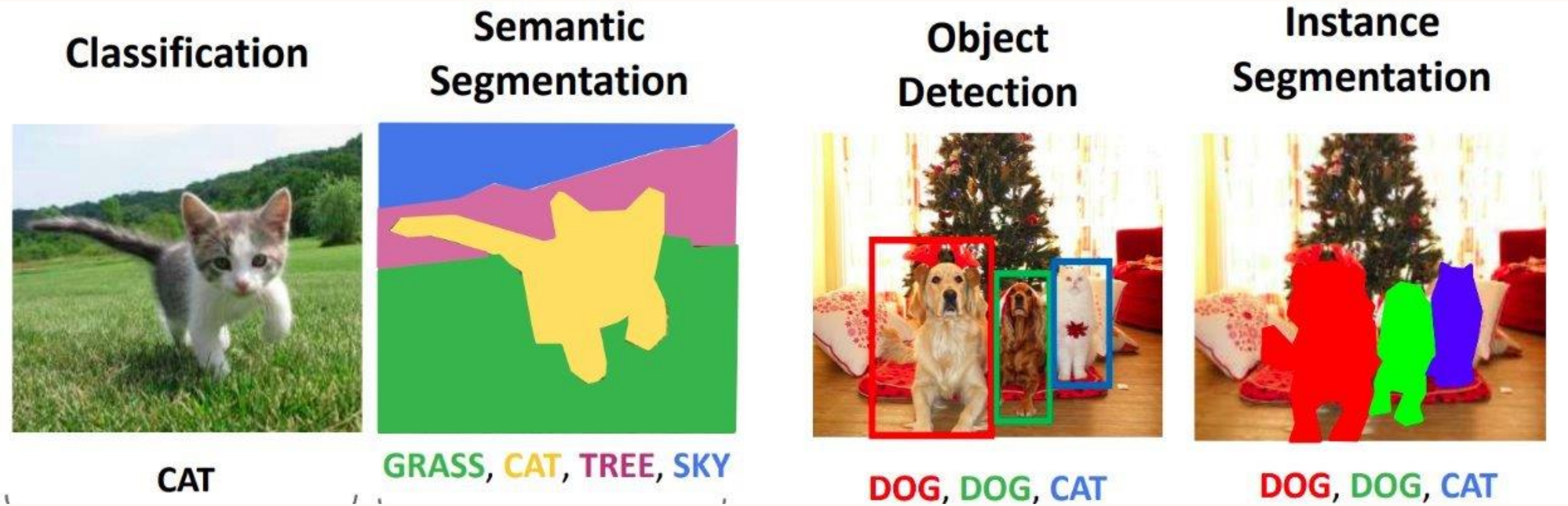
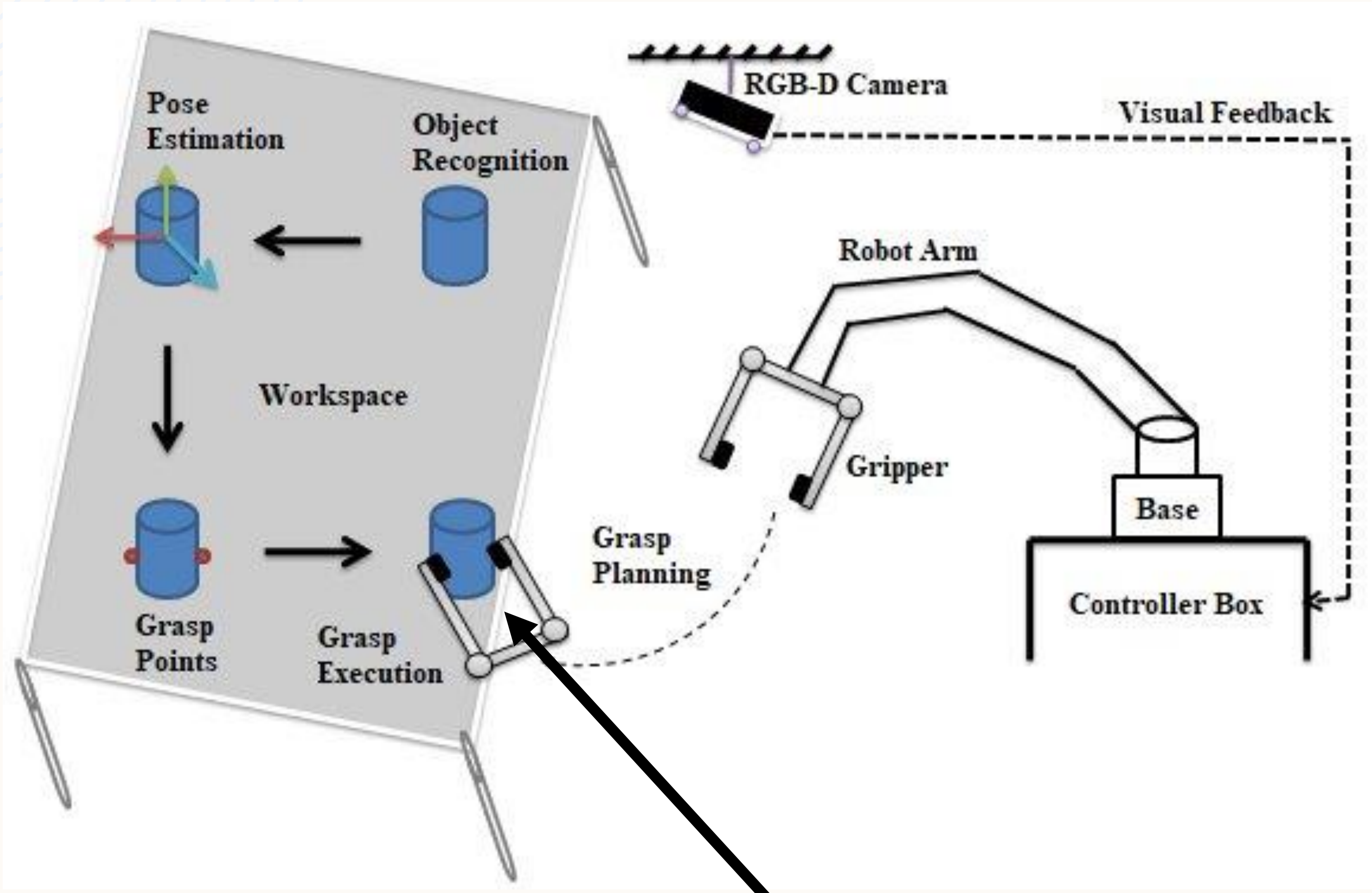
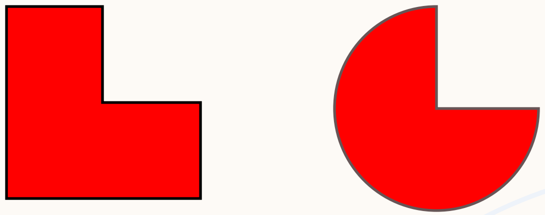
AI IN ROBOTICS – SEGMENTATION

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

PRESENTER : HASSAN YOUSEFZADE


Classification
Semantic Segmentation
Object Detection
Instance Segmentation
CAT
GRASS, CAT, TREE, SKY
DOG, DOG, CAT
DOG, DOG, CAT





If this object is complex

Object detection vs object segmentation

Segmentation

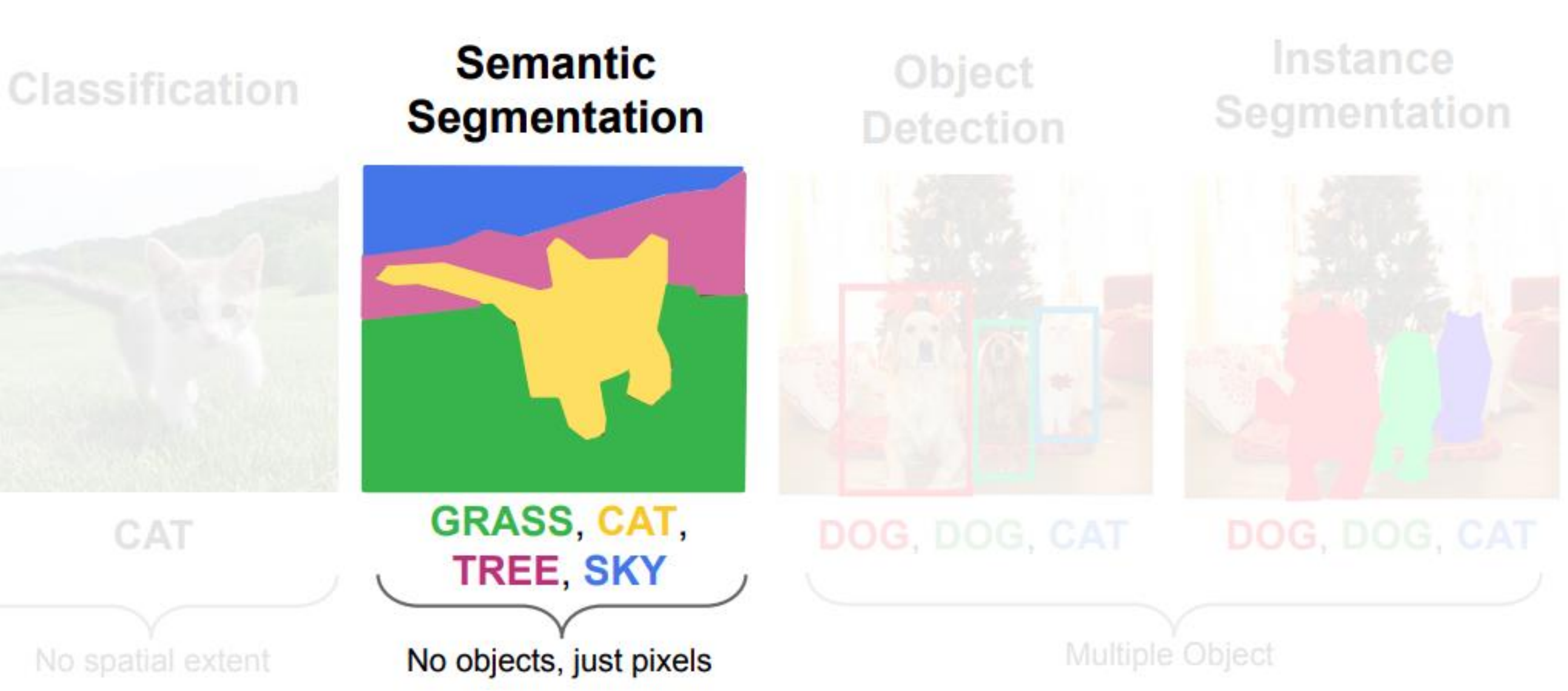
- **High Detail**: Image segmentation helps robots assign each pixel to a class, thus identifying the precise shape and boundaries of objects. This high accuracy is useful for grasping and manipulating complex objects.
- **Practical Application**: This allows robots to grasp objects with complex and irregular shapes, such as car parts, precise tools, or food items.

Object detection

- **Fast Identification**: Object detection allows robots to quickly identify objects and determine their approximate position in the environment.
- **Practical Application**: Useful for manipulation tasks that require fast object identification, such as picking items off a conveyor belt.

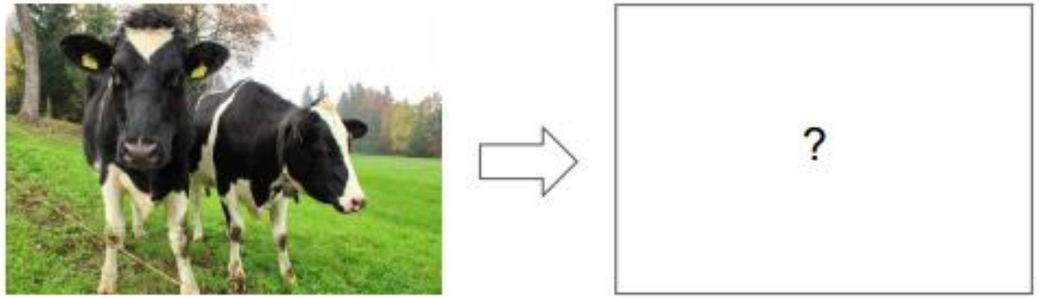
Environmental Complexity

- **Image Segmentation**: Useful in complex environments requiring precise differentiation of objects from the background and detailed object identification.
- **Object Detection**: Useful in simpler environments or applications requiring quick object identification.

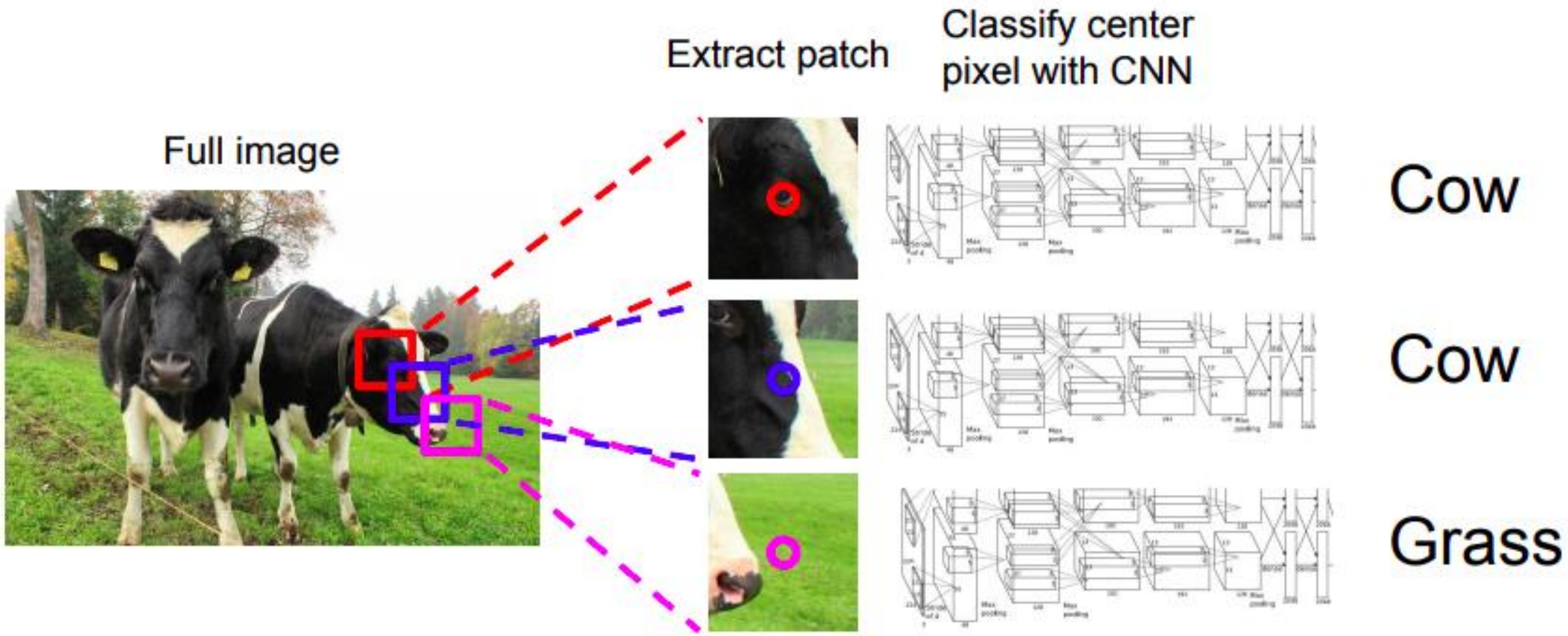

Classification
Semantic Segmentation
Object Detection
Instance Segmentation
CAT
GRASS, CAT, TREE, SKY
DOG, DOG, CAT
DOG, DOG, CAT
No spatial extent
No objects, just pixels
Multiple Object

GRASS, CAT,
TREE, SKY, ...

