Computer Vision

CS308 Autumn

Feng Zheng

SUSTech CS Vision Intelligence and Perception

南方科技大学

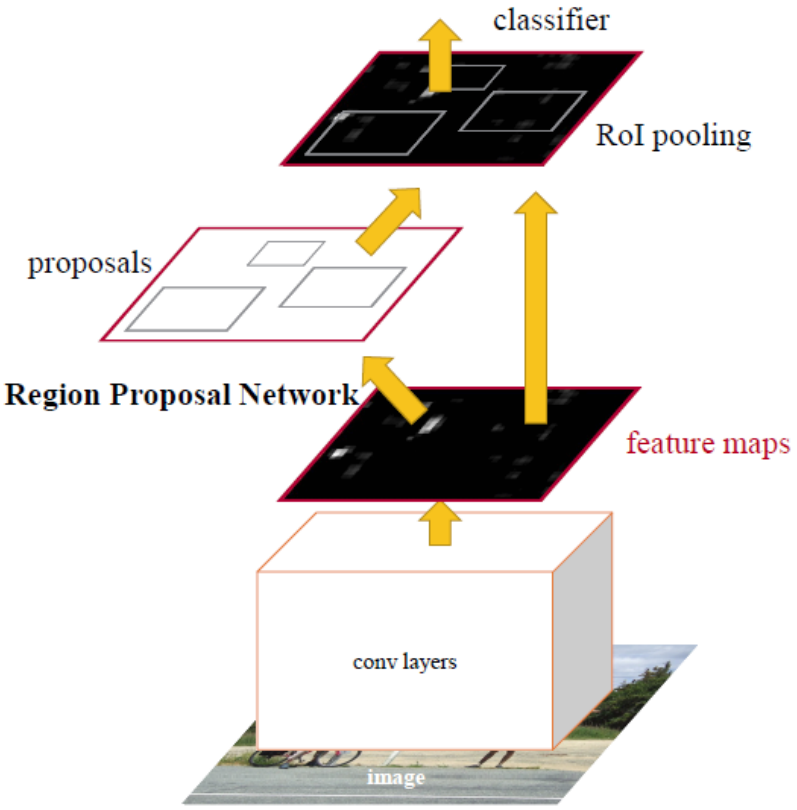
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Content

- Brief Review
- Semantic Segmentation
- Video Object Segmentation

Brief Review

Faster R-CNN



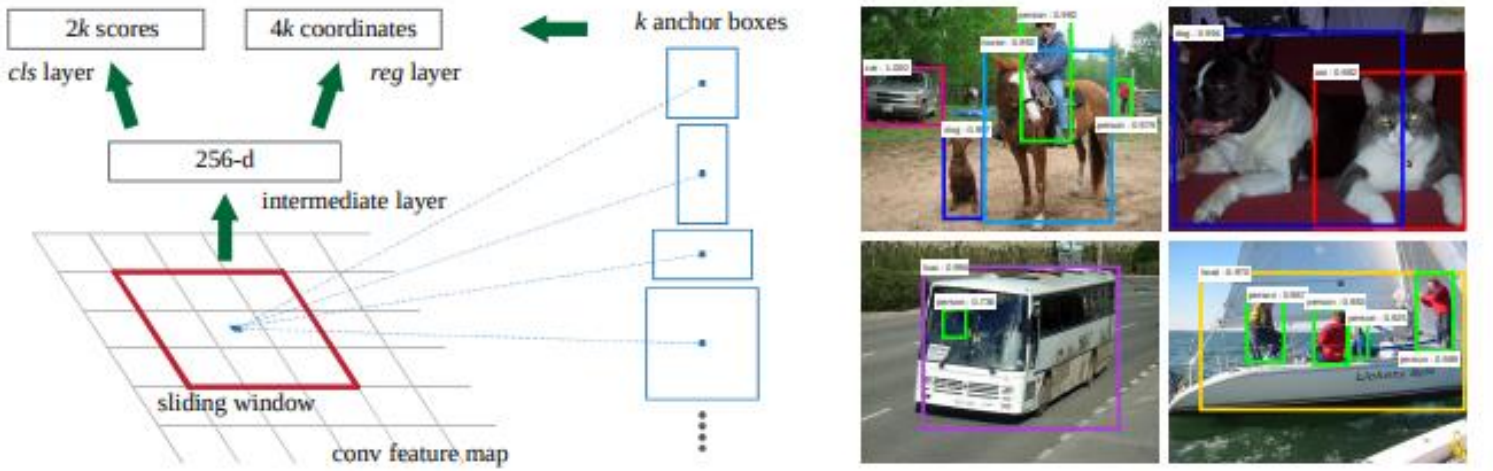
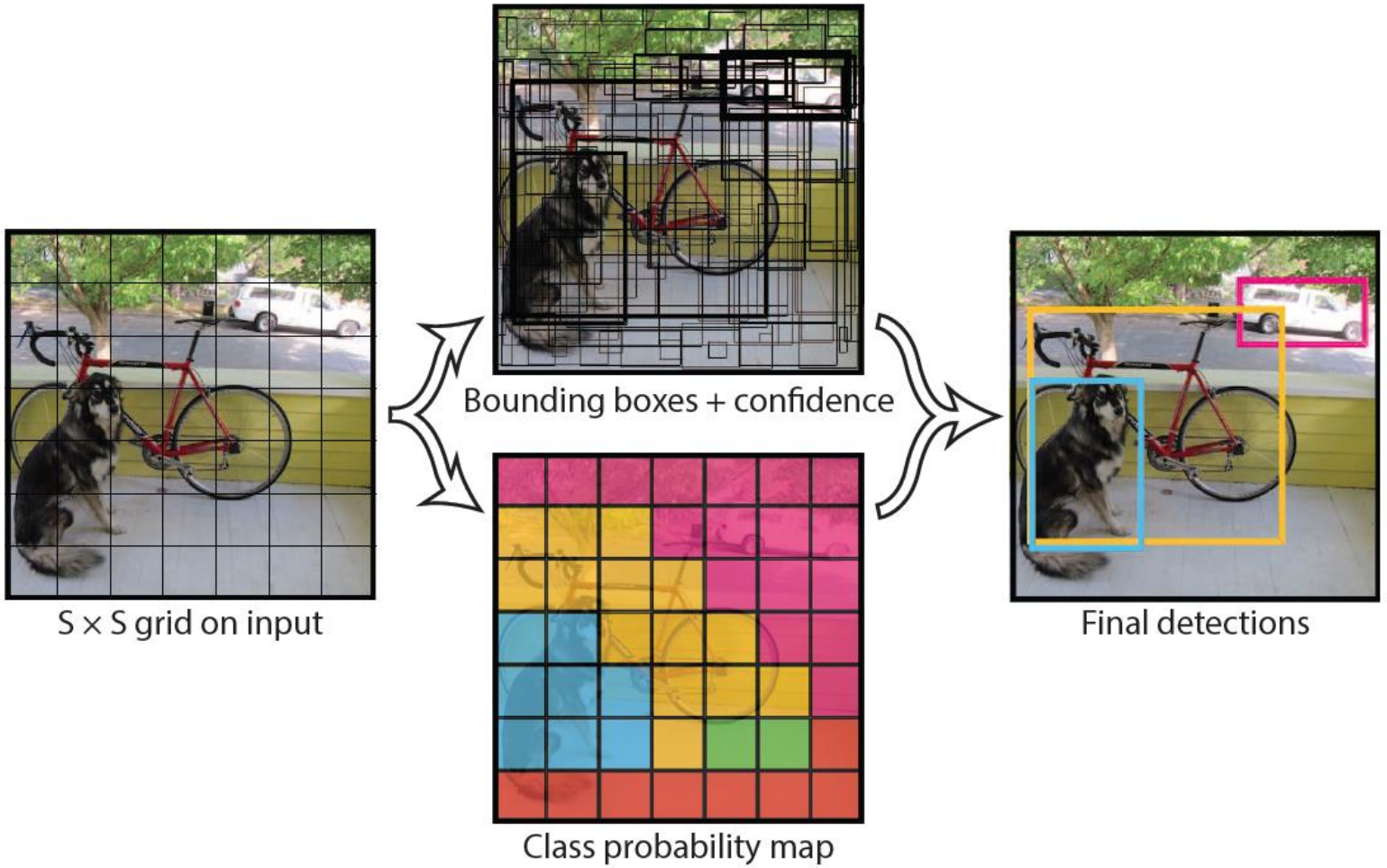


Figure 3: **Left**: Region Proposal Network (RPN). **Right**: Example detections using RPN proposals on PASCAL VOC 2007 test. Our method detects objects in a wide range of scales and aspect ratios.

YOLO Framework



$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

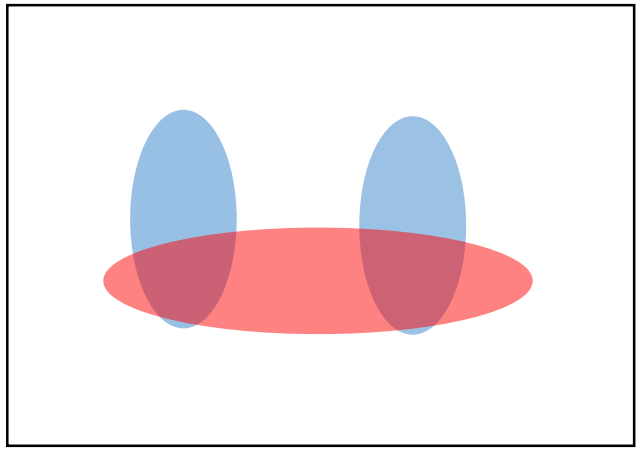
At test time we multiply the conditional class probabilities and the individual box confidence predictions

Semantic Segmentation

The Segmentation Task

Evaluation metric

- Pixel classification!
- Accuracy?
  - Heavily unbalanced
  - Common classes are over-emphasized
- *Intersection over Union*
  - Average across classes and images
- Per-class accuracy
  - Compute accuracy for every class and then average



Things vs Stuff

THINGS

- Person, cat, horse, etc
- Constrained shape
- Individual instances with separate identity
- May need to look at objects

STUFF

- Road, grass, sky etc
- Amorphous, no shape
- No notion of instances
- Can be done at pixel level
- "texture"

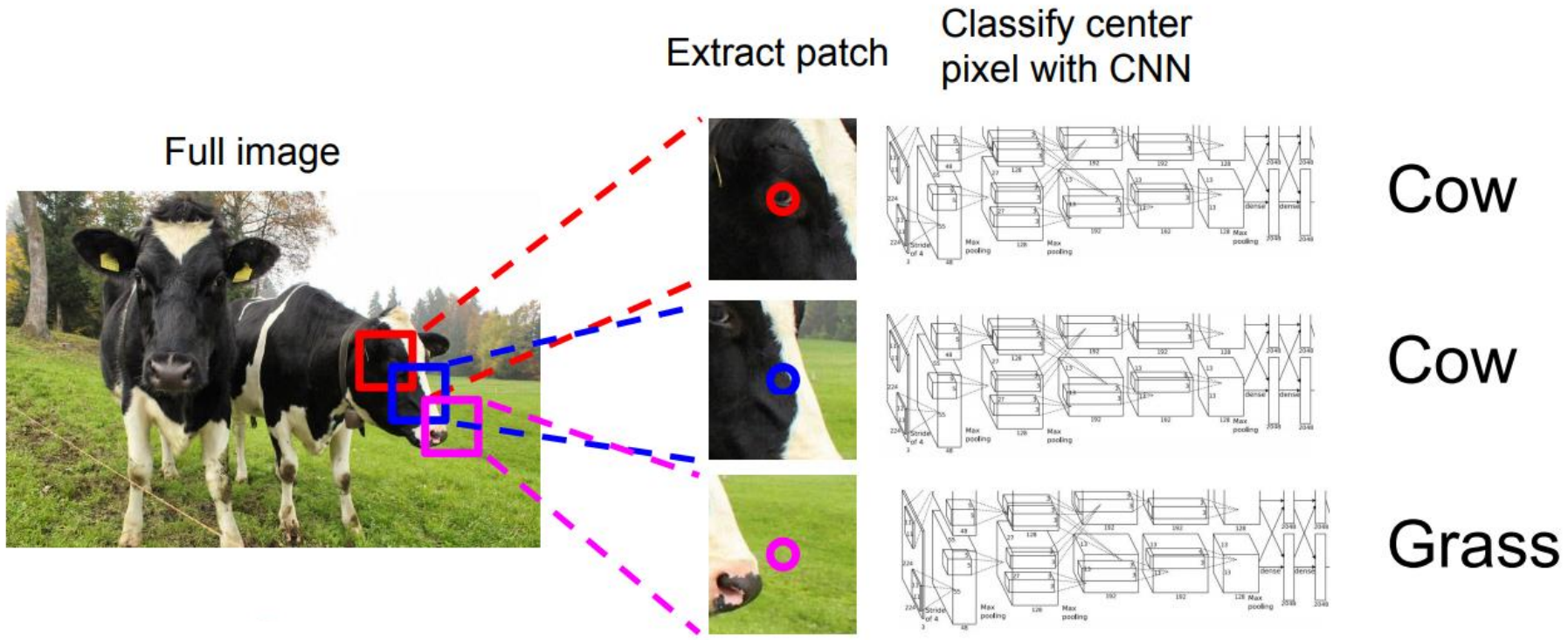
Challenges in data collection

- Precise localization is hard to annotate
- Annotating every pixel leads to heavy tails
  - Common solution: annotate few classes (often things), mark rest as "Other"
  - Weakly-supervised labels: scribble and points
- Common datasets:
  - PASCAL VOC 2012 (~1500 images, 20 categories),
  - COCO (~100k images, 20 categories)
  - The number of classes is limited

Classical semantic segmentation (Pre-convnet)

- Things
  - Do object detection, then segment out detected objects
- Stuff
  - "Texture classification"
  - Compute histograms of filter responses
  - Classify local image patches
- Classical segmentation methods
  - Graph cut
  - Clustering

Using Sliding Window for CNN Features

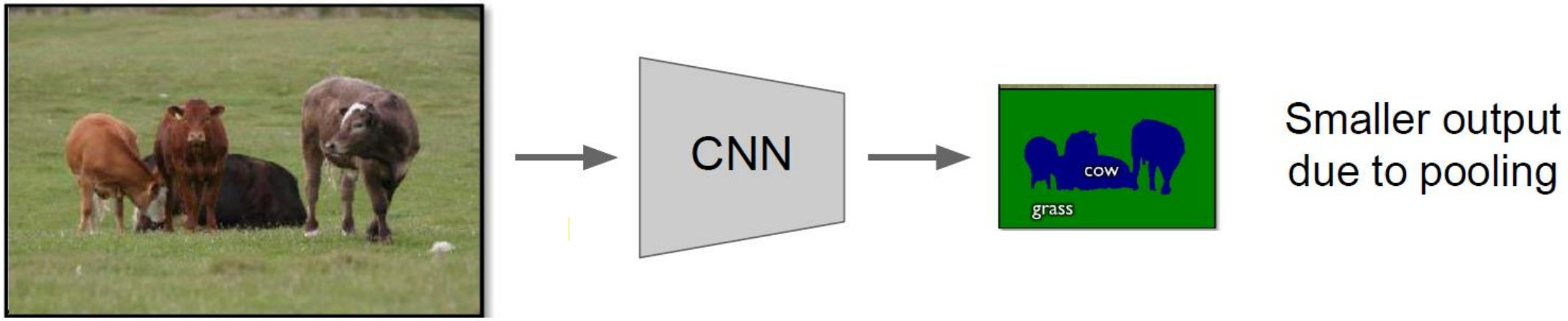


What's the problem?

Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

Using Convolution

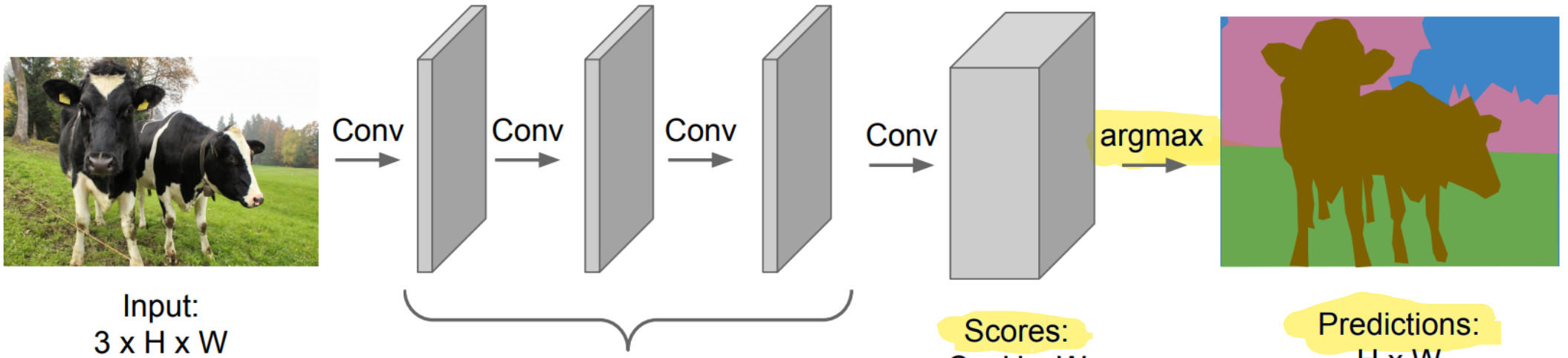
- We can use pooling:



What's the problem?

Using Convolution

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

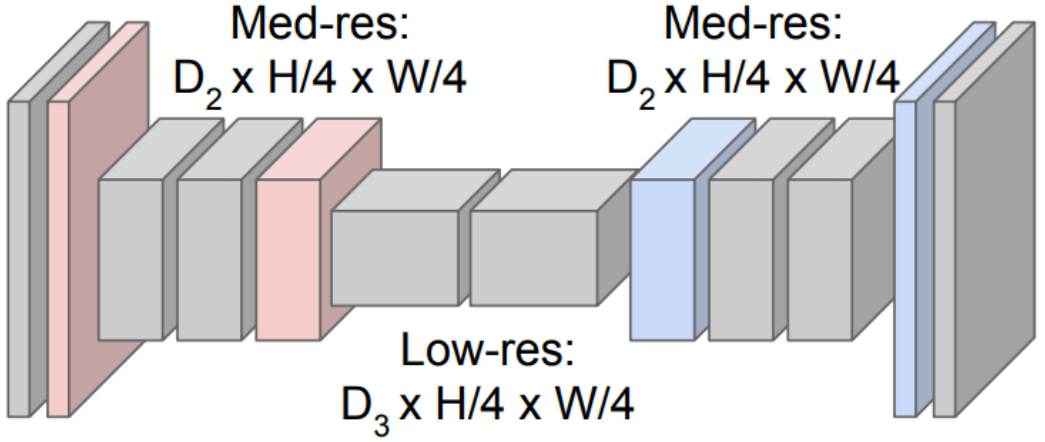


What's the problem?

Fully convolutional

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

Input:
3 x H x W

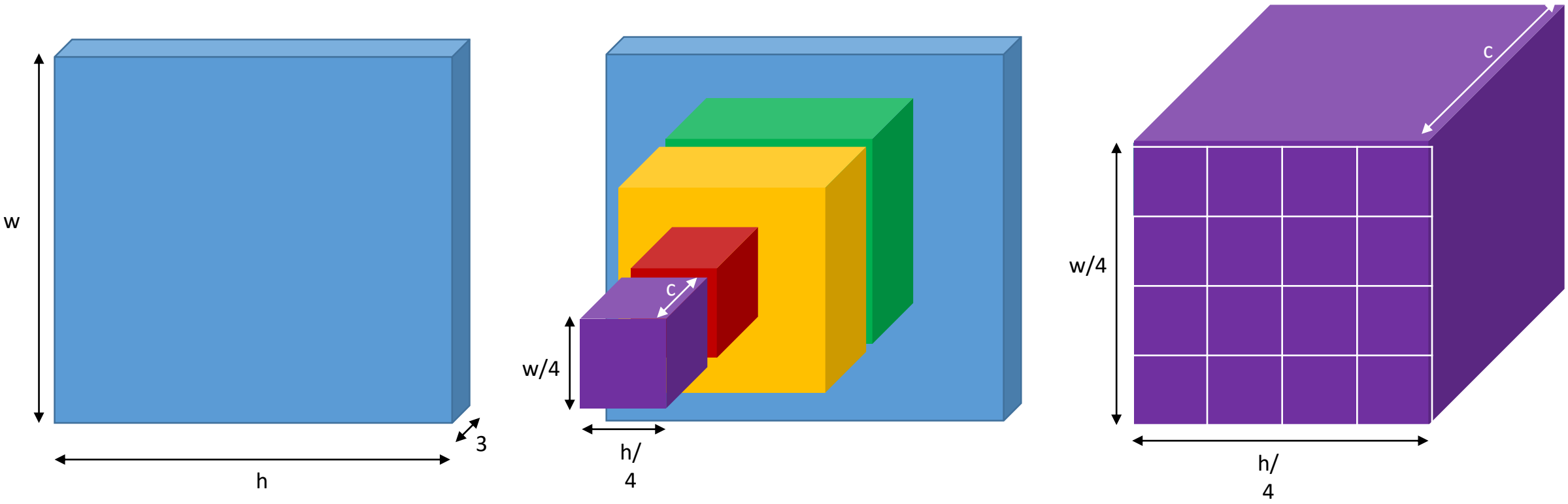


High-res:
D_1 x H/2 x W/2

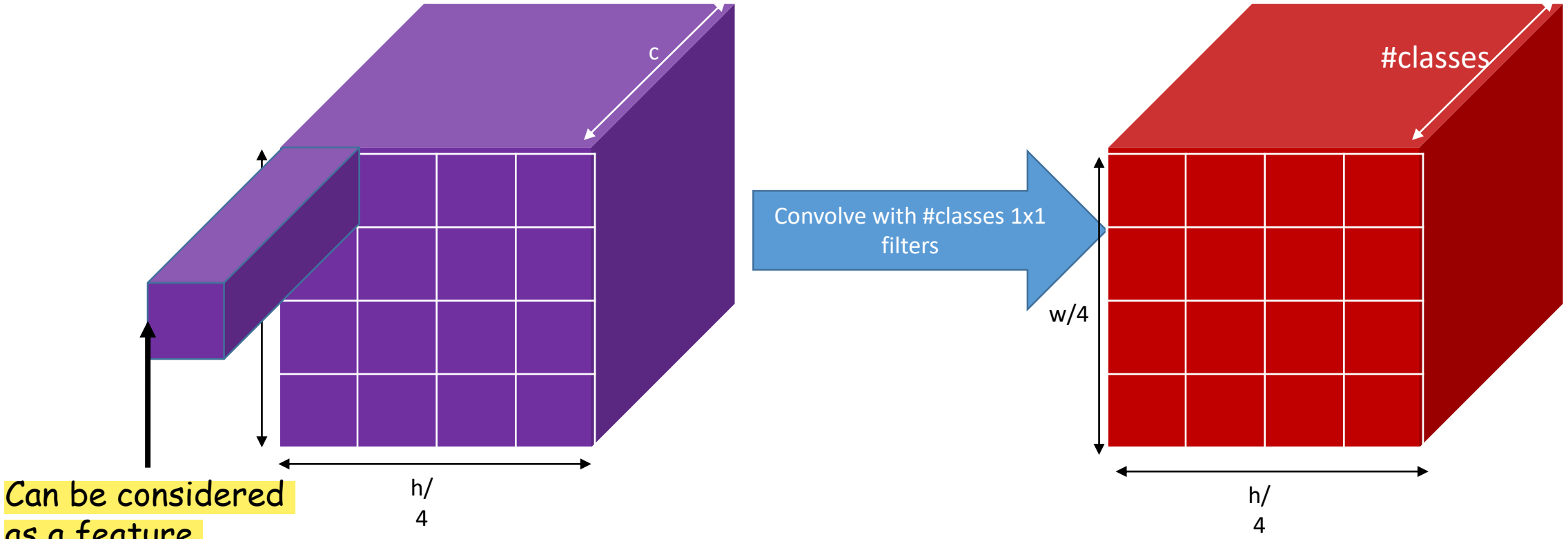
High-res:
D_1 x H/2 x W/2

Predictions:
H x W

Semantic segmentation using convolutional networks



Semantic segmentation using convolutional networks



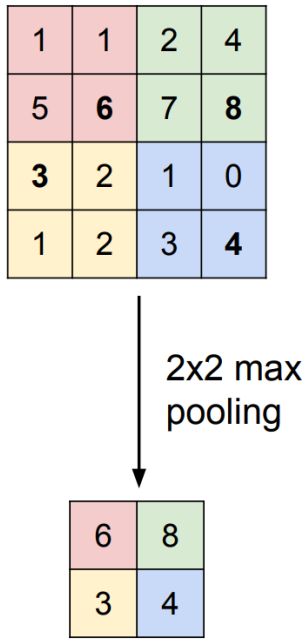
Can be considered as a feature vector for a pixel

Semantic segmentation using convolutional networks

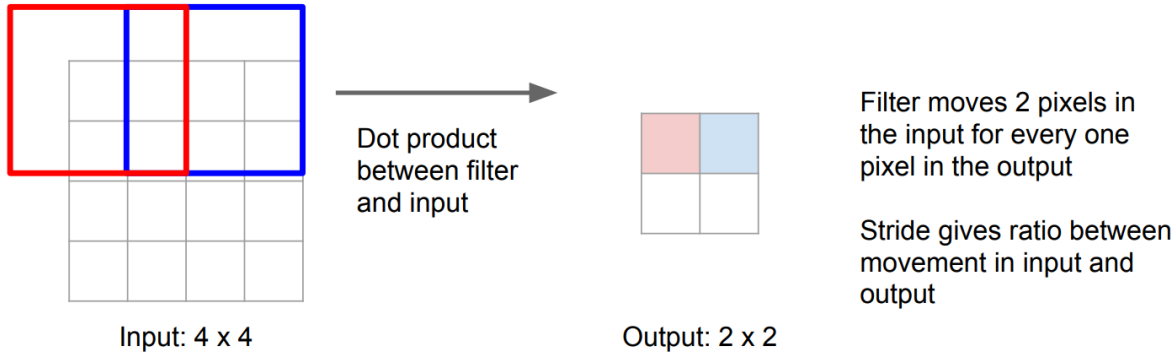
- Creates a feature map, then downsample it and so on…
- The convolution is not on full size image.
- All layers are Convolutional, no fully connected layer at the end.
- The output size of the last layer will be the same size as the input image.
- The **number of channels** is according to the **number of classes** we want to classify (car, road)
- Training is the same (as in previous methods) by cross-entropy loss for every pixel.

Downsampling

Pooling



Strided Convolution



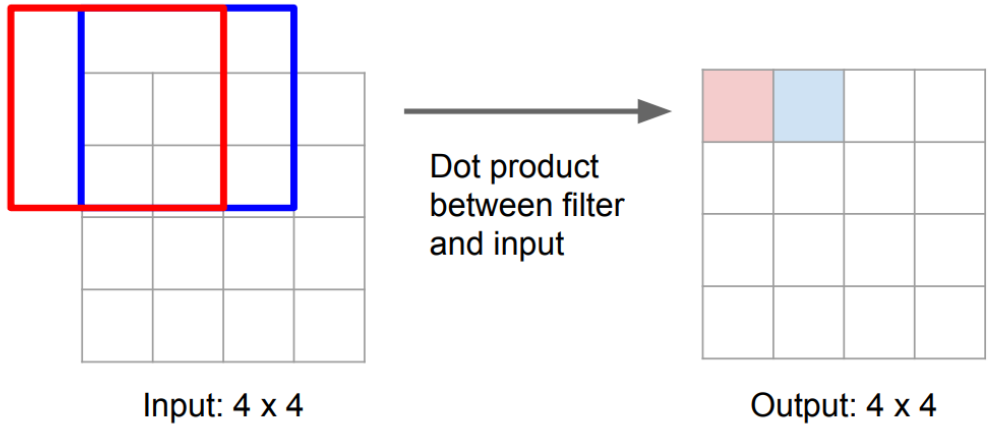
Move 2 pixels

Recall: Strided convolution

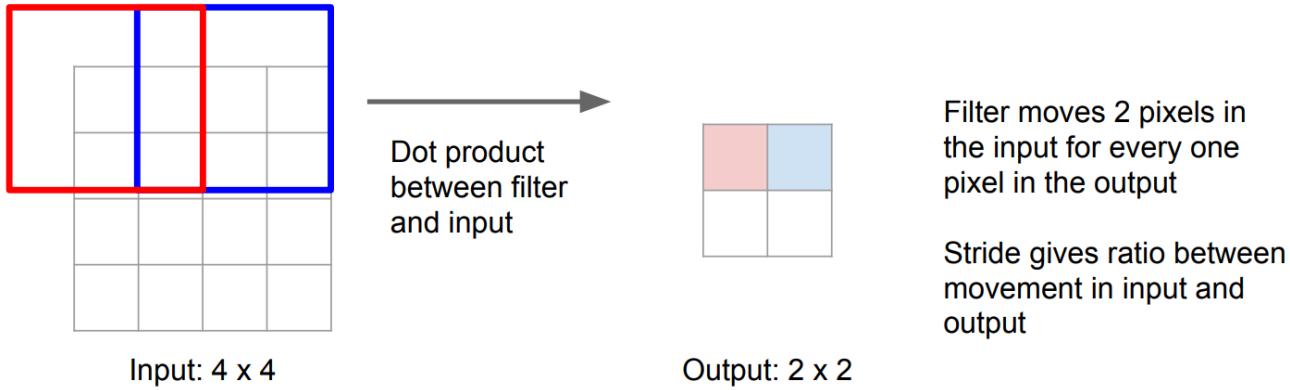
Normal convolution: 3x3, pad 1, Stride 1

$$d = \frac{n+2p-f}{s}+1$$

f – filter size
p – padding
n – input image size
s – stride
d – output image size

