Semantic Segmentation

Dr. Konda Reddy Mopuri
Deep Learning for Computer Vision (DL4CV)
IIT Guwahati
Aug-Dec 2022

Semantic Segmentation

- Label each pixel in the image with a category
- No differentiation among multiple instances of the same category

Semantic Segmentation

Input

Output

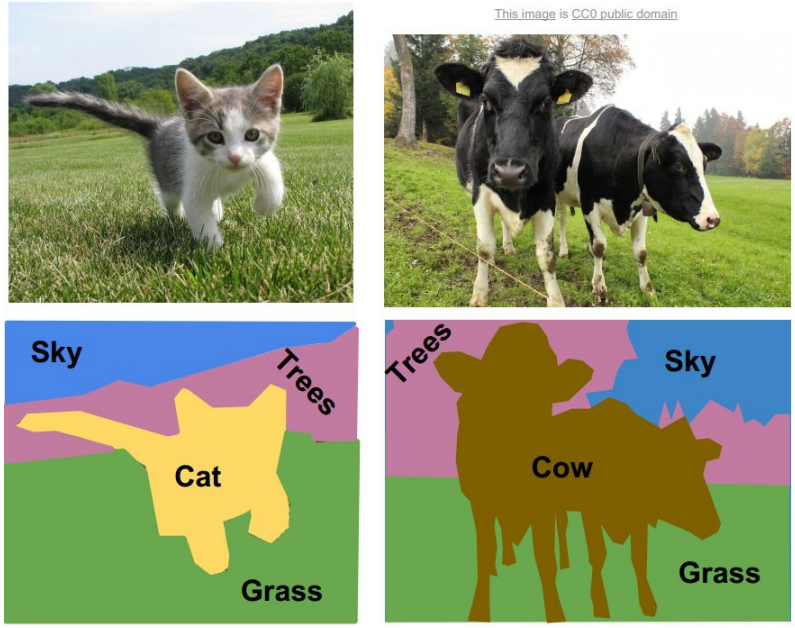


Figure credits CS231N, Stanford

Data labeling for semantic segmentation

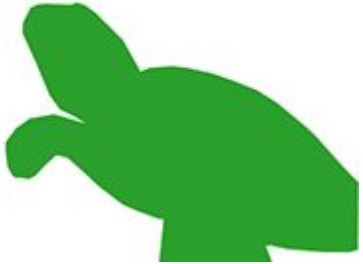
Input

Output

Figure credits: AWS Amazon

