Master IMA – RDFIA 2026-2027
Reconnaissance des formes pour l'analyse et l'interprétation d'images
Semantic Segmentation

Clément Rambour

clement.rambour@isir.upmc.fr

Sorbonne Université
ISIR - équipe MLIA

Année universitaire 2026-2027

Outline

1. Segmentation: definitions and data

2. Supervised segmentation with deep ConvNets
   - Fully Convolutional Networks
   - Deconvolution networks and U-Net
   - Atrous convolution: the DeepLab approach

3. Going further: segmentation after DeepLab

Segmentation: definitions



- What is in the image, and **where**?

Def1: Semantic segmentation

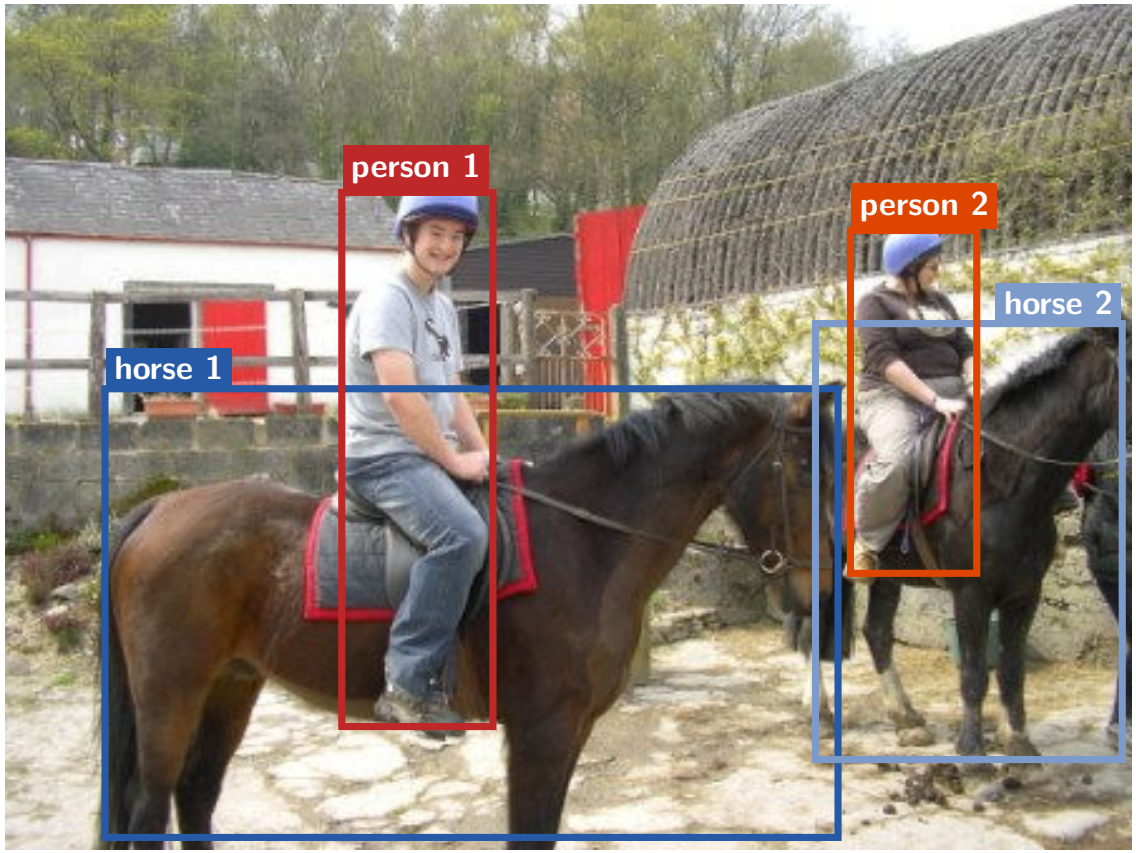
- Label each **pixel** with a category label



- Both horses: **same** label, no instances

Object detection

- One **box** per **instance**



Def2: Instance segmentation

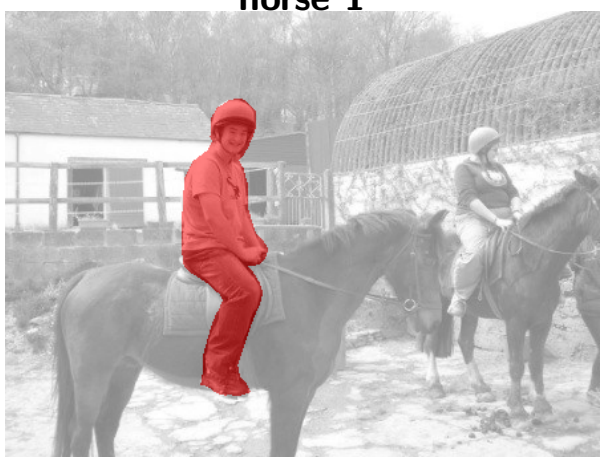
- One **mask** per **instance** (detection + segmentation)



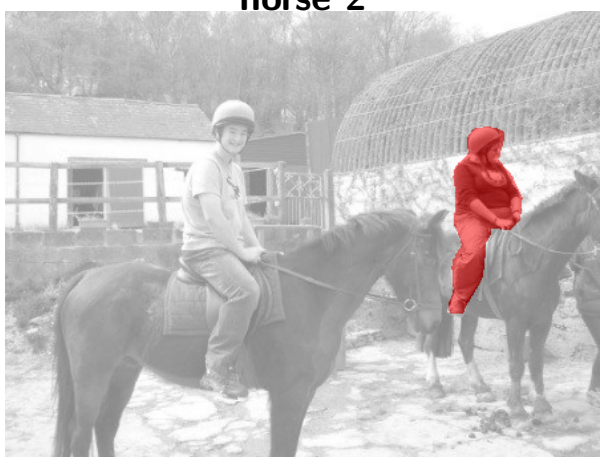
horse 1



horse 2



person 1



person 2

Three tasks, three outputs

Image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$, C classes

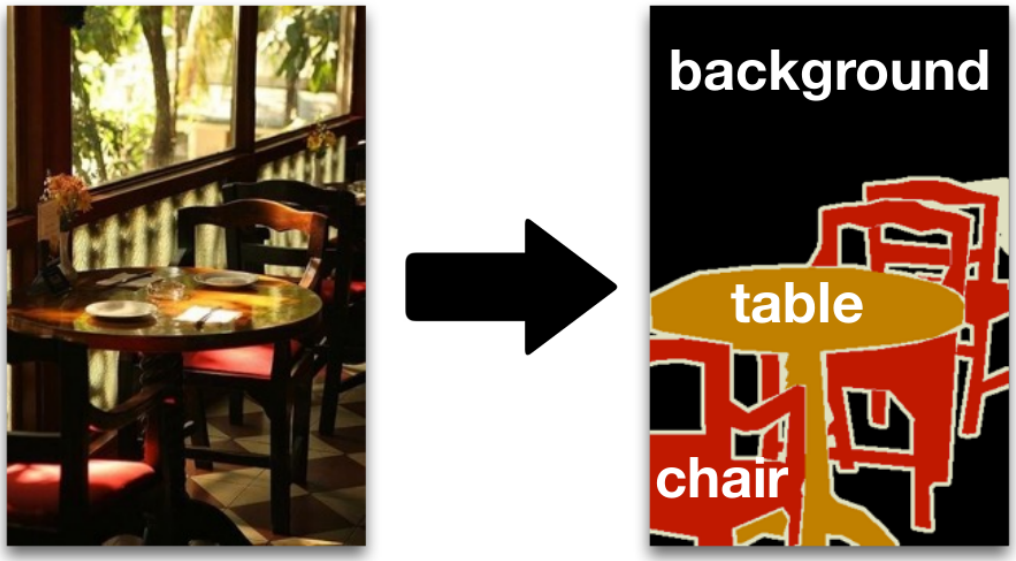
Task	Output	Metric
Semantic segmentation	a label map $\mathbf{y} \in \{1, \dots, C\}^{H \times W}$	mean IoU over classes
Object detection	a set $\{(b_k, c_k, s_k)\}_k$: box, class, score	AP (box IoU $\geq$ threshold)
Instance segmentation	a set $\{(m_k, c_k, s_k)\}_k$, binary masks $m_k \in \{0, 1\}^{H \times W}$	AP (mask IoU $\geq$ threshold)

- Semantic segmentation = **dense classification**
- **This lecture:** semantic segmentation

Multi-class image segmentation

Goal

Assign a class label to each pixel in the image





Datasets: PASCAL VOC 2012

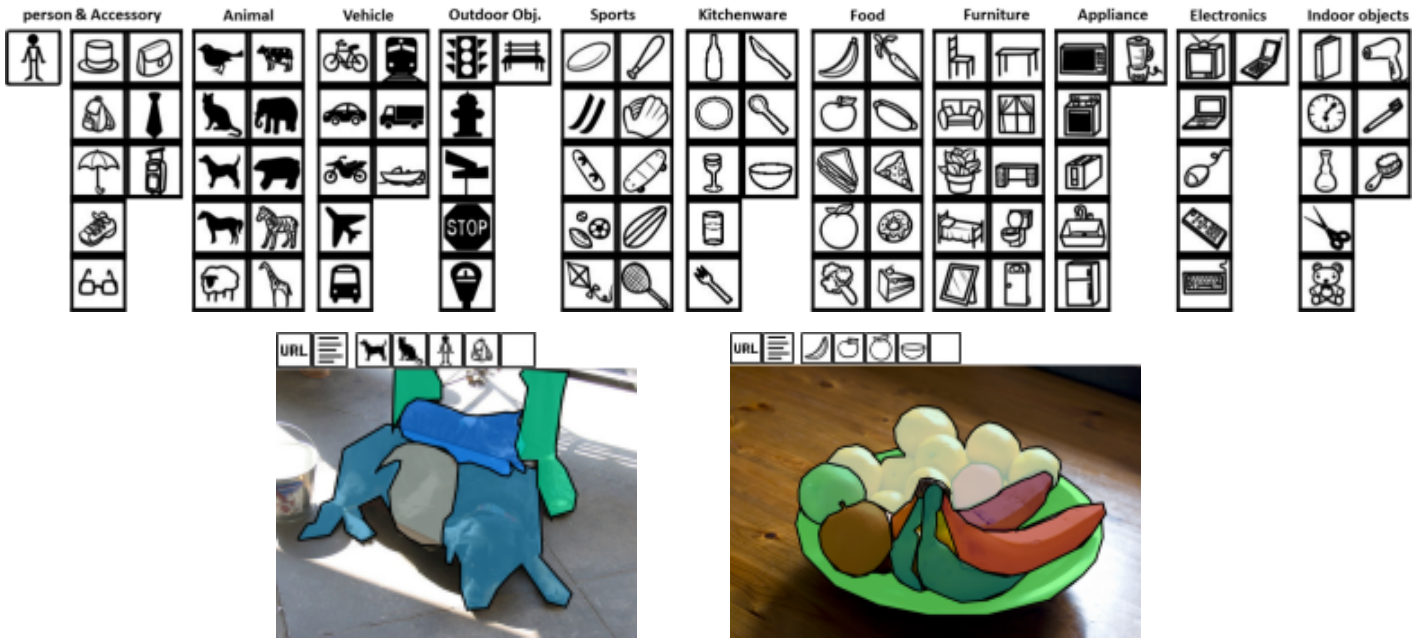
- Train 1464 images / Val 1449 images / Test 1456 images
- 20 object classes + background
- Metric: mean IoU [Everingham et al., 2010]





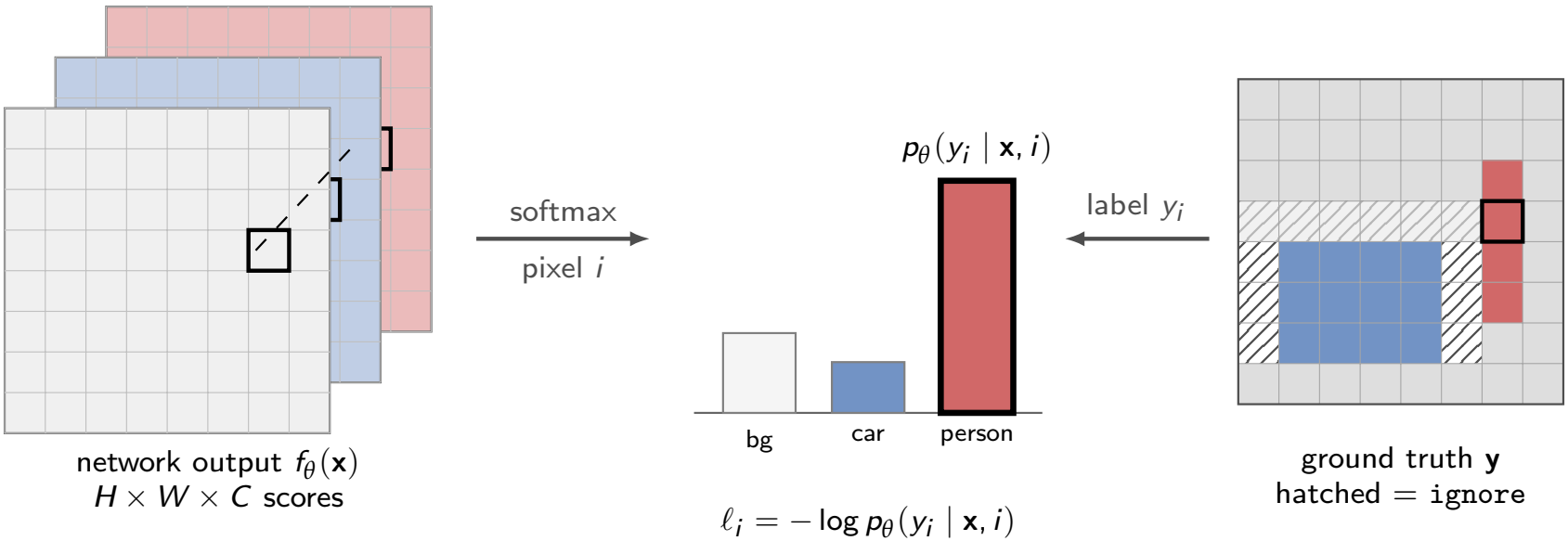

Datasets: MS COCO

- 118k train / 5k val images, 80 object categories [Lin et al., 2014]
- Detection, instance and panoptic segmentation



Many small, **non-centred** objects per image

Training: per-pixel cross-entropy

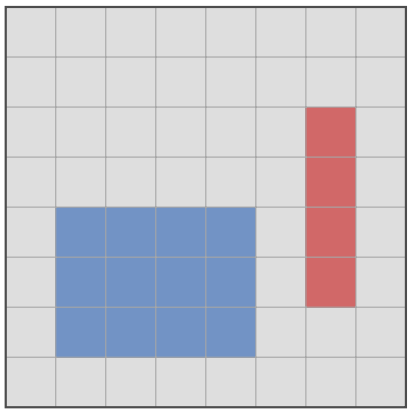


$$\mathcal{L}(\theta) = \frac{1}{|\Omega|} \sum_{i \in \Omega} w_{y_i} \ell_i = -\frac{1}{|\Omega|} \sum_{i \in \Omega} w_{y_i} \log p_\theta(y_i \mid \mathbf{x}, i)$$

- Ω: annotated pixels (`ignore` label excluded)
- w_c: optional class weights against imbalance