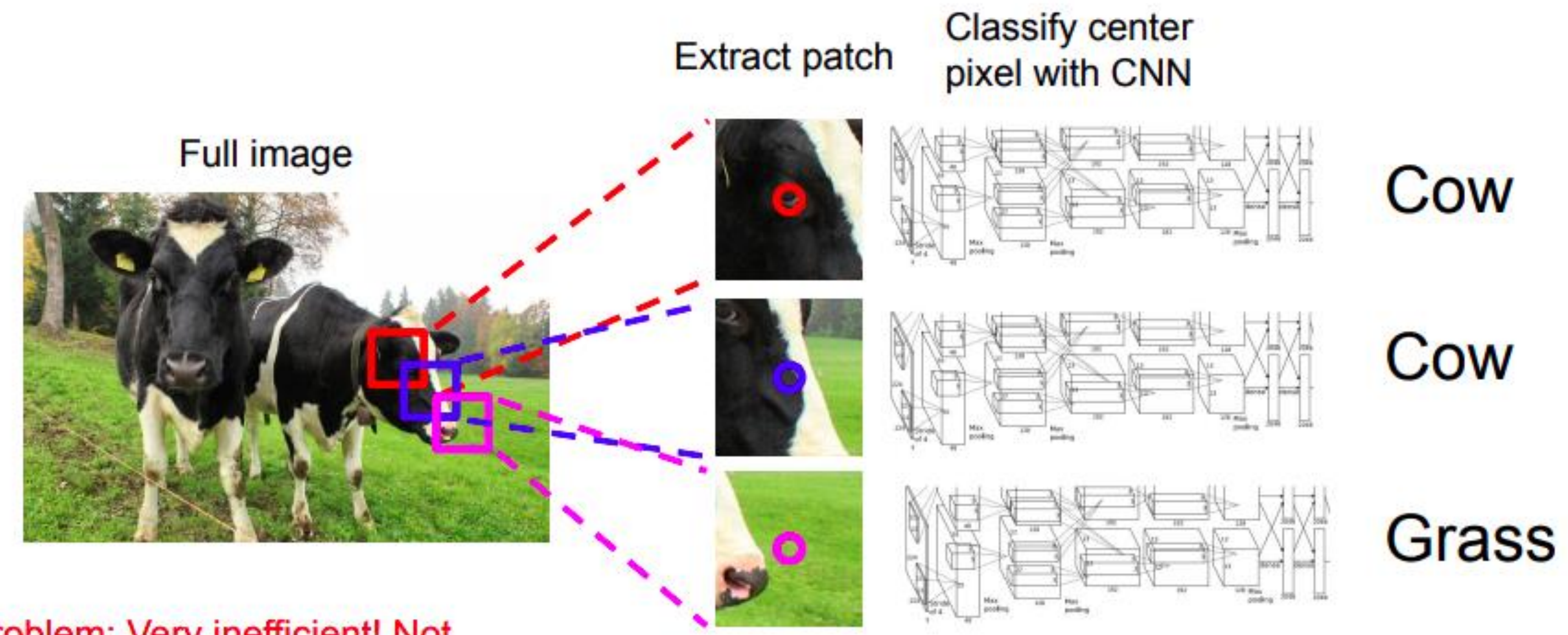
Paired training data: for each training image, each pixel is labeled with a semantic category.



At test time, classify each pixel of a new image.



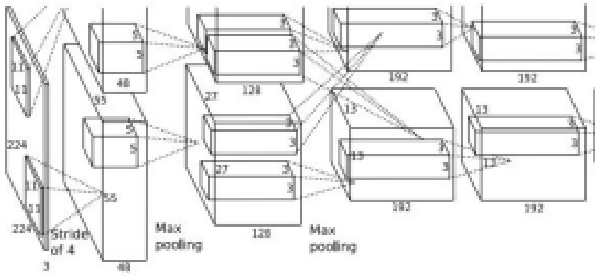
Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014



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

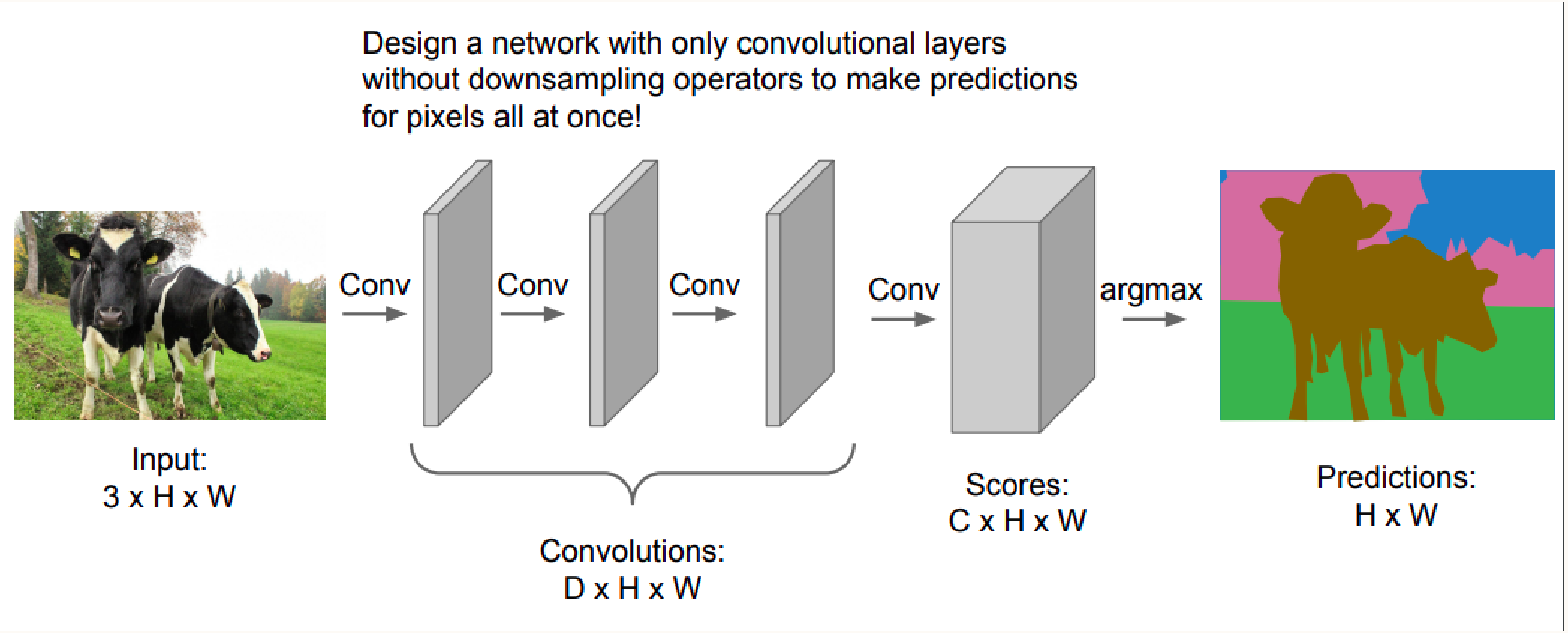
Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

Full image

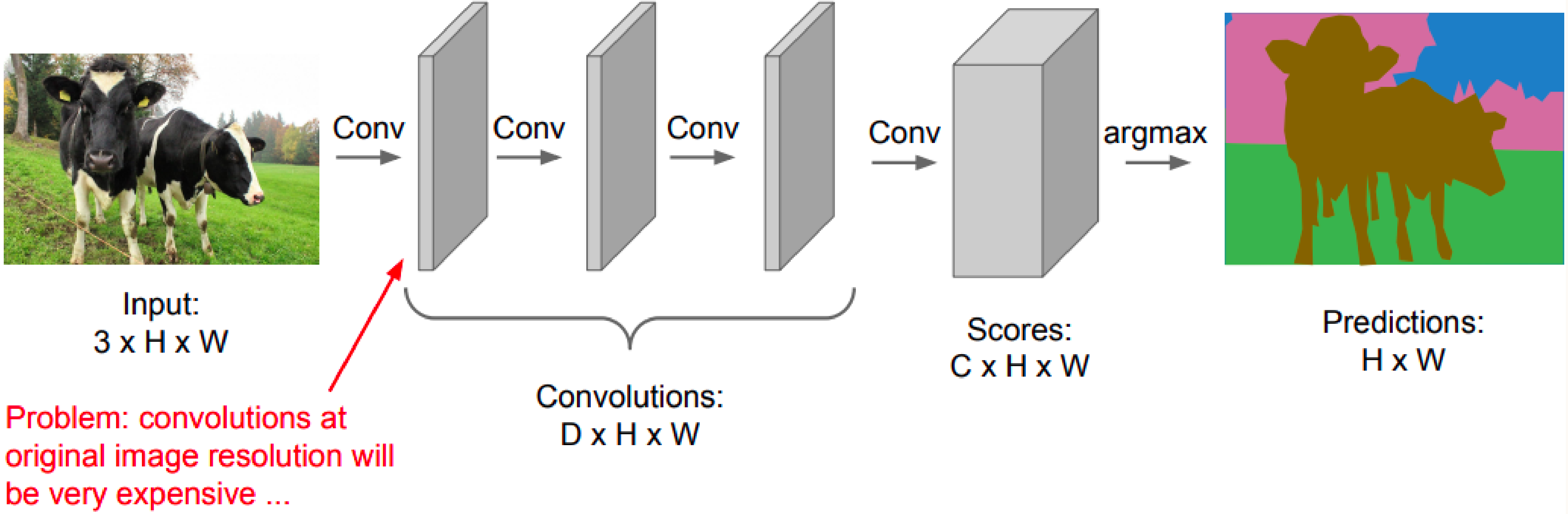


An intuitive idea: encode the entire image with conv net, and do semantic segmentation on top.

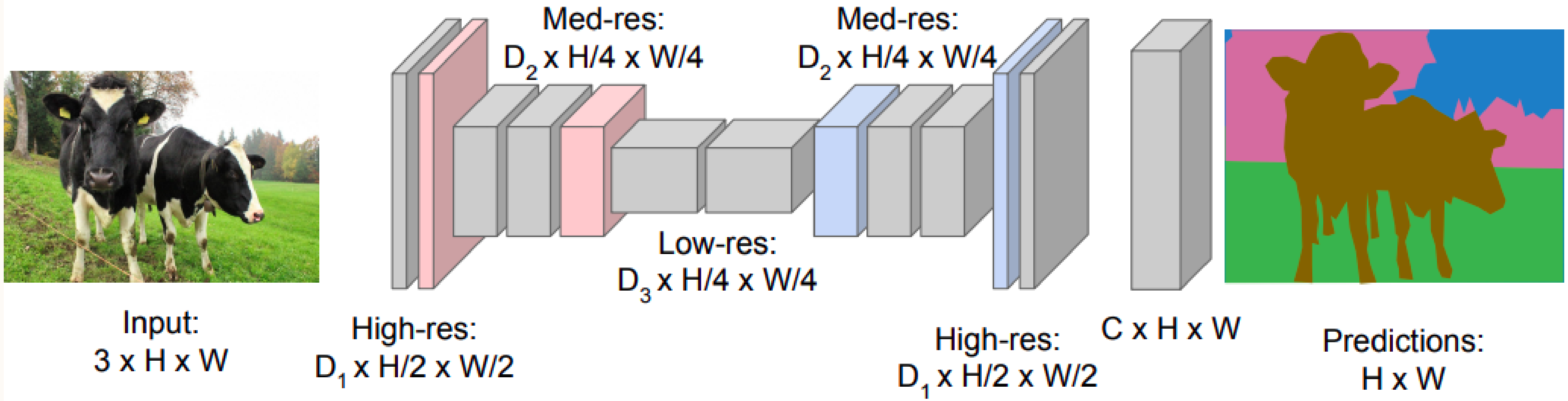
Problem: classification architectures often reduce feature spatial sizes to go deeper, but semantic segmentation requires the output size to be the same as input size.

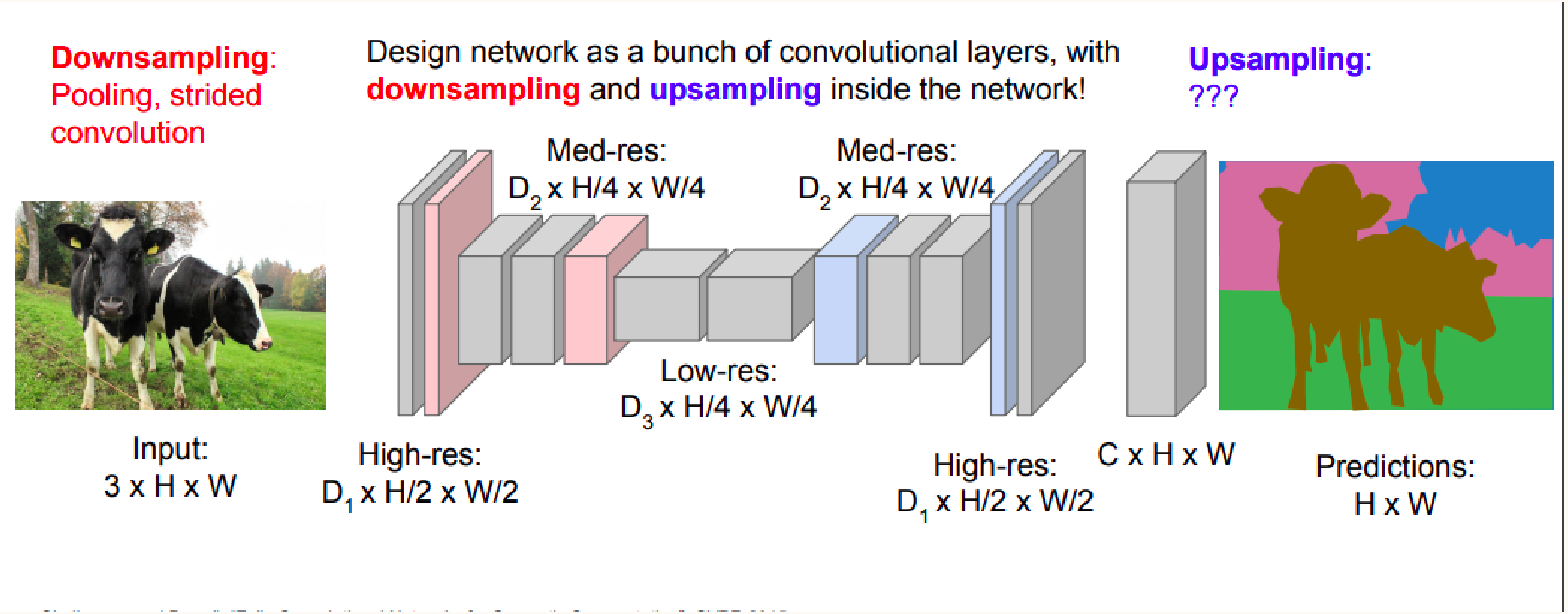

Design a network with only convolutional layers
without downsampling operators to make predictions
for pixels all at once!
Conv
Conv
Conv
Conv
argmax
Input:
3 x H x W
Convolutions:
D x H x W
Scores:
C x H x W
Predictions:
H x W

Design a network with only convolutional layers without downsampling operators to make predictions for pixels all at once!