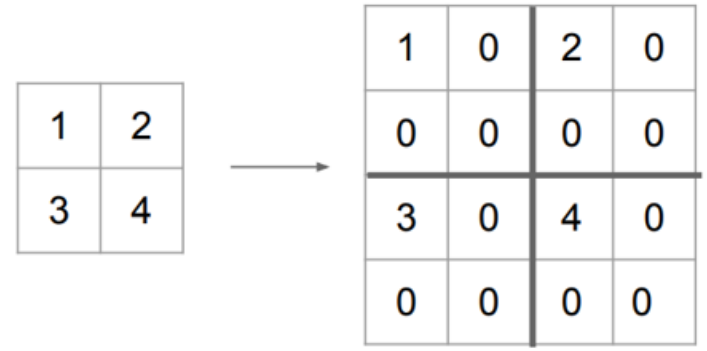


Normal convolution: 3x3, pad 1, Stride 2

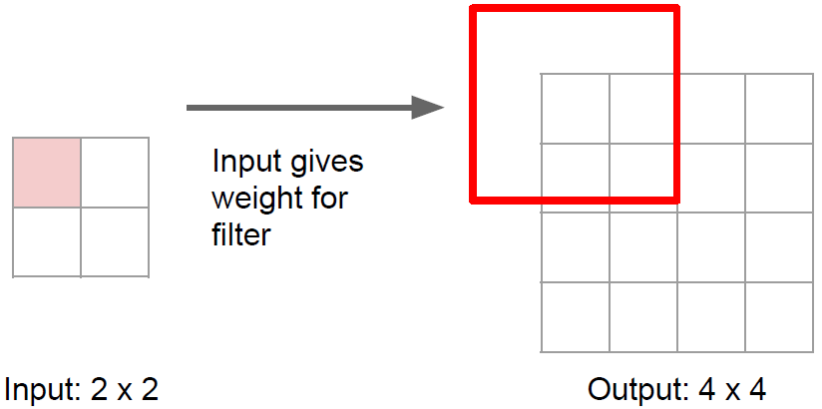


How Does Upsampling Work?

UpPooling

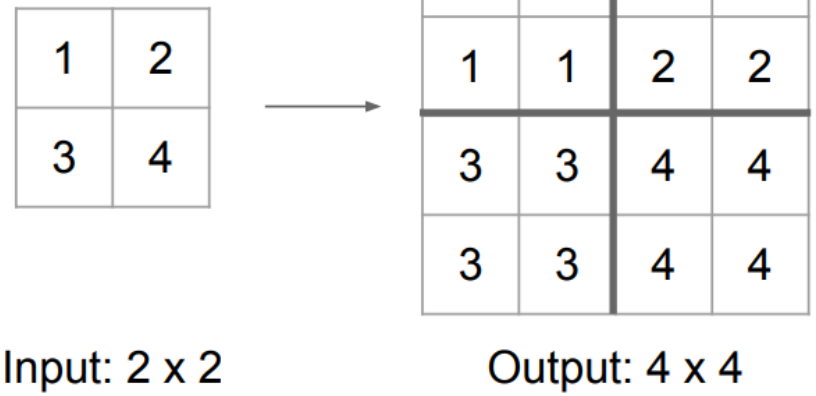


Transpose Convolution



Uppooling Methods

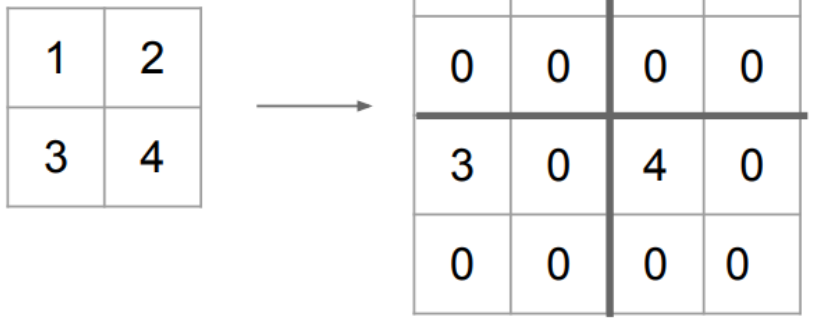
Nearest Neighbor



Input: 2 x 2

Output: 4 x 4

“Bed of Nails”



Input: 2 x 2

Output: 4 x 4

Uppooling Methods

Max Pooling
Remember which element was max!

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

Input: 4 x 4

5	6
7	8

Output: 2 x 2

Rest of the network

Max Unpooling
Use positions from pooling layer

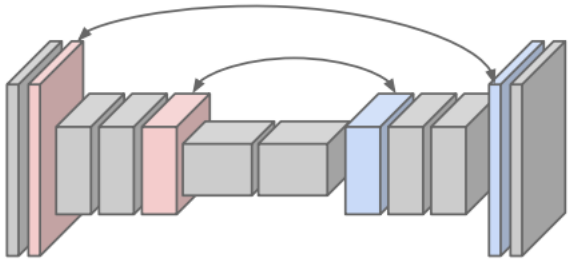
1	2
3	4

Input: 2 x 2

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

Output: 4 x 4

Corresponding pairs of downsampling and upsampling layers

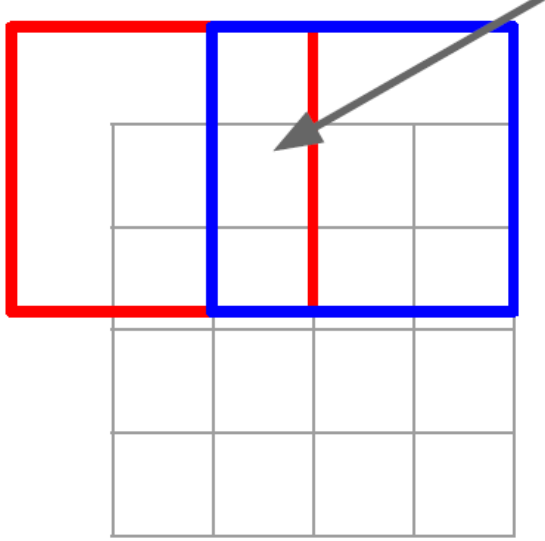


Transpose Convolution

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

3 x 3 **transpose** convolution, stride 2 pad 1

Input gives weight for filter



Sum where output overlaps

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

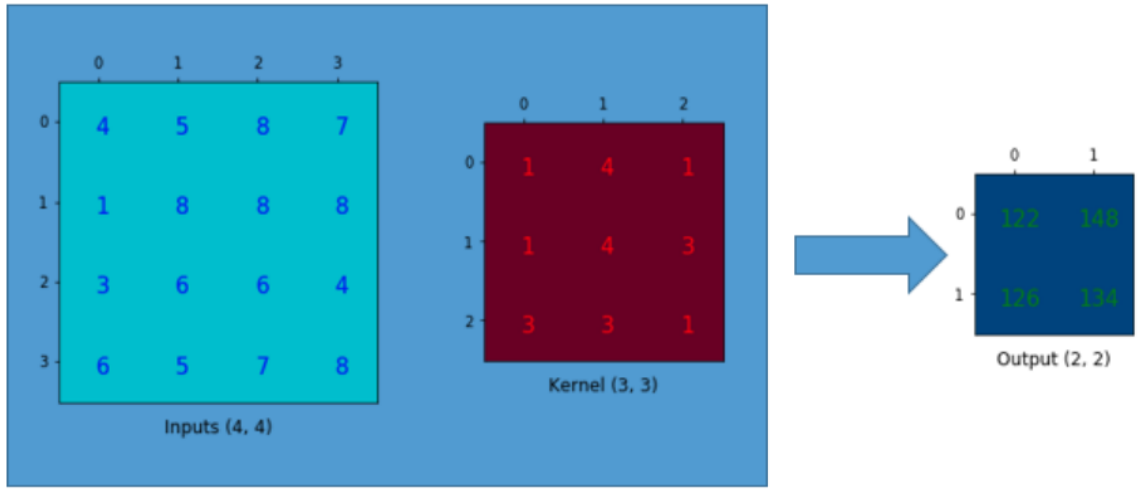
Stride gives ratio between movement in output and input

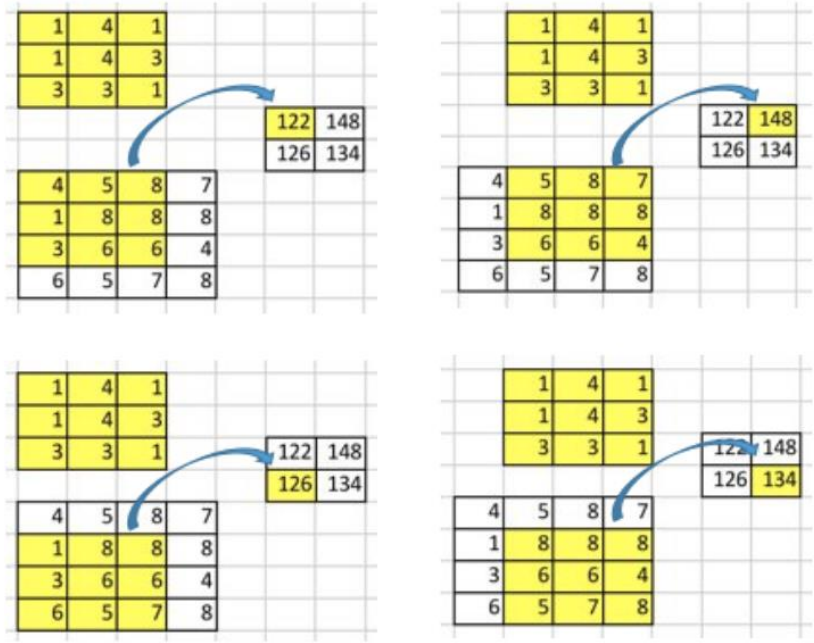
Input: 2 x 2

Output: 4 x 4

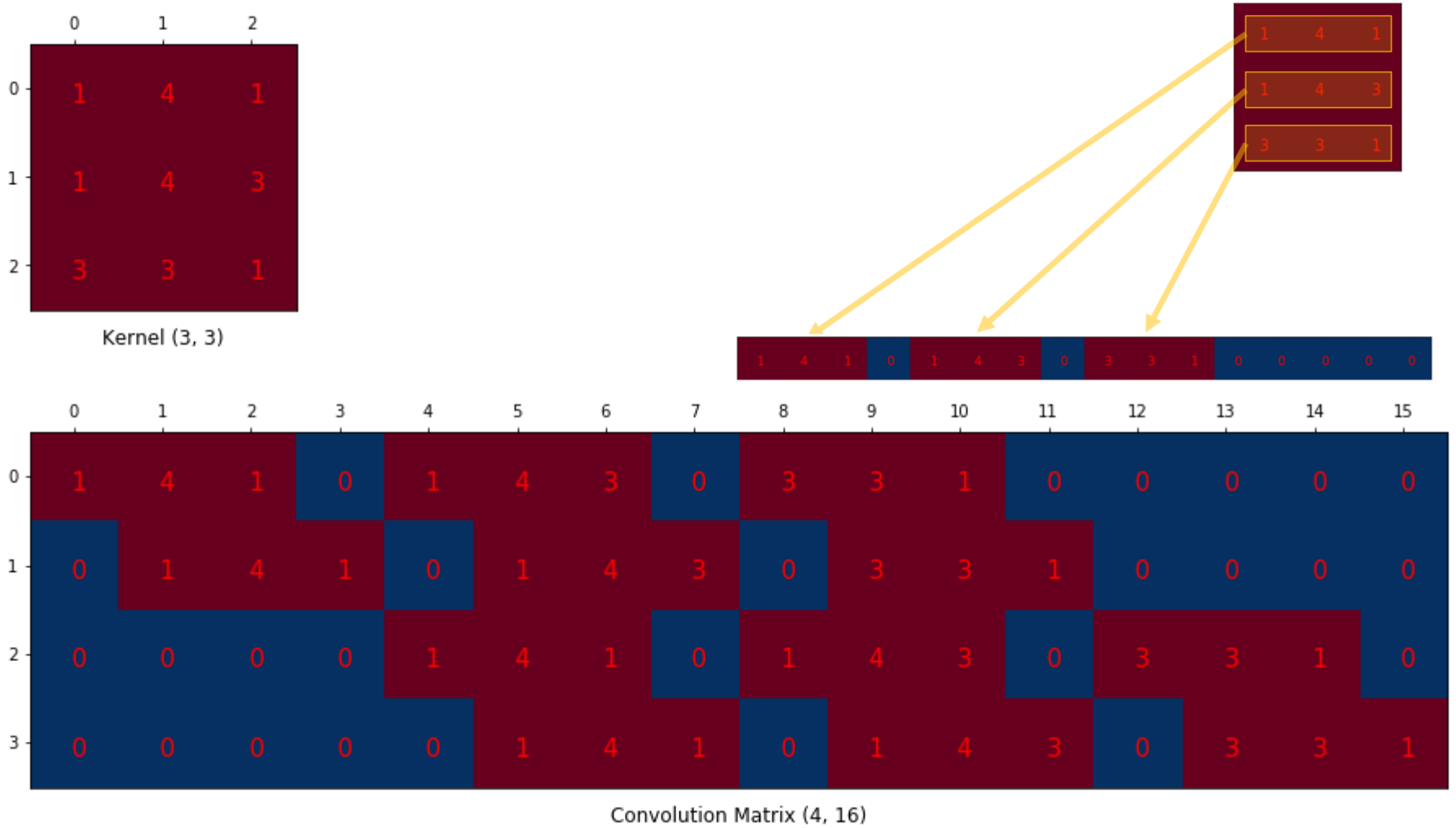
Weighting?

