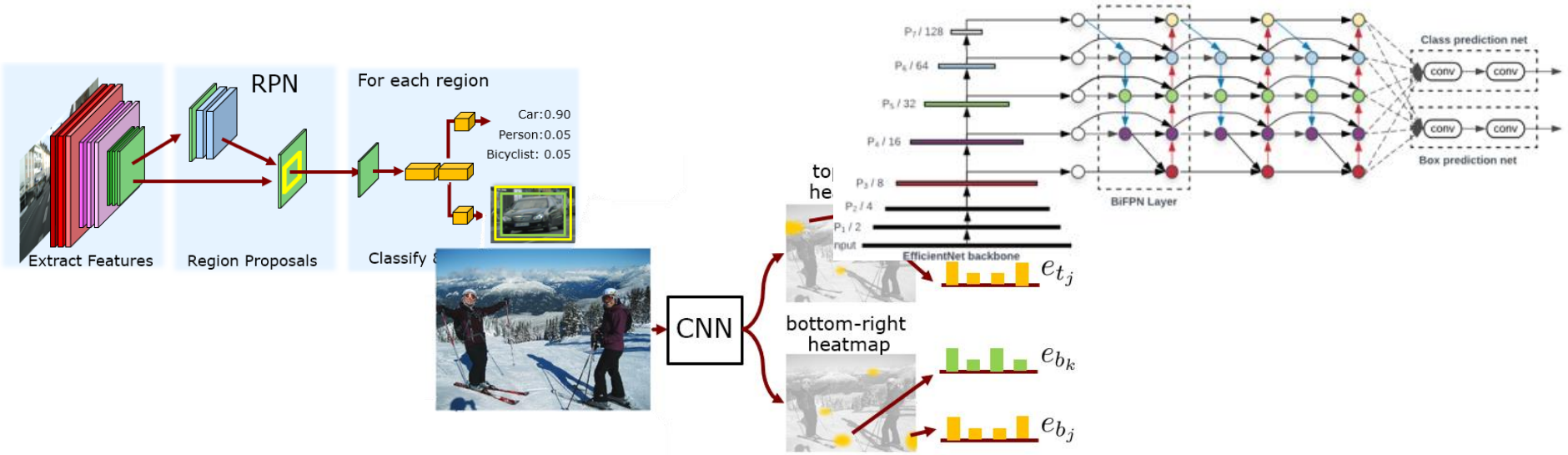
Photogrammetry & Robotics Lab

Machine Learning for Robotics and Computer Vision

Segmentation with CNNs

Jens Behley

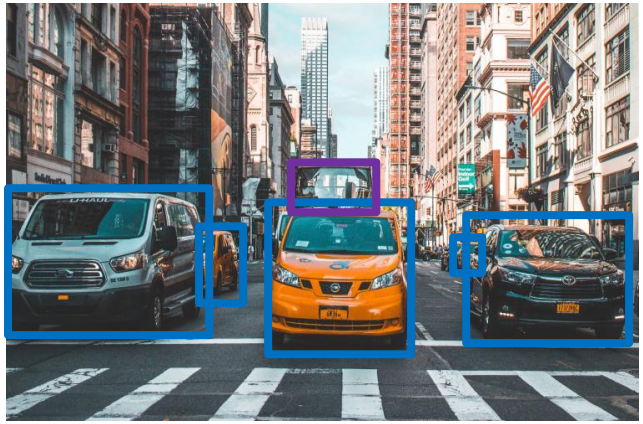
Recap: Last Lecture



- Discussed two-stage and single-stage detectors
- Nowadays, single-stage detectors on-par with two-stage detectors
- Anchor-based vs. anchor-free detectors

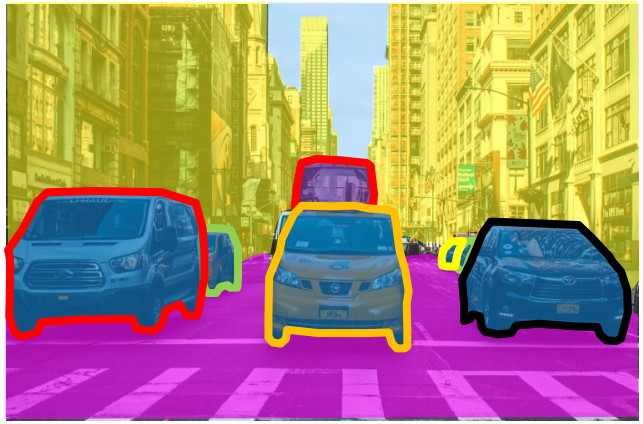
Perception Tasks

Classification



Object Detection

Semantic Segmentation



Panoptic Segmentation

Photo by Marisol Benitez on Unsplash

Semantic Segmentation

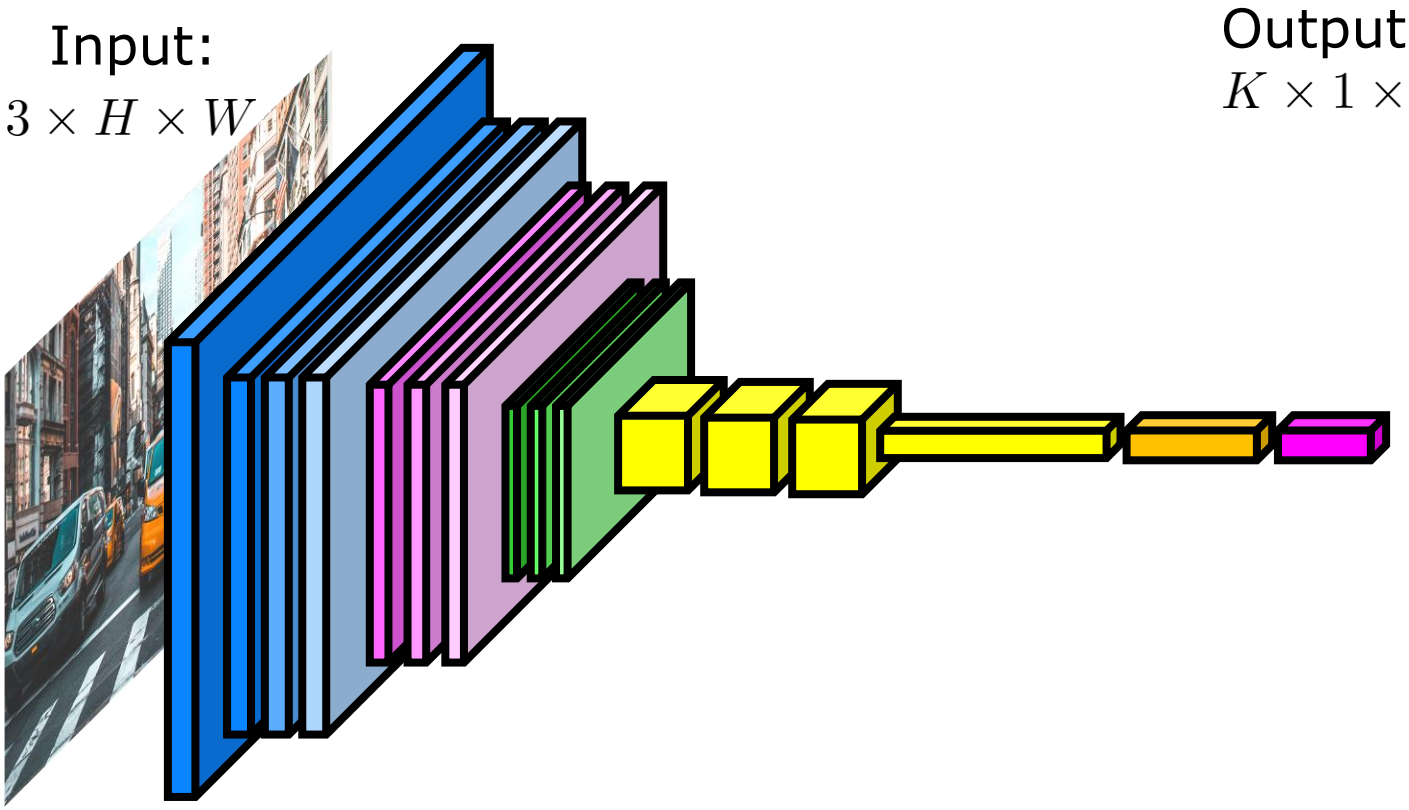
CNN

Building Road Car Bus

- **Goal:** Provide label $y_{i,j} \in \{1, \ldots, K\}$ for each pixel in the image.