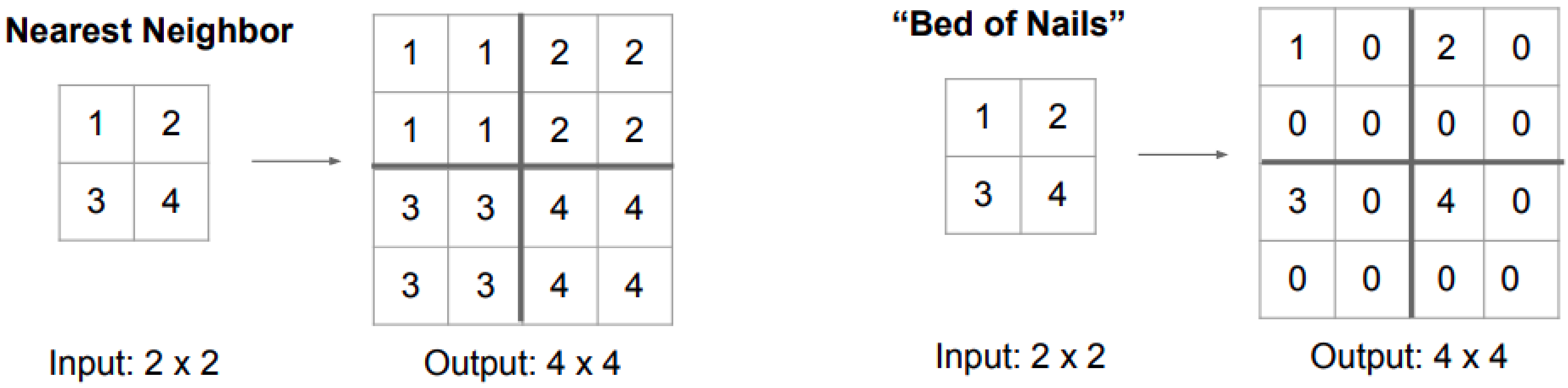


Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

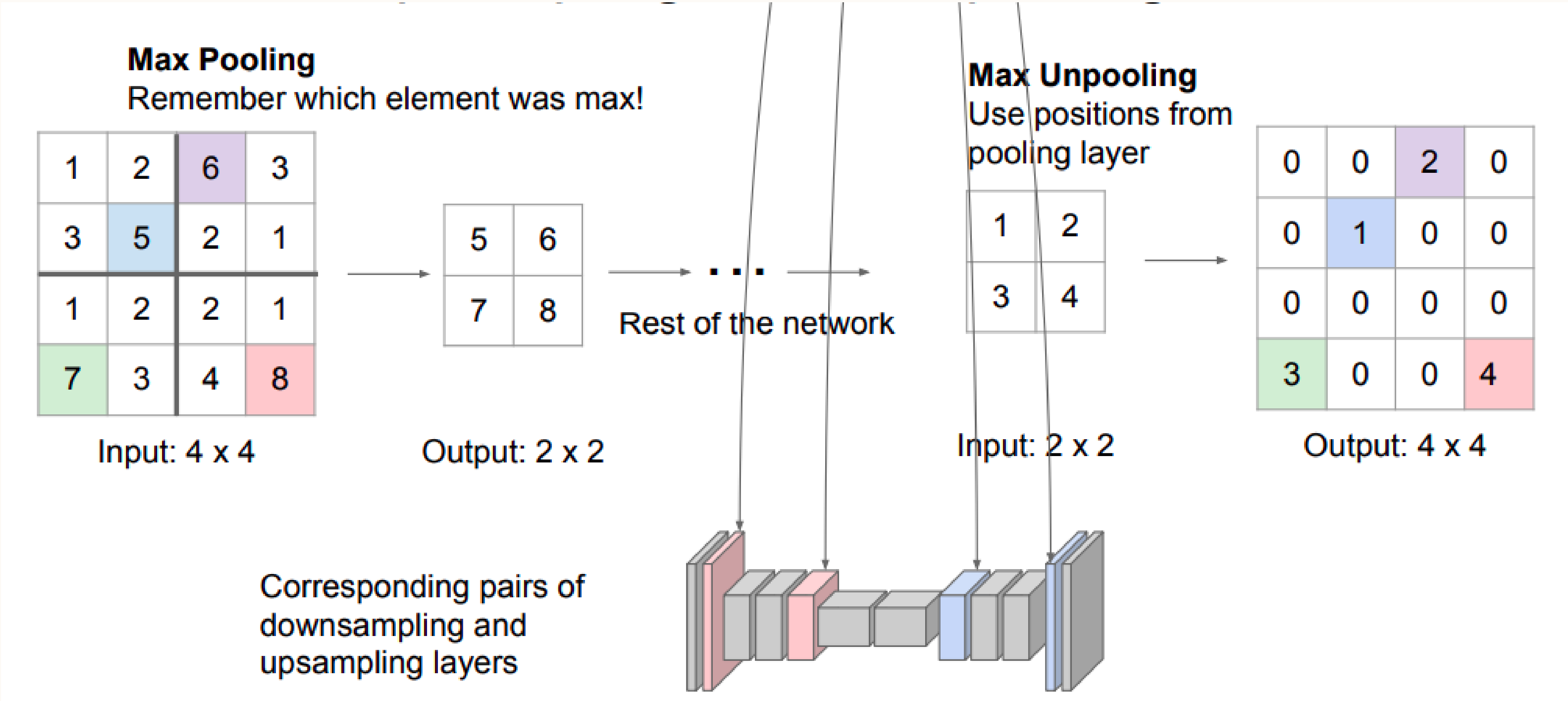



Downsampling:
Pooling, strided convolution
Design network as a bunch of convolutional layers, with downsampling and upsampling inside the network!
Upsampling:
???
Med-res:
D2 x H/4 x W/4
Med-res:
D2 x H/4 x W/4
Low-res:
D3 x H/4 x W/4
Input:
3 x H x W
High-res:
D1 x H/2 x W/2
High-res:
D1 x H/2 x W/2
C x H x W
Predictions:
H x W

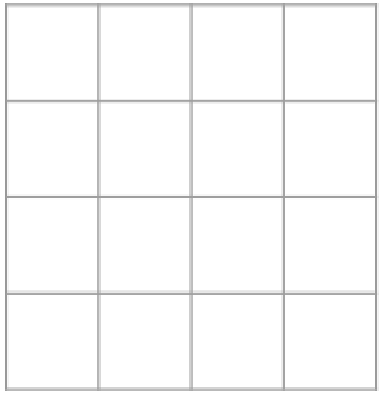
pooling


Nearest Neighbor
1 2
3 4
1 1 2 2
1 1 2 2
3 3 4 4
3 3 4 4
Input: 2 x 2
Output: 4 x 4
"Bed of Nails"
1 2
3 4
1 0 2 0
0 0 0 0
3 0 4 0
0 0 0 0
Input: 2 x 2
Output: 4 x 4

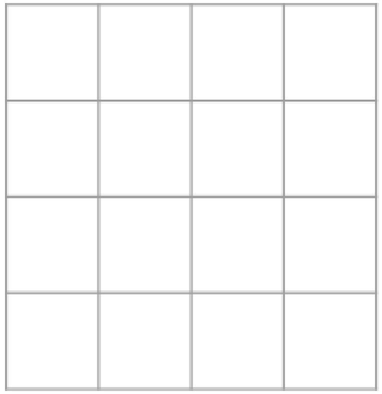
Max - pooling


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

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

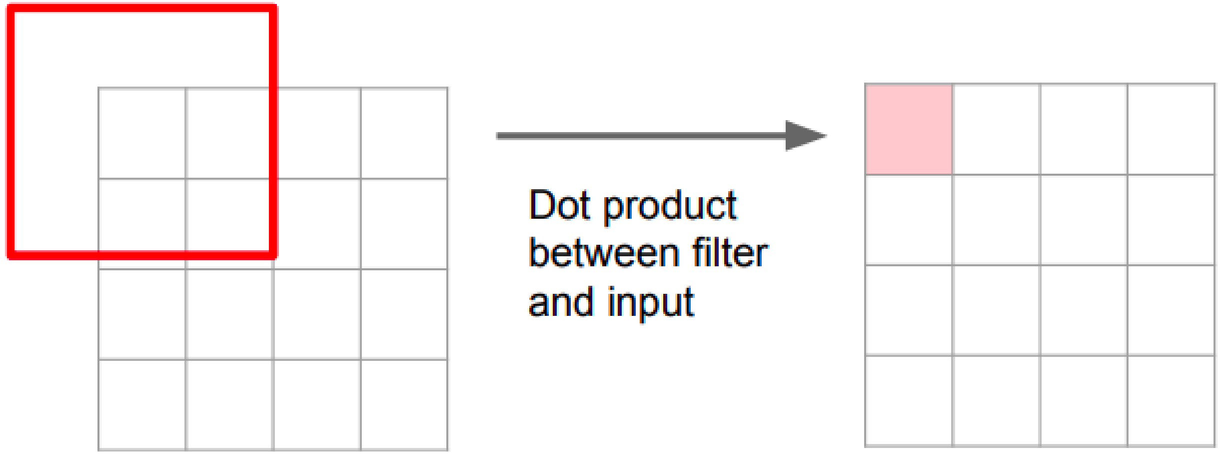


Input: 4 x 4



Output: 4 x 4

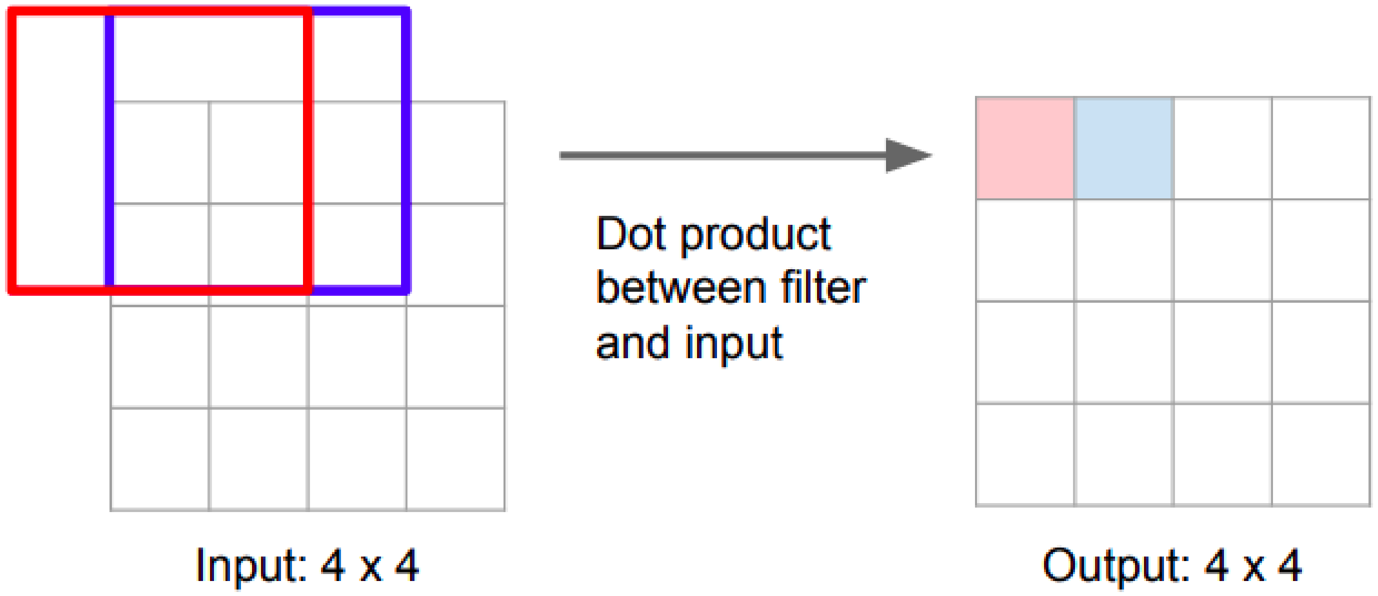
Recall: Normal 3 x 3 convolution, stride 1 pad 1



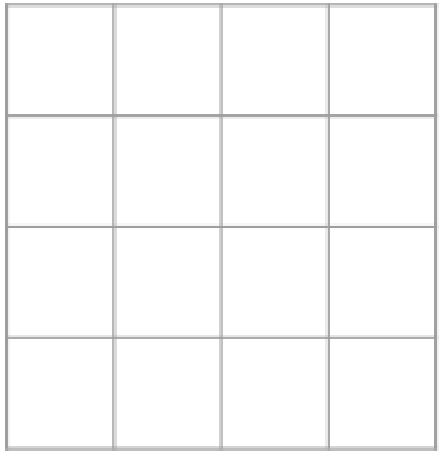
Input: 4 x 4

Output: 4 x 4

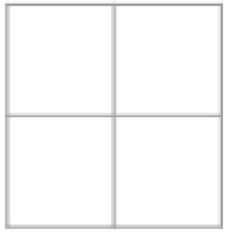
Recall: Normal 3 x 3 convolution, stride 1 pad 1



