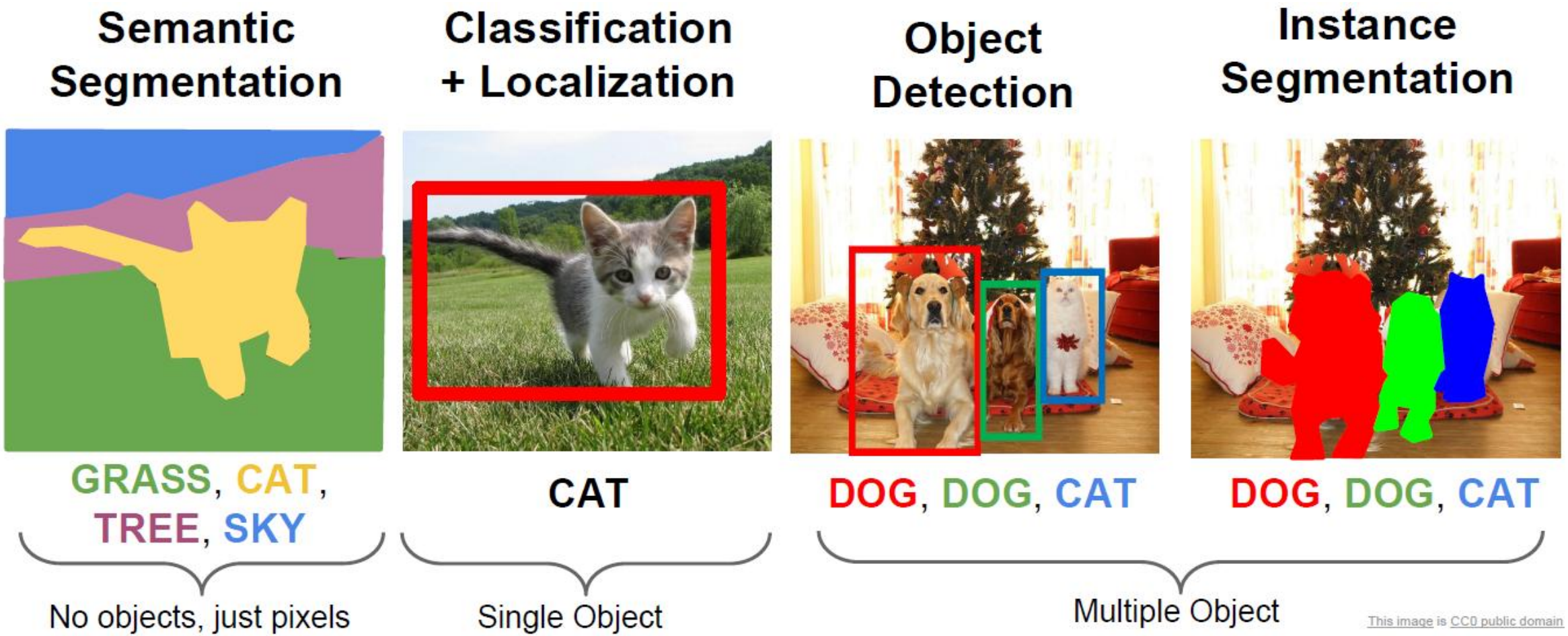
Lecture 8 Semantic Segmentation and Object Detection

Tian Sheuan Chang

Outline

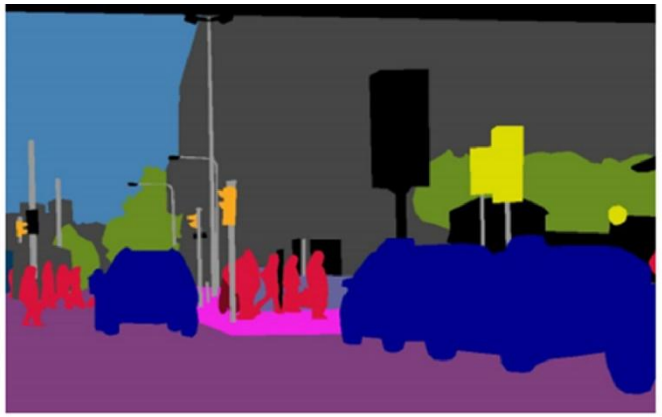
- Semantic segmentation
- Object detection
- Instance segmentation

Object Detection and Semantic Segmentation



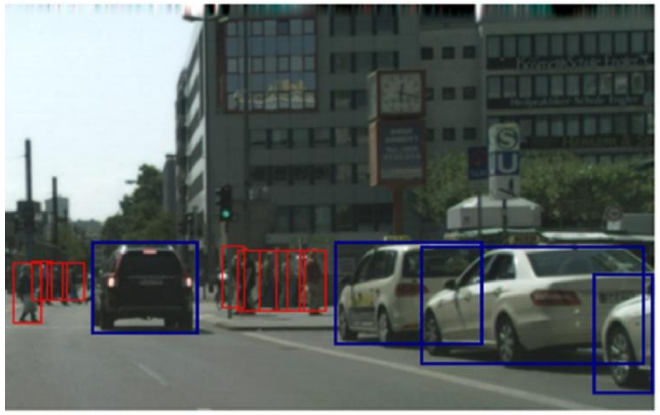
Source: cs231n

(a) Image

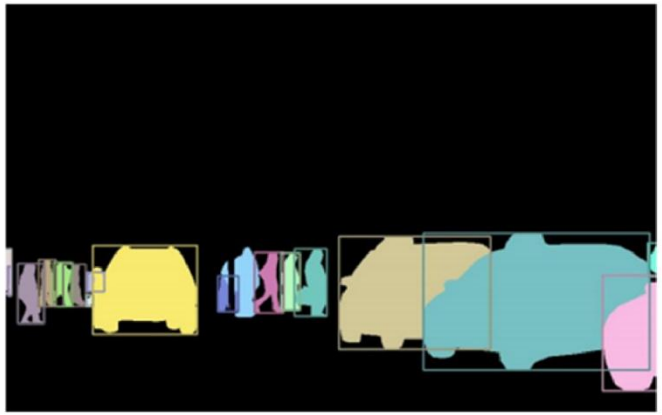


(b) Semantic segmentation

(c) Image classification



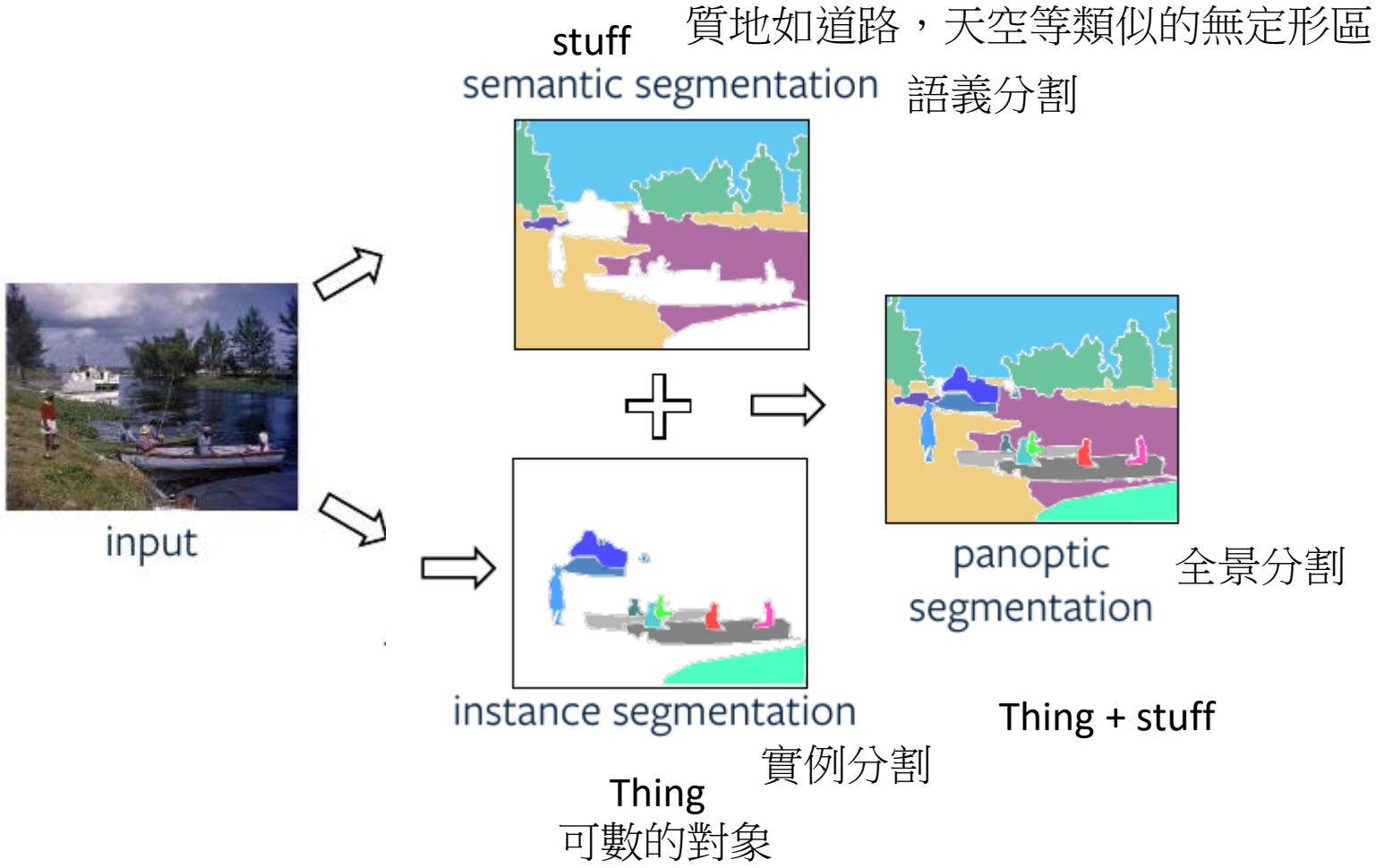
(d) Object detection



(e) Instance segmentation

(f) Panoptic segmentation

A Brief Survey on Semantic Segmentation with Deep Learning



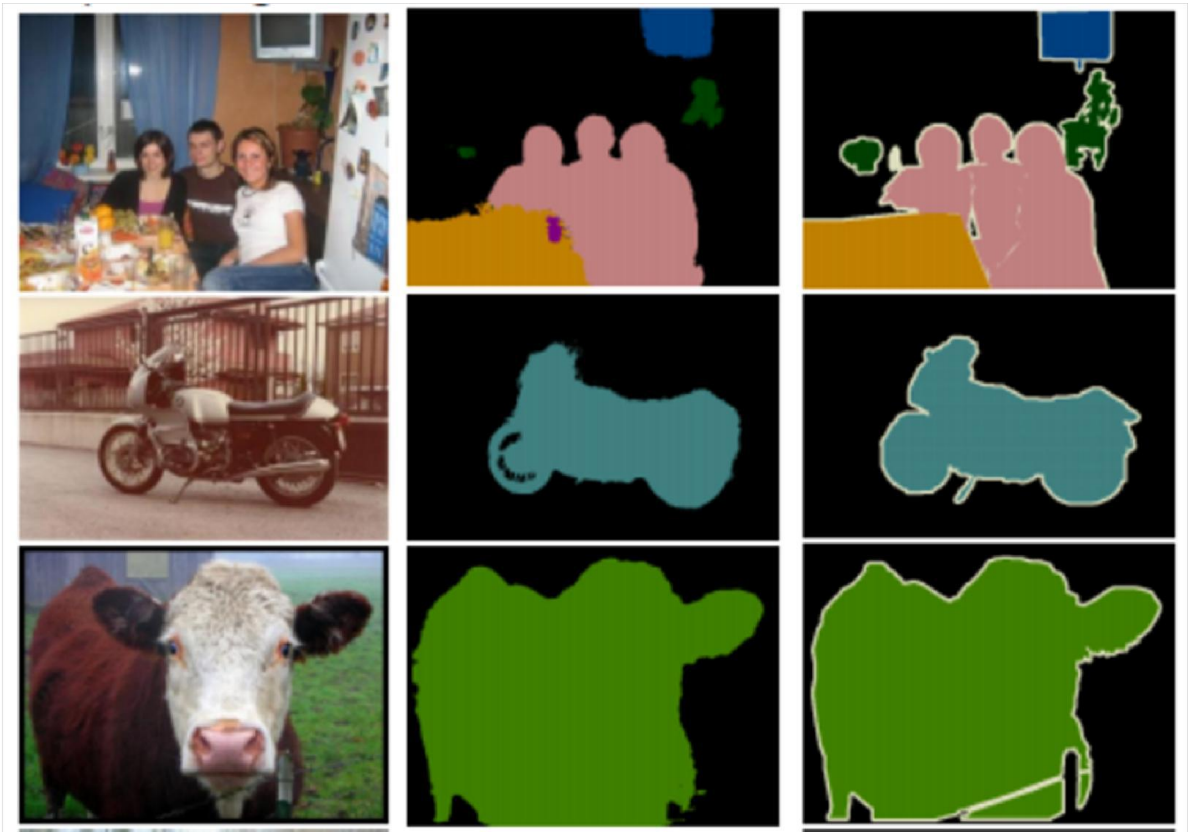
https://kharshit.github.io/blog/2019/10/18/introduction-to-panoptic-segmentation-tutorial

SEMANTIC SEGMENTATION

https://nanonets.com/blog/semantic-image-segmentation-2020/

Semantic Segmentation

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



Use Cases of Image Segmentation

Google portrait mode

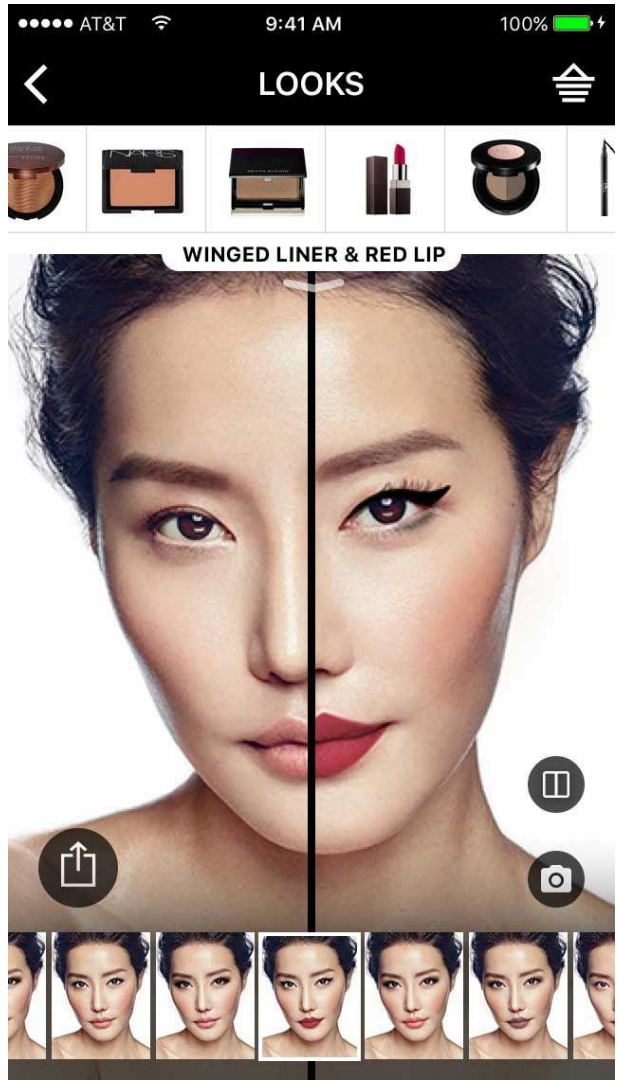
separate foreground from background

background blurred out

YouTube stories

different backgrounds

while creating stories.

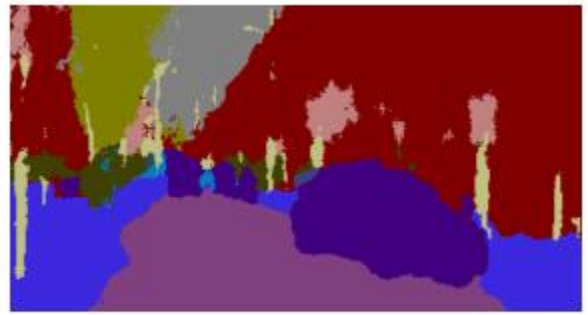


Virtual make-up

Why Semantic Segmentation

- Scene understanding
  - Useful for autonomous navigation of cars and drones

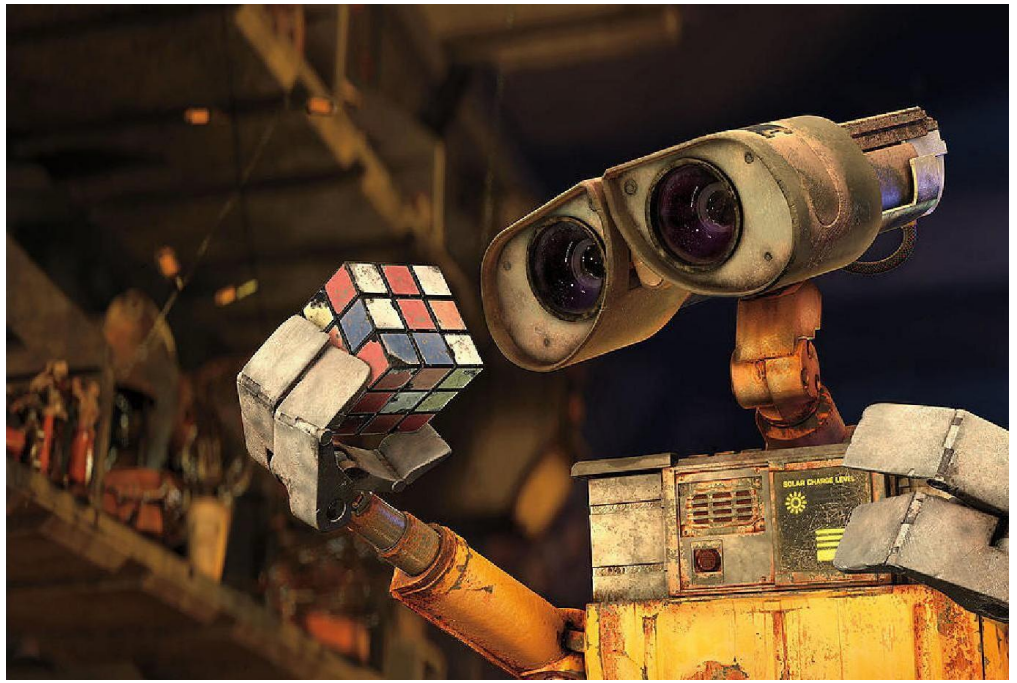
(a) Input Image



(b) Semantic Segmentation

Why Semantic Segmentation

- Help partially sighted people by highlighting important objects
- Let robots segment objects so that they can grasp them



Why Semantic Segmentation

- Medical purposes
  - e.g. segmenting tumours, dental cavities, ...

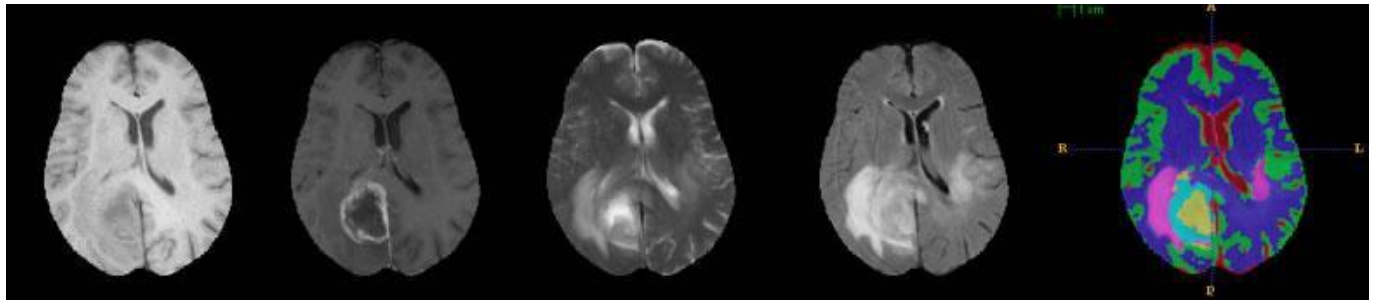
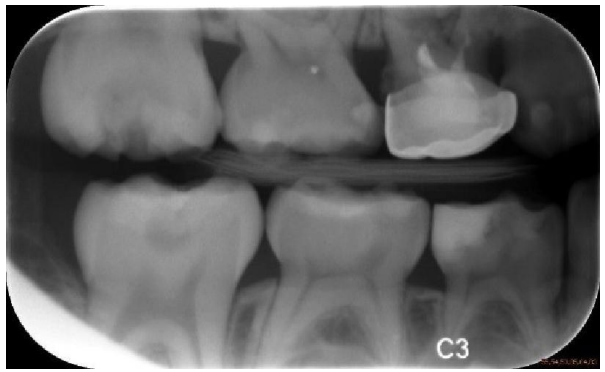
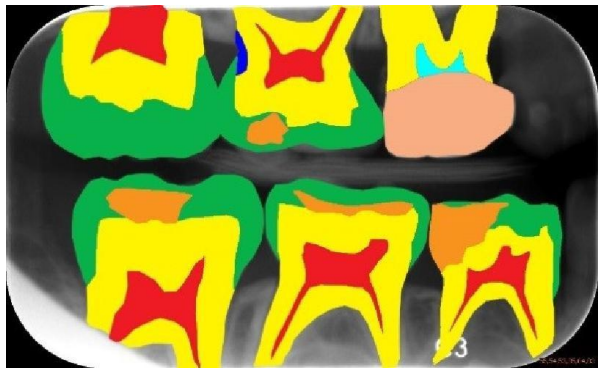


Image taken from Mauricio Reyes





ISBI Challenge 2015, dental x-ray images

Datasets for Semantic/Instance Segmentation

Pascal Visual Object Classes

- 20 categories
- +10,000 images
- Semantic segmentation GT
- Instance segmentation GT

Pascal Context

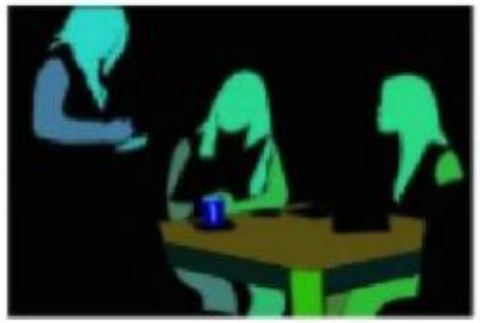
- Real indoor & outdoor scenes
- 540 categories
- +10,000 images
- Dense annotations
- Semantic segmentation GT
- Objects + stuff

Datasets for Semantic/Instance Segmentation

CityScapes

- Real driving scenes
- 30 categories
- +25,000 images
- 20,000 partial annotations
- 5,000 dense annotations
- Semantic segmentation GT
- Instance segmentation GT
- Depth, GPS and other metadata
- Objects and stuff

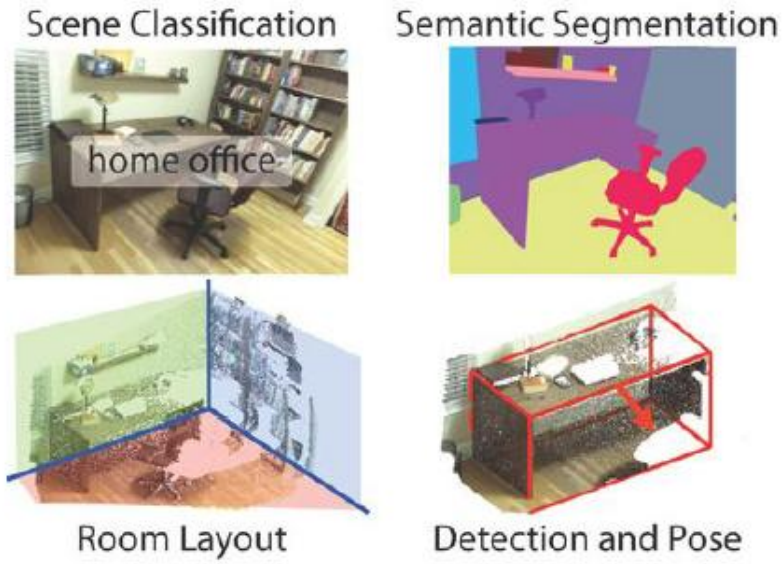
ADE20K



- Real general scenes
- +150 categories
- +22,000 images
- Semantic segmentation GT
- Instance + parts segmentation GT
- Objects and stuff

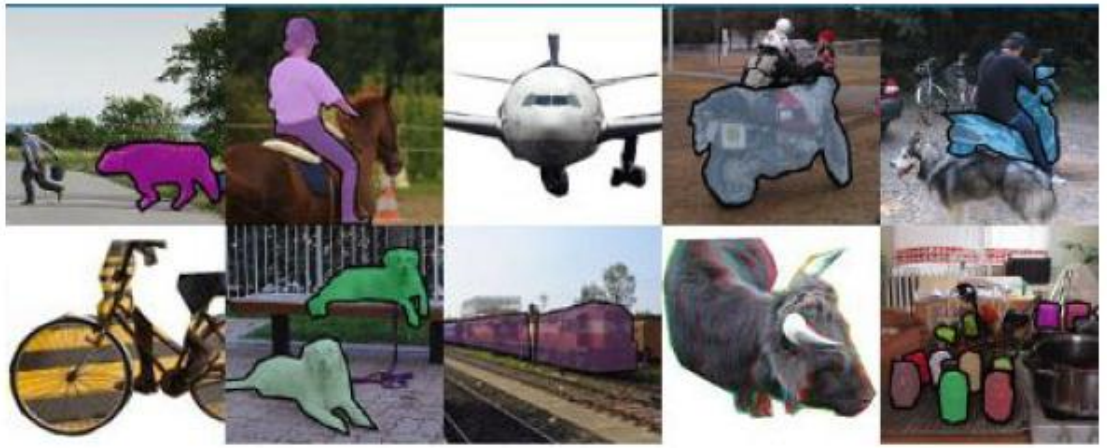
Datasets for Semantic/Instance Segmentation

SUN RGB-D



- Real indoor scenes
- 10,000 images
- 58,658 3D bounding boxes
- Dense annotations
- Instances GT
- Semantic segmentation GT
- Objects + stuff

COCO Common Objects in Context



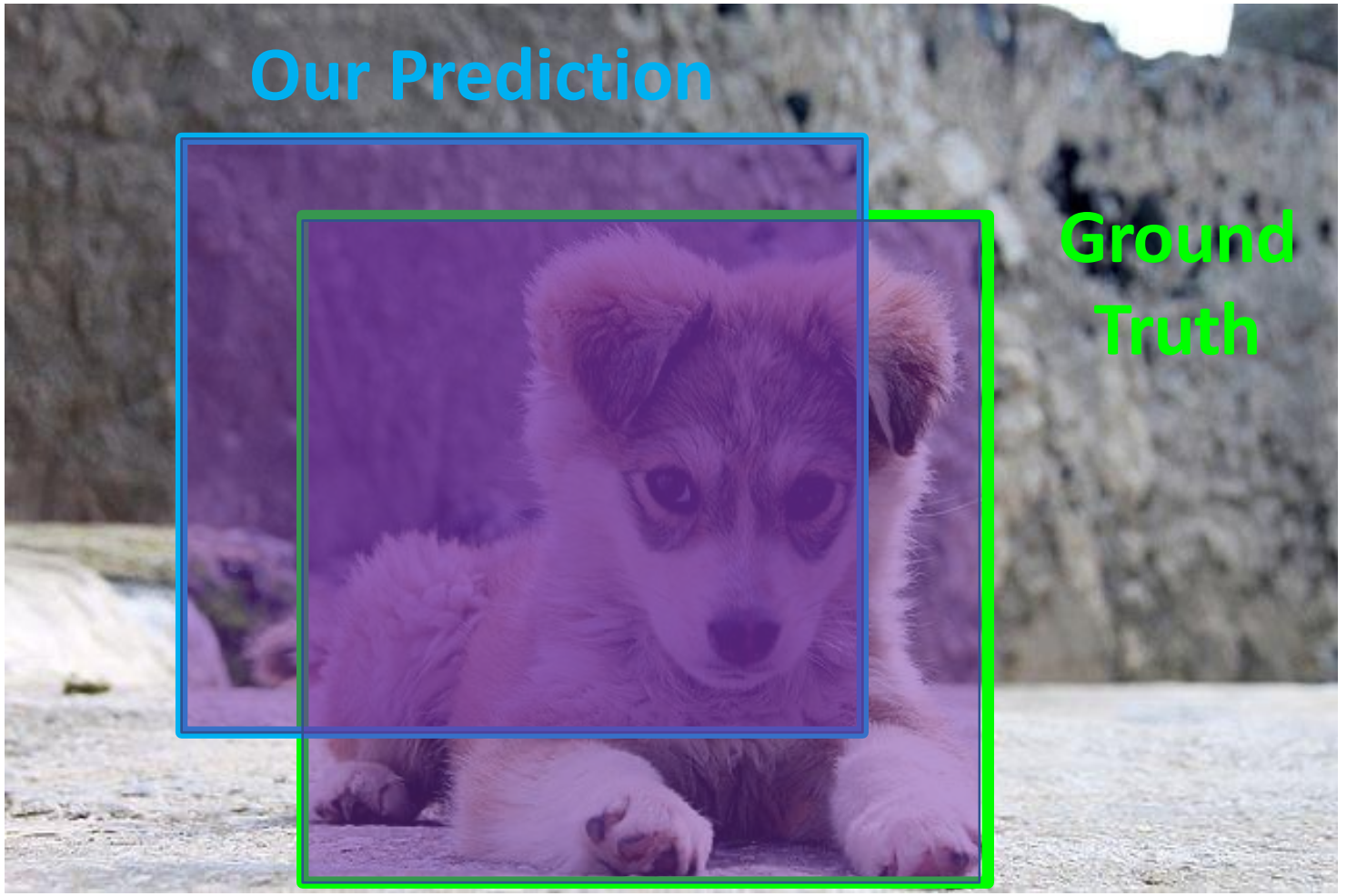
- Real indoor & outdoor scenes
- 80 categories
- +300,000 images
- 2M instances
- Partial annotations
- Semantic segmentation GT
- Instance segmentation GT
- Objects, but no stuff

Comparing Boxes: Intersection over Union (IoU)

How can we compare our prediction to the ground-truth box?

Intersection over Union (IoU)
(Also called “Jaccard similarity” or “Jaccard index”):

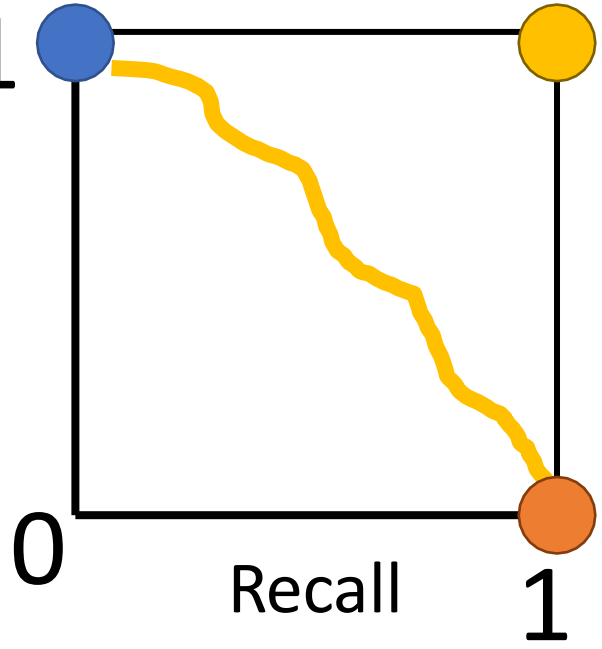
$$\frac{Area\ of\ Intersection}{Area\ of\ Union}$$



Precision & Recall

- True detection: high intersection over union
  - Choose IoU threshold
- Precision: #true detections / #detections
- Recall: #true detections / #true positives

Reject everything: no mistakes



Ideal!

