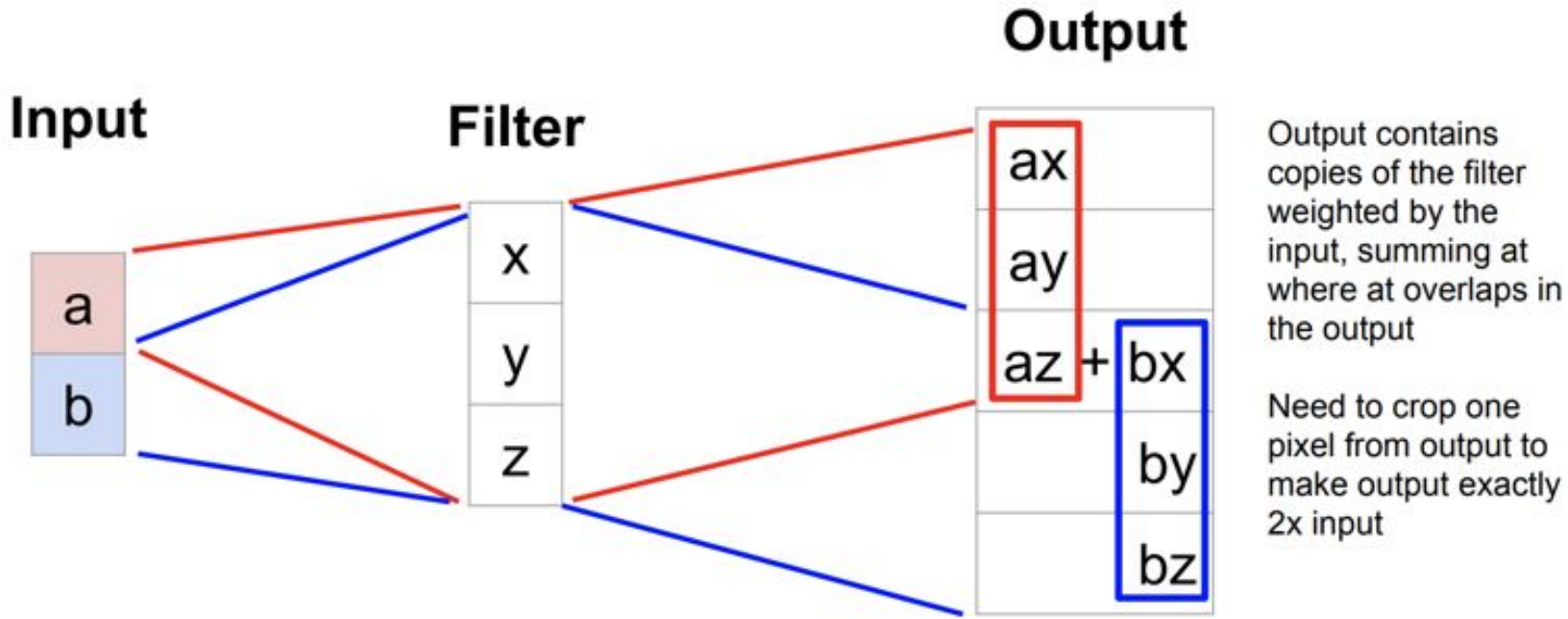
Transpose Convolution

Other names:
-Deconvolution (bad)
-Upconvolution
-Fractionally strided convolution
-Backward strided convolution



Transpose Convolution

Transpose Convolution: 1D Example



Output contains copies of the filter weighted by the input, summing at where at overlaps in the output

Need to crop one pixel from output to make output exactly 2x input

Convolution as Matrix Multiplication (1D Example)

We can express convolution in terms of a matrix multiplication

$$\vec{x} * \vec{a} = X\vec{a}$$

$$\begin{bmatrix} x & y & z & 0 & 0 & 0 \\ 0 & x & y & z & 0 & 0 \\ 0 & 0 & x & y & z & 0 \\ 0 & 0 & 0 & x & y & z \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ ax + by + cz \\ bx + cy + dz \\ cx + dy \end{bmatrix}$$

Example: 1D conv, kernel size=3, stride=1, padding=1

Convolution transpose multiplies by the transpose of the same matrix:

$$\vec{x} *^T \vec{a} = X^T\vec{a}$$

$$\begin{bmatrix} x & 0 & 0 & 0 \\ y & x & 0 & 0 \\ z & y & x & 0 \\ 0 & z & y & x \\ 0 & 0 & z & y \\ 0 & 0 & 0 & z \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} ax \\ ay + bx \\ az + by + cx \\ bz + cy + dx \\ dz + dy \\ dy \end{bmatrix}$$

When stride=1, convolution transpose is just a regular convolution (with different padding rules)

Convolution as Matrix Multiplication (1D Example)

We can express convolution in terms of a matrix multiplication

$$\vec{x} * \vec{a} = X\vec{a}$$

$$\begin{bmatrix} x & y & z & 0 & 0 & 0 \\ 0 & 0 & x & y & z & 0 \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ bx + cy + dz \end{bmatrix}$$

Example: 1D conv, kernel size=3, stride=2, padding=1

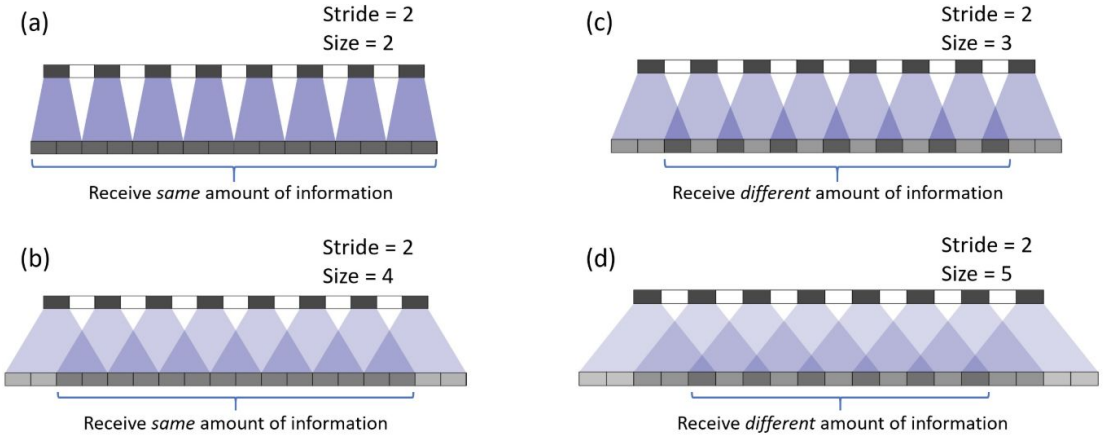
Convolution transpose multiplies by the transpose of the same matrix:

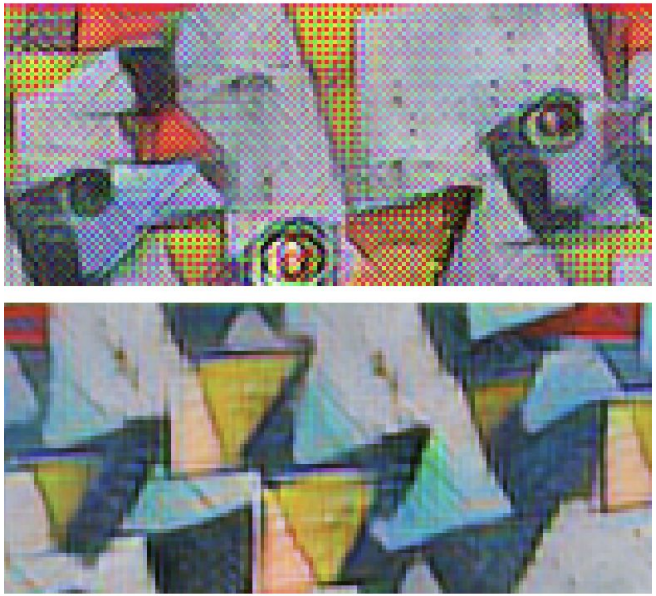
$$\vec{x} *^T \vec{a} = X^T \vec{a}$$

$$\begin{bmatrix} x & 0 \\ y & 0 \\ z & x \\ 0 & y \\ 0 & z \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} ax \\ ay \\ az + bx \\ by \\ bz \\ 0 \end{bmatrix}$$

When stride>1, convolution transpose is no longer a normal convolution!

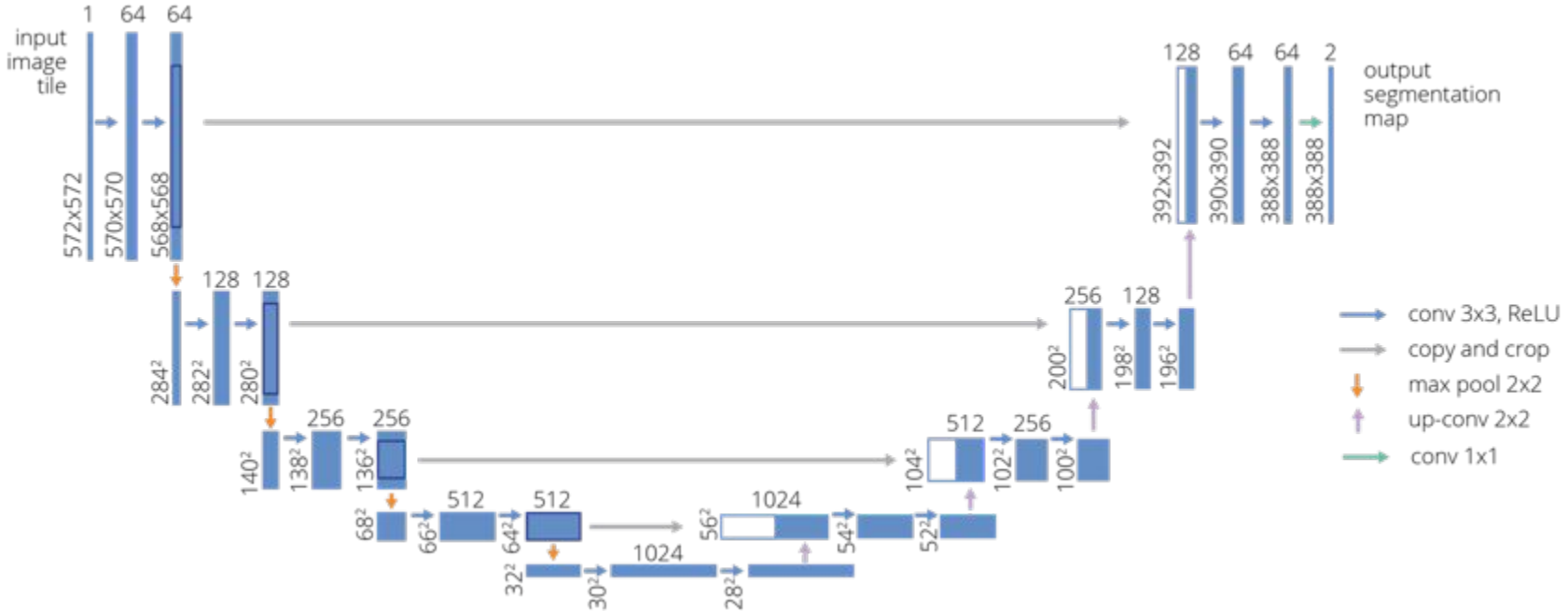
Transpose Convolutions and Checkerboard Artifacts