Convolution Matrix

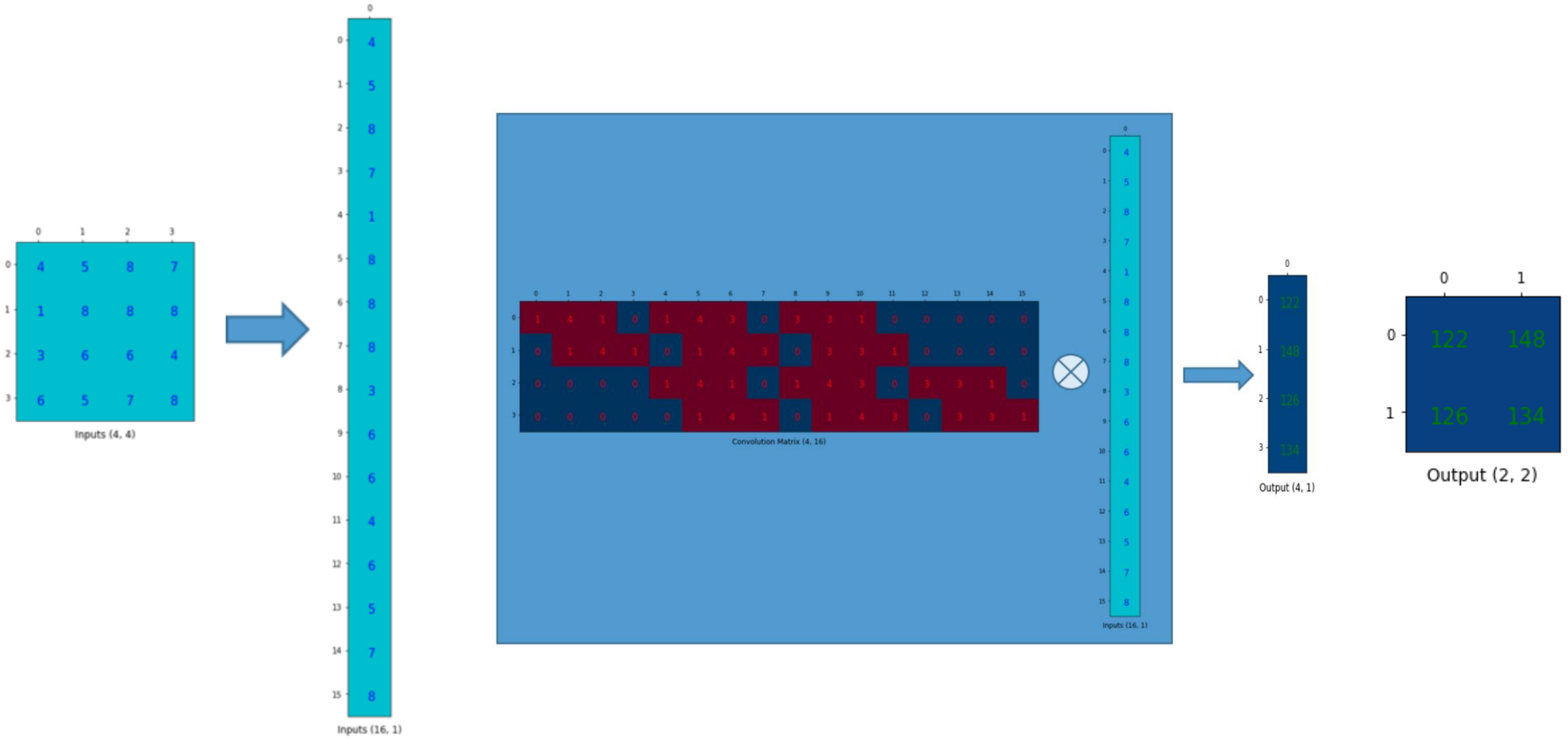




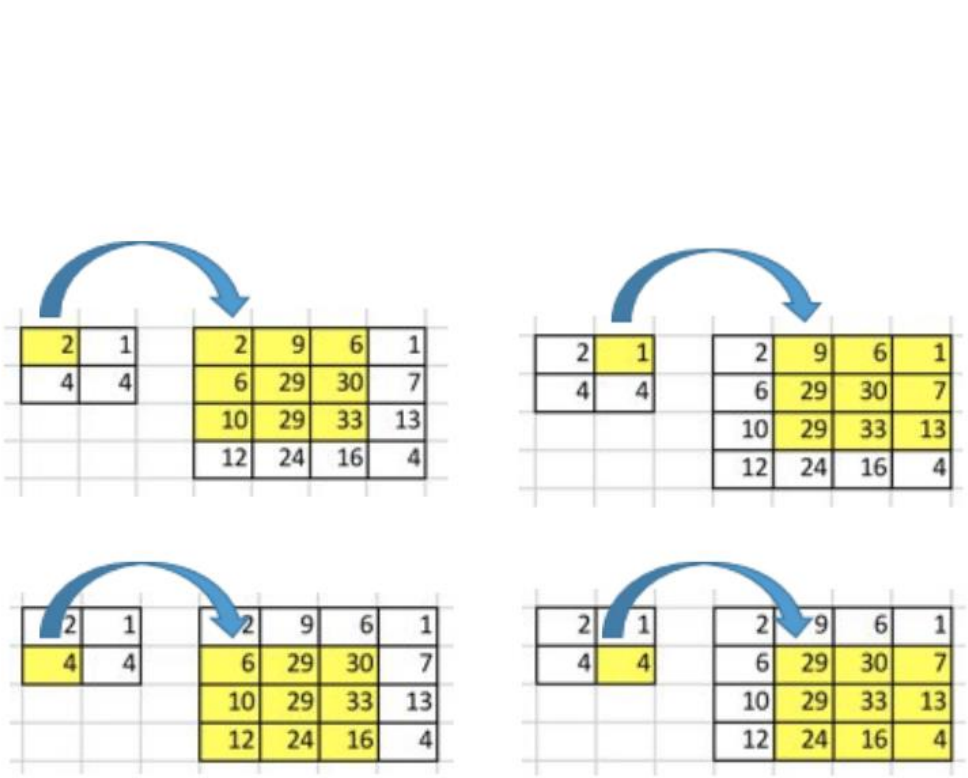
Convolution Matrix

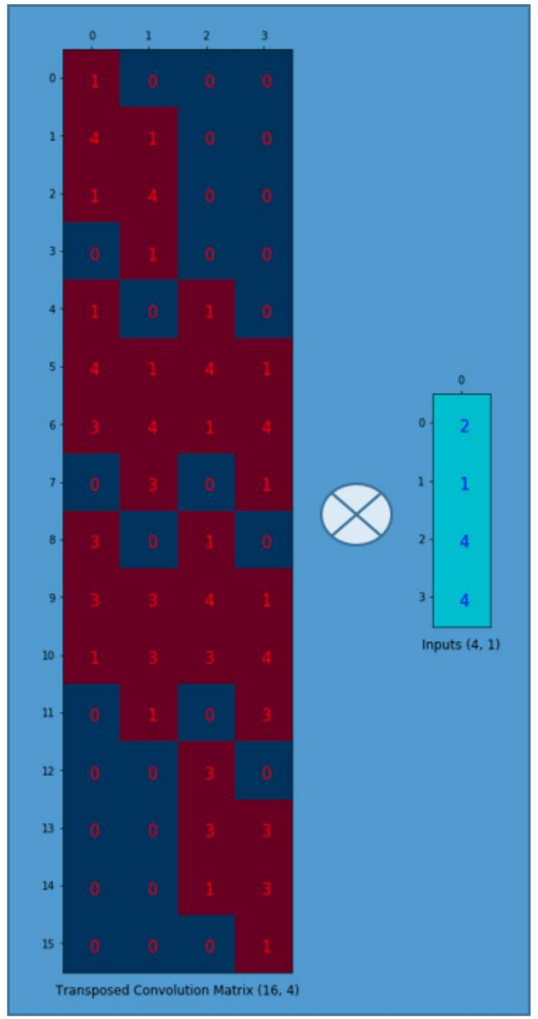


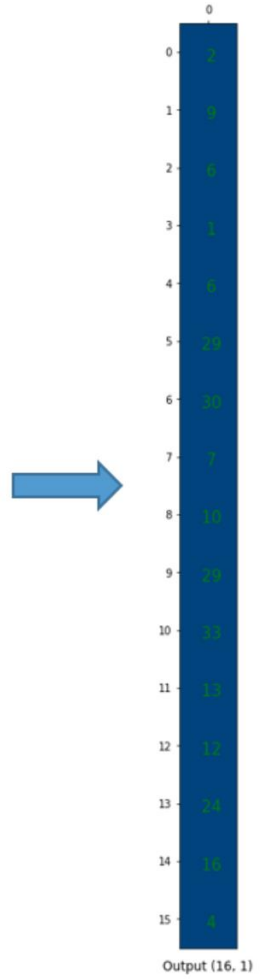
Convolution Matrix

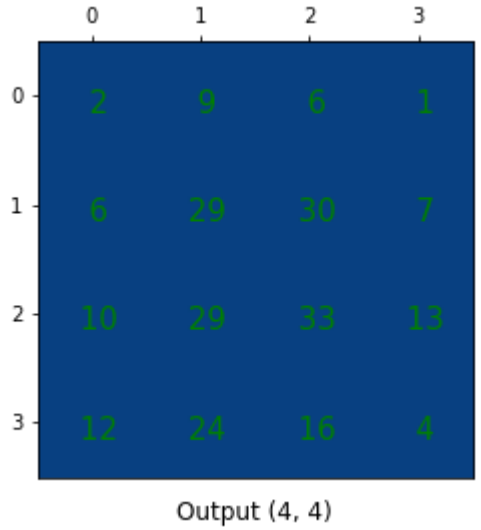


Transposed Convolution Matrix





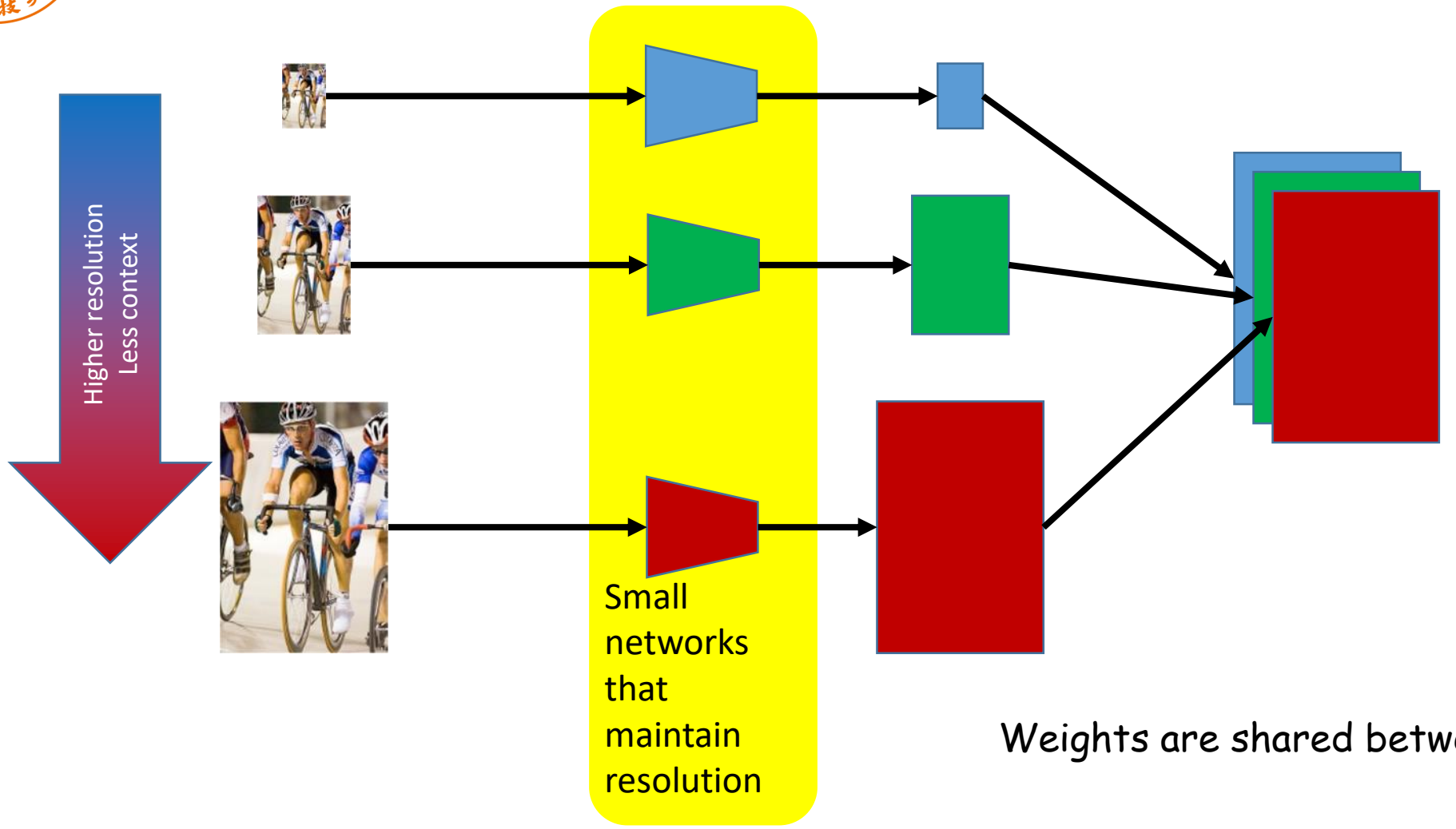




The resolution issue

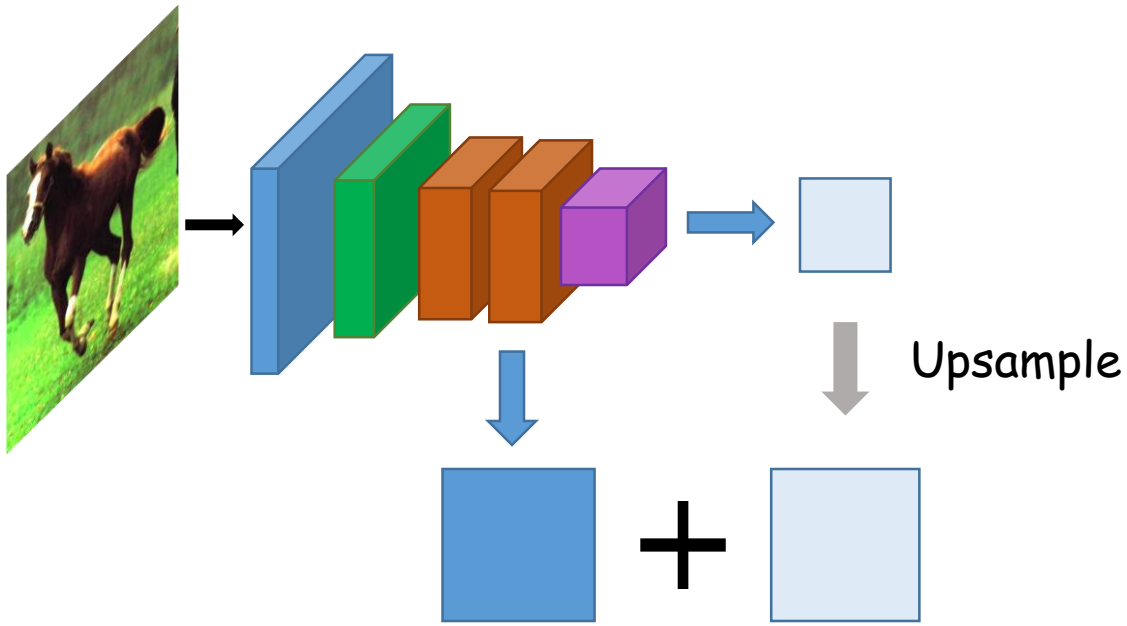
- Problem: Need fine details!
- Shallower network / earlier layers?
  - Deeper networks work better: more abstract concepts
  - Shallower network => Not very semantic!
- Remove subsampling?
  - Subsampling allows later layers to capture larger and larger patterns
  - Without subsampling => Looks at only a small window!