Data labeling for semantic segmentation

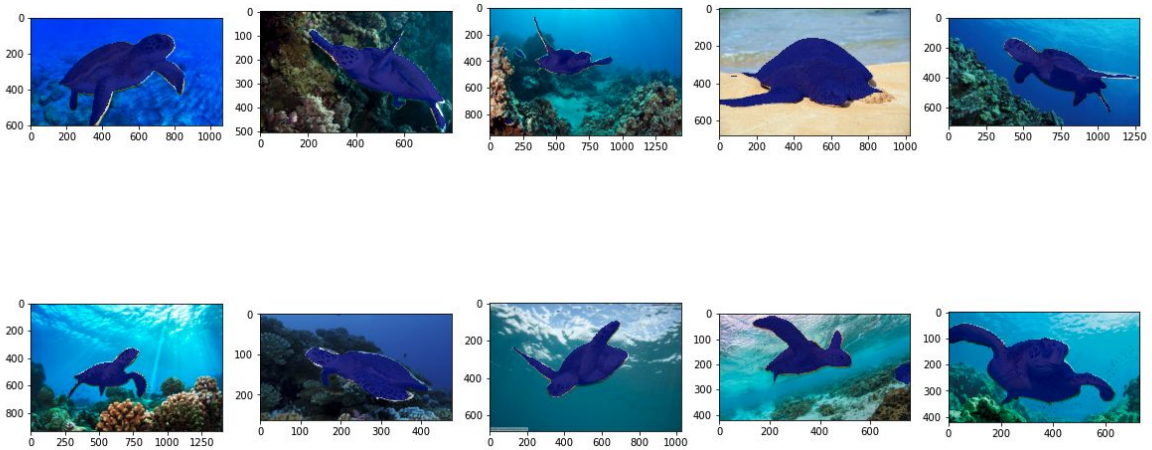
Input



Segmentation mask

Figure credits: AWS Amazon

Data labeling for semantic segmentation

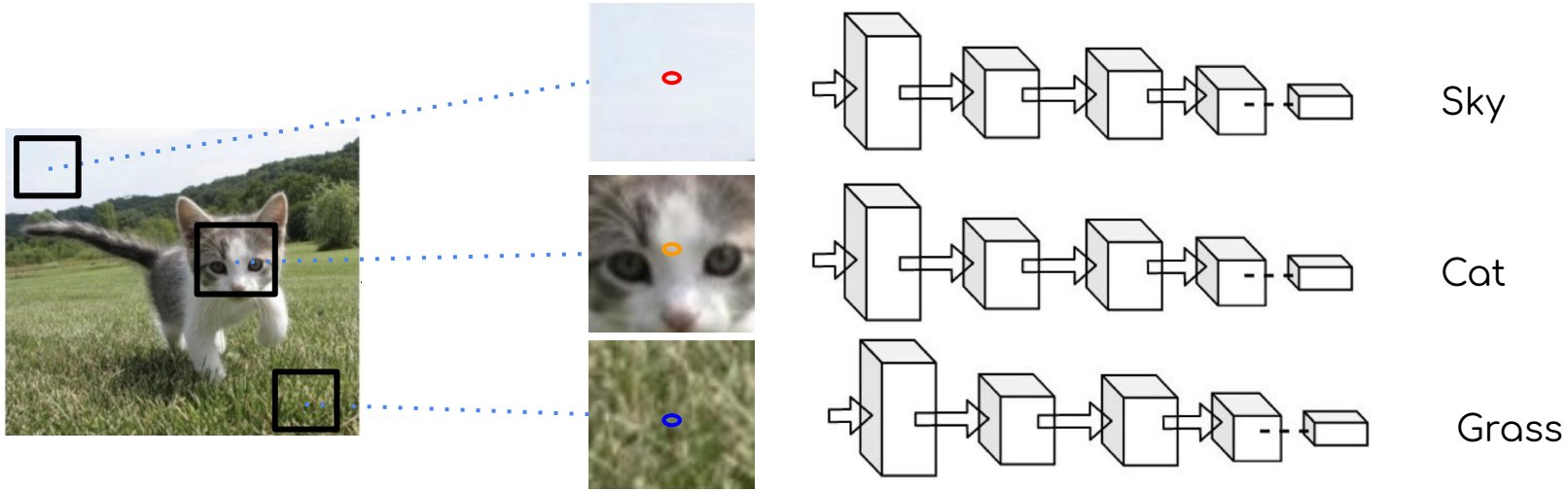


Inputs with Segmentation mask

Figure credits: AWS Amazon

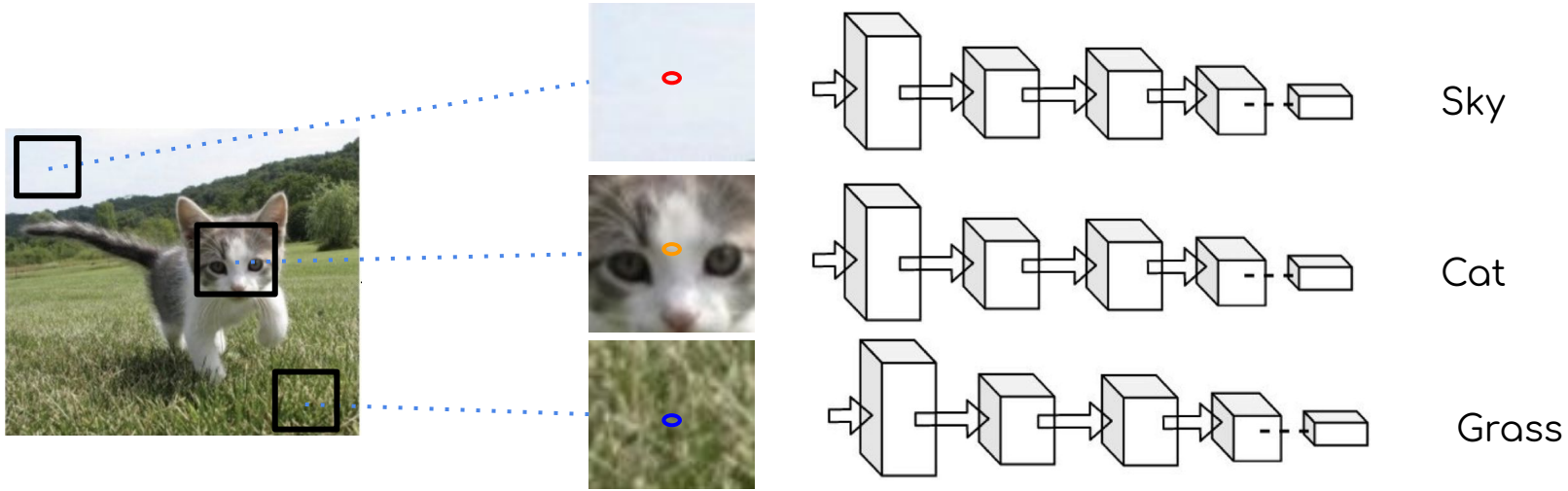
Semantic Segmentation: A Simple Approach

Extract patches



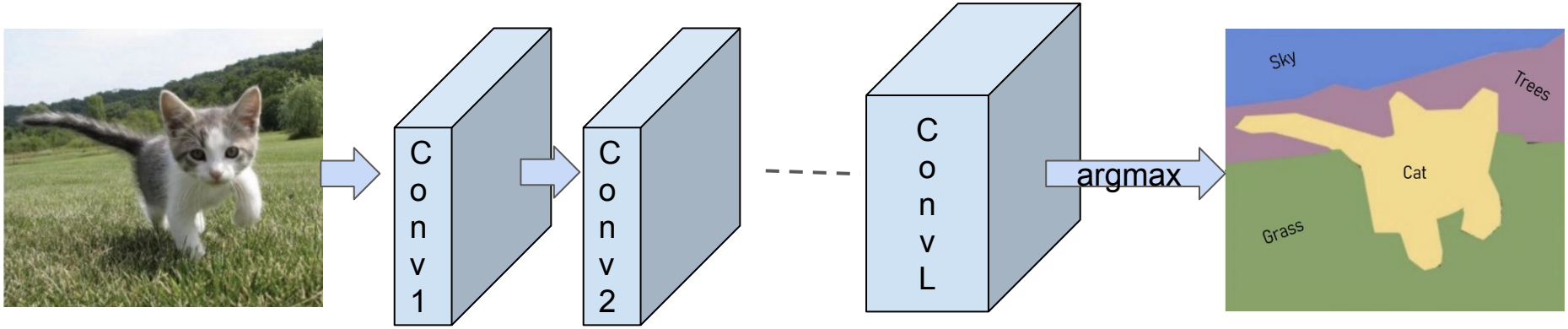
Semantic Segmentation: A Simple Approach