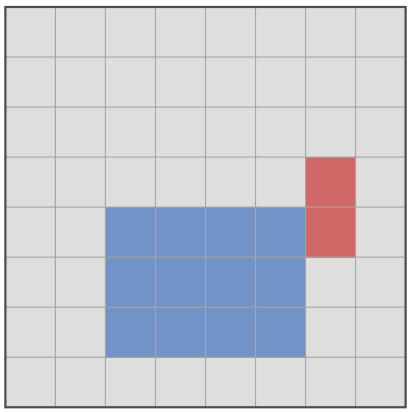
Evaluation: a toy example

ground truth



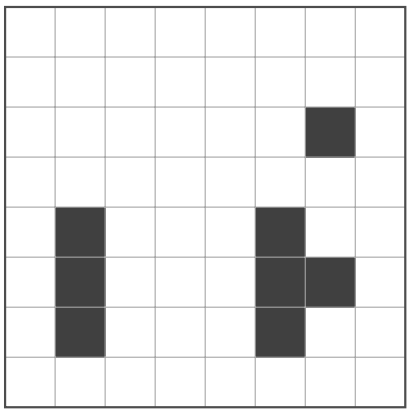
bg 48 px, car 12 px,
person 4 px

prediction



car shifted by one column,
person half missed

errors



8 wrong pixels
out of 64

background car person

Pixel accuracy

$$\text{pixel acc.} = \frac{\#\,\text{correct pixels}}{\#\,\text{pixels}} = \frac{56}{64} = 87.5\,\%$$

- "All background" already gets 75 %

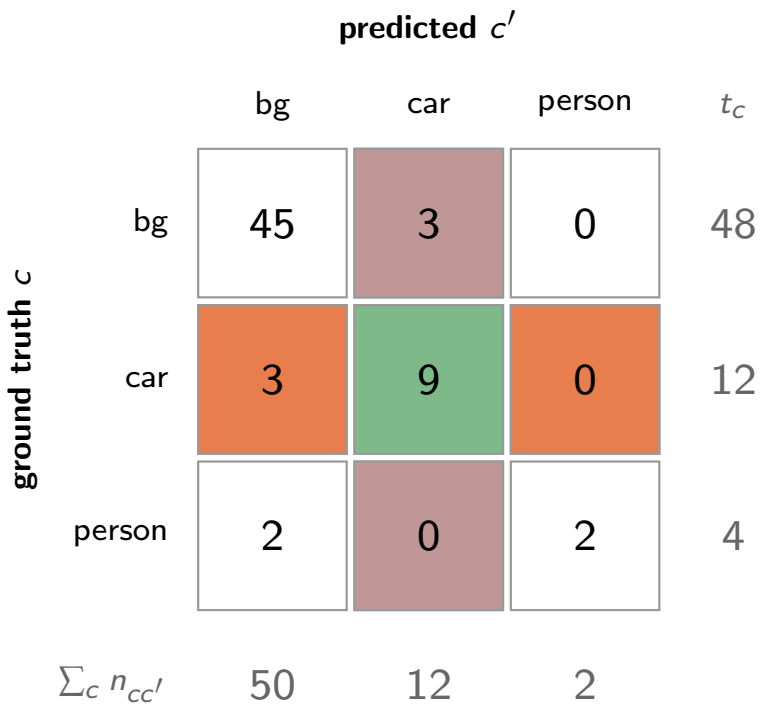
The confusion matrix

ground truth c \ predicted c'	bg	car	person	t_c
bg	45	3	0	48
car	3	9	0	12
person	2	0	2	4
$\sum_c n_{cc'}$	50	12	2	

- $n_{cc'}$: pixels of class c predicted c'
- Row sum t_c: size of class c

- Accumulated over the **whole dataset**

The confusion matrix



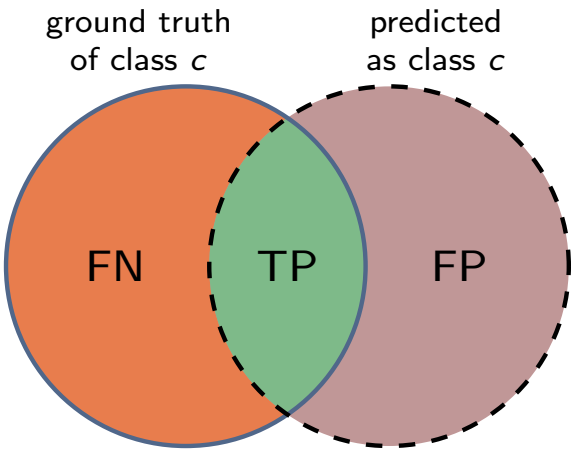
- $n_{cc'}$: pixels of class c predicted c'
- Row sum t_c: size of class c

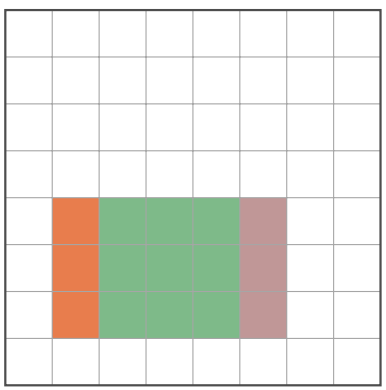
For class **car**:

- TP $= n_{cc} = 9$
- FN $=$ rest of the row $= 3$
- FP $=$ rest of the column $= 3$

- Accumulated over the **whole dataset**

Intersection over Union





class car:
TP 9, FN 3, FP 3

$$\mathrm{IoU}_c = \frac{|\,\mathrm{GT}_c \cap \mathrm{Pred}_c\,|}{|\,\mathrm{GT}_c \cup \mathrm{Pred}_c\,|} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}$$

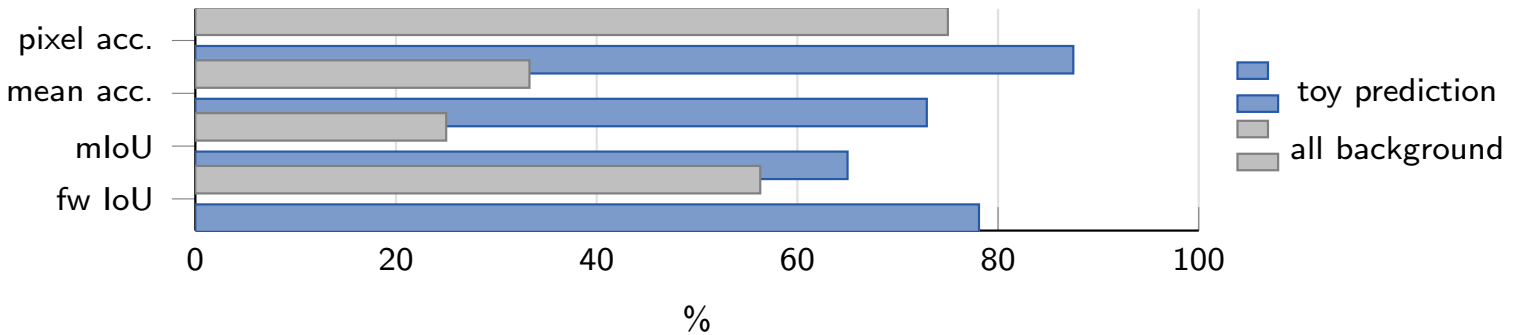
$$= \frac{n_{cc}}{t_c + \sum_{c'} n_{c'c} - n_{cc}}$$

	TP	FP + FN	IoU
background	45	$5+3$	$45/53 = 0.85$
car	9	$3+3$	$9/15 = 0.60$
person	2	$0+2$	$2/4 = 0.50$

- Penalises misses **and** false alarms

Which metric?

	formula	**toy prediction**	**all background**
pixel accuracy	$\frac{\sum_c n_{cc}}{\sum_c t_c}$	87.5	75.0
mean accuracy	$\frac{1}{C}\sum_c \frac{n_{cc}}{t_c}$	72.9	33.3
mean IoU	$\frac{1}{C}\sum_c \mathrm{IoU}_c$	**65.0**	**25.0**
frequency-weighted IoU	$\frac{1}{\sum_c t_c}\sum_c t_c \mathrm{IoU}_c$	78.1	56.3



- **mIoU**: every class counts equally $\Rightarrow$ the standard

Supervised image segmentation methods

Full supervision

- Precise annotation ✓
- Expensive ✗: $>$ 1.5 h per Cityscapes image [Cordts et al., 2016]





Weakly supervised image segmentation methods

Weak supervision

- Image tags only ✗
- Cheap: $\approx$ 1 s per label
  ✓[Bearman et al., 2016]



✓ background
✗ aeroplane
✗ cat
✓ chair
✗ dog
✗ person
✗ sheep
✓ table
✗ tv/monitor

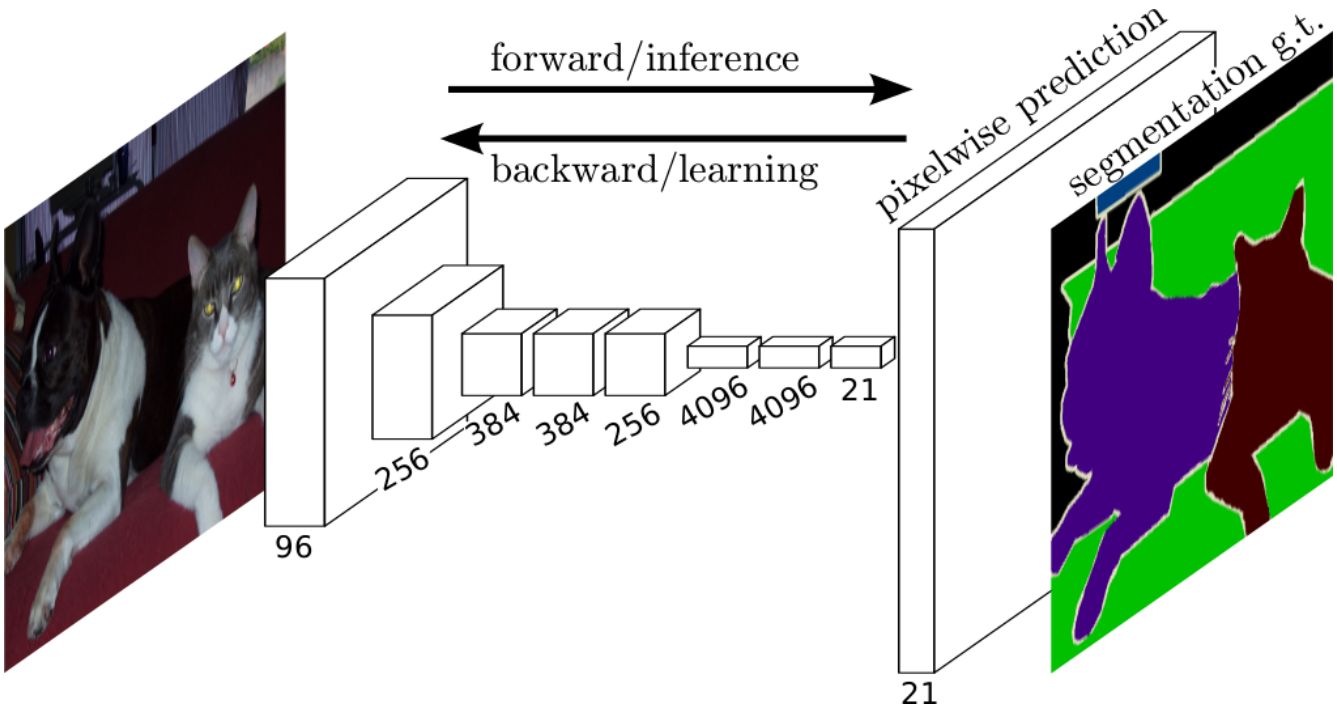
- *What*, not *where* — **this lecture:** full supervision

Outline

1. Segmentation: definitions and data
2. Supervised segmentation with deep ConvNets
   - Fully Convolutional Networks
   - Deconvolution networks and U-Net
   - Atrous convolution: the DeepLab approach
3. Going further: segmentation after DeepLab

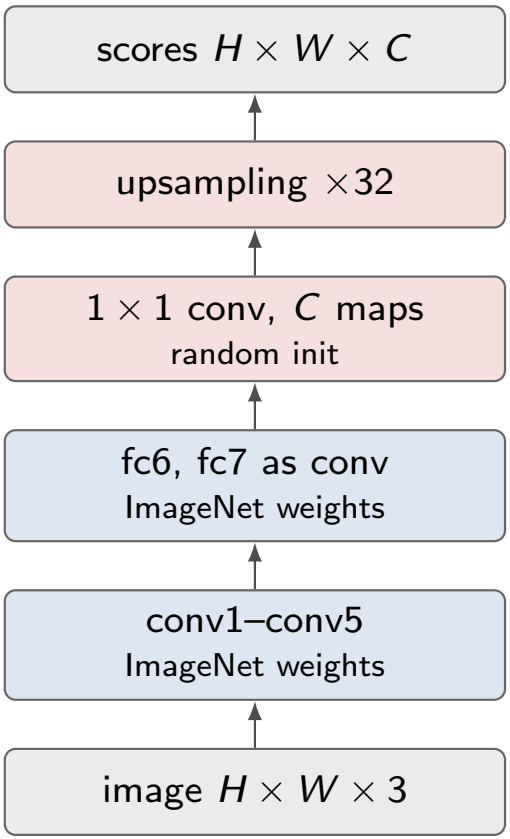
FCN: Fully Convolutional Network [Long et al., 2015]

- Classify each pixel, fully convolutionally
- Upsample the output (bilinear or "deconvolution")
- Loss: per-pixel cross-entropy



FCN = transfer learning from ImageNet

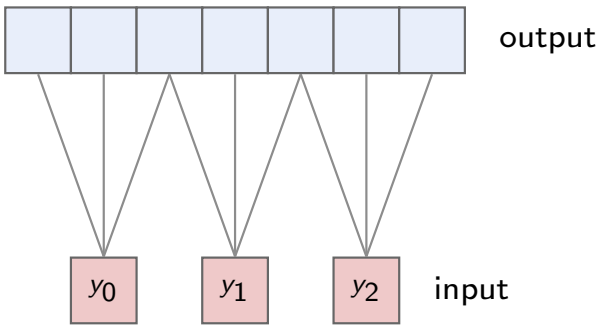
1. ImageNet classifier, last layer removed
2. fc $\rightarrow$ conv (same weights)
3. 1×1 conv, C outputs
4. **Upsample** to input size
5. Fine-tune everything



Fine-tuning as in lecture 4, with a **dense** target task

Learnable upsampling: transposed convolution

- Strided conv: $\mathbf{y} = A\,\mathbf{x}, \quad \mathbb{R}^n \to \mathbb{R}^{n/s}$
- **Transposed conv:** $\mathbf{x}' = A^\top \mathbf{y}'$, $\mathbb{R}^{n/s} \to \mathbb{R}^n$
- Learnable upsampling



kernel 3, stride 2: overlaps are summed

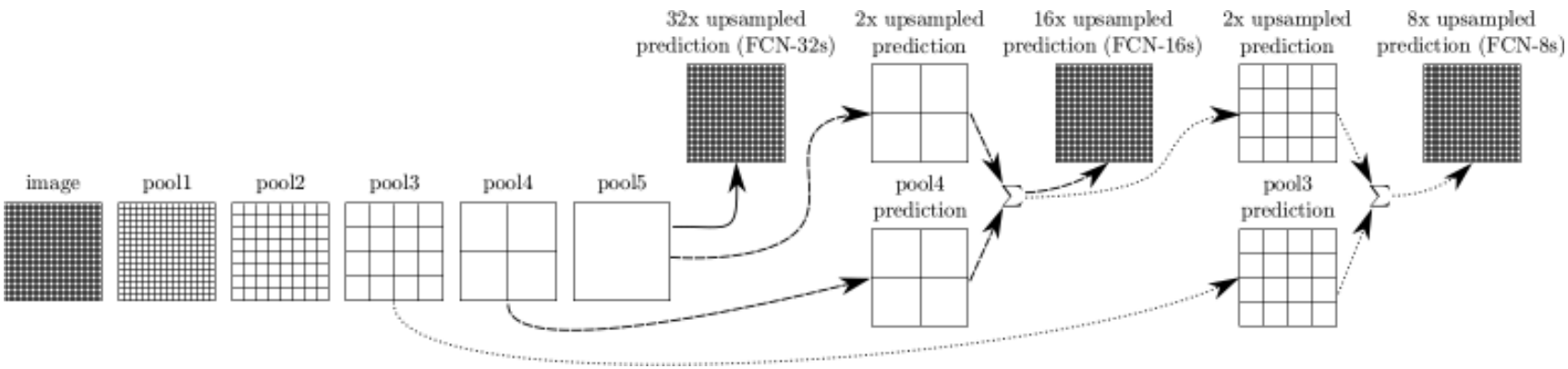
Output size (per dimension)

$$n_{\text{out}} = s\,(n_{\text{in}} - 1) + k - 2p$$

- "Deconvolution" is a misnomer: **not** an inverse
- FCN: initialised to **bilinear** interpolation
- Uneven overlap ⇒ checkerboard artefacts [Odena et al., 2016]