Solution 1: Image pyramids



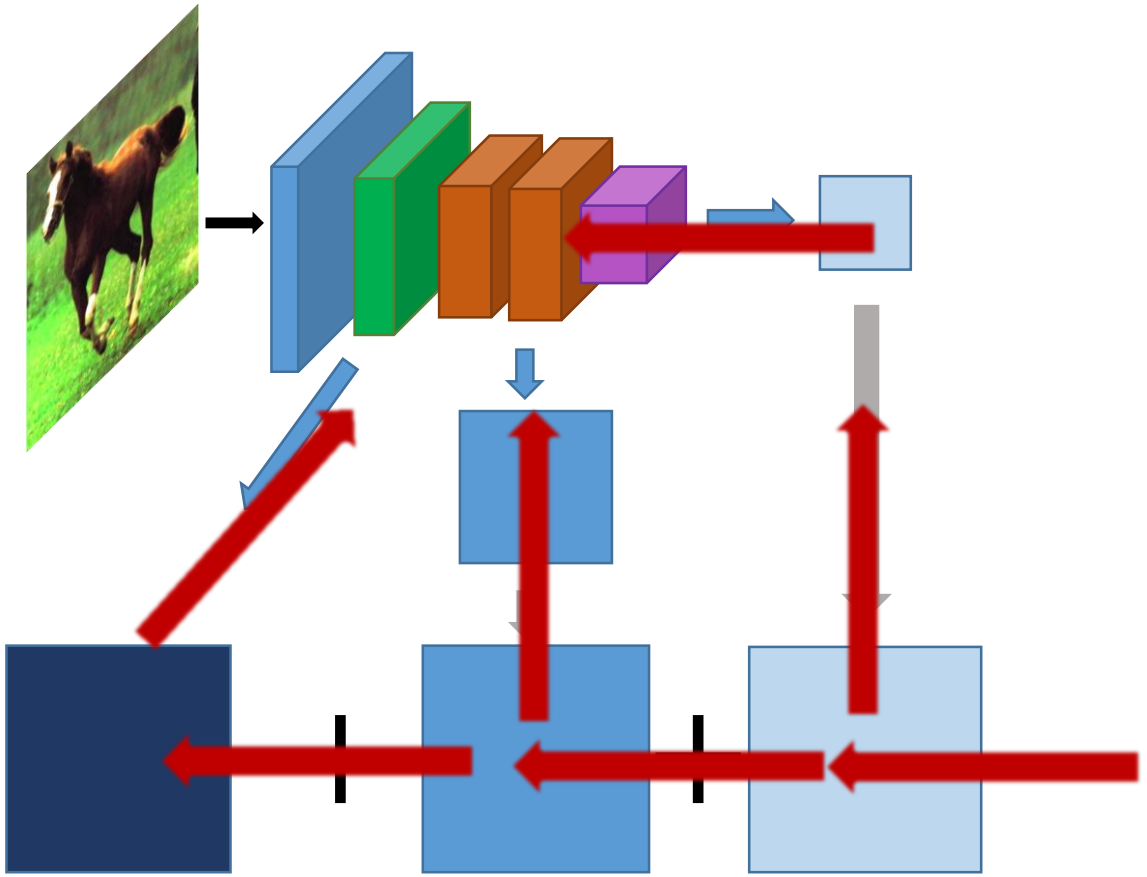
Learning Hierarchical Features for Scene Labeling. Clement Farabet, Camille Couprie, Laurent Najman, Yann LeCun. In *TPAMI,* 2013.

Solution 2: Skip connections



Compute class scores at multiple layers, then upsample and add

Solution 2: Skip connections



Red arrows indicate backpropagation

Solution 2: Skip connections

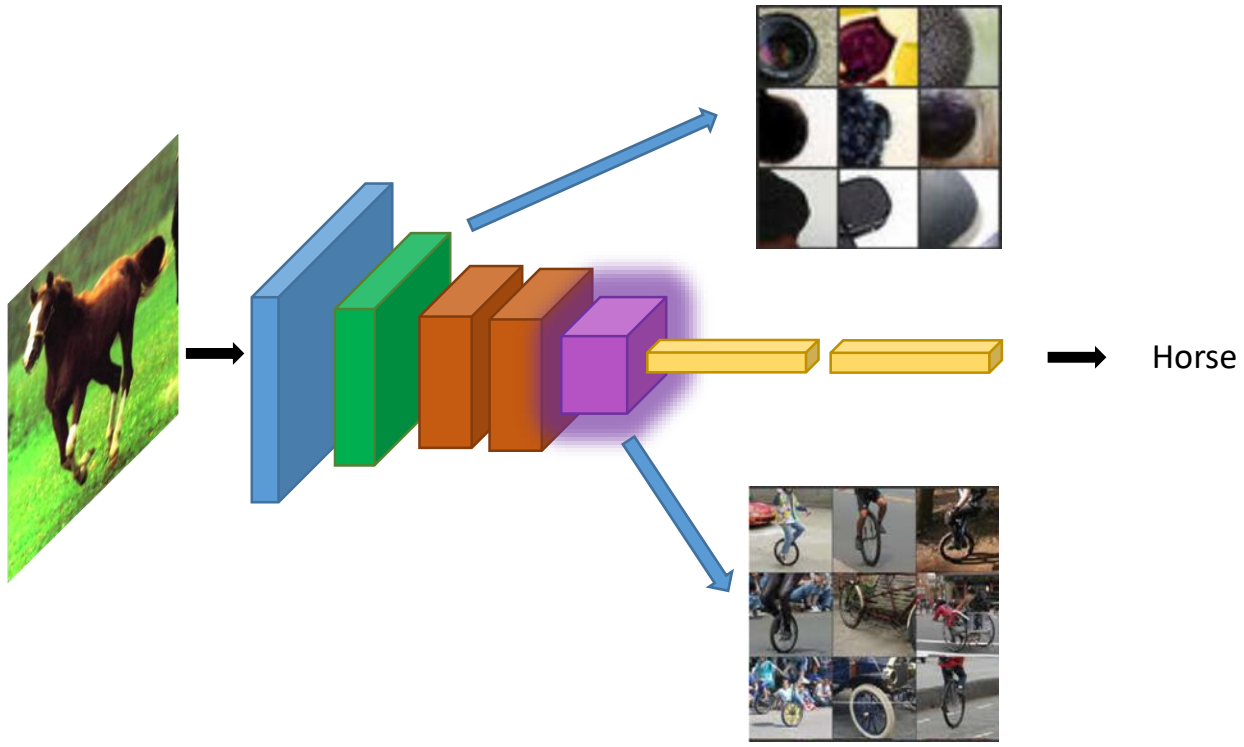
without skip

with skip

Fully convolutional networks for semantic segmentation. Evan Shelhamer, Jon Long, Trevor Darrell. In *CVPR* 2015

Solution 2: Skip connections

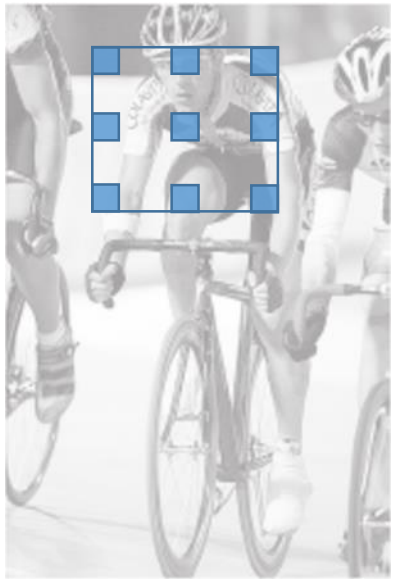
- Problem: early layers not semantic

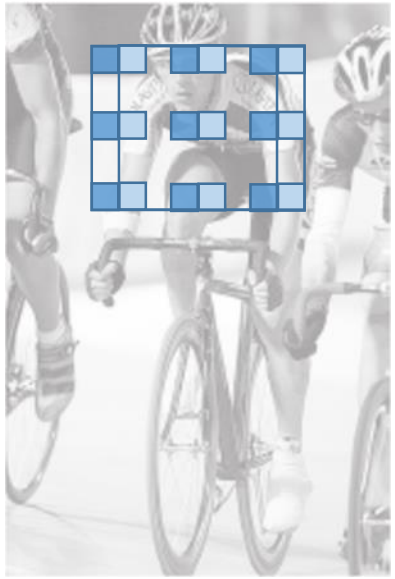


Visualizations from : M. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. In *ECCV* 2014.

Solution 3: Dilation

- Need subsampling to allow convolutional layers to capture large regions with small filters
  - Can we do this without subsampling?





$$(F * k)(\mathbf{p}) = \sum_{\mathbf{s}+\mathbf{t}=\mathbf{p}} F(\mathbf{s})\, k(\mathbf{t}). \quad (F *_l k)(\mathbf{p}) = \sum_{\mathbf{s}+l\mathbf{t}=\mathbf{p}} F(\mathbf{s})\, k(\mathbf{t}).$$

Standard Convolution (Left)
Dilated Convolution (Right)

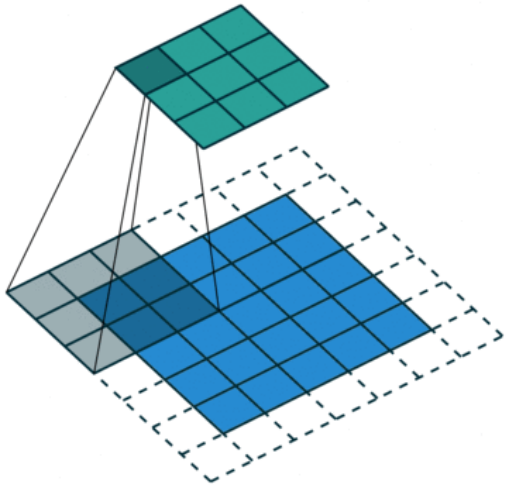
When l=1, it is standard convolution.
When l>1, it is dilated convolution.

Solution 3: Dilation

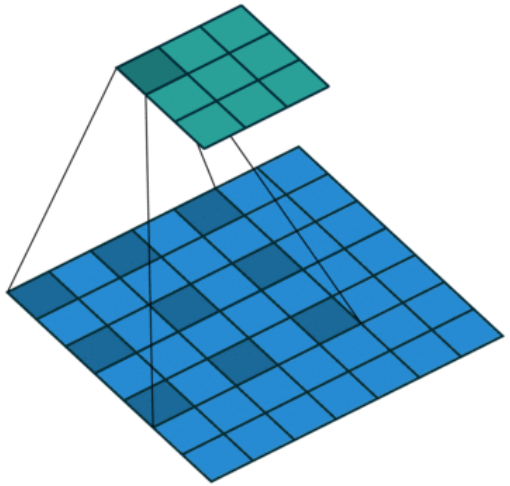
- Instead of subsampling by factor of 2: dilate by factor of 2
- Dilation can be seen as:
  - Using a much larger filter, but with most entries set to 0
  - Taking a small filter and "exploding"/ "dilating" it
- Not panacea: without subsampling, feature maps are much larger: memory issues
- The idea of Dilated Convolution is come from the wavelet decomposition. It is also called "**atrous convolution**", "**algorithme àtrous**" and "**hole algorithm**". Thus, any ideas from the past are still useful if we can turn them into the deep learning framework.

Solution 3: Dilation

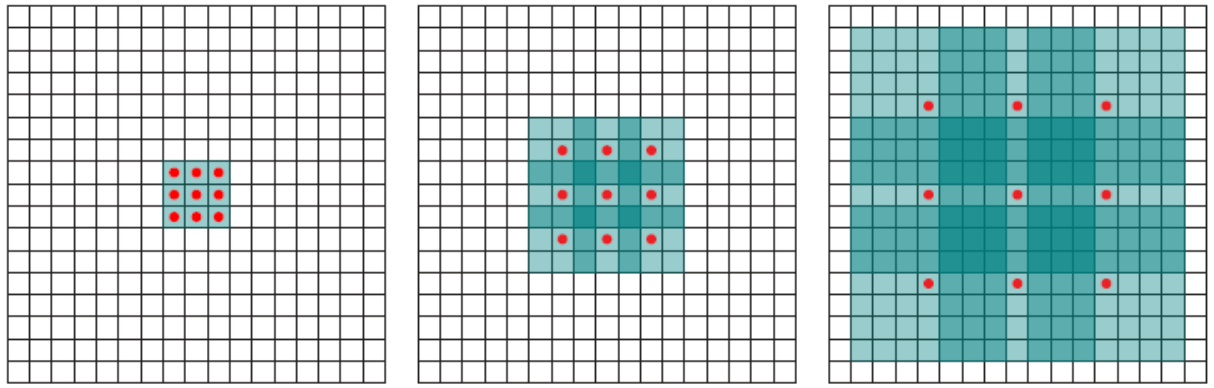
- The figures illustrate an example of **dilated convolution when *l*=2**. We can see that **the receptive field is larger** compared with the standard one.



Standard Convolution (l=1)

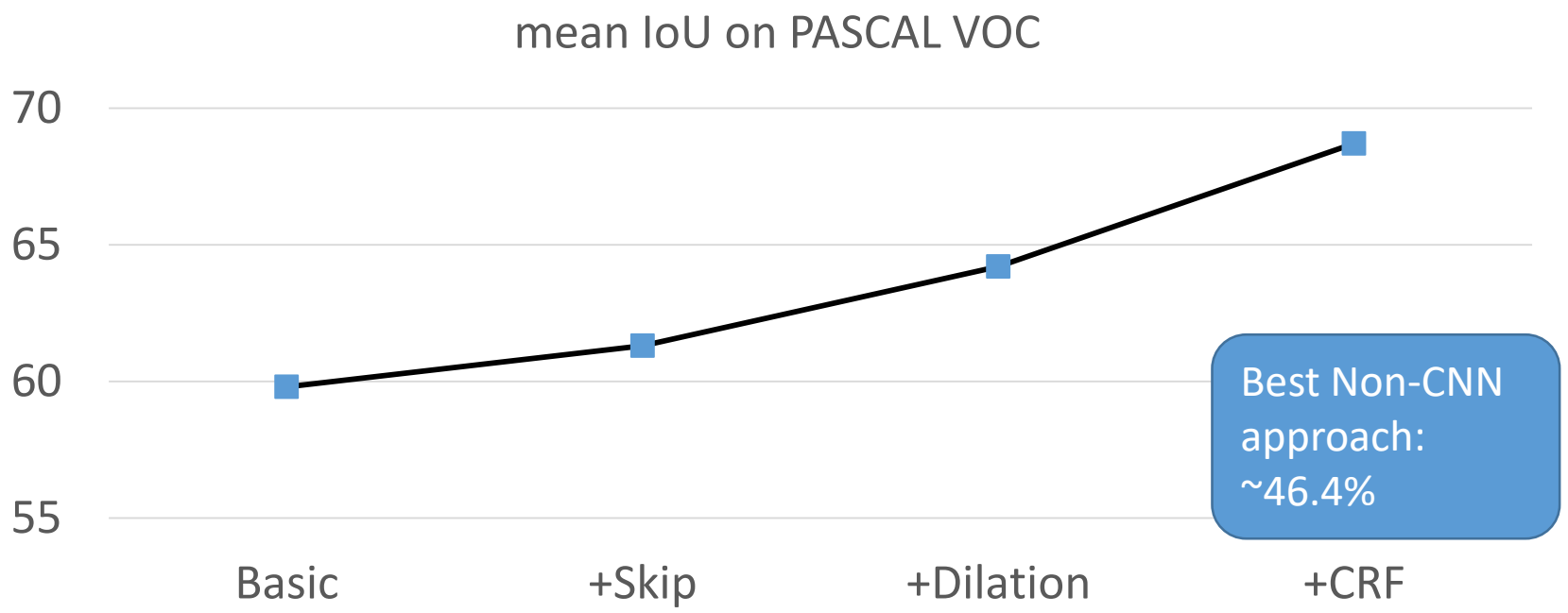


Dilated Convolution (l=2)



l=1 (left), l=2 (Middle), l=4 (Right)

Putting it all together



Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan Yuille. In *ICLR,* 2015.