Summarize by area under curve (avg. precision)

Accept everything: Miss nothing

Car AP = 0.65

Cat AP = 0.80

Dog AP = 0.86

mAP@0.5 = 0.77

mAP@0.5 = 0.77

mAP@0.55 = 0.71

mAP@0.60 = 0.65

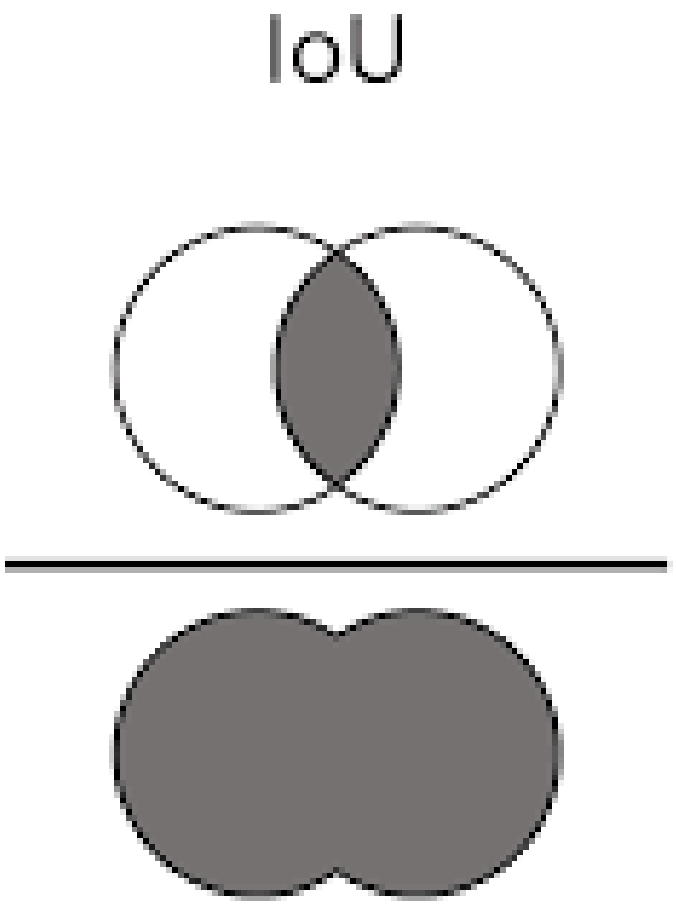
…

mAP@0.95 = 0.2

COCO mAP = 0.4

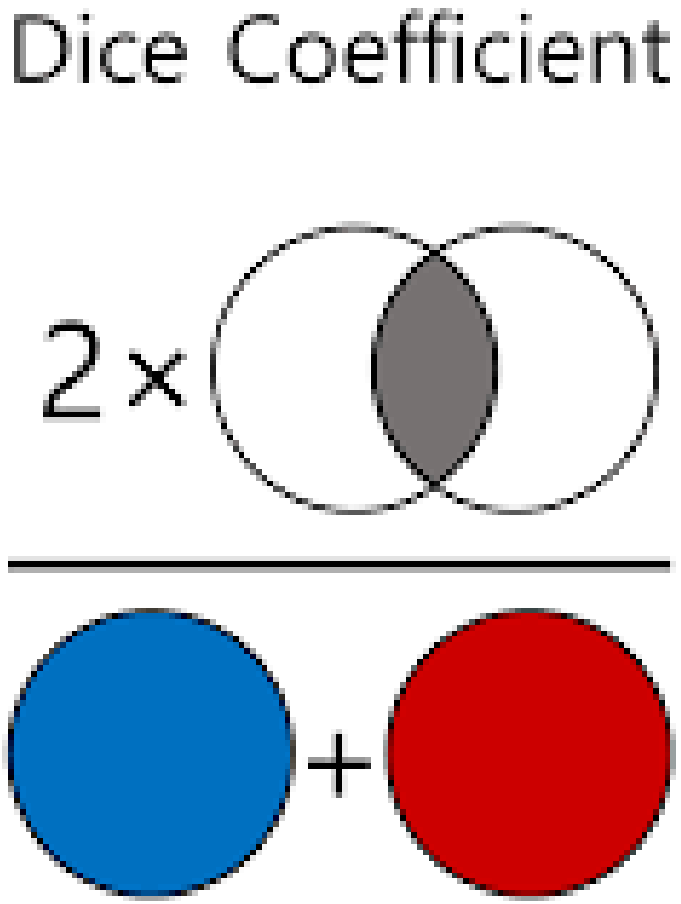
How to measure segmentation accuracy?

IoU (Intersection over Union)



通用場景 (General Purpose) / 標準競賽

DICE Score (F1 score)



嚴重類別不平衡（例如，前景像素只佔 0.1%）時的首選
小目標 / 類別不平衡 (Small Objects / Imbalance)：使用 Dice這是 Dice 最重要的應用場景，尤其在醫療影像（如腫瘤、器官分割）中

SEMANTIC SEGMENTATION

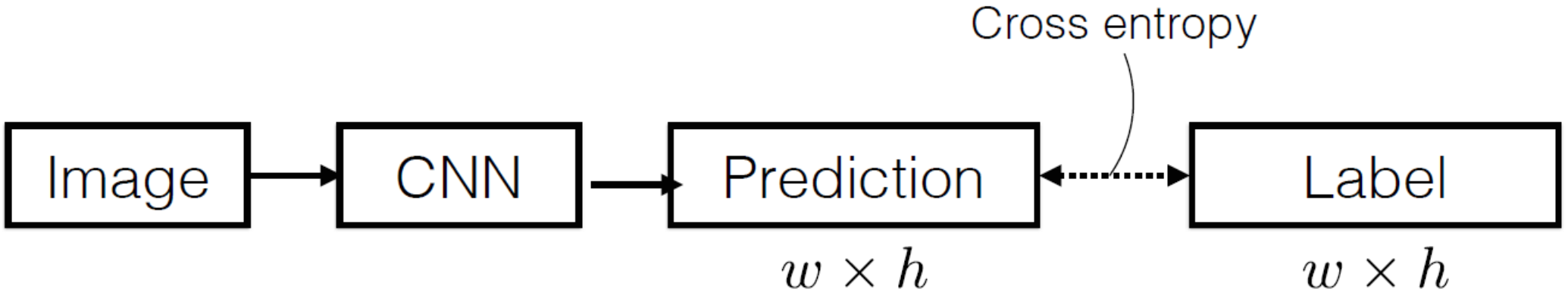
FROM CLASSIFICATION TO SEMANTIC SEGMENTATION

FULLY CONVOLUTIONAL NETWORK

J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in CVPR, 2015

Typical formulation

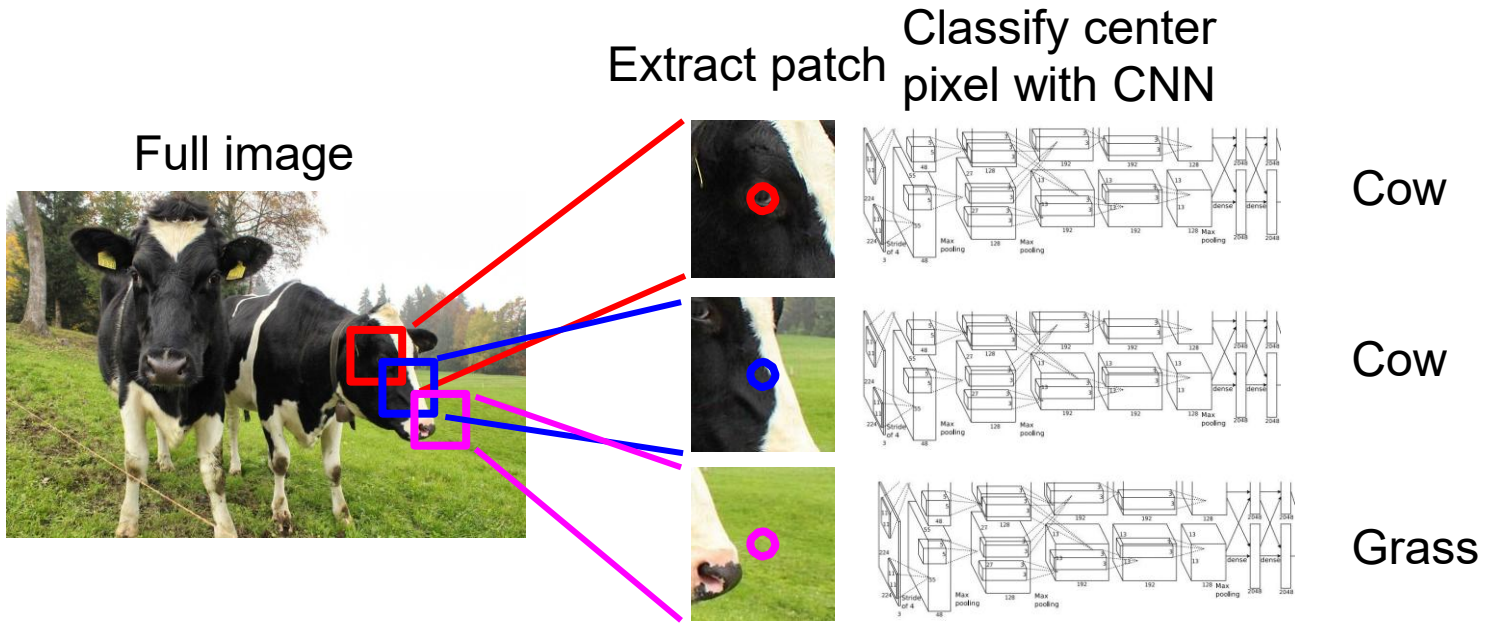
- Tackling this problem with CNN, usually it's formulated as:



- The loss is calculated for **each pixel independently**
- It leads to the problem: "How can the model consider the context to make a single prediction for a pixel?"
  - leverage context information
  - dense prediction (aka. Pixel level prediction)

Semantic Segmentation

- A straightforward idea: pixel level classification with sliding window



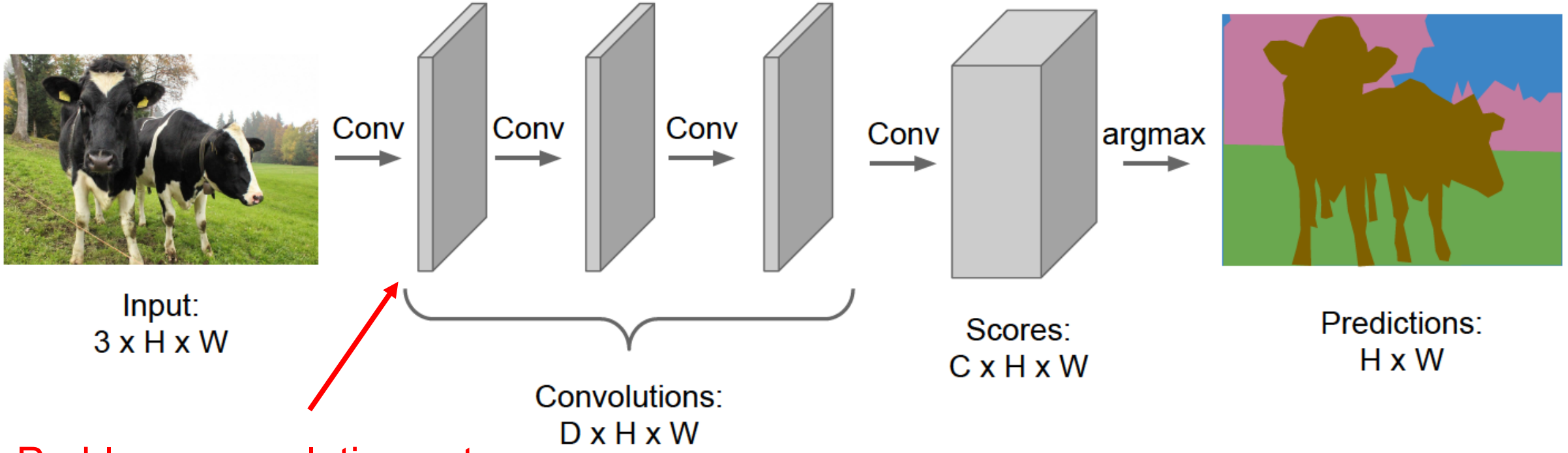
Problem: Very inefficient! Not reusing shared features between overlapping patches

Source: cs231n

Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013
Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, IC

Semantic Segmentation Idea: Fully Convolutional Network

Design a network as a bunch of convolutional layers to make predictions for pixels all at once!

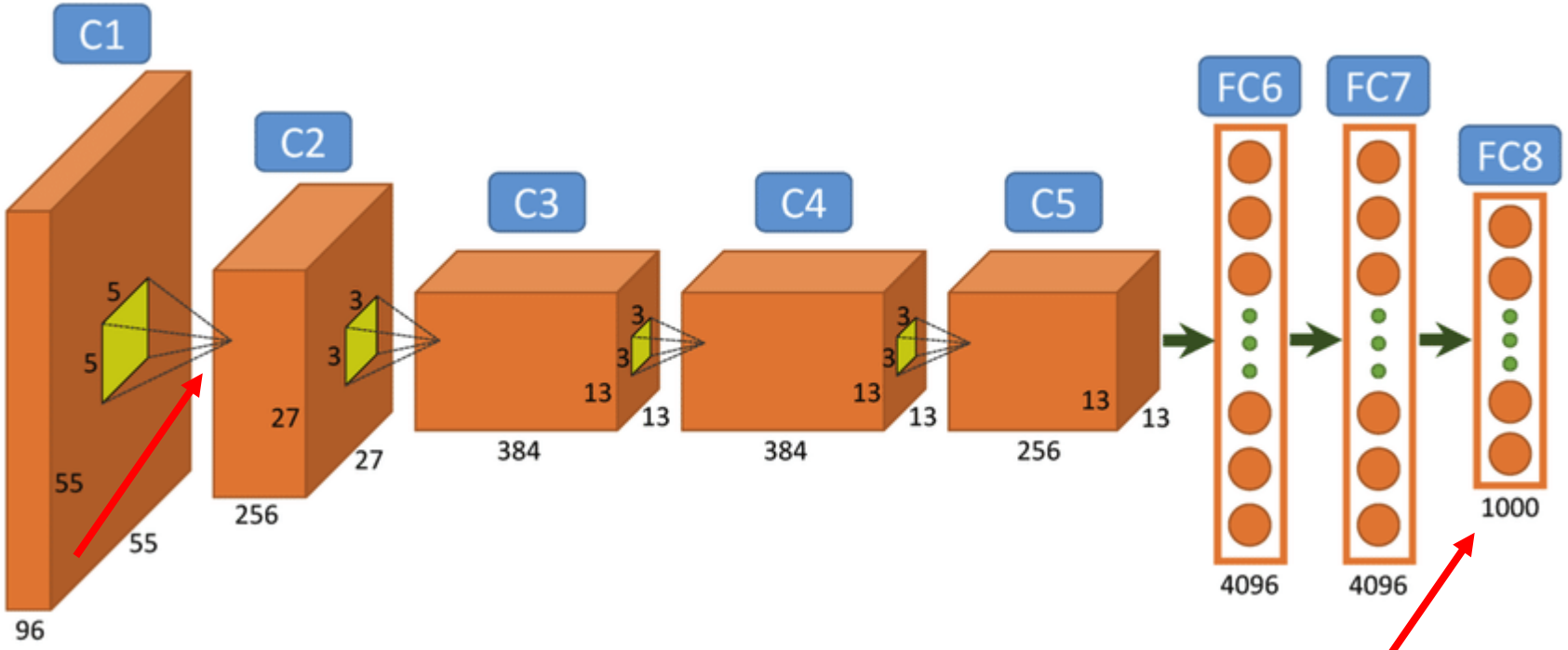


Problem: convolutions at original image resolution will be very expensive ... → Try downsampling => pooling? Strided convolution

Source: cs231n

Semantic Segmentation Idea: Fully Convolutional Network

- Problems to convert classification CNN to semantic segmentation



Problem: pooling reduces spatial resolution and also loss position information
=> upsampling

Problem: Fully connected layer has fixed size input/output
=> Fully convolutional layer

From classification to semantic segmentation

- Target: dense prediction
  - fully convolutional, end to end, pixel-to-pixel network

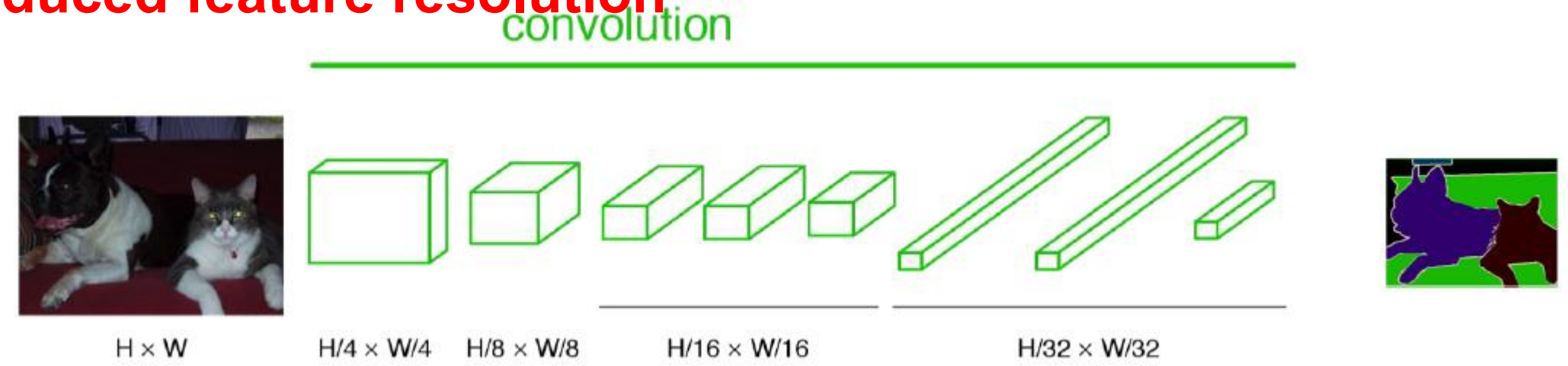
Challenges

- Resolution
  - high computational cost for high resolution images
    - Solution: add pooling or strided convolution to reduce image size
  - Problems with pooling layers or strided convolution
    - Output is much smaller than input → Add upsampling layers
    - Output is very coarse → Add fine details from previous layers
- Multi-scale: existence of objects at multiple scales
  - Multi-scale feature merging

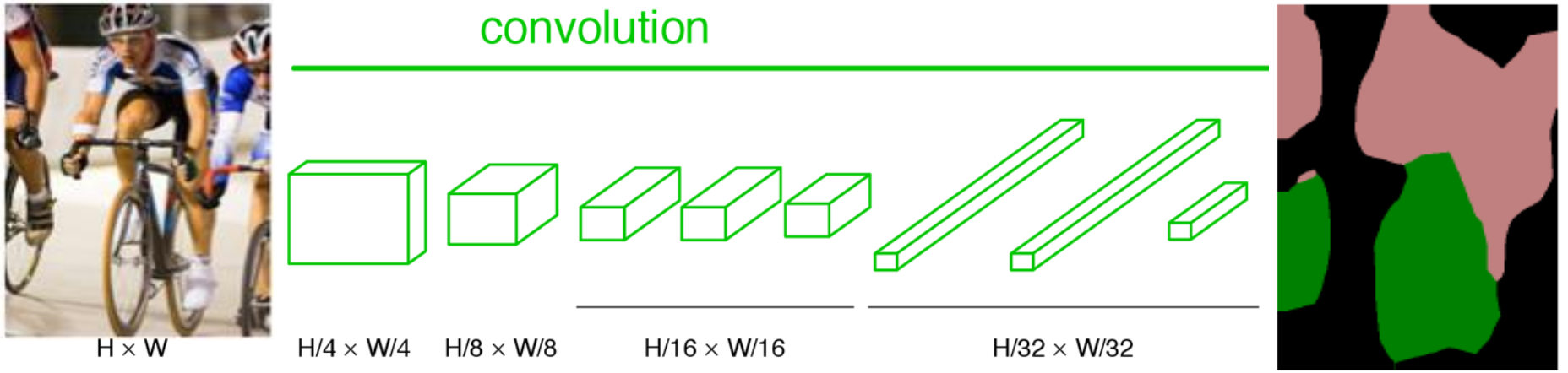
Credit: Shelhamer, Long

Challenges of Semantic Convolution

- **Reduced feature resolution**

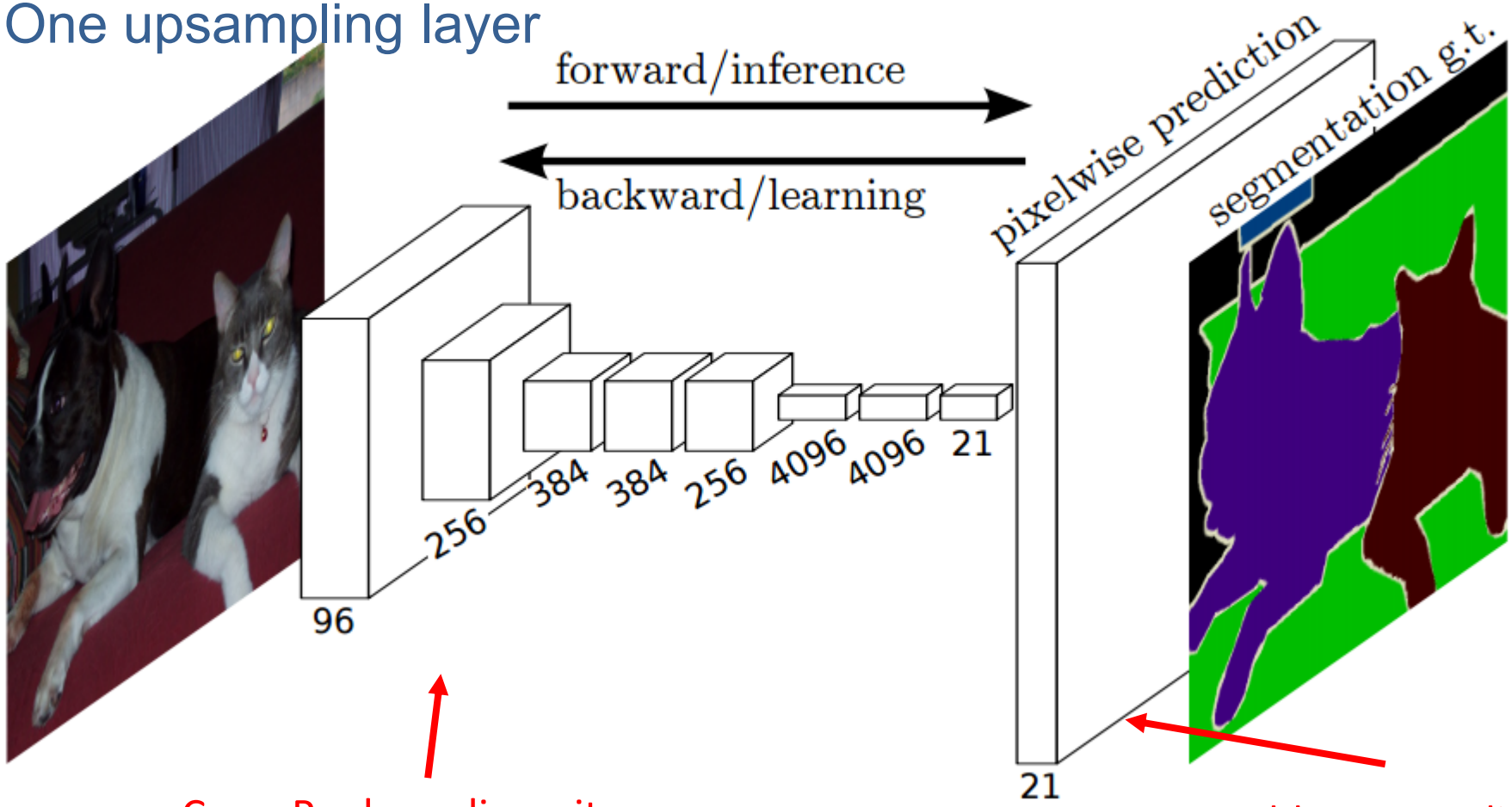


- **Coarse prediction output**



From classification to semantic segmentation

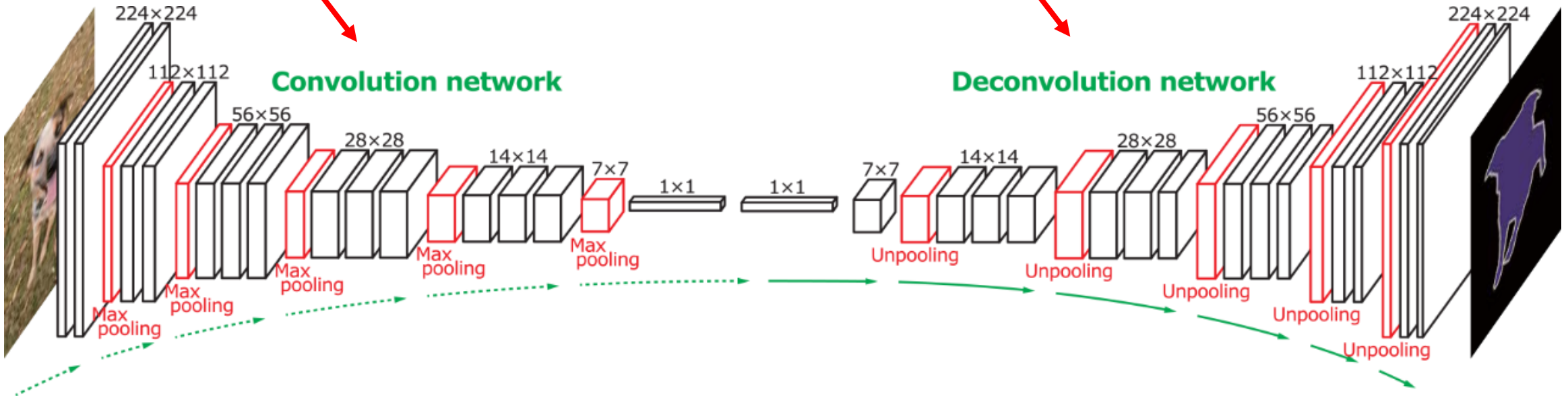
- fully convolutional, end to end, pixel-to-pixel network
  - One upsampling layer



Source: cs231n

More than one upsampling layer

- DeconvNet:
  - VGG-16 (conv+Relu+MaxPool) + mirrored VGG (Unpooling+'deconv'+Relu)



Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

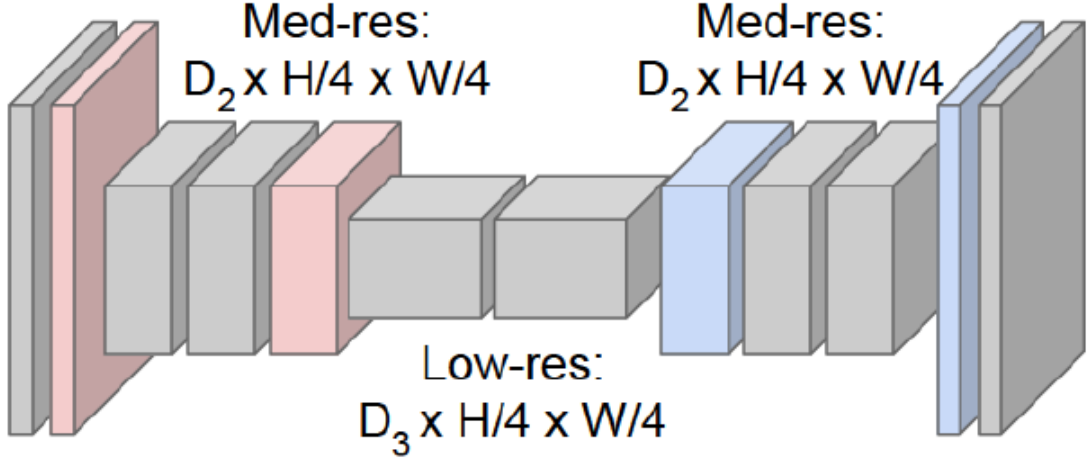
Semantic Segmentation Idea: Fully Convolutional Network

Downsampling:
Pooling, dilated Convolution (strided)

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Upsampling:
???