Extract patches



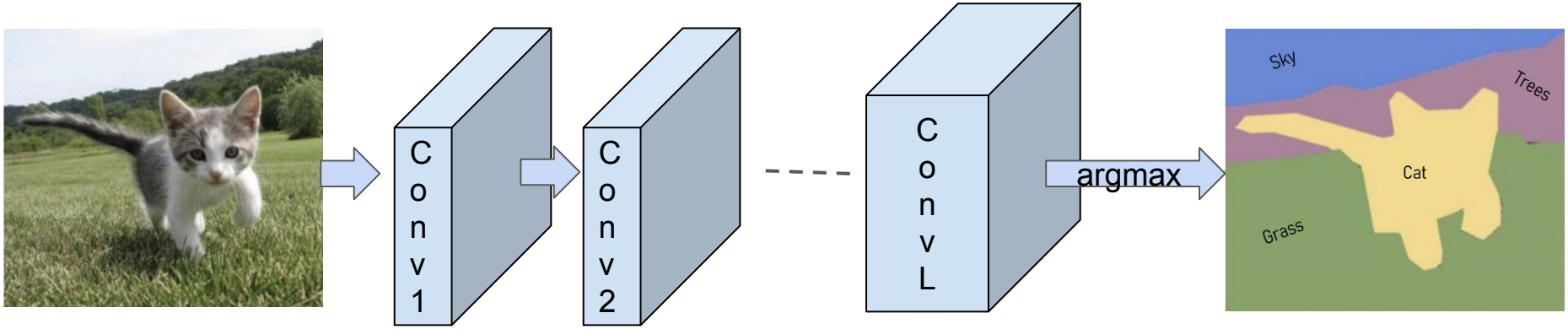
Very inefficient since it does not reuse features among neighboring patches!

Fully Convolutional Network (Lang et al. CVPR 2015)



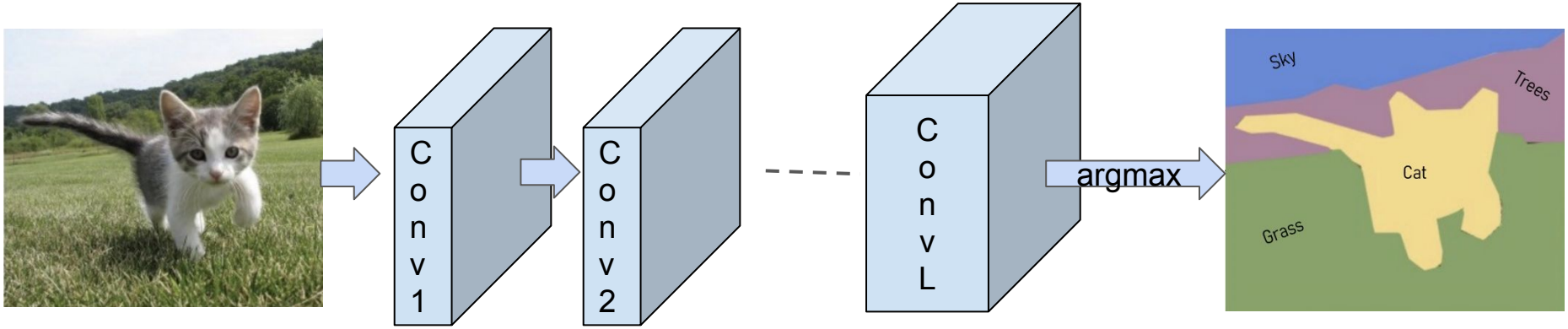
All conv layers; predictions per pixel simultaneously for all the pixels

Fully Convolutional Network (Lang et al. CVPR 2015)



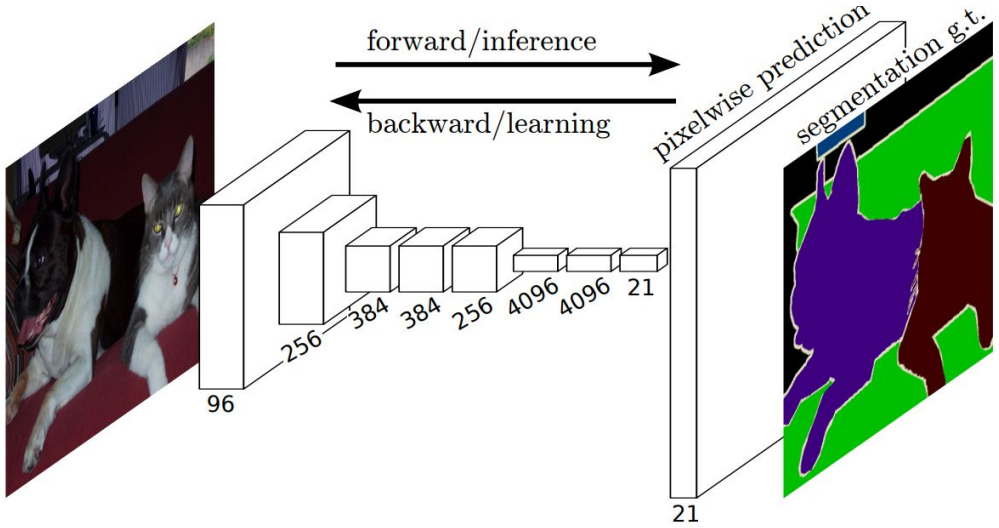
Loss: Cross-entropy per pixel

Fully Convolutional Network (Lang et al. CVPR 2015)



Issue: receptive field grows slowly with depth → needs more conv layers → processing high-res images is very expensive!