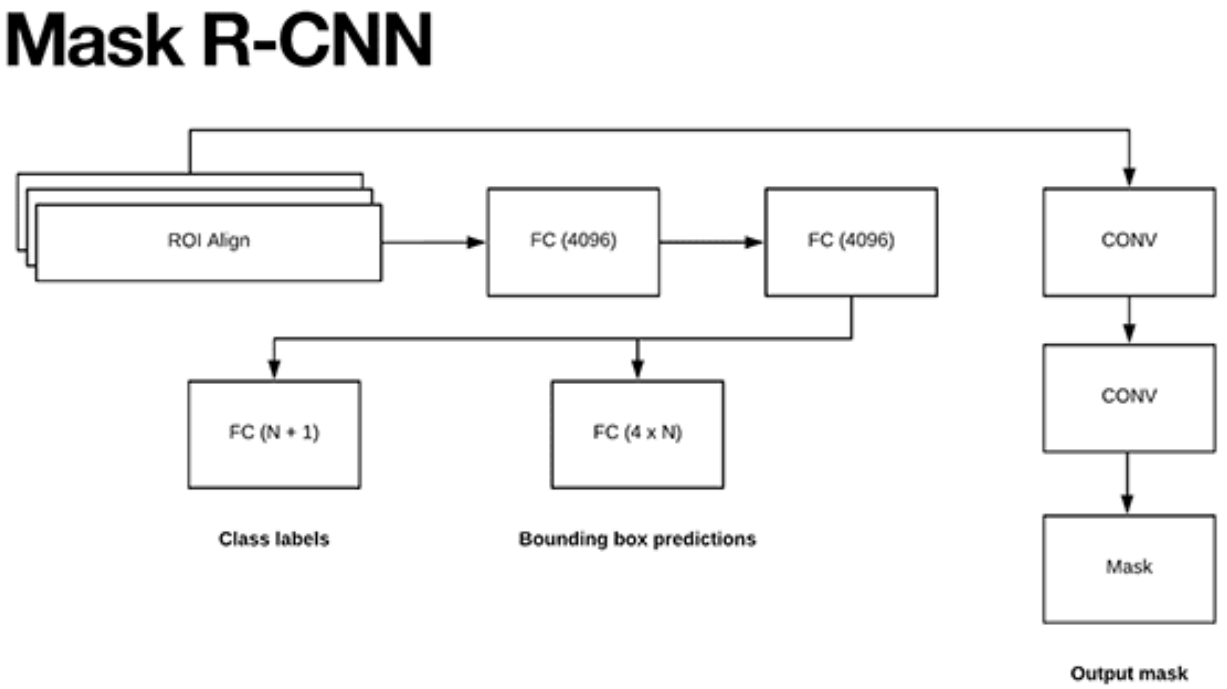
Other additions

Method	mean IoU (%)
VGG16 + Skip + Dilation	65.8
ResNet101	68.7
ResNet101 + Pyramid	71.3
ResNet101 + Pyramid + COCO	74.9

DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan Yuille. Arxiv 2016.

Mask R-CNN

- Faster R-CNN
  - **Region Proposal Network (RPN)** <-- Selective Search algorithm
- Mask R-CNN
  - Replacing the ROI Pooling module with a more accurate ROI Align module
  - Inserting an additional branch out of the ROI Align module

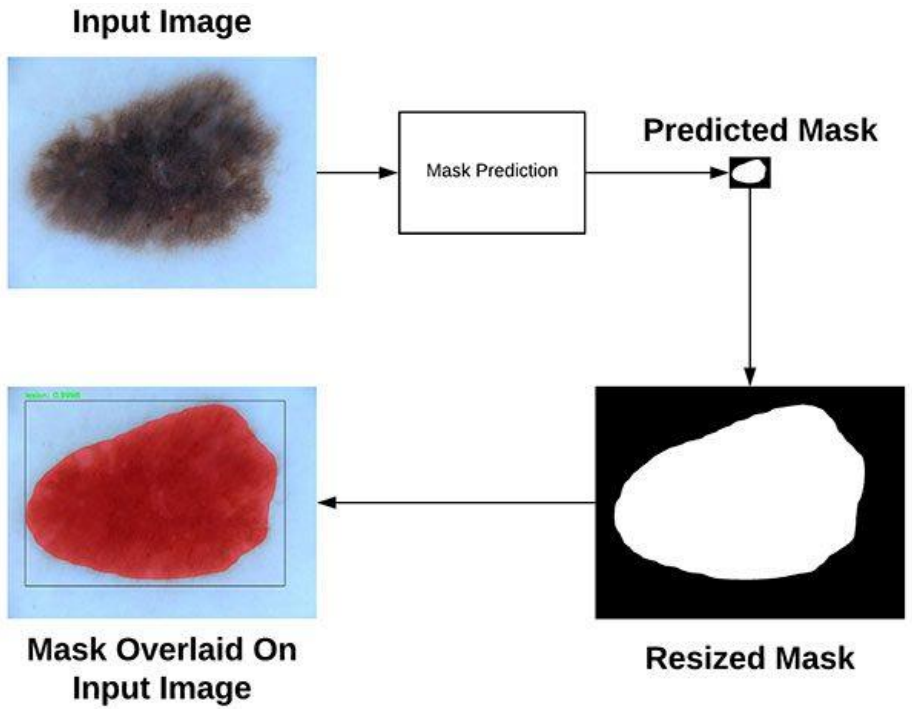


Mask R-CNN: Kaiming He, Georgia Gkioxari, Piotr Dollar and Ross Girshick
https://arxiv.org/pdf/1703.06870.pdf

Mask R-CNN

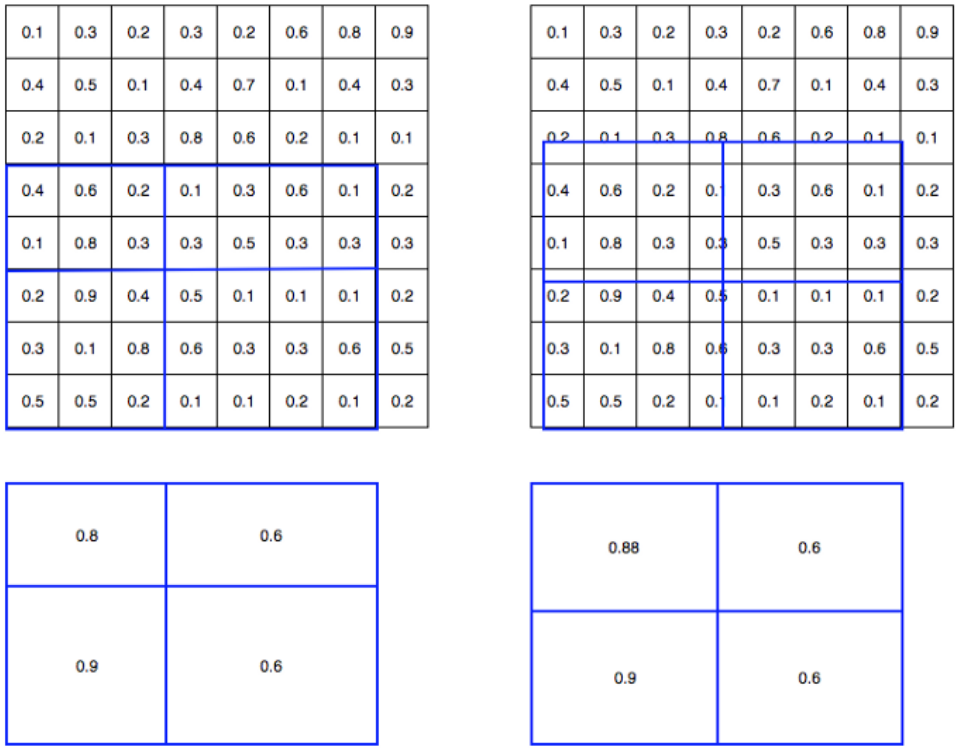
- Each of the 300 selected ROIs go through three parallel branches of the network:
  - Label prediction
  - Bounding box prediction
  - Mask prediction
- A multi-task loss
  - Average binary cross-entropy loss
  - Per pixel: K classes

$$L = L_{cls} + L_{box} + L_{mask}$$



Mask R-CNN

- **ROI Align**
  - Removes this quantization which is causes this misalignment
  - For each bin, you regularly sample 4 locations and do bilinear interpolation
  - Result are not sensitive to exact sampling location or the number of samples



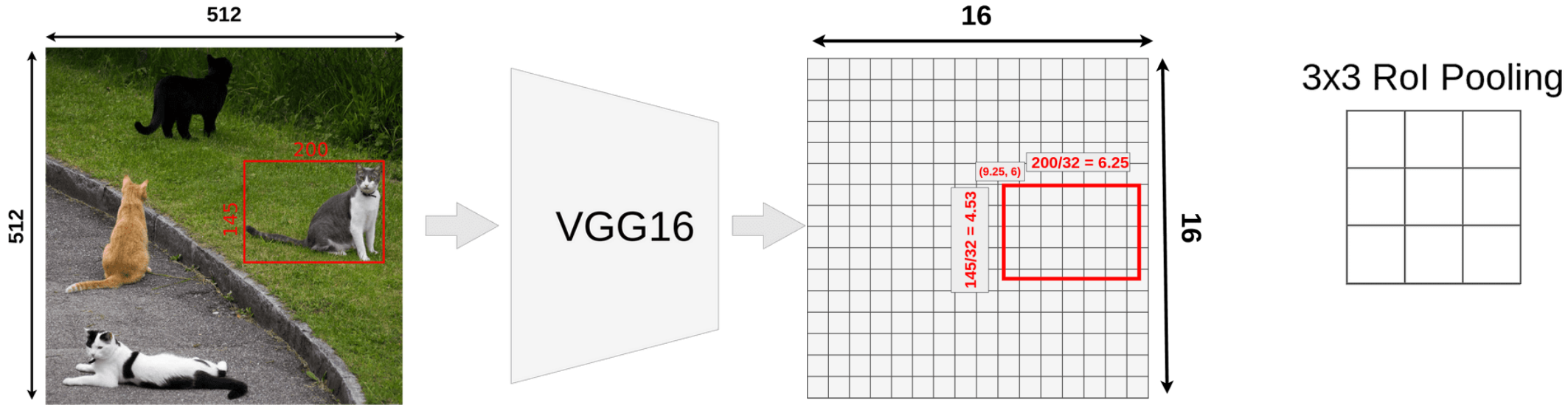
	AP	AP_{50}	AP_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
RoIPool	23.6	46.5	21.6	28.2	52.7	26.9
RoIAlign	**30.9**	**51.8**	**32.1**	**34.0**	**55.3**	**36.4**
	+7.3	*+ 5.3*	*+10.5*	*+5.8*	*+2.6*	*+9.5*

https://erdem.pl/2020/02/understanding-region-of-interest-part-2-ro-i-align

ROI Align

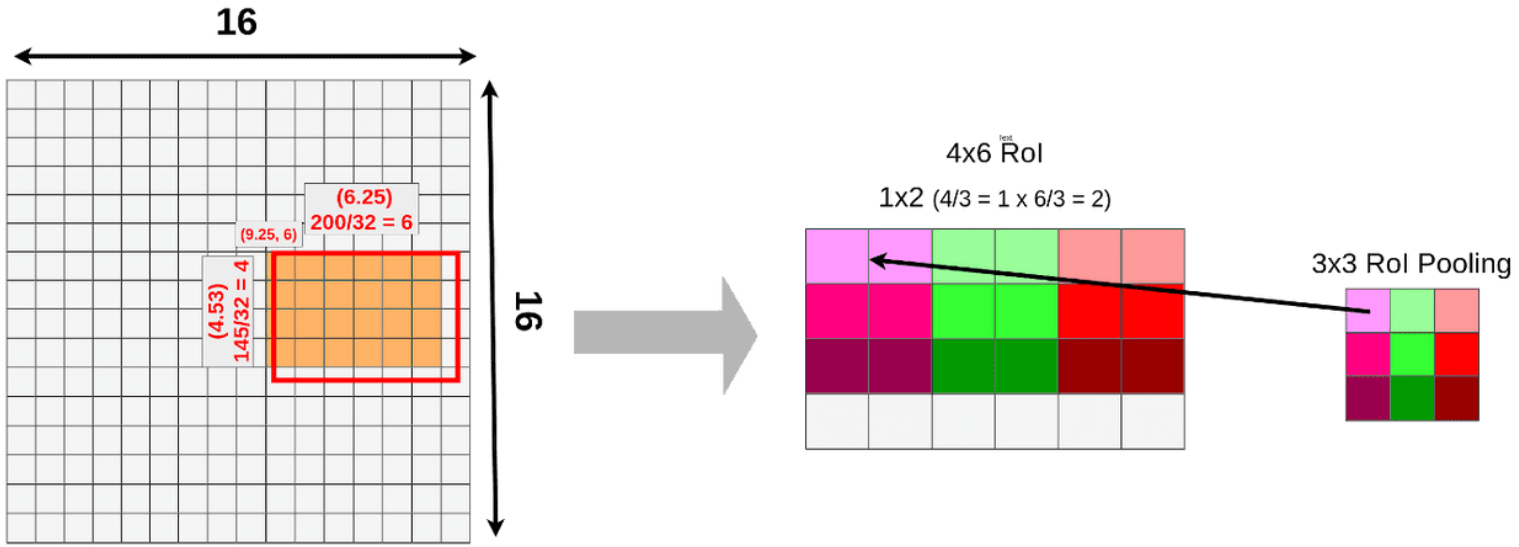
- An image input of size 512x512x3
- A 16x16x512 feature map
- The proposed RoIs (145x200 box)
- 3*3 RoI Pooling