FCN: Fully Convolutional Network

- Pooling and striding lose resolution
- **Skip connections**: fuse scores from shallower layers



FCN-32s, 16s, 8s: fusing scores across scales

- S_ℓ: class scores on pool_ℓ (resolution $\frac{1}{2^\ell}$), S_7 on fc7
- up_r: transposed conv, factor r

$$\begin{aligned} \text{FCN-32s:} \quad & \text{up}_{32}(S_7) \\ \text{FCN-16s:} \quad & \text{up}_{16}(\text{up}_2(S_7) + S_4) \\ \text{FCN-8s:} \quad & \text{up}_8(\text{up}_2(\text{up}_2(S_7) + S_4) + S_3) \end{aligned}$$

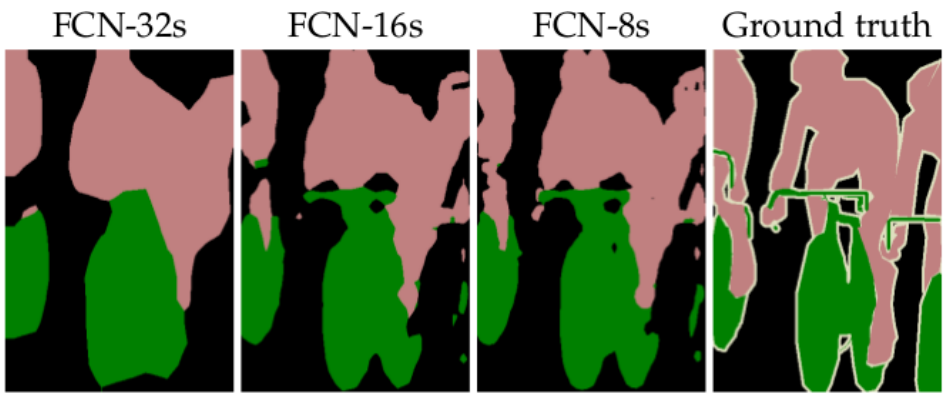
- Deep layers: **what** Shallow layers: **where**

FCN: Fully Convolutional Network

	pixel acc.	mean acc.	mean IoU	f.w. IoU
FCN-32s-fixed	83.0	59.7	45.4	72.0
FCN-32s	89.1	73.3	59.4	81.4
FCN-16s	90.0	75.7	62.4	83.0
FCN-8s	**90.3**	**75.9**	**62.7**	**83.2**

PASCAL VOC 2011 val subset [Long et al., 2015]

FCN-32s-fixed: only the last layer is fine-tuned



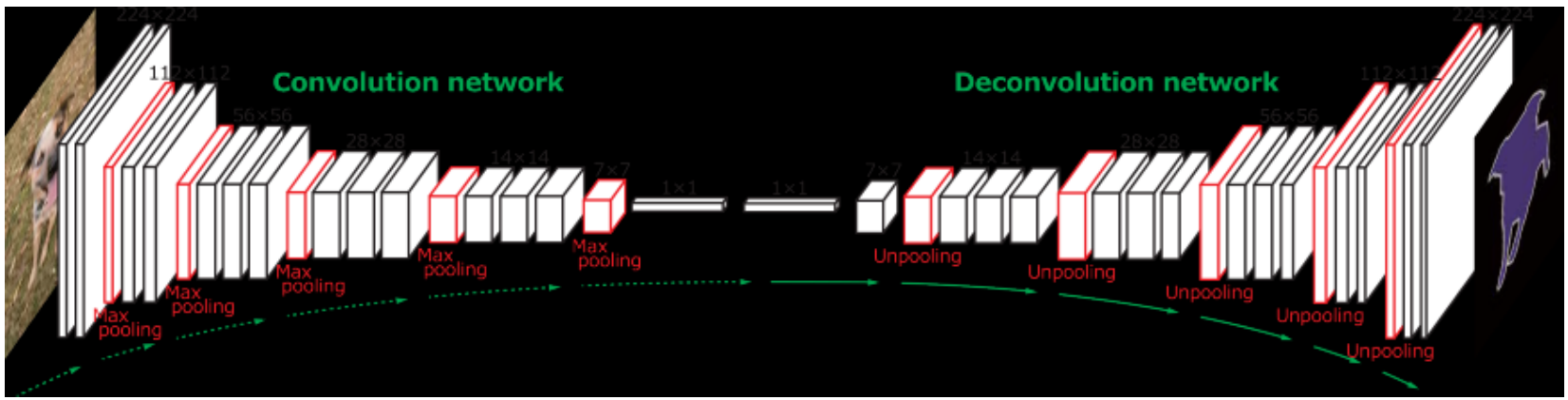
- Fine-tuning vs frozen: +14 mIoU Skips: +3 mIoU

Outline

1. Segmentation: definitions and data
2. Supervised segmentation with deep ConvNets
   - Fully Convolutional Networks
   - Deconvolution networks and U-Net
   - Atrous convolution: the DeepLab approach
3. Going further: segmentation after DeepLab

Deconvolution network [Noh et al., 2015]

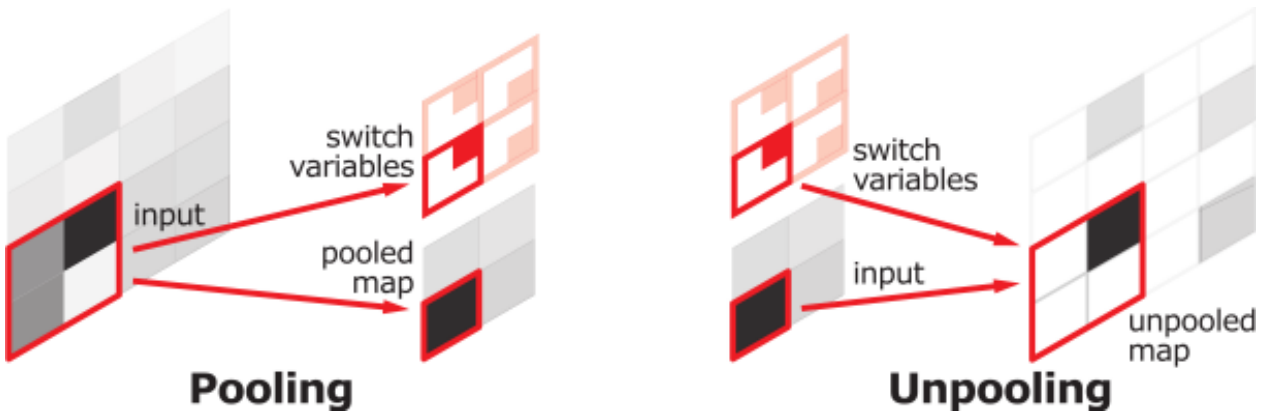
- Encoder: convolution network (feature extractor)
- Decoder: its **mirror**, a shape generator



Deconvolution network

Unpooling

- Reverse of pooling: back to the original size
- Output: **sparse** map



Unpooling with switch variables

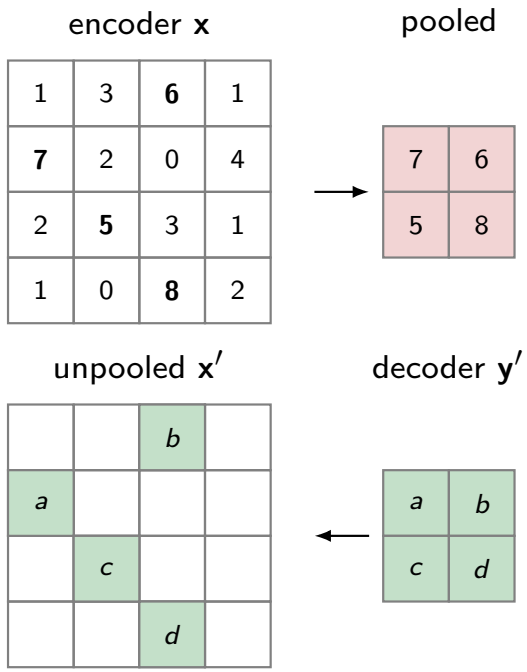
Max pooling over a window $\mathcal{R}_u$:

$$y_u = \max_{v \in \mathcal{R}_u} x_v, \qquad \sigma(u) = \arg\max_{v \in \mathcal{R}_u} x_v$$

the **switch** $\sigma(u)$ is stored

Unpooling of a decoder map $\mathbf{y}'$:

$$x'_v = \begin{cases} y'_u & \text{if } v = \sigma(u) \\ 0 & \text{otherwise} \end{cases}$$

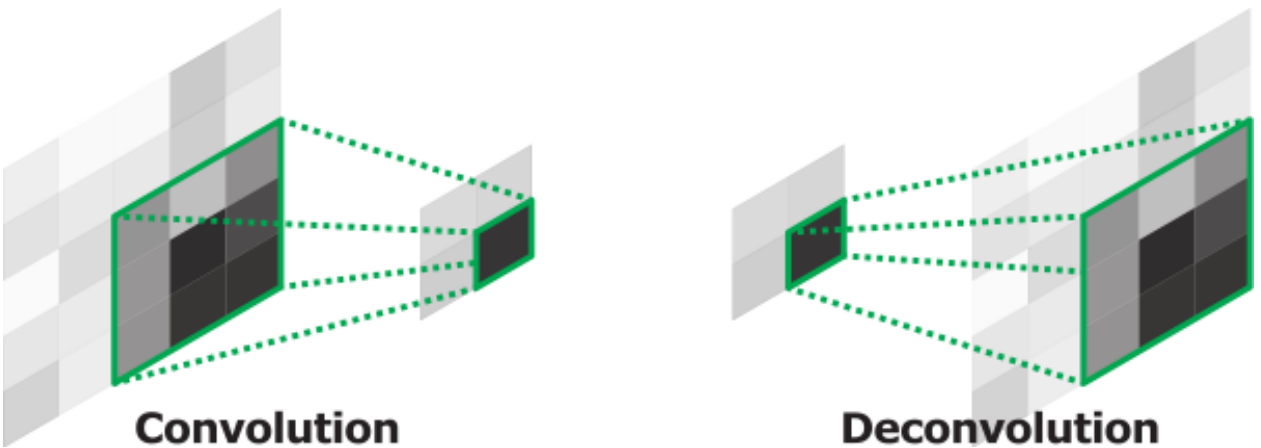


- Each activation goes back **where it came from**
- Only indices are stored (SegNet [Badrinarayanan et al., 2017])