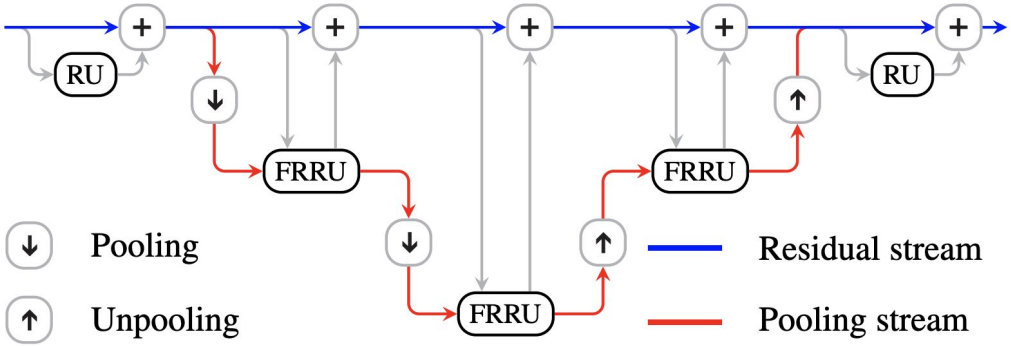
Odena, et al., 2016

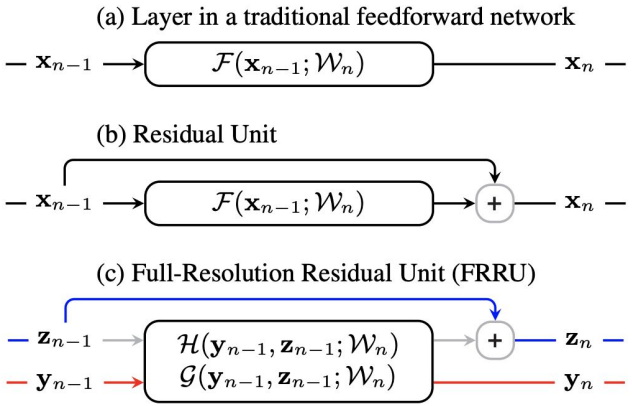
U-Net



Ronneberger et al. 2015

Full-Resolution Residual Networks

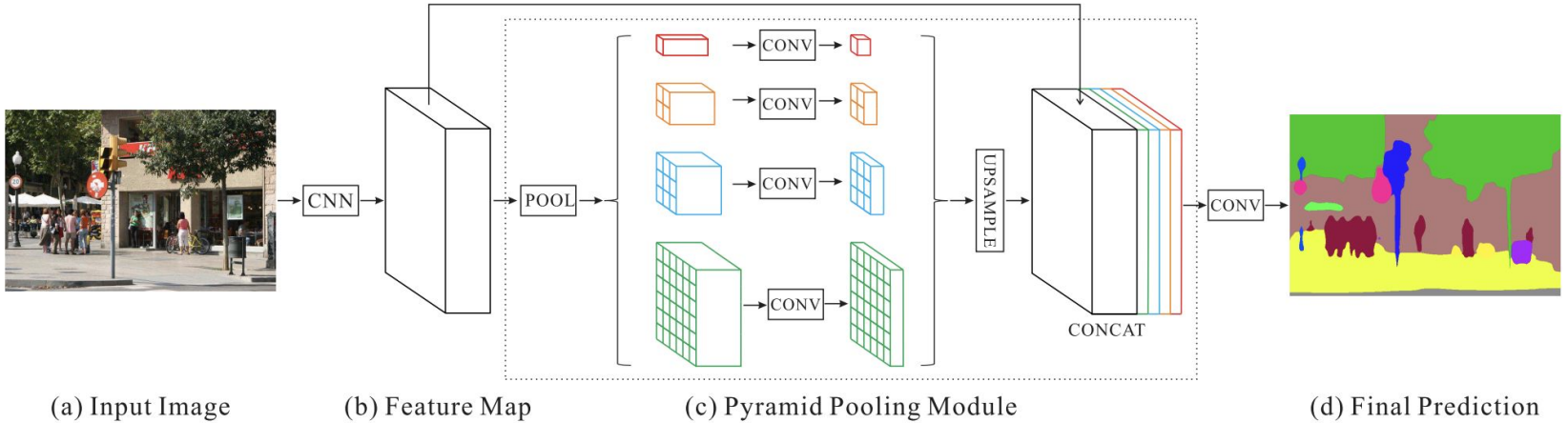


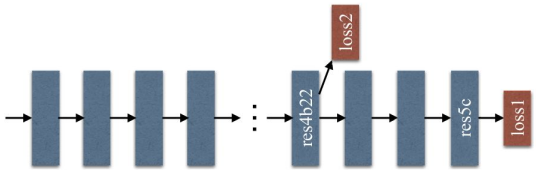


$$\mathbf{z}_n = \mathbf{z}_{n-1} + \mathcal{H}(\mathbf{y}_{n-1}, \mathbf{z}_{n-1}; \mathcal{W}_n)$$
$$\mathbf{y}_n = \mathcal{G}(\mathbf{y}_{n-1}, \mathbf{z}_{n-1}; \mathcal{W}_n),$$

Pohlen et al. 2016

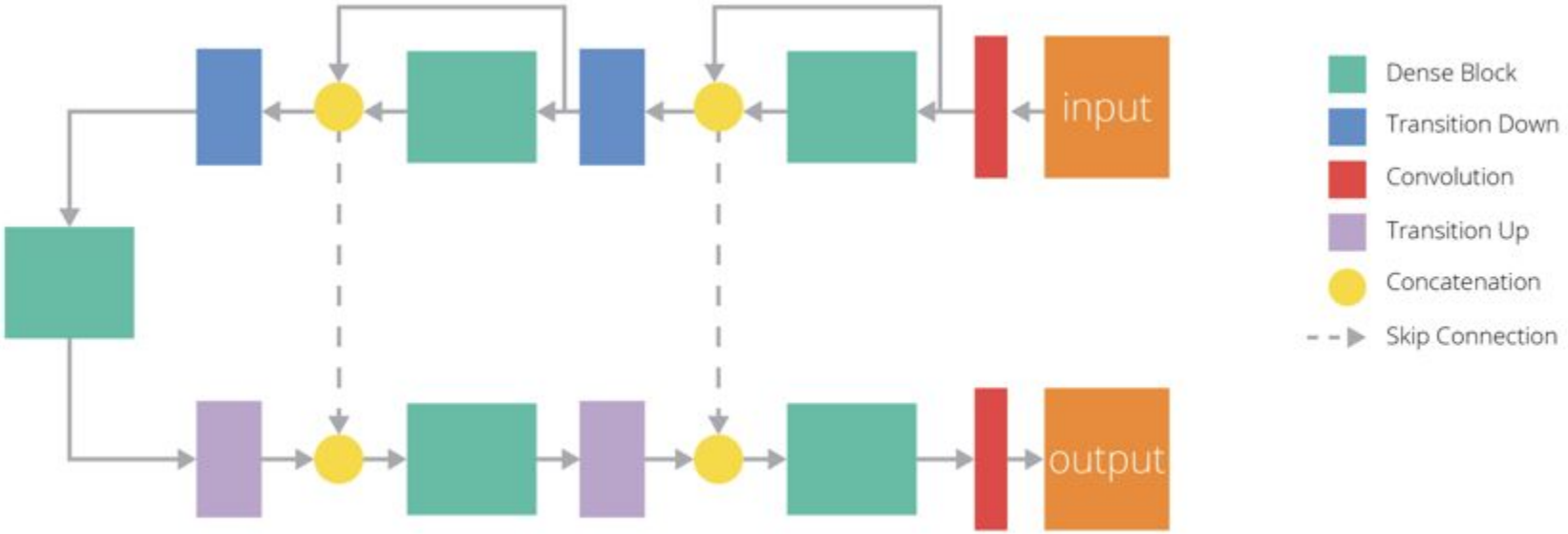
Pyramid Scene Parsing Network (PSPNet)