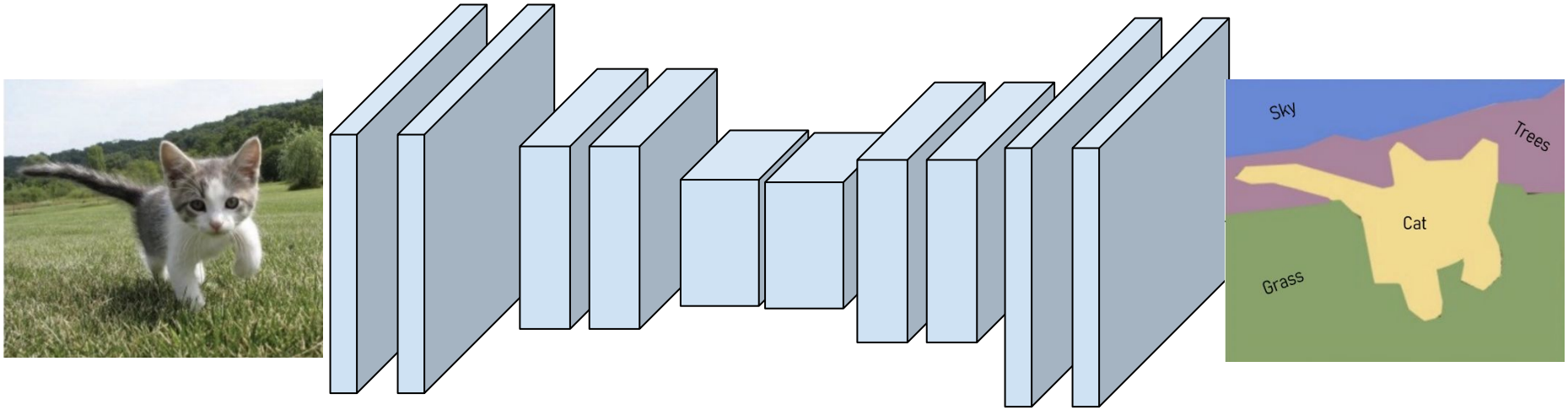
Fully Convolutional Network (Lang et al. CVPR 2015)



Use downsampling and upsampling inside the network

Deconvolution for Semantic Segmentation

(Noh et al. ICCV 2015)

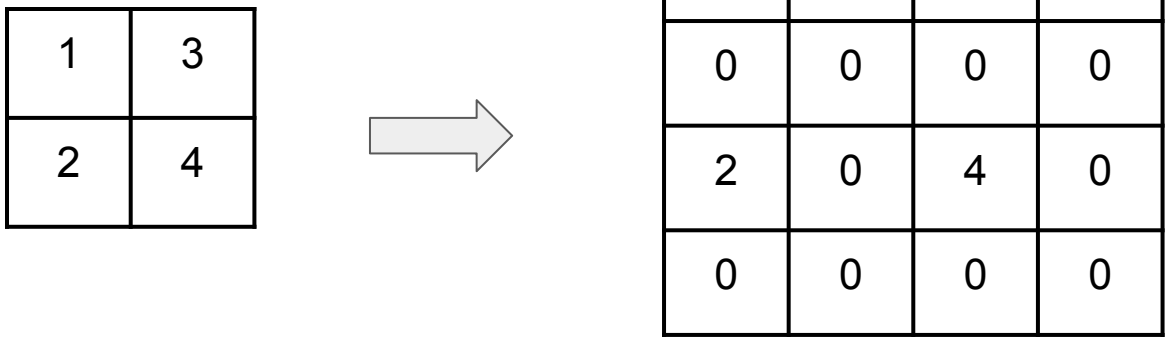


Use downsampling and upsampling inside the network

Upsampling, How ??

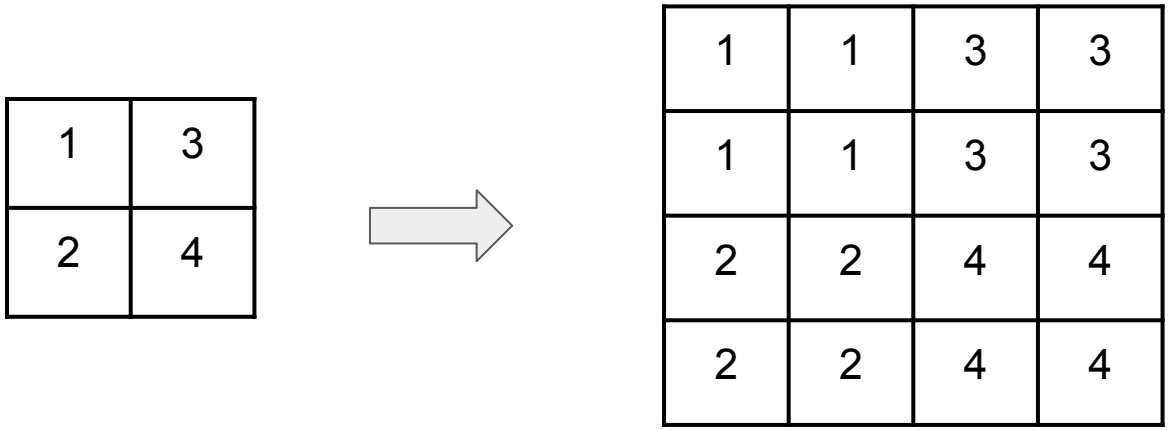
In-net upsampling

- Naive Unpooling (Bed of nails)



In-net upsampling

- Nearest Neighbor unpooling

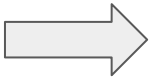


In-net upsampling

- Bilinear interpolation (use two closest neighbors and linear approx.)