Deconvolution network

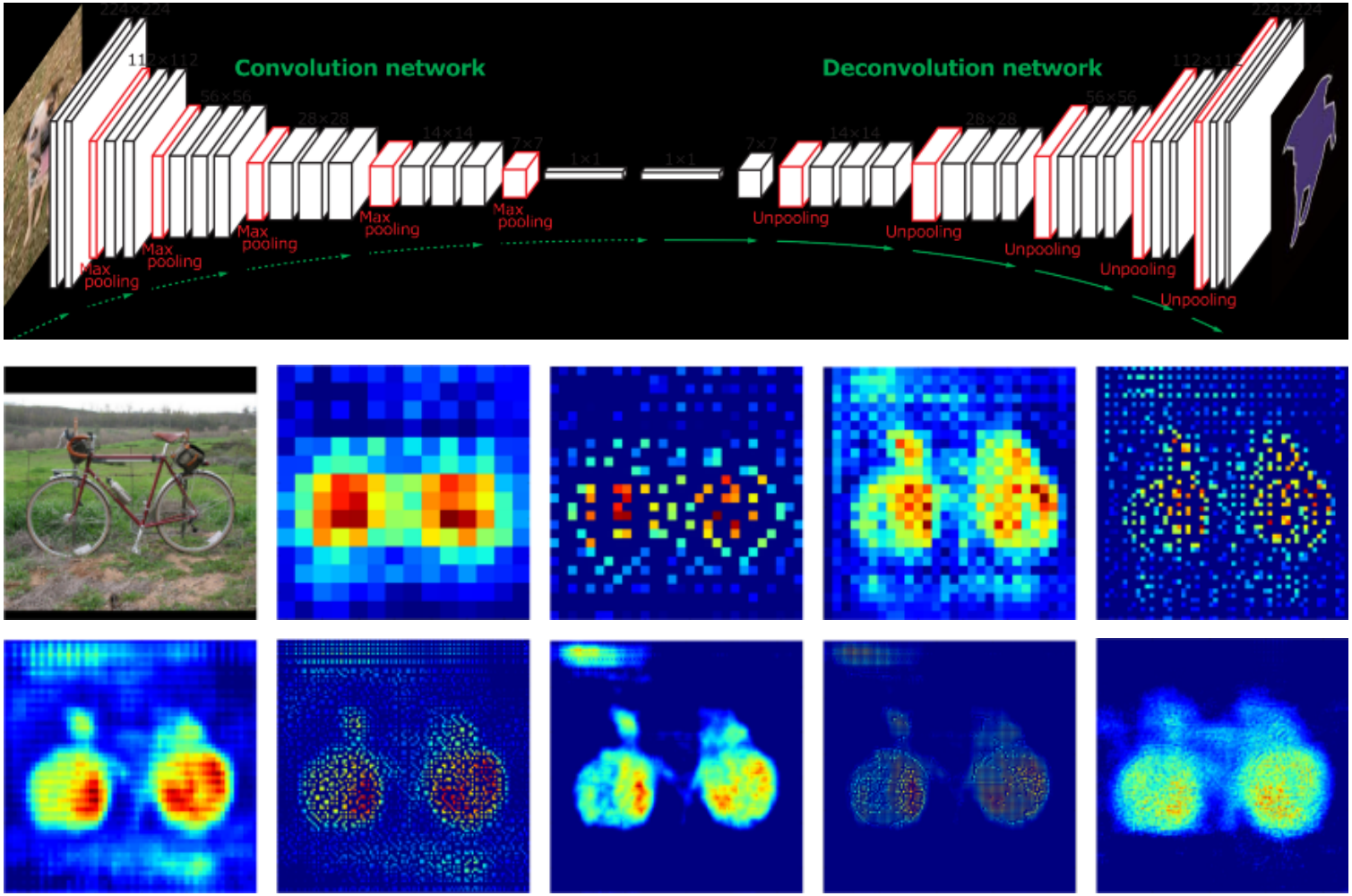
Deconvolution

- One input $\rightarrow$ many outputs (transposed conv)
- Output: **dense** map



Deconvolution network

- Hourglass structure + unpooling with switch variables



Decoder activations, coarse to fine

Deconvolution network + shortcut connection

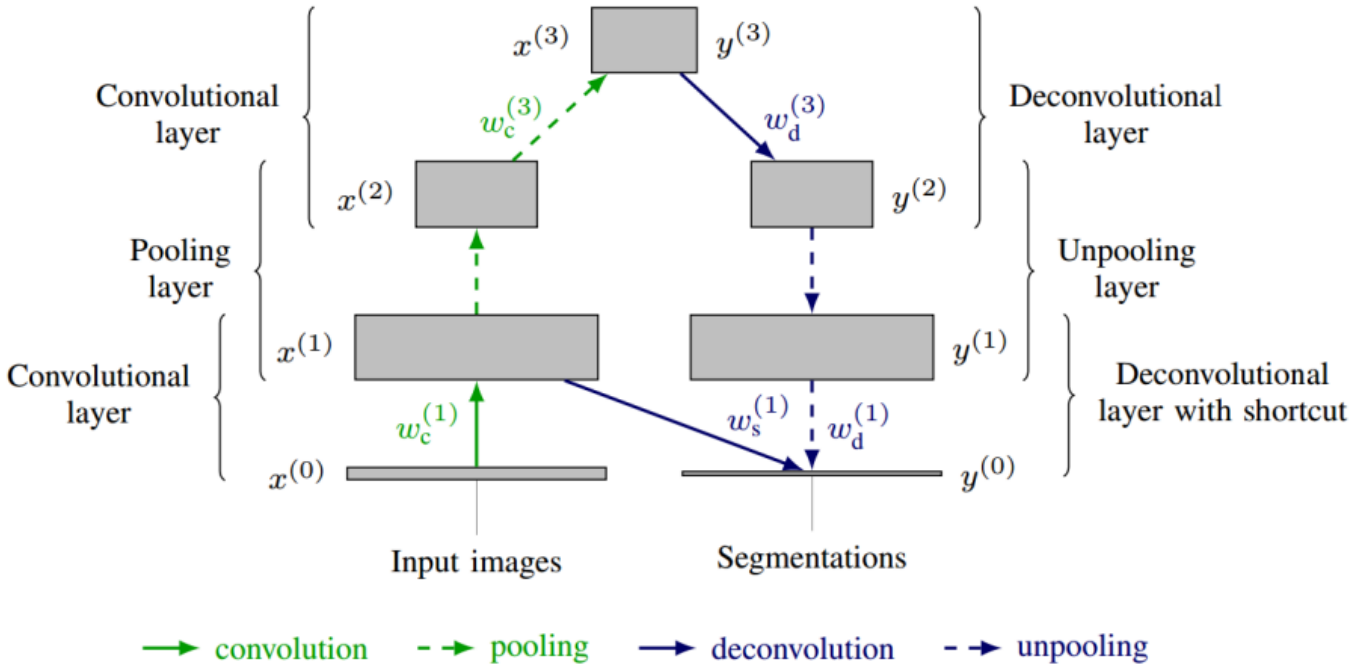


Fig. 3: CEN-s: The proposed architecture by Brosch et al. [11].

- CEN-s [Brosch et al., 2016]: shortcut from fine encoder layers to the decoder

U-Net [Ronneberger et al., 2015]

- Encoder–decoder + skips by **concatenation**
- Works with little data (medical imaging)

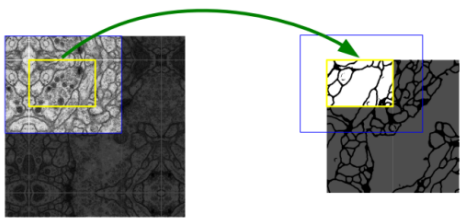


Fig. 2. Overlap-tile strategy for seamless segmentation of arbitrary large images (here segmentation of neuronal structures in EM stacks). Prediction of the segmentation in the yellow area, requires image data within the blue area as input. Missing input data is extrapolated by mirroring

Overlap-tile: borders are mirrored

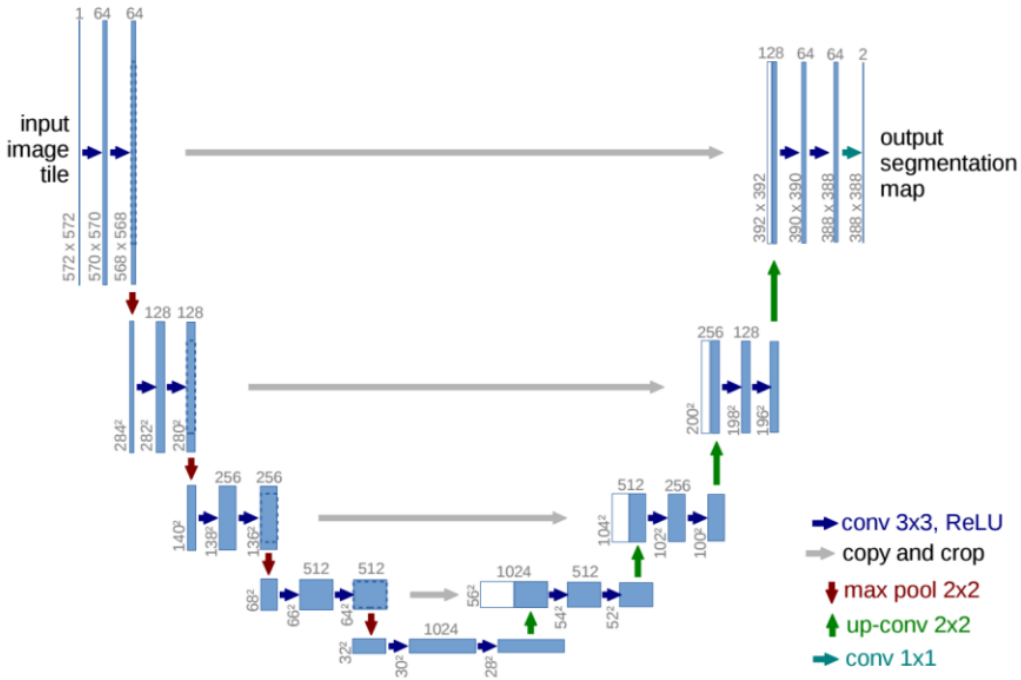


Fig. 2: U-Net: The proposed architecture by Ronneberger et al. [68].

U-Net in detail

- **Encoder:** 2 conv 3×3 + max pool; channels $\times 2$ per level
- **Decoder:** up-conv 2×2, **concatenate** encoder map, 2 conv 3×3

Weighted cross-entropy to separate touching cells

$$\mathcal{L} = -\sum_i w(i)\, \log p(y_i \mid \mathbf{x}, i), \qquad w(i) = w_c(i) + w_0 \exp\Big(-\frac{\big(d_1(i) + d_2(i)\big)^2}{2\sigma^2} \Big)$$

d_1, d_2: distances to the two nearest cells

- 30 training images: heavy **elastic** augmentation
- Today: default dense architecture (medical, diffusion models)

Recovering resolution: summary

Model	Upsampling	Encoder → decoder	Decoder
FCN	transposed conv (bilinear init)	sum of *scores*	none (score fusion)
DeconvNet	unpooling + transposed conv	pooling *indices*	mirror of the encoder
SegNet	unpooling	pooling *indices*	mirror of the encoder
U-Net	up-conv 2×2	concatenation of *features*	symmetric, as deep as the encoder

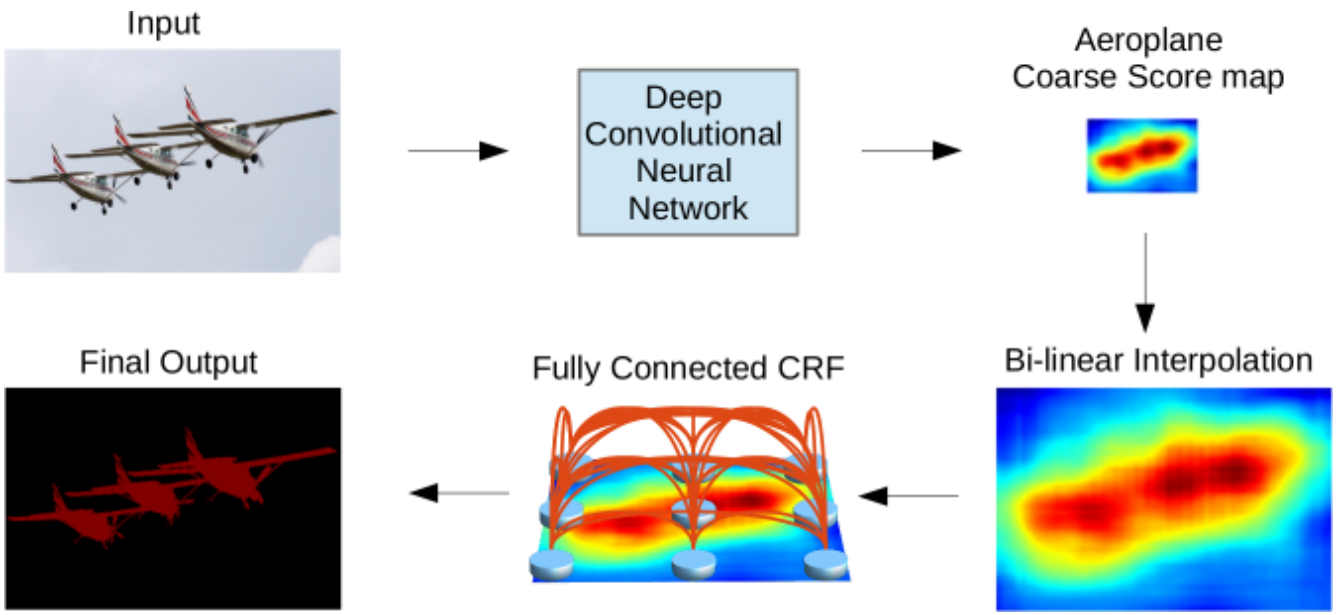
- All **downsample, then upsample**
- Alternative: **do not downsample** $\Rightarrow$ DeepLab

Outline

1. Segmentation: definitions and data
2. Supervised segmentation with deep ConvNets
   - Fully Convolutional Networks
   - Deconvolution networks and U-Net
   - Atrous convolution: the DeepLab approach
3. Going further: segmentation after DeepLab

DeepLab [Chen et al., 2015]

1. Keep resolution: **atrous** convolution
2. Sharpen boundaries: fully connected **CRF** [Krähenbühl and Koltun, 2011]



DeepLab: atrous filtering

- **Remove** the last downsamplings
- **Dilate** the filters instead (rate r):

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

r: sampling stride of the input
$r = 1$: standard convolution

$r = 1$ 

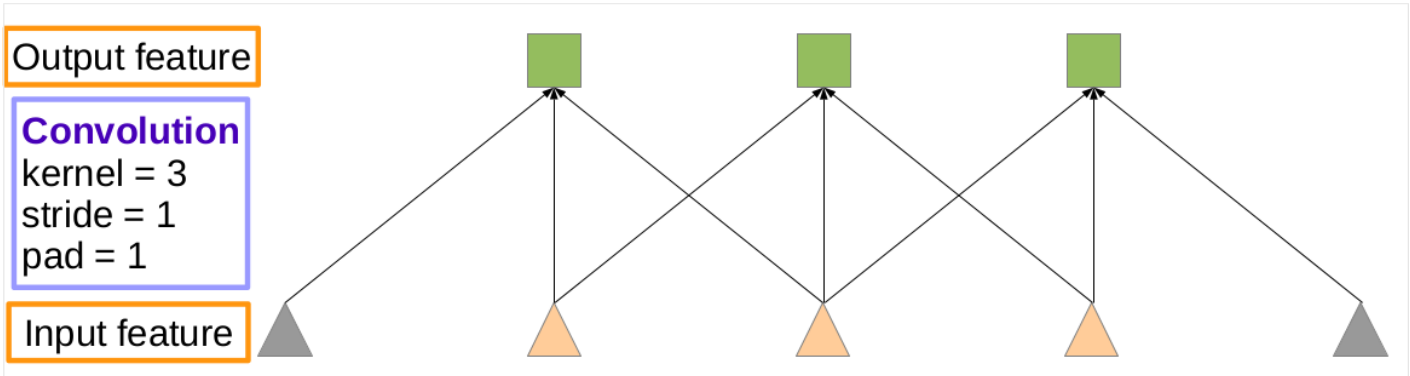
$r = 2$ 

introduce zeros between filter values

- "*à trous*" = with holes; a.k.a. **dilated** convolution [Yu and Koltun, 2016]

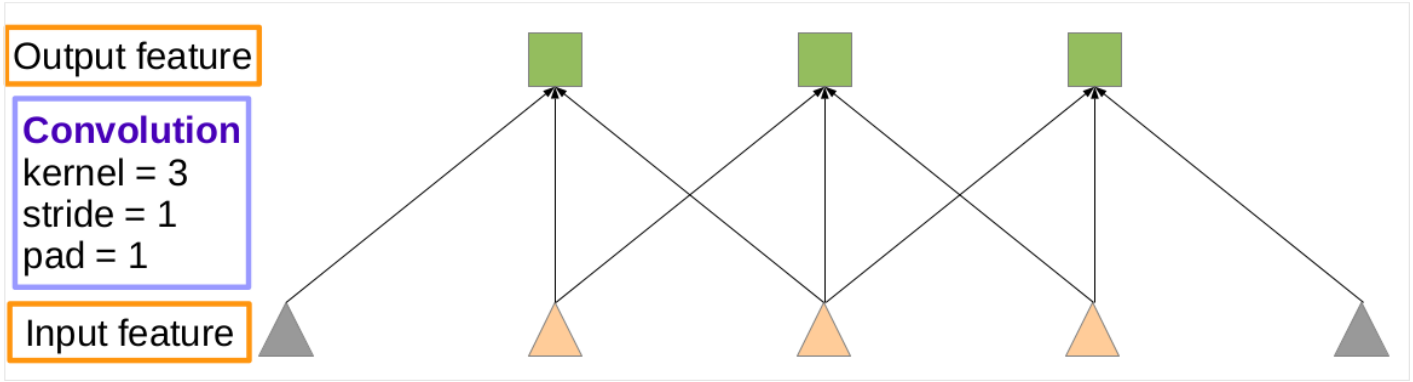
DeepLab: atrous (dilated) convolution

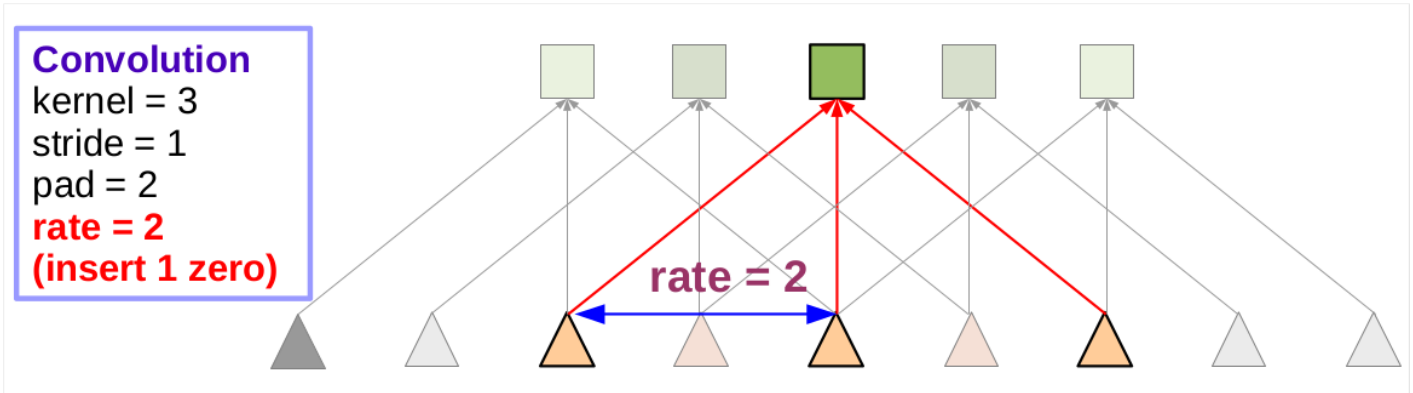
Classical filtering / pooling / downsampling