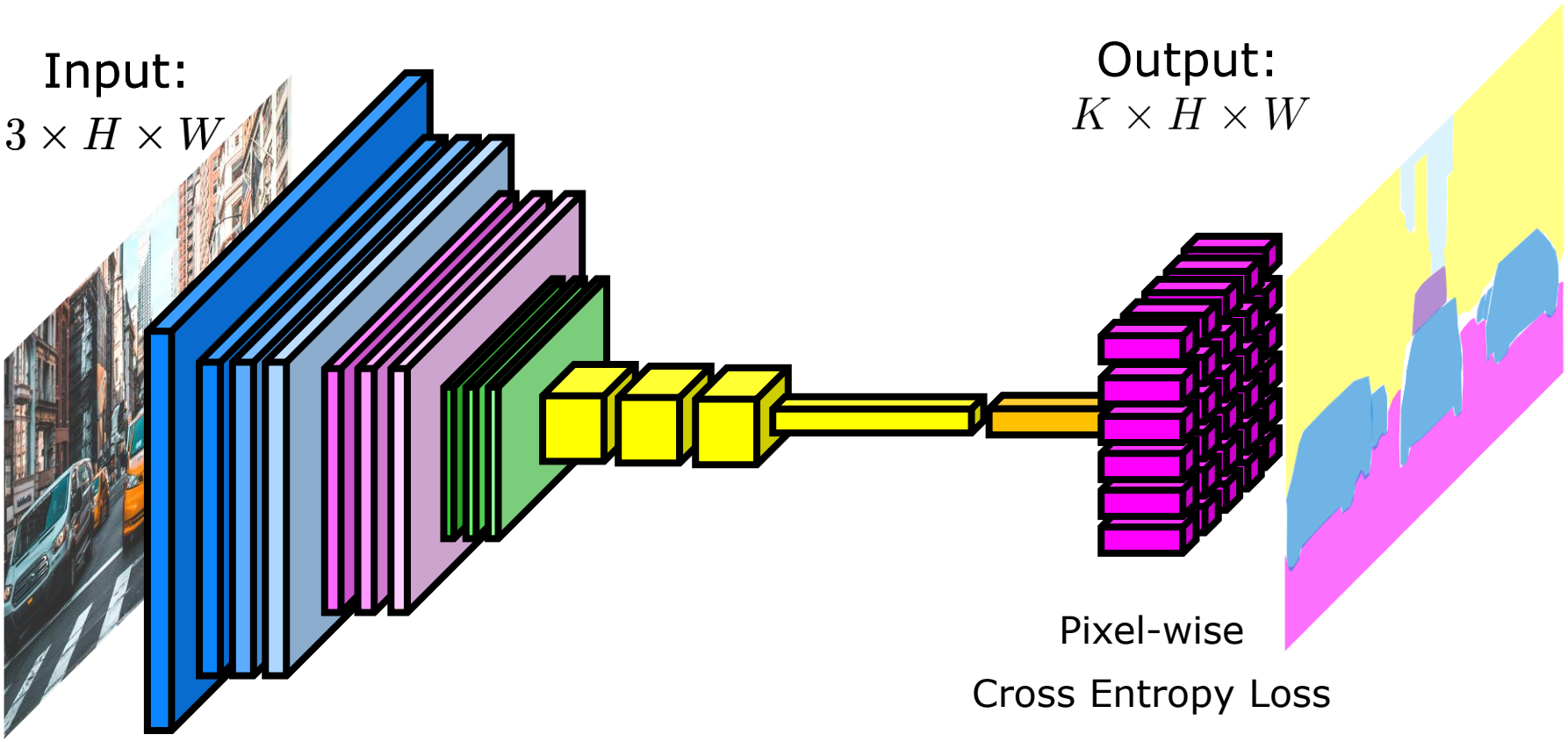
Photo by Marisol Benitez on Unsplash

Image Classification



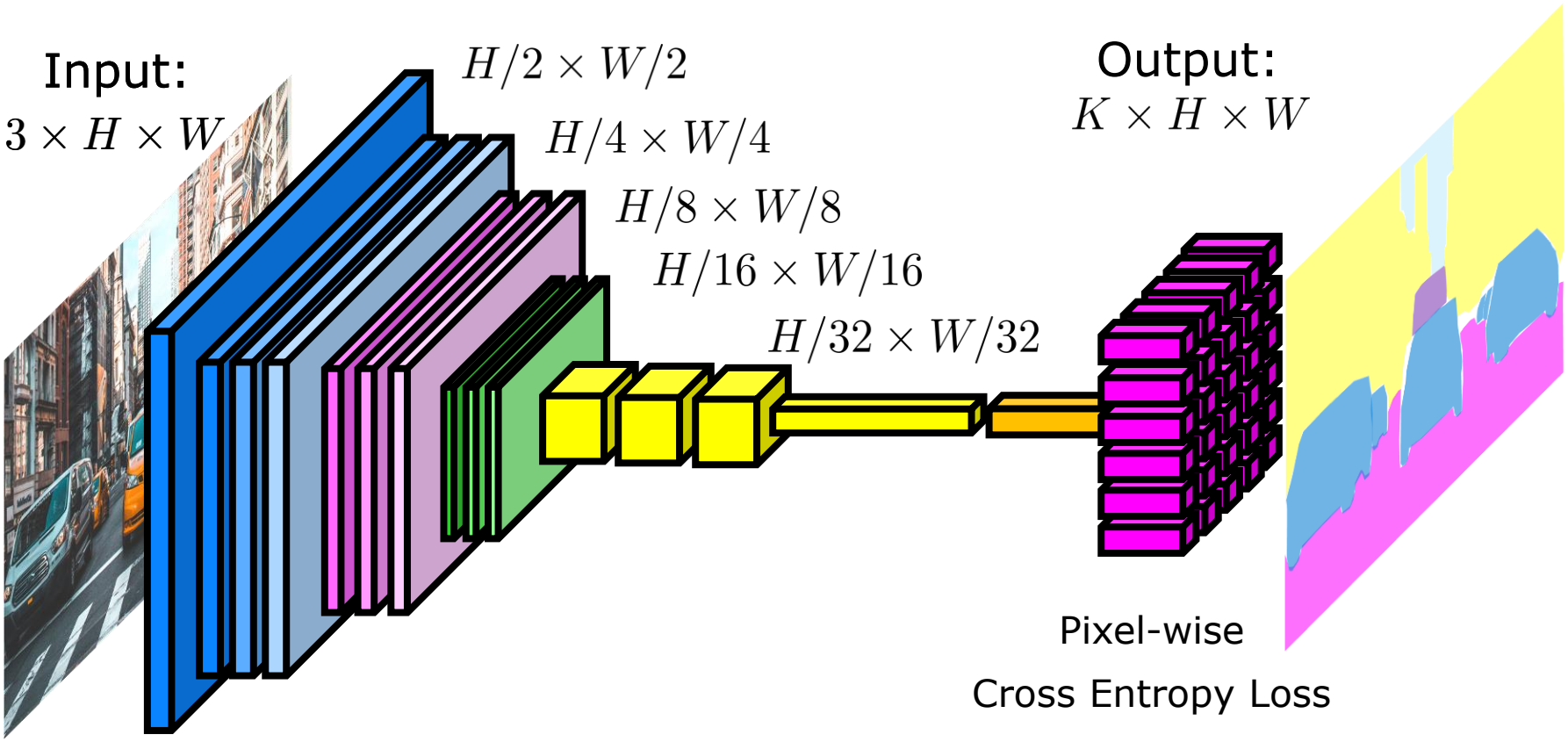
- In image classification, we produce K scores

Semantic Segmentation



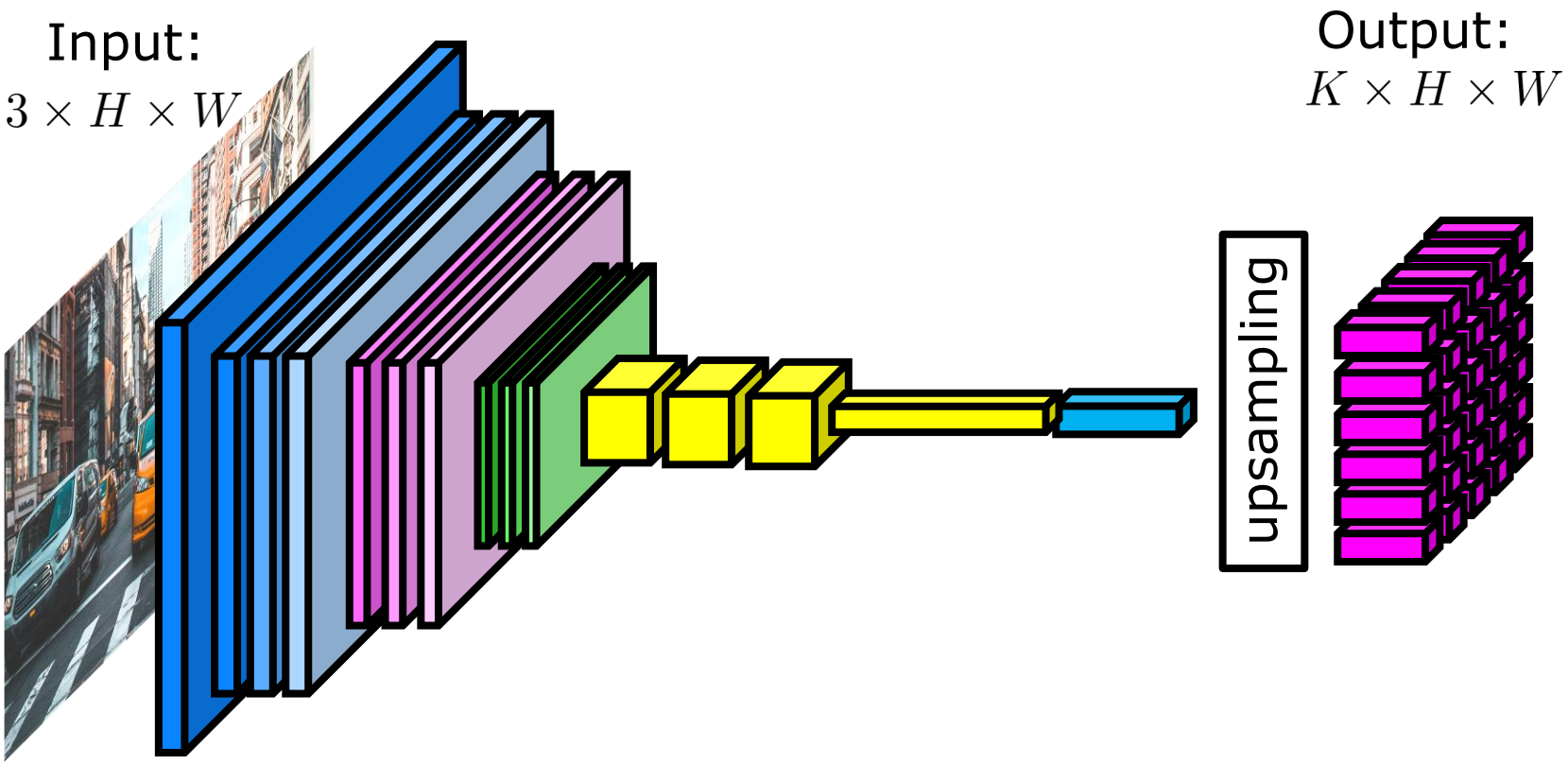
- Correspondingly, in semantic segmentation we want to have K scores for each pixel
- Loss: Pixel-wise cross entropy loss with $K \times H \times W$

Pixel-wise Predictions?



- **Problem:** After P pooling/strided convolutions, last convolutional layer has size $H/2^P \times W/2^P$

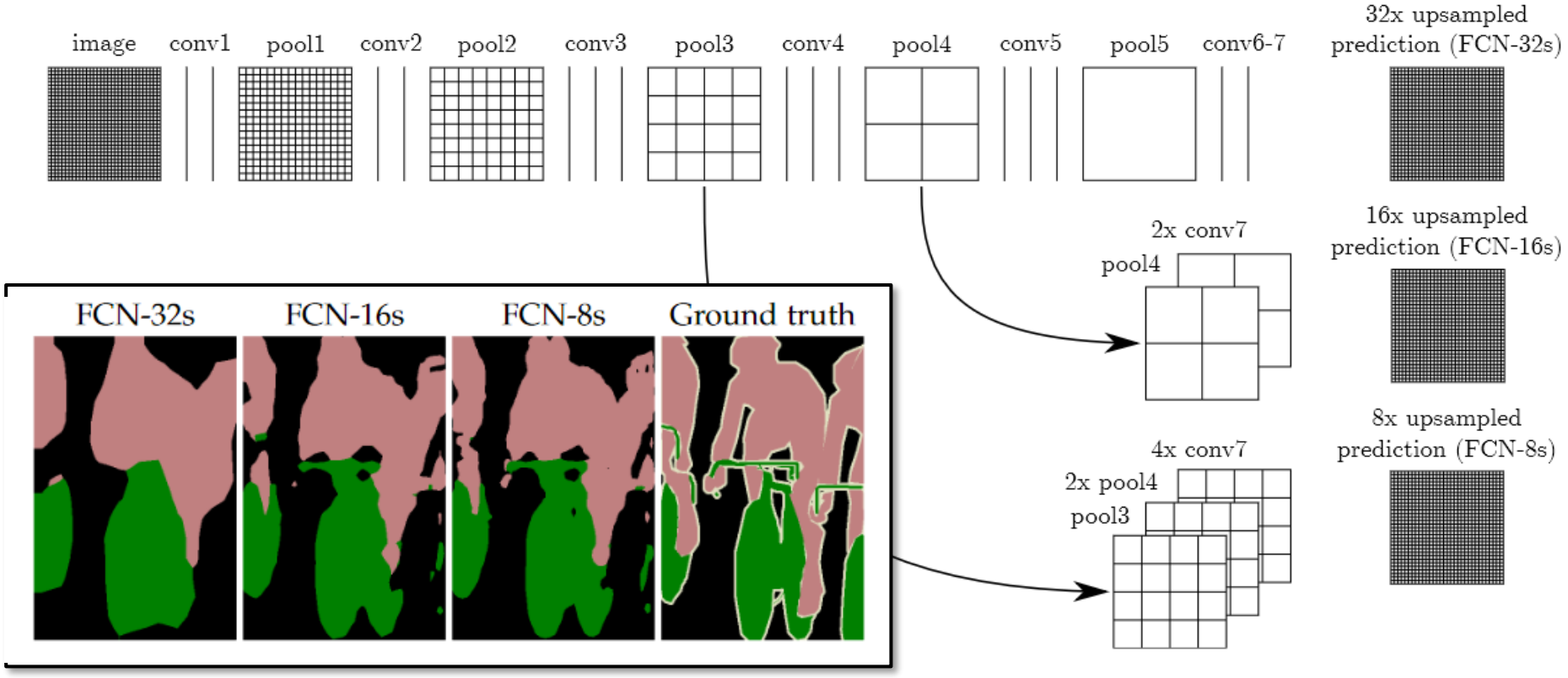
Fully Convolutional Net



- **Idea:** Up-sample last conv layer outputs
- Replace FC layer with 1x1 convolutions to enable arbitrary image sizes → FCN

[Long, 2014][Shelhammer, 2016]

Upsampling in FCN

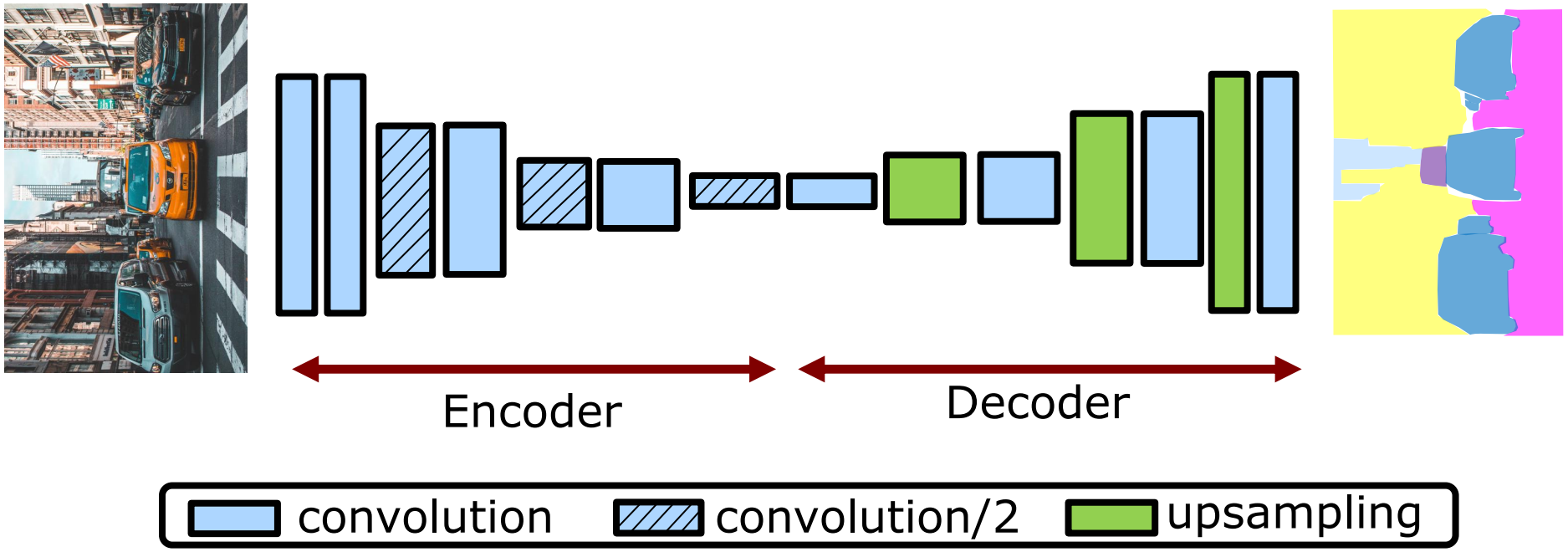


- Use earlier layer outputs in up-sampling to retain fine details
- Upsampling via bilinear interpolation/transpose conv