Input:
3 x H x W



High-res:
D_1 x H/2 x W/2

High-res:
D_1 x H/2 x W/2

Predictions:
H x W

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Source: cs231n

In-Network Upsampling: “Unpooling”

Nearest Neighbor

1	2
3	4

Input: 2 x 2

→

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

Output: 4 x 4

“Bed of Nails”

1	2
3	4

Input: 2 x 2

→

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

Output: 4 x 4

For average pooling

Source: cs231n

In-Network Upsampling: “Max Unpooling”

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

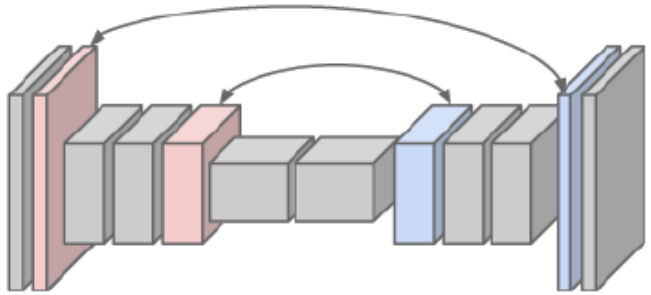
Input: 2 x 2

→

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

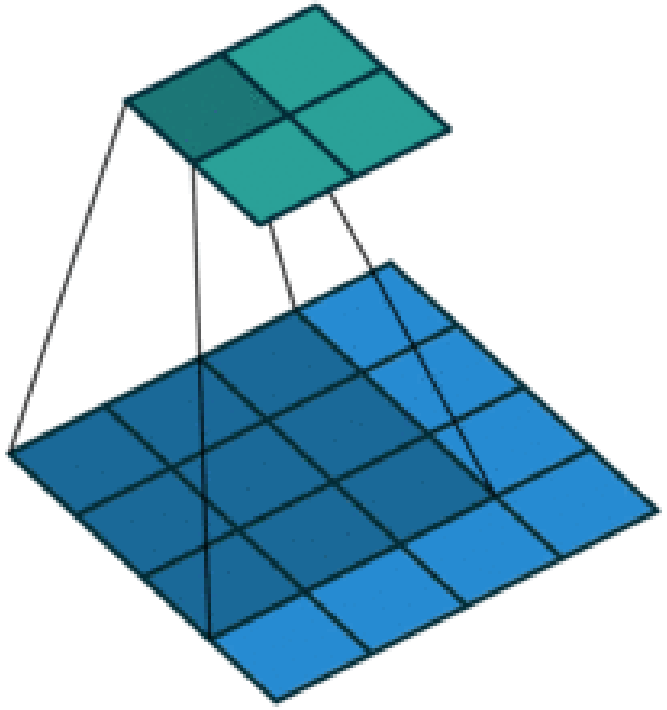
Output: 4 x 4

Corresponding pairs of downsampling and upsampling layers

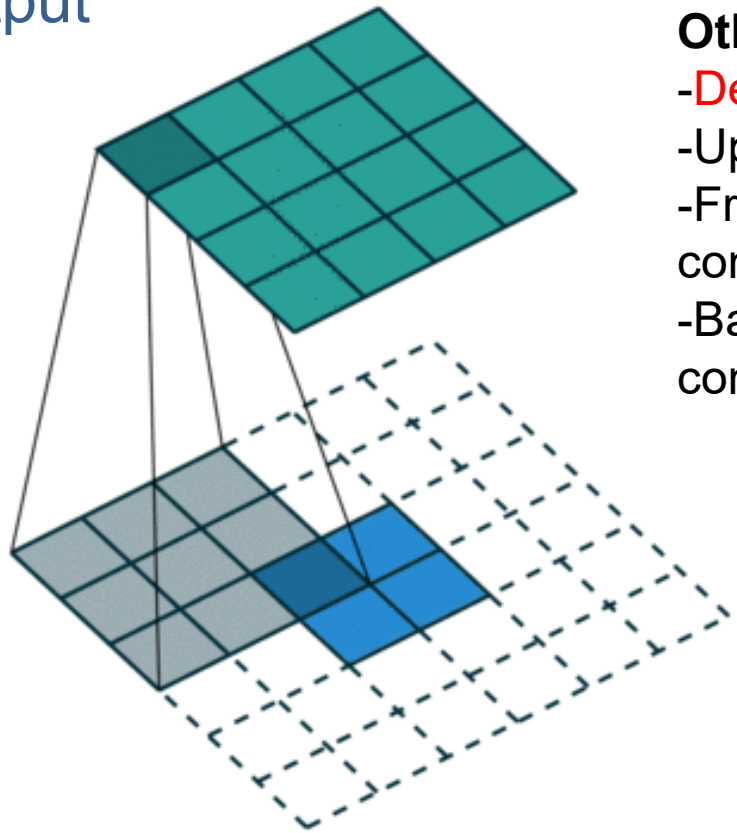


Learnable Unsampling: Transpose Convolution

- Convolution
  - 4x4 input x 3x3 filter
  - =>2x2 output



- Transpose convolution (Note. Not Inverse Conv)
  - 2x2 input x 3x3 filters (transposed)
  - => 4x4 output

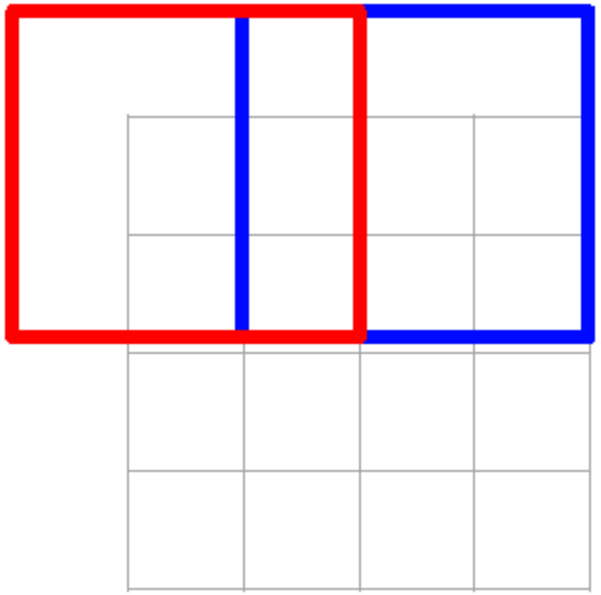


Other names:
-Deconvolution (bad)
-Upconvolution
-Fractionally strided convolution
-Backward strided convolution

Source: http://deeplearning.net/software/theano_versions/dev/tutorial/conv_arithmetic.html#transposed-convolution-arithmetic

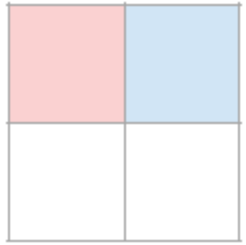
Learnable Unsampling: Transpose Convolution

Recall: Normal 3 x 3 convolution, stride 2 pad 1



Input: 4 x 4

Dot product between filter and input



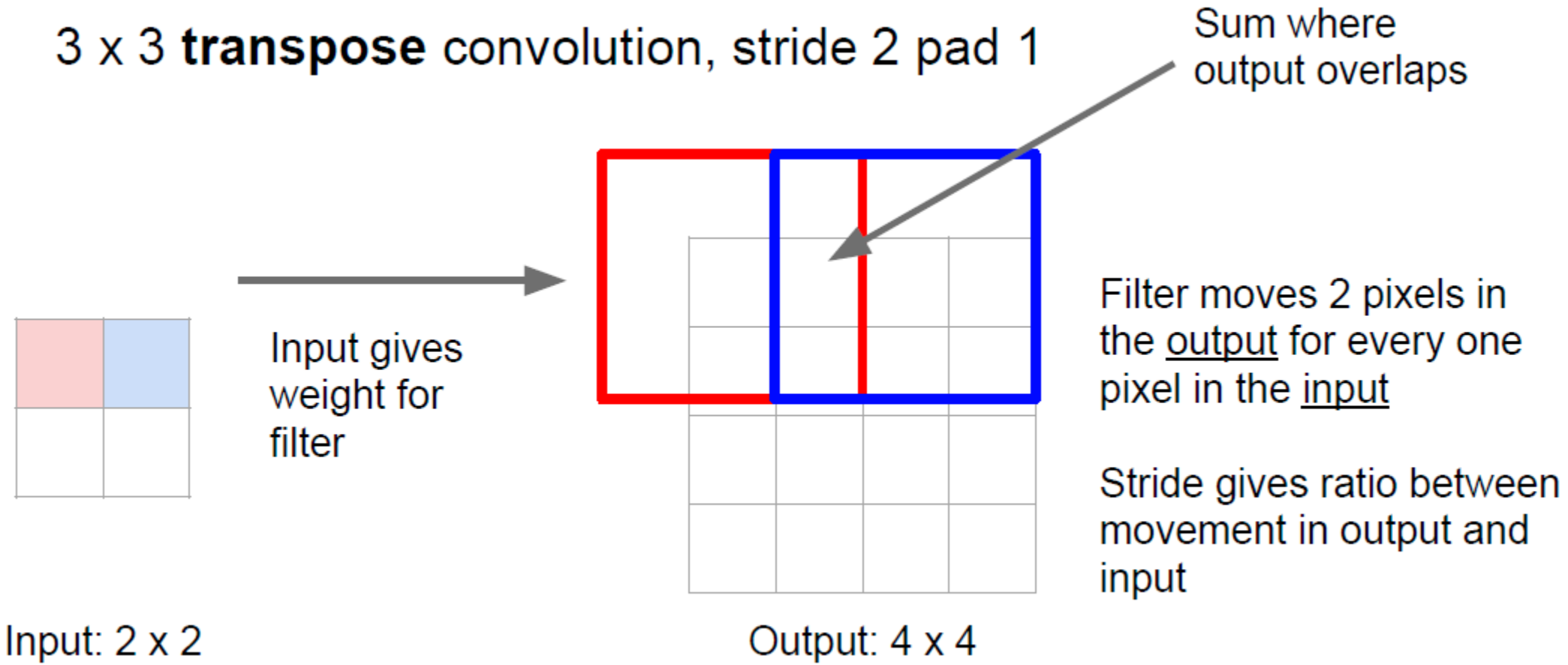
Output: 2 x 2

Filter moves 2 pixels in the input for every one pixel in the output

Stride gives ratio between movement in input and output

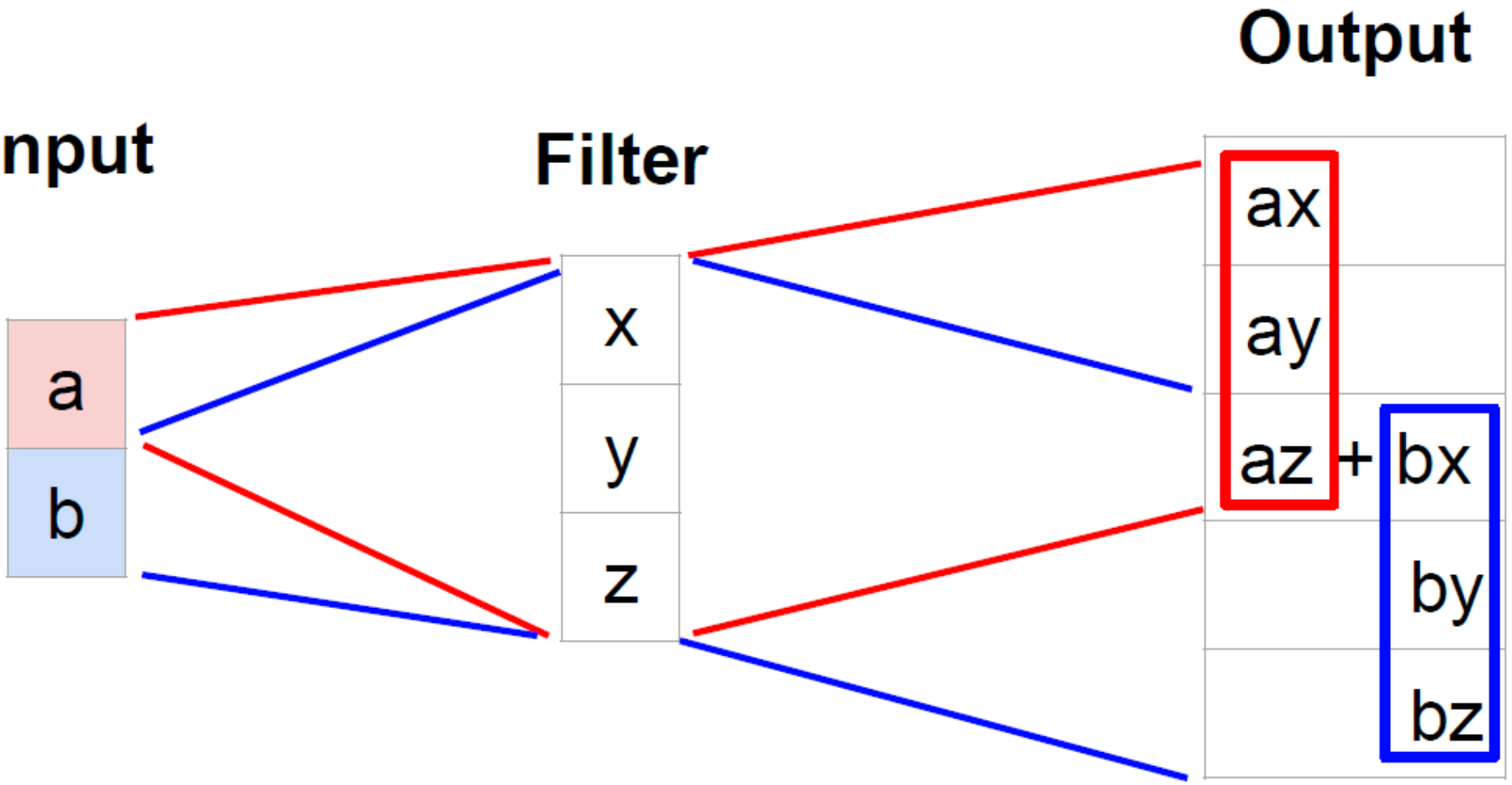
Source: cs231n

Learnable Unsampling: Transpose Convolution



Source: cs231n

Learnable Unsampling: Transpose Convolution 1-D 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

Source: cs231n

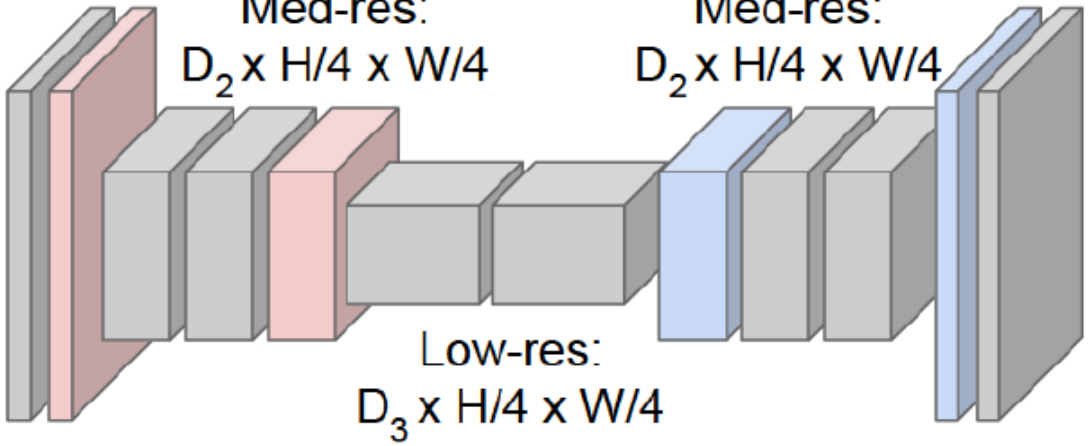
Semantic Segmentation Idea: Fully Convolutional Network

Downsampling:
Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Upsampling:
Unpooling or strided transpose convolution

Input:
3 x H x W



High-res:
D_1 x H/2 x W/2

High-res:
D_1 x H/2 x W/2

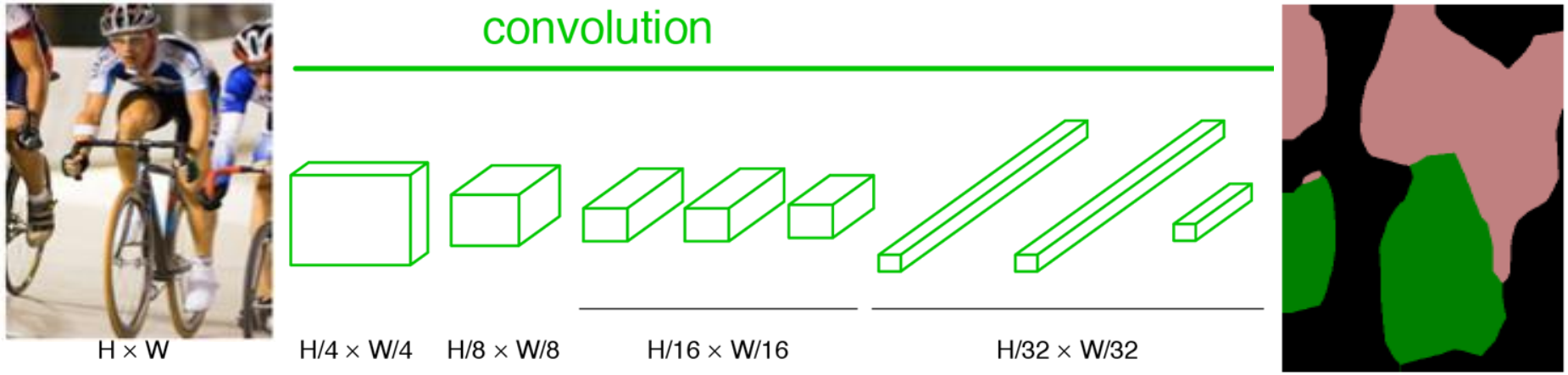
Predictions:
H x W

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

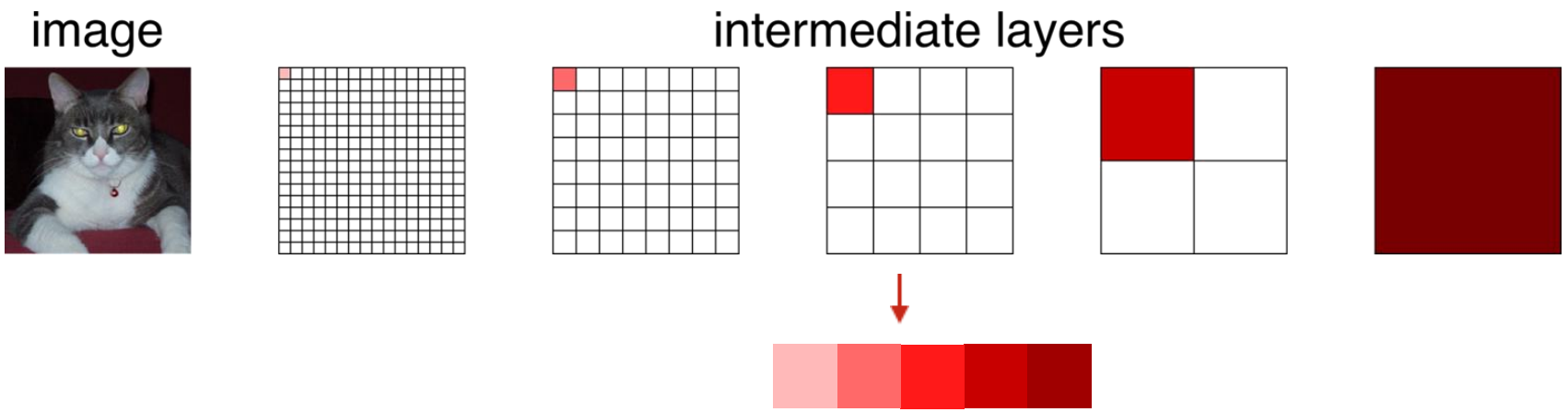
Source: cs231n

How to detect multi-scale object? Fully Convolutional Network Case

Problem: coarse output



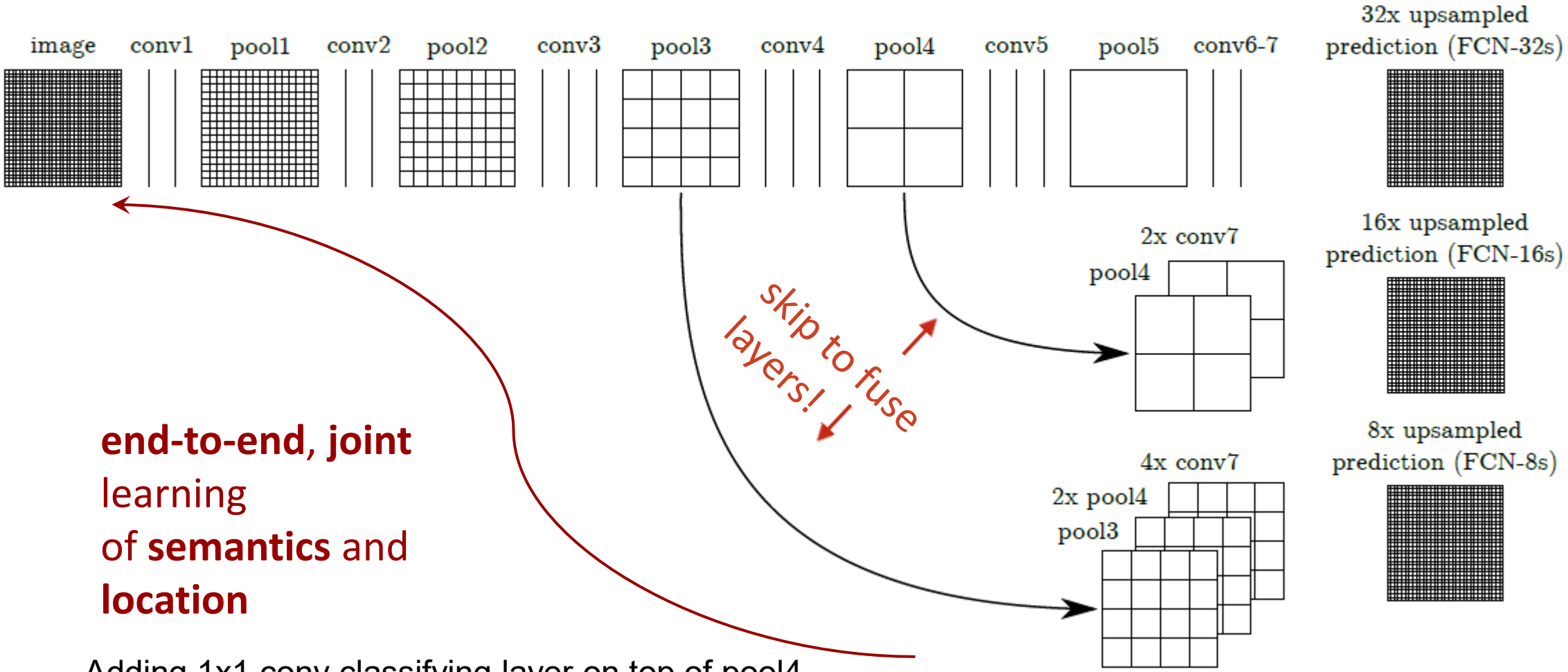
- combine *where* (local, shallow) with *what* (global, deep)



fuse features into **deep jet** (cf. Hariharan et al. CVPR15 "hypercolum

Credit: Shelhamer, Long

Fine details: skip connections



end-to-end, joint learning of **semantics** and **location**

Adding 1x1 conv classifying layer on top of pool4,
Then upsample x2 (init to bilinear and then learned)
conv7 prediction, sum both, and upsample x16 for output

Credit: Shelhamer, Long

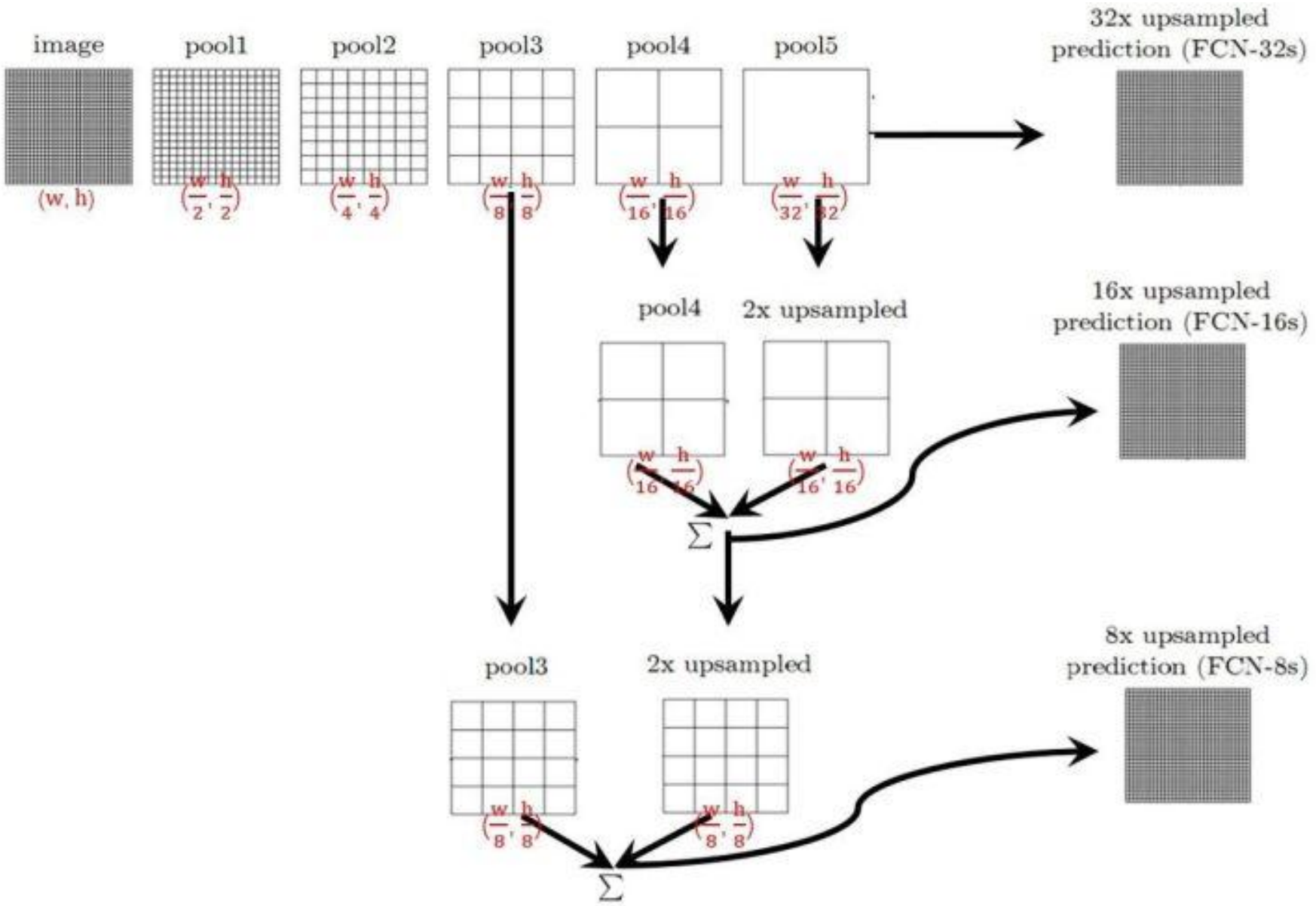
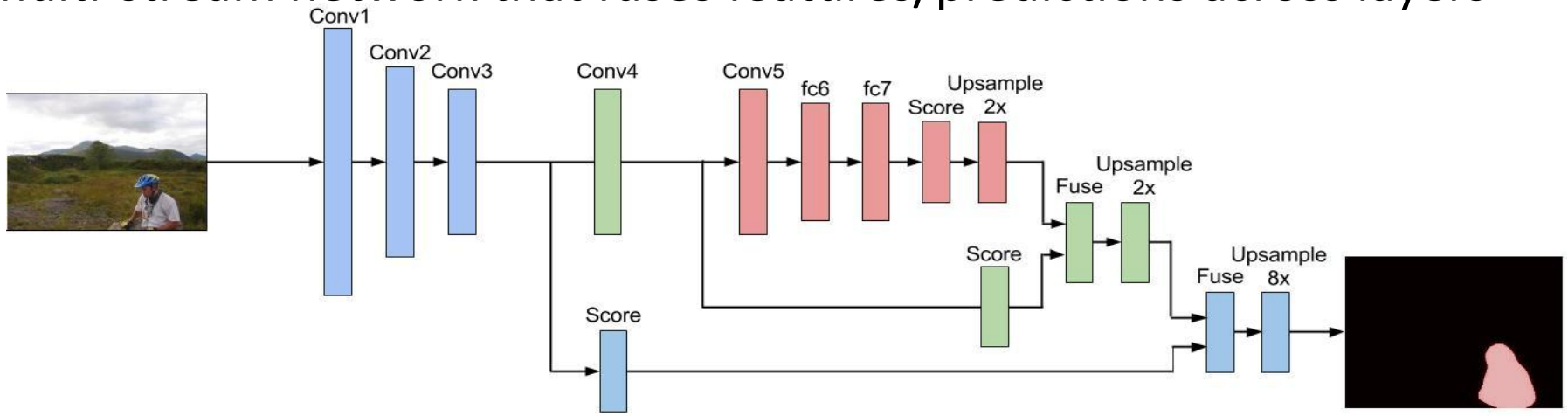


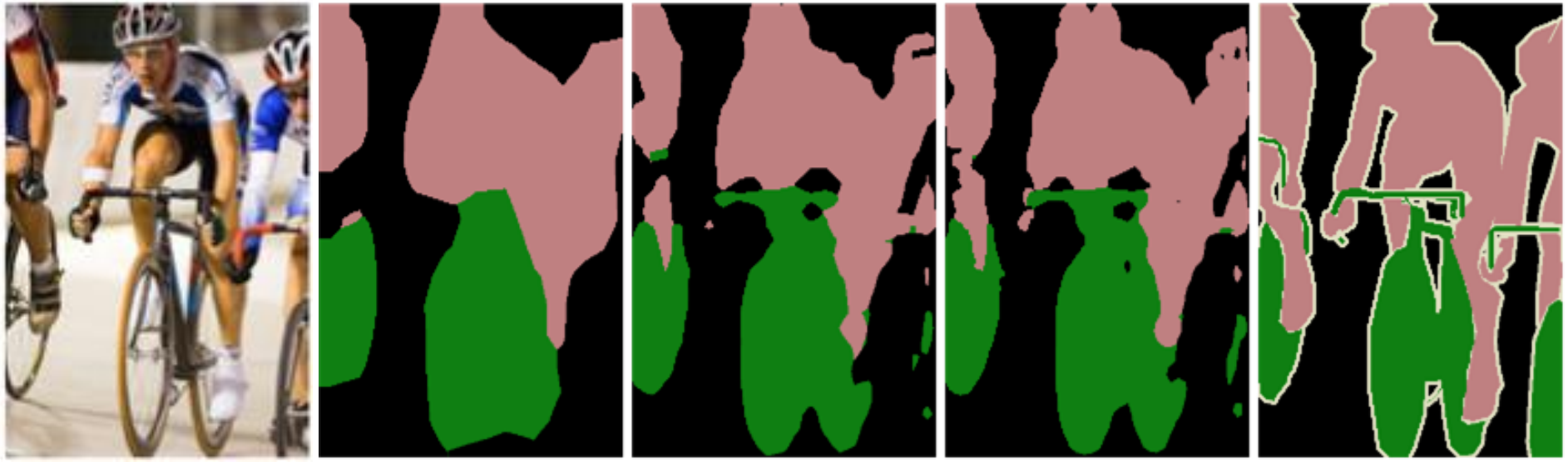
image
pool1
pool2
pool3
pool4
pool5
(w, h)
32x upsampled
prediction (FCN-32s)
pool4
2x upsampled
16x upsampled
prediction (FCN-16s)
pool3
2x upsampled
8x upsampled
prediction (FCN-8s)