DeepLab: atrous (dilated) convolution

No downsampling + upsampled filter





Same weights, same field of view, output everywhere

DeepLab: atrous (dilated) convolution

- Small field of view $\rightarrow$ localisation; large $\rightarrow$ context
- Effective filter size:

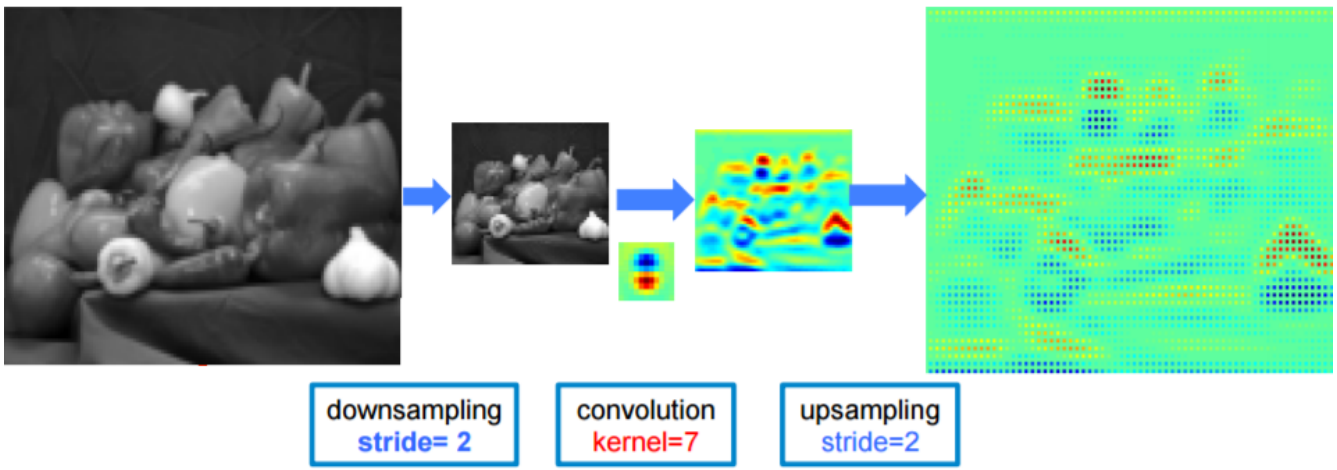
$$k_e = k + (k-1)(r-1)$$

- Same number of parameters and operations
- DeepLab-LargeFOV: 3×3, $r = 12 \Rightarrow k_e = 25$

Standard vs atrous convolution

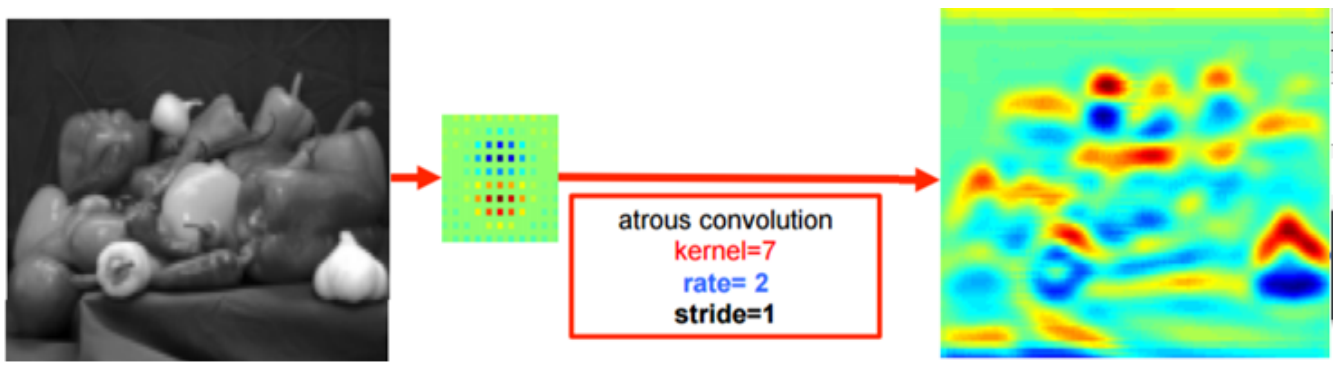
Standard convolution

$\frac{1}{4}$ of the positions



Atrous convolution

all positions



Output stride and receptive field

Output stride (OS) = input resolution / output resolution

Stride 1 in the last blocks, rates multiplied accordingly

	block 4	block 5	OS
classification	stride 2	stride 2	32
DeepLab OS 16	stride 2	stride 1, rate 2	16
DeepLab OS 8	stride 1, rate 2	stride 1, rate 4	8

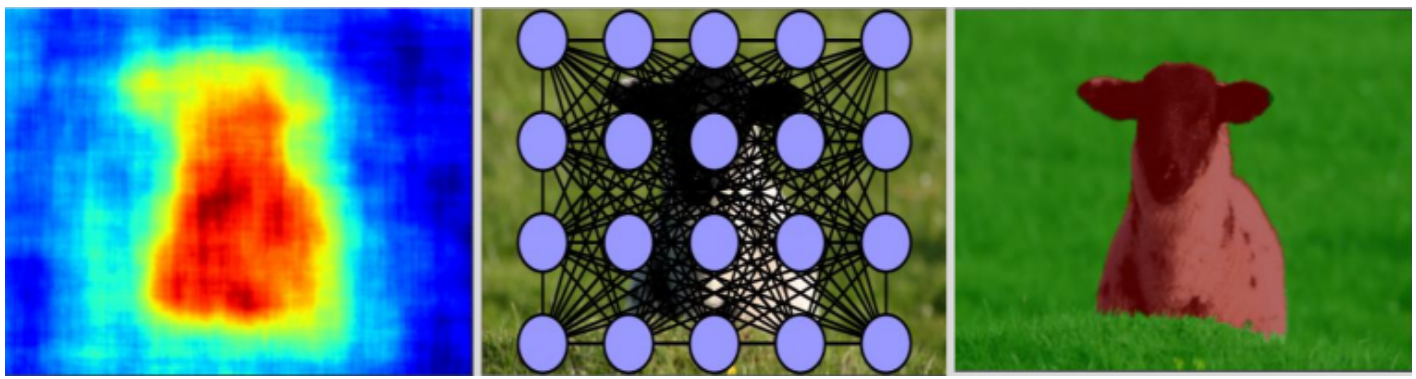
- Receptive field: $\mathrm{RF} = 1 + 2\sum_\ell r_\ell$ (rates 1, 2, 4, 8: 31 vs 9)
- **Cost:** 4× (OS 16) or 16× (OS 8) memory
- **Gridding** with a constant rate [Wang et al., 2018]

DeepLab: fully connected CRF

- CNN outputs are blurry at object boundaries

$$E(\mathbf{y}) = \sum_i \theta_i(y_i) + \sum_{i,j} \theta_{ij}(y_i, y_j)$$

- Unary: CNN output Pairwise: similar pixels, same label



Fully connected CRF in detail [Krähenbühl and Koltun, 2011]

- Unary: $\theta_i(y_i) = -\log p_\theta(y_i \mid \mathbf{x}, i)$
- Pairwise over **all** pixel pairs (position $\mathbf{p}$, colour $\mathbf{I}$):

$$\theta_{ij}(y_i, y_j) = [y_i \neq y_j]\; k(\mathbf{f}_i, \mathbf{f}_j)$$

$$k(\mathbf{f}_i, \mathbf{f}_j) = w_1 \underbrace{\exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\sigma_\alpha^2} - \frac{\|\mathbf{I}_i - \mathbf{I}_j\|^2}{2\sigma_\beta^2} \right)}_{\text{appearance}} + w_2 \underbrace{\exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\sigma_\gamma^2} \right)}_{\text{smoothness}}$$

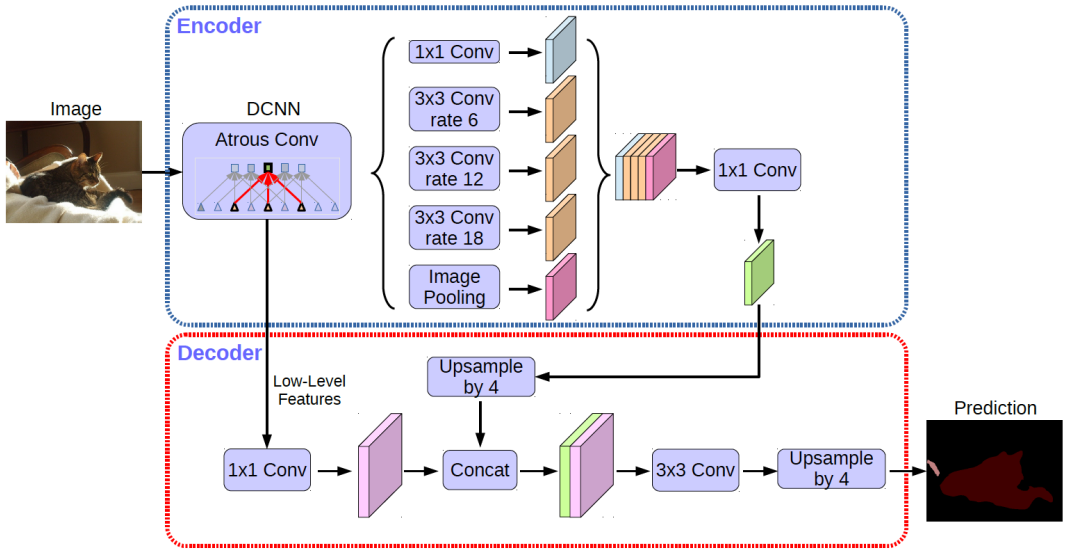
k: similarity of pixels i, j $\Rightarrow$ different labels on similar pixels are penalised

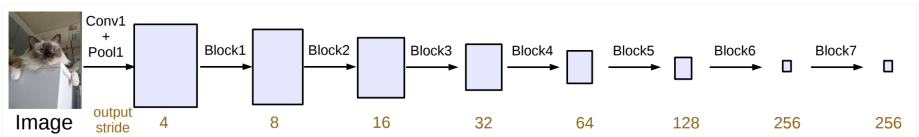
- **Mean-field** inference, $Q(\mathbf{y}) = \prod_i Q_i(y_i)$:

$$Q_i(c) \propto \exp\left(-\theta_i(c) - \sum_{c' \neq c} \sum_{j \neq i} k(\mathbf{f}_i, \mathbf{f}_j)\, Q_j(c') \right)$$

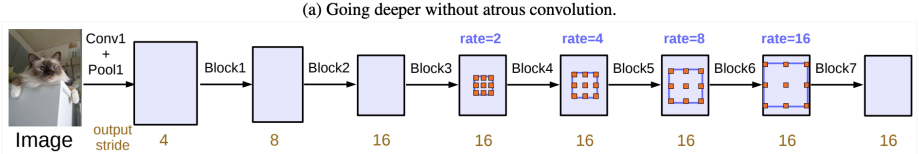
- Sum over j = Gaussian filtering $\Rightarrow$ linear cost
- Dropped from DeepLab v3 on (CRF-as-RNN [Zheng et al., 2015])

DeepLab v3+ [Chen et al., 2018b]





(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.

- ASPP encoder + light decoder

Atrous Spatial Pyramid Pooling and the DeepLab family

ASPP: several fields of view in parallel on the same map

$$\mathrm{ASPP}(\mathbf{F}) = \mathrm{conv}_{1\times1}\Big[\mathrm{conv}_{1\times1}(\mathbf{F}) \,\|\, \mathrm{conv}^{r=6}_{3\times3}(\mathbf{F}) \,\|\, \mathrm{conv}^{r=12}_{3\times3}(\mathbf{F}) \,\|\, \mathrm{conv}^{r=18}_{3\times3}(\mathbf{F}) \,\|\, \mathrm{up}\big(\mathrm{GAP}(\mathbf{F})\big)\Big]$$

($\|$: concatenation)

- Several scales at once; GAP = global context (cf. PSPNet [Zhao et al., 2017])
- v3: no CRF v3+: + decoder with a skip at OS 4

From FCN to DeepLab v3+

Version	Backbone	Context	Boundaries	mIoU†
FCN-8s	VGG16	–	skips	62.2
DeepLab v1	VGG16	atrous, LargeFOV	dense CRF	71.6
DeepLab v2	ResNet-101	ASPP	dense CRF	79.7
DeepLab v3	ResNet-101	ASPP + image pooling, BN	–	–
DeepLab v3+	Xception	ASPP	decoder (skip at OS 4)	89.0*

† PASCAL VOC 2012 test * with extra pre-training on JFT-300M

[Long et al., 2015, Chen et al., 2015, Chen et al., 2018a, Chen et al., 2017, Chen et al., 2018b]

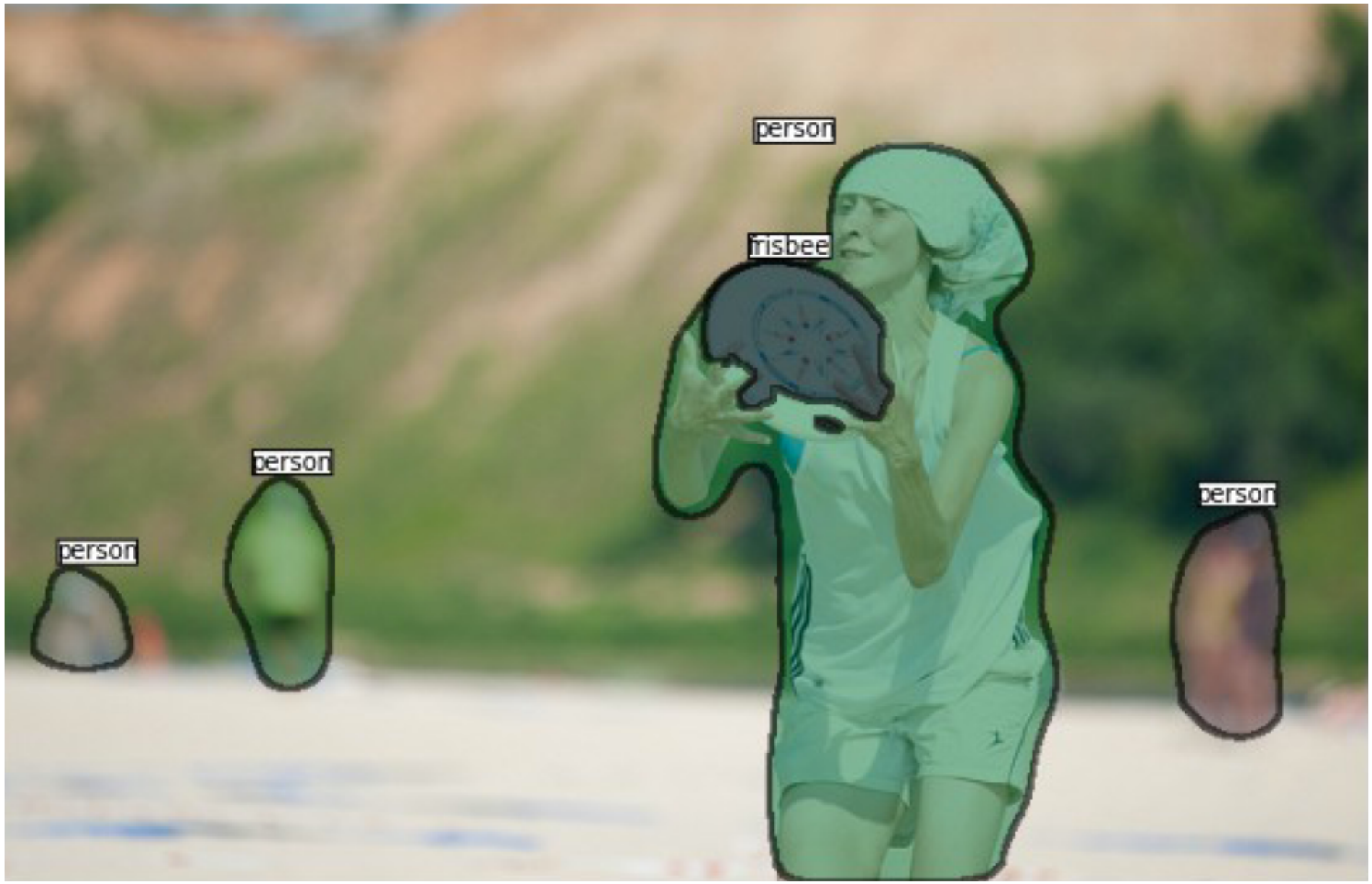
- +27 mIoU in three years: backbone, context, boundaries

