Computer Vision

Image Segmentation

16%

Naeemullah Khan
naeemullah.khan@kaust.edu.sa

KAUST Academy
King Abdullah University of Science and Technology

January 28, 2026

1. Data →
2. Models
3. Error | Loss →
4. Optimisation.

correct ✓

Computer Vision

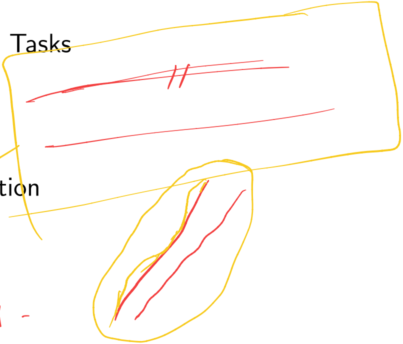
Naeemullah Khan
naeemullah.khan@kaust.edu.sa

KAUST Academy
King Abdullah University of Science and Technology

January 28, 2026

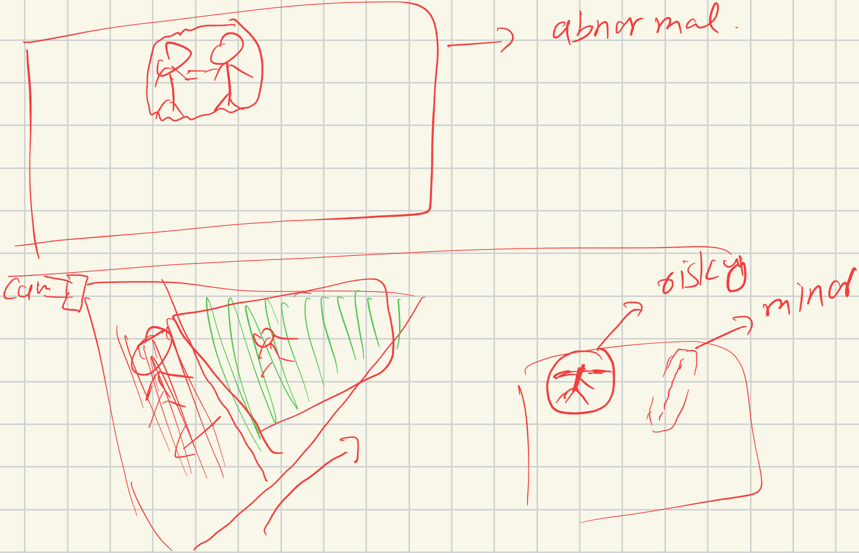
Table of Contents

KAUST ACADEMY أكاديمية كاوست



crack

Image Segmentation.


abnormal.
Cam
risky
minor

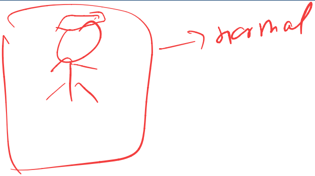
Learning Outcomes

- Understand the fundamentals of image segmentation and its importance.
- Understand how Convolutional Neural Networks (CNNs) are adapted for segmentation tasks.
- Understand different upsampling techniques used in segmentation models.
- Understand the role of residual connections and the U-Net architecture in segmentation.
- Differentiate between instance segmentation and panoptic segmentation.