1	3
2	4

Input



1	1.5	2.5	3
1.25	1.75	2.75	3.25
1.75	2.25	3.25	3.75
2	2.5	3.5	4

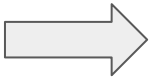
Output

In-net upsampling

- Bicubic interpolation (use 3 nearest neighbors, cubic approx.)

1	3
2	4

Input

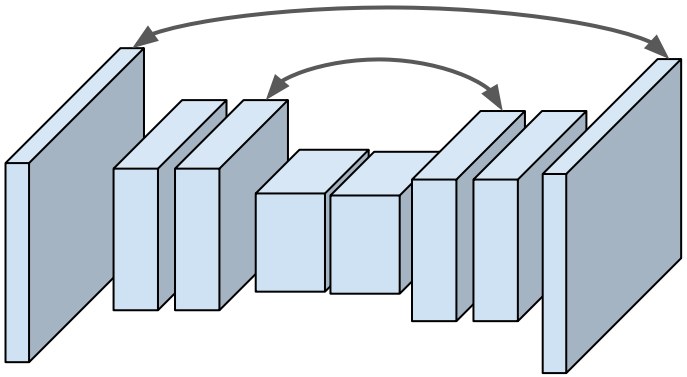


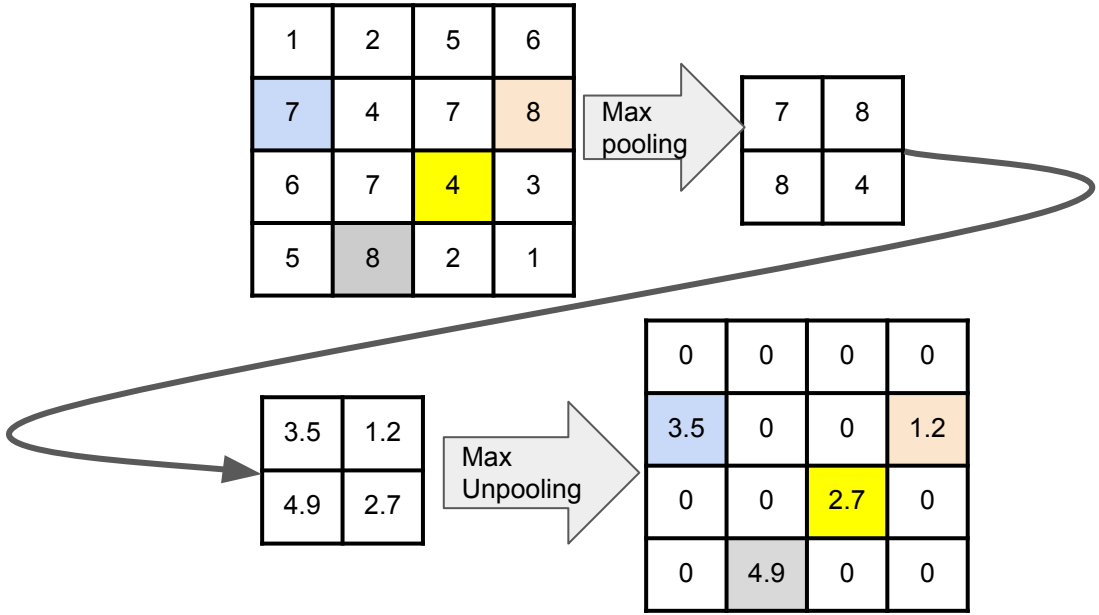
.68	1.35	2.44	3.11
1.02	1.68	2.77	3.44
1.56	2.23	3.32	3.98
1.89	2.56	3.65	4.32

Output

In-net upsampling

- Max Unpooling (Remembers the position of max values)





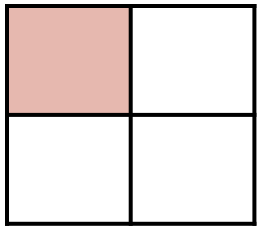
In-net upsampling

- So far, these approaches have no learnable params
- Learnable upsampling → Transposed Convolution

Transposed Convolution

- Convolution reduces the size (division by stride)
- Same Convolution can increase if the stride < 1