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

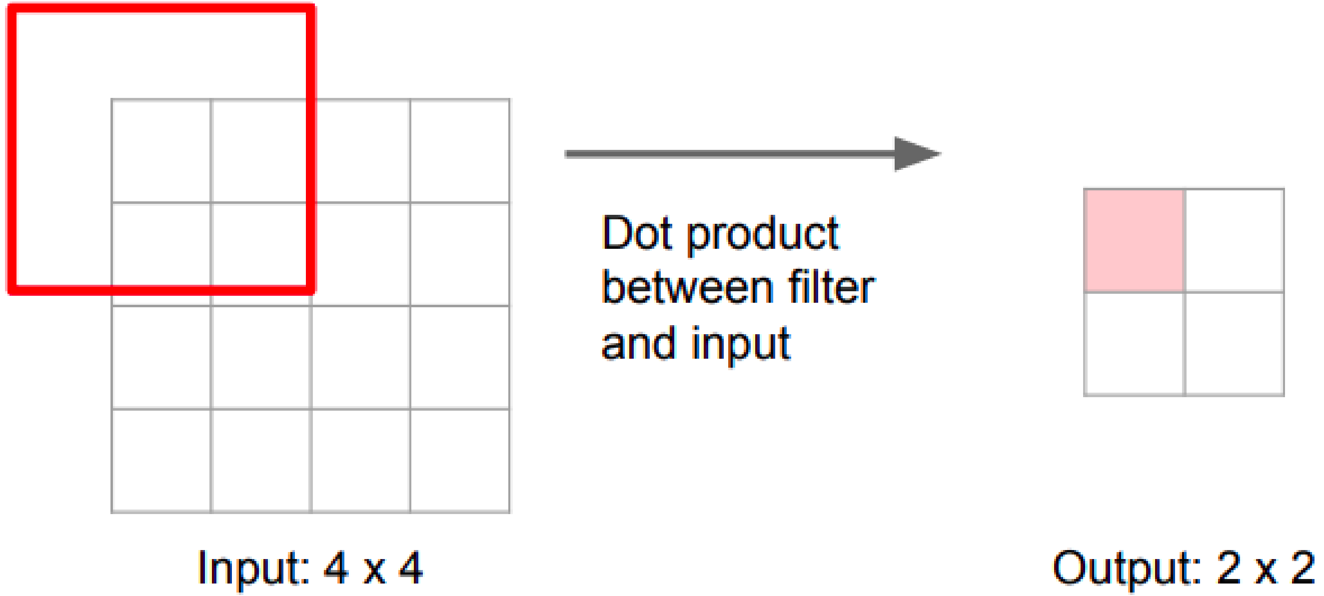


Input: 4 x 4

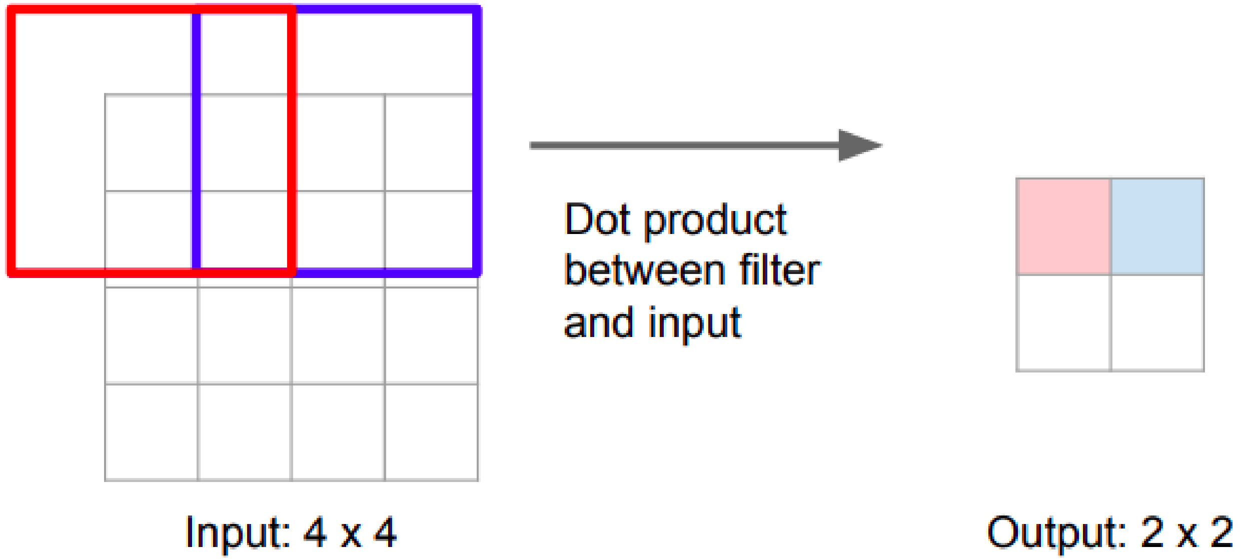


Output: 2 x 2

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



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

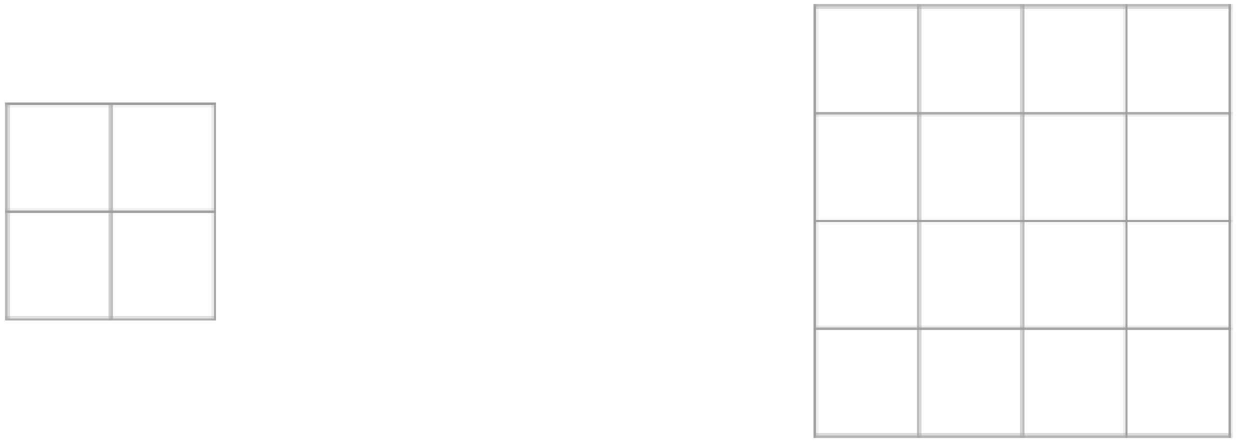


Filter moves 2 pixels in the input for every one pixel in the output

Stride gives ratio between movement in input and output

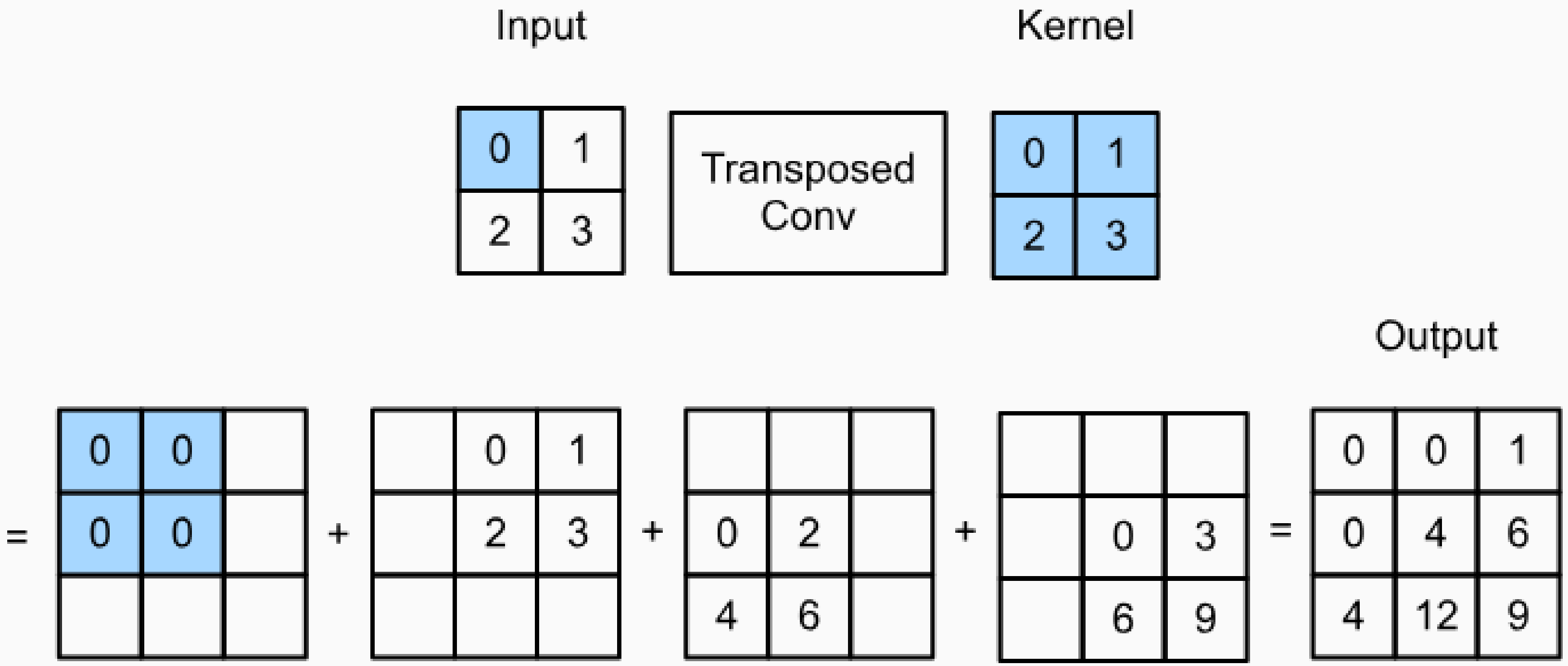
We can interpret strided convolution as “learnable downsampling”.

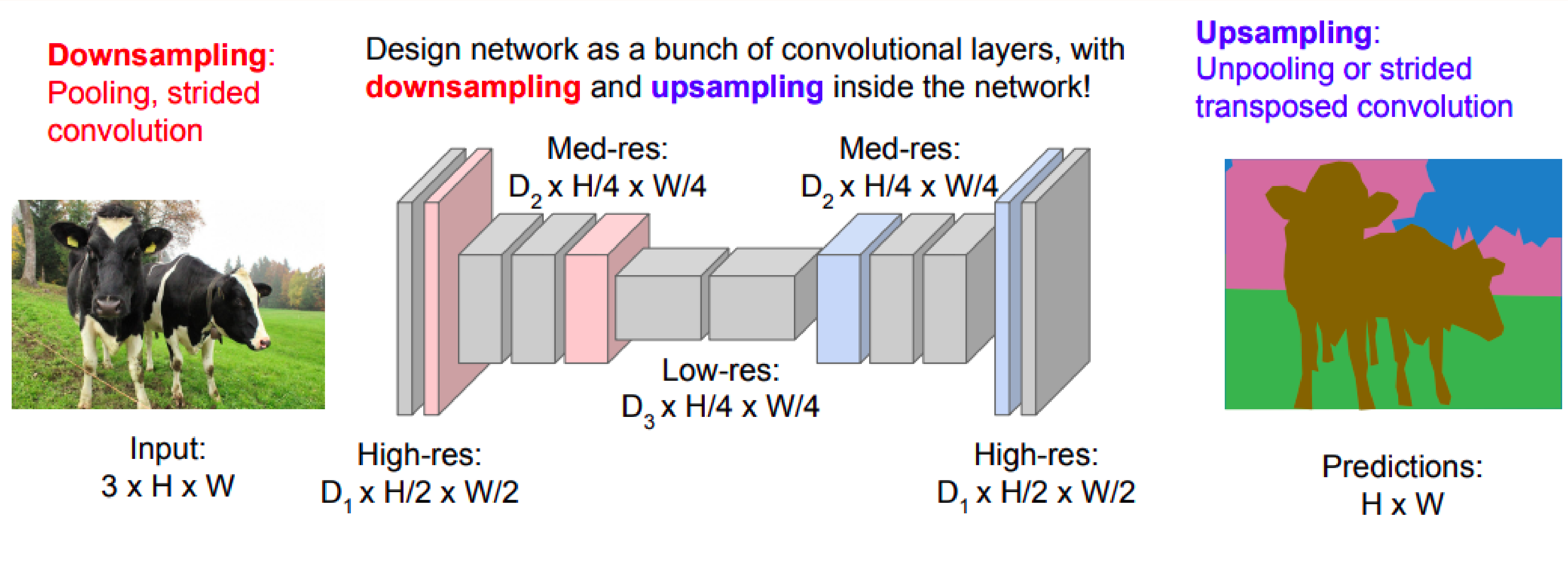
3 x 3 **transposed** convolution, stride 2 pad 1



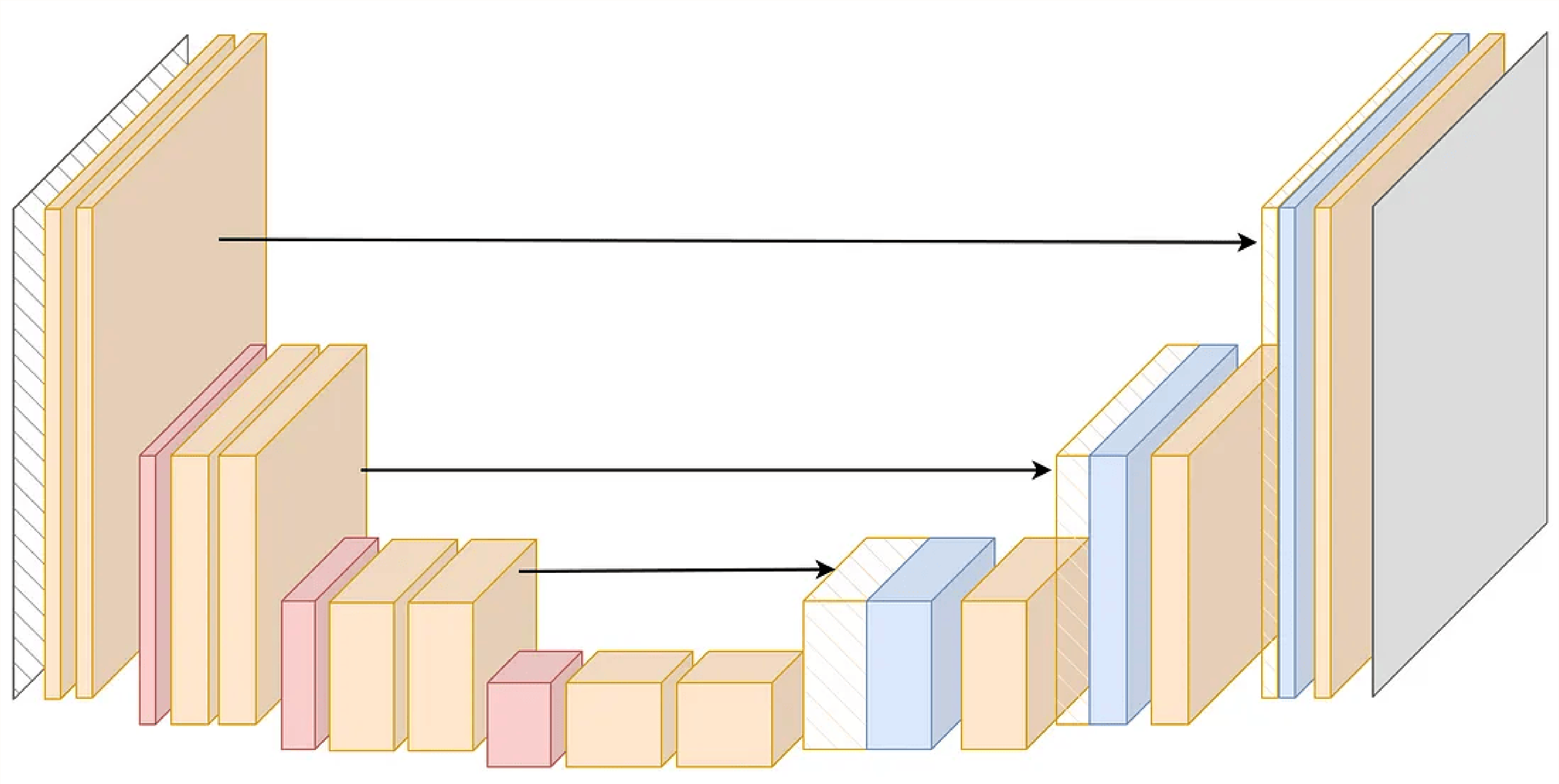
Input: 2 x 2

Output: 4 x 4


Input
0
1
2
3
Transposed Conv
Kernel
0
1
2
3
Output
=
0 0
0 0
+
0 1
2 3
+
0 2
4 6
+
0 3
6 9
=
0 0 1
0 4 6
4 12 9