Skip connections

- A multi-stream network that fuses features/predictions across layers



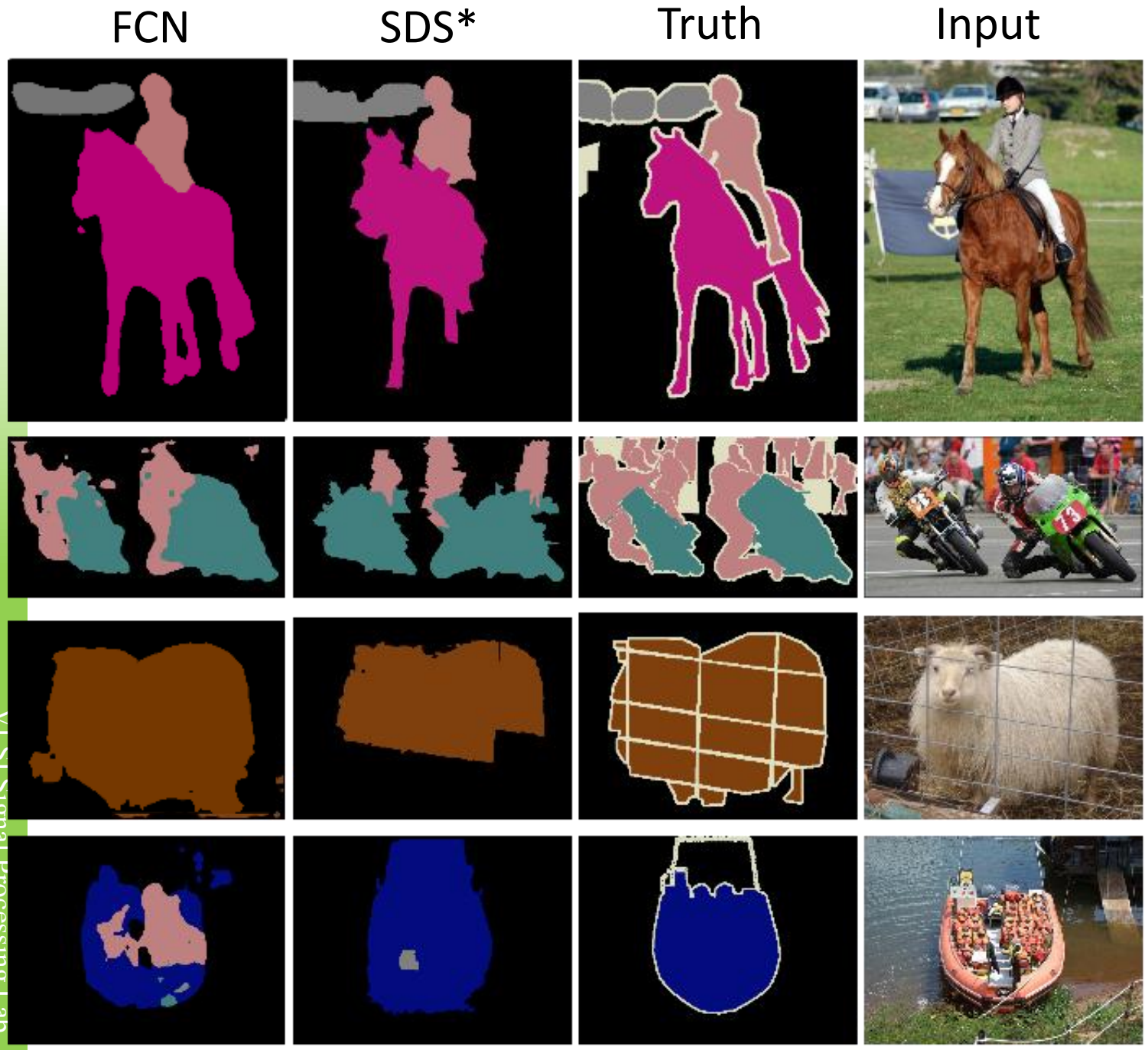
input image | stride 32 | stride 16 | stride 8 | ground truth



no skips | 1 skip | 2 skips

Credit: Shelhamer, Long

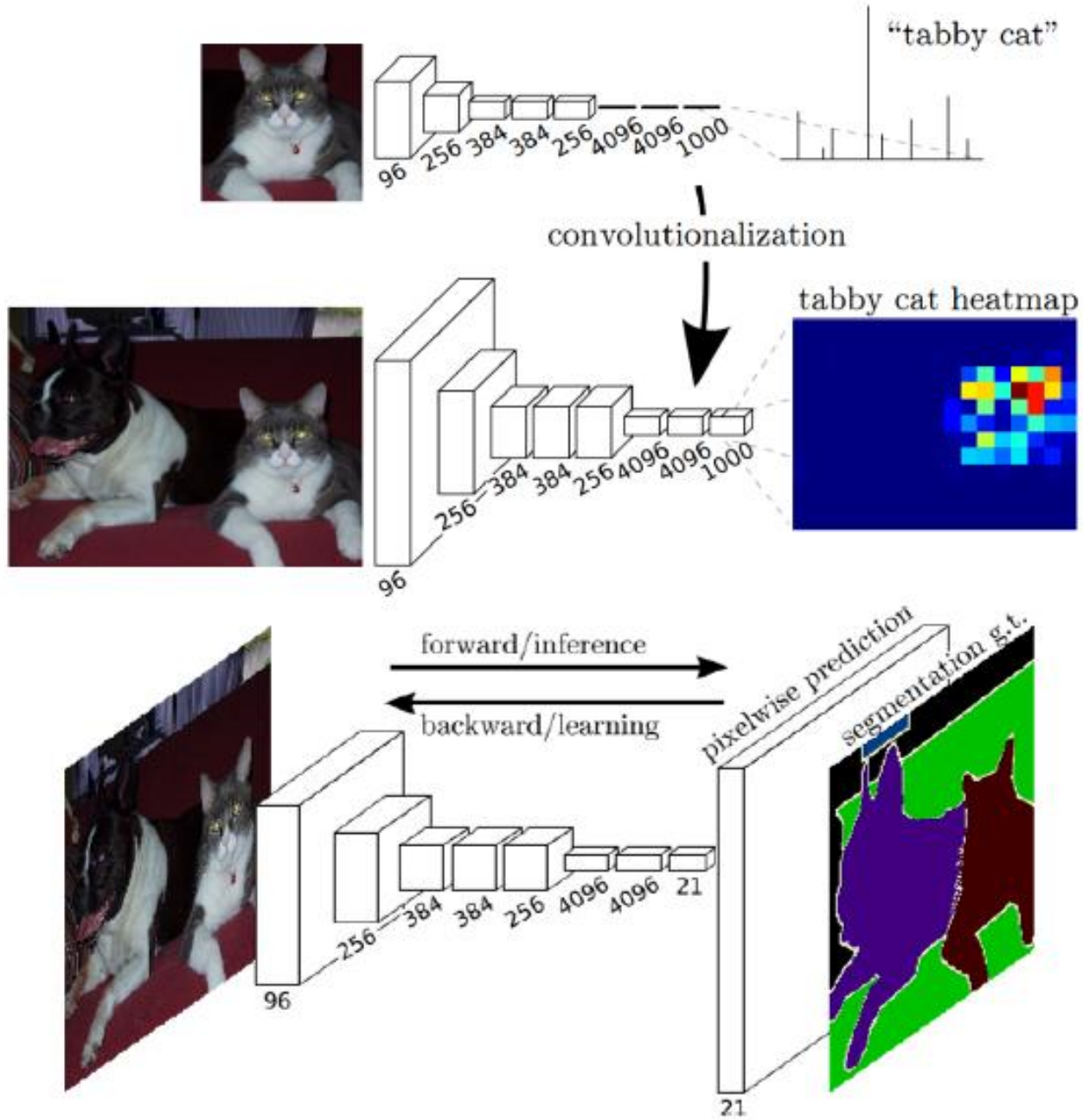
Source: cs231n



Relative to prior state-of-the-art SDS:

- 30% relative improvement for mean IoU
- 286× faster

*Simultaneous Detection and Segmentation Hariharan et al. ECCV14


"tabby cat"
96
256
384
384
256
4096
4096
1000
convolutionalization
tabby cat heatmap
forward/inference
backward/learning
pixelwise prediction
segmentation g.t.
21

SEMANTIC SEGMENTATION
STATE OF THE ART METHODS

Architecture of Semantic Segmentation

- ImageNet pretrained
- Dilated convolution

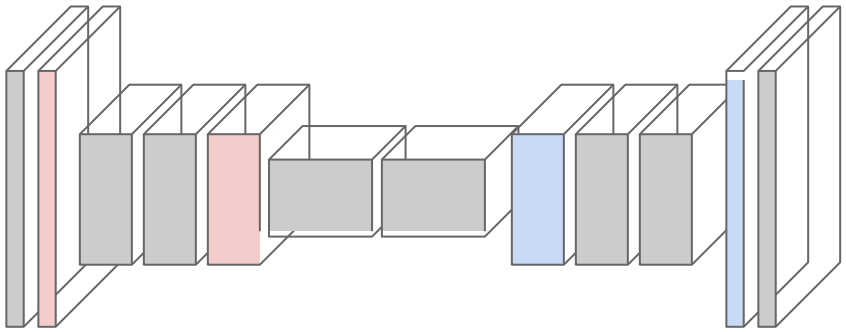
- Multi-scale feature aggregation

- Bilinear interpolation
- Single/multiple upsampling layer (learned)

- Conditional random field (optional)

Common Semantic Segmentation Styles

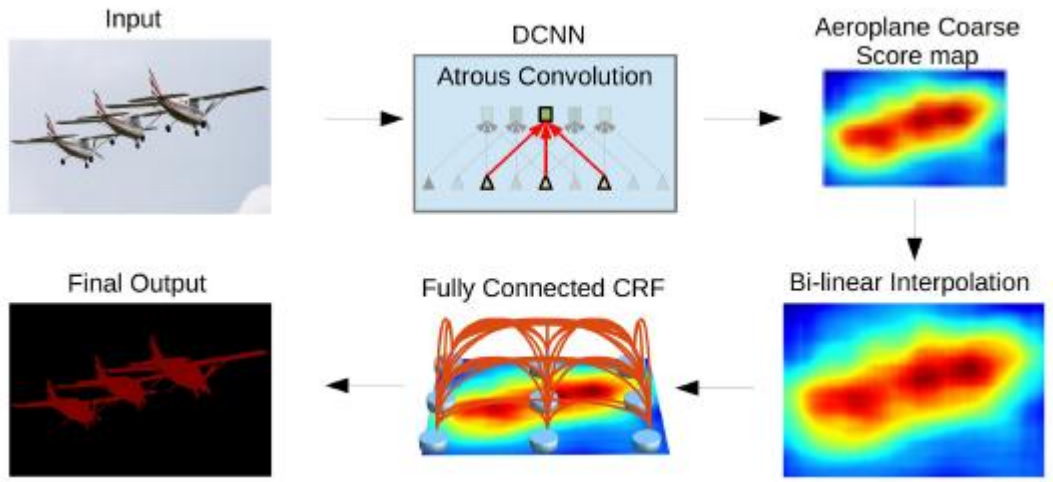
- Downsampling path: extracts coarse features (Semantic)
  - As a transfer learning work from classification to semantic segmentation
    - Base classification network: AlexNet, VGG, ResNet….
  - Dilated convolution (Yu, 2016)
- Upsampling path: recovers input image resolution (Resolution)
  - Upsampled version of above downsampling path
  - One layer
    - Skip connections: recovers detailed information
  - Multiple layers
- Multi-scale detection
- Post-processing (optional): refines predictions (CRF)
  - CRF: condition random field



Source: cs231n

Architectures for Semantic Segmentation

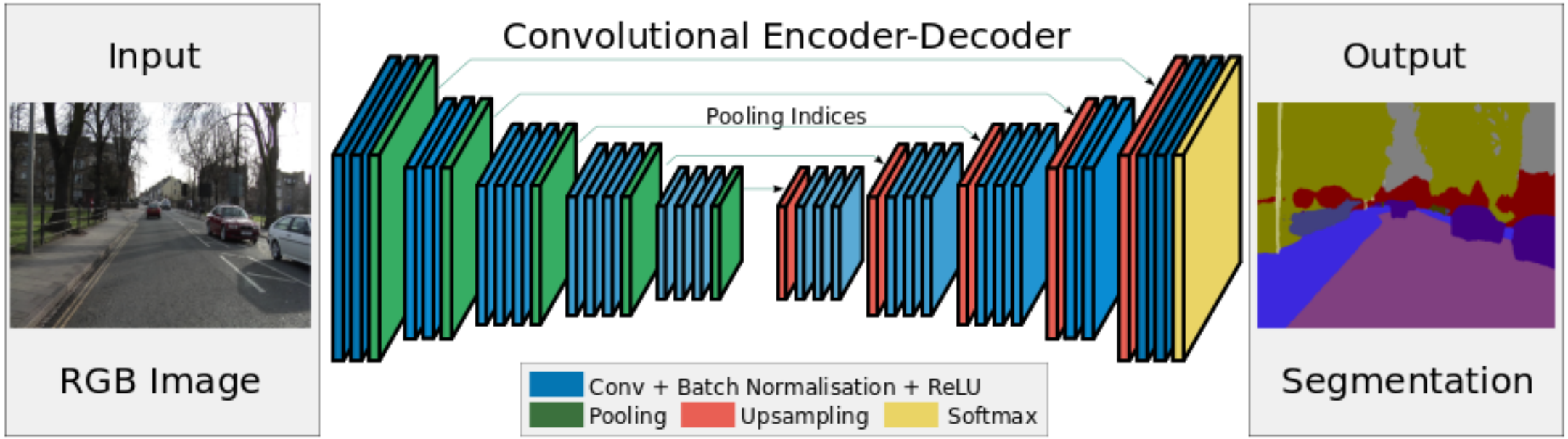
- Encoder-decoder
  - SegNet
  - U-Net (Ronnemberger, 2015)
    - Fully Convolutional DenseNets (Jégou, 2016)
- Feature enhanced
  - DeepLab (V1, V2, V3, V3+): ‘atrous’ convolutions + spatial pyramid + CRF (Chen, ICLR 2015)
  - Dilated convolutions (Yu, 2016)
- Object detection
  - Mask R-CNN



Source: cs231n

SegNet

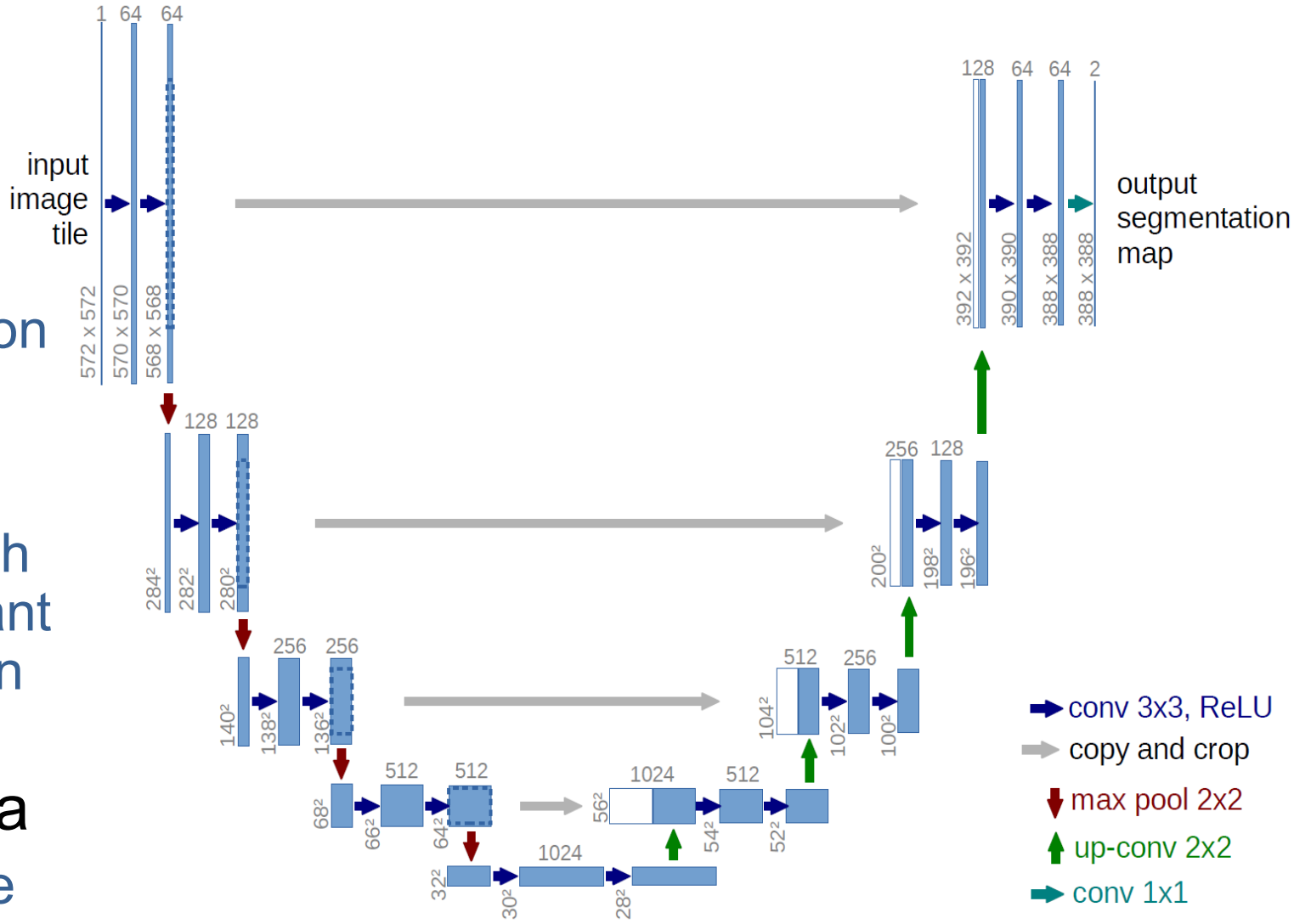
- Unpooling without learning
  - Get index from encoder side
  - Pros. help keep the high-frequency information intact
  - Cons: miss neighbouring information when unpooling from low-resolution



Source: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, 2015

U-Net

- symmetric contracting /expanding path
  - capture context
  - enables precise localization
- skip concatenation connections
  - allows the decoder at each stage to learn back relevant features that are lost when pooled in the encoder
- Learn from very little data
  - Popular for medical image segmentation



特徵融合方式：拼接

Winner of
CAD Caries challenge ISBI 2015
Cell tracking challenge ISBI 2015

Ronneberger et al, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, arXiv 2015

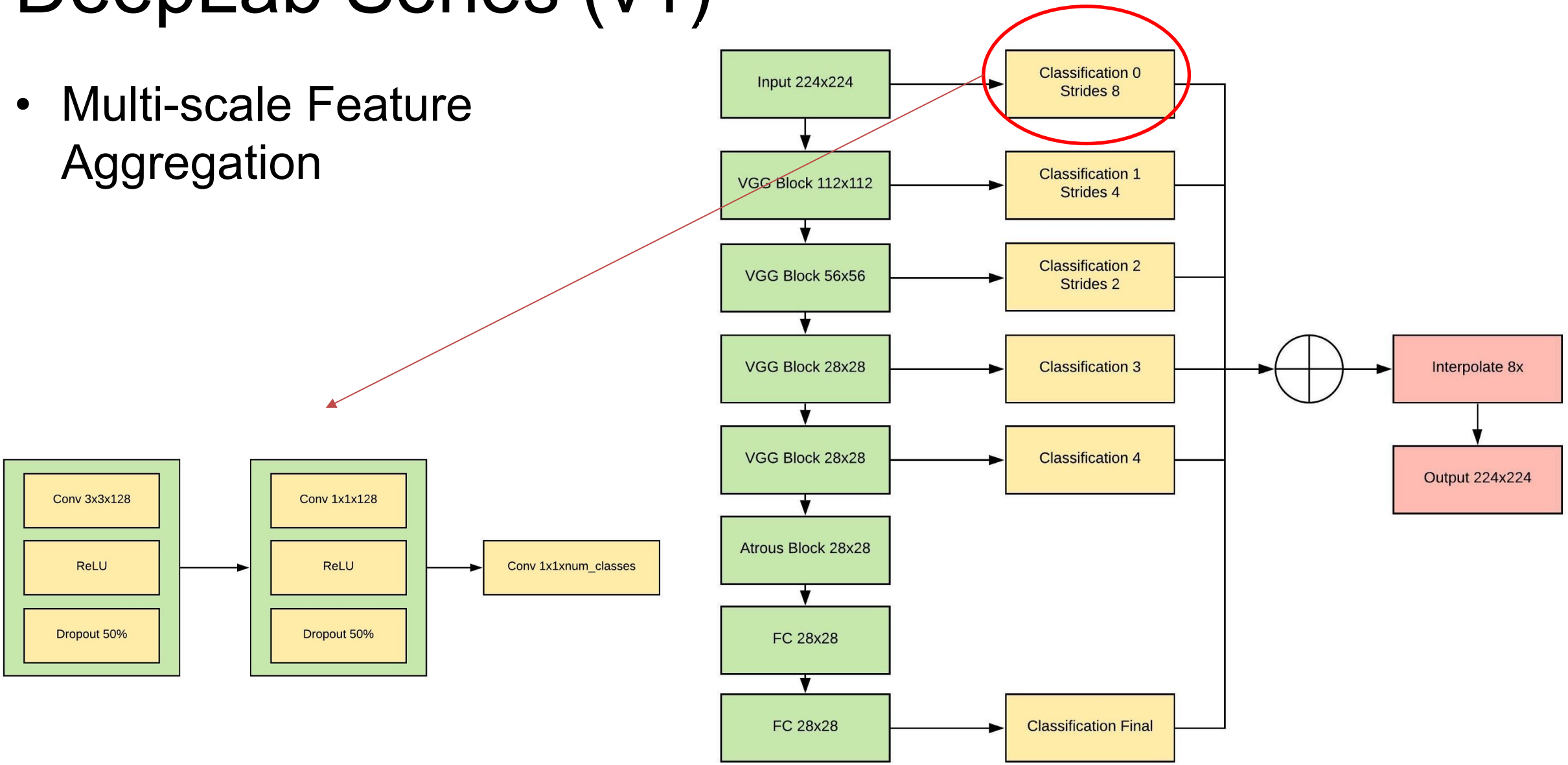
Deeplab Series v1

Three challenges

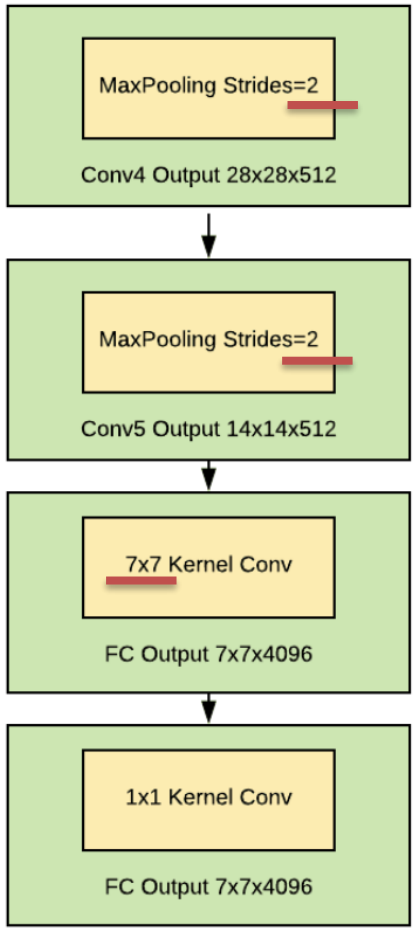
- Reduced feature resolution due to pooling and downsampling (strided convolution)
  - Atrous convolution (dilated convolution)
- existence of objects at multiple scales
  - Inspired by SPP (spatial pyramid pooling), atrous spatial pyramid pooling
- reduced localization accuracy due to DCNN invariance
  - Fully connected CRF

DeepLab Series (v1)

- Multi-scale Feature Aggregation

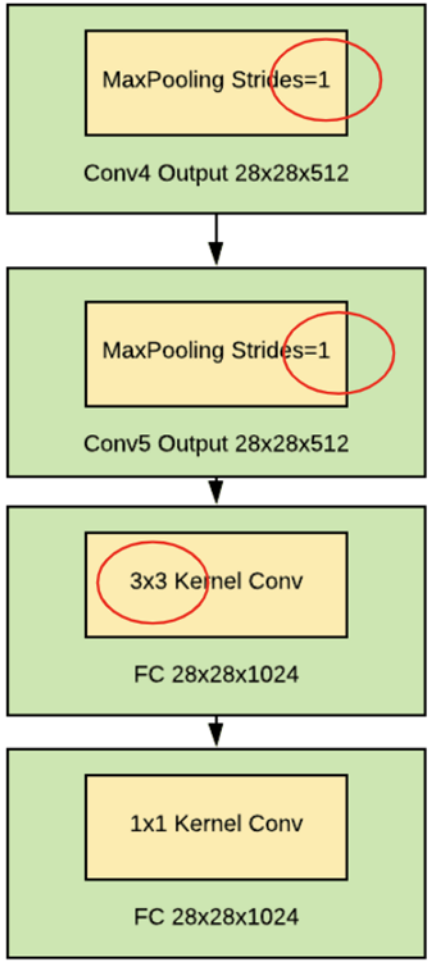


https://yanjia.li/witnessing-the-progression-in-semantic-segmentation-deeplab-series-from-v1-to-v3/



原始VGG輸出7x7太小，不利segmentation

FCN



uses atrous convolution

uses atrous convolution

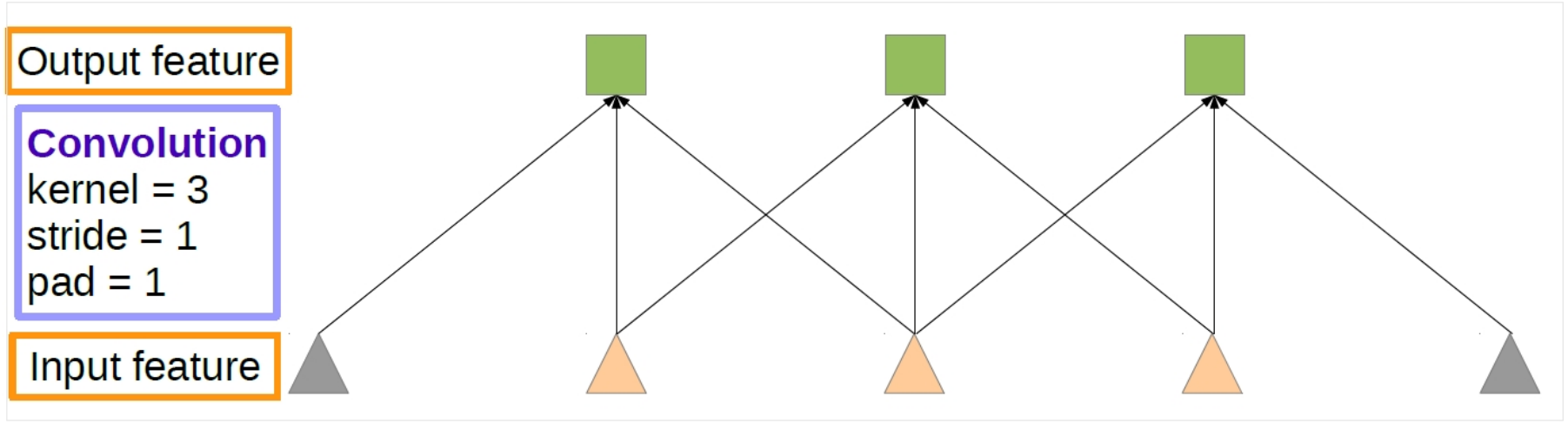
28x28 feature map

DeepLabv1

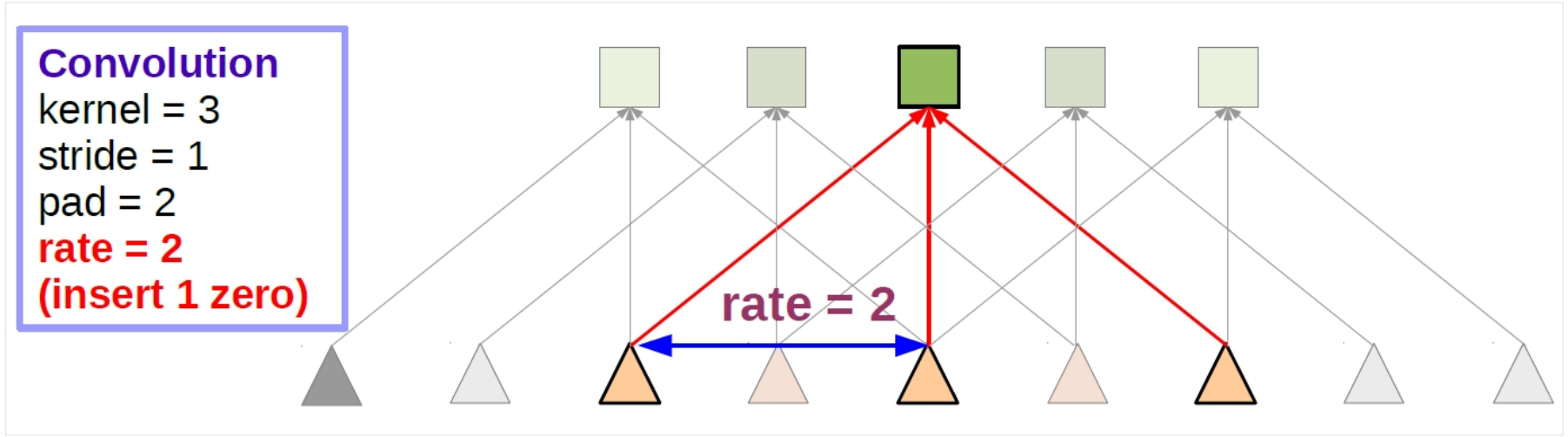
將最後一個VGG塊的MaxPooling層的步幅設置為1，這樣即使對於最後幾層，特徵圖也可以保持在28x28
確保了我們可以預測更大範圍的特徵，但同時又引入了另一個問題：現在的計算成本更高，並且還丟失了從7x7特徵圖中提取的一些全局信號=>conv4, conv5 uses atrous convolution

https://yanjia.li/witnessing-the-progression-in-semantic-segmentation-deeplab-series-from-v1-to-v3/

Astrous Convolution



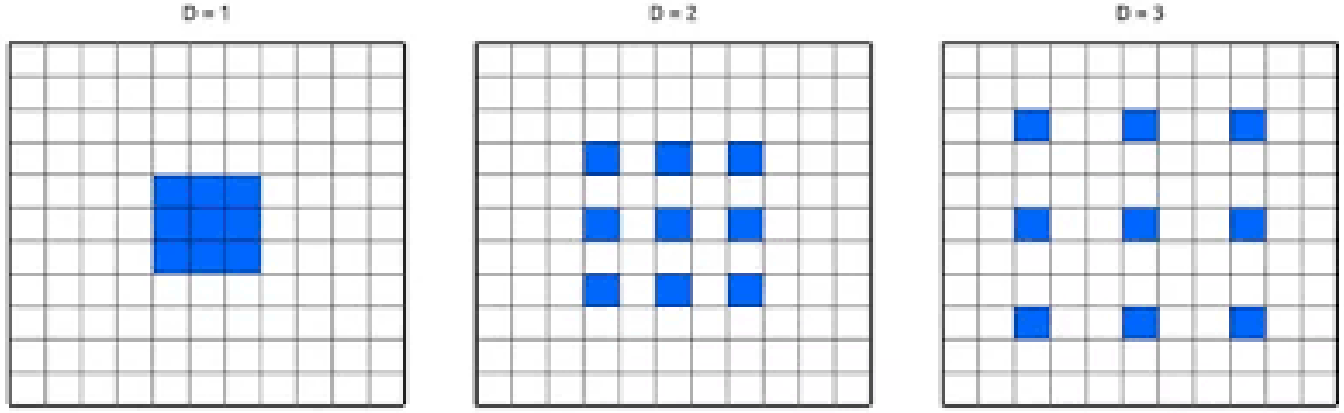
(a) Sparse feature extraction

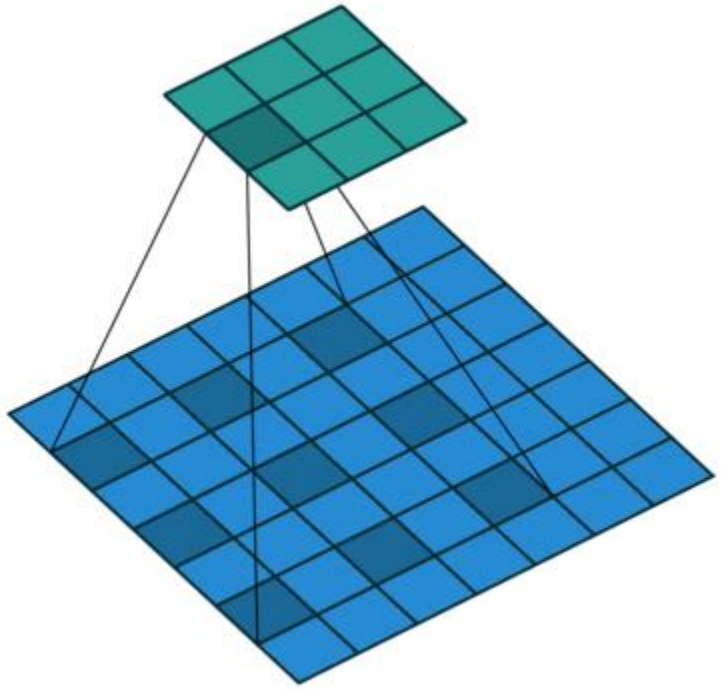


(b) Dense feature extraction

Deeplab v1: atrous convolution (dilated convolution)