[Shelhammer, 2016]

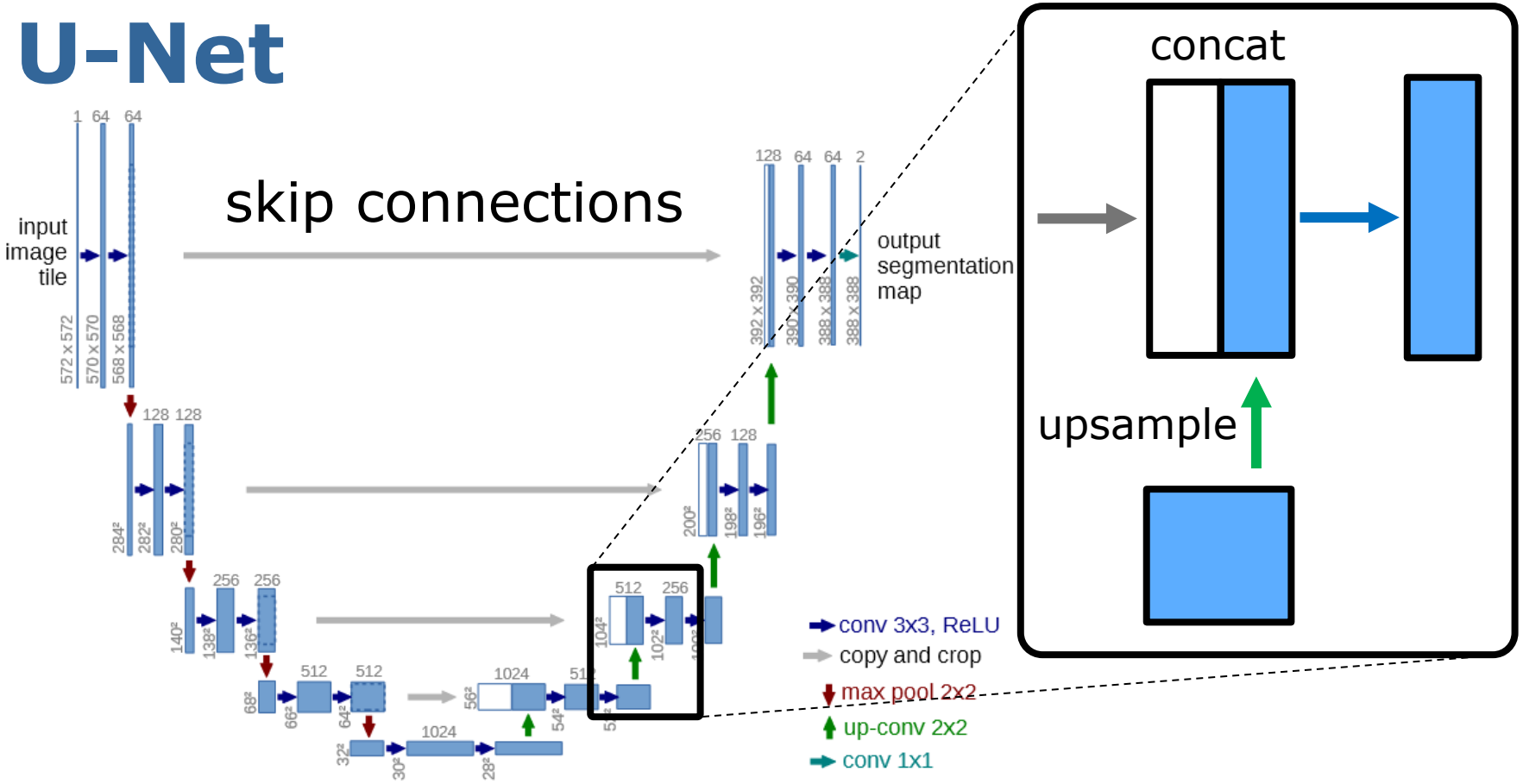
Figures from [Shelhammer, 2016]

Encoder-Decoder Architecture



- Combine up-sampling with convolutional layers to regain spatial resolution

U-Net

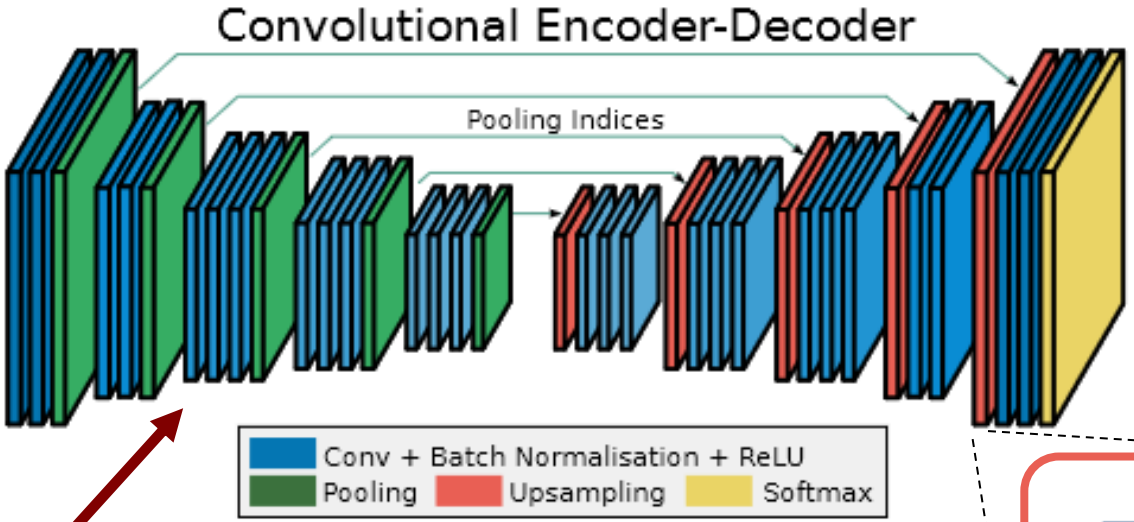


- Skip connections help to retain fine-grained results
- Concatenate feature volumes from encoder and upsampled feature volumes from decoder
- Convolve to reduce number of channels

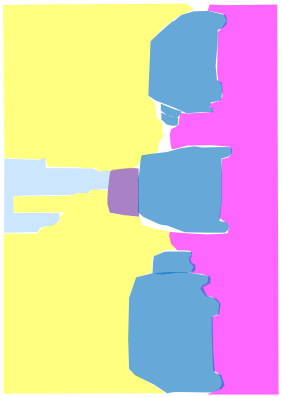
[Ronneberger, 2015]

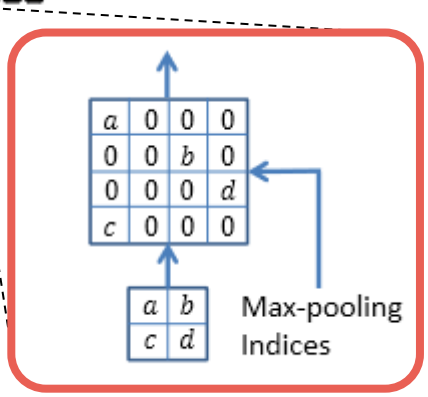
Figures from [Ronneberger, 2015]

SegNet



VGG-16



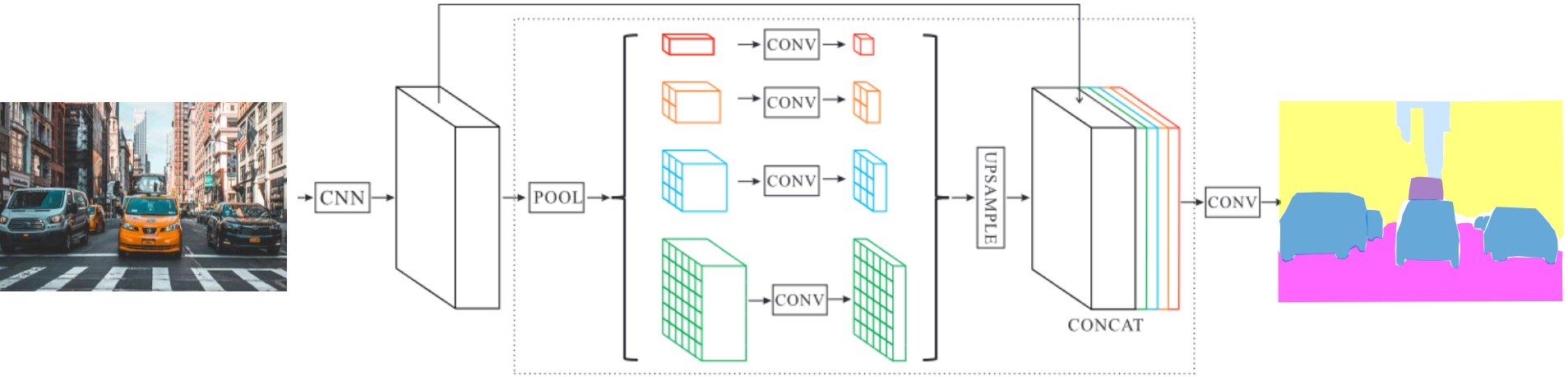


- SegNet uses encoder-decoder architecture
- Upsampling by un-pooling with maximums from max-pool layer of encoder

[Badrinarayanan, 2017]

Figures from [Badrinarayanan, 2017]

Pyramid Scene Parsing Network

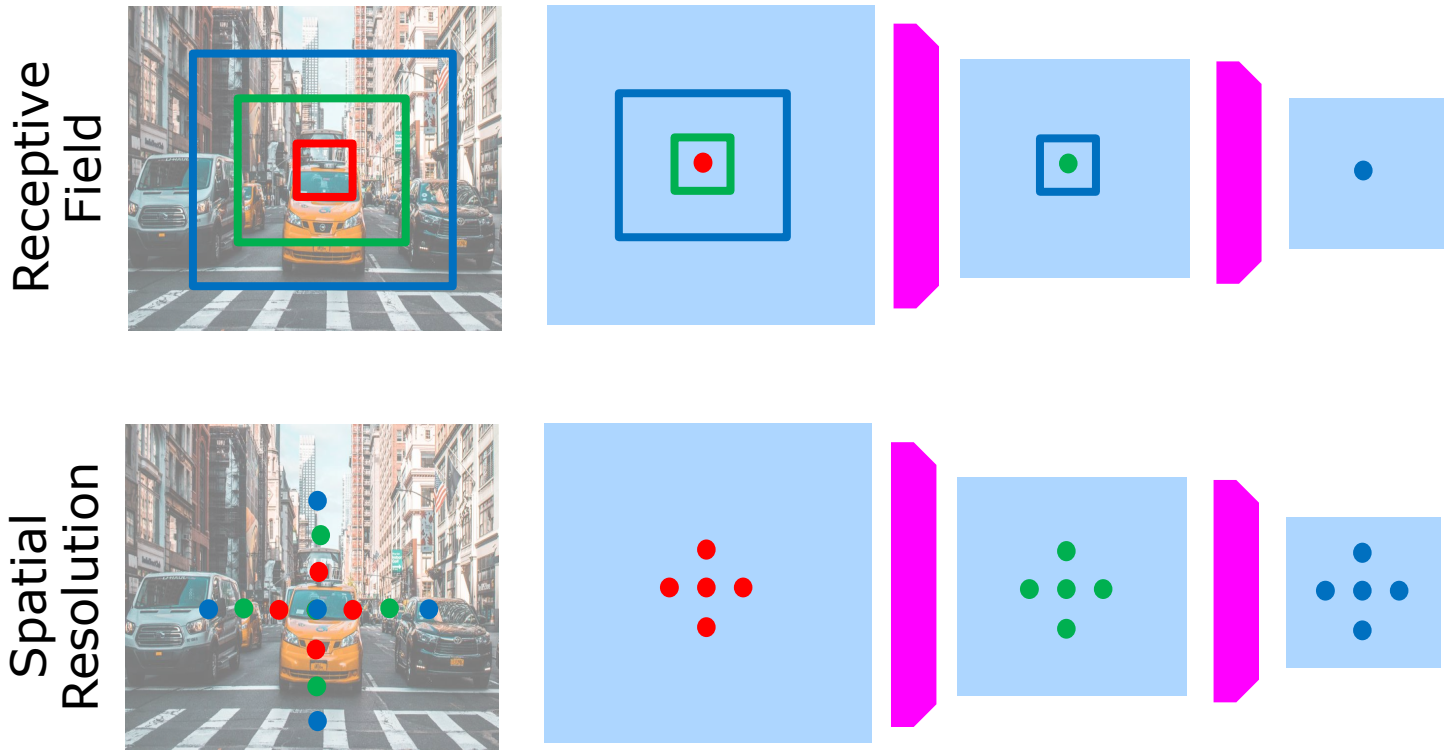


- Generate feature pyramid by pooling operations
- Additional contextual information (larger receptive field of pooled regions)