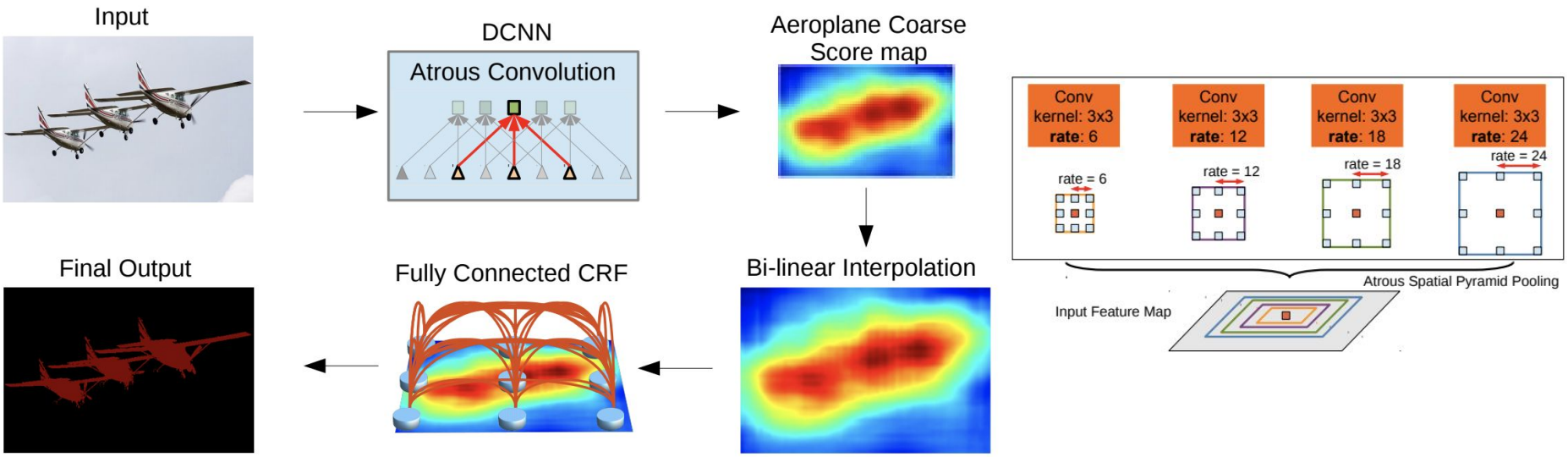
Zhao et al. 2016

The One Hundred Layers Tiramisu



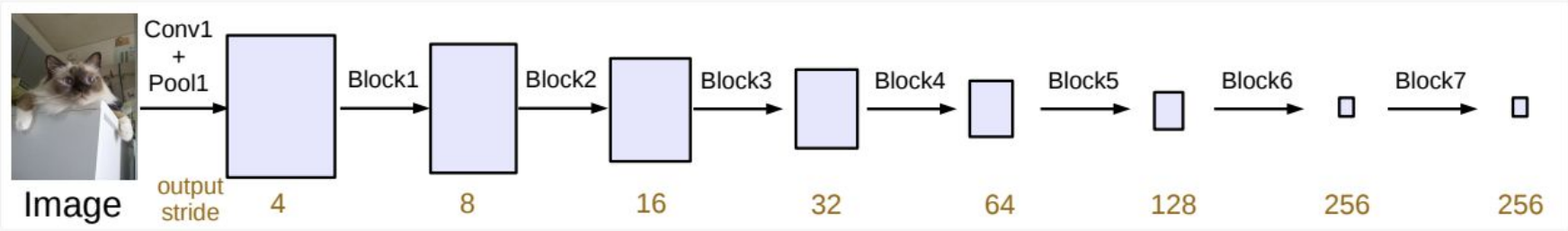
Jégou et al. 2016

DeepLab V1 and DeepLab V2

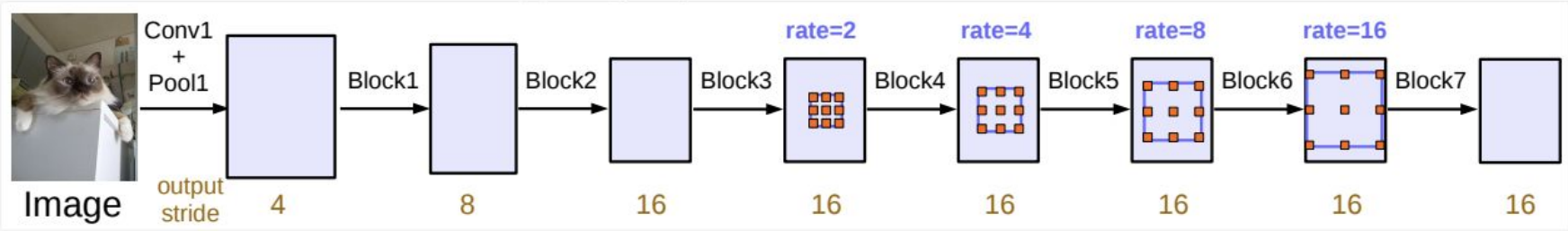


[Chen et al. 2016]

DeepLab V1 and DeepLab V2



(a) Going deeper without atrous convolution.

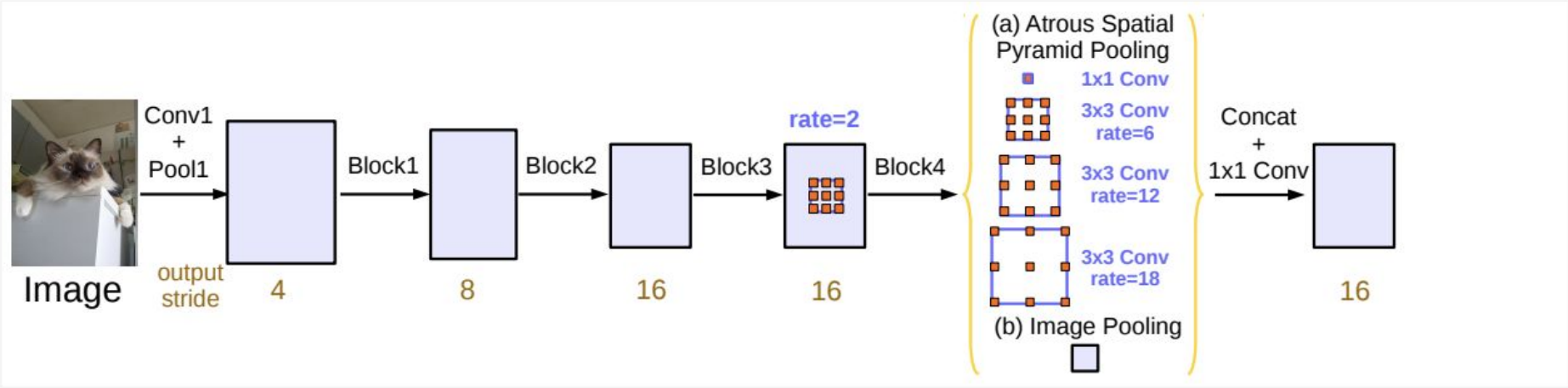


(b) Going deeper with atrous convolution. Atrous convolution with $rate > 1$ is applied after block3 when $output\_stride = 16$.

Figure 3. Cascaded modules without and with atrous convolution.