- Problems with pooling for downsampling
  - Loss position information
  - Keep position information while expand receptive filed exponentially => dilated convolution
    - Gridding effect for the same scaling factor due to aliasing





在不降低空間維度的前提下增大了相應的receiptive field，計算量不變

Deeplab v1: atrous convolution (dilated convolution)

- Dilated convolution

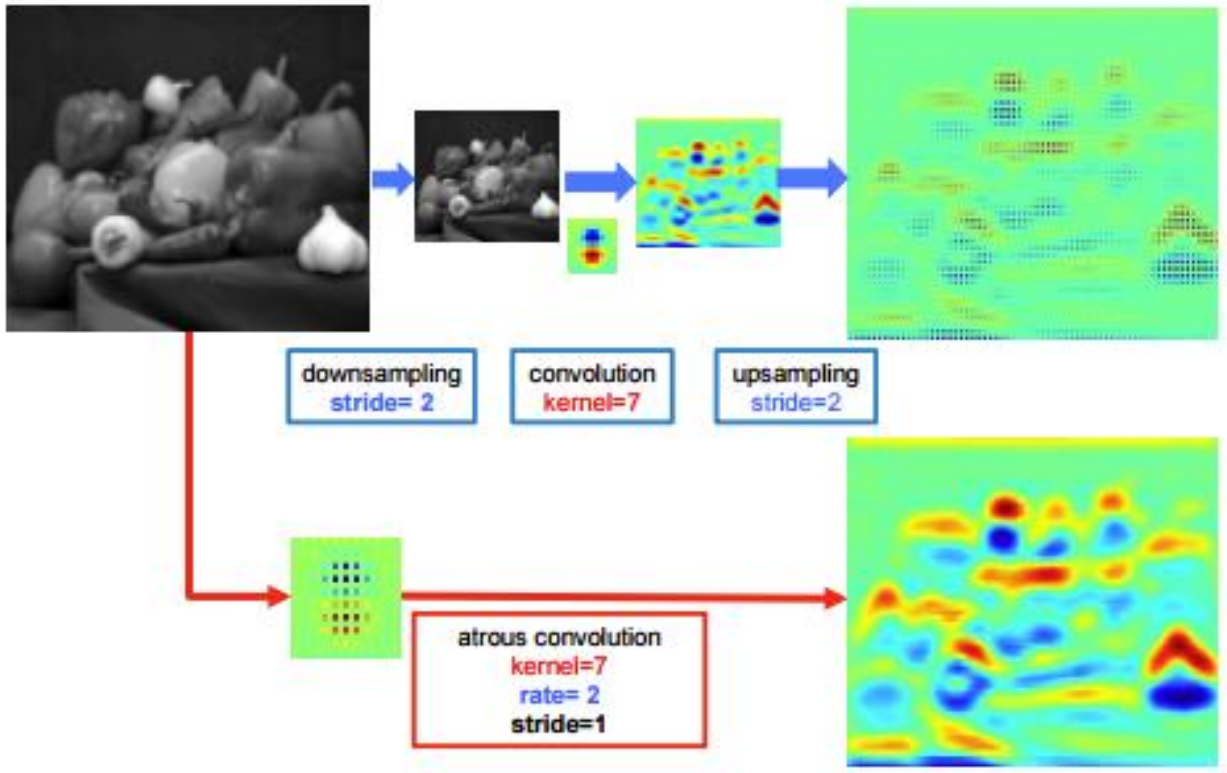


Fig. 3: Illustration of atrous convolution in 2-D. Top row: sparse feature extraction with standard convolution on a low resolution input feature map. Bottom row: Dense feature extraction with atrous convolution with rate $r = 2$,

Deeplab v1: Conditional Random Field

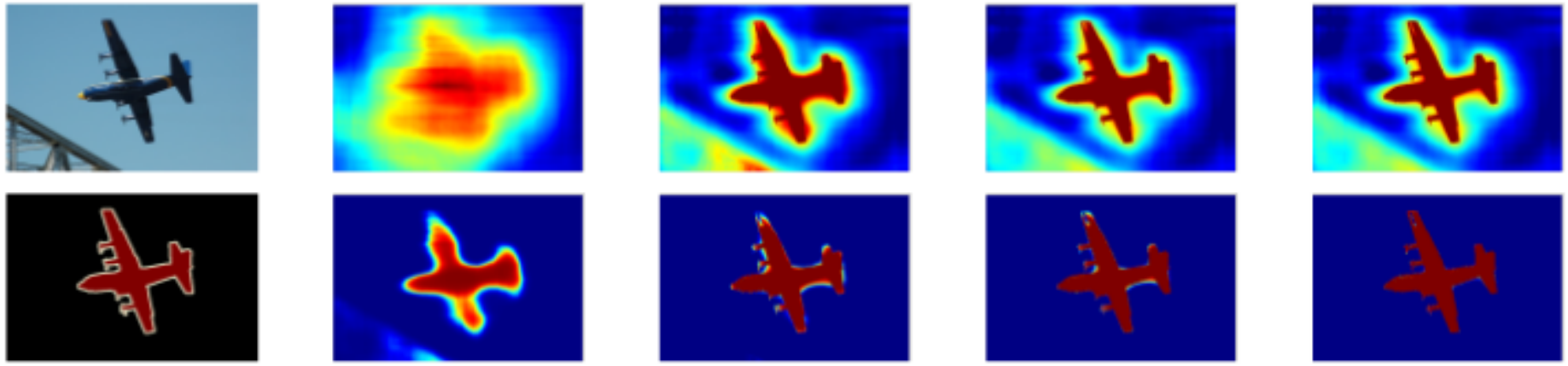


Fig. 5: Score map (input before softmax function) and belief map (output of softmax function) for Aeroplane. We show the score (1st row) and belief (2nd row) maps after each mean field iteration. The output of last DCNN layer is used as input to the mean field inference.

Source: 2015, DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs

Deeplab v1

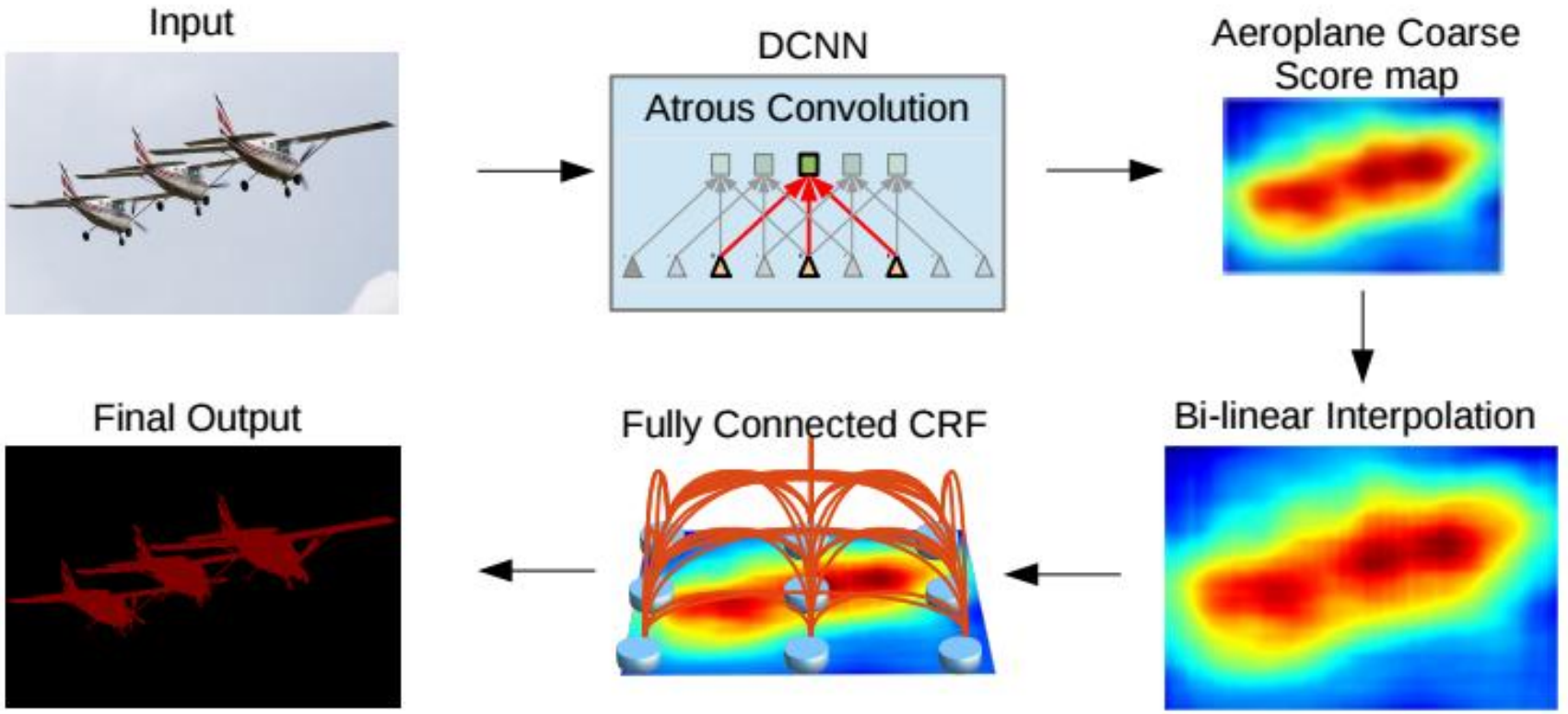


Fig. 1: Model Illustration. A Deep Convolutional Neural Network such as VGG-16 or ResNet-101 is employed in a fully convolutional fashion, using atrous convolution to reduce the degree of signal downsampling (from 32x down 8x). A bilinear interpolation stage enlarges the feature maps to the original image resolution. A fully connected CRF is then applied to refine the segmentation result and better capture the object boundaries.

Source: 2015, DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs

DeepLab v1: Result before/after CRF

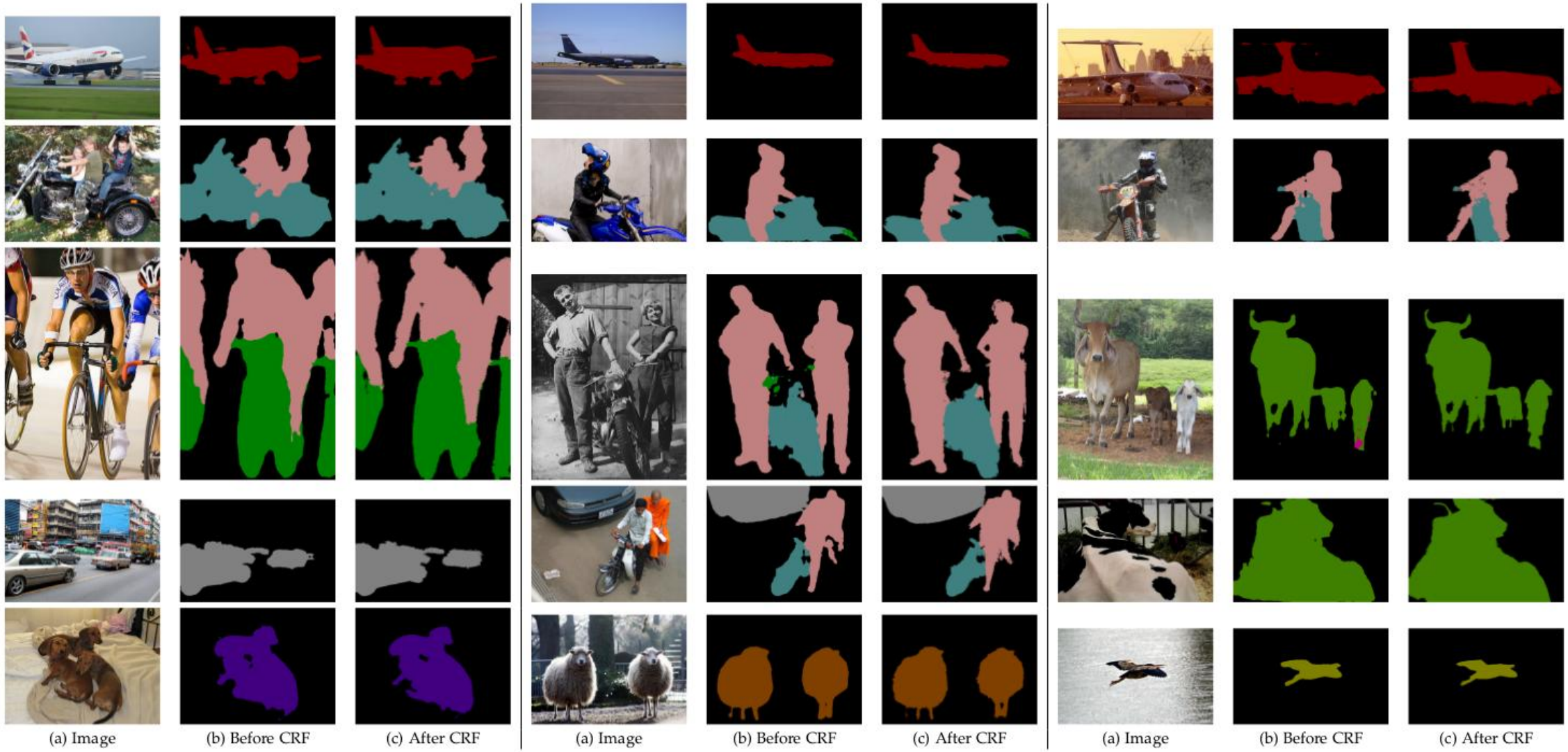


Fig. 6: PASCAL VOC 2012 *val* results. Input image and our DeepLab results before/after CRF.

Deeplab v2: atrous spatial pyramid pooling (ASPP)

如何更好地提取出圖像特徵
使用了多個並行的、不同採樣率的捲積核

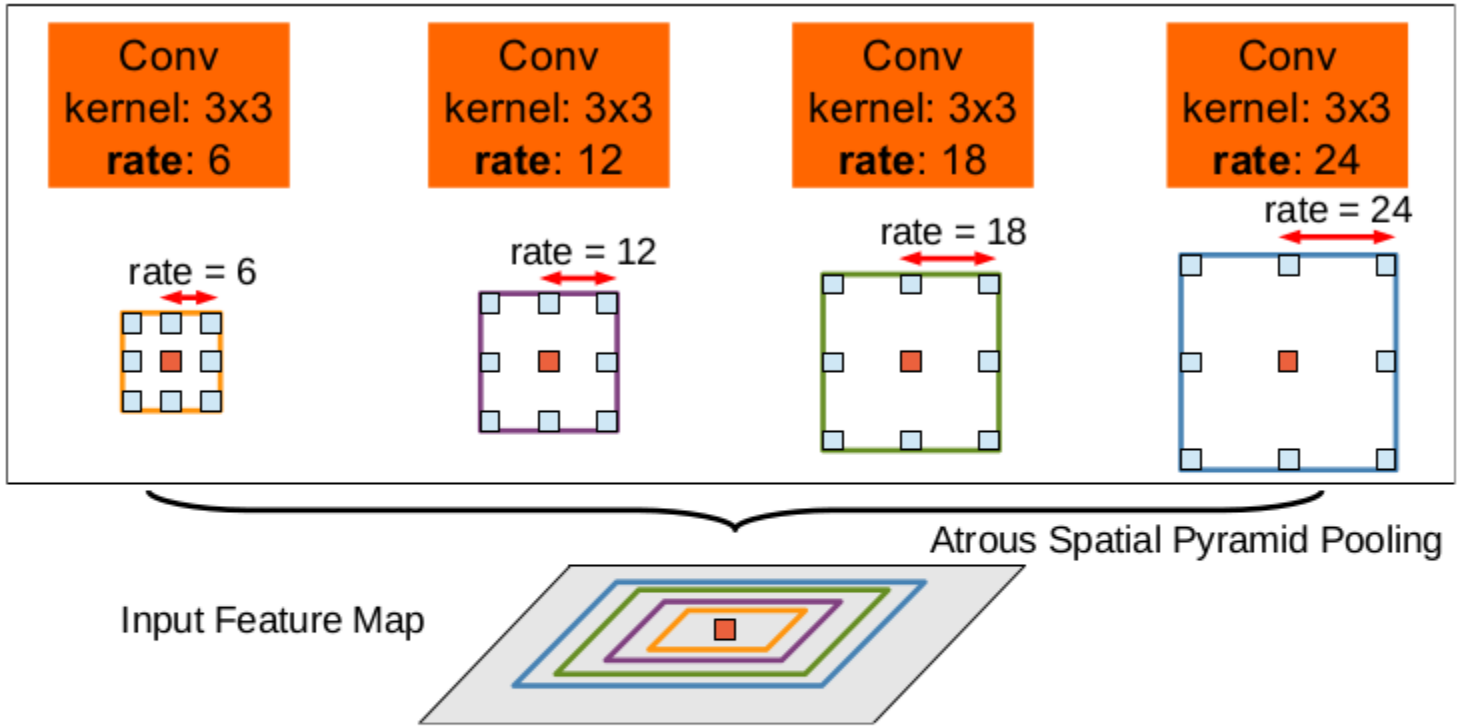


Fig. 4: Atrous Spatial Pyramid Pooling (ASPP). To classify the center pixel (orange), ASPP exploits multi-scale features by employing multiple parallel filters with different rates. The effective Field-Of-Views are shown in different colors.

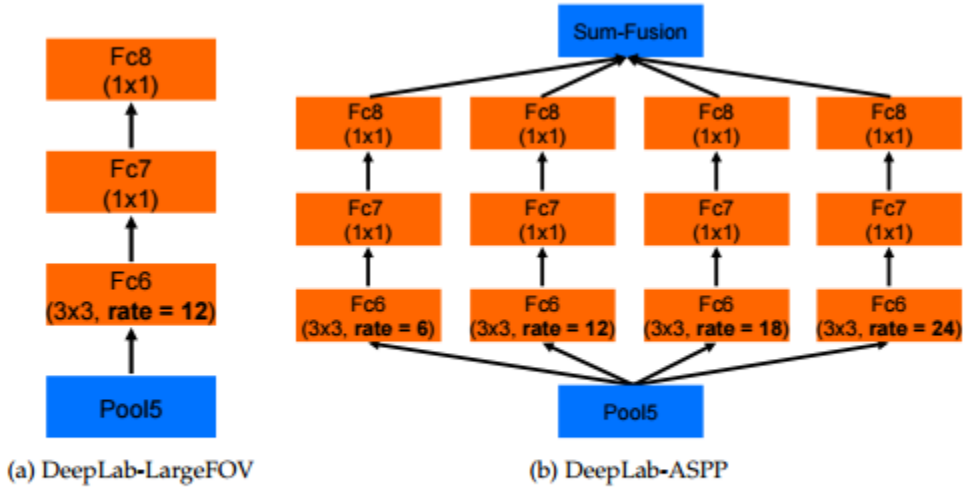
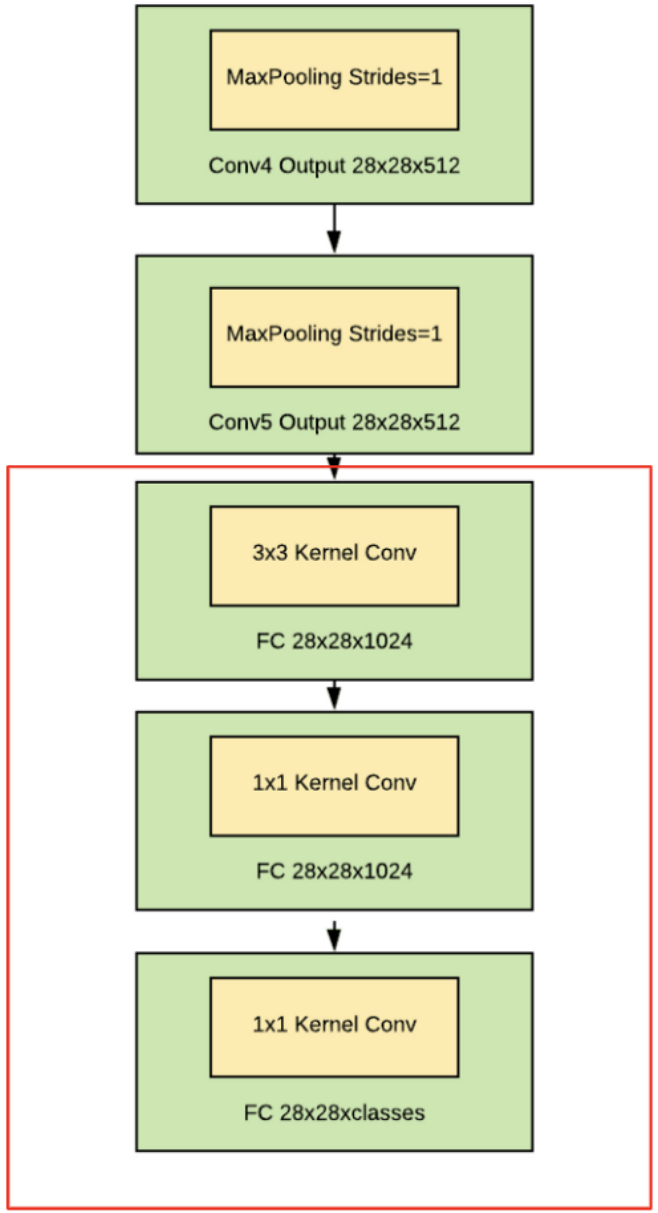
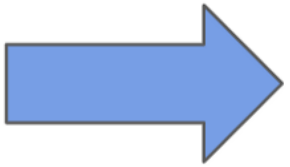


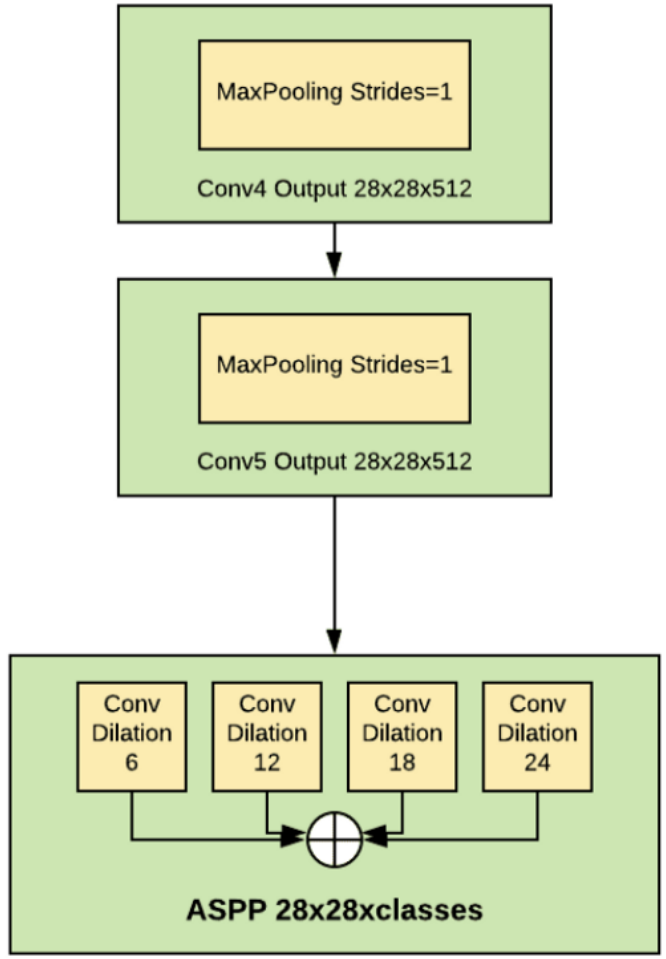
Fig. 7: DeepLab-ASPP employs multiple filters with different rates to capture objects and context at multiple scales.

- ResNet
- Atrous spatial pyramid pooling
- CRF



DeepLabV1



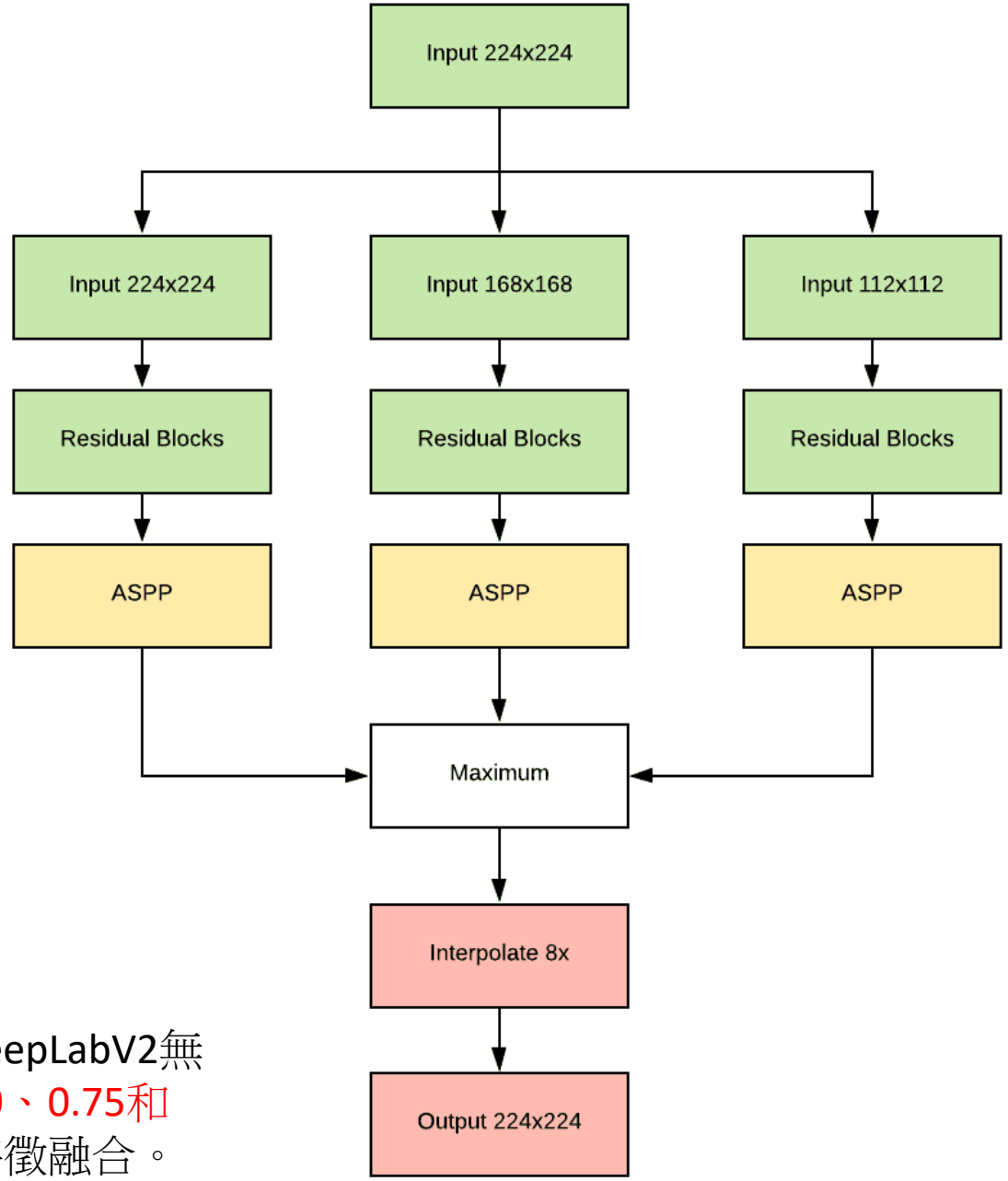


DeepLabV2

https://yanjia.li/witnessing-the-progression-in-semantic-segmentation-deeplab-series-from-v1-to-v3/ 為了更好提取多尺度圖像信息，引入空間金字塔池化

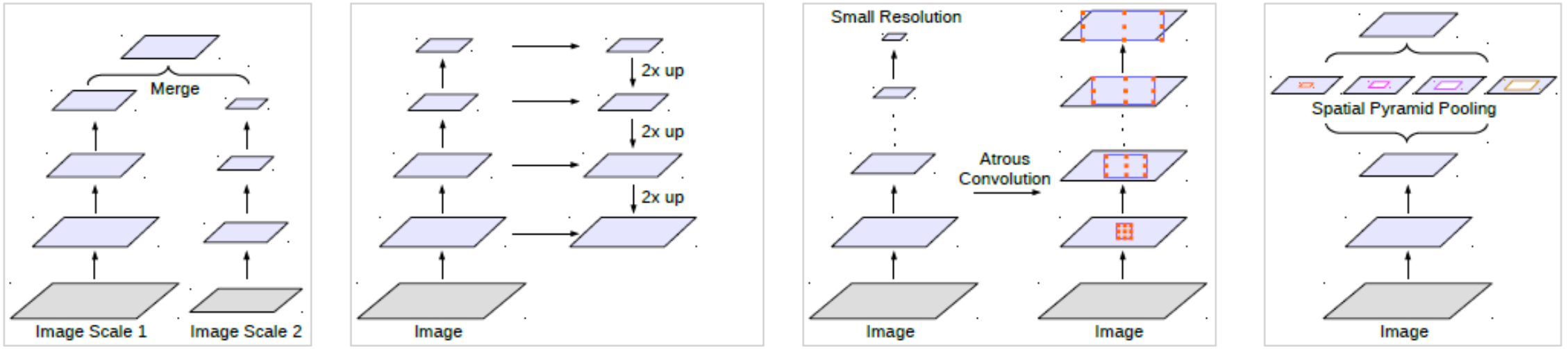
Deeplab v2

- Multi-scale input
- ResNet backbone



V1和V2之間的另一個區別是新的多尺度結構。DeepLabV2無需在不同比例的計算特徵上進行分類，而是在1.0、0.75和0.5的縮小圖像上並行運行三遍，以實現多尺度特徵融合。