[Zhao, 2016]

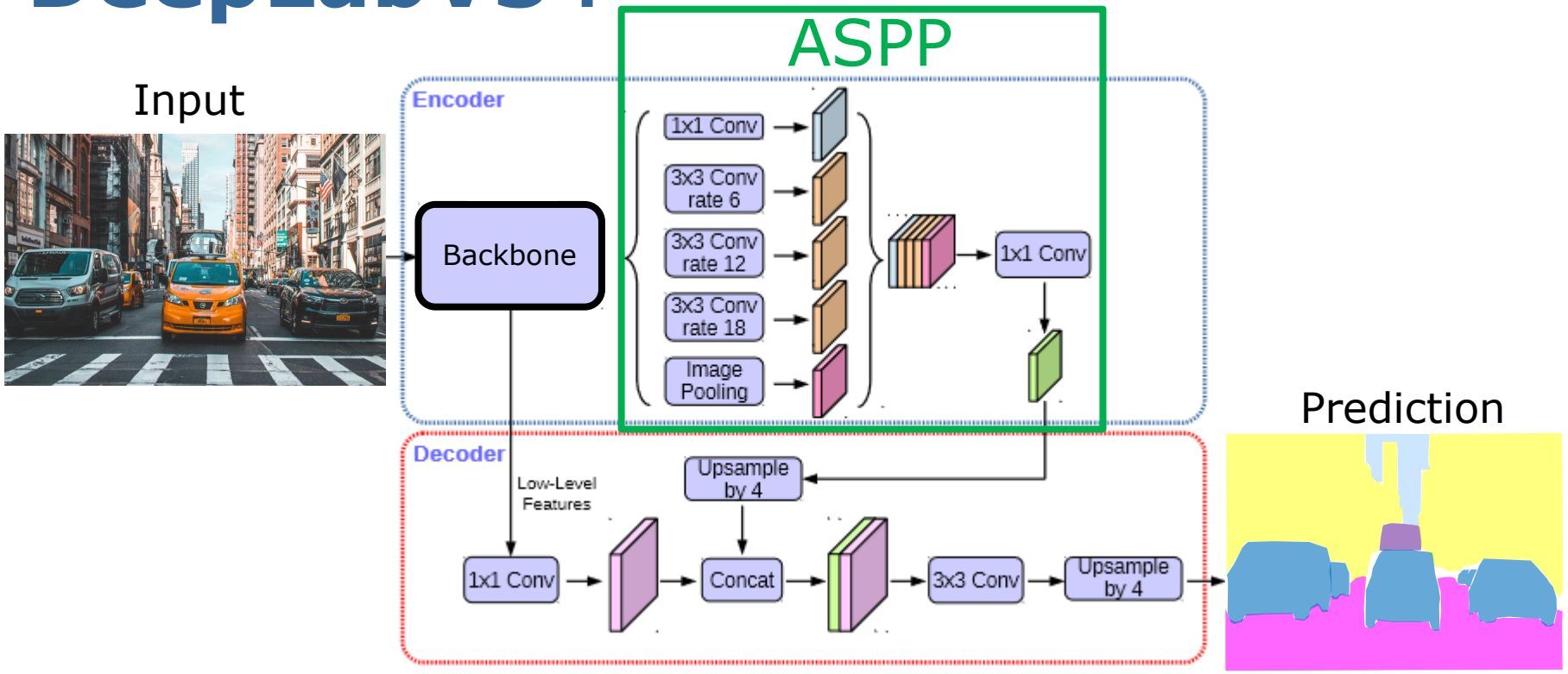
Figures from [Zhao, 2016]

Receptive Field vs. Spatial Resolution



- Max Pooling/Strided convolution effectively increases receptive field of convolutions
- **Problem:** Deeper layers have reduced spatial resolution → Possible solution: Upsampling decoder

DeepLabV3+

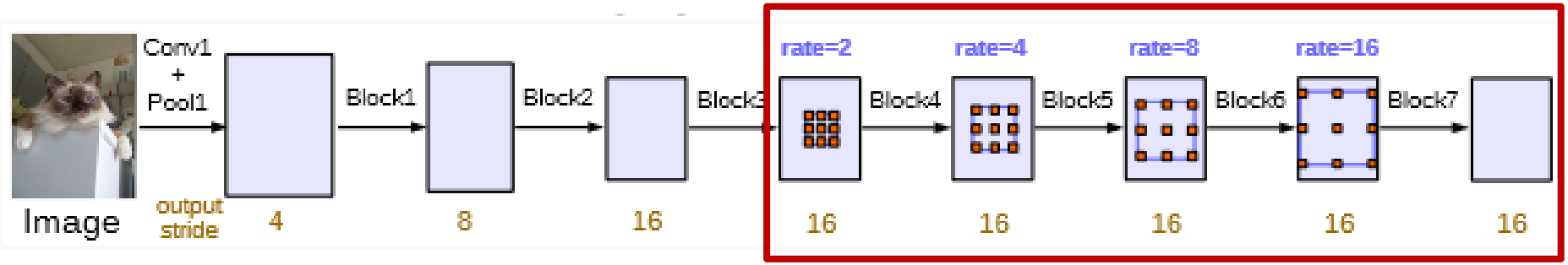


- Line of DeepLab models solves the problem via **dilated convolutions** (atrous convolutions)
- **Atrous Spatial Pyramid Pooling (ASPP)** module

[Chen, 2018]

Figure from [Chen, 2018]

Backbone with Dilated Convs

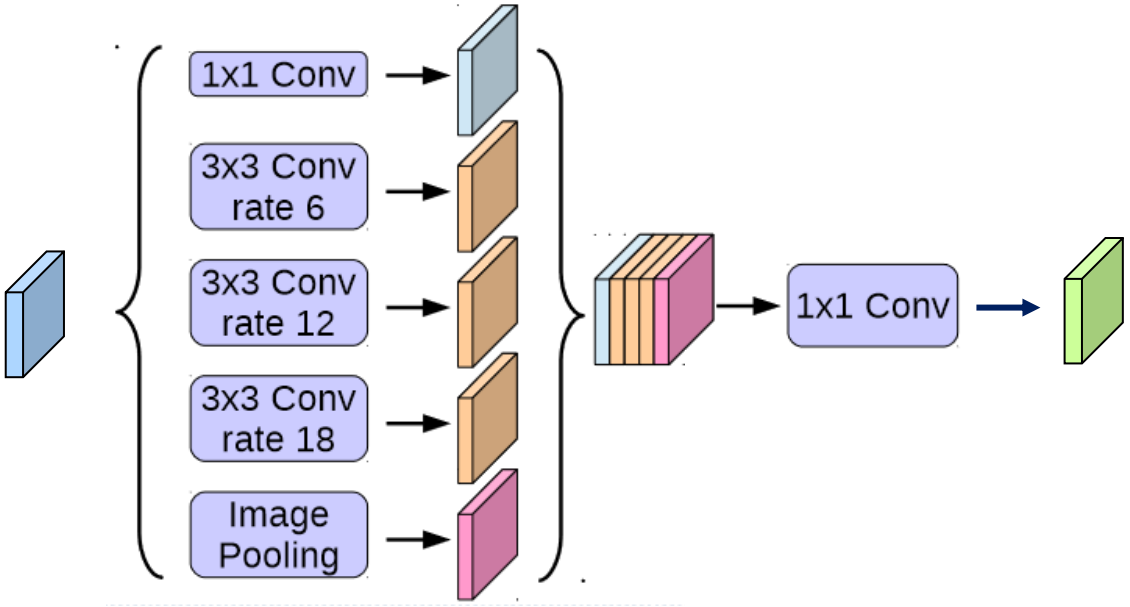


- **Output stride** corresponds to stride of feature map in respect to input image
- Use **dilated convolutions** in later layers to increase receptive field at same output stride

[Chen, 2018]

Figure from [Chen, 2018]

Atrous Spatial Pyramid Pooling

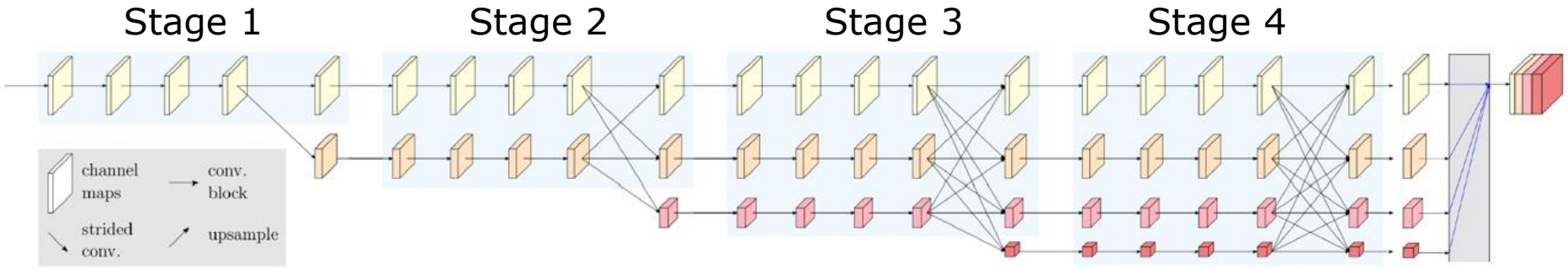


- PSPNet used different pooling operation to capture context
- Here, apply different *dilated* convolution on last conv layer output
- Capture contextual features at different scales

[Chen, 2018]

Figure from [Chen, 2018]

HRNet

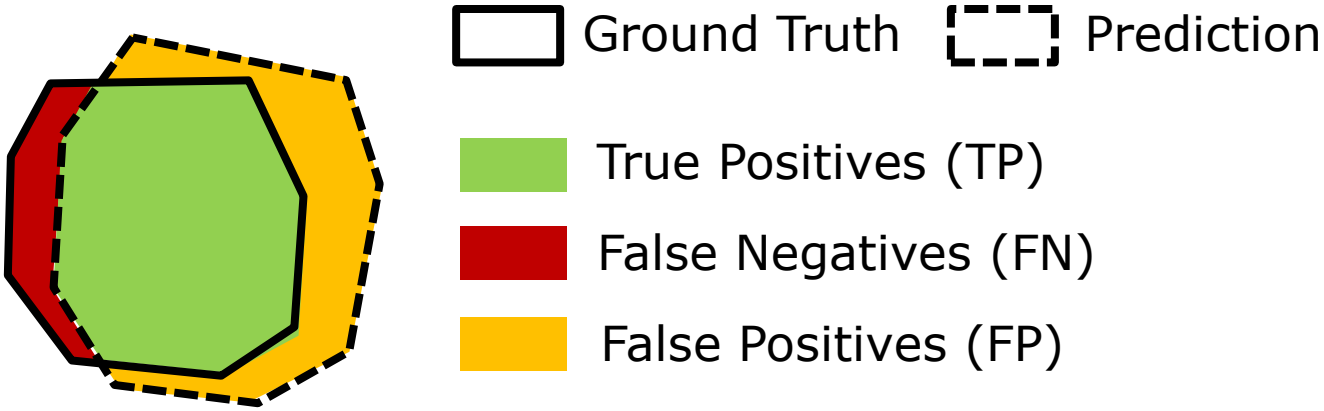


- Multiple branches at different strides
- Fusion of branch outputs in every stage to keep high-resolution information
- Downsampling: strided convolutions
  Upsampling: bilinear interpolation