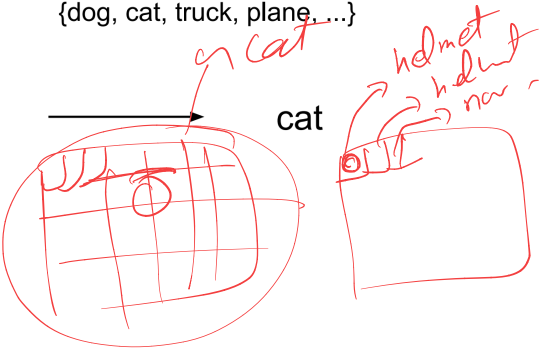
Image Classification

أكاديمية كاوست
KAUST ACADEMY

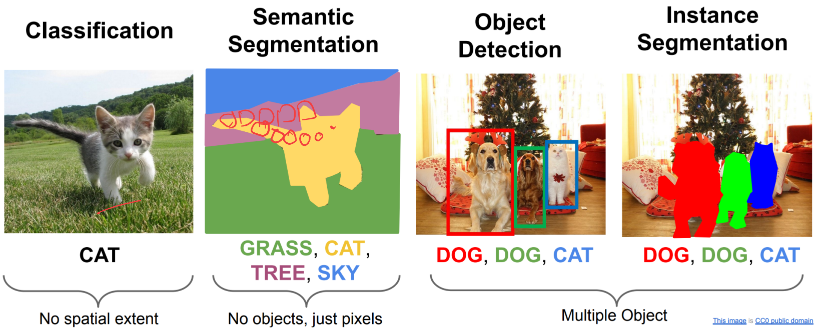
- Previously, we discussed Image Classification
- A core task in Computer Vision



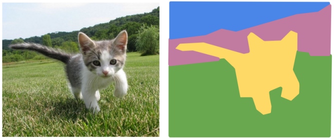
(assume given a set of possible labels)
{dog, cat, truck, plane, ...}



Computer Vision Tasks

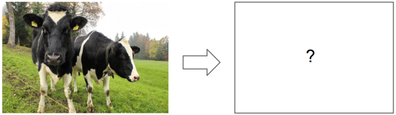


Semantic Segmentation

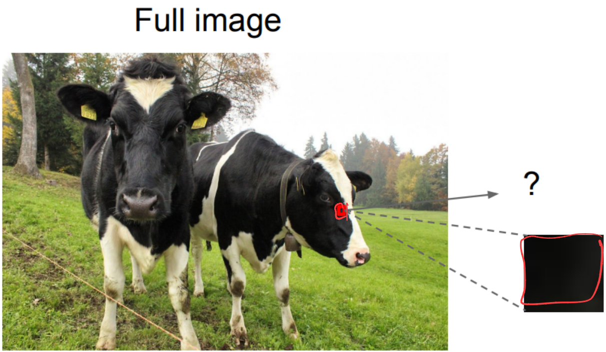


GRASS, CAT,
TREE, SKY, ...

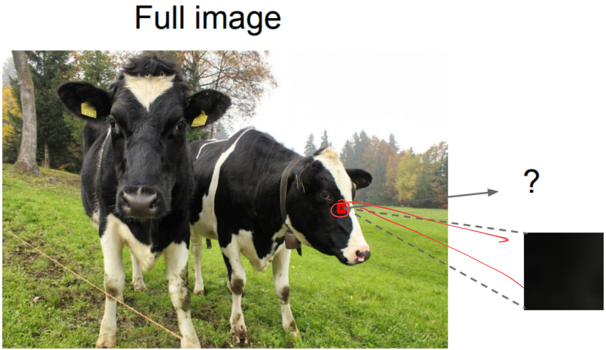
Paired training data: for each training image, each pixel is labeled with a semantic category.



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

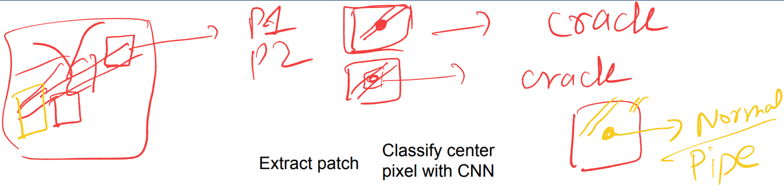

Full image
?