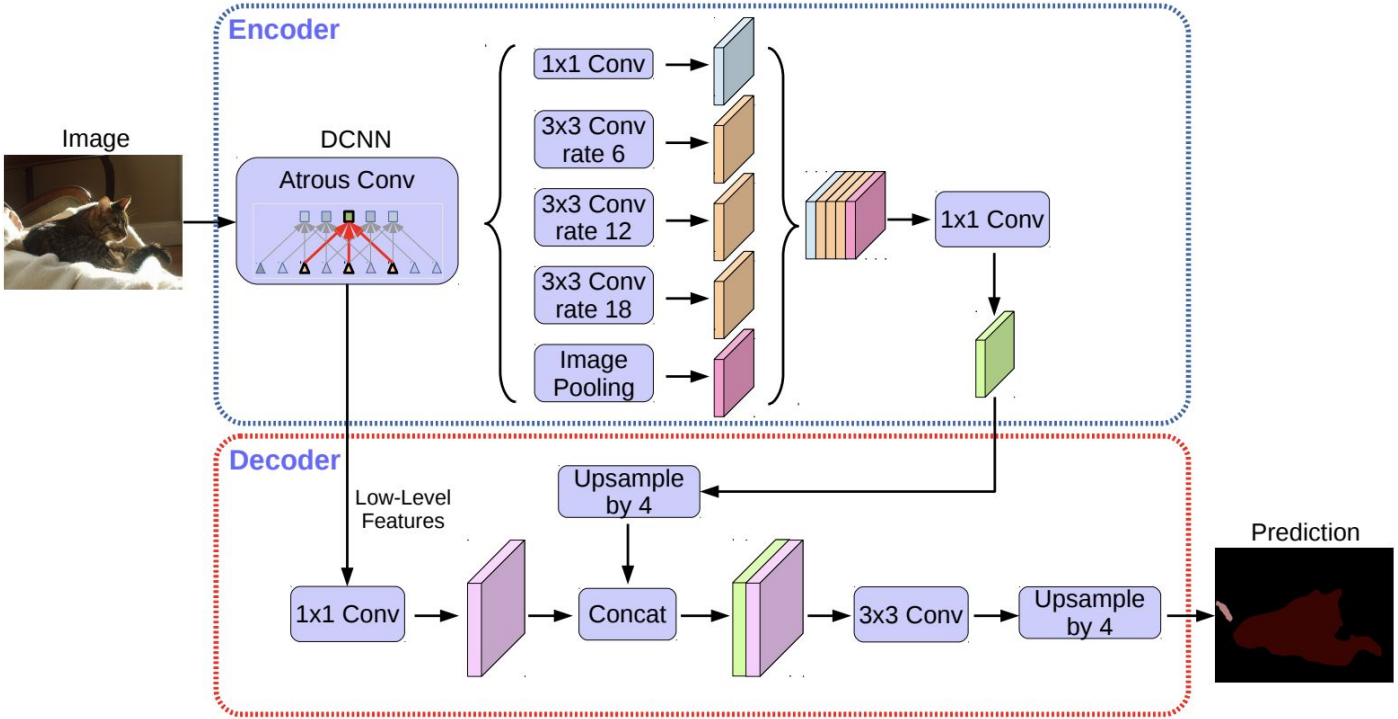
Chen et al. 2016

DeepLab V3



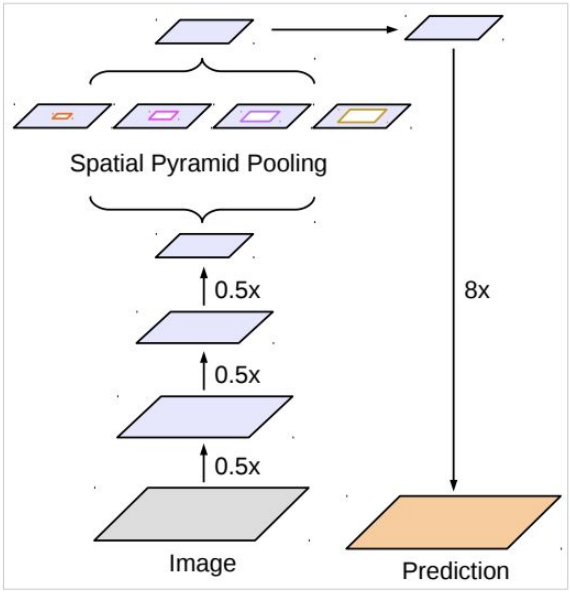
Chen et al. 2017

DeepLab V3+

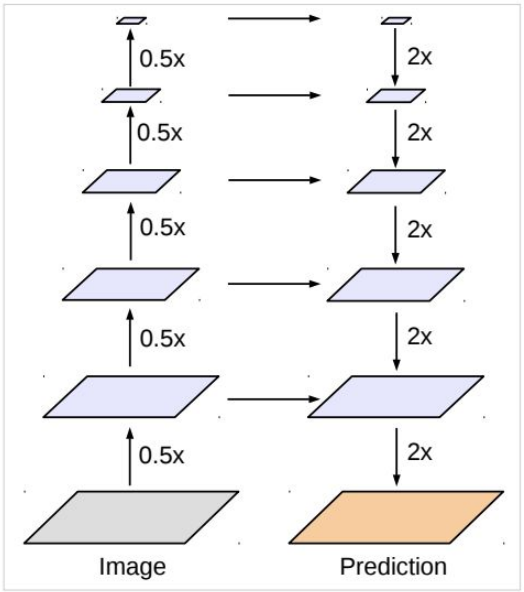


Chen et al. 2018

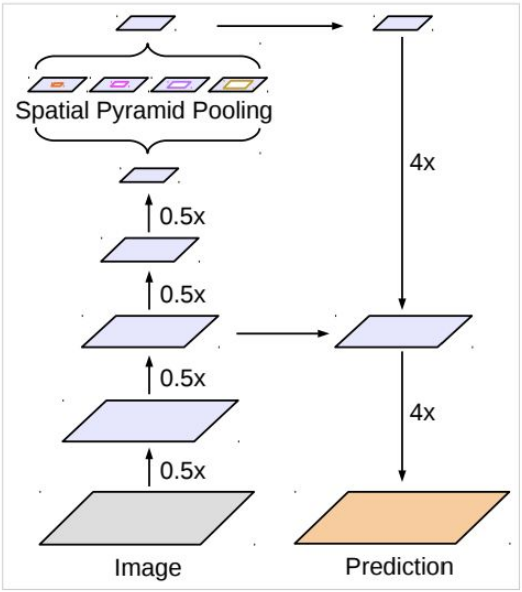
DeepLab V3+ vs DeepLab V3



(a) Spatial Pyramid Pooling



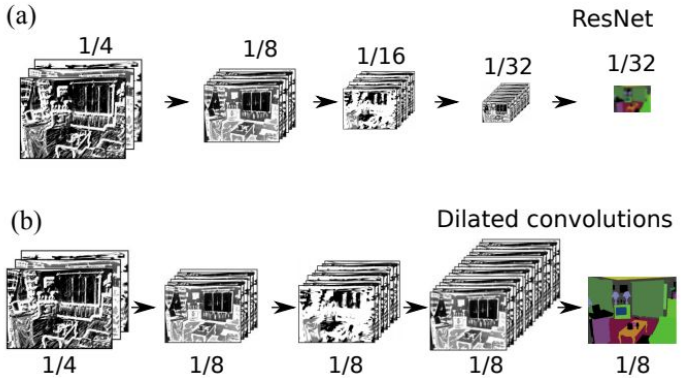
(b) Encoder-Decoder

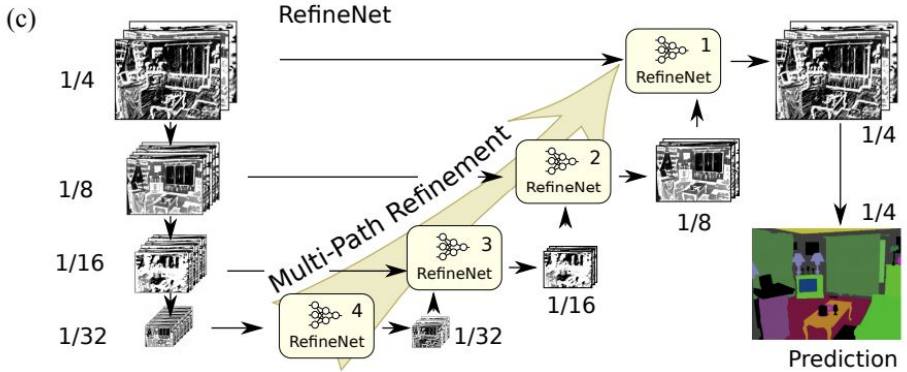


(c) Encoder-Decoder with Atrous Conv

Chen et al. 2018

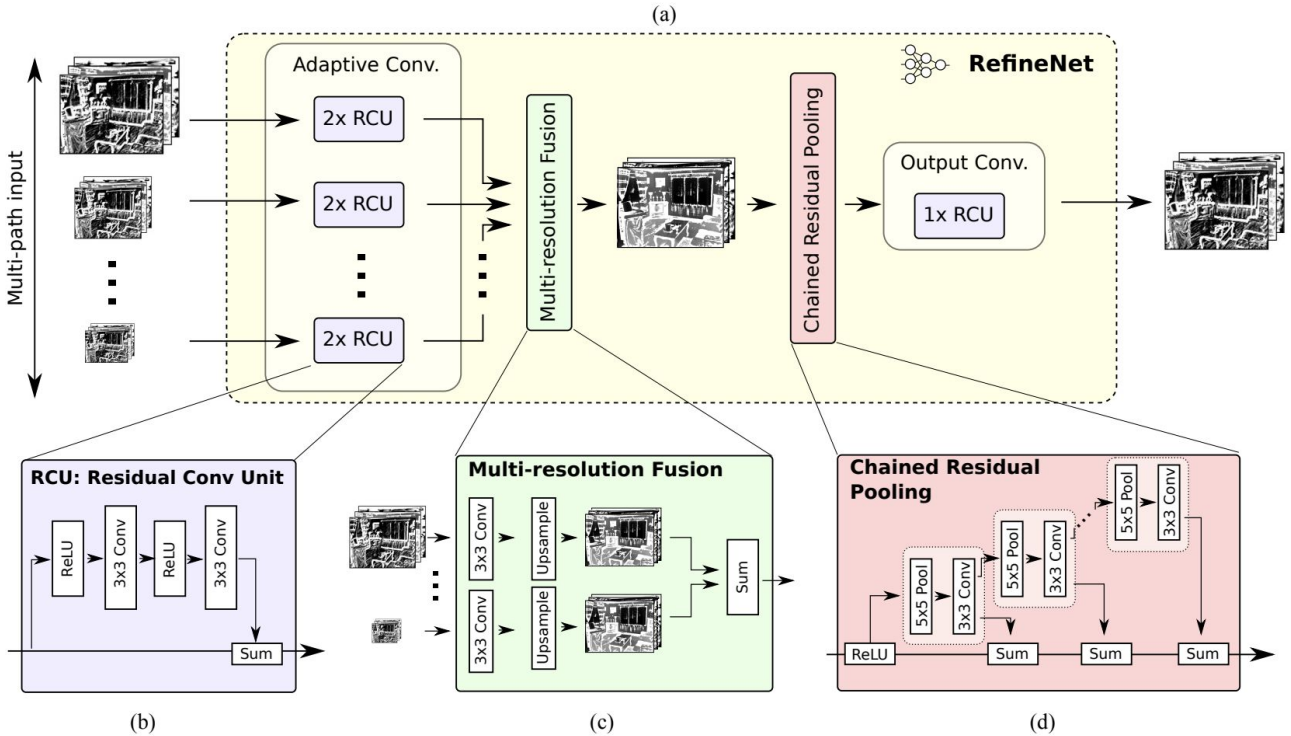
Multi-Path Refinement Networks (RefineNet)





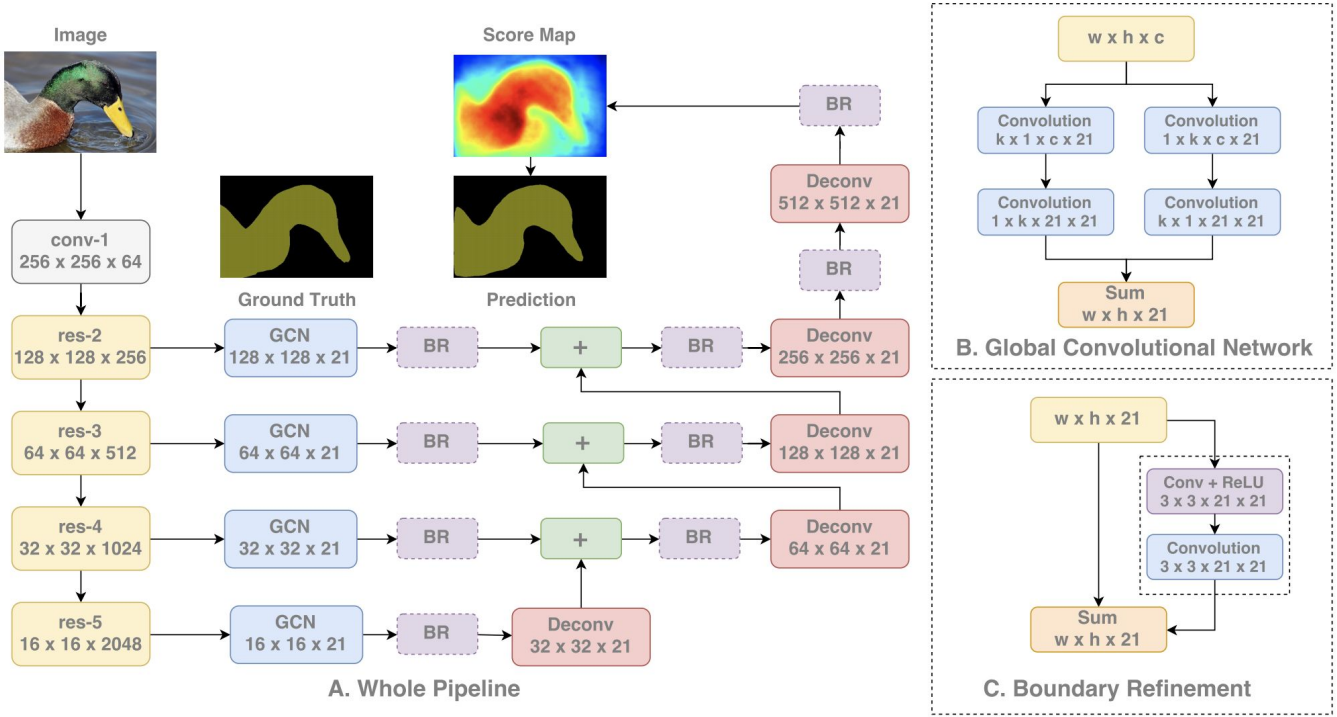
[Lin et al. 2016]

Multi-Path Refinement Networks (RefineNet)



Lin et al. 2016

Large Kernel Matters: GCN



[Peng et al. 2017]