DeepLabv3: How to Capture Multi-Scale Context



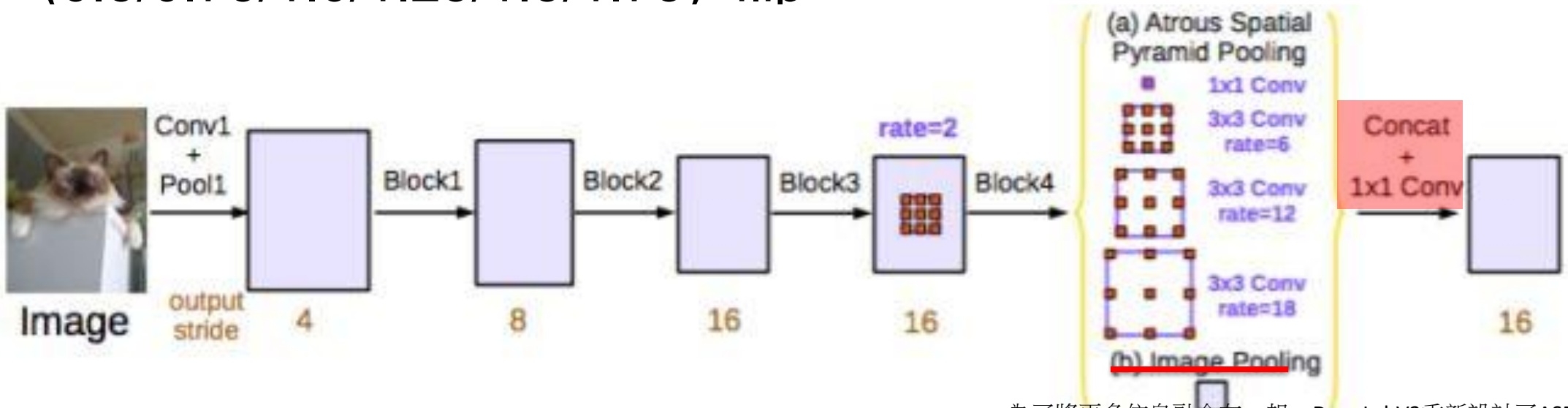
(a) Image Pyramid (b) Encoder-Decoder (c) Deeper w. Atrous Convolution (d) Spatial Pyramid Pooling

Figure 2. Alternative architectures to capture multi-scale context.

v1 v2

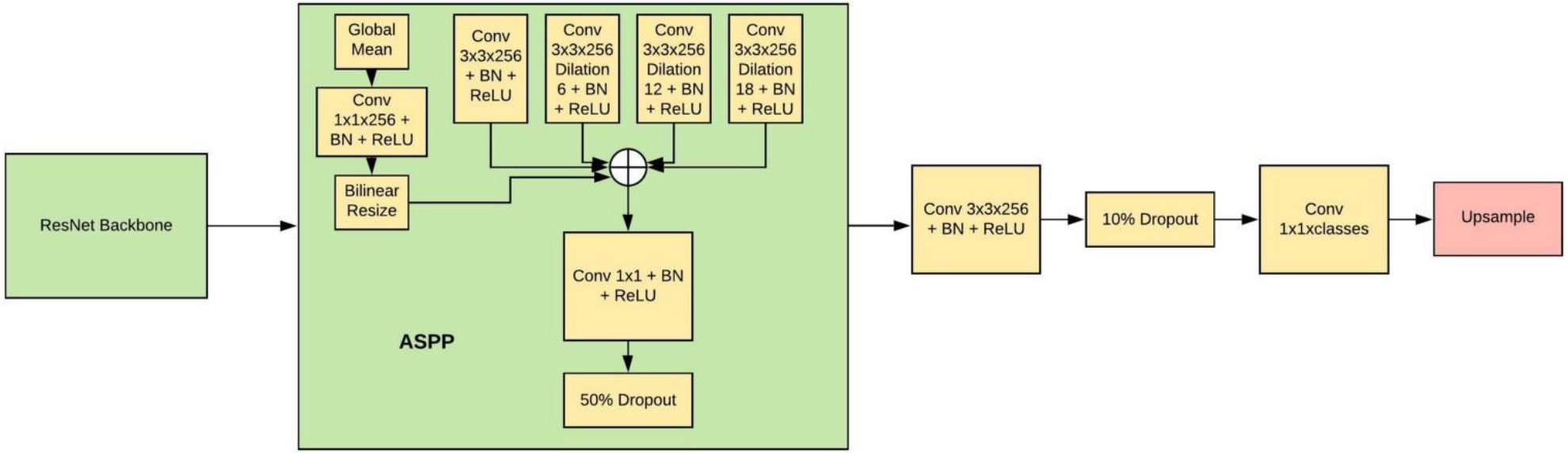
DeepLab v3

- +Batch normalization
- +improved ASPP (+1x1 +pooling +rate (6, 8,12))
- -CRF
- +Test time augmentation: input size (0.5/0.75/1.0/1.25/1.5/1.75)+flip

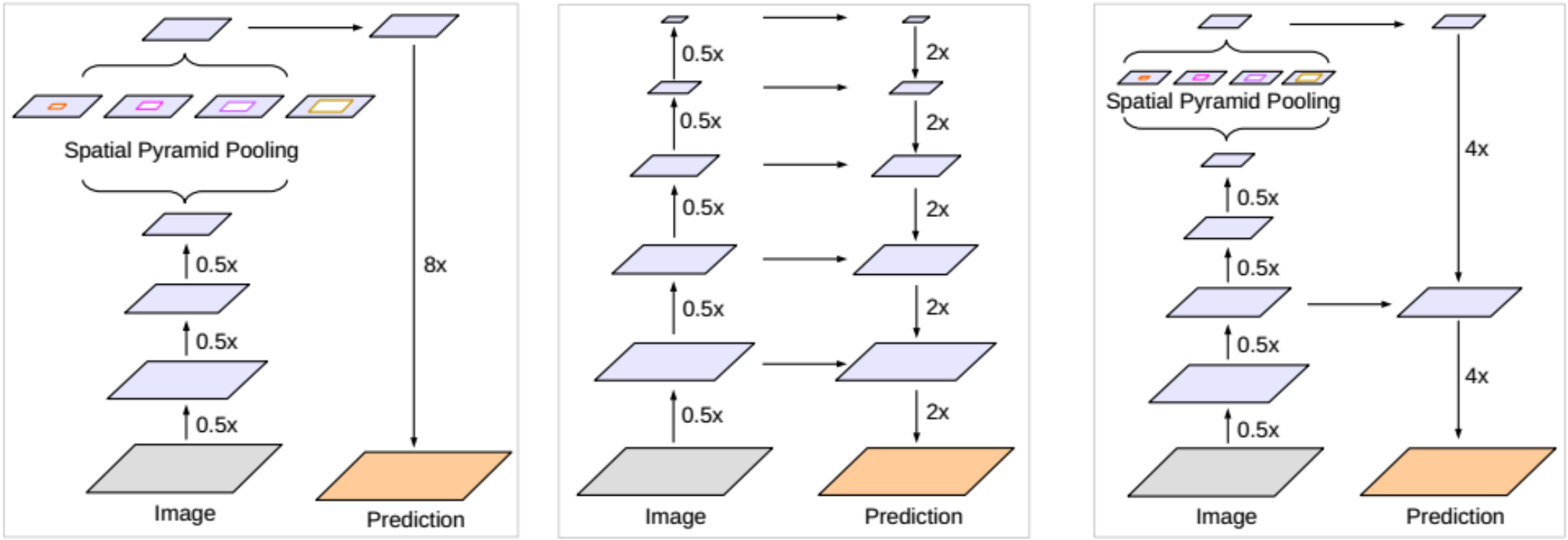


Rethinking Atrous Convolution for Semantic Image Segmentation

為了將更多信息融合在一起，DeepLabV3重新設計了ASPP模塊，使其具有單獨的全局圖像池通道以包含全局特徵，然後將結果特徵向量與ASPP輸入進行1x1卷積級聯，以使用細粒度的細節


ResNet Backbone
Global Mean
Conv 1x1x256 + BN + ReLU
Bilinear Resize
Conv 3x3x256 + BN + ReLU
Conv 3x3x256 Dilation 6 + BN + ReLU
Conv 3x3x256 Dilation 12 + BN + ReLU
Conv 3x3x256 Dilation 18 + BN + ReLU
Conv 1x1 + BN + ReLU
50% Dropout
ASPP
Conv 3x3x256 + BN + ReLU
10% Dropout
Conv 1x1xclasses
Upsample

DeepLab v3+ (=v3 + encoder-decoder)



(a) Spatial Pyramid Pooling (b) Encoder-Decoder (c) Encoder-Decoder with Atrous Conv.

Figure 1. We propose to improve DeepLabv3, which employs the spatial pyramid pooling module (a), with the encoder-decoder structure (b). The proposed model, DeepLabv3+, contains rich semantic information from the encoder module, while the detailed object boundaries are recovered by the simple yet effective decoder module. The encoder module allows us to extract features at an arbitrary resolution by applying atrous convolution.

DeepLab v3+

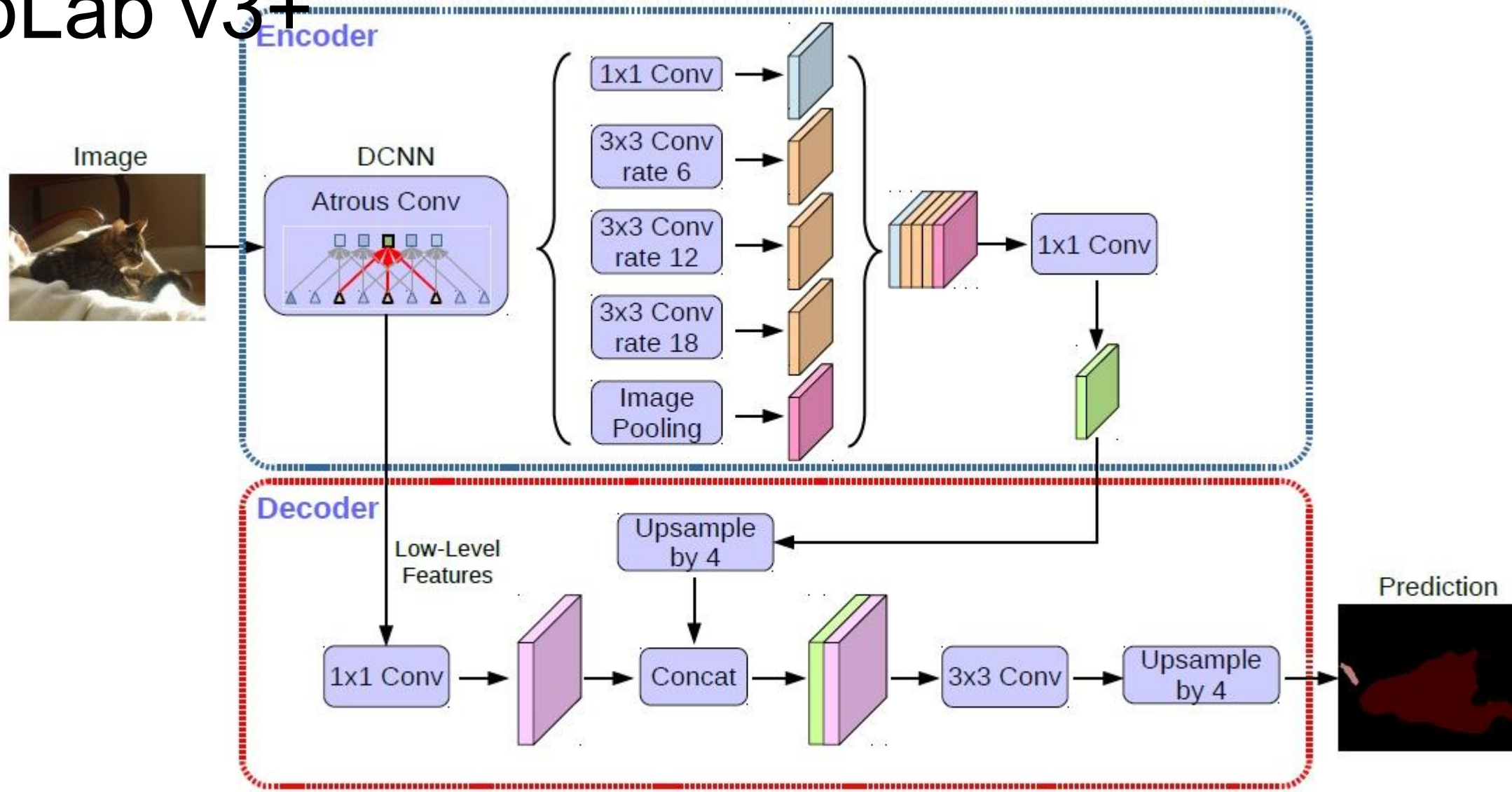


Figure 2. Our proposed DeepLabv3+ extends DeepLabv3 by employing a encoder-decoder structure. The encoder module encodes m scale contextual information by applying atrous convolution at multiple scales, while the simple yet effective decoder module refine segmentation results along object boundaries.

Method	mIOU
Adelaide_VeryDeep_FCN_VOC [85]	79.1
LRR_4x_ResNet-CRF [25]	79.3
DeepLabv2-CRF [11]	79.7
CentraleSupelec Deep G-CRF [8]	80.2
HikSeg_COCO [80]	81.4
SegModel [75]	81.8
Deep Layer Cascade (LC) [52]	82.7
TuSimple [84]	83.1
Large_Kernel_Matters [68]	83.6
Multipath-RefineNet [54]	84.2
ResNet-38_MS_COCO [86]	84.9
PSPNet [95]	85.4
IDW-CNN [83]	86.3
CASIA_IVA_SDN [23]	86.6
DIS [61]	86.8
DeepLabv3	85.7
DeepLabv3-JFT	86.9

Table 7. Performance on PASCAL VOC 2012 *test* set.

Method	mIOU
Deep Layer Cascade (LC) [42]	82.7
TuSimple [75]	83.1
Large_Kernel_Matters [57]	83.6
Multipath-RefineNet [43]	84.2
ResNet-38_MS_COCO [77]	84.9
PSPNet [81]	85.4
IDW-CNN [73]	86.3
CASIA_IVA_SDN [20]	86.6
DIS [50]	86.8
DeepLabv3 [10]	85.7
DeepLabv3-JFT [10]	86.9
DeepLabv3+ (Xception)	87.8
DeepLabv3+ (Xception-JFT)	89.0

Table 6. PASCAL VOC 2012 *test* set results with top-performin
models. We refer interested readers to leaderboard for details.

See More Than Once -- Kernel-Sharing Atrous Convolution for Semantic Segmentation

- using KSAC instead of ASPP 62% of the parameters are saved when dilation rates of 6,12 and 18 are used

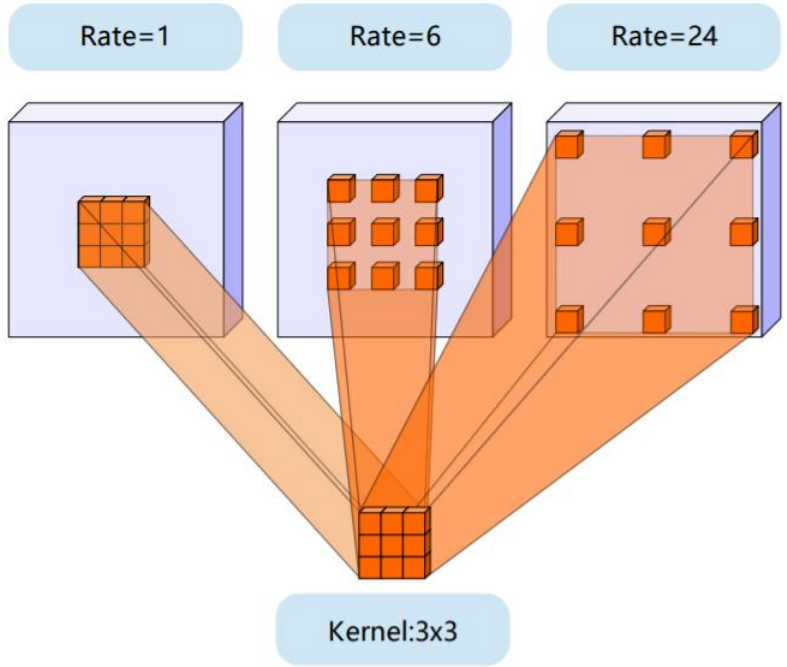
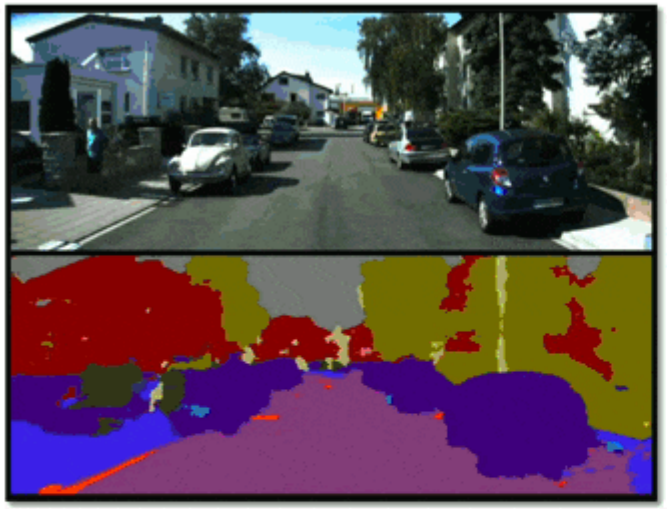


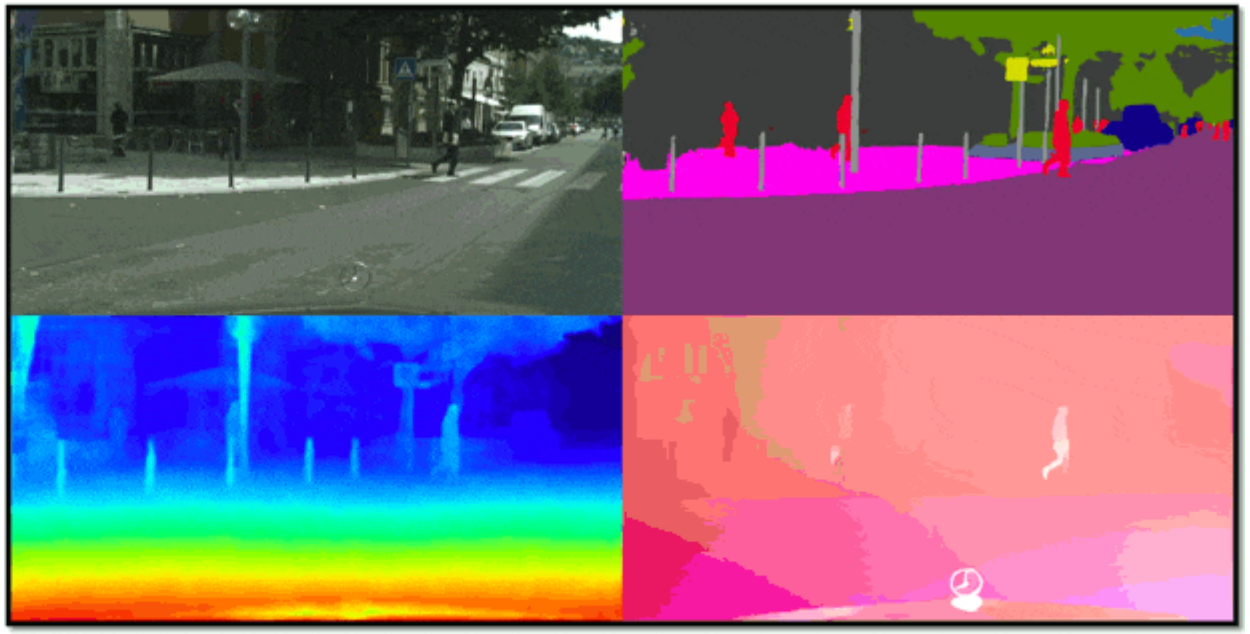
Figure 2. Illustration of our proposed Kernel-Sharing Atrous Convolution structure. The single 3 × 3 kernel is shared by three parallel branches with different atrous rates.

ASPP不是每個並行層都有不同的kernel，而是共享一個kernel，增強了泛化能力

MASK R-CNN

- At the back of the slides





Progression of computer vision from

2015

... to 2018

Deep Layer Aggregation

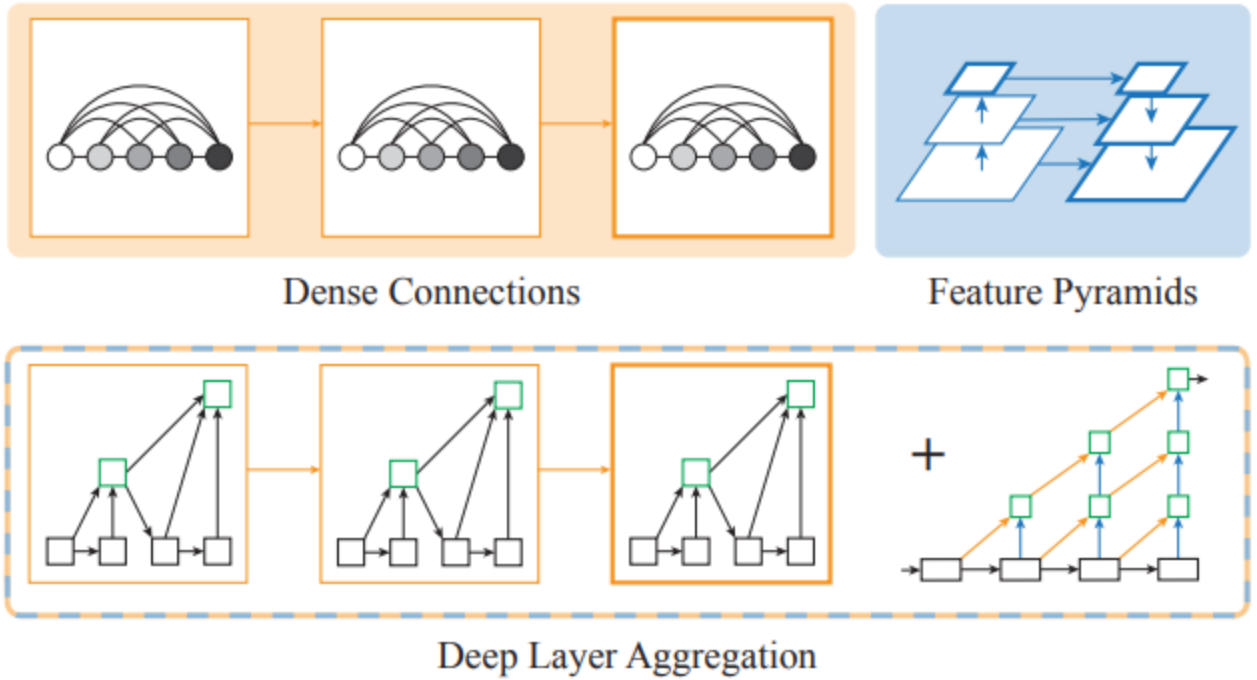


Figure 1: Deep layer aggregation unifies semantic and spatial fusion to better capture what and where. Our aggregation architectures encompass and extend densely connected networks and feature pyramid networks with hierarchical and iterative skip connections that deepen the representation and refine resolution.

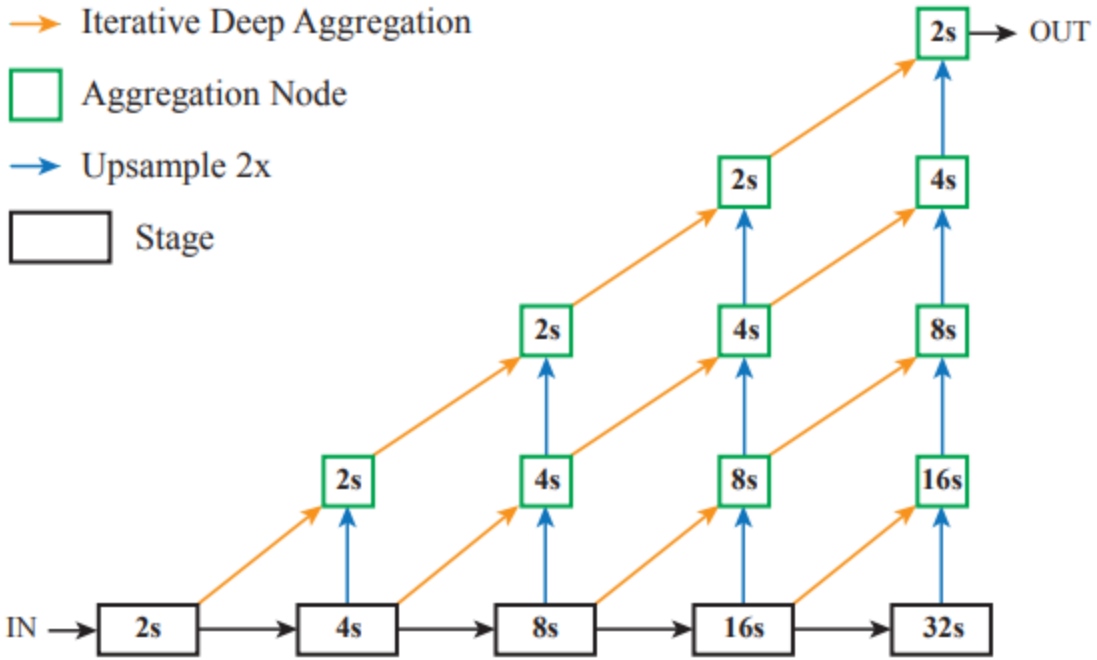


Figure 4: Interpolation by iterative deep aggregation. Stages are fused from shallow to deep to make a progressively deeper and higher resolution decoder.

Semantic segmentation
Dense Prediction Networks

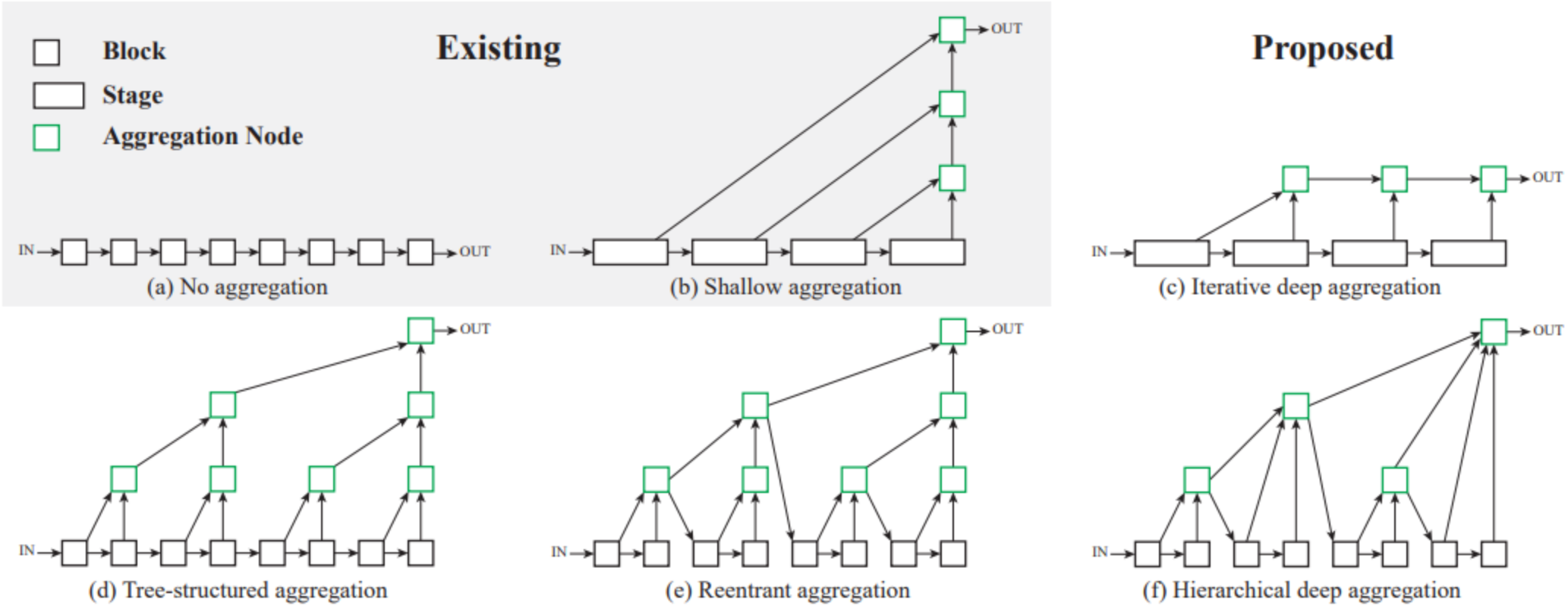
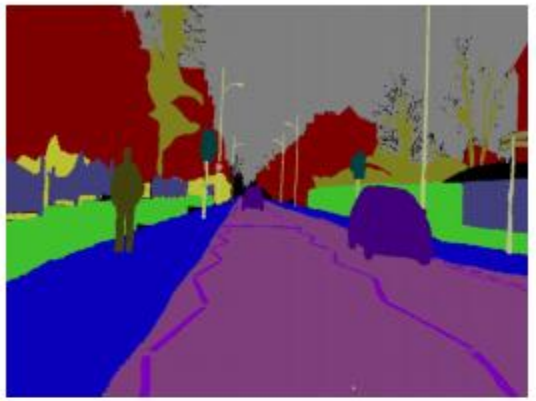


Figure 2: Different approaches to aggregation. (a) composes blocks without aggregation as is the default for classification and regression networks. (b) combines parts of the network with skip connections, as is commonly used for tasks like segmentation and detection, but does so only shallowly by merging earlier parts in a single step each. We propose two deep aggregation architectures: (c) aggregates iteratively by reordering the skip connections of (b) such that the shallowest parts are aggregated the most for further processing and (d) aggregates hierarchically through a tree structure of blocks to better span the feature hierarchy of the network across different depths. (e) and (f) are refinements of (d) that deepen aggregation by routing intermediate aggregations back into the network and improve efficiency by merging successive aggregations at the same depth. Our experiments show the advantages of (c) and (f) for recognition and resolution.

Semantic Segmentation

Method	Split	mIoU
DLA-34 8s	Val	73.5
DLA-34 2s		75.1
DLA-102 2s		74.4
FCN-8s [35]	Test	65.3
RefineNet-101 [28]		73.6
DLA-102		75.3
DLA-169		**75.9**

Table 4: Evaluation on Cityscapes to compare strides on validation and to compare against existing methods on test. DLA is the best-in-class among methods in the same setting.



Method	mIoU
SegNet [2]	46.4
DeepLab-LFOV [7]	61.6
Dilation8 [45]	65.3
FSO [24]	66.1
DLA-34 8s	66.7
DLA-34 2s	68.6
DLA-102 2s	**71.0**

Table 5: Evaluation on CamVid. Higher depth and resolution help. DLA is state-of-the-art.