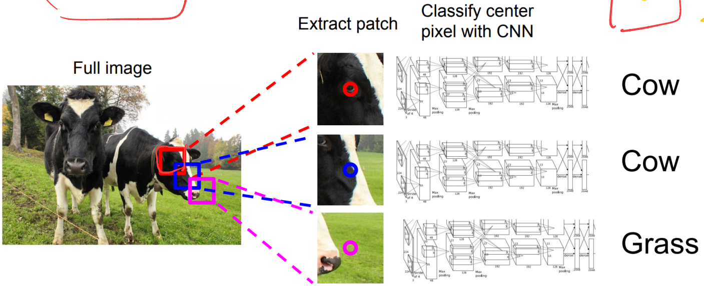
Semantic Segmentation



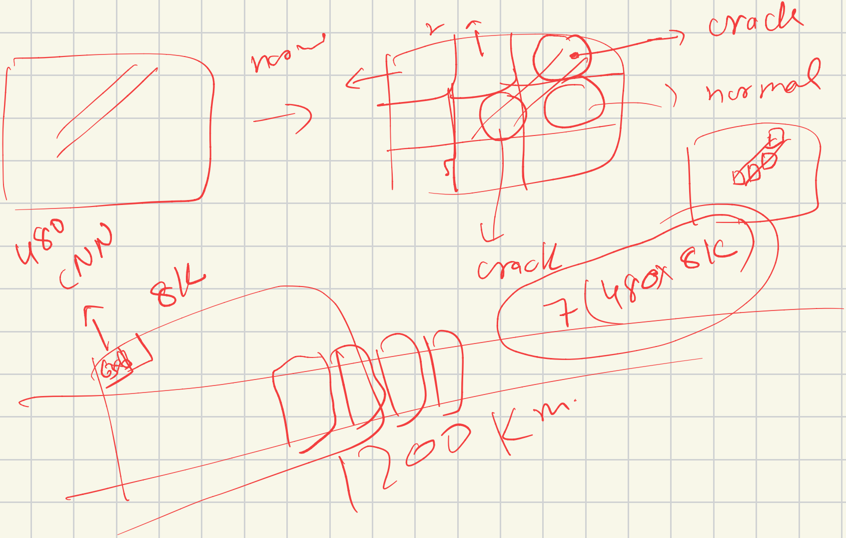
- Impossible to classify without context
- How do we include context?

Semantic Segmentation Idea: Sliding Window

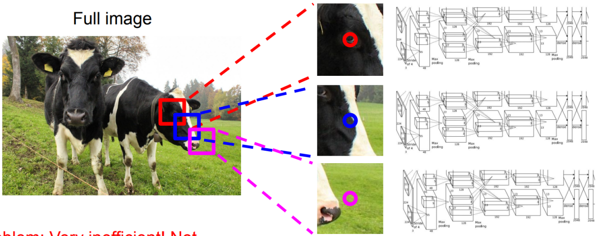




Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014


now
crack
normal
crack
480
CNN
8K
7(480x8K)
1200 Km.

Semantic Segmentation Idea: Sliding Window (cont.)

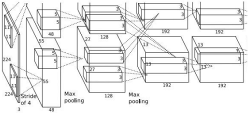


Problem: Very inefficient! Not reusing shared features between overlapping patches

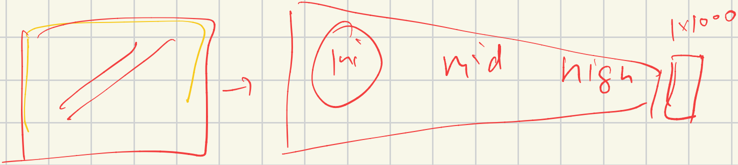
Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