Segmentation results



Images: Y. LeCun

Segmentation results



Images: Y. LeCun

- One mask **per object**: instance segmentation (Mask R-CNN [He et al., 2017])

Outline

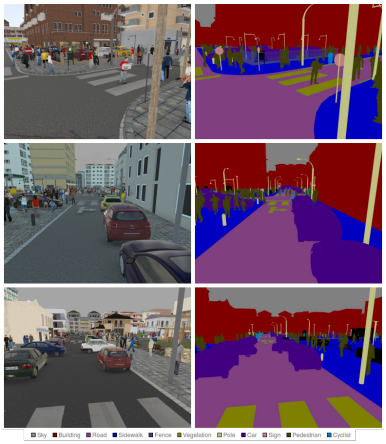
1. Segmentation: definitions and data
2. Supervised segmentation with deep ConvNets
3. Going further: segmentation after DeepLab

Where segmentation went after DeepLab

1. **Labels are expensive** $\Rightarrow$ synthetic data + domain adaptation
2. **Richer outputs:** instance, panoptic
3. **Transformers:** SETR, MaskFormer, Mask2Former
4. **Foundation models:** SAM, open vocabulary

1/ Pixel labels are expensive: back to lecture 4

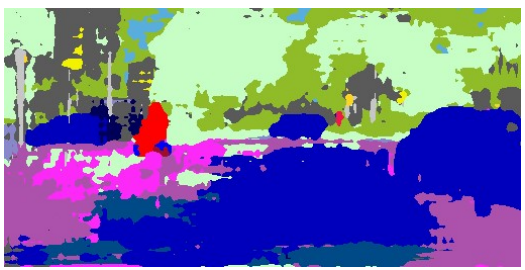
- Cityscapes: > 1.5 h per image [Cordts et al., 2016]
- GTA5 [Richter et al., 2016]: labels for free
- But a **domain gap**



source: synthetic, labelled



target: real (Cityscapes)



source only



after UDA (ADVENT)

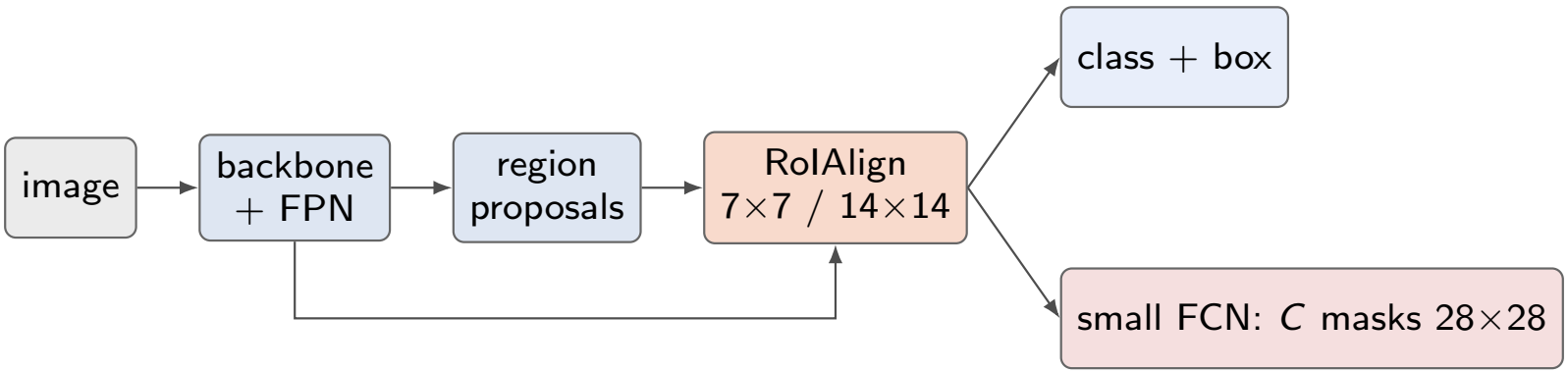
ADVENT [Vu et al., 2019] = **DeepLab-v2** + an adaptation loss

1/ Why adapt the *output* space for segmentation?

- Output maps share a **layout** across domains (sky above, road below)
- AdaptSegNet [Tsai et al., 2018]: discriminator on softmax maps
- ADVENT: entropy minimisation on target pixels

$$\min_{\theta} \frac{1}{HW} \sum_{i} - \sum_{c} p_{\theta}(c \mid \mathbf{x}^{t}, i) \log p_{\theta}(c \mid \mathbf{x}^{t}, i)$$

2/ Instance segmentation: Mask R-CNN [He et al., 2017]



- Detector + a small **FCN per box**
- $\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{box} + \mathcal{L}_{mask}$, binary mask per class
- **RoIAlign**: bilinear sampling, no rounding

2/ Panoptic segmentation [Kirillov et al., 2019]

- **Things** (car, person): instances **Stuff** (sky, road): regions
- Every pixel: a class **and** an instance id

	semantic	instance	panoptic
stuff (road, sky)	✓	✗	✓
things (car, person)	✓, merged	✓, separated	✓, separated
overlaps allowed	no	yes	no

Panoptic Quality

Segments matched if $\mathrm{IoU} > 0.5$

$$\mathrm{PQ} = \frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p,g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} = \underbrace{\frac{\sum_{TP} \mathrm{IoU}(p,g)}{|TP|}}_{\text{segmentation quality}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{recognition quality}}$$

3/ Transformers for segmentation

Self-attention: global context from the first layer

Per-pixel classification with a ViT

- SETR [Zheng et al., 2021]: ViT + conv decoder
- SegFormer [Xie et al., 2021]: hierarchical, MLP decoder
- Output: $H \times W \times C$

Mask classification

- MaskFormer [Cheng et al., 2021] $\rightarrow$ Mask2Former [Cheng et al., 2022]
- A **set** of N (mask, class) pairs
- One model for all tasks

Same transfer recipe as FCN, new backbone

SETR: segmentation as sequence-to-sequence [Zheng et al., 2021]

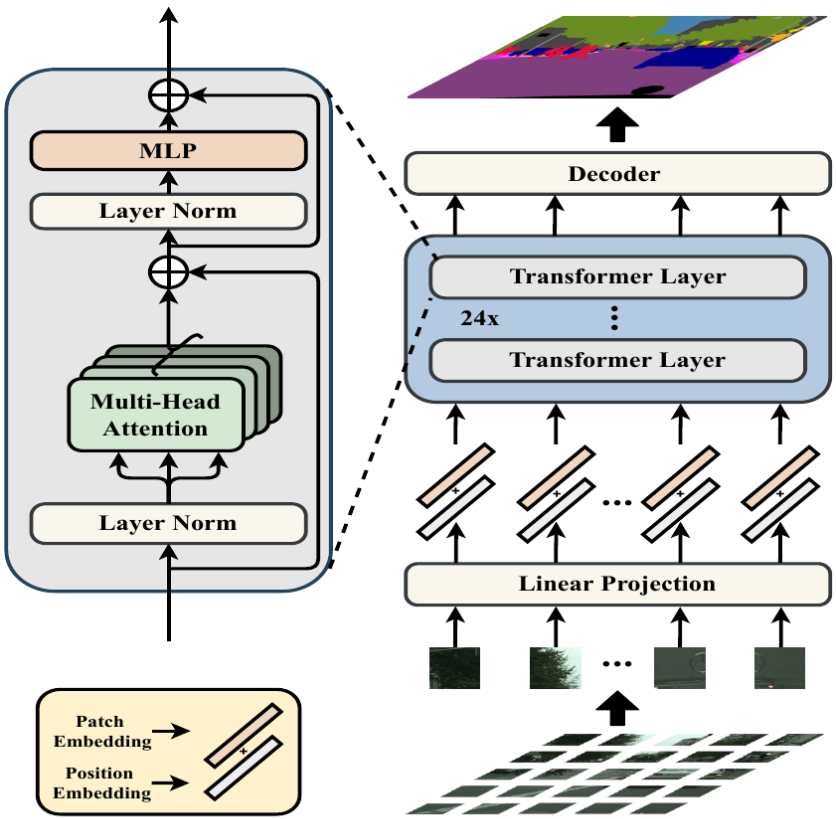


Figure from [Zheng et al., 2021]

- 16×16 patches $\rightarrow L = \frac{HW}{256}$ tokens
- Encoder: **ViT-Large**, 24 layers, ImageNet-21k
- **No downsampling**: $\frac{1}{16}$ everywhere
- **Global** receptive field
- Decoder: tokens $\rightarrow H \times W \times C$

SETR: the encoder in equations

Tokens

Patches $\mathbf{x}_p^i$, linear embedding E, position embeddings $\mathbf{p}_i$

$$Z^0 = \left[E\,\mathbf{x}_p^1 + \mathbf{p}_1;\ \ldots;\ E\,\mathbf{x}_p^L + \mathbf{p}_L \right] \in \mathbb{R}^{L \times D}$$

Transformer layer $\ell = 1..24$

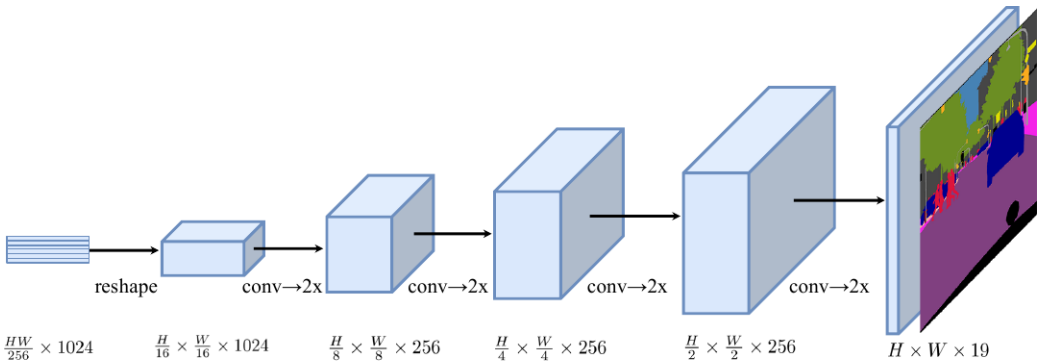
$$\tilde{Z}^\ell = Z^{\ell-1} + \mathrm{MSA}\big(\mathrm{LN}(Z^{\ell-1})\big), \qquad Z^\ell = \tilde{Z}^\ell + \mathrm{MLP}\big(\mathrm{LN}(\tilde{Z}^\ell)\big)$$

$$\mathrm{SA}(Z) = \mathrm{softmax}\Big(\frac{QK^\top}{\sqrt{d}}\Big)V, \qquad Q = ZW_Q,\ K = ZW_K,\ V = ZW_V$$

- Z^{24} reshaped to $\frac{H}{16} \times \frac{W}{16} \times D$
- Attention is $L \times L$: **quadratic** cost

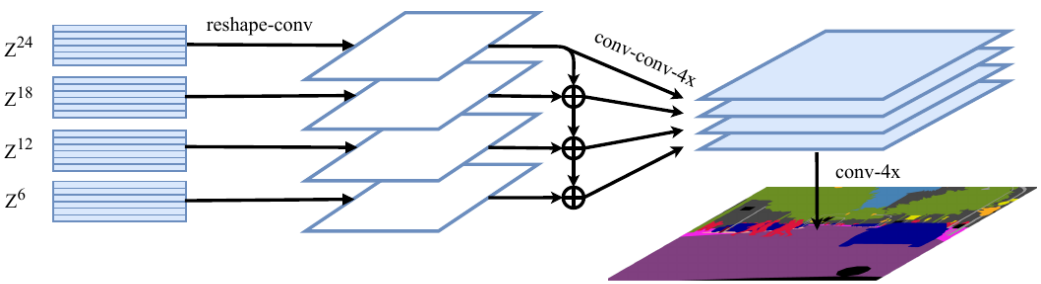
SETR: three decoders

PUP: progressive upsampling



(b)

MLA: multi-level feature aggregation



(c)

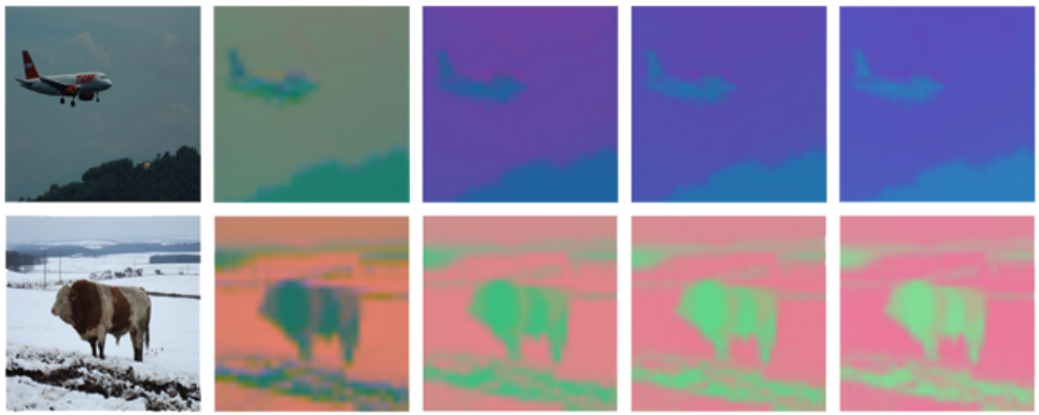
Figures from [Zheng et al., 2021]

- **Naive**: 1×1 conv + bilinear $\times 16$
- **PUP**: $4 \times$ (conv + up $2\times$)
- **MLA**: fuse $Z^6, Z^{12}, Z^{18}, Z^{24}$

MLA: skips across *depth*, not scales

SETR: what the encoder learns

Output features of layers Z^1, Z^9, Z^{17}, Z^{24}



Attention maps



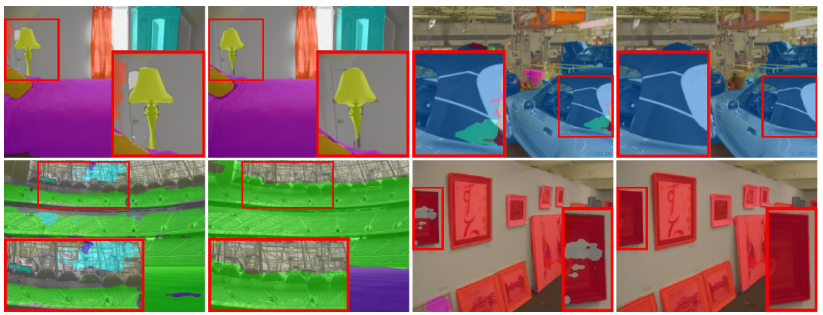
- Objects grouped early, without pooling