ResNeSt 之語義分割

RANK	METHOD	VALIDATION MIOU	TEST SCORE	PAPER TITLE	YEAR
1	**ResNeSt-269**	47.60		ResNeSt: Split-Attention Networks	2020
2	**ResNeSt-101**	46.91		ResNeSt: Split-Attentio	
3	CPN (ResNet-101)	46.27		Context Prior for Scene	
4	LaU-regression-loss	45.02	0.5632	Location-aware Upsam	
5	PSPNet	44.94	0.5538	Pyramid Scene Parsing	
6	CFNet (ResNet-101)	44.89		Co-Occurrent Features	
7	EncNet	44.65	0.5567	Context Encoding for S	

	Method	Backbone	pixAcc%	mIoU%
Prior Work	UperNet [59]	ResNet101	81.01	42.66
	PSPNet [69]	ResNet101	81.39	43.29
	EncNet [65]	ResNet101	81.69	44.65
	CFNet [66]	ResNet101	81.57	44.89
	OCNet [63]	ResNet101	-	45.45
	ACNet [17]	ResNet101	81.96	45.90
Ours	DeeplabV3 [7]	ResNet50 [21]	80.39	42.1
		ResNet101 [21]	81.11	44.14
		ResNeSt-50 (ours)	81.17	45.12
		ResNeSt-101 (ours)	**82.07**	**46.91**
		ResNeSt-269 (ours)	82.62	47.60

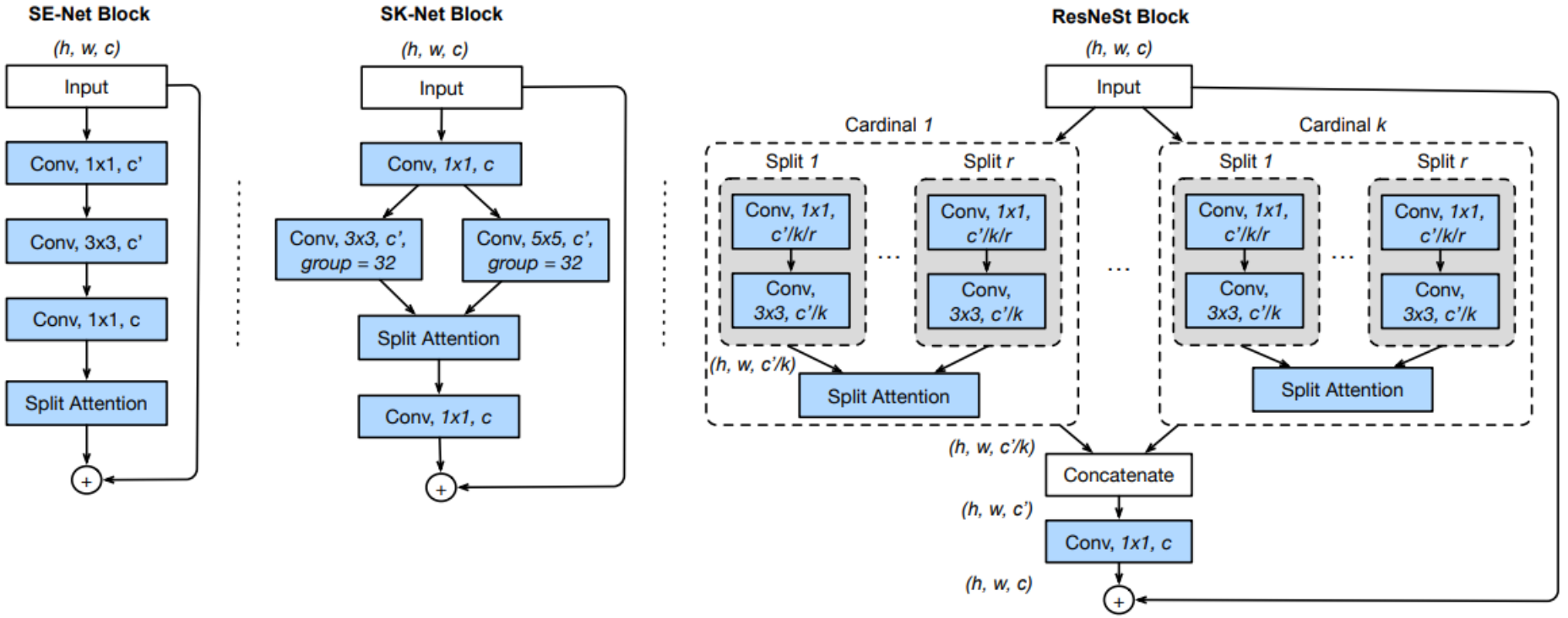


Fig. 1: Comparing our ResNeSt block with SE-Net [30] and SK-Net [38]. A detailed view of Split-Attention unit is shown in Figure 2. For simplicity, we show ResNeSt block in cardinality-major view (the featuremap groups with same cardinal group index reside next to each other). We use radix-major in the real implementation, which can be modularized and accelerated by group convolution and standard CNN layers (see supplementary material).

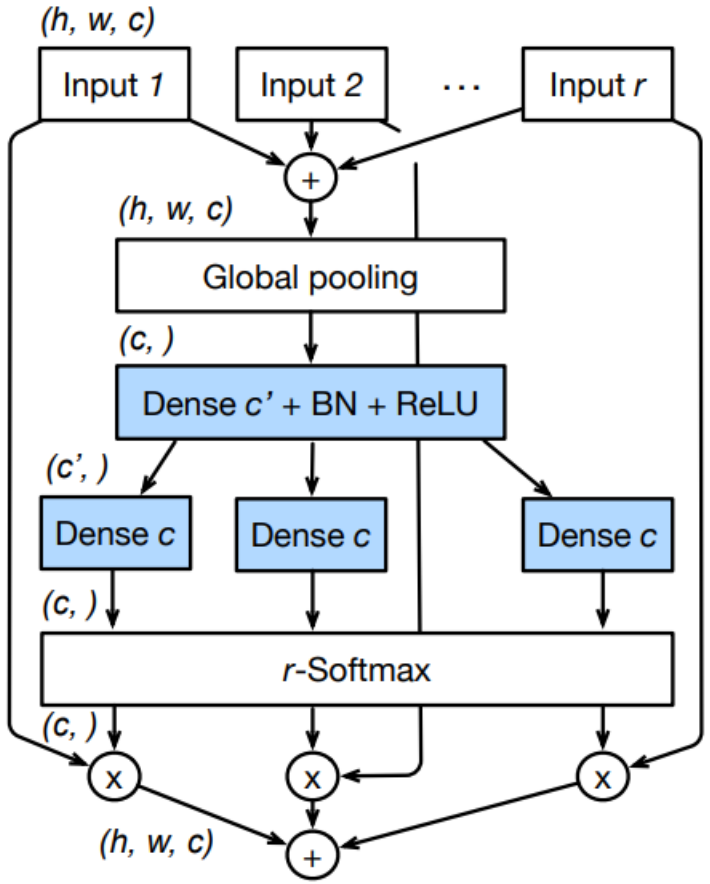


Fig. 2: Split-Attention within a cardinal group. For easy visualization in the figure, we use $c = C/K$ in this figure.

TRANSFORMER FOR SEGMENTATION

Segmenter

Pretrained ViT

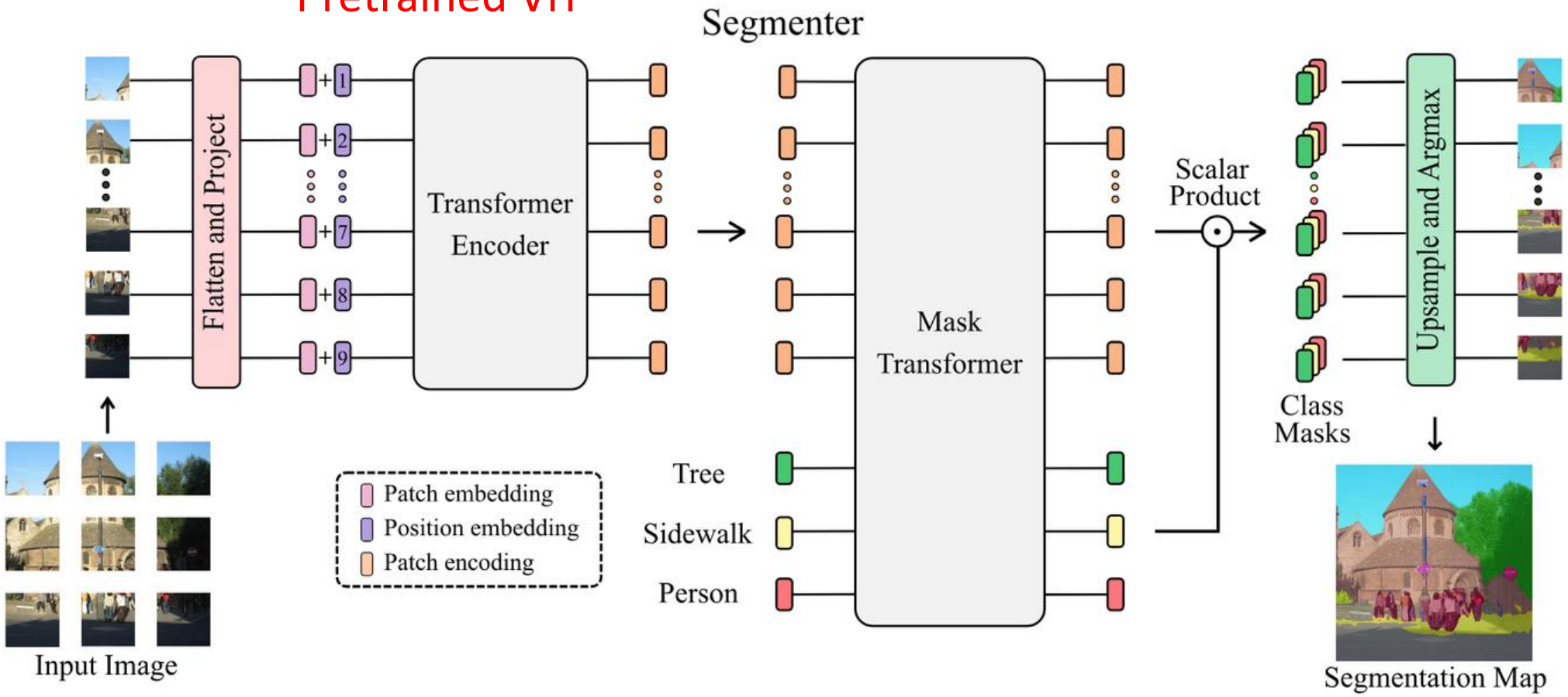
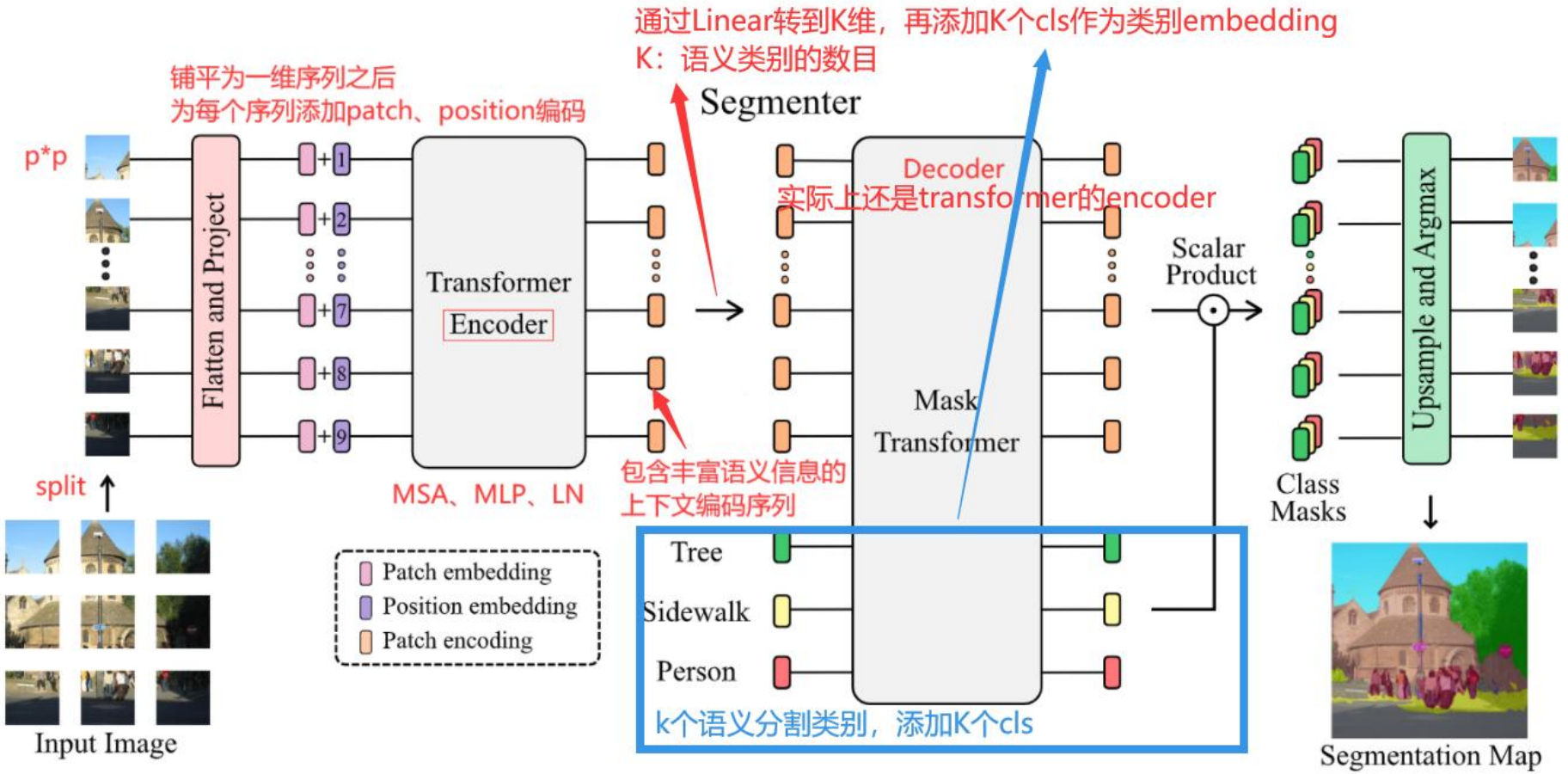


Figure 2: Overview of our approach Segmenter. (Left) Encoder: The image patches are projected to a sequence of embeddings and then encoded with a transformer. (Right) Decoder: A mask transformer takes as input the output of the encoder and class embeddings to predict segmentation masks. See text for details.



借鉴了DETR、Max-DeepLab、SOLO-v2，
利用对象embedding来产生实例掩码

优势在于：指先添加patch embedding 跑一次transformer，再添加
class embedding 跑一次transformer；计算负担比先前的方法要小

Method	Backbone	Patch size	ImNet acc.	mIoU
DeepLab V3+	ResNet50-D	-	-	43.95
DeepLab V3+	ResNet101-D	-	-	**45.47**
Seg-S†/16	DeiT-S	16	81.20	42.40
Seg-B†/16	DeiT-B	16	**85.20**	**47.10**
Seg-B/32	ViT-B	32	81.65	40.92
Seg-B/16	ViT-B	16	83.95	45.69
Seg-B/8	ViT-B	8	**85.35**	**48.06**
Seg-L/32	ViT-L	32	81.68	42.64
Seg-L/16	ViT-L	16	**84.95**	**48.60**

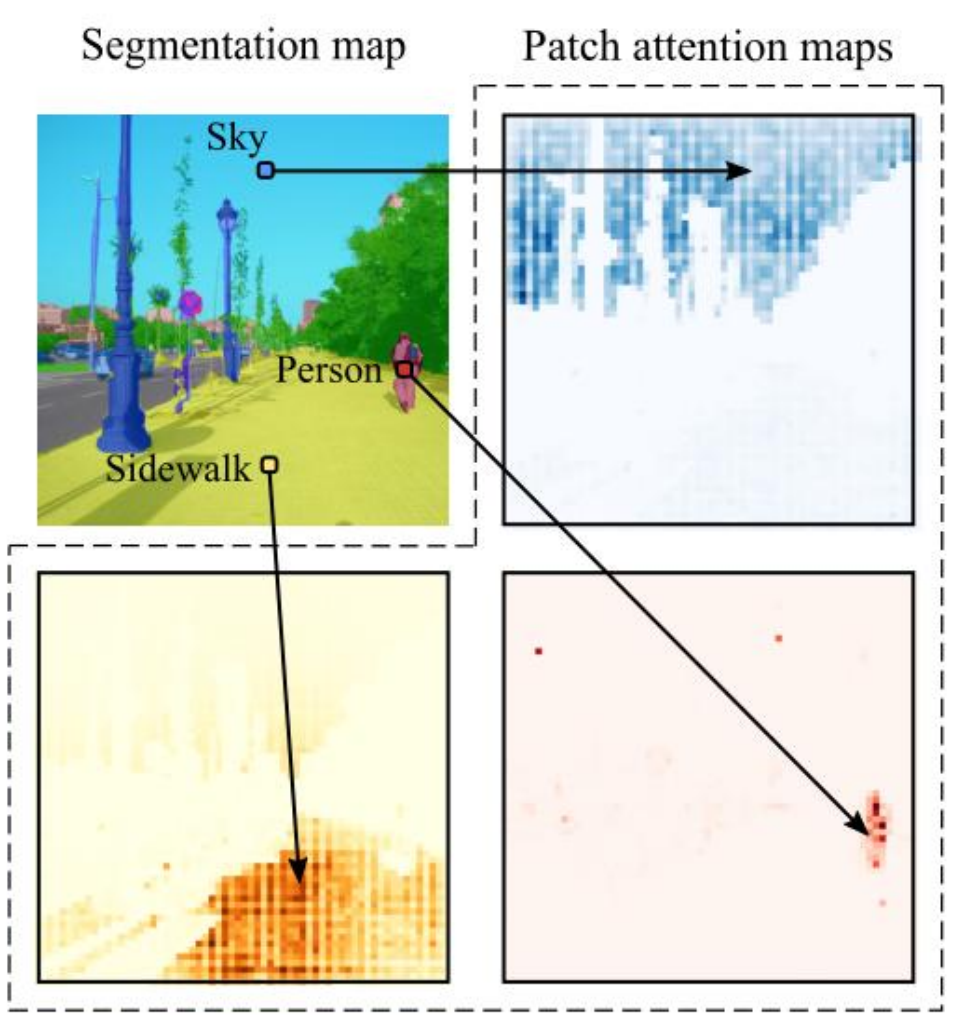


Figure 1: Our approach for semantic segmentation is purely transformer based. It leverages the global image context at every layer of the model. Attention maps from the first Segmenter layer are displayed for three 8×8 patches and highlight the early grouping of patches into semantically meaningful categories. The original image (top-left) is overlayed with segmentation masks produced by our method.

SegFormer

Hierarchical Transformer Encoder
Positional Encoding-Free Design
Lightweight All-MLP Decoder

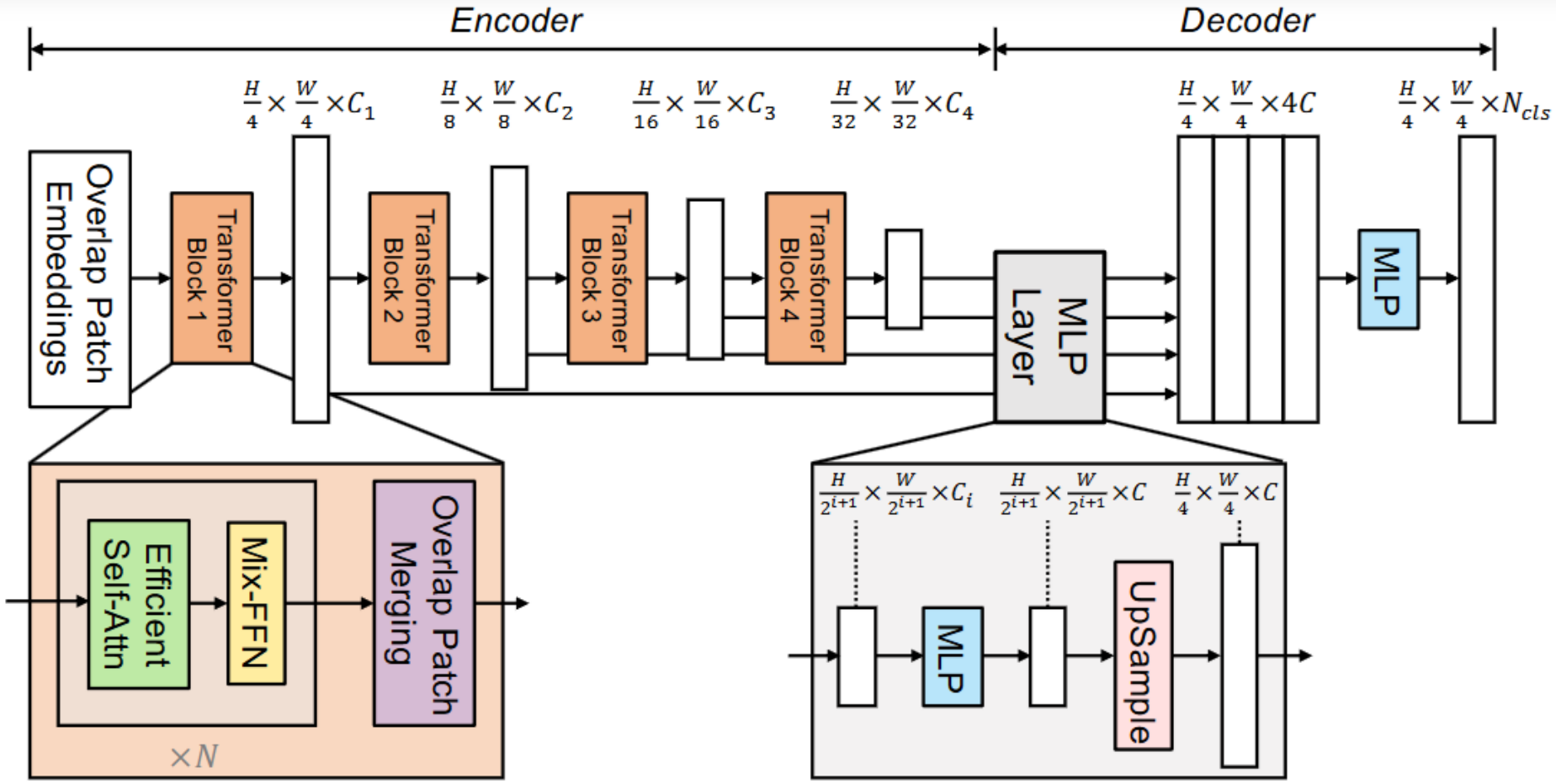
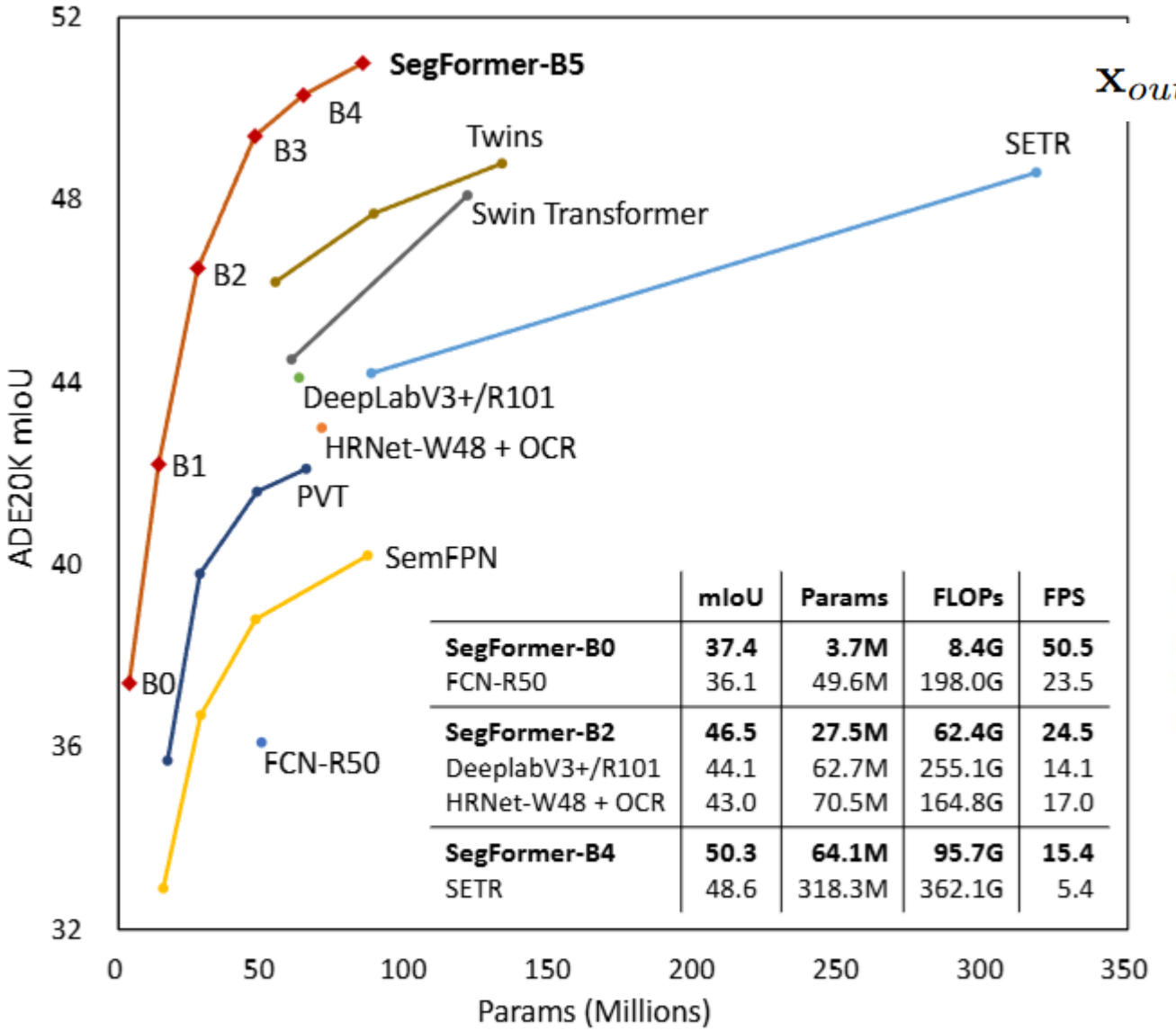


Figure 2: **The proposed SegFormer framework** consists of two main modules: A hierarchical Transformer encoder to extract coarse and fine features; and a lightweight All-MLP decoder to directly fuse these multi-level features and predict the semantic segmentation mask. "FFN" indicates feed-forward network.



	mIoU	Params	FLOPs	FPS
SegFormer-B0	**37.4**	**3.7M**	**8.4G**	**50.5**
FCN-R50	36.1	49.6M	198.0G	23.5
SegFormer-B2	**46.5**	**27.5M**	**62.4G**	**24.5**
DeeplabV3+/R101	44.1	62.7M	255.1G	14.1
HRNet-W48 + OCR	43.0	70.5M	164.8G	17.0
SegFormer-B4	**50.3**	**64.1M**	**95.7G**	**15.4**
SETR	48.6	318.3M	362.1G	5.4

Figure 1: **Performance *vs.* model efficiency on ADE20K.** All results are reported with single model and single-scale inference. SegFormer achieves a new state-of-the-art 51.0% mIoU while being significantly more efficient than previous methods.

MIX-FFN

$$\mathbf{x}_{out} = \mathrm{MLP}(\mathrm{GELU}(\mathrm{Conv}_{3\times3}(\mathrm{MLP}(\mathbf{x}_{in})))) + \mathbf{x}_{in},$$

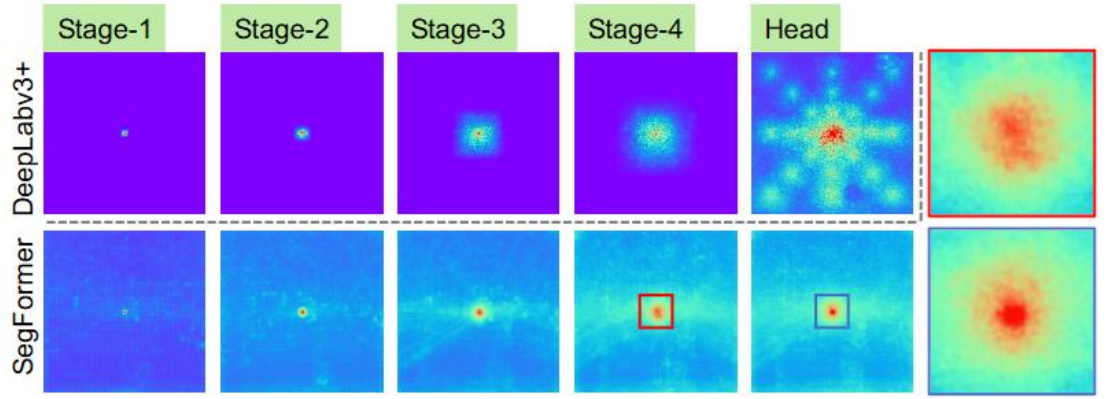


Figure 3: **Effective Receptive Field (ERF) on Cityscapes** (average over 100 images). Top row: Deeplabv3+. Bottom row: SegFormer. ERFs of the four stages and the decoder heads of both architectures are visualized. Best viewed with zoom in.