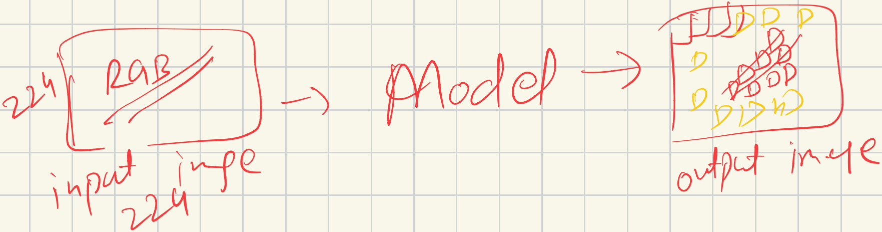
Semantic Segmentation Idea: Convolution

Full image



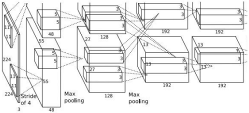
An intuitive idea: encode the entire image with conv net, and do semantic segmentation on top.


mid
high
1x1000


224
RGB
input
224
Model
output

Semantic Segmentation Idea: Convolution (cont.)

Full image

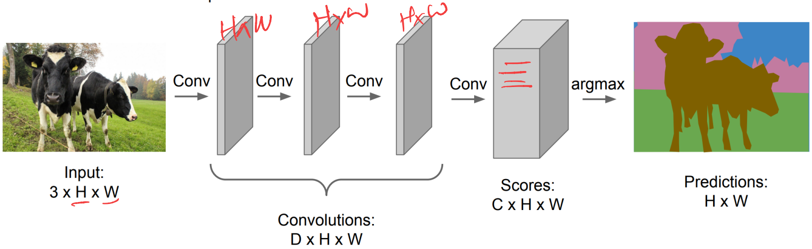


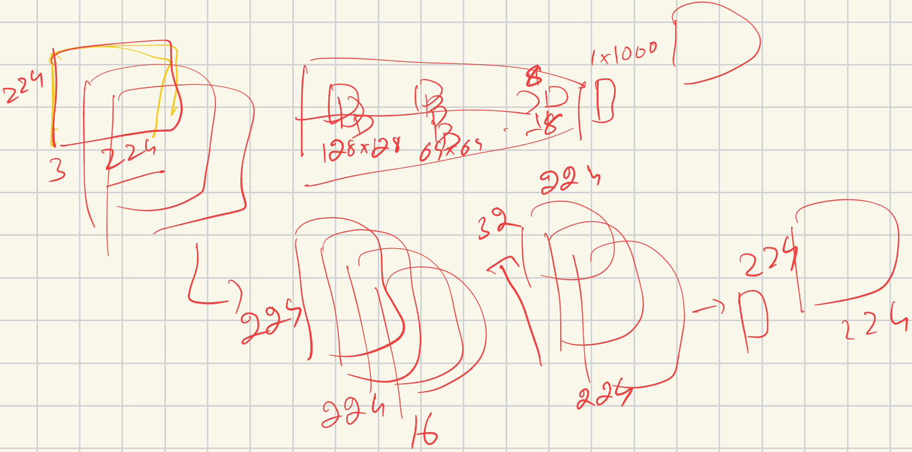
An intuitive idea: encode the entire image with conv net, and do semantic segmentation on top.

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.

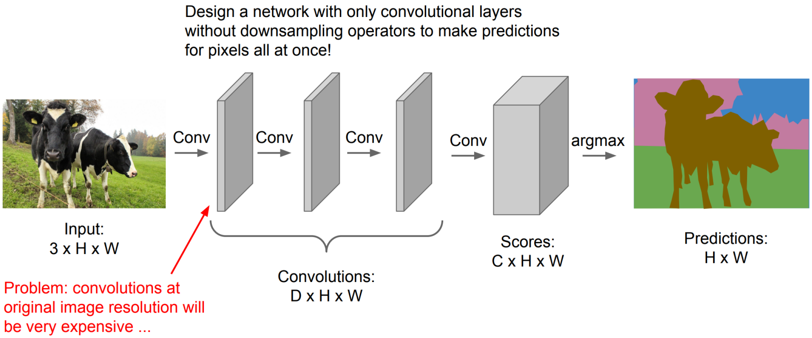
Semantic Segmentation Idea: Fully Convolutional