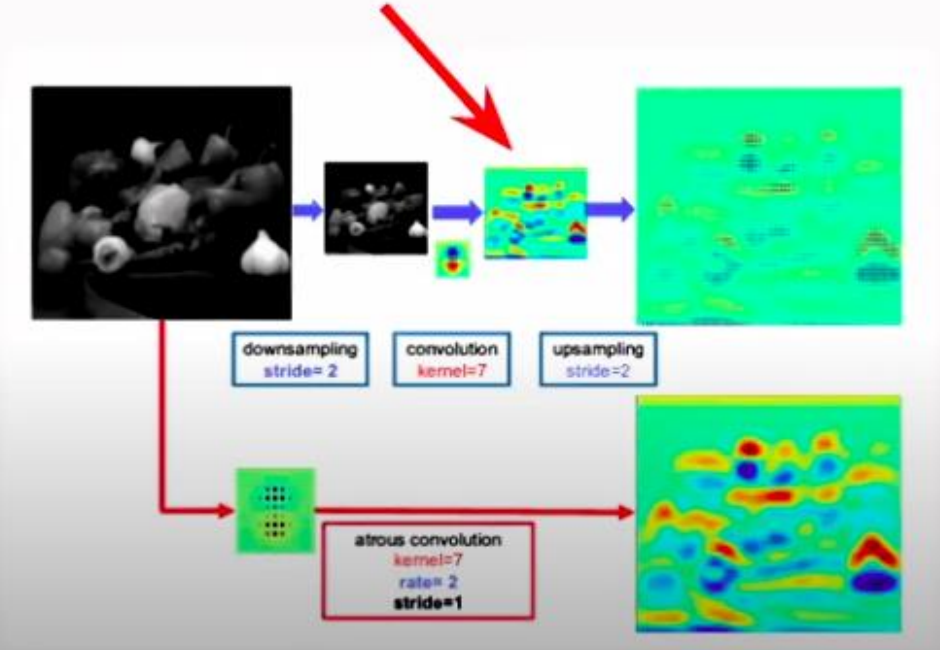

Downsampling:
Pooling, strided convolution
Design network as a bunch of convolutional layers, with downsampling and upsampling inside the network!
Upsampling:
Unpooling or strided transposed convolution
Med-res:
D_2 x H/4 x W/4
Med-res:
D_2 x H/4 x W/4
Low-res:
D_3 x H/4 x W/4
Input:
3 x H x W
High-res:
D_1 x H/2 x W/2
High-res:
D_1 x H/2 x W/2
Predictions:
H x W

UNET



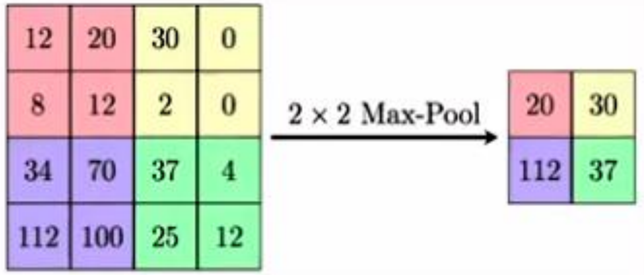
DEEPLAB

1) Problem: Reduced feature resolution



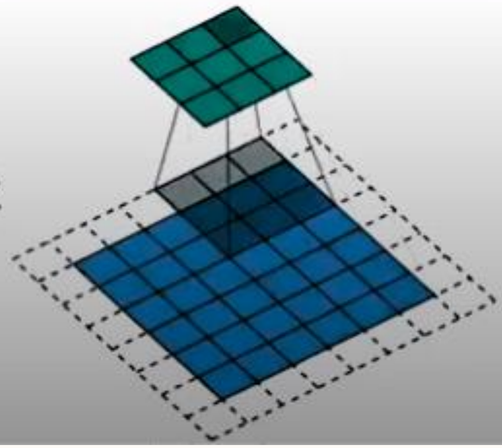
Causes:

- Maxpooling



Source: computersciencewiki.org

- Conv, strides = 2



DEEPLAB

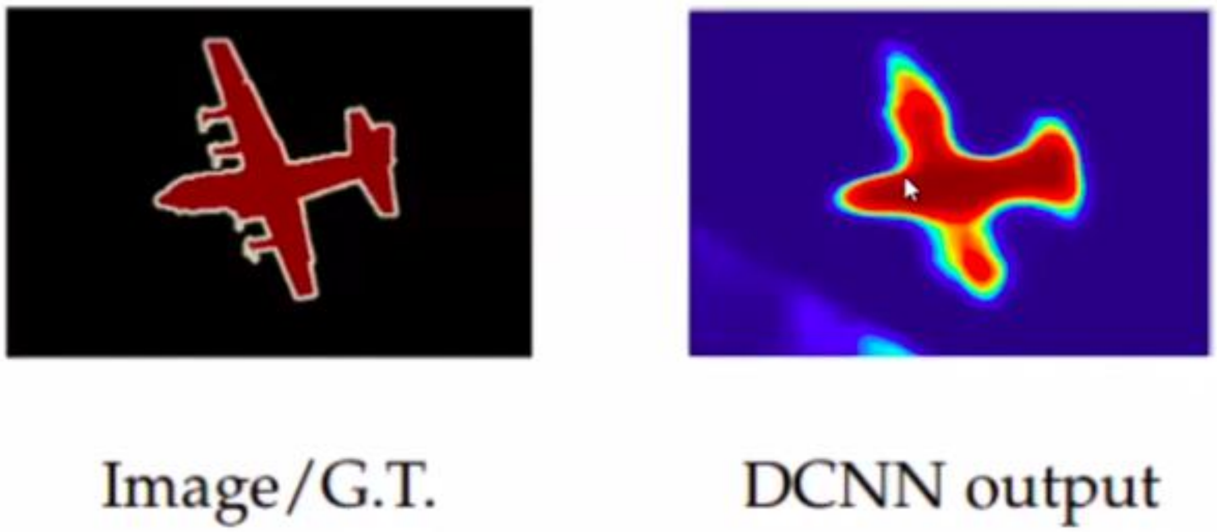
2) Problem: Existence of multiple-scale objects

Same class

Different size (FoV)

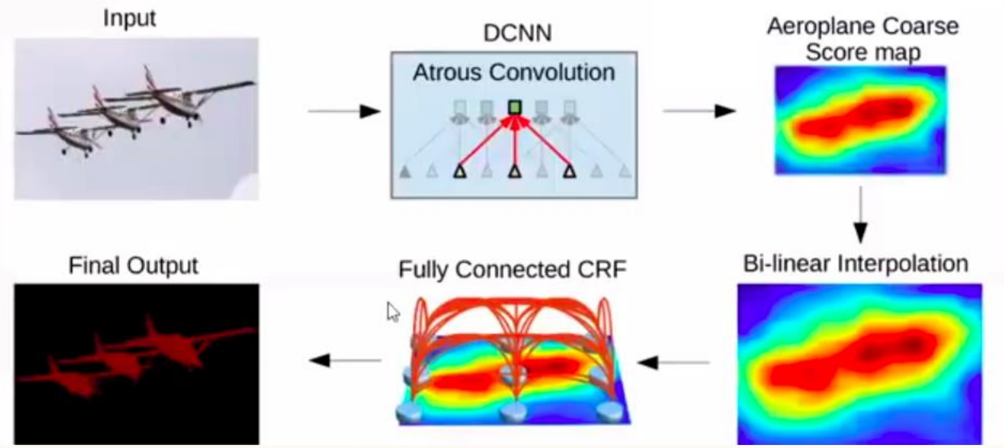
DEEPLAB

3) Reduced accuracy (in borders)



DEEPLAB

1) Atrous convolution (or dilated convolution)
2) Atrous Spatial Pyramid Pooling (ASPP)
3) Conditional Random Fields (CRFs)



Pipeline

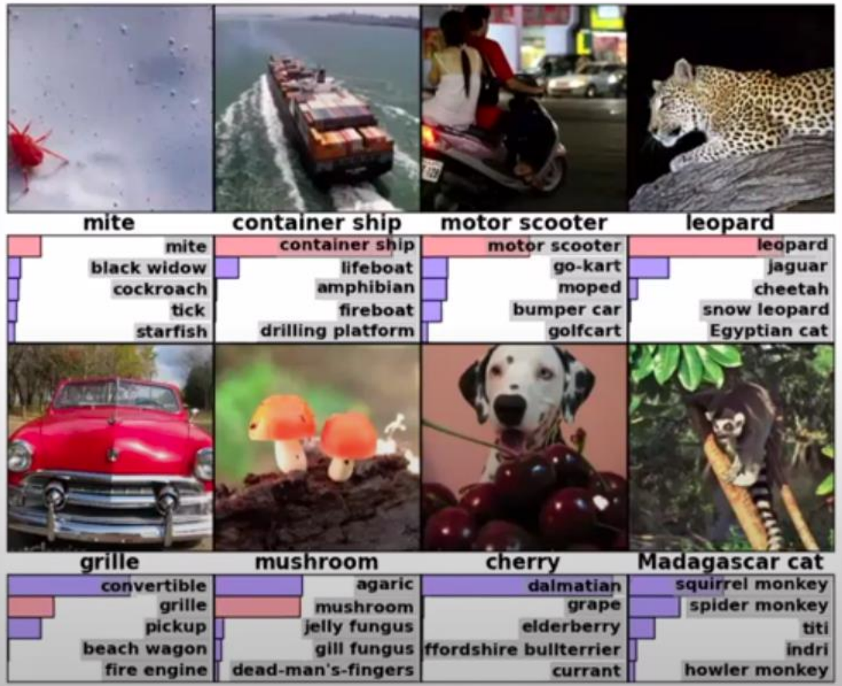
Backbone
ASPP
Upsample x8
CRF

DEEPLAB

3) Reduced accuracy (in borders)

Cause: Downsampling, maxpooling, useful to achieve invariance in **classification**

Image/G.T. DCNN output

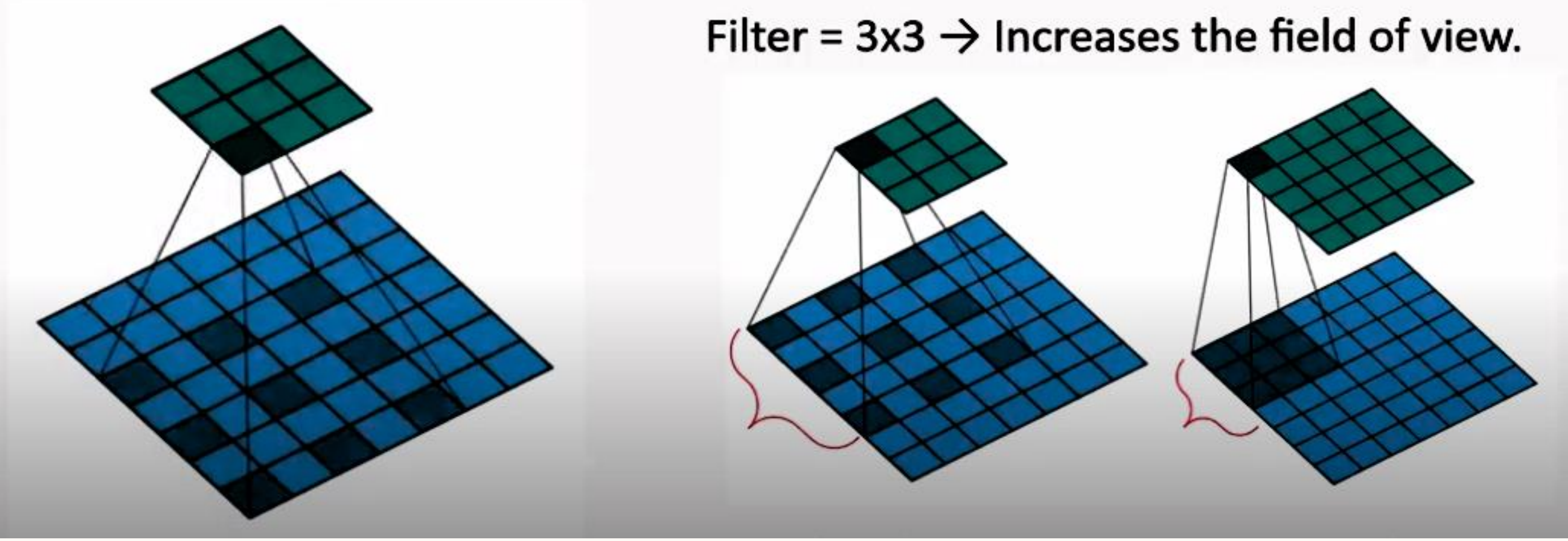


In Segmentation we want to preserve **spatial information**

DEEPLAB

1) Atrous convolution (or dilated convolution)

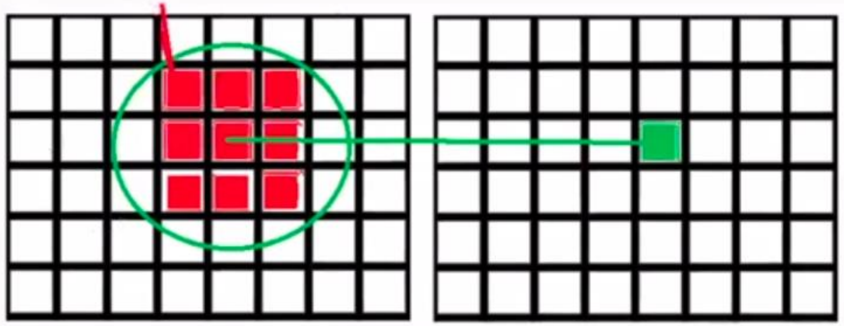
Filter = 3x3 → Increases the field of view.