- (9.25, 6) – top left corner
- 6.25 – width
- 4.53 – height



ROI Align

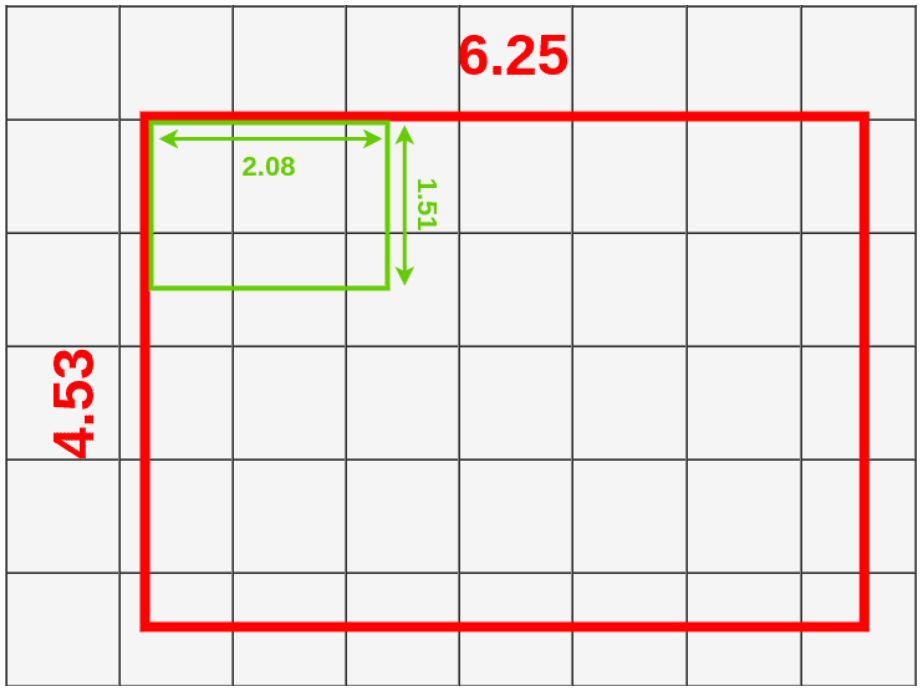
- The main difference between RoI Pooling and RoI Align is quantization (applying quantization twice)
  - First time in the mapping process
  - Second time during the pooling process



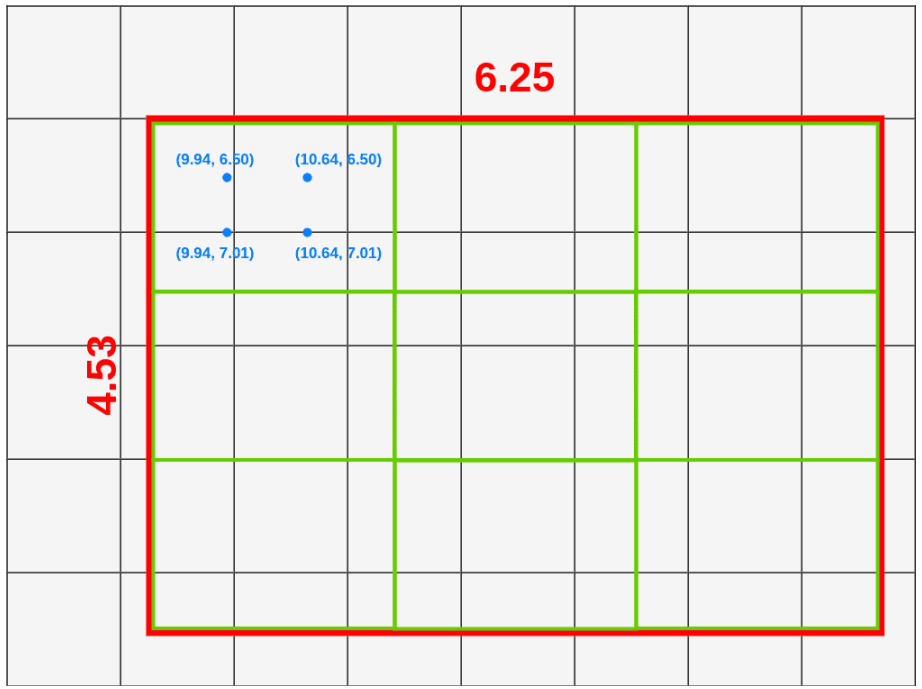
Quantization when mapping and pooling

ROI Align

- RoI Align is not using quantization for data pooling



1. Divide mapped RoI (6.25x4.53) by 3

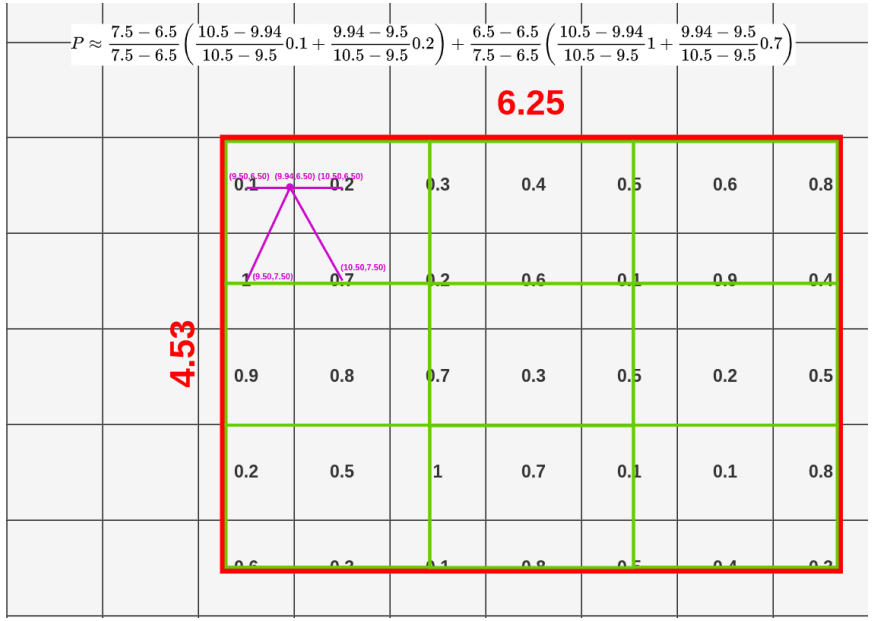


2. Create **four sampling points** inside that box (divide box by 3)

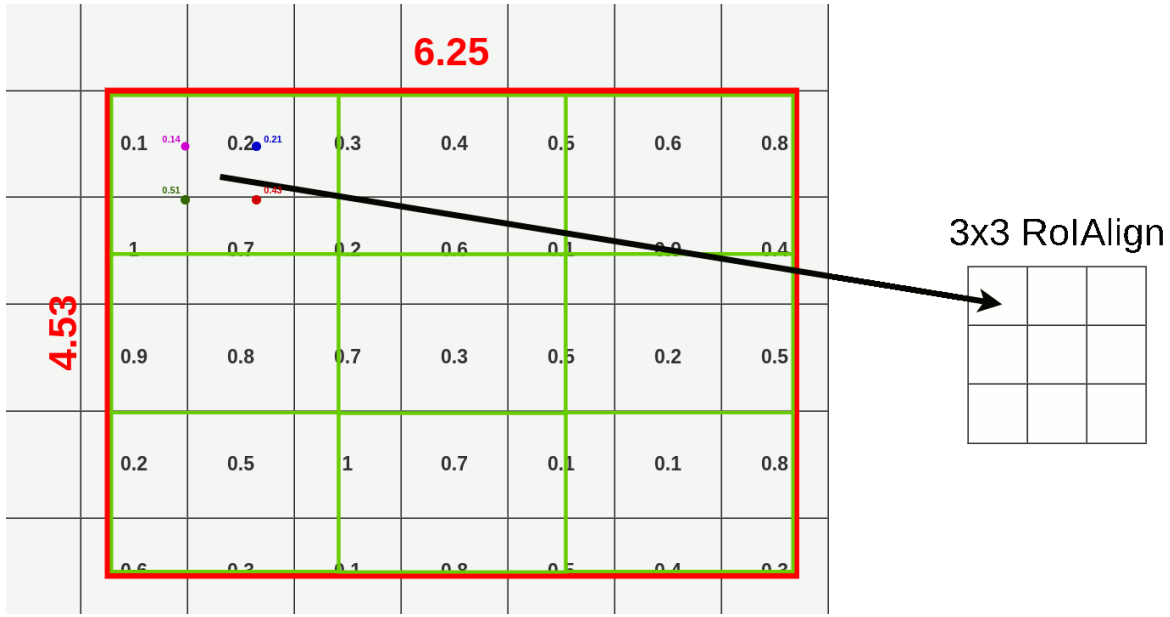
ROI Align

- RoI Align is not using quantization for data pooling

$$P \approx \frac{y_2 - y}{y_2 - y_1}\left(\frac{x_2 - x}{x_2 - x_1}Q_{11} + \frac{x - x_1}{x_2 - x_1}Q_{21}\right) + \frac{y - y_1}{y_2 - y_1}\left(\frac{x_2 - x}{x_2 - x_1}Q_{12} + \frac{x - x_1}{x_2 - x_1}Q_{22}\right)$$

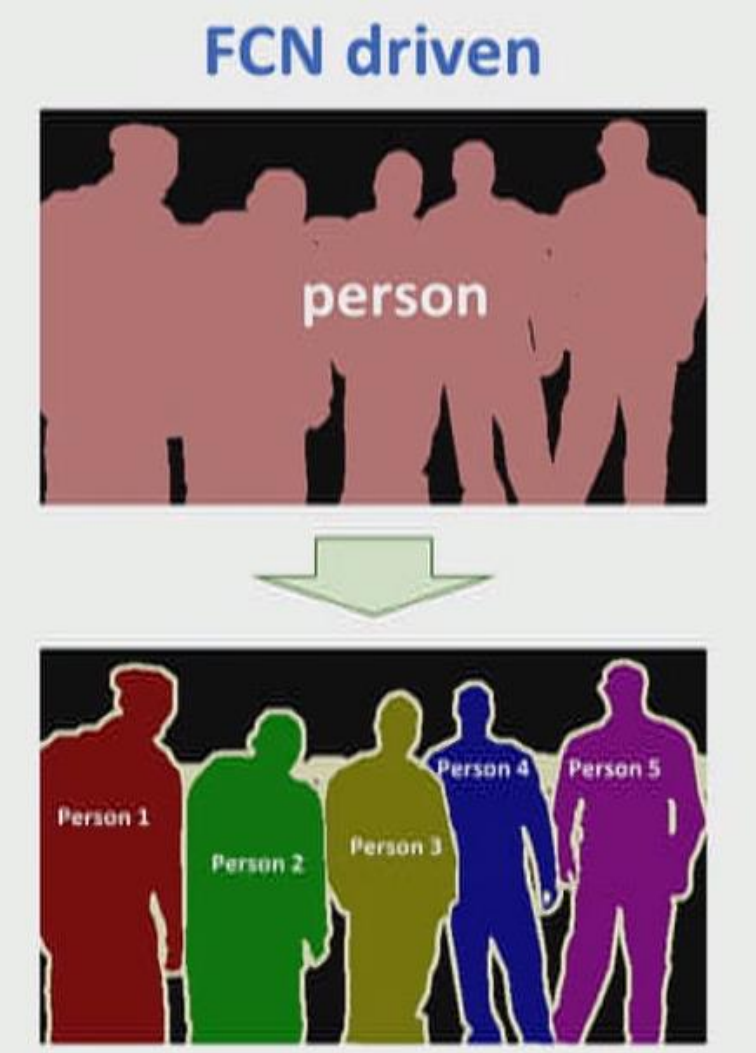


3. Bilinear Interpolation equation



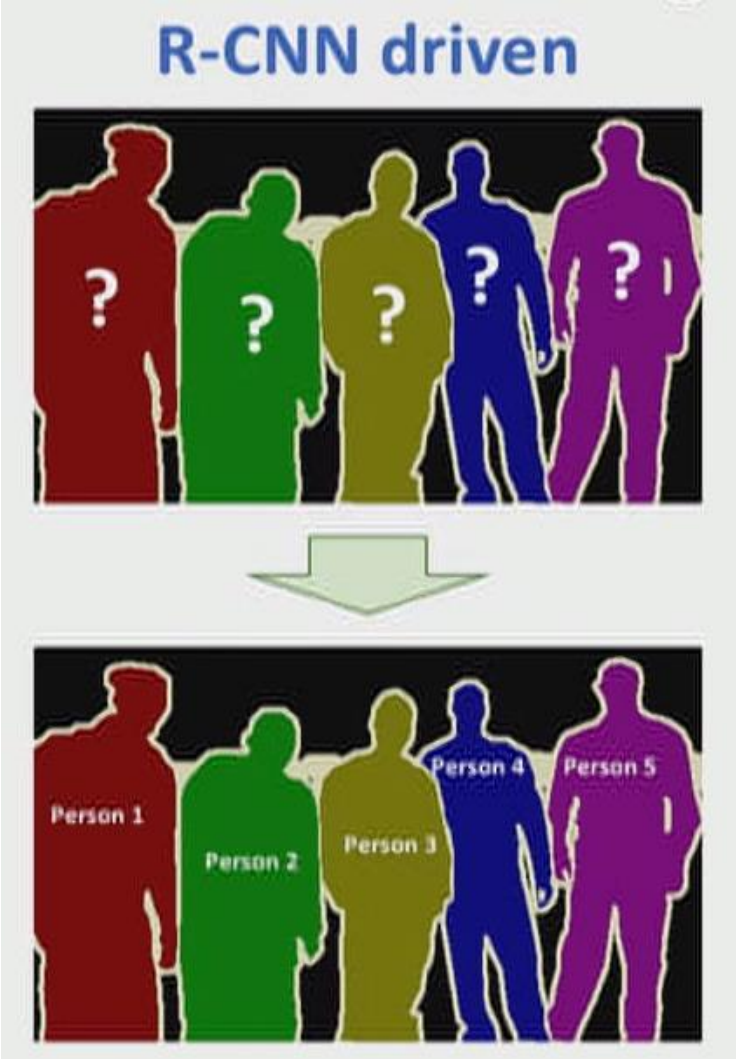
4. RoIAlign pooling process

Instance Segmentation Methods



FCN driven methods:
start from a segmentation result,
then learn how divide the results into individual instances.

Instance Segmentation Methods



RCNN driven methods:
start from segmentation level proposals, then train a classifier to classify these proposals into semantic categories.

Video Object Segmentation

Video Object Segmentation

Object segmentation

Video object segmentation

Video Object Segmentation

- Goal: Generate accurate and temporally consistent pixel masks for objects in a video sequence.