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




224
3
224
1x1000
128x128
64x64
8
224
32
224
224
16
224
224
224

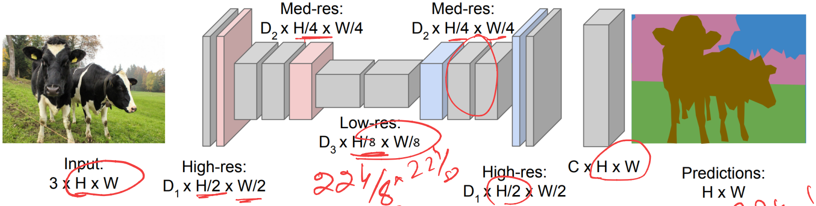
Semantic Segmentation Idea: Fully Convolutional (cont.)



Semantic Segmentation Idea: Fully Convolutional (cont.)

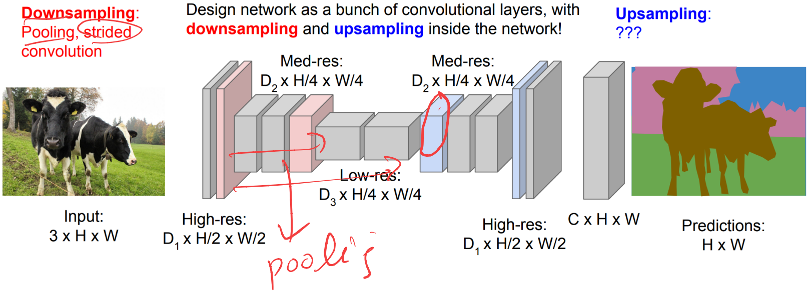
KAUST ACADEMY أكاديمية كاوست

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



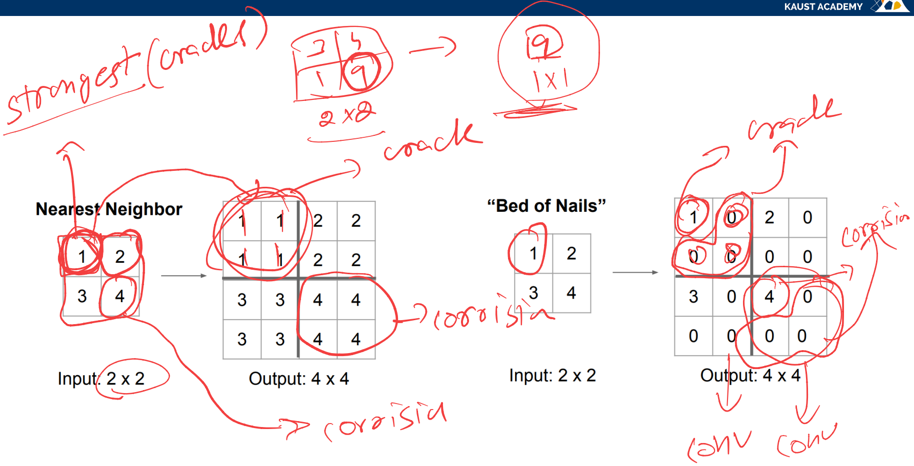
Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Semantic Segmentation Idea: Fully Convolutional (cont.)