DEEPLAB

(Field of view, reminder)

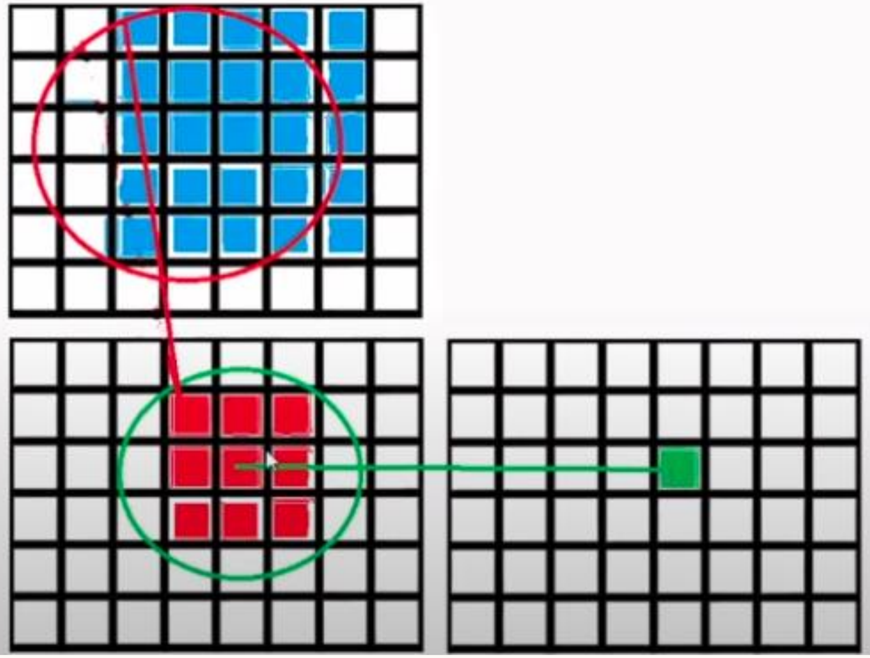
Operation	Field of view (size)
Conv 3x3	3



DEEPLAB

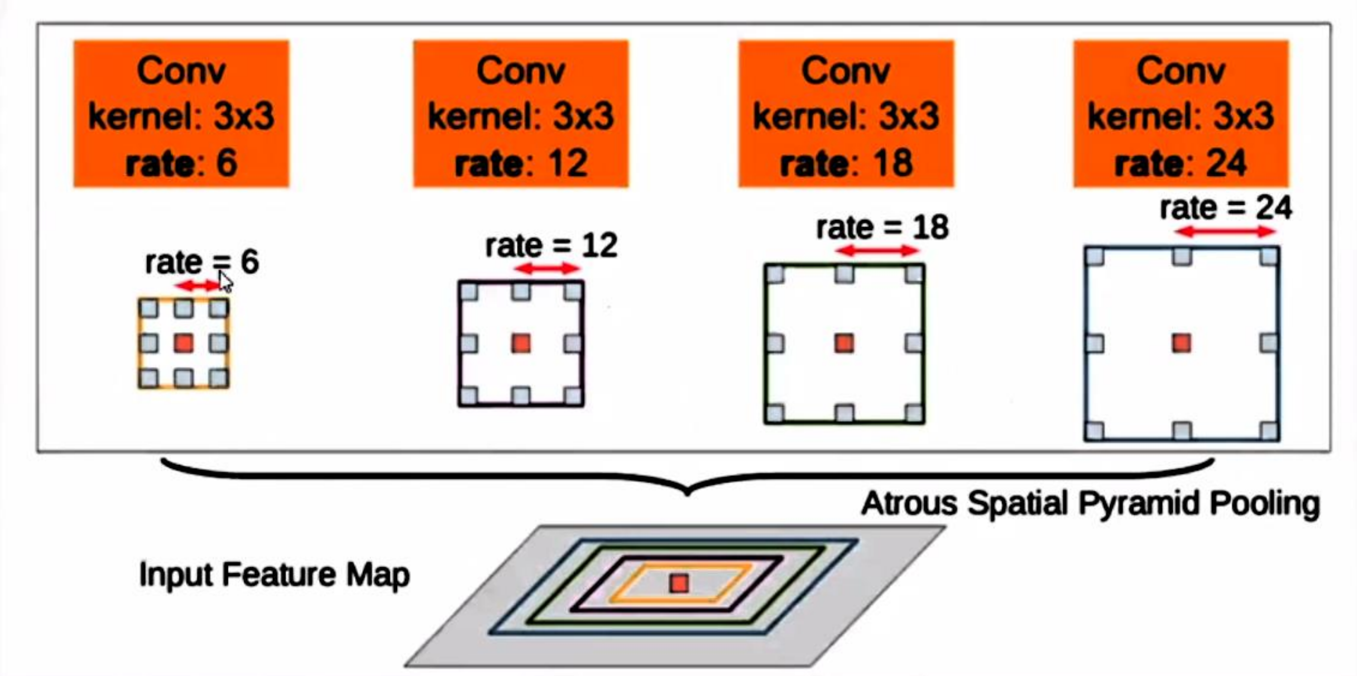
(Field of view, reminder)

Operation	Field of view (size)
Conv 3x3	3
Conv 3x3	3 + 2 = 5
...	...



DEEPLAB

2) Atrous Spatial Pyramid Pooling (ASPP)



DEEPLAB

3) Conditional Random Fields (CRFs)

$$E(x) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j) \qquad (2)$$

$$\mu(x_i, x_j) = 1 \text{ if } x_i \neq x_j,$$

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \left[w_1 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\alpha^2} - \frac{||I_i - I_j||^2}{2\sigma_\beta^2} \right) + w_2 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\gamma^2} \right) \right] \qquad (3)$$

p = coordinates

I = RGB intensity values

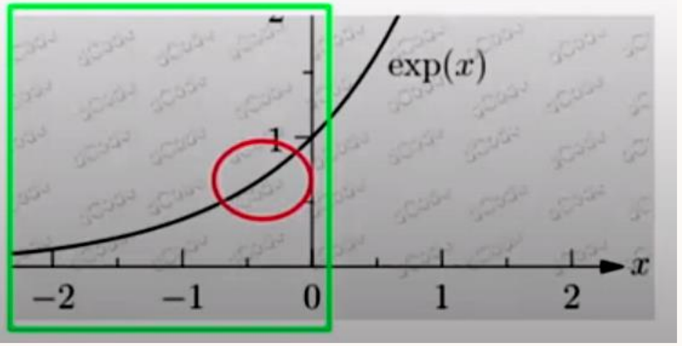
DEEPLAB

3) Conditional Random Fields (CRFs)

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j)\left[w_1 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\alpha^2} - \frac{||I_i - I_j||^2}{2\sigma_\beta^2}\right) + w_2 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\gamma^2}\right)\right] \quad (3)$$

- Small distance, similar intensities

Small negative values → large penalty



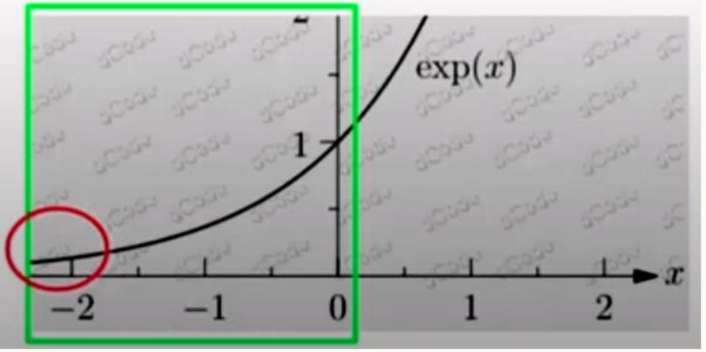
DEEPLAB

3) Conditional Random Fields (CRFs)

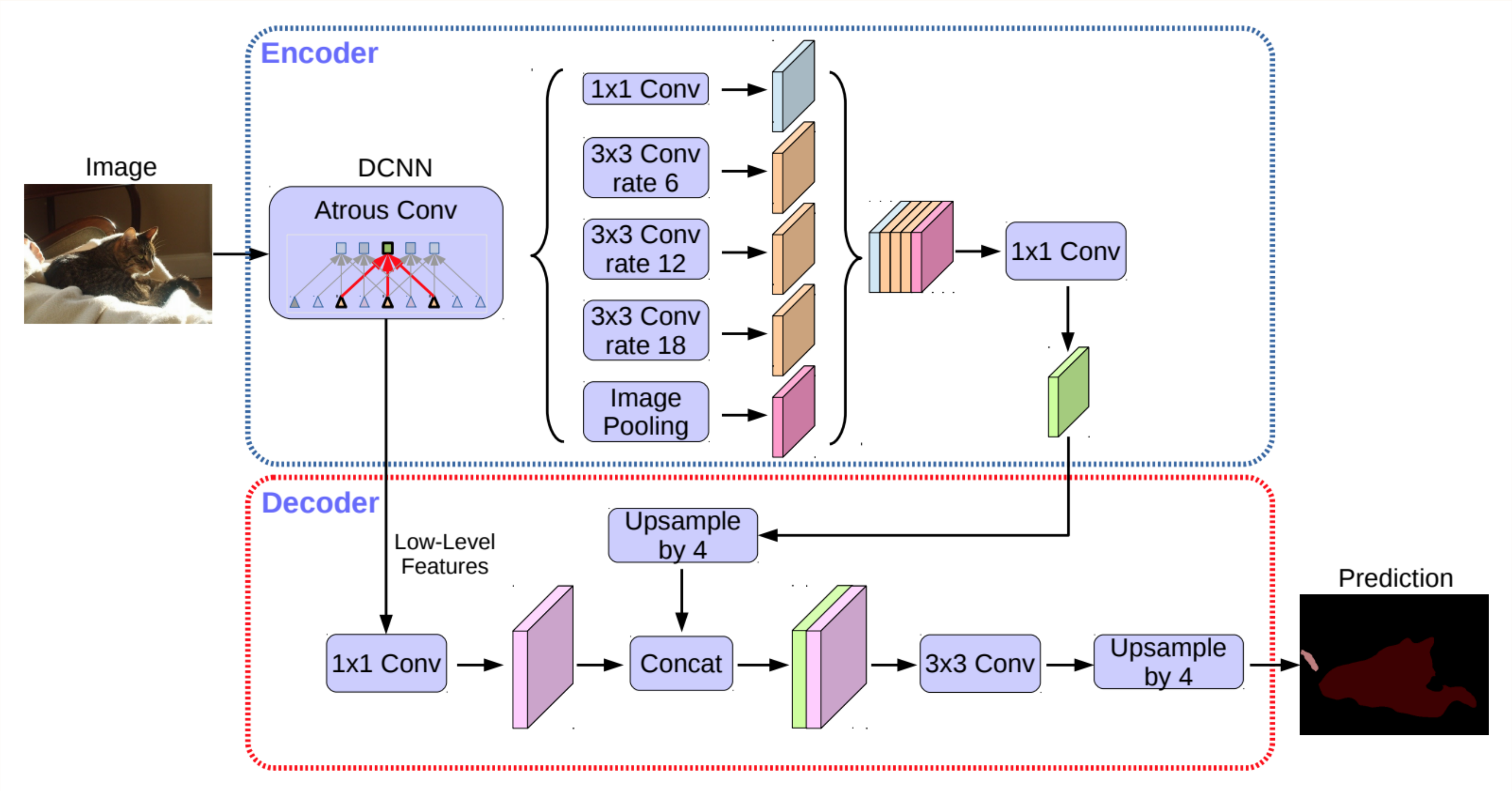
$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j)\Bigg[w_1 \exp\Big(-\frac{||p_i - p_j||^2}{2\sigma_\alpha^2} - \frac{||I_i - I_j||^2}{2\sigma_\beta^2}\Big) + w_2 \exp\Big(-\frac{||p_i - p_j||^2}{2\sigma_\gamma^2}\Big)\Bigg] \quad (3)$$