Figures from [Zheng et al., 2021], trained on Pascal Context

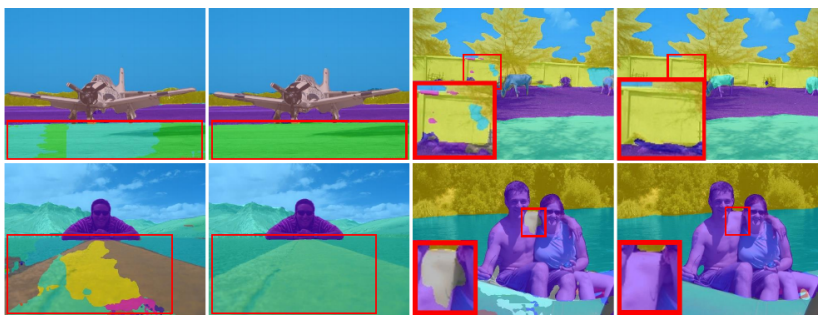
SETR: results



Cityscapes: dilated FCN (left) vs SETR (right)



ADE20K: FCN (left) vs SETR (right)



Pascal Context: FCN (left) vs SETR (right)

- Better on large, context-dependent regions (red boxes)
- **Limits:** $\frac{1}{16}$ only, quadratic cost $\Rightarrow$ hierarchical transformers

MaskFormer: per-pixel vs mask classification [Cheng et al., 2021]

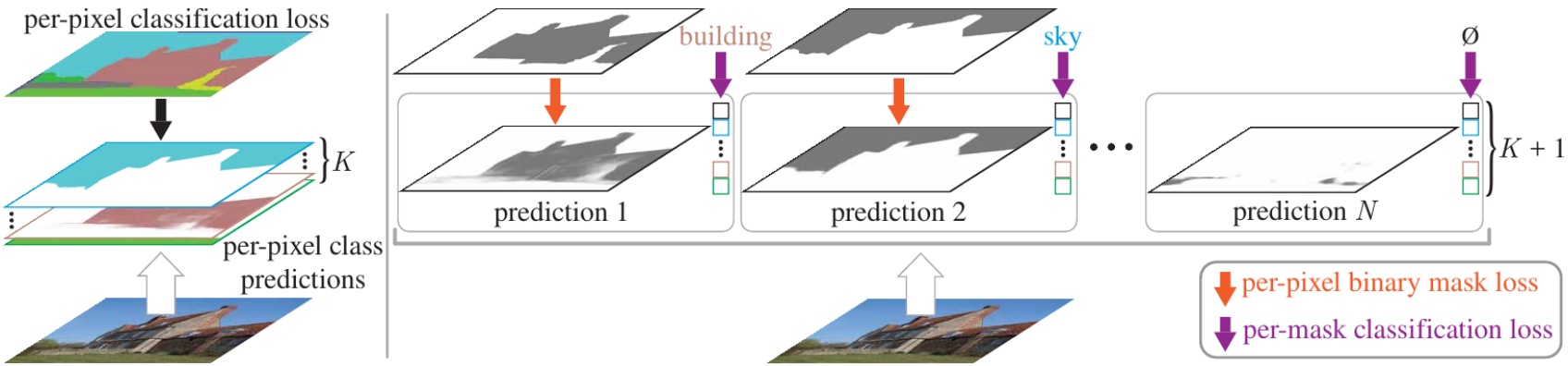


Figure from [Cheng et al., 2021]

Per-pixel classification (left)

- $H \times W$ distributions over K classes
- No instances

Mask classification (right)

- N pairs (mask m_i, class $p_i \in \Delta^{K+1}$)
- $+1$: "no object" $\varnothing$
- Instances for free

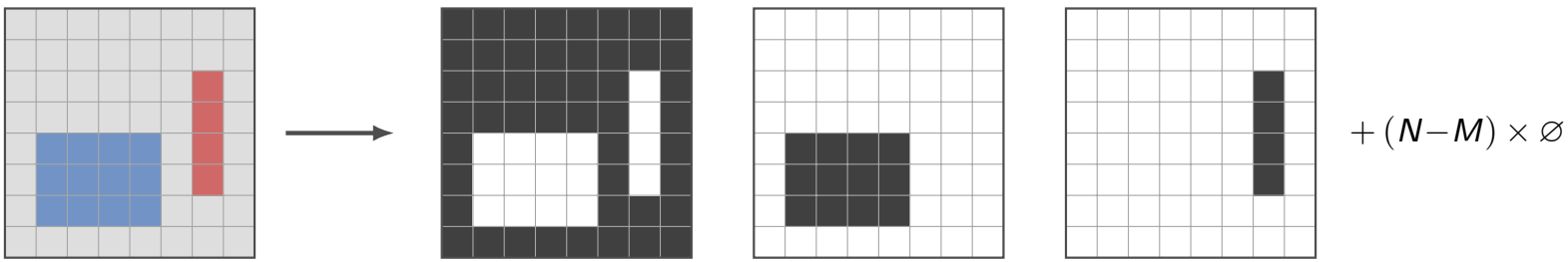
Per-pixel vs mask classification: formally

	Per-pixel	**Mask classification**
Output	$\{p_{hw}\}_{h,w}, \quad p_{hw} \in \Delta^K$	$z = \{(p_i, m_i)\}_{i=1}^N, \quad p_i \in \Delta^{K+1}$, $m_i \in [0,1]^{H \times W}$
Unit	pixel	segment
Size	fixed: $H \times W$	N slots, $M \leq N$ used
Loss	$\sum_{h,w} -\log p_{hw}(y_{hw})$	matching, then set loss
Pixel label	$\arg\max_c p_{hw}(c)$	$\arg\max_c \sum_i p_i(c)\, m_i[h,w]$
Instances	✗	✓

- Per-pixel = special case: $N = K$, query $i \leftrightarrow$ class i, fixed matching
- Mask classification: **same model and loss** for semantic, instance, panoptic

The ground-truth set

$$z^{gt} = \{(c_j, m_j^{gt})\}_{j=1}^{M}, \qquad c_j \in \{1..K\}, \qquad m_j^{gt} \in \{0,1\}^{H \times W}$$



Task	one pair per ...
Semantic	class present (all its pixels)
Instance	object instance (things)
Panoptic	instance for things, class region for stuff

- M varies per image; "ignore" pixels belong to no mask

MaskFormer: architecture

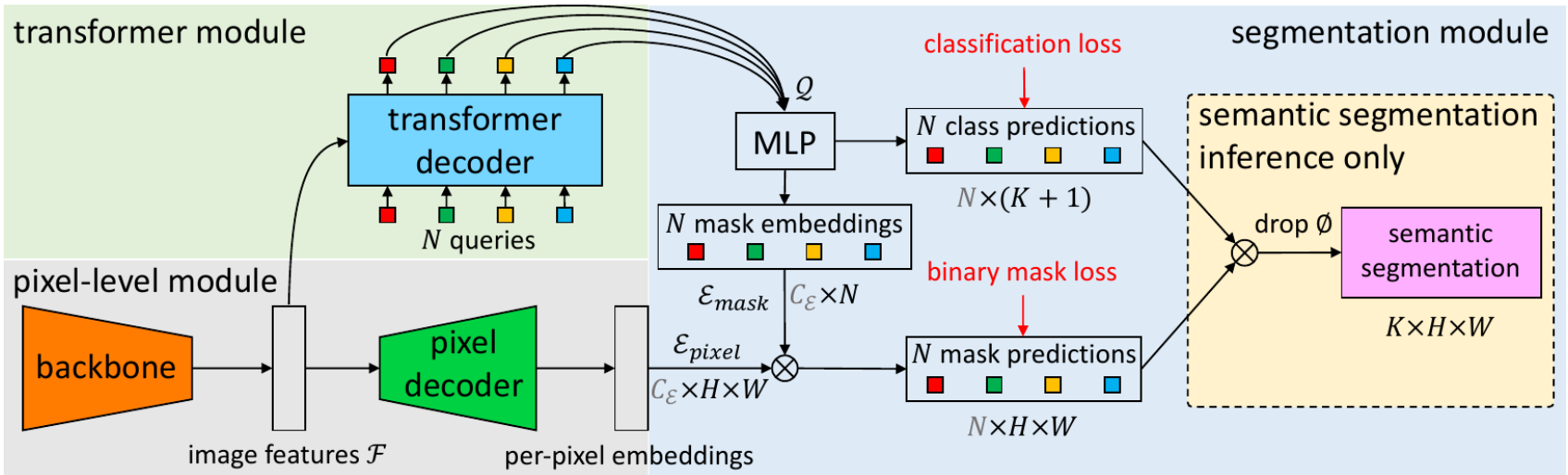


Figure from [Cheng et al., 2021]

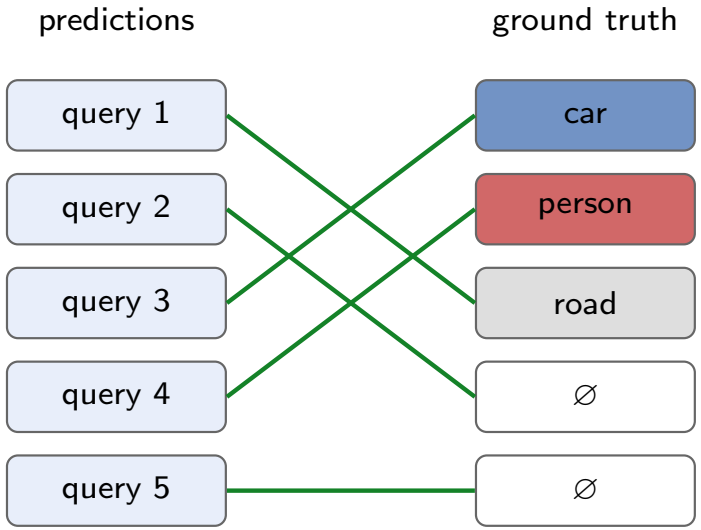
1. **Pixel module:** backbone + pixel decoder $\rightarrow \mathcal{E}_{pixel}$
2. **Transformer module:** N queries $\rightarrow$ N segment embeddings
3. **Segmentation module:** $m_i[h, w] = \sigma\left(\mathcal{E}_{mask}[i] \cdot \mathcal{E}_{pixel}[:, h, w]\right)$

MaskFormer: training with bipartite matching

- Ground truth padded with $\varnothing$ to N
- **Hungarian** matching, cost:

$$\mathcal{C}(i,j) = -p_i(c_j^{gt}) + \mathcal{L}_{mask}(m_i, m_j^{gt})$$

- One query per segment: no NMS



Loss on the matched pairs

$$\mathcal{L}_{mask\text{-}cls} = \sum_{j=1}^{N} \Big[-\log p_{\sigma(j)}(c_j^{gt}) + \mathbb{1}_{c_j^{gt} \neq \varnothing} \, \mathcal{L}_{mask}\big(m_{\sigma(j)}, m_j^{gt}\big) \Big]$$

The matching objective

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{j=1}^{N} \mathcal{C}\big(\sigma(j), j\big), \qquad \mathcal{C}(i,j) = \underbrace{-\,p_i(c_j)}_{\text{right class?}} + \underbrace{\mathbb{1}_{c_j \neq \varnothing}\, \mathcal{L}_{mask}(m_i, m_j^{gt})}_{\text{right shape?}}$$

	car	person	road	
query 1	1.2	1.1	−0.6	
query 2	0.9	0.8	0.7	→ ∅
query 3	−0.5	0.6	1.0	
query 4	0.7	−0.3	1.1	
query 5	0.4	0.2	0.9	→ ∅

duplicate of query 4

toy cost matrix $\mathcal{C}(i,j)$, $N = 5$, $M = 3$

- Low cost: right class **and** mask
- **Hungarian** [Kuhn, 1955], $O(N^3)$
- One-to-one ⇒ duplicates get ∅: no NMS
- Each query gets the segment it already explains best
- $-p$, not $-\log p$: bounded [Carion et al., 2020]

Matching cost $\neq$ training loss

	Matching cost $\mathcal{C}$	Loss $\mathcal{L}_{\text{mask-cls}}$
Role	**choose** targets	**update** weights
Gradient	none ($\hat{\sigma}$ discrete)	backpropagated
Mask term	decides the assignment	trains the masks

Why a mask term in the loss?

- $-\log p_{\sigma(j)}(c_j)$ does not depend on m_i
- No gradient to the pixel decoder
- $\Rightarrow$ masks never trained

Why a mask term in the cost?

- Two "person" segments: same class cost
- Mask cost breaks the tie
- $\Rightarrow$ stable assignment across iterations

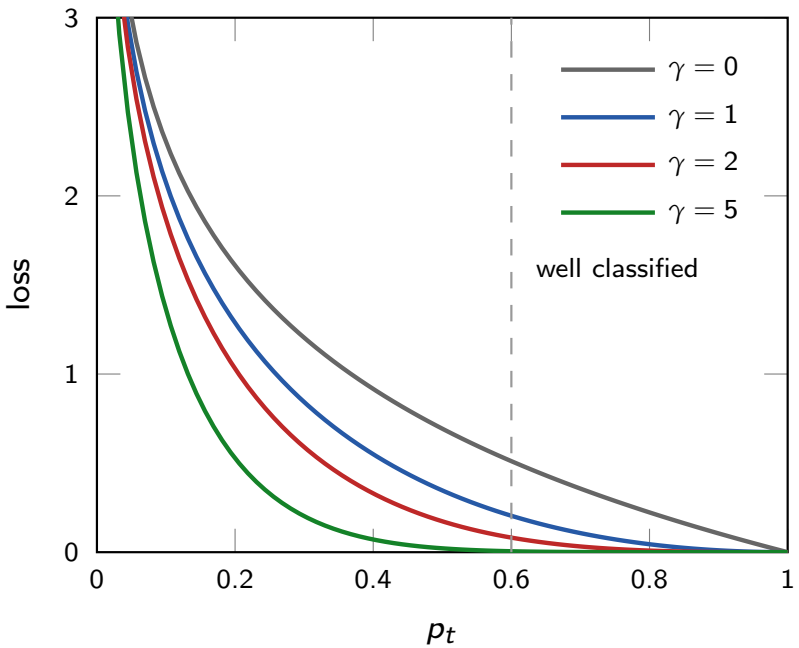
- Per iteration: match (no grad) $\rightarrow$ loss $\rightarrow$ backprop; at **every** decoder layer

Mask loss 1/2: focal loss [Lin et al., 2017]

Predicted $m \in [0,1]^{H \times W}$, target $g \in \{0,1\}^{H \times W}$

$$p_t = \begin{cases} m_p & g_p = 1 \\ 1 - m_p & g_p = 0 \end{cases}$$

$$\mathcal{L}_{focal} = -\frac{1}{HW} \sum_p \alpha_t \left(1 - p_t\right)^\gamma \log p_t$$

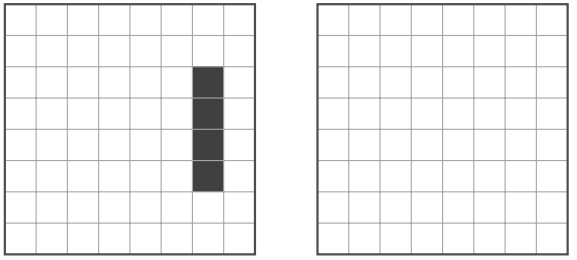


- $\gamma = 0$: (weighted) BCE
- $\gamma = 2$, $\alpha = 0.25$
- Easy pixels down-weighted
  - **Local**: per pixel, sharp and calibrated masks

Mask loss 2/2: dice loss [Milletari et al., 2016]

$$\mathcal{L}_{dice} = 1 - \frac{2 \sum_p m_p \, g_p + 1}{\sum_p m_p + \sum_p g_p + 1}$$

- Soft overlap (F1) of the whole mask
- **Global**: independent of object size



target "person" empty prediction

6 % of pixels wrong: BCE small

dice loss ≈ 1

Final mask loss

$$\mathcal{L}_{mask} = \lambda_{focal} \, \mathcal{L}_{focal} + \lambda_{dice} \, \mathcal{L}_{dice}$$

	weights	computed on
MaskFormer	20 focal + 1 dice	all pixels
Mask2Former	5 BCE + 5 dice	112^2 sampled points (uncertain ones favoured)