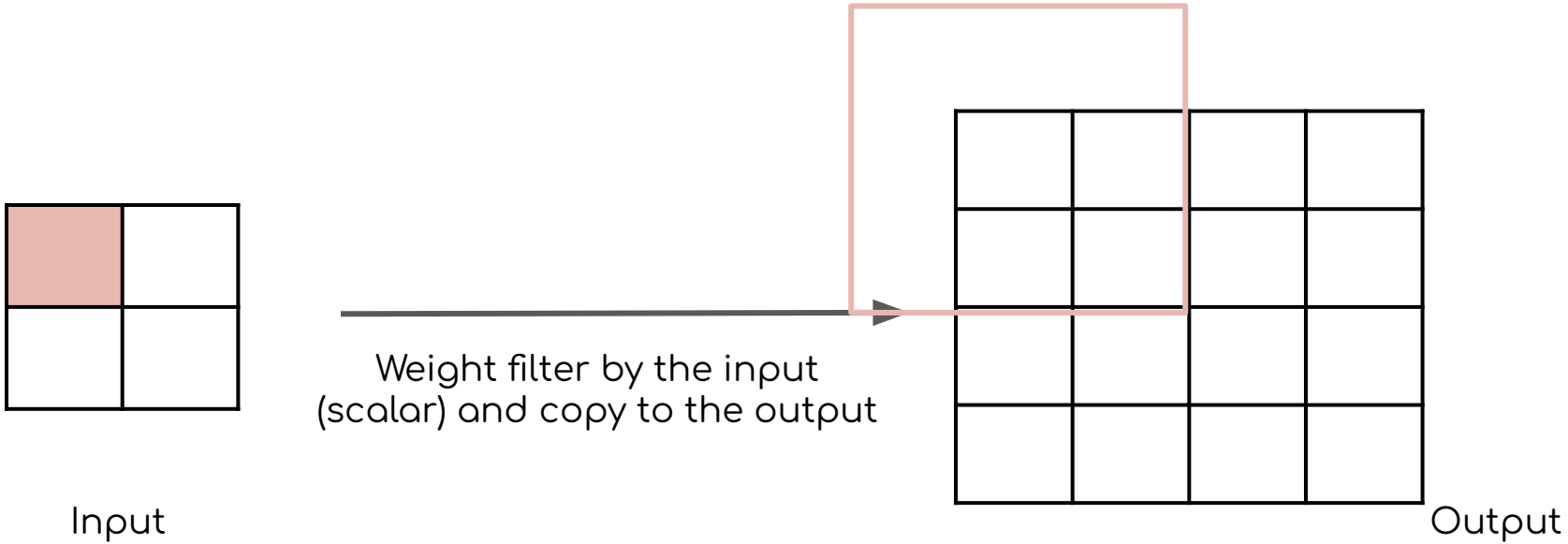
Transposed Convolution



Input

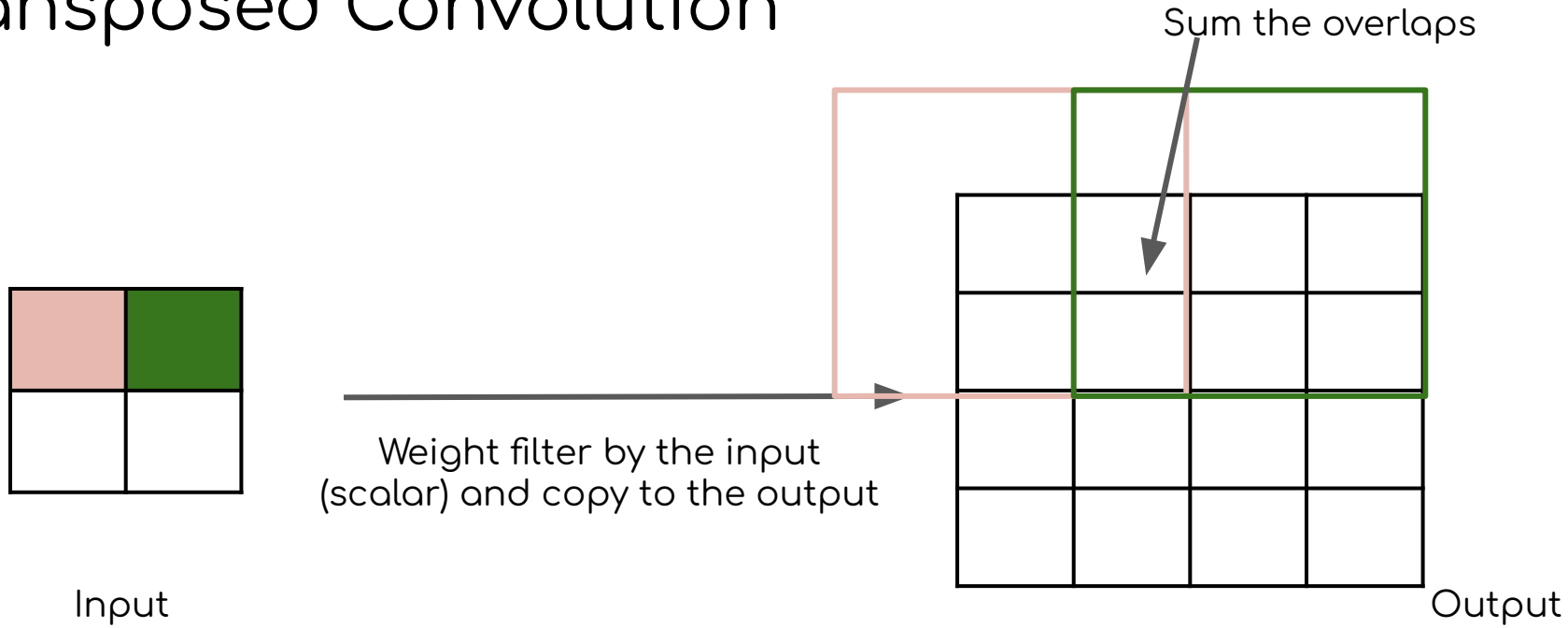
Filter is 3 X 3 convolution transpose, stride 2