[Wang, 2020]

Figure from [Wang, 2020]

Evaluation Metric: mIoU



$$\text{IoU}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c + \text{FN}_c} \qquad \text{mIoU} = \frac{1}{K} \sum_c \text{IoU}_c$$

- For each class, determine pixe-wise intersection-over-union (IoU)
- Mean over class-wise IoUs gives mean Intersection-over-union **(mIoU)**

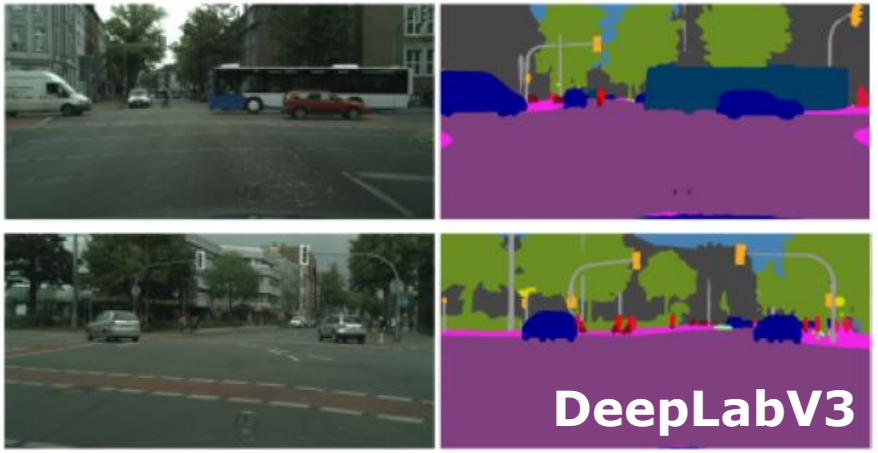
Common Datasets

name	#categories	#images	task
MS COCO (2014)	80	118k	I,P
Cityscapes (2016)	30(19)	5k	S,I,P
Mapillary Vistas (2017)	66	25k	S,I
ADE20K (2017)	2,693(150)	25k	S,I

S = Semantic Seg., I = Instance Seg., P = Panoptic Seg.

[Lin, 2014][Cordts, 2016][Neuhold, 2017][Zhou, 2017]

Results on ADE20K/Cityscapes



Approach	mIoU
SegNet	21.6
FCN-8s	29.4
HRNetV2	43.2
PSPNet	44.9

ADE20K validation set

Approach	mIoU
SegNet	57.0
FCN-8s	65.3
PSPNet	80.2
HRNetV2	81.8
DeepLabV3+	82.1

Cityscapes test set

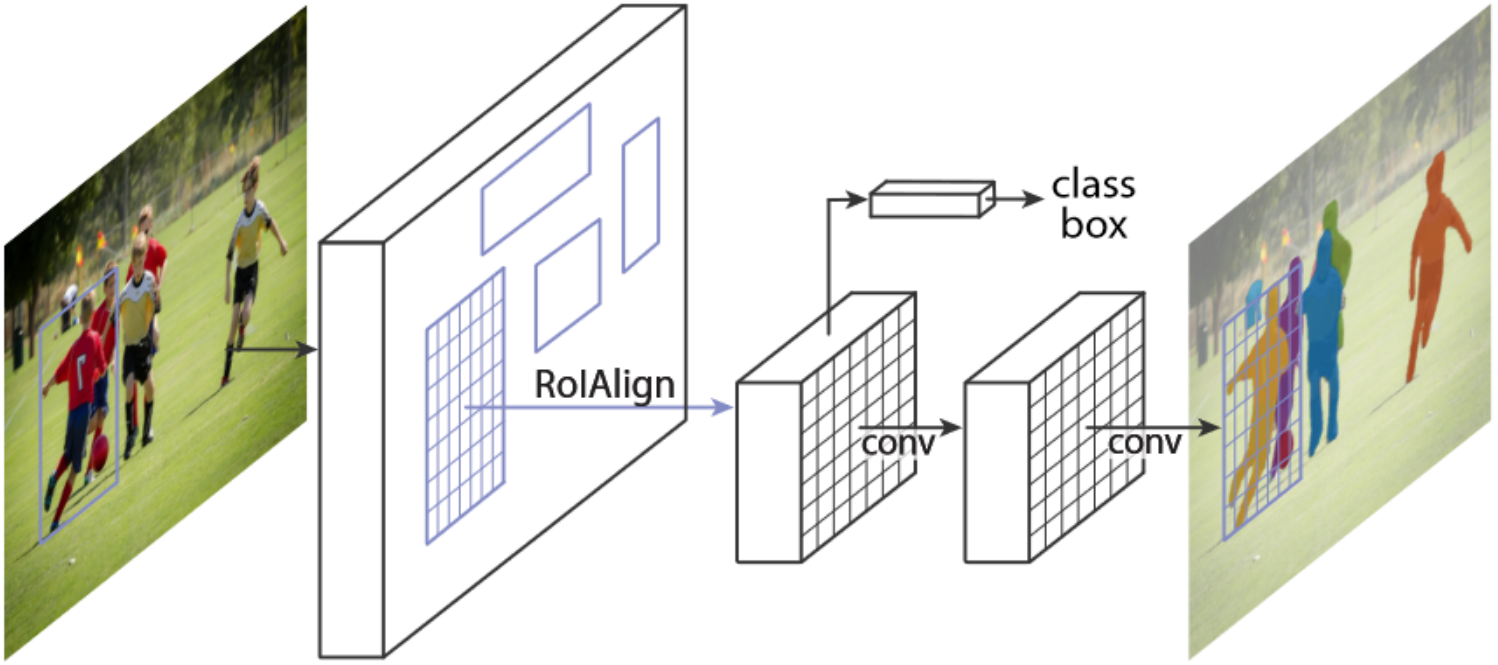
Instance Segmentation

Semantic Segmentation

Instance Segmentation

- Semantic segmentation provides only class labels
- Instance segmentation aims at distinguishing different instances of the same object class

Mask R-CNN

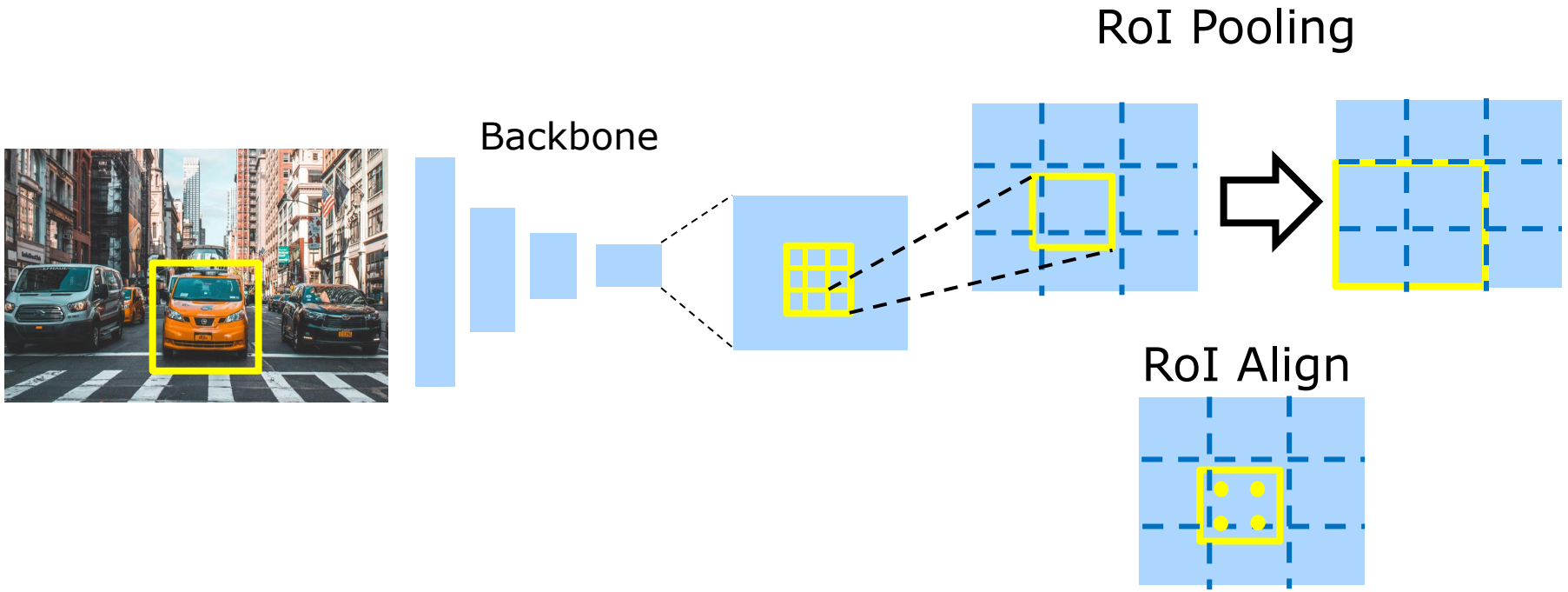


- **Idea:** Localize objects and determine binary mask
- Add mask prediction head to Faster R-CNN
- Determine foreground/background for each pixel of Region-of-Interest (RoI)

[He, 2017]

Figure from [He, 2017]

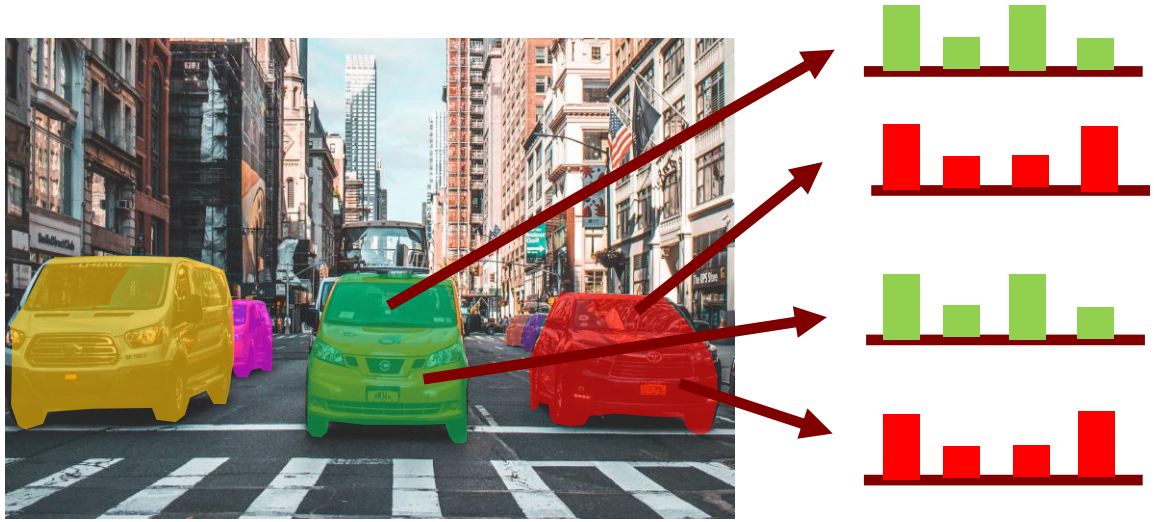
RoI Pooling vs. RoI Align



- RoI pooling from Faster RCNN suffers from rounding/discretization errors
- RoI Align uses real coordinates and bilinear interpolation to compute RoI feature map at sample locations

[He, 2017]