Large Kernel Matters: GCN

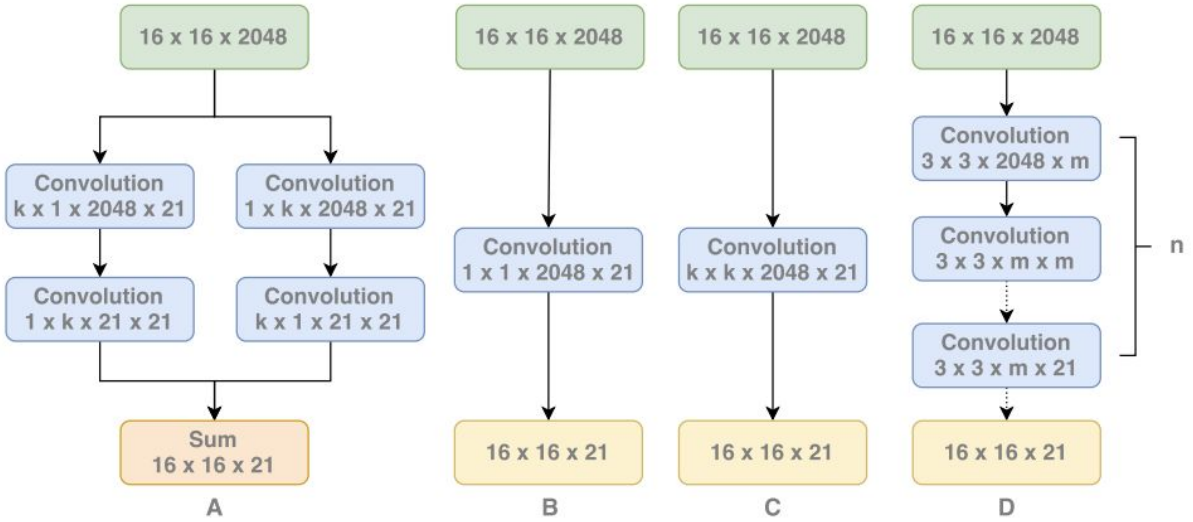
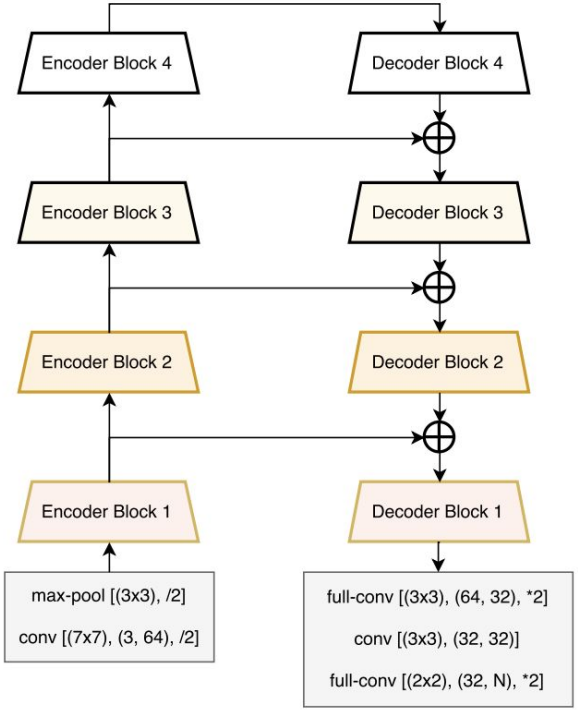
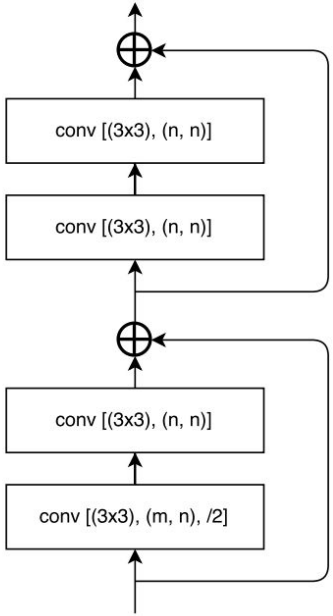


Figure 4. (A) Global Convolutional Network. (B) 1×1 convolution baseline. (C) $k \times k$ convolution. (D) stack of 3×3 convolutions.

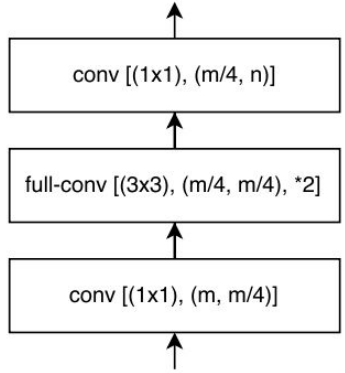
Peng et al. 2017

LinkNet





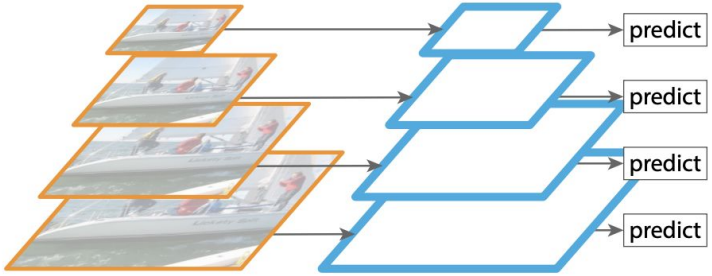
: Convolutional modules in *encoder-block (i)*



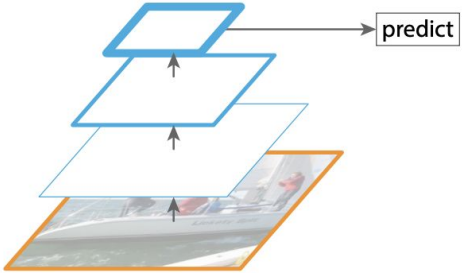
Convolutional modules in *decoder-block (i)*

[Chaurasia et al. 2017]

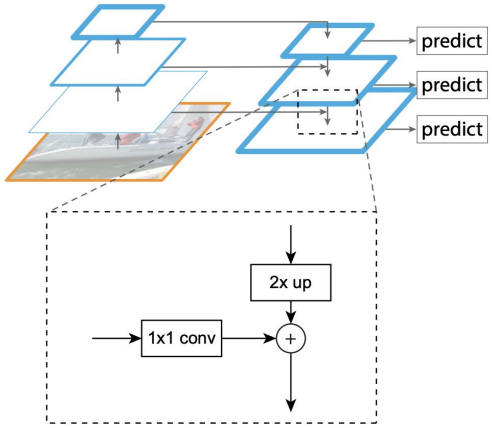
Feature Pyramid Networks

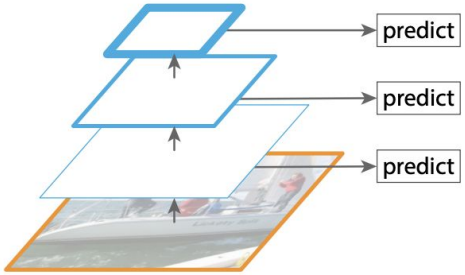


(a) Featurized image pyramid

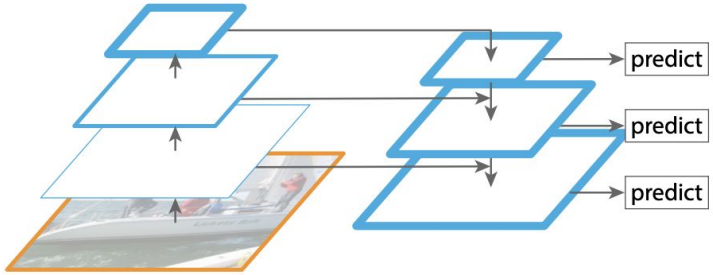


(b) Single feature map





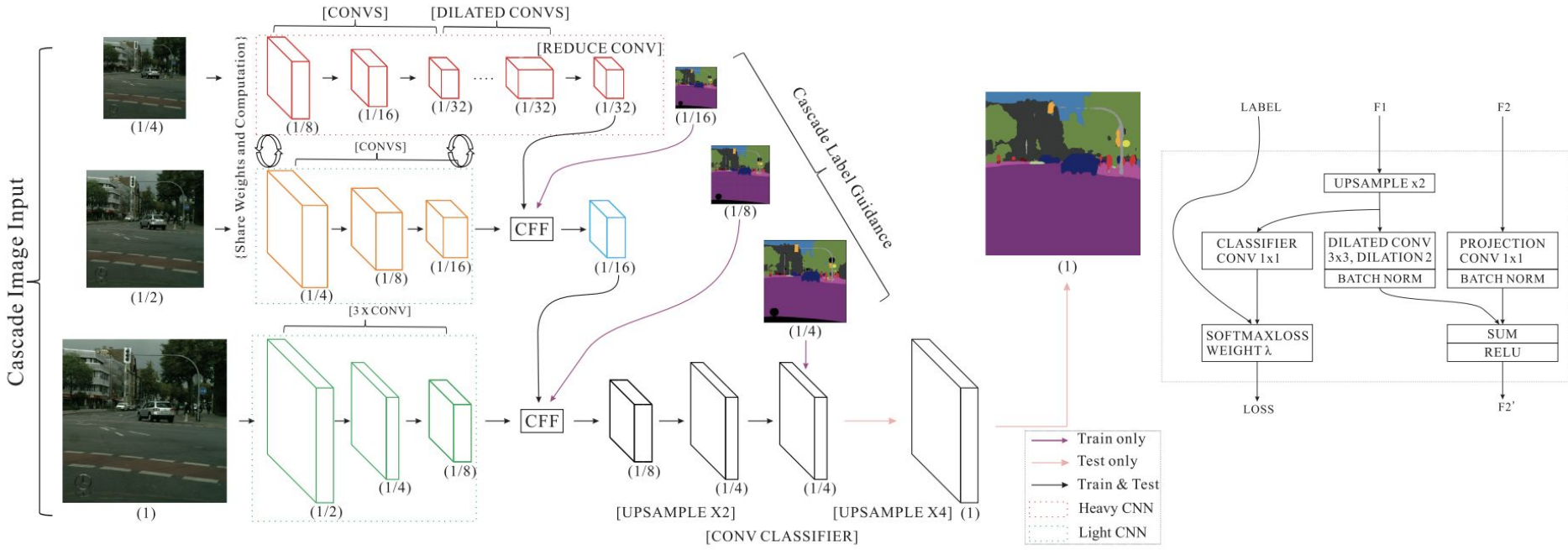
(c) Pyramidal feature hierarchy



(d) Feature Pyramid Network

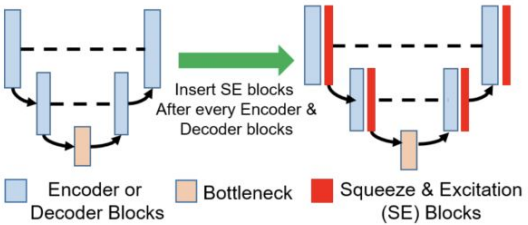
Lin et al. 2017

ICNet

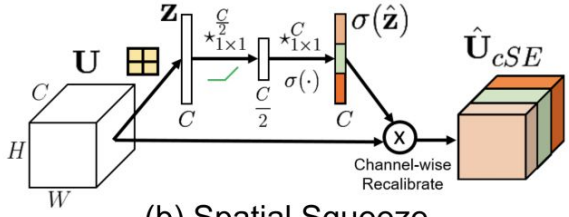


[Zhao et al. 2018]