Transposed Convolution



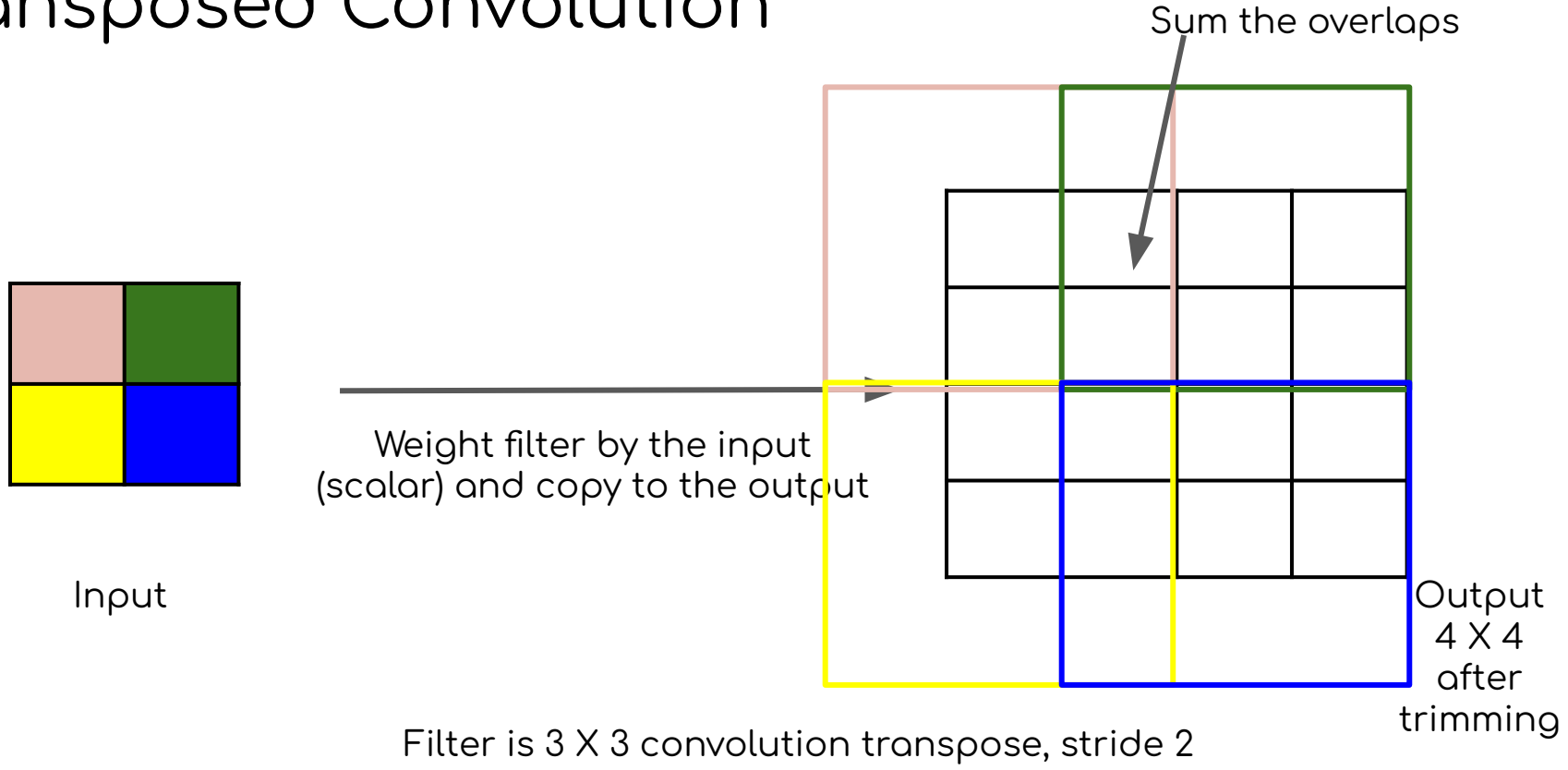
Filter is 3 X 3 convolution transpose, stride 2

Transposed Convolution



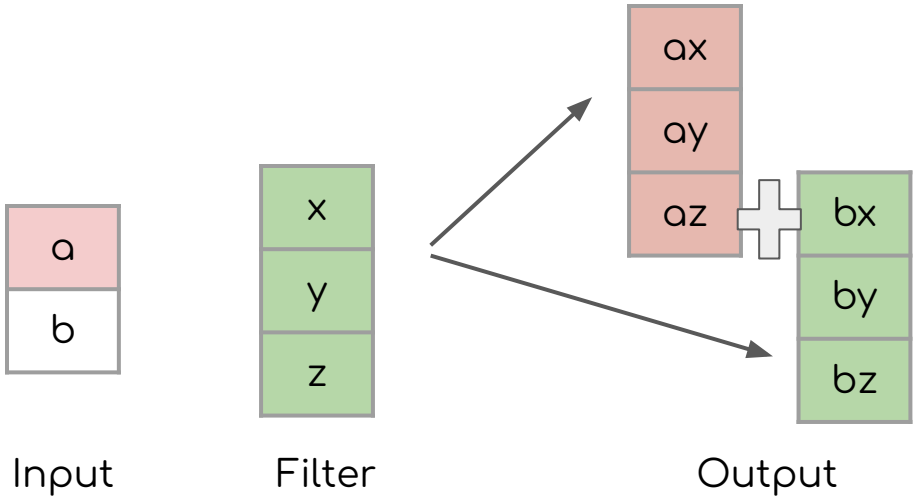
Filter is 3 X 3 convolution transpose, stride 2