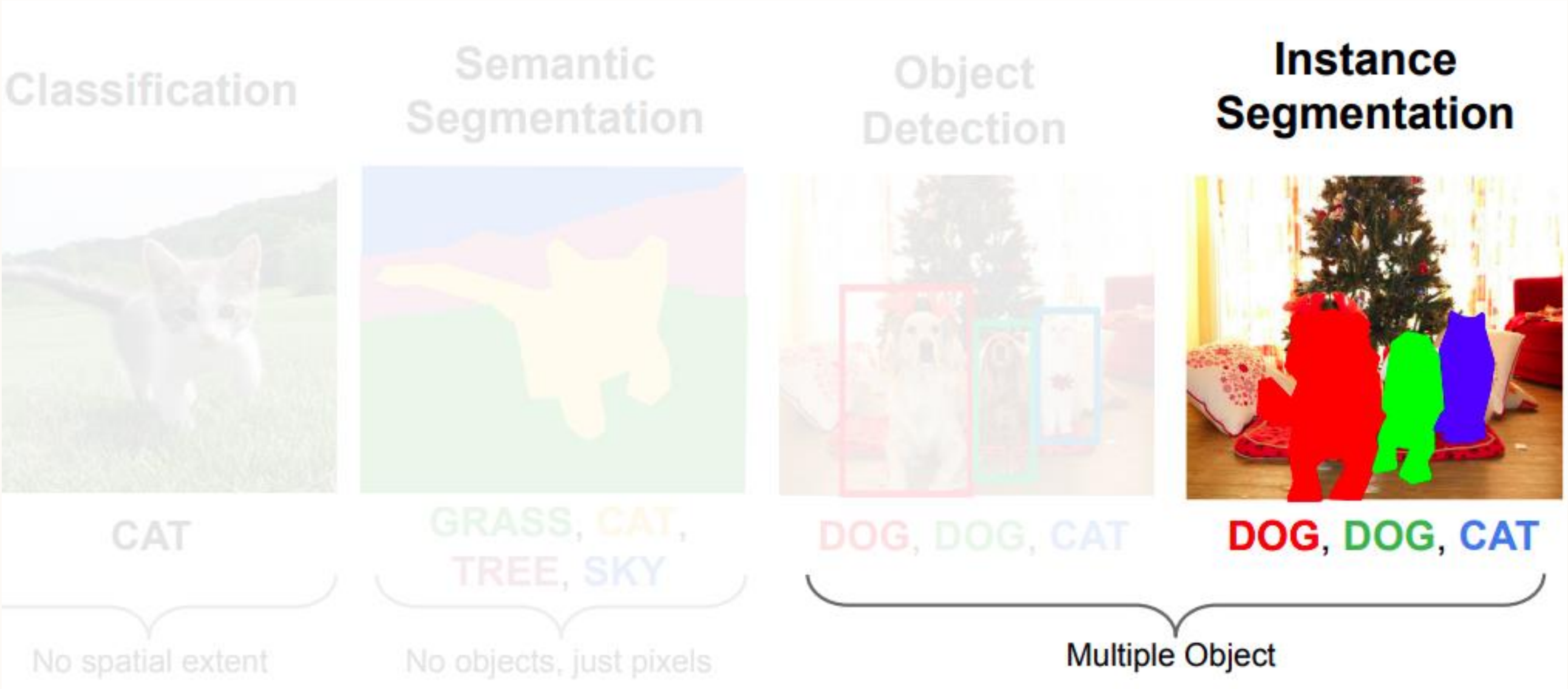
- Large distance, different intensities

Large negative values → very small penalty

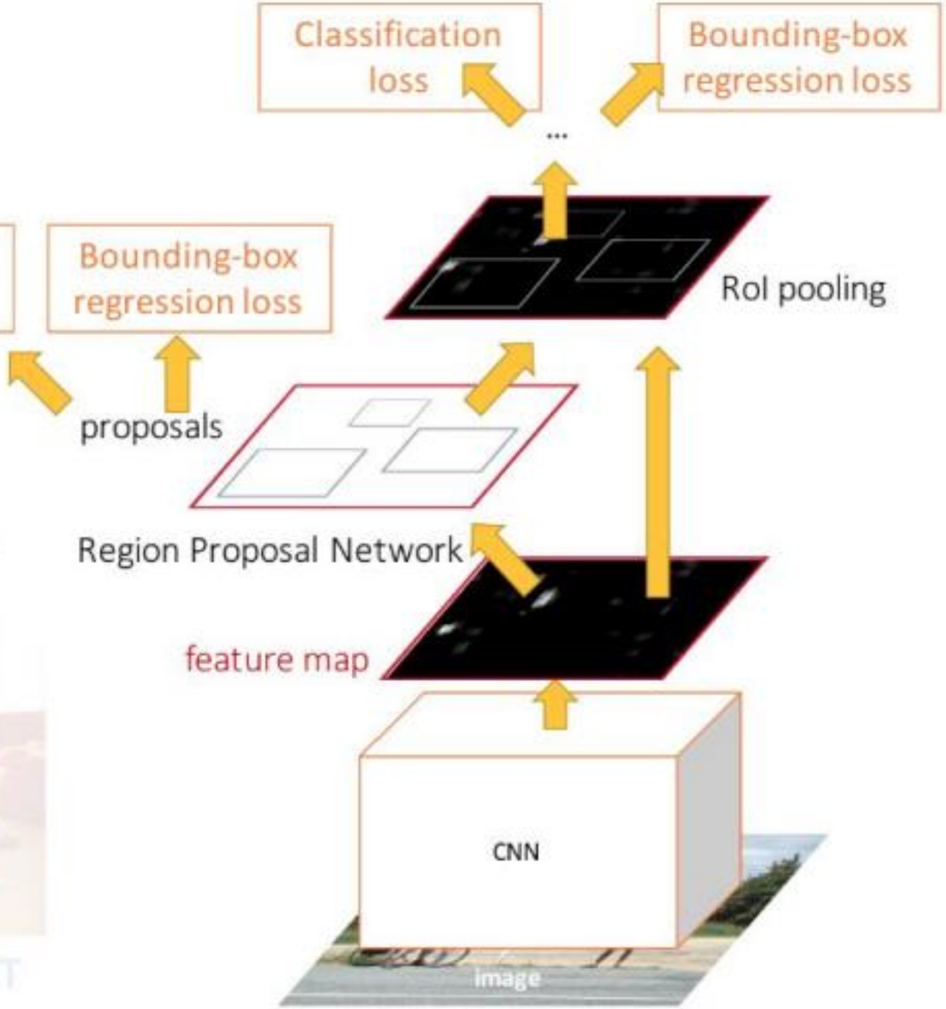


DEEPLAB-V3


Encoder
Image
DCNN
Atrous Conv
1x1 Conv
3x3 Conv rate 6
3x3 Conv rate 12
3x3 Conv rate 18
Image Pooling
1x1 Conv
Decoder
Low-Level Features
Upsample by 4
1x1 Conv
Concat
3x3 Conv
Upsample by 4
Prediction


Classification
Semantic Segmentation
Object Detection
Instance Segmentation
CAT
GRASS, CAT, TREE, SKY
DOG, DOG, CAT
DOG, DOG, CAT
No spatial extent
No objects, just pixels
Multiple Object

Object Detection: Faster R-CNN



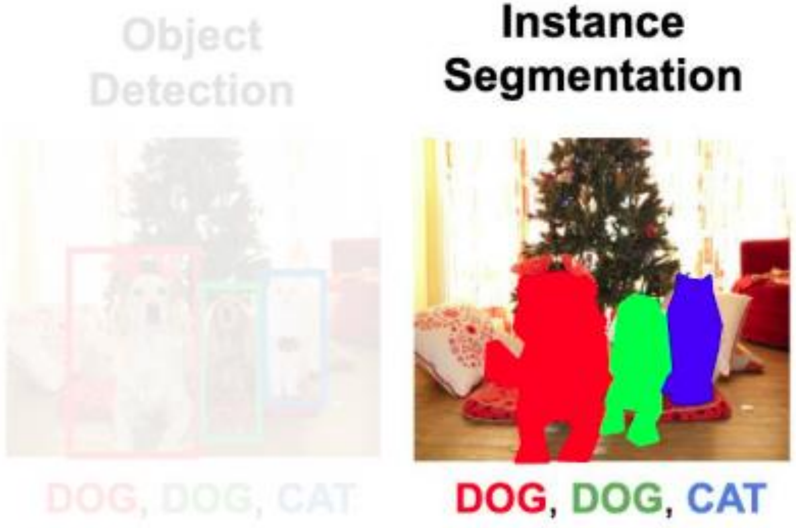
Object Detection

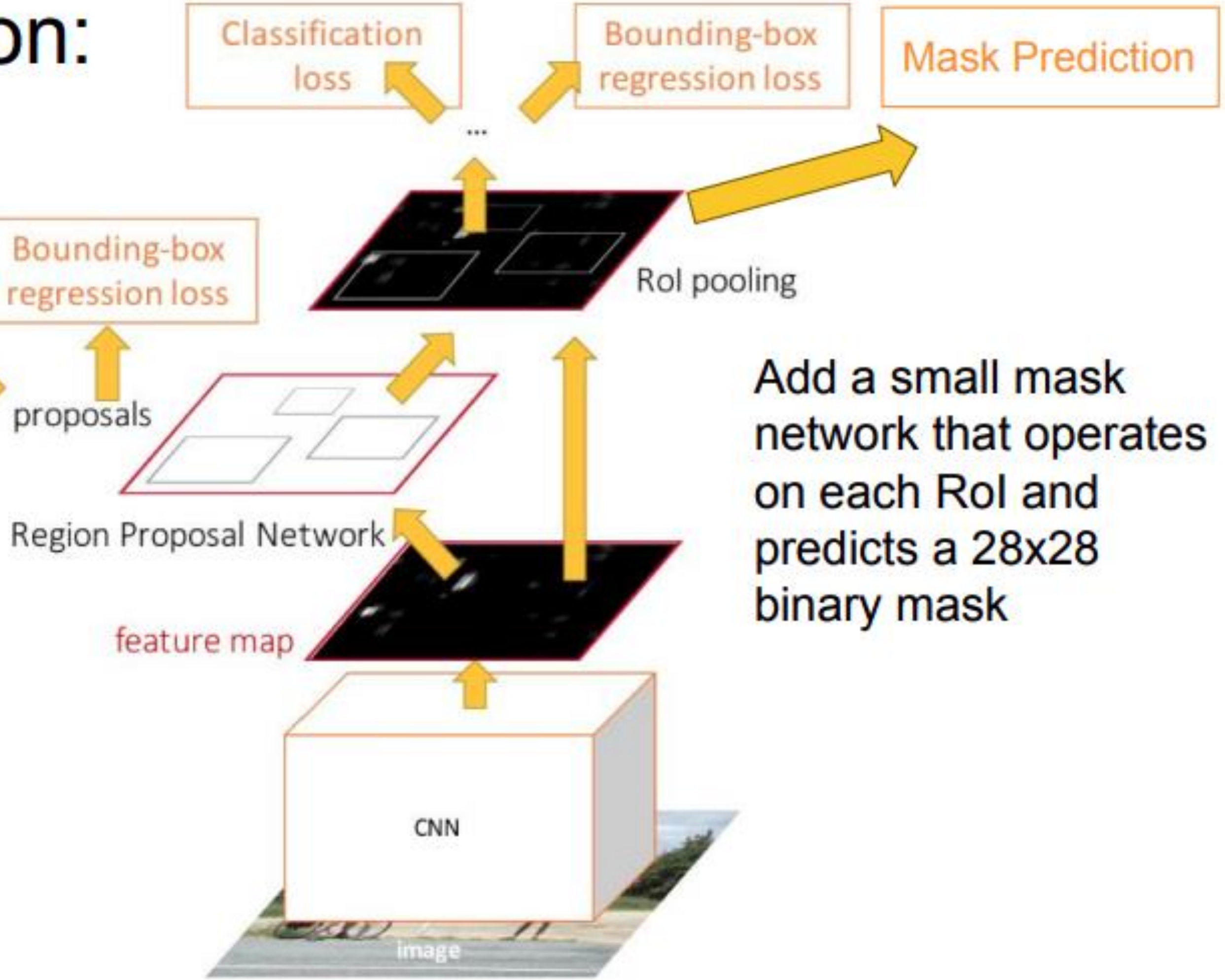
DOG, DOG, CAT

Instance Segmentation

DOG, DOG, CAT

Instance Segmentation: Mask R-CNN

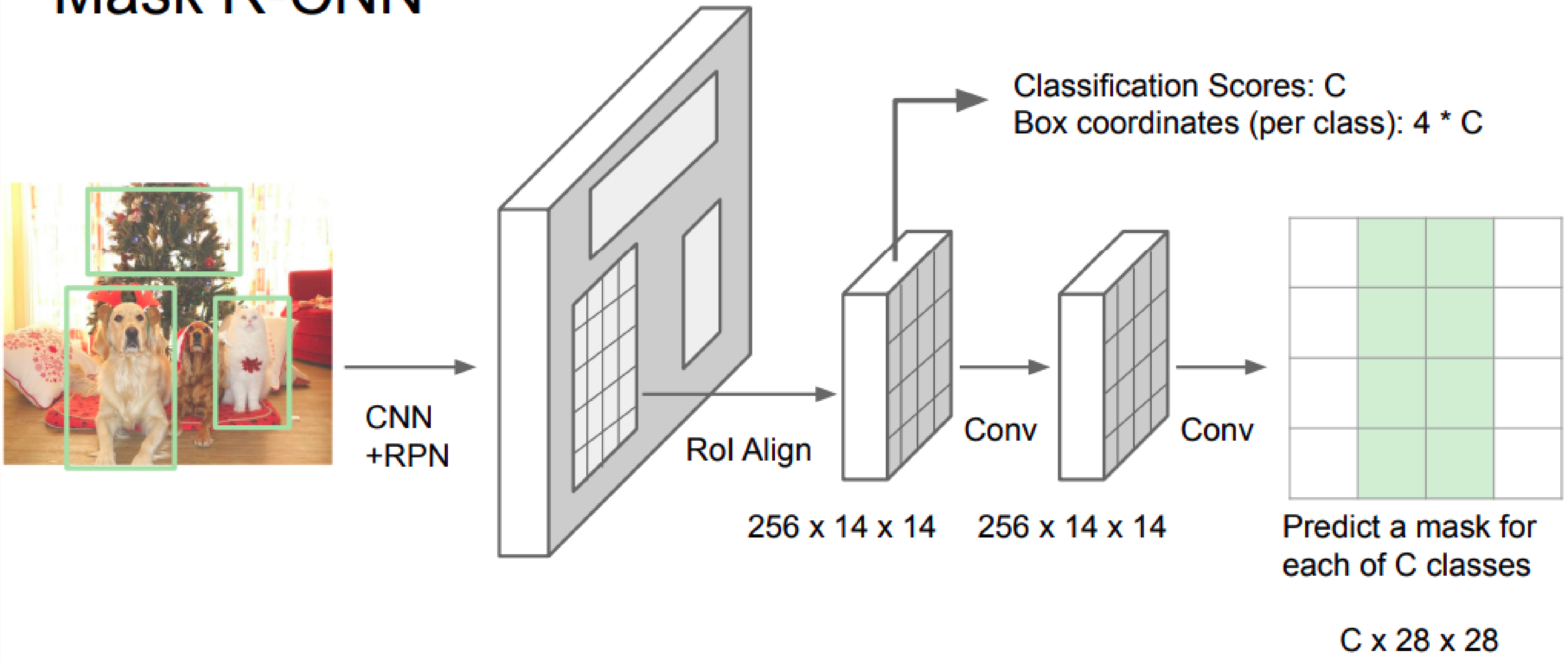




Add a small mask network that operates on each RoI and predicts a 28x28 binary mask

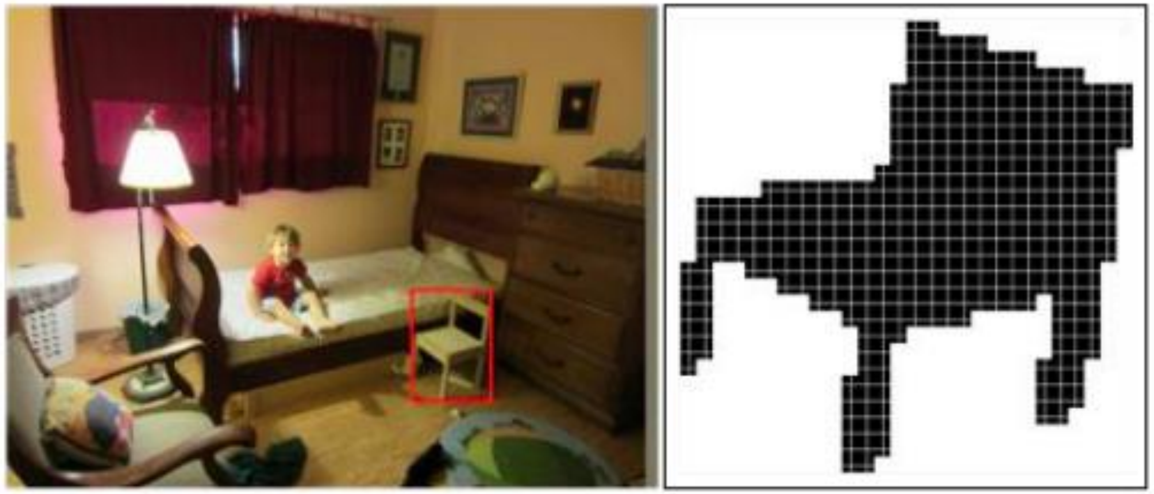
He et al, "Mask R-CNN", ICCV 2017

Mask R-CNN

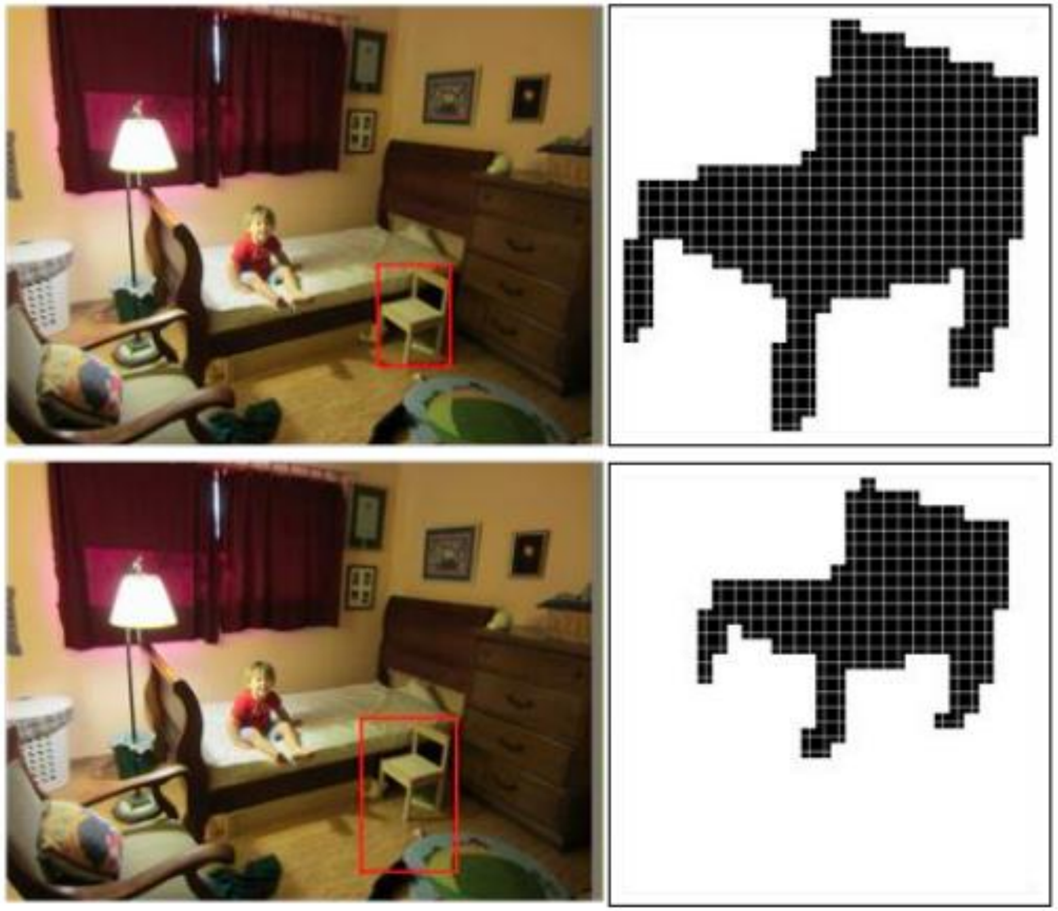


CNN
+RPN
RoI Align
Classification Scores: C
Box coordinates (per class): 4 * C
Conv
Conv
256 x 14 x 14
256 x 14 x 14
Predict a mask for
each of C classes
C x 28 x 28