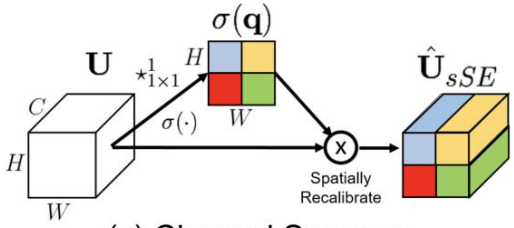
Concurrent Squeeze and Excitation



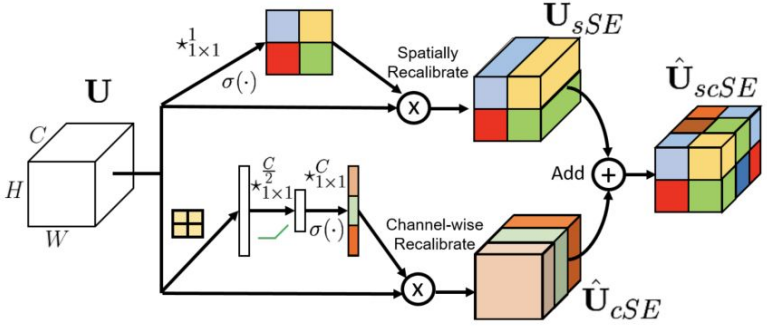
(a) SE block in F-CNN



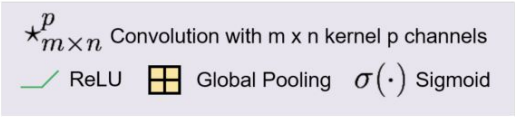
(b) Spatial Squeeze
and Channel Excitation (cSE)



(c) Channel Squeeze
and Spatial Excitation (sSE)

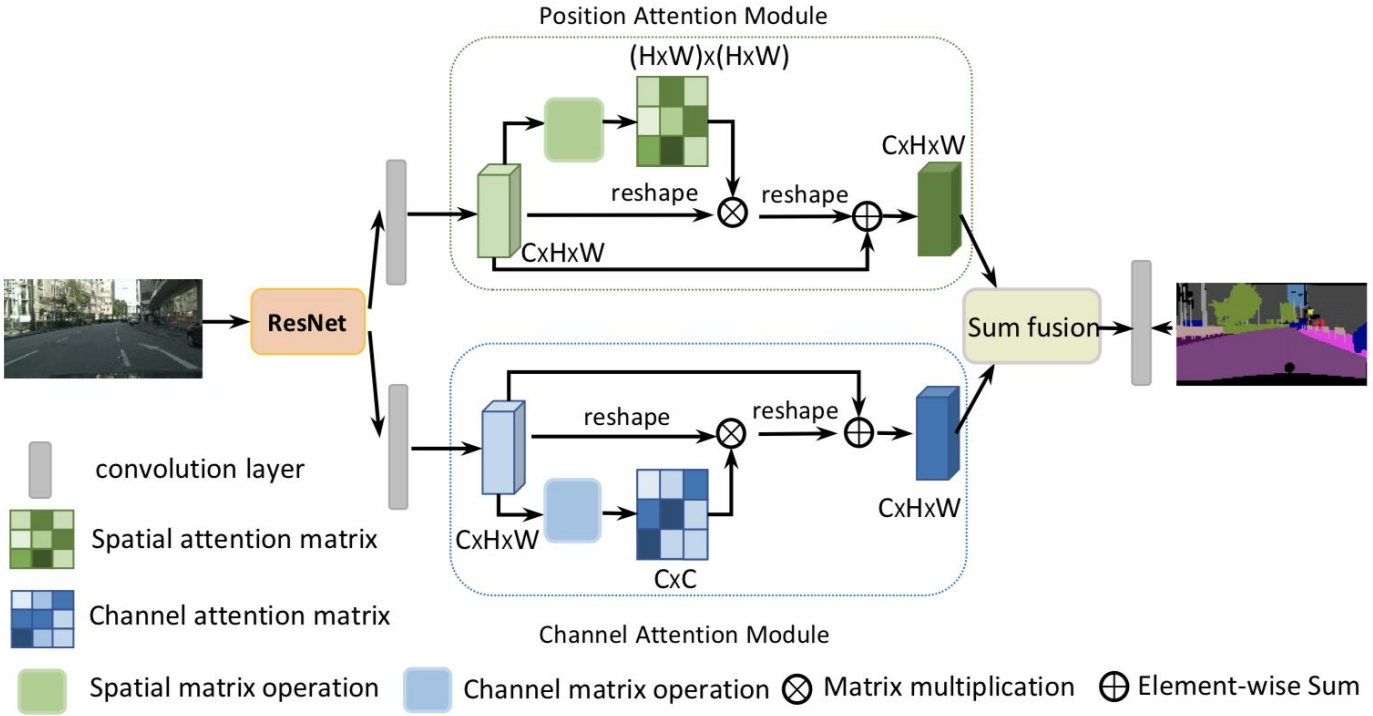


(d) Concurrent Spatial and Channel
Squeeze and Channel Excitation (scSE)



[Roy et al. 2018]

Dual Attention Network (DANet)



Fu et al. 2019

Cross-entropy

$$\mathrm{CE}\left(p, \hat{p}\right) = -\left(p \log(\hat{p}) + (1-p) \log(1-\hat{p})\right)$$

Weighted Cross-entropy

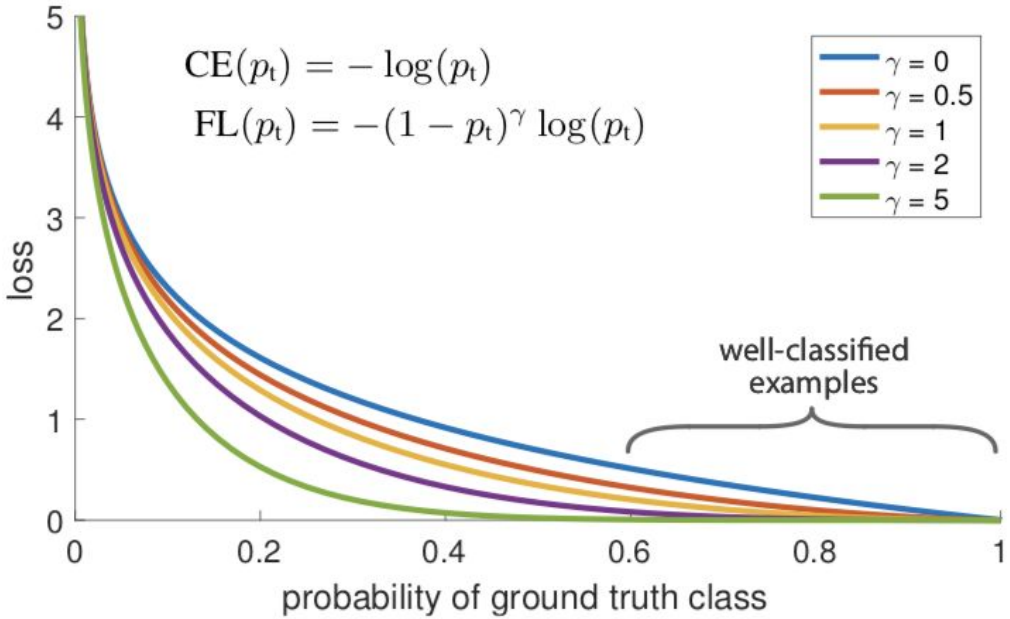
$$\mathrm{WCE}\left(p, \hat{p}\right) = -\left(\beta p \log(\hat{p}) + (1-p) \log(1-\hat{p})\right)$$

Balanced Cross-entropy

$$\mathrm{BCE}\left(p, \hat{p}\right) = -\left(\beta p \log(\hat{p}) + (1-\beta)(1-p) \log(1-\hat{p})\right)$$

Focal Loss

$$\mathrm{FL}\left(p, \hat{p}\right) = -\left(\alpha(1-\hat{p})^{\gamma} p \log(\hat{p}) + (1-\alpha)\hat{p}^{\gamma}(1-p)\log(1-\hat{p})\right)$$



Lin et al. 2017

Dice, Jaccard and Tversky Losses

$$\mathrm{DC} = \frac{2TP}{2TP + FP + FN} = \frac{2|X \cap Y|}{|X| + |Y|}$$

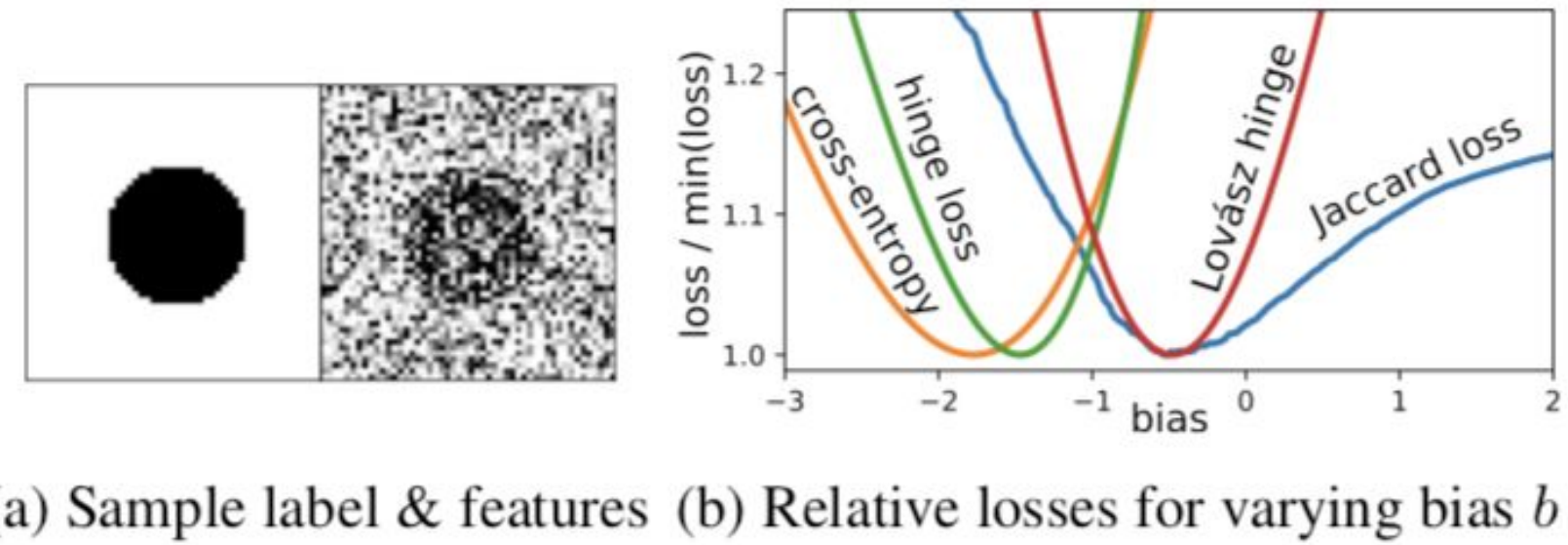
$$\mathrm{IoU} = \frac{TP}{TP + FP + FN} = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}$$

$$\mathrm{DL}\left(p, \hat{p}\right) = 1 - \frac{2p\hat{p} + 1}{p + \hat{p} + 1} \qquad \mathrm{TI}\left(p, \hat{p}\right) = \frac{p\hat{p}}{p\hat{p} + \beta(1-p)\hat{p} + (1-\beta)p(1-\hat{p})}$$