Transposed Convolution



Filter is 3 X 3 convolution transpose, stride 2

Transposed Convolution - 1D Example

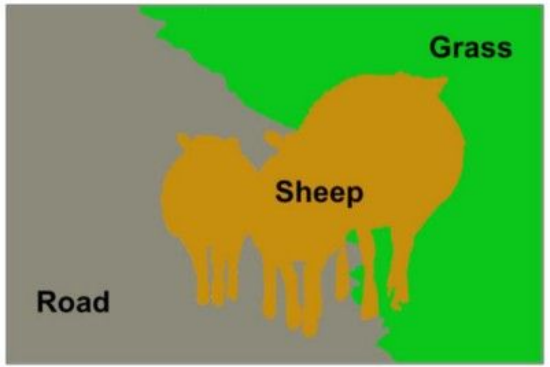


Transposed Convolution

- Different names
  - Deconvolution
  - Upconvolution
  - Convolution with fractional stride
  - Convolution with backward stride
  - Transposed Convolution

Semantic Segmentation

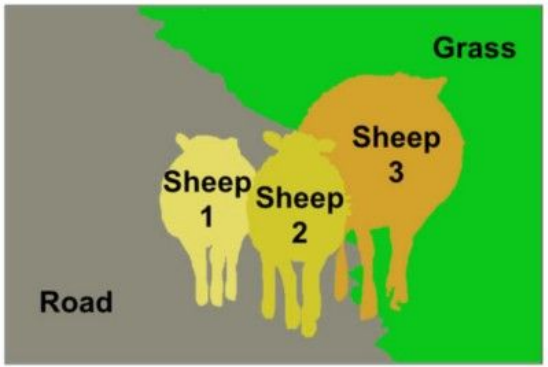
- Gives per-pixel labels
- Merges different instances of the objects
- Stuff vs things
  - Things: Categories that can be separated into instances (cat, dog, cow, person, etc.)
  - Stuff: Categories that can't be separated into instances (sky, grass, trees, etc.)



Semantic Segmentation

Instance Segmentation

- Detect all the objects in the image
- Identify pixels that belong to each object (only for things!)

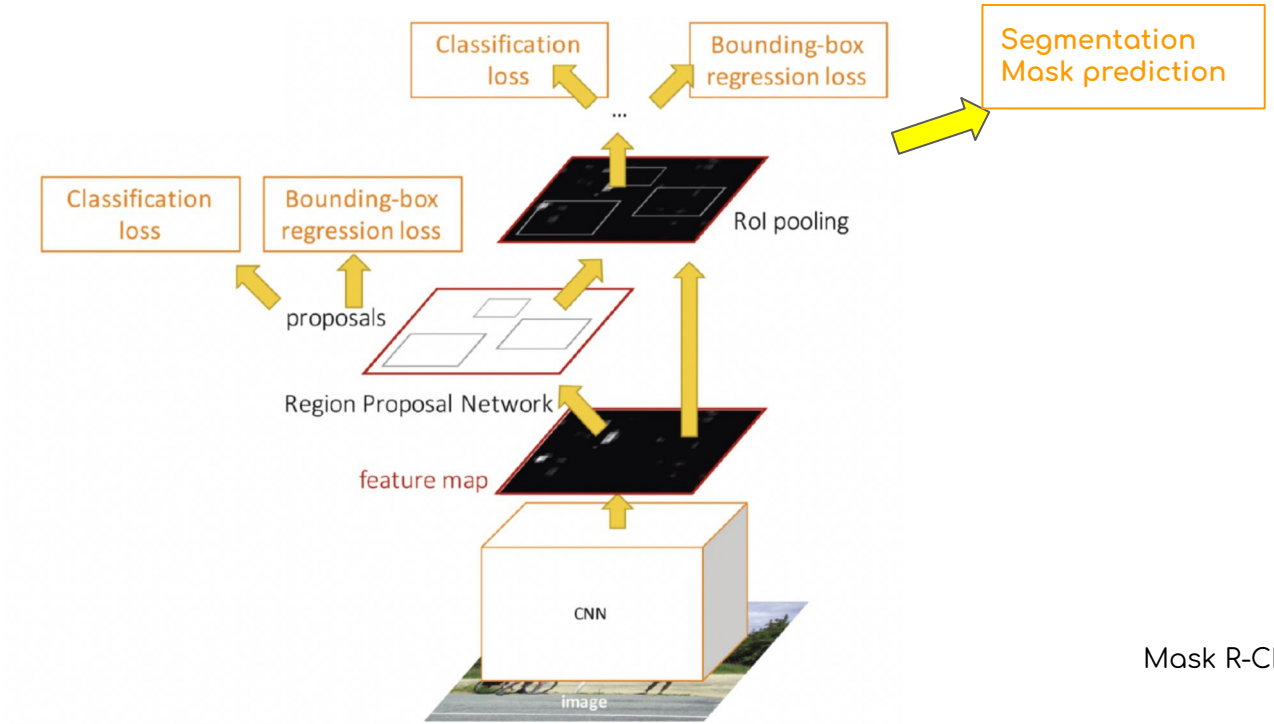


Instance Segmentation

Instance Segmentation: Approach

1. Object detection
2. Semantic segmentation for each of the detected objects

Faster R-CNN



Mask R-CNN, ICCV 2017

Next class

- Video tasks