對於語義分割來說最重要的問題就是如何增大感受野，對於CNN encoder來說，有效感受野是比較小且局部的，所以需要一些decoder 的設計來增大有效感受野，

但是對於Transformer encoder來說，由於 self-attention這一操作，有效感受野變得非常大，因此 decoder 不需要更多操作來提高感受野

Promptable and Language-Driven Segmentation

SEGMENTATION WITH FOUNDATION MODELS

Foundation Models

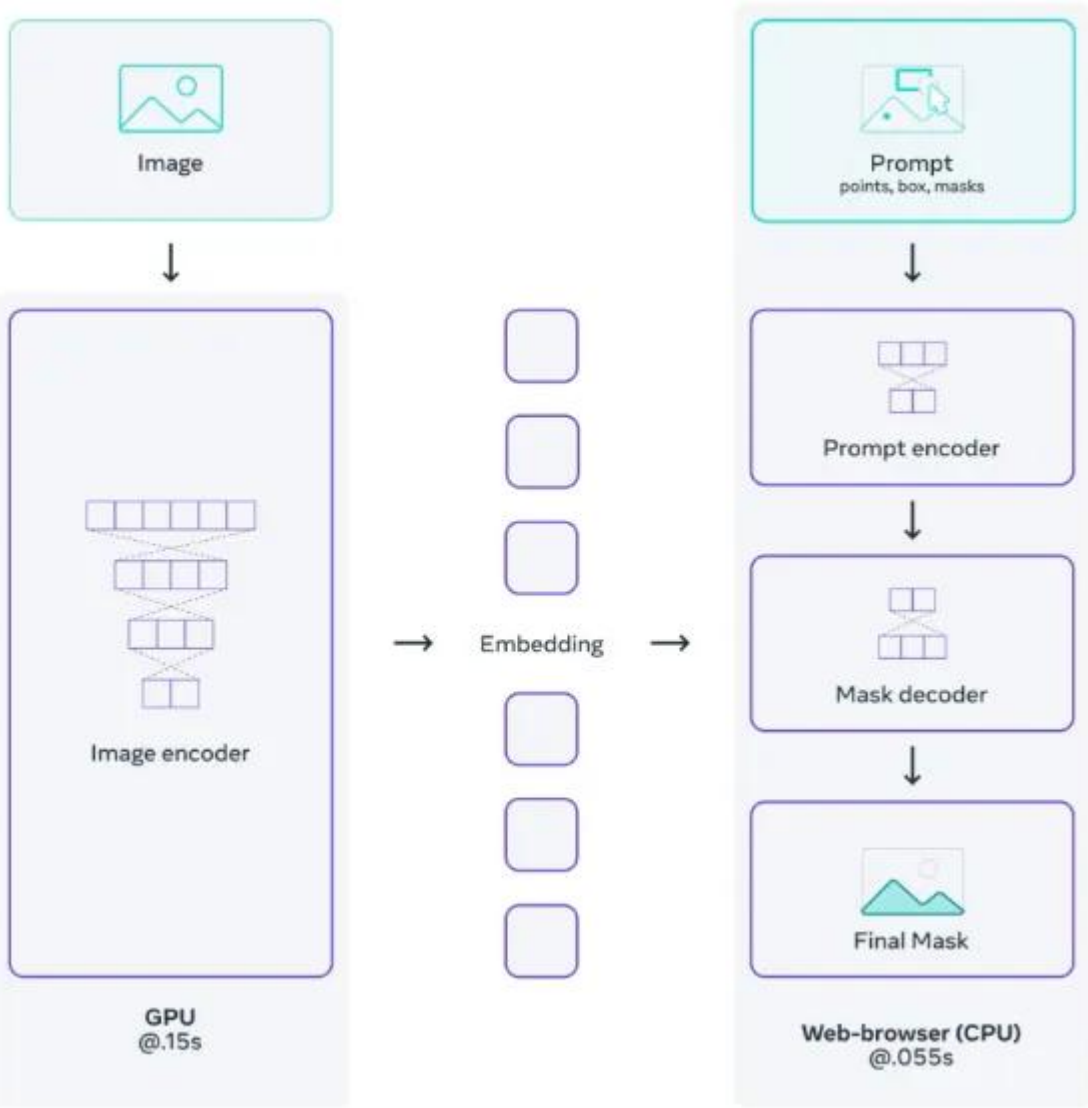
- foundation model
  - a large-scale machine learning or deep learning model that has been trained on vast datasets, enabling it to be applied across a wide range of tasks and applications.
  - The term was popularized by researchers at Stanford University in 2021, who defined foundation models as those trained on broad data using self-supervision, capable of being adapted for various downstream tasks

Key Characteristics of Foundation Models

- Generalization
  - perform well across multiple tasks without needing extensive retraining for each specific application
  - Zero short or few shot learning
- Large dataset
  - Trained on large, unlabeled dataset
  - Often by self-supervised learning
- Multimodel capabilities
- Resource intensive
- Example
  - GPT, Dalle-E, SAM

SAM: Segment Anything

- Models
  - Image encoder, prompt encoder, mask encoder
- Features
  - Generalization Across Tasks
  - Promptable Segmentation
- Dataset collection
  - 三個階段
    - 輔助手動標註、半自動標註、全自動標註
  - 最大的影像分割數據集 SA-1B，其中包含了10億個遮罩(Mask)和1,100萬張經過授權的圖像，影像平均的大小為 3,300 x 4,950



Segment Anything

Research by Meta AI

Home Demo Dataset Blog Paper

Tools

Upload Gallery

Hover & Click

Box

Everything

Find all the objects in the image automatically.

Cut out all objects

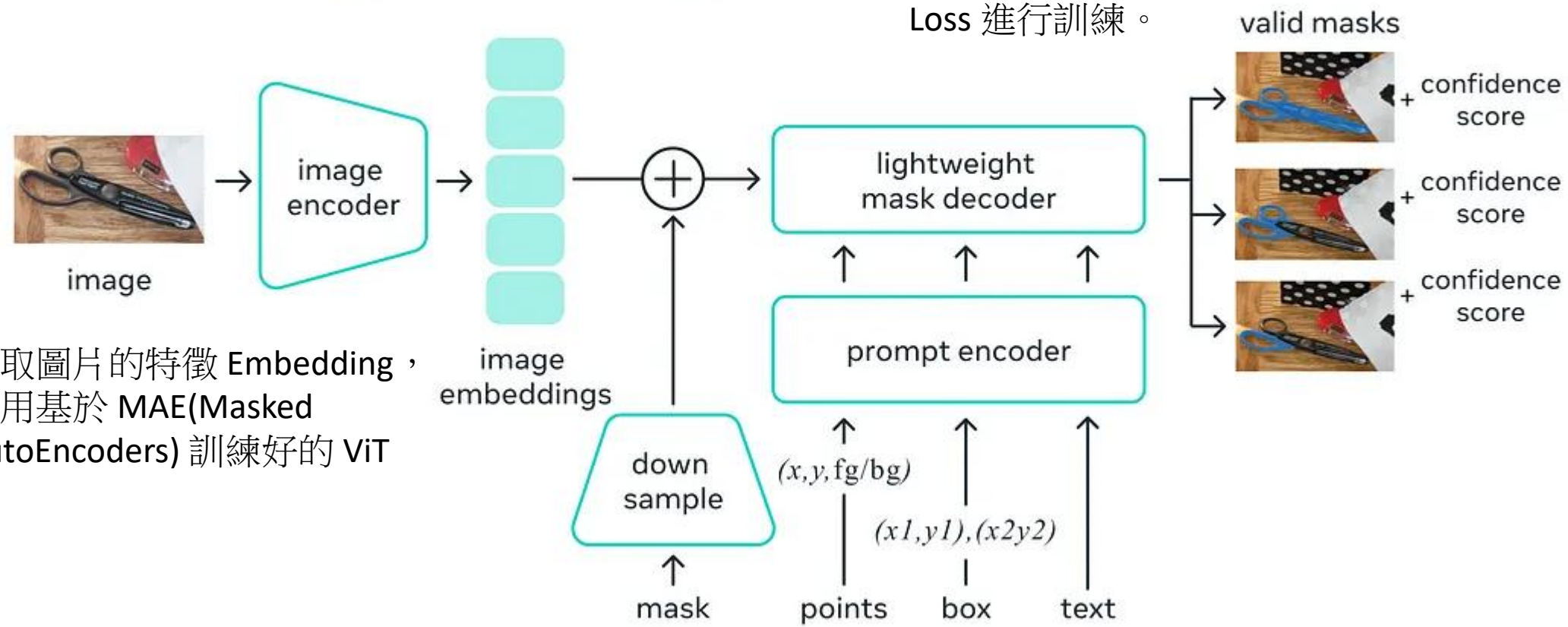
Cut-Outs

Prompting SAM with each point in the grid

SAM 可能會錯過精細結構，有時會產生小的斷開連接的組件，並且生成的邊界不如計算量更大的方法清晰

https://ai.meta.com/blog/segment-anything-foundation-model-image-segmentation/

參考 DETR 以及 MaskFormer 的 Mask Classification 架構，可以同時預測出 Semantic 以及 Instance-level 的影像分割結果，整體訓練方式簡單，使用 Focal Loss + Dice Loss 進行訓練。



萃取圖片的特徵 Embedding，使用基於 MAE(Masked AutoEncoders) 訓練好的 ViT

Prompt 支援文字和空間資訊，空間的資訊可以分成兩個種類稀疏(Sparse)型態的 Prompt，如 Points, boxes，以及密集(Dense)型態的 Prompt，如 Mask。

針對密集(Dense)型態的 Prompt，是先將 Mask 結合 Convolution 進行 Downsampling 後，成為 Mask embedding，直接與 Image embedding 相加。

針對稀疏(Sparse)型態的 Prompt，會基於空間上的座標進行轉換後變成 Positional embedding，再加上各自型態所學出的 embedding(Box embedding, Point embedding 等)，直接與 Image embedding 相加。

Faster Segment Anything: Towards Lightweight SAM for Mobile Applications

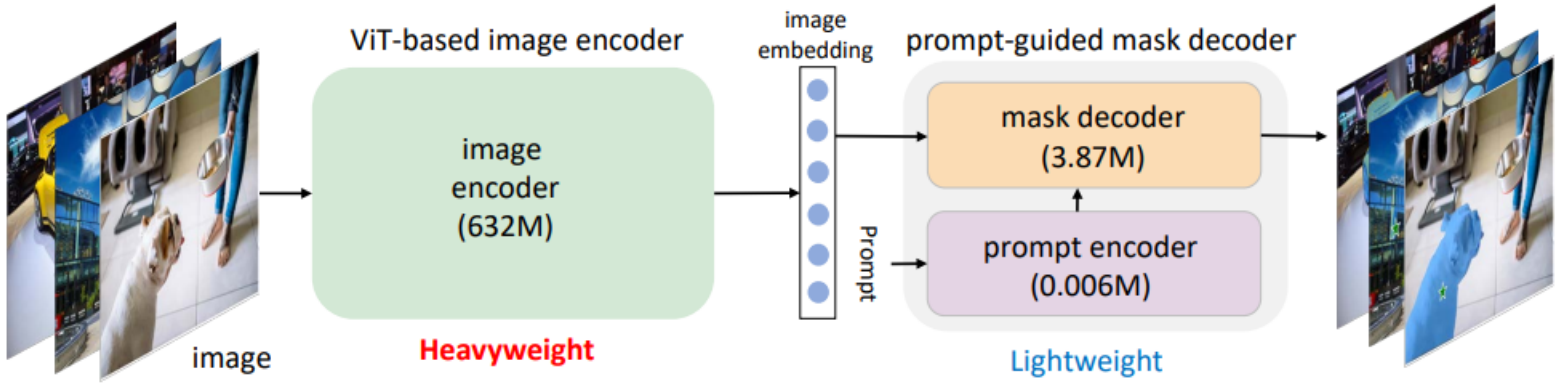


Figure 1: The overview of Segment Anything Model.

Table 1: Parameters SAM with different image encoders.

Parameters	SAM (ViT-H)	SAM (ViT-L)	SAM (ViT-B)
ViT-based encoder	632M	307M	86M
prompt-guided encoder	0.006M	0.006M	0.006M

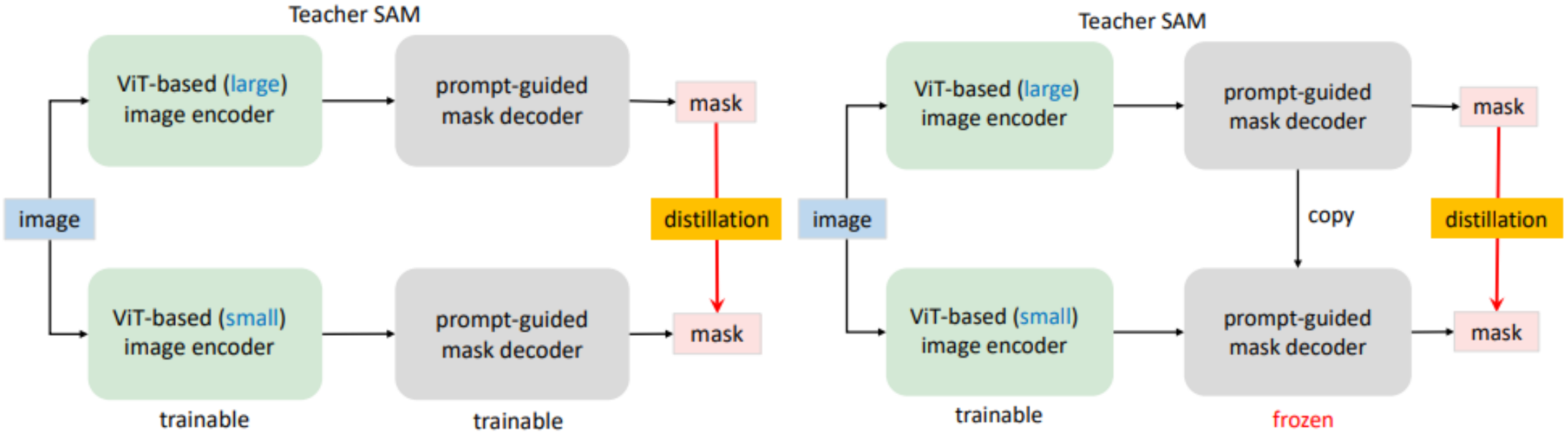


Figure 2: Coupled knowledge distillation of SAM. The left subfigure denotes the fully-coupled distillation, while the right one represents the semi-coupled distillation.

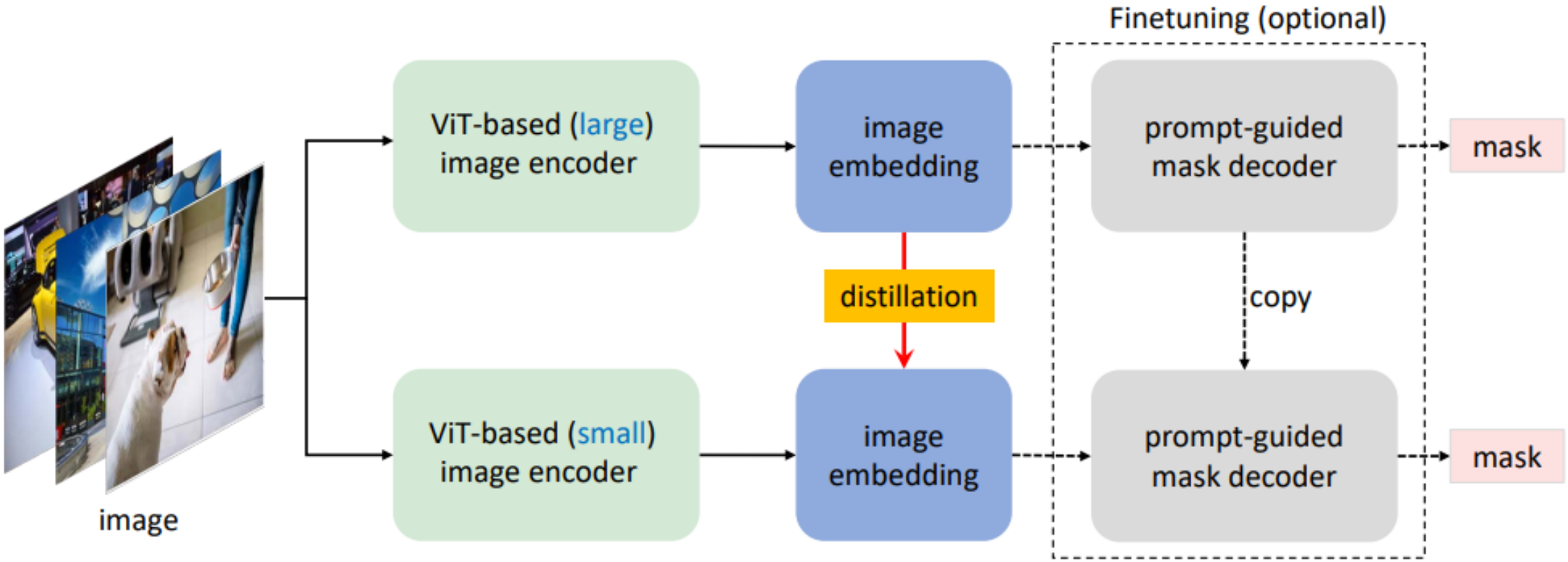


Figure 3: Decoupled distillation for SAM.

SUMMARY

Fully Convolutional Networks for Semantic Segmentation

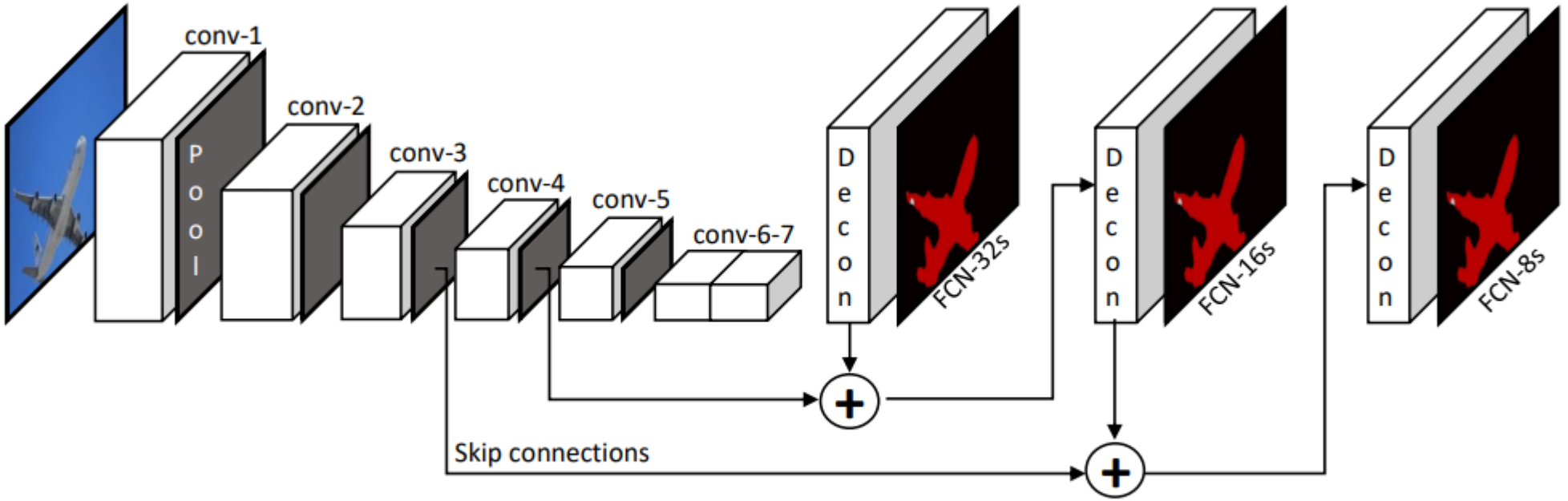
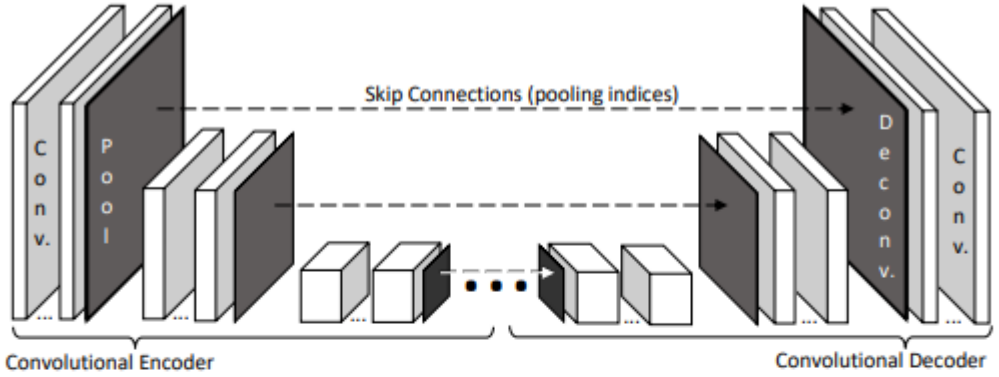
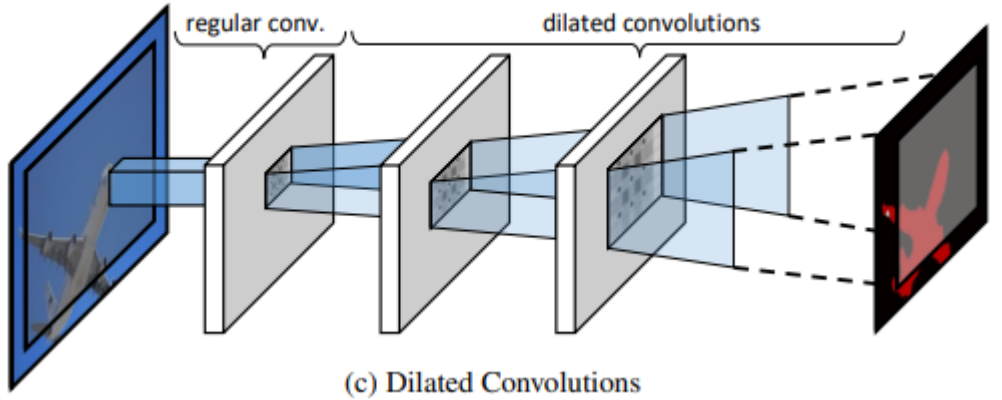


Figure 3: Fully convolutional networks (FCNs) are trained end-to-end and are designed to make dense predictions for per-pixel tasks like semantic segmentation. FCNs consist of no fully connected layers .

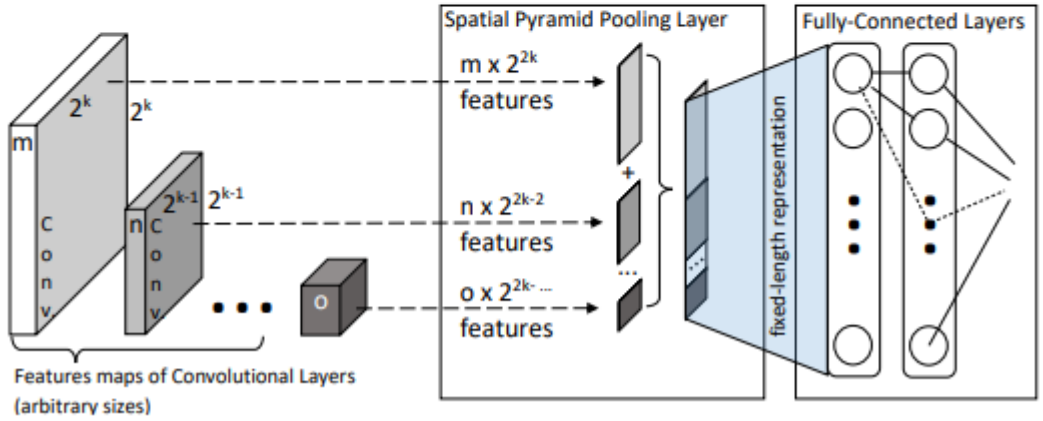
Techniques for Fine-grained Localisation



(a) Encoder-Decoder Architecture.



(c) Dilated Convolutions

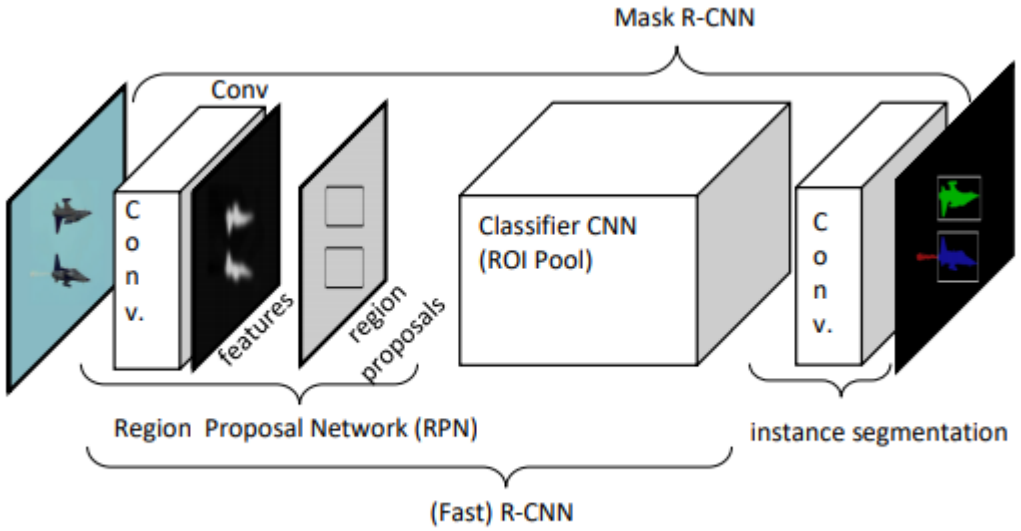


(b) Spatial-Pyramid Pooling Layer

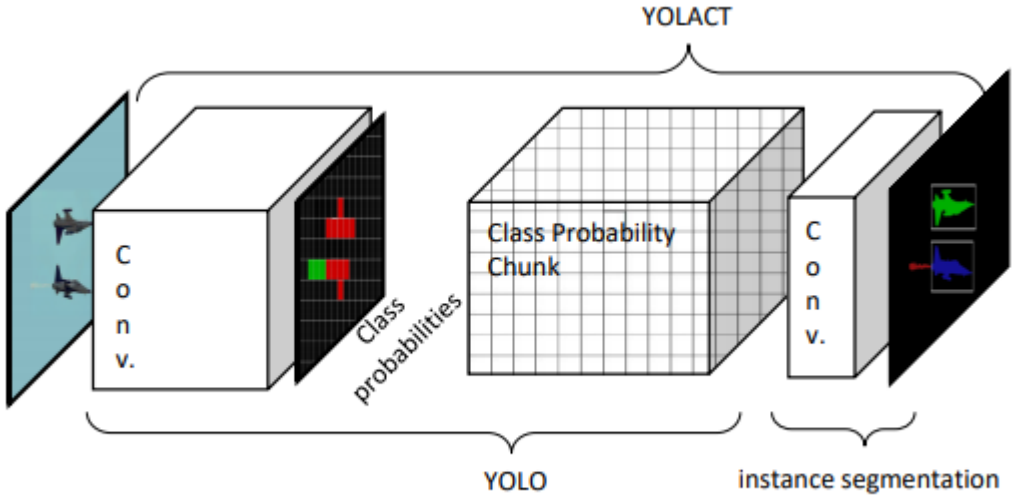
A Survey on Deep Learning-based Architectures for Semantic Segmentation on 2D images

Object Detection Based Semantic Segmentation

- Object bounding box + Semantic mask



(a) Regions with CNN features-based (Mask-RCNN) architecture

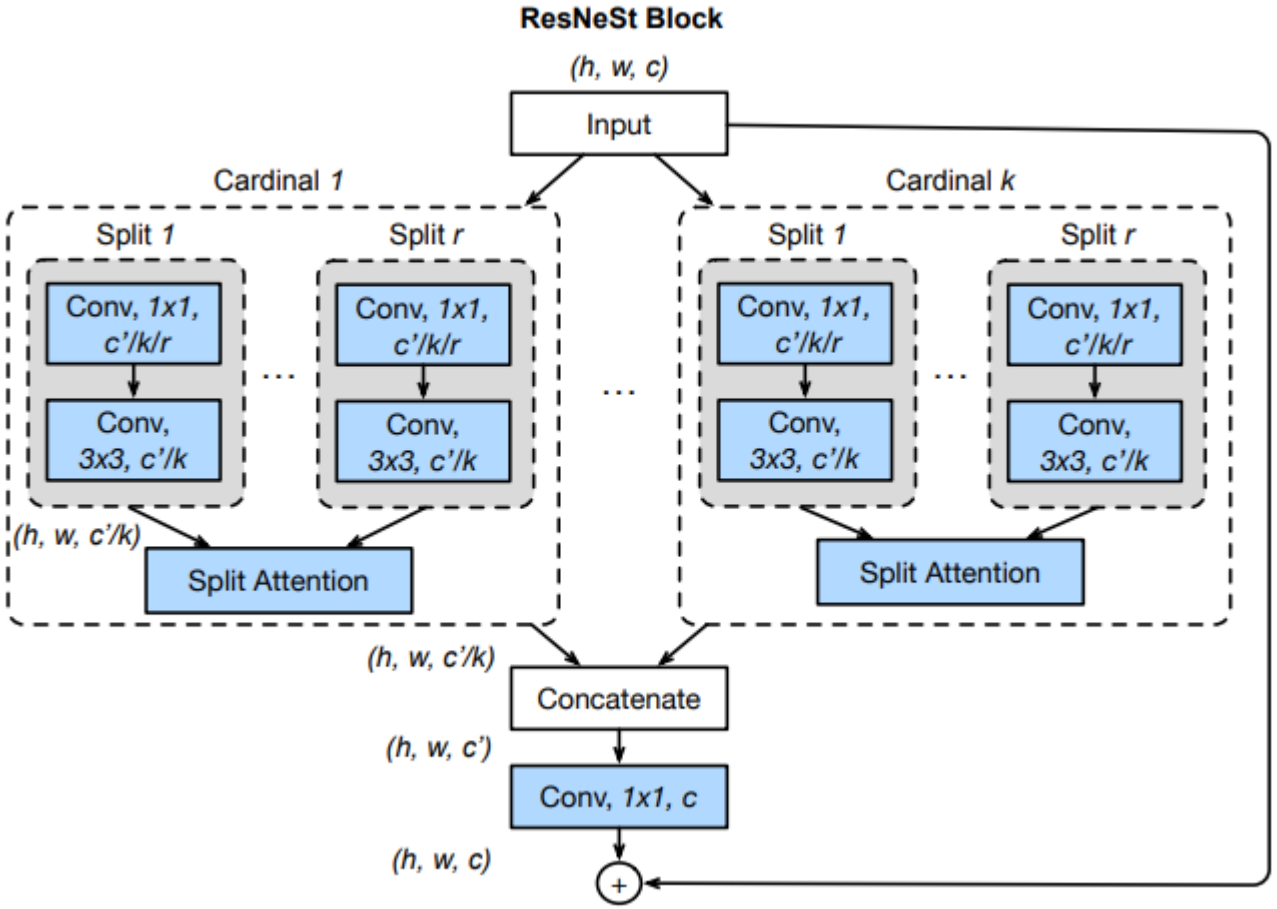


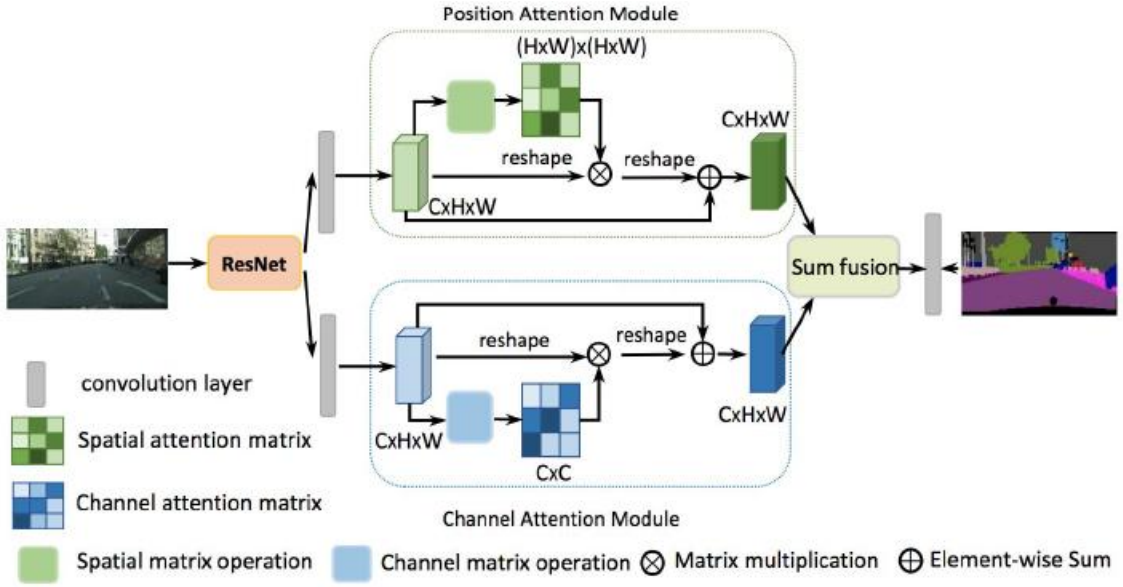
(b) Fully Convolutional Object Detector-based (YOLO)-based architecture

A Survey on Deep Learning-based Architectures for Semantic Segmentation on 2D images

Attention

- In backbone or feature fusion





Dual attention

INSTANCE SEGMENTATION

Ref: ICCV 2017 tutorial, Mask R-CNN: A perspective on equivariance