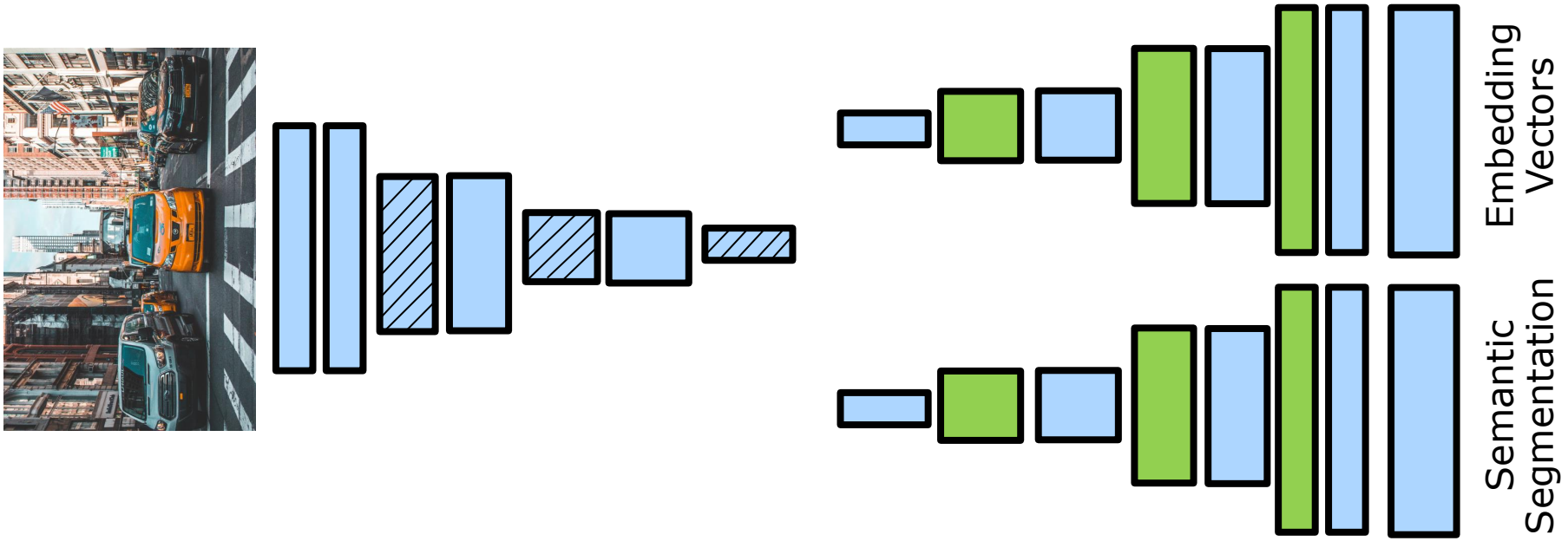
Embedding-based Association



- **Idea:** Learn embedding vectors such that vectors of same instance are closer then embedding vectors of different instance
- **Cluster** embedding vectors to determine instances

[De Brabandere, 2017]

Multi-class Embeddings



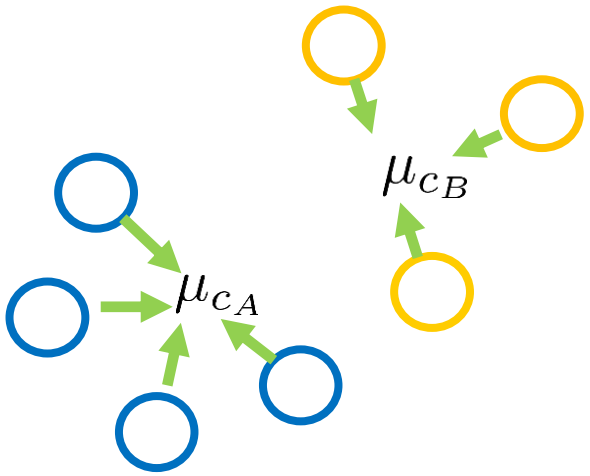
- Shared encoder, separate decoder for instances and semantic segmentation
- Per-class embeddings are optimized to be distinct

[De Brabandere, 2017]

Embedding Loss

$$[\cdot]_+ = \max(0, x)$$

$$L_{var} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}[\|\mu_c - x_i\| - \delta_v]_+^2$$



- Embedding vectors of the same instance should have small distance to mean embedding

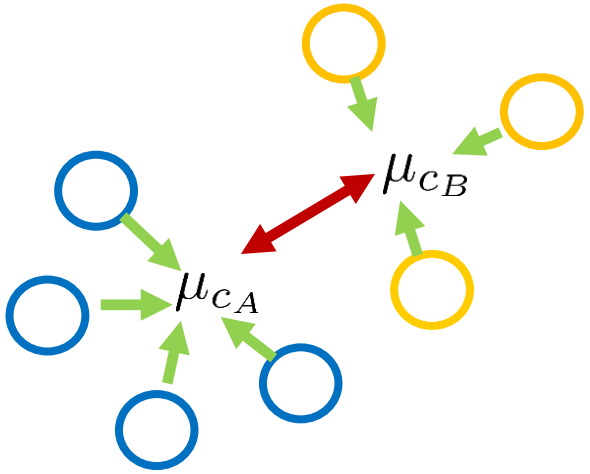
[De Brabandere, 2017]

Embedding Loss

$$[\cdot]_+ = \max(0, x)$$

$$L_{var} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}[\|\mu_c - x_i\| - \delta_v]_+^2$$

$$L_{dist} = \frac{1}{C(C-1)}\sum_{\substack{c_A=1 \\ c_A \neq c_B}}^{C}\sum_{c_B=1}^{C}[2\delta_d - \|\mu_{c_A} - \mu_{c_B}\|]_+^2$$



- Embedding vectors of the same instance should have small distance to mean embedding
- Mean embedding vectors of different instances should have large distance

[De Brabandere, 2017]

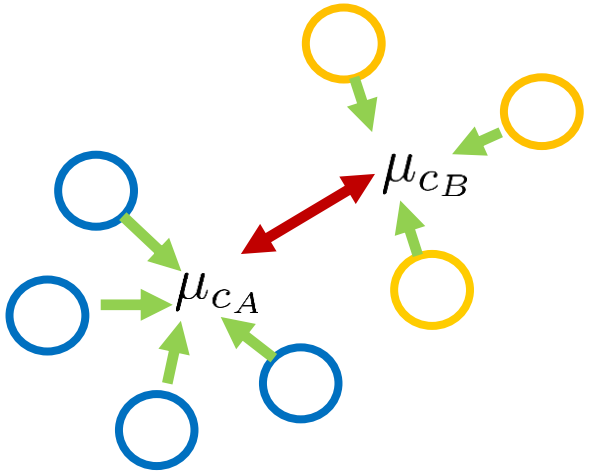
Embedding Loss

$$[\cdot]_+ = \max(0, x)$$

$$L_{var} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}[\|\mu_c - x_i\| - \delta_v]_+^2$$

$$L_{dist} = \frac{1}{C(C-1)}\sum_{\substack{c_A=1 \\ c_A \neq c_B}}^{C}\sum_{c_B=1}^{C}[2\delta_d - \|\mu_{c_A} - \mu_{c_B}\|]_+^2$$

$$L_{reg} = \frac{1}{C}\sum_{c=1}^{C}\|\mu_c\|$$



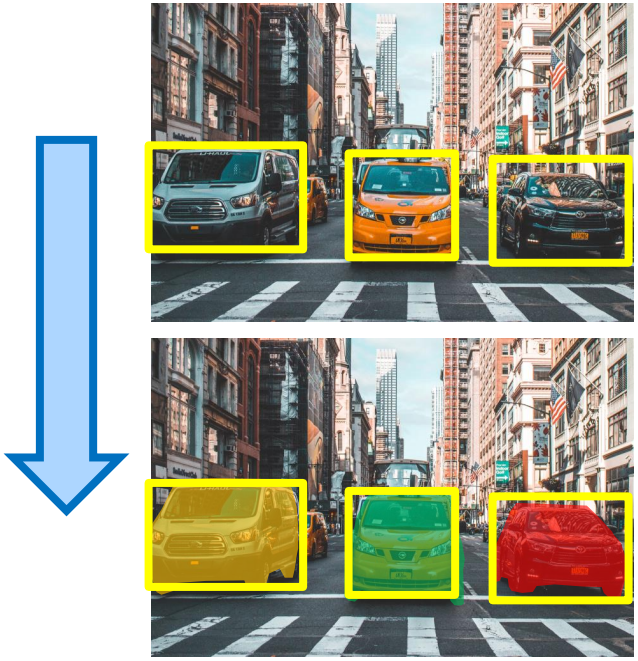
- Embedding vectors of the same instance should have small distance to mean embedding
- Mean embedding vectors of different instances should have large distance
- Regularizer ensures that embeddings are bounded

[De Brabandere, 2017]

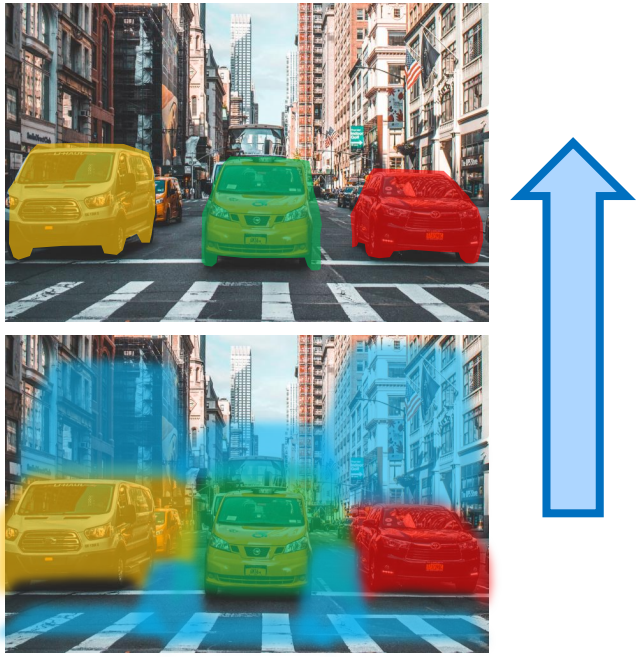
Instance Clustering

- Clustering on embedding vectors via mean-shift variant
  1. Select pixel of specific class and set mean to its embedding.
  2. Get all pixels within embedding distance δ_v to mean.
  3. Recompute mean of thresholded embeddings, repeat 2.

Top-down vs. Bottom-up Instance Segmentation



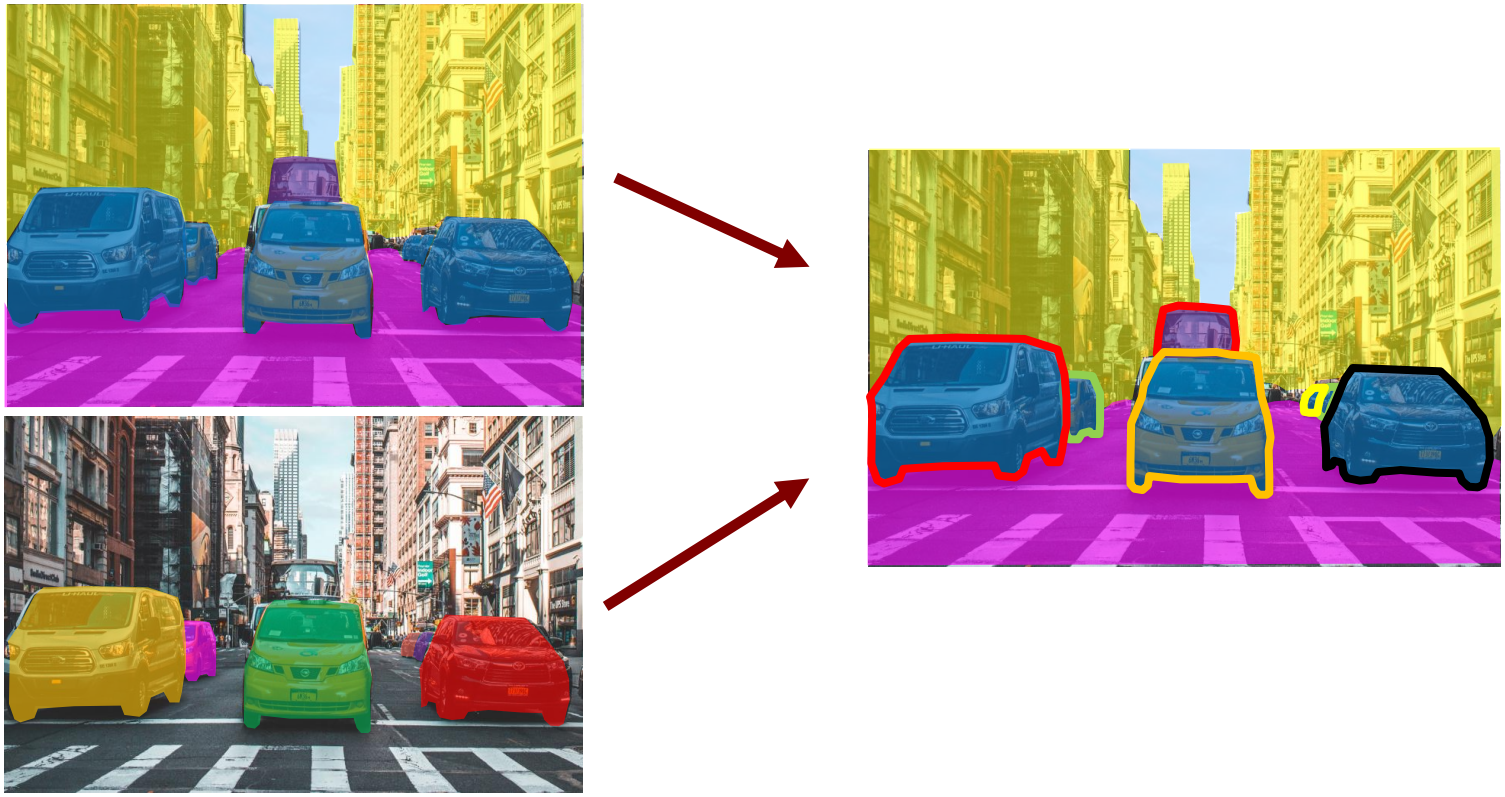
Top-down Approach



Bottom-up Approach

- **Top-down**: Instances are first determined and then foreground/background mask estimated
- **Bottom-up**: Determine per-pixel properties that are then used to cluster instances