Milletari et al. 2016

Salehi et al. 2017

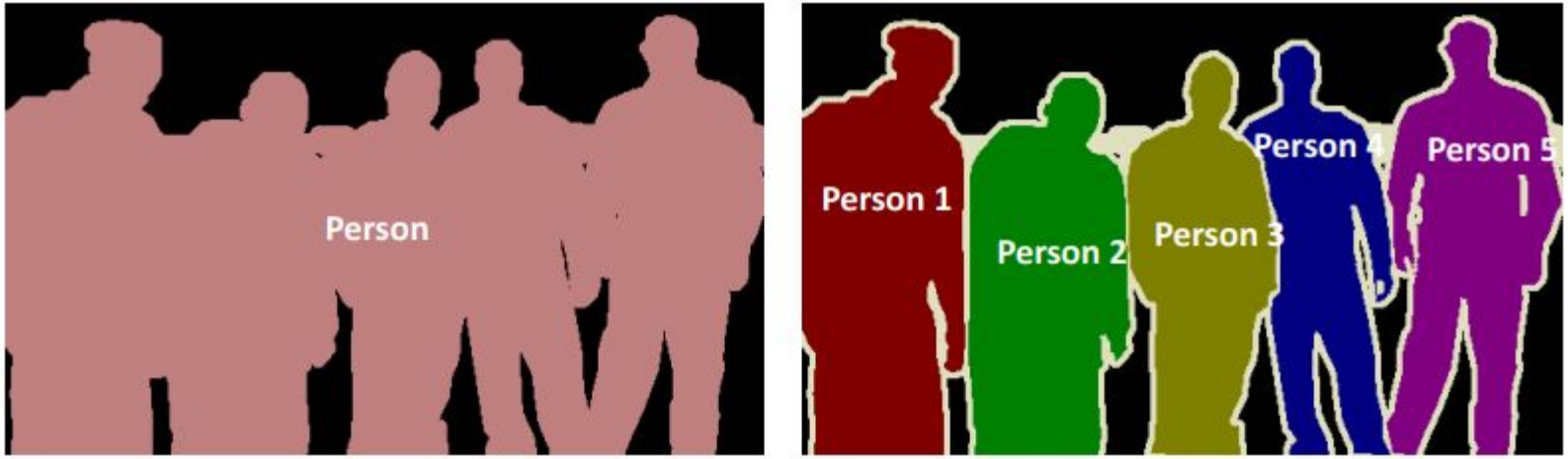
Lovasz Loss



(a) Sample label & features (b) Relative losses for varying bias b

Berman et al. 2018

Instance Segmentation

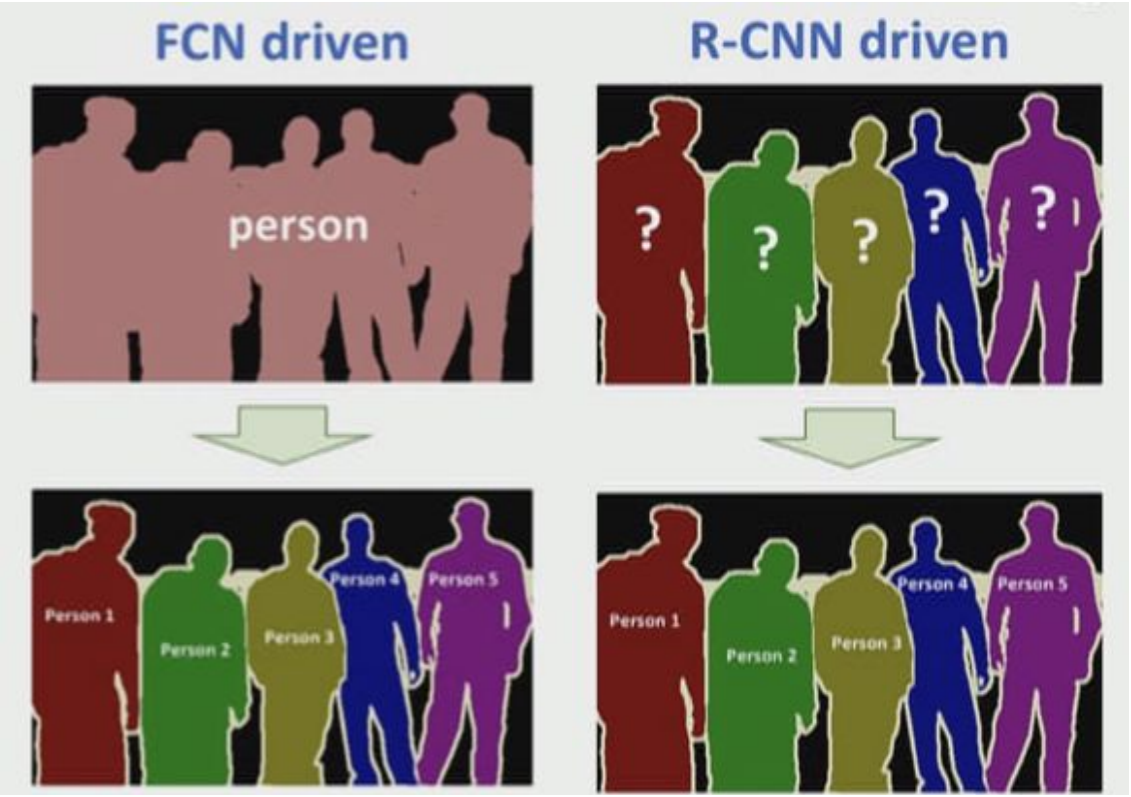


Semantic Segmentation

Instance Segmentation

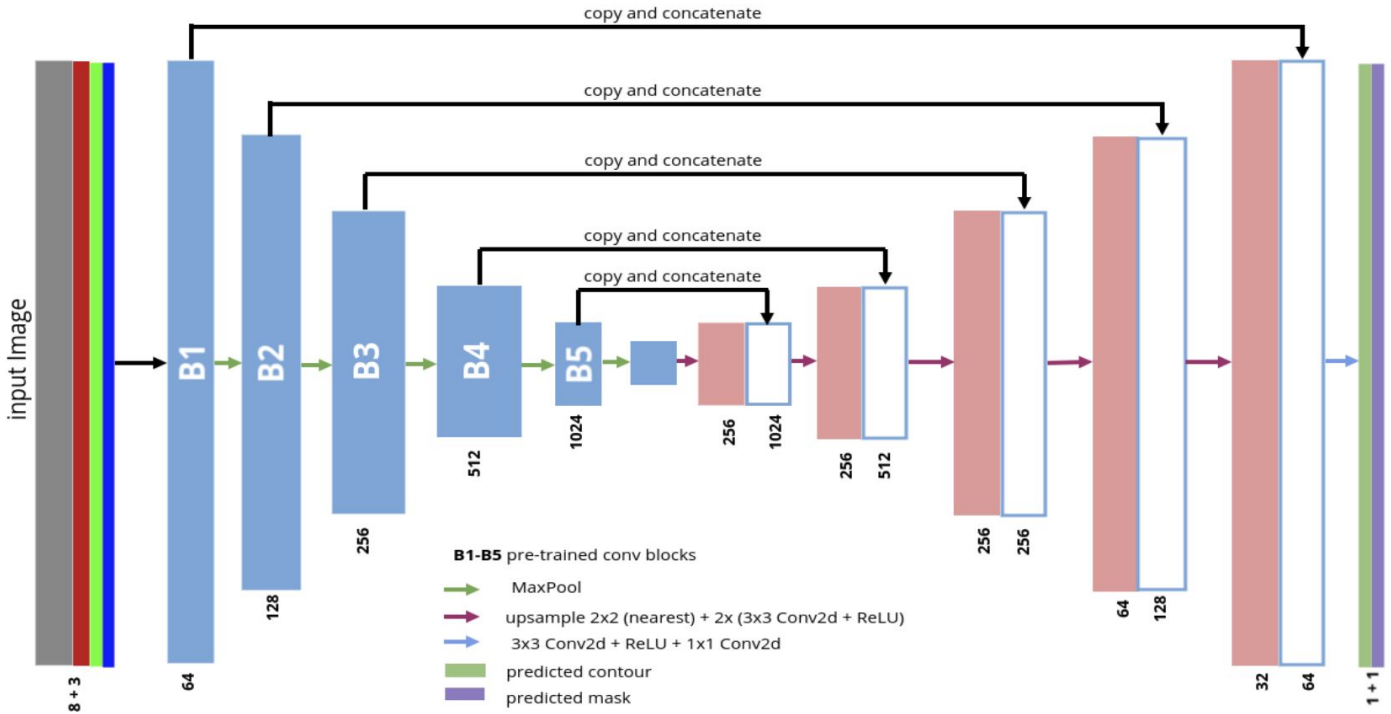
- Detect instances, give category, label pixels
- “Simultaneous detection and segmentation”