(a) Video sequence

(b) Object mask

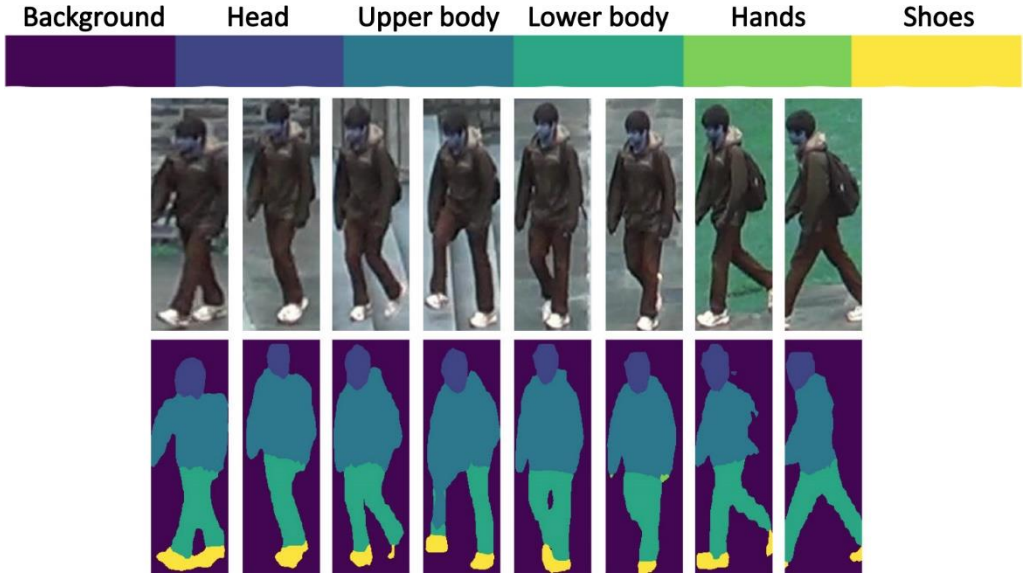
(c) Segmented object

VOS - Some applications

- Video Editing:

VOS - Some applications

- Video Surveillance and Action Recognition: .



video surveillance [1]

action recognition [2]

[1] Human semantic parsing for person re-identification, In *CVPR* 2018.

[2] Mask r-cnn, In *ICCV* 2017.

VOS - Some challenges

- Appearance changes:



VOS results by the method [1], where green circles highlight the false results due to appearance changes.

[1] OSVOS: One-Shot Video Object Segmentation, In *CVPR* 2017.

VOS - Some challenges

- Occlusions:



VOS results by the method [1], where yellow circles highlight the false results due to occlusions.

[1] Video Segmentation via Object Flow, In CVPR 2016.

VOS - Some challenges

- Distraction from similar backgrounds:



VOS results by the method [1], where yellow circles highlight the false results due to the distraction from similar backgrounds. Both the target object (break-dancer) and the segmented background object wear red sweater.

[1] OSVOS: One-Shot Video Object Segmentation, In *CVPR* 2017.

VOS: tasks

- Based on the supervision level required during inference, there are mainly four types of VOS tasks:
  - Semi-supervised (one-shot) Video Object Segmentation
  - Unsupervised (zero-shot) Video Object Segmentation
  - Interactive Video Object Segmentation
  - Language-guided Video Object Segmentation

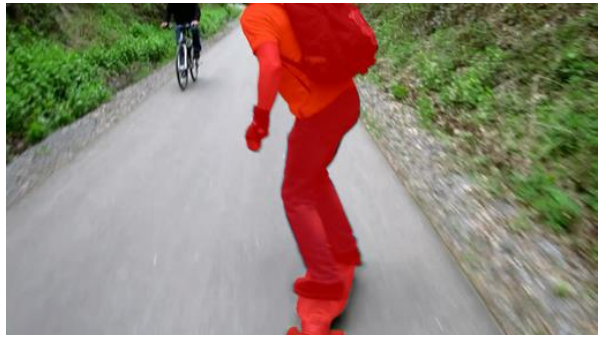
Semi-supervised (one-shot) VOS

Given: First frame ground truth

Goal: Complete video segmentation

- We get the first frame ground truth mask, we know what object to segment

Unsupervised (zero-shot) VOS



Given: Raw video sequence

Goal: Complete video segmentation

- We just get a raw video, we have to find the objects as well as their masks (generally based on saliency or motion information).

Interactive VOS

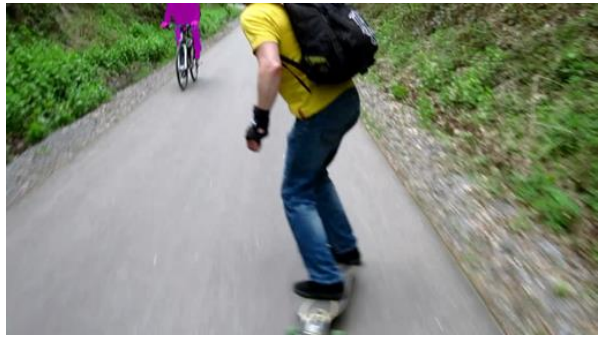



Given: Coarse annotations (e.g., scribbles)
Red: target object, Blue: background

Goal: Complete video segmentation

- We get coarse annotations (e.g., scribbles) to indicate what object to segment, simpler to provide than ground truth masks.
- we can also refine the unsatisfactory results by providing additional scribbles

Language-guided VOS

Given: A sentence “The man on a bicycle”

Goal: Complete video segmentation

- We get a sentence describing what object to segment.

Datasets

- Two benchmark datasets for VOS training and evaluation:
  - DAVIS (Densely Annotated Video Segmentation) [1] (evaluate temporal continuity)
  - YouTube-VOS [2] (more objects, frames, evaluate generalization, and long-range temporal consistency)

DATASETS	Annotation type	Resolution	# Videos	# Objects	Usage
DAVIS-2016	Dense annotation	854 x 480	50	50	SVOS, UVOS, IVOS
DAVIS-2017	Dense annotation	854 x 480	150	376	SVOS, UVOS
YouTube-VOS-2018	Every 5 frames	1280 x 720	4,453	7,754	SVOS
YouTube-VOS-2019	Every 5 frames	1280 x 720	4,519	8,614	SVOS
YouTube-VIS	Every 5 frames	1280 x 720	2,883	4,883	UVOS
Refer-YouTube-VOS	Every 5 frames	1280 x 720	3,975	7,451	LVOS

[1] A benchmark dataset and evaluation methodology for video object segmentation, In *CVPR* 2016.

[2] Youtube-vos: A large-scale video object segmentation benchmark, In *Arxiv* 2018.

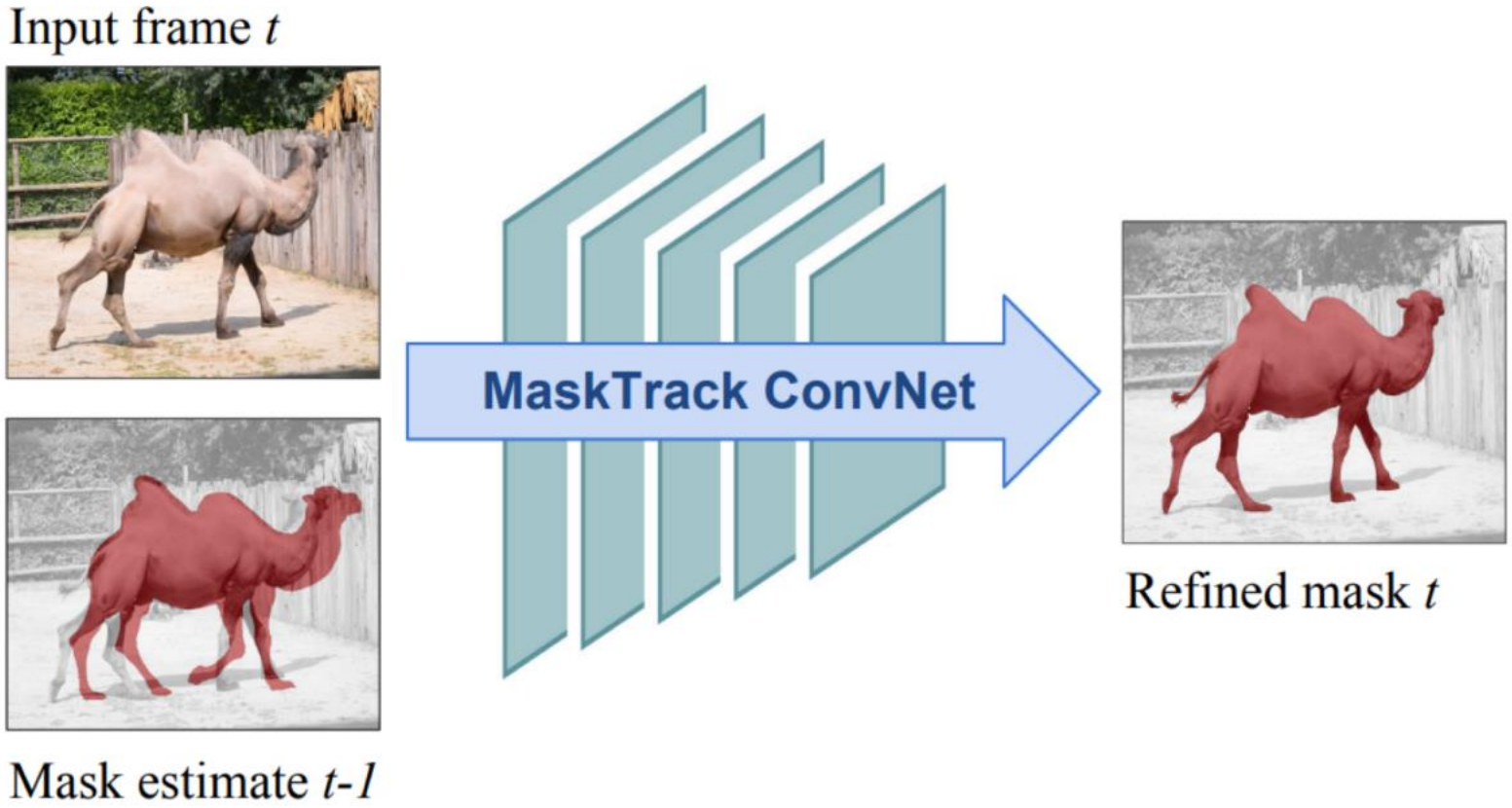
One classical SVOS model: MaskTrack

- Assumption: an object, i.e., a mask, does not move a lot from frame to frame.
- We can often start with an approximate mask (either from previous frame or from coarse estimate).
- We can then use a refinement network to accurately refine the mask estimate.

MaskTrack: Learning Video Object Segmentation from Static Images, CVPR 2017.

One classical SVOS model: MaskTrack

- Overview: MaskTrack takes a tensor with four channels (RGB + previous frame mask) as input, predicts object masks for the current frame.



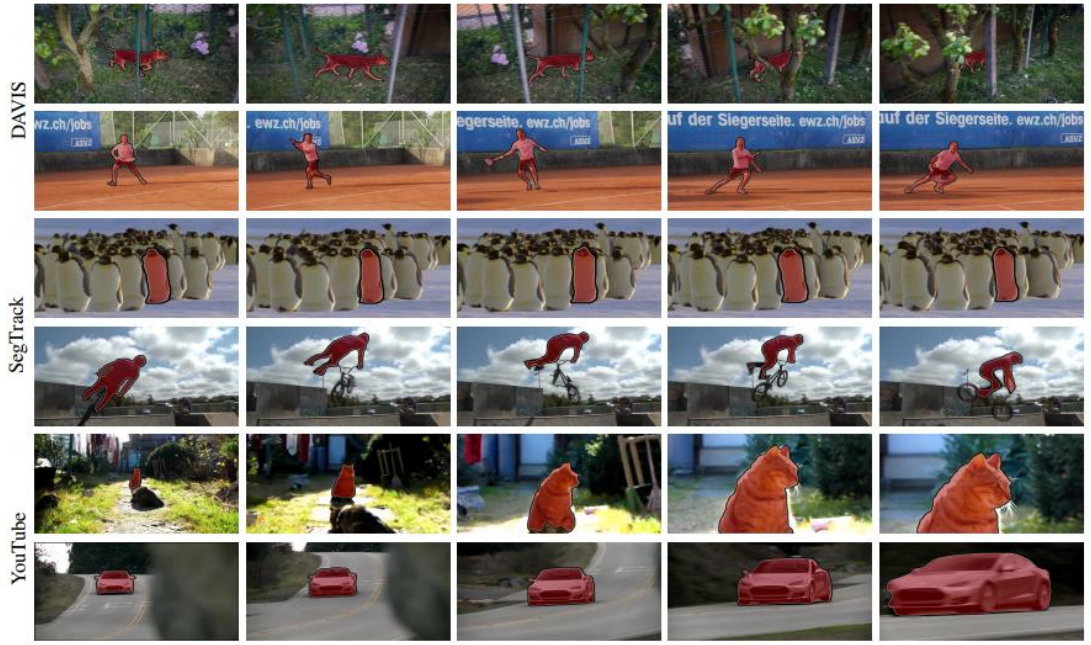
One classical SVOS model: MaskTrack

- Network
  - DeepLabv2-VGG network
  - pre-trained on ImageNet
  - Extra mask channel of filters: gaussian initialization
- Offline training
  - Does not require pixel-label annotations on videos
  - Images and masks: ECSSD, MSRA10K, SOD, and PASCAL-S
  - Deforming the binary segmentation masks

(a) Annotated image (b) Example training masks

One classical SVOS model: MaskTrack

- Online training
  - 200 iterations
  - 1000 augmented training samples from the first frame
  - Same learning parameters as for offline training

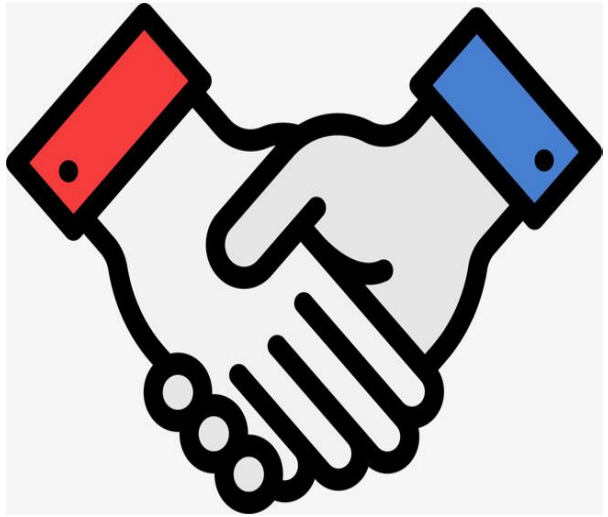


1st frame, GT segment Results with MaskTrack, the frames are chosen equally distant based on the video sequence length

Conclusions

Conclusion

- Object Tracking: General
- Semantic Segmentation
- Video Object Segmentation



Thanks

zhengf@sustc.edu.cn