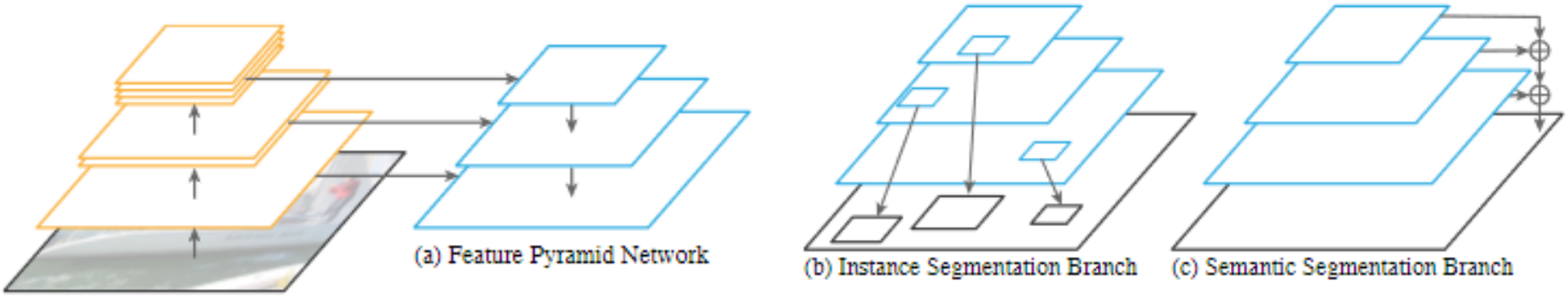
Panoptic Segmentation



- Panoptic Segmentation unifies semantic and instance segmentation
- Distinguish **stuff** (e.g., vegetation, road, …) and **thing** classes (e.g., car, pedestrian, …)

[Kirillov, 2019]

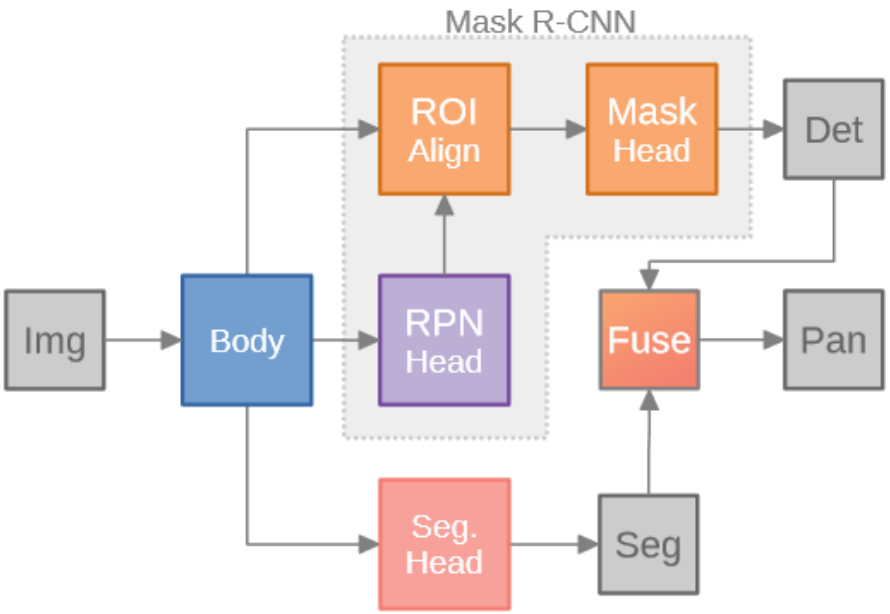
PanopticFPN



- Feature Pyramid Network (FPN)-based instance segmentation (Mask R-CNN)
- Mask R-CNN provides segmentation of things
- Semantic segmentation branch provides segmentation of stuff (+ "thing" class) by upsampling FPN maps

[Kirillov, 2019]

Concurrent Approaches

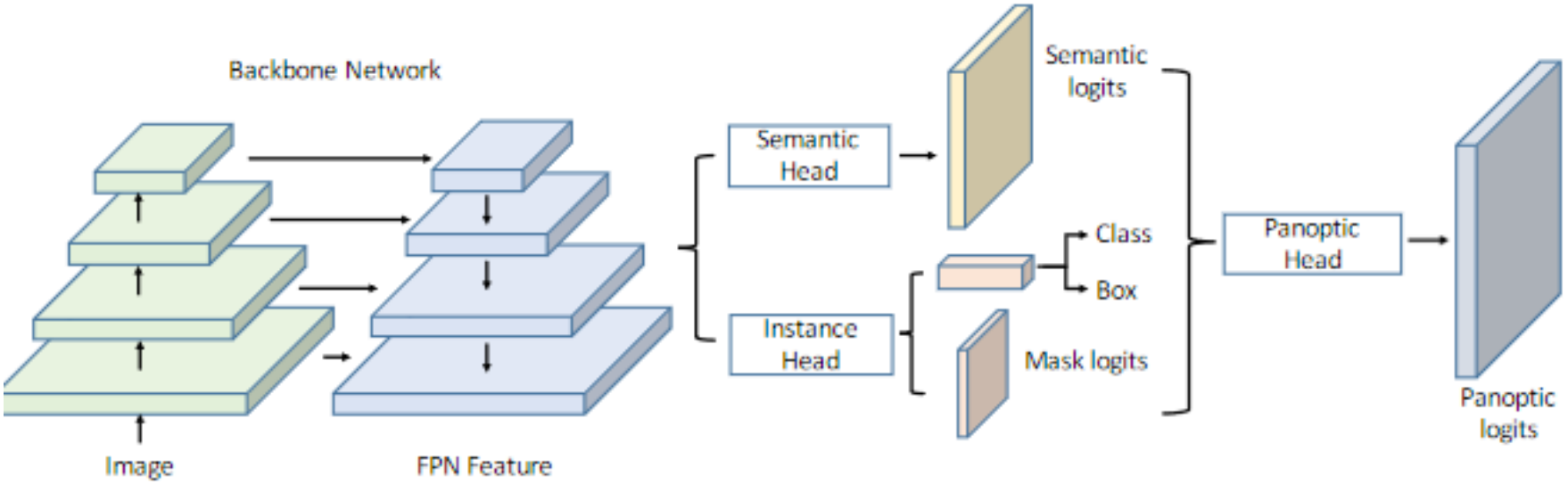


- Concurrently other FPN-based top-down methods:
  - Seamless Segmentation [Porzi, 2019]

[Porzi, 2019]

Figure from [Porzi, 2019]

Concurrent Approaches

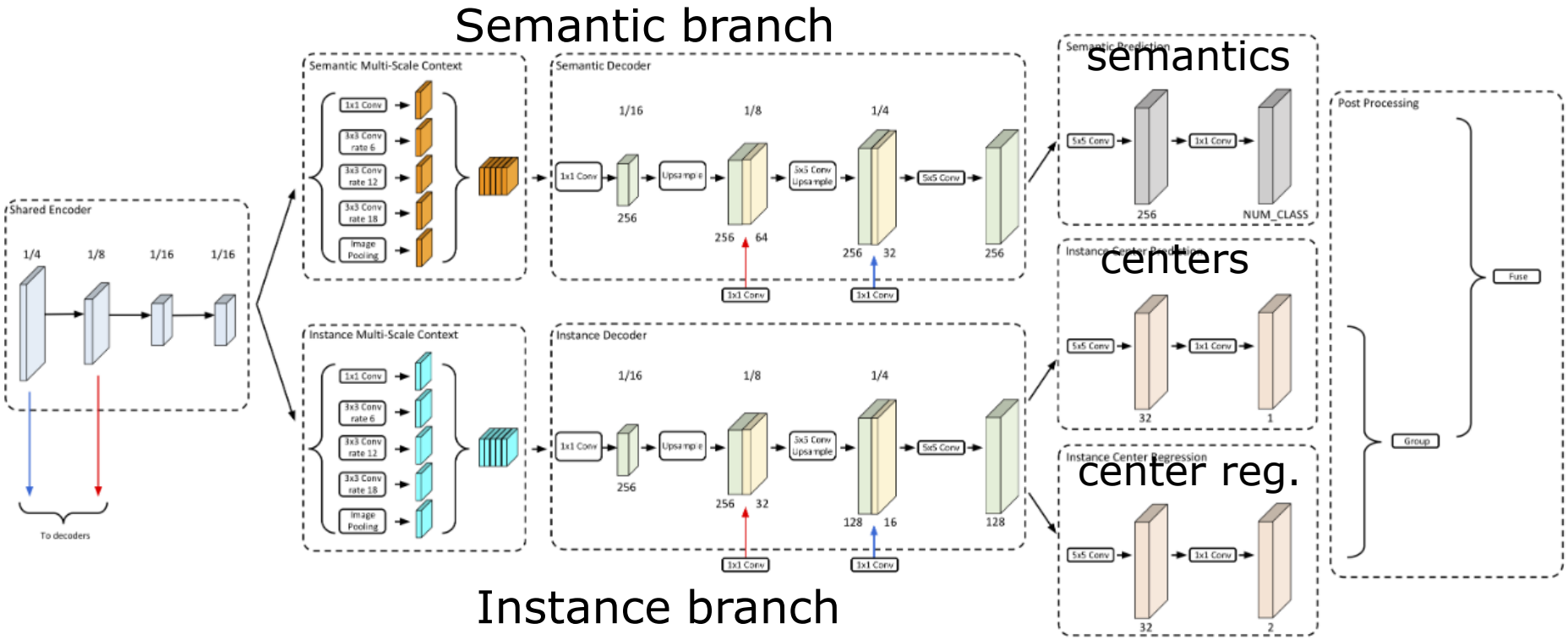


- Concurrently other FPN-based top-down methods:
  - Seamless Segmentation [Porzi, 2019]
  - Unified Panoptic Segmentation (UPSNet) [Xiong, 2019]

[Xiong, 2019]

Figure from [Xiong, 2019]

Panoptic-DeepLab



- **Bottom-up approach** using separate branches for semantic and instance segmentation
- Use semantic labels to filter instances & majority vote on instances to assign instance labels

[Cheng, 2020]

Figure from [Cheng, 2019]

Panoptic-DeepLab: Instances

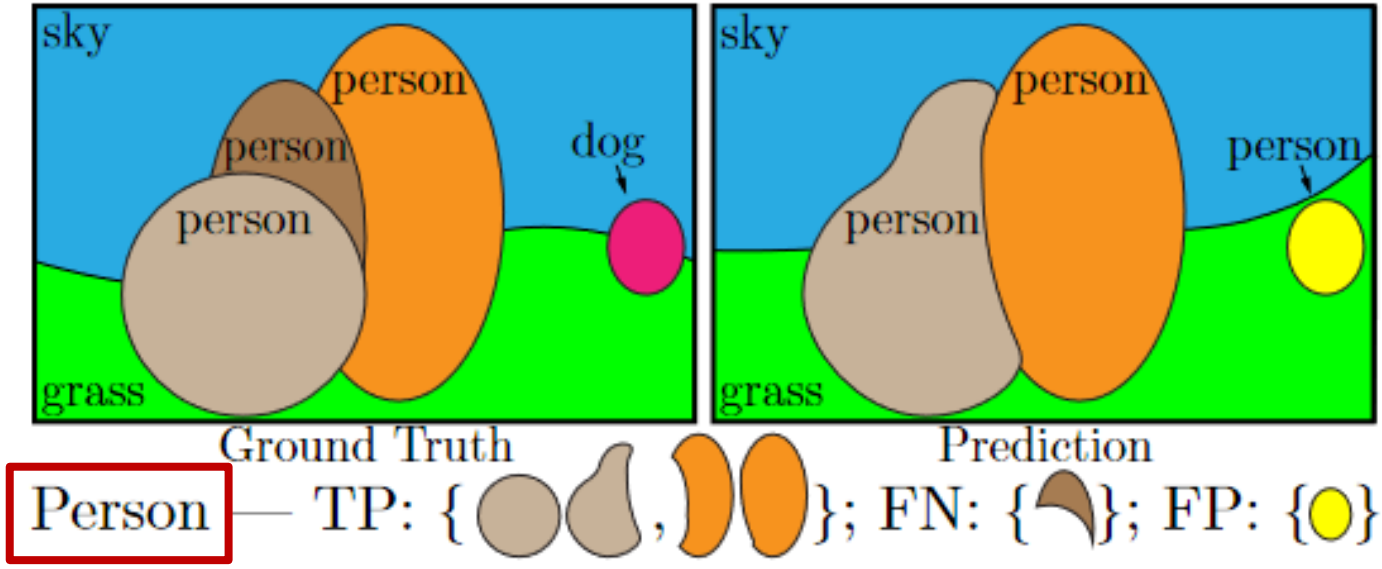
center

center offsets

- For instance segmentation:
  - Instance center prediction (= center of mass)
  - For each instance pixel: estimate offset vector that points towards instance center
- At inference time: Use offsets to assign instance id of closest instance center.