Instance Segmentation Methods



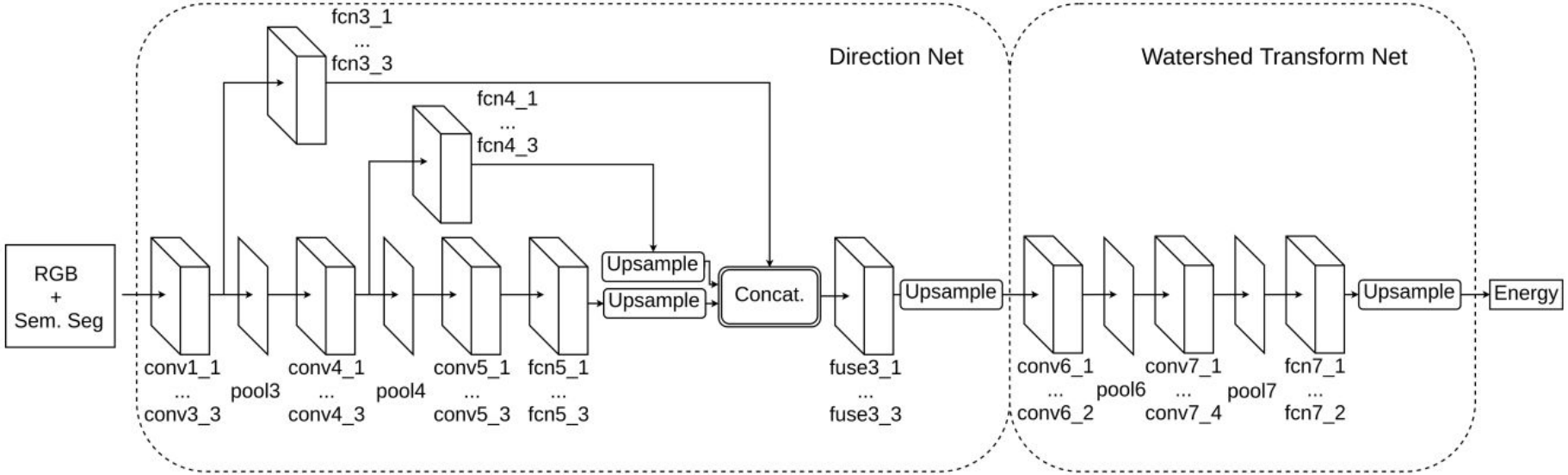
- RCNN driven methods: start from segmentation level proposals, then train a classifier to classify these proposals into somatic categories.
- FCN driven methods: start from a segmentation result, then learn how divide the results into individual instances.

TernausNetV2



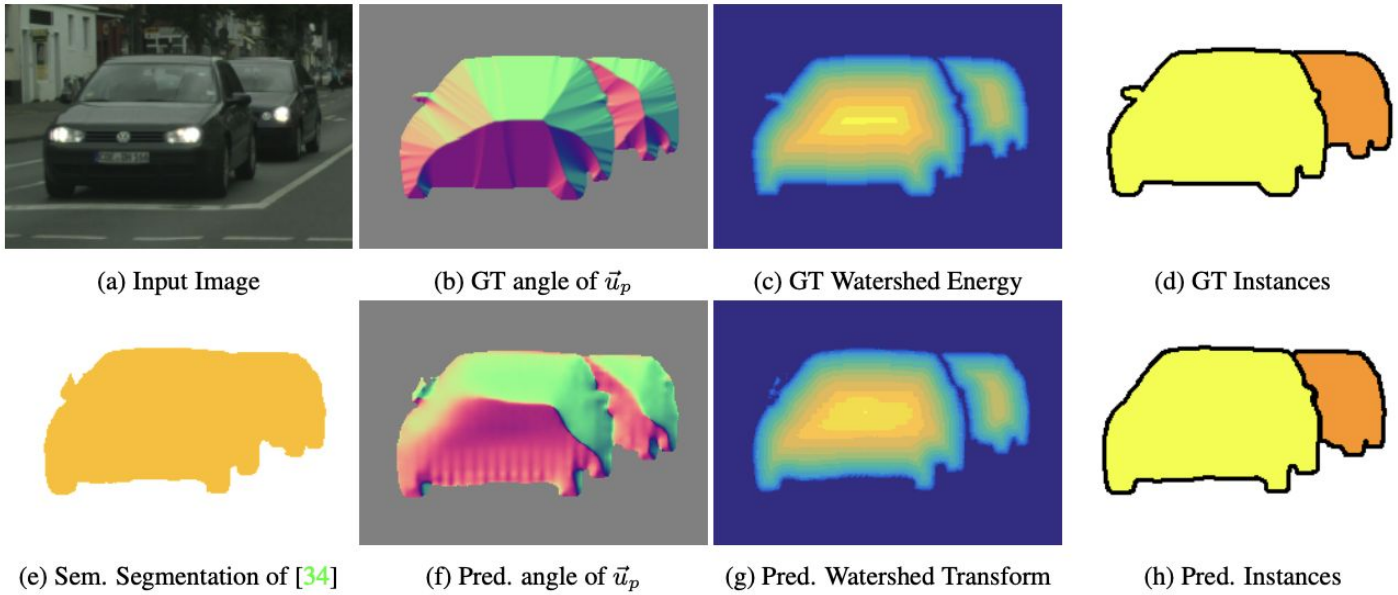
[Iglovikov et al. 2018]

Deep Watershed Transform



Bai et al. 2017

Deep Watershed Transform

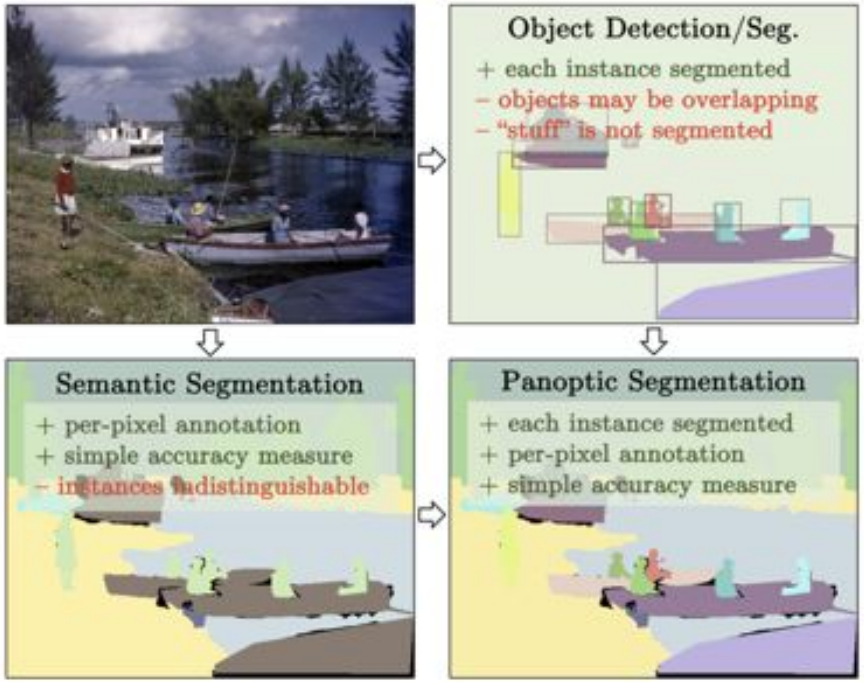


(a) Input Image (b) GT angle of $\vec{u}_p$ (c) GT Watershed Energy (d) GT Instances

(e) Sem. Segmentation of [34] (f) Pred. angle of $\vec{u}_p$ (g) Pred. Watershed Transform (h) Pred. Instances

Bai et al. 2017

Panoptic Segmentation



Previous attempts to explore the combined task:

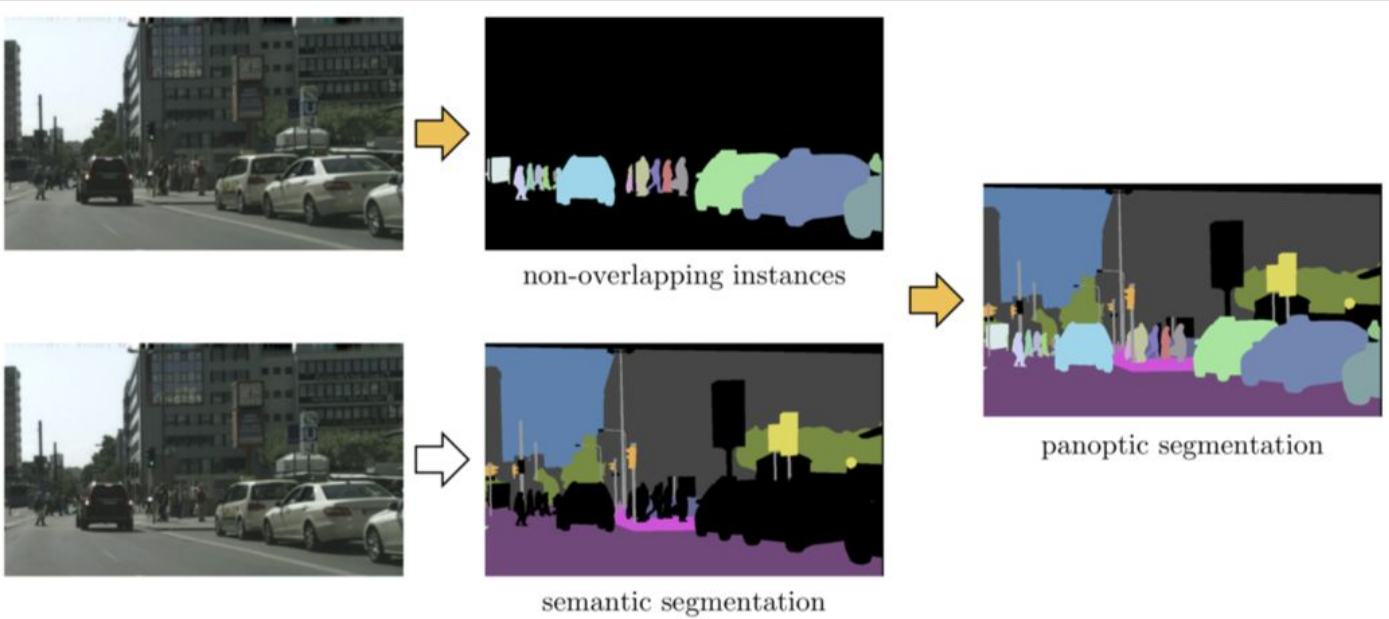
- Scene Parsing
- Image Parsing
- Holistic Understanding

Kirillov et al. 2018

Human Performance: images annotated twice

Cityscapes	Mapillary Vistas	ADE20k	COCO
30 images	46 images	64 images	5000 images

Machine Performance: Baselines



Notes:

- Run instance segmentation independently
- Run semantic segmentation independently
- Simple post-processing to merge

Thank You!

Useful Links

- https://github.com/vdumoulin/conv_arithmetic
- https://github.com/mrgloom/awesome-semantic-segmentation
- https://github.com/kamruleee51/Recommendation-for-understanding-of-semantic-segmentation-using-CNN
- **http://tiny.cc/i96bez**
-

Classwork

Homework

Homework