MaskFormer: inference and results

Semantic inference

$$\hat{y}[h, w] = \arg \max_{c \in \{1..K\}} \sum_{i=1}^{N} p_i(c)\ m_i[h, w]$$

Instance / panoptic inference

Drop $\varnothing$ queries; each pixel goes to $\arg\max_i\ p_i(c_i)\ m_i[h, w]$

- 55.6 mIoU (ADE20K), 52.7 PQ (COCO)
- Weak: instances (40.1 AP), slow training

Mask2Former: masked-attention mask transformer [Cheng et al., 2022]

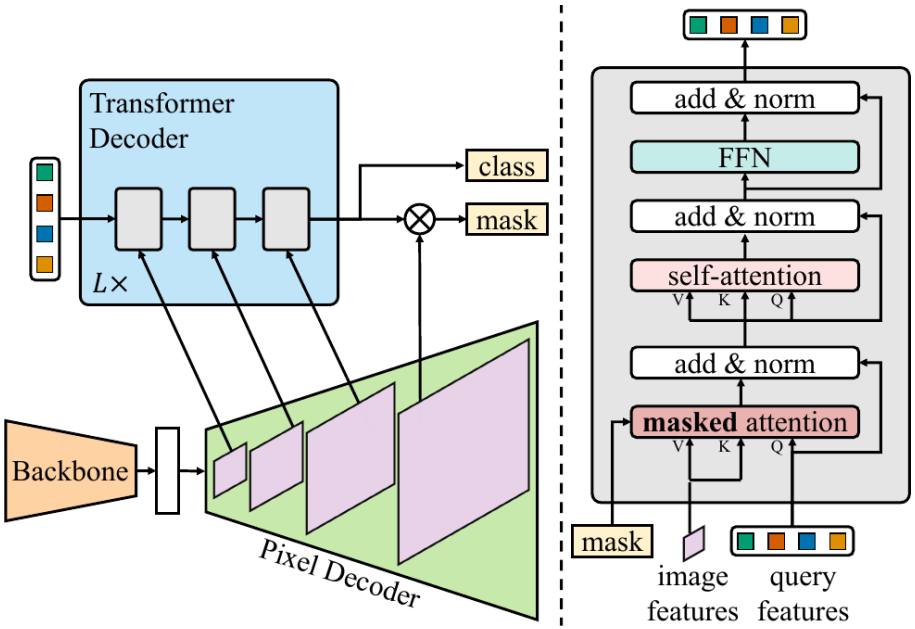


Figure from [Cheng et al., 2022]

MaskFormer + a new decoder:

1. **Masked** attention
2. **Multi-scale** features
3. Learnable queries
4. Loss on **sampled points**

Mask2Former: masked attention

Cross-attention:

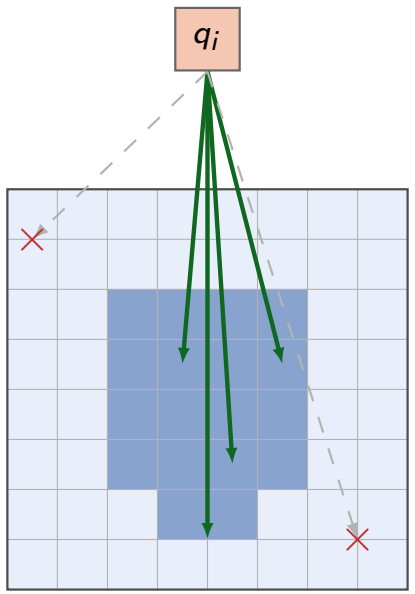
$$X_l = \text{softmax}\left(Q_l K_l^\top\right) V_l + X_{l-1}$$

Masked attention:

$$X_l = \text{softmax}\left(\mathcal{M}_{l-1} + Q_l K_l^\top\right) V_l + X_{l-1}$$

$$\mathcal{M}_{l-1}(x,y) = \begin{cases} 0 & \text{if } M_{l-1}(x,y) = 1 \\ -\infty & \text{otherwise} \end{cases}$$

M_{l-1}: mask of the previous layer



query i attends only to pixels of its mask M_{l-1}^i

- **Local** features, refined layer after layer; faster convergence

Mask2Former: the other ingredients

- **Multi-scale:** $\frac{1}{32}$, $\frac{1}{16}$, $\frac{1}{8}$, one per decoder layer
- **Learnable queries** act as region proposals
- **Sampled points** for the mask loss: $3\times$ less memory

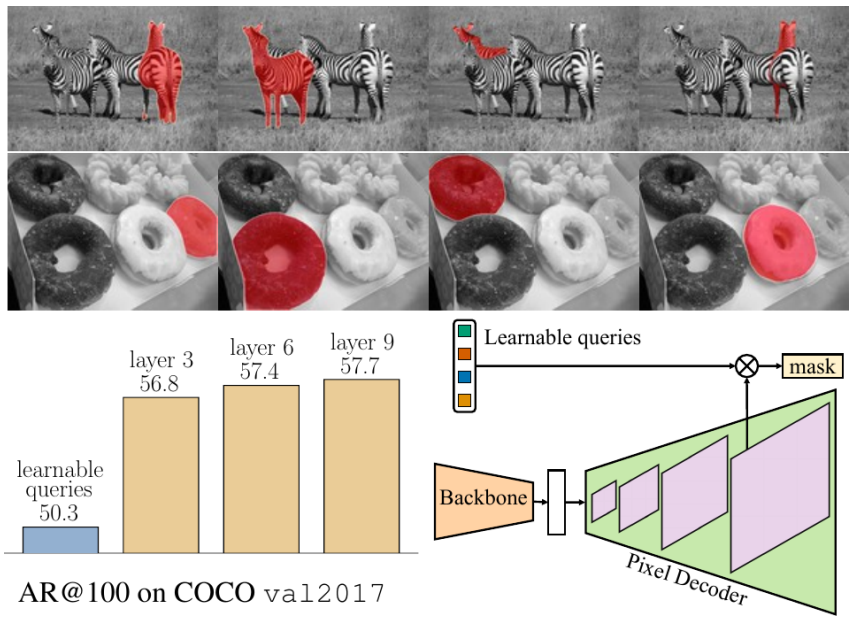


Figure from [Cheng et al., 2022]

Mask2Former: one model for three tasks

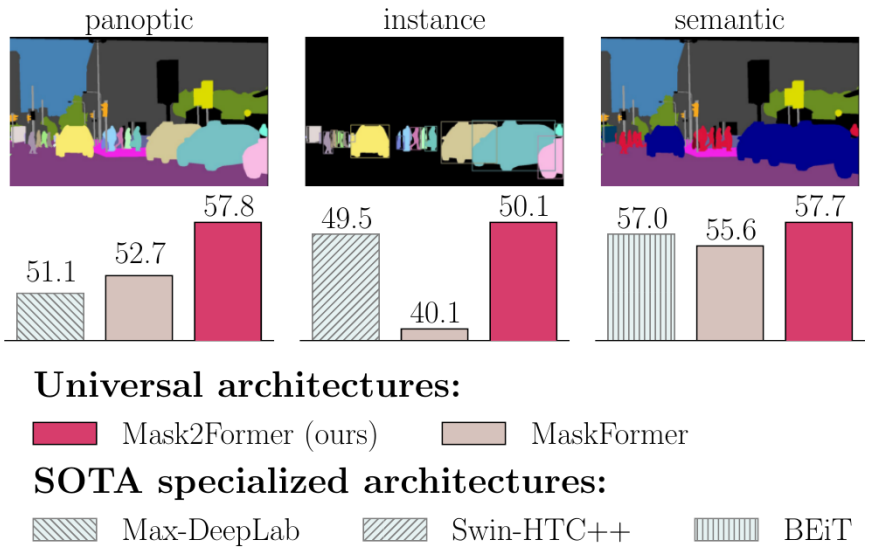


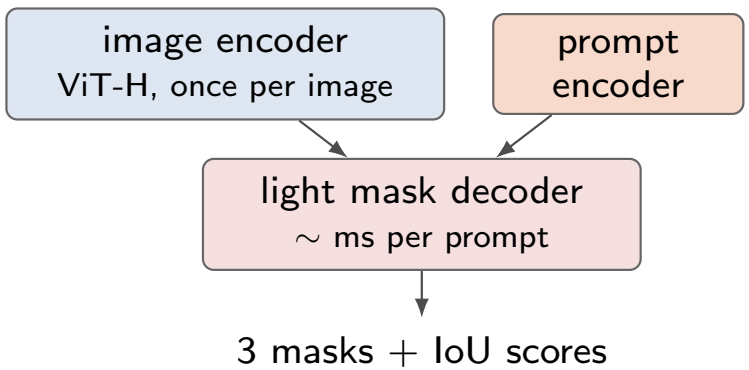
Figure from [Cheng et al., 2022]

- Only the annotations change
- Beats **specialised** models:
  - 57.8 PQ, 50.1 AP, 57.7 mIoU

	output	decoder	loss
FCN / DeepLab	$H \times W \times K$ scores	upsampling, ASPP	per-pixel CE
SETR	$H \times W \times K$ scores	conv on ViT tokens	per-pixel CE
MaskFormer	N (mask, class)	DETR decoder	matching
Mask2Former	N (mask, class)	masked attention	matching

4/ Segment Anything (SAM) [Kirillov et al., 2023]

- **Prompt** (points, box) → mask
- No classes
- Ambiguity: 3 masks + scores
- SA-1B: 11M images, 1.1B masks



- New domains (medical, satellite): fine-tuning again (lecture 4)
- SAM 2 [Ravi et al., 2024]: video

4/ Open-vocabulary segmentation

- Classes given as **text** (CLIP [Radford et al., 2021])

$$p(c \mid \mathbf{x}, i) = \frac{\exp\left(\langle \mathbf{f}_i(\mathbf{x}), \mathbf{t}_c \rangle / \tau\right)}{\sum_{c'} \exp\left(\langle \mathbf{f}_i(\mathbf{x}), \mathbf{t}_{c'} \rangle / \tau\right)}, \qquad \mathbf{t}_c = \text{text encoder}(\text{“a photo of a } c\text{”})$$

- LSeg [Li et al., 2022]: pixel embeddings matched to text
- New class = new sentence: **zero-shot** (lecture 4)
- DINOv2 [Oquab et al., 2024]: frozen features + **linear** head

Take-away

	2015–2018 (this lecture)	**Today**
Backbone	ImageNet CNN	ViT, foundation model
Context	atrous, ASPP, CRF	self-attention
Resolution	skips, decoders	multi-scale decoder
Output	$H \times W \times C$	(mask, class) pairs
Classes	fixed list	text, or none (SAM)

- Still there: **mIoU**, **skips**, **transfer from pre-training**

References I



Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017).

SegNet: A deep convolutional encoder-decoder architecture for image segmentation.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12):2481–2495.



Bearman, A., Russakovsky, O., Ferrari, V., and Fei-Fei, L. (2016).

What's the point: Semantic segmentation with point supervision.
In *European Conference on Computer Vision (ECCV)*.



Brosch, T., Tang, L. Y. W., Yoo, Y., Li, D. K. B., Traboulsee, A., and Tam, R. (2016).

Deep 3D convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation.
IEEE Transactions on Medical Imaging, 35(5):1229–1239.



Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020).

End-to-end object detection with transformers.
In *European Conference on Computer Vision (ECCV)*.



Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2015).

Semantic image segmentation with deep convolutional nets and fully connected CRFs.
In *International Conference on Learning Representations (ICLR)*.



Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018a).

DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4):834–848.



Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017).

Rethinking atrous convolution for semantic image segmentation.
arXiv preprint arXiv:1706.05587.



Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018b).

Encoder-decoder with atrous separable convolution for semantic image segmentation.
In *European Conference on Computer Vision (ECCV)*.

References II



Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., and Girdhar, R. (2022).

Masked-attention mask transformer for universal image segmentation.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.



Cheng, B., Schwing, A. G., and Kirillov, A. (2021).

Per-pixel classification is not all you need for semantic segmentation.
In *Advances in Neural Information Processing Systems (NeurIPS)*.



Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016).

The Cityscapes dataset for semantic urban scene understanding.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.



Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010).

The PASCAL visual object classes (VOC) challenge.
International Journal of Computer Vision, 88(2):303–338.



He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017).

Mask R-CNN.
In *IEEE International Conference on Computer Vision (ICCV)*.



Kirillov, A., He, K., Girshick, R., Rother, C., and Dollár, P. (2019).

Panoptic segmentation.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.



Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. (2023).

Segment anything.
In *IEEE International Conference on Computer Vision (ICCV)*.



Krähenbühl, P. and Koltun, V. (2011).

Efficient inference in fully connected CRFs with Gaussian edge potentials.
In *Advances in Neural Information Processing Systems (NeurIPS)*.

References III



Kuhn, H. W. (1955).

The Hungarian method for the assignment problem.
Naval Research Logistics Quarterly, 2(1–2):83–97.



Li, B., Weinberger, K. Q., Belongie, S., Koltun, V., and Ranftl, R. (2022).

Language-driven semantic segmentation.
In *International Conference on Learning Representations (ICLR)*.



Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017).

Focal loss for dense object detection.
In *IEEE International Conference on Computer Vision (ICCV)*.



Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014).

Microsoft COCO: Common objects in context.
In *European Conference on Computer Vision (ECCV)*.



Long, J., Shelhamer, E., and Darrell, T. (2015).

Fully convolutional networks for semantic segmentation.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.



Milletari, F., Navab, N., and Ahmadi, S.-A. (2016).

V-net: Fully convolutional neural networks for volumetric medical image segmentation.
In *International Conference on 3D Vision (3DV)*.



Noh, H., Hong, S., and Han, B. (2015).

Learning deconvolution network for semantic segmentation.
In *IEEE International Conference on Computer Vision (ICCV)*.



Odena, A., Dumoulin, V., and Olah, C. (2016).

Deconvolution and checkerboard artifacts.
Distill.

References IV



Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., et al. (2024).

DINOv2: Learning robust visual features without supervision.
Transactions on Machine Learning Research (TMLR).



Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021).

Learning transferable visual models from natural language supervision.
In *International Conference on Machine Learning (ICML).*



Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., et al. (2024).

SAM 2: Segment anything in images and videos.
arXiv preprint arXiv:2408.00714.



Richter, S. R., Vineet, V., Roth, S., and Koltun, V. (2016).

Playing for data: Ground truth from computer games.
In *European Conference on Computer Vision (ECCV).*



Ronneberger, O., Fischer, P., and Brox, T. (2015).

U-Net: Convolutional networks for biomedical image segmentation.
In *Medical Image Computing and Computer-Assisted Intervention (MICCAI).*



Tsai, Y.-H., Hung, W.-C., Schulter, S., Sohn, K., Yang, M.-H., and Chandraker, M. (2018).

Learning to adapt structured output space for semantic segmentation.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*



Vu, T.-H., Jain, H., Bucher, M., Cord, M., and Pérez, P. (2019).

ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*



Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., and Cottrell, G. (2018).

Understanding convolution for semantic segmentation.
In *IEEE Winter Conference on Applications of Computer Vision (WACV).*

References V



Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. (2021).

SegFormer: Simple and efficient design for semantic segmentation with transformers.
In *Advances in Neural Information Processing Systems (NeurIPS)*.



Yu, F. and Koltun, V. (2016).

Multi-scale context aggregation by dilated convolutions.
In *International Conference on Learning Representations (ICLR)*.



Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017).

Pyramid scene parsing network.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.



Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. S. (2015).

Conditional random fields as recurrent neural networks.
In *IEEE International Conference on Computer Vision (ICCV)*.



Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H. S., and Zhang, L. (2021).

Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.