[Cheng, 2020]

Metric: Panoptic Quality

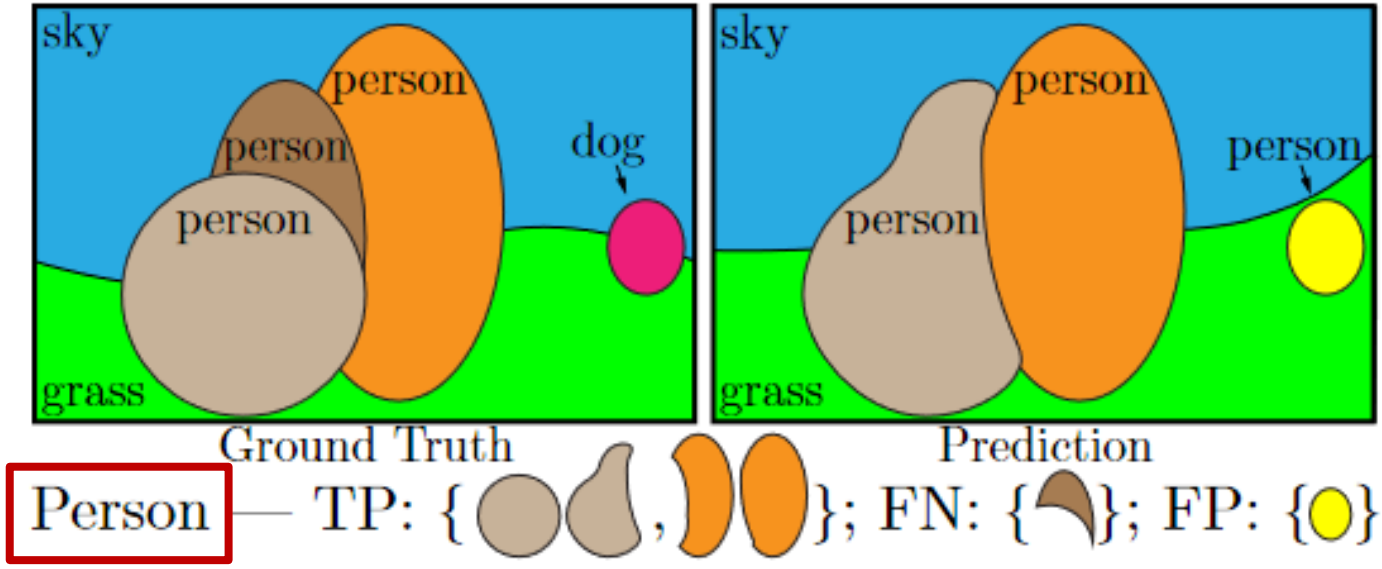


- Computed class-wise over segment assignments:
  - Matched segments: $\mathrm{TP}_c = \{g \in \mathcal{G}_c, p \in \mathcal{P}_c | \mathrm{IoU}(p, q) > 0.5\}$
  - Unmatched predictions FP_c: no gt segment with IoU > 0.5
  - Unmatched ground truth FN_c: no prediction with IoU > 0.5

[Kirillov, 2019]

Figure from [Kirillov, 2019]

Metric: Panoptic Quality



- Class-wise PQ is defined as:

$$\mathrm{PQ}_c = \frac{\sum_{(p,g)\in \mathrm{TP}_c} \mathrm{IoU}(p,g)}{|\mathrm{TP}_c| + \frac{1}{2}|\mathrm{FP}_c| + \frac{1}{2}|\mathrm{FN}_c|}$$

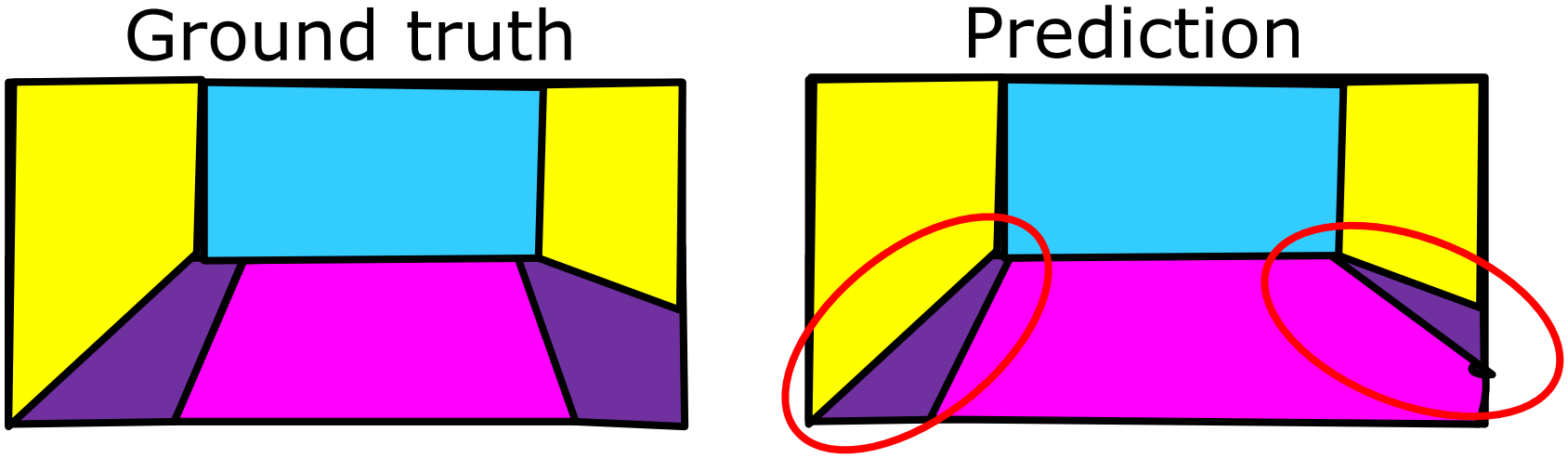
- Overall panoptic quality (PQ)

$$PQ = \frac{1}{K}\sum_c \mathrm{PQ}_c$$

[Kirillov, 2019]

Figure from [Kirillov, 2019]

Metric: Panoptic Quality*



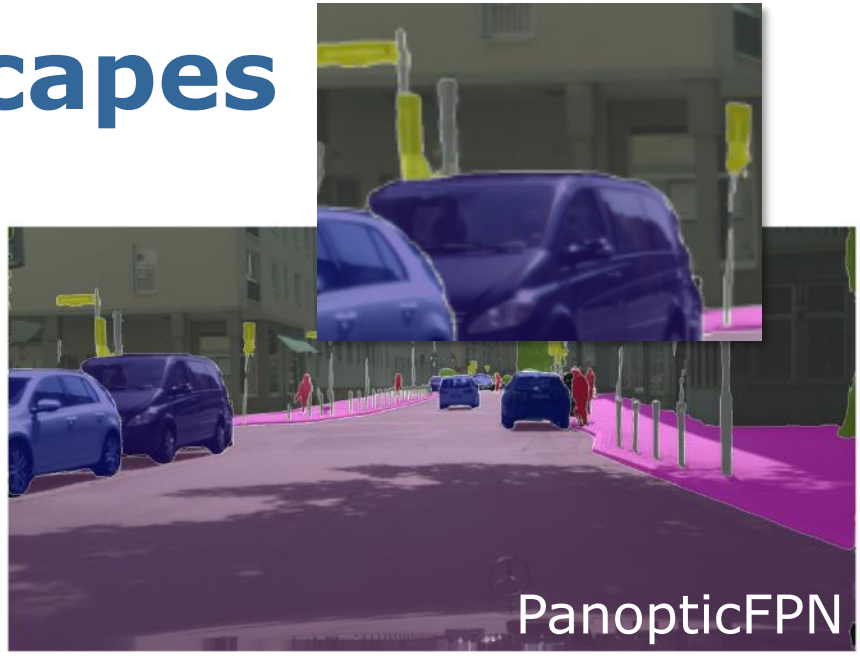
- For **stuff** regions, the IoU-based definition of TP will penalize large segments
- Thus, compute PQ only for thing classes, use IoU for stuff classes:

$$\mathcal{M}_c = \{(s, \hat{s}) \in \mathcal{S}_c \times \hat{\mathcal{S}}_c | \mathrm{IoU}(s, \hat{s}) > 0\}$$

$$\mathrm{PQ}_c^{\dagger} = \begin{cases} \frac{1}{|\mathcal{S}_c|} \sum_{(s,\hat{s}) \in \mathcal{M}_c} \mathrm{IoU}(s, \hat{s}), & \text{if } c \text{ is stuff class} \\ \mathrm{PQ}_c, & \text{otherwise.} \end{cases}$$

[Porzi, 2019]

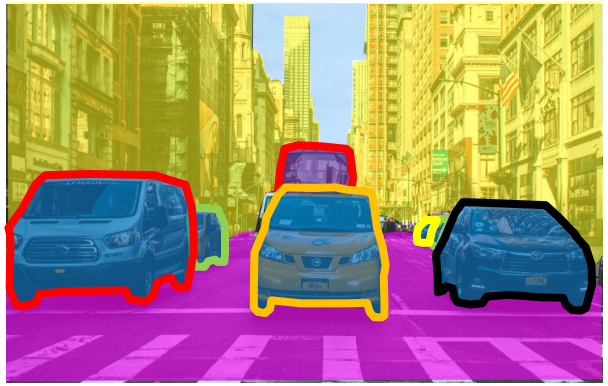
Results on Cityscapes



Approach	PQ	PQ^{Th}	PQ^{St}	mIoU
PanopticFPN	58.1	52.0	62.4	75.7
UPSNet	59.3	54.9	62.7	75.2
Seamless Segmentation	60.3	56.1	63.3	77.5
Panoptic-DeepLab	62.3	-	-	79.4

Summary





- Fine-grained scene understanding:
  - Semantic Segmentation
  - Instance Segmentation
  - Panoptic Segmentation
- Discussed common, popular approaches for segmentation in these domains.

See you next week!

References

- Badrinarayanan et al. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, PAMI, 2017.
- Cheng et al. Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-up Panoptic Segmentation, CVPR, 2020.
- Cordts et al. The Cityscapes Dataset for Semantic Urban Scene Understanding, CVPR, 2016.
- De Brabandere et al. Semantic Instance Segmentation with a Discriminative Loss Function, CVPRW, 2017.
- Chen et al. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, CVPR, 2018.
- Gupta et al. LVIS: A Dataset for Large Vocabulary Instance Segmentation, CVPR, 2019.
- He et al. Mask R-CNN, ICCV, 2017.
- Kirillov et al. Panoptic Feature Pyramid Networks, CVPR, 2019.
- Kirillov et al. Panoptic Segmentation, CVPR, 2019.
- Lin et al. Microsoft COCO: Common Objects in Context, ECCV, 2014.
- Long et al. Fully Convolutional Networks for Semantic Segmentation, CVPR, 2015.
- Neuhold et al. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes, ICCV, 2017.
- Porzi et al. Seamless Scene Segmentation, CVPR, 2019.
- Ronneberger et al. U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI, 2015.
- Shelhamer et al. Fully Convolutional Models for Semantic Segmentation, PAMI, 2016.
- Xiong et al. UPSNet: A Unified Panoptic Segmentation Network, CVPR, 2019.
- Zhao et al. Pyramid Scene Parsing Network, CVPR, 2017.
- Zhou et al. Scene Parsing through ADE20K Dataset, CVPR, 2017.