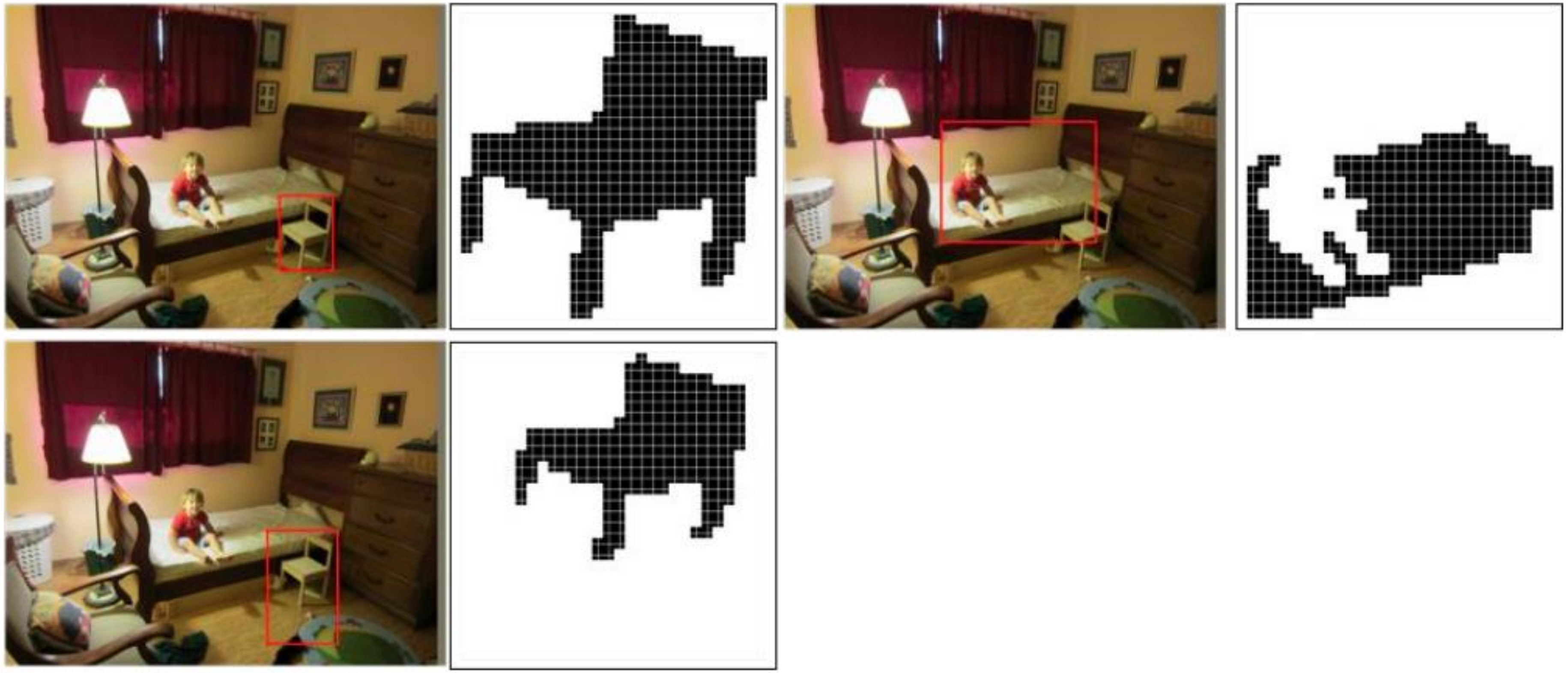
Mask R-CNN: Example Mask Training Targets



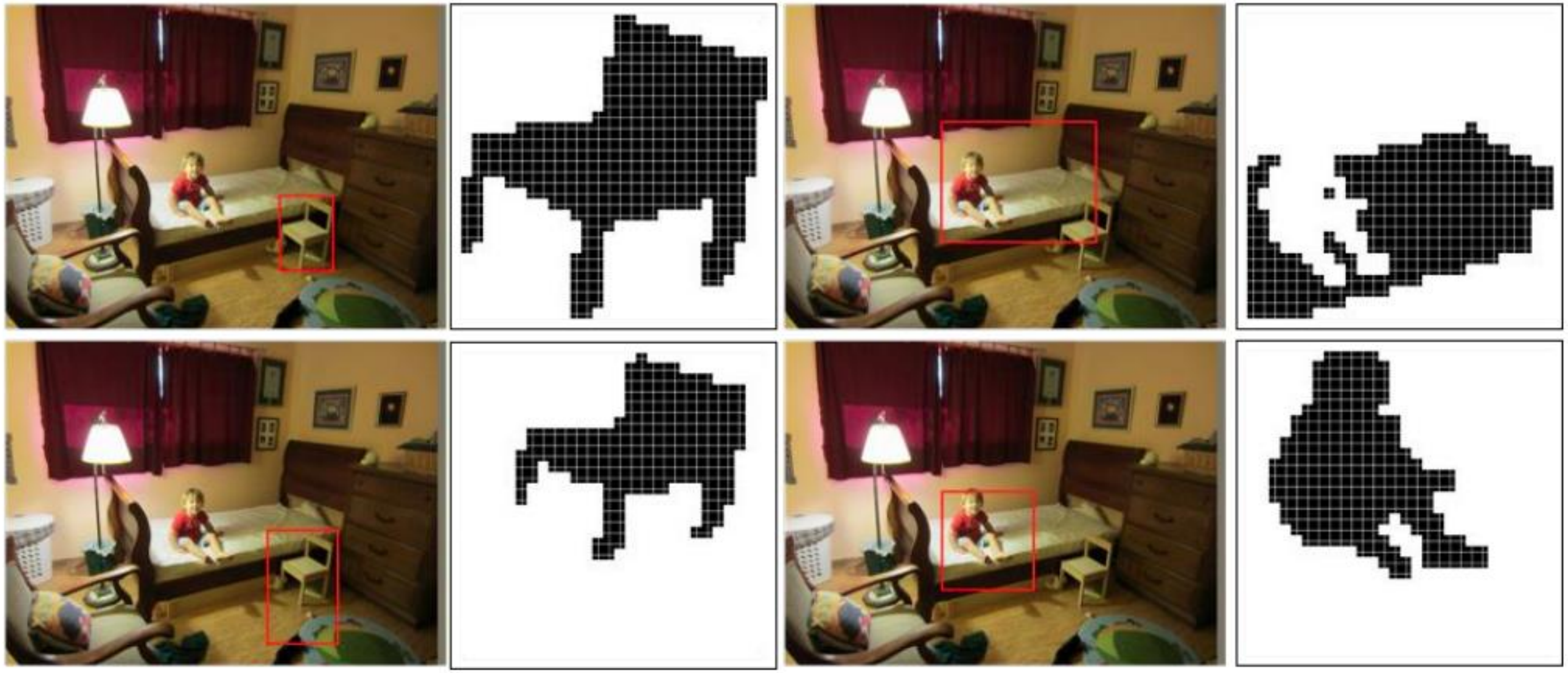
Mask R-CNN: Example Mask Training Targets



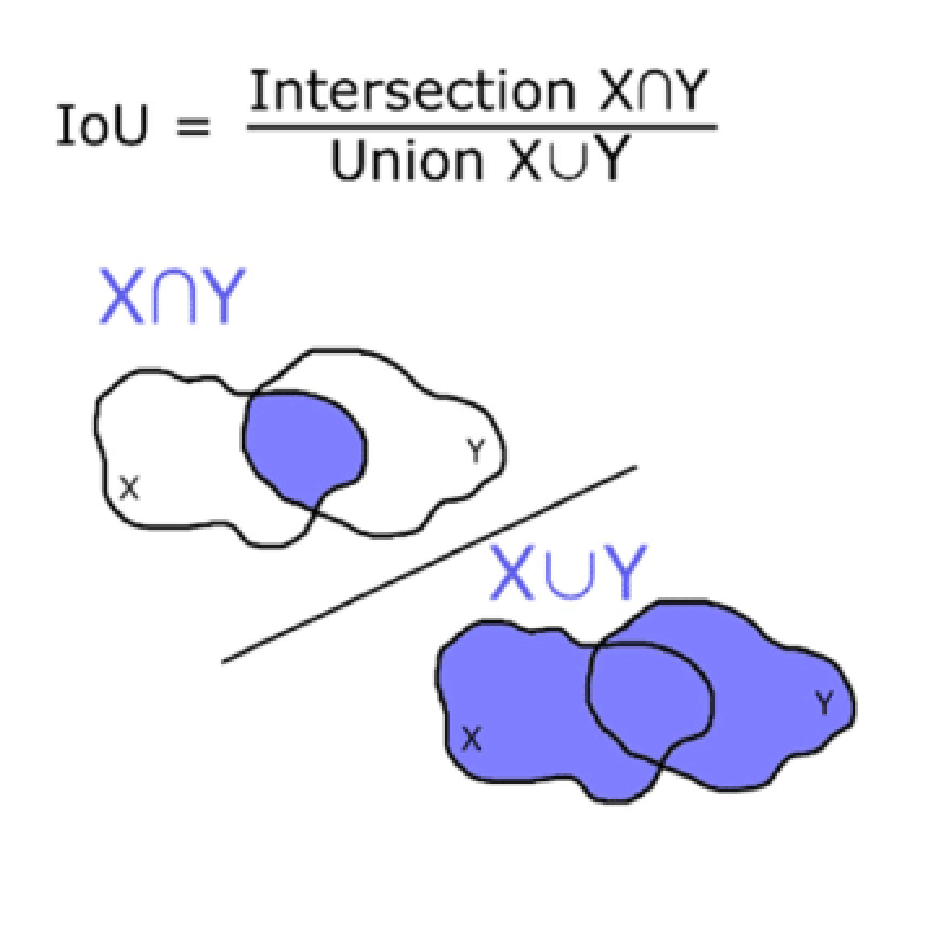
Mask R-CNN: Example Mask Training Targets



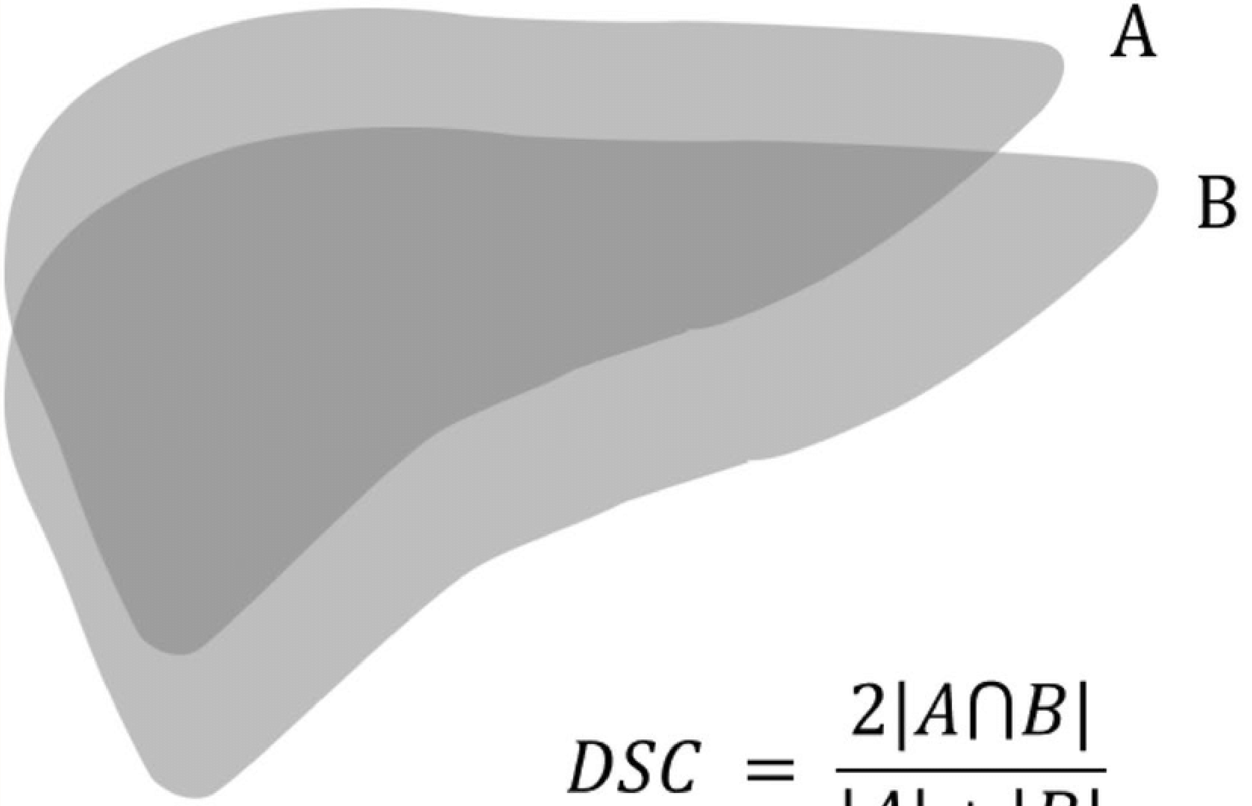
Mask R-CNN: Example Mask Training Targets



Evaluation metrics in segmentation



Evaluation metrics in segmentation



$$DSC = \frac{2|A \cap B|}{|A| + |B|}$$

DSC: Dice similarity coefficient

Evaluation metrics in segmentation

$$Dice\ Coefficient = \frac{2 * TP}{FN + (2 * TP) + FP}$$

$$Jaccard\ Index = \frac{TP}{TP + FN + FP}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

TP - true positive
TN - true negative
FP - false positive
FN - false negative

Manual Segmentation

Automated Segmentation