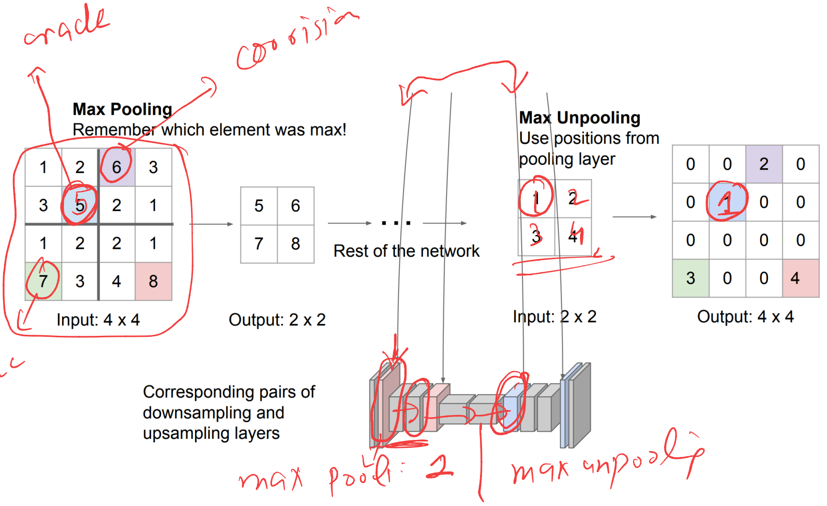


Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

In-Network Upsampling: Unpooling

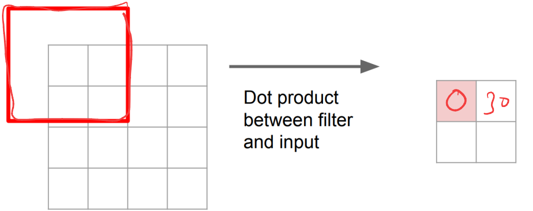


In-Network Upsampling: Max Unpooling



Learnable Upsampling: Transposed Convolution

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

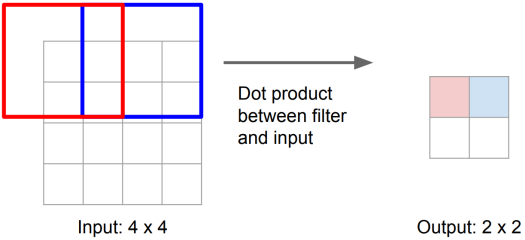


Input: 4 x 4

Output: 2 x 2

Learnable Upsampling: Transposed Convolution (cont.)

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



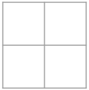
Filter moves 2 pixels in the input for every one pixel in the output

Stride gives ratio between movement in input and output

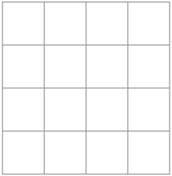
We can interpret strided convolution as "learnable downsampling".

Learnable Upsampling: Transposed Convolution (cont.)

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

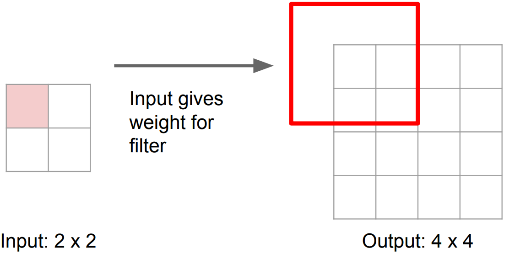


Input: 2 x 2



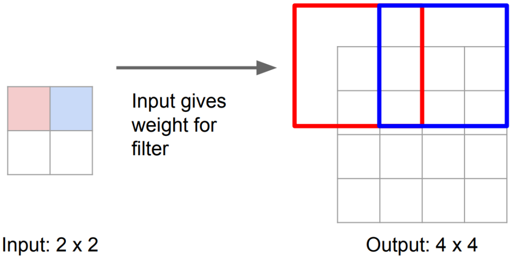
Output: 4 x 4

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



Learnable Upsampling: Transposed Convolution (cont.)

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

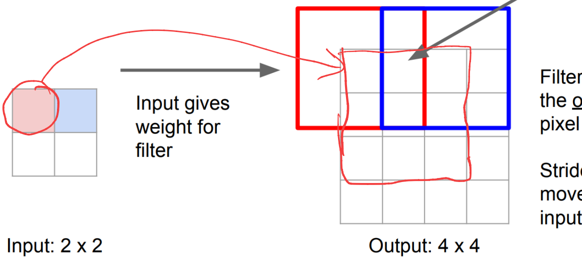


Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

Learnable Upsampling: Transposed Convolution (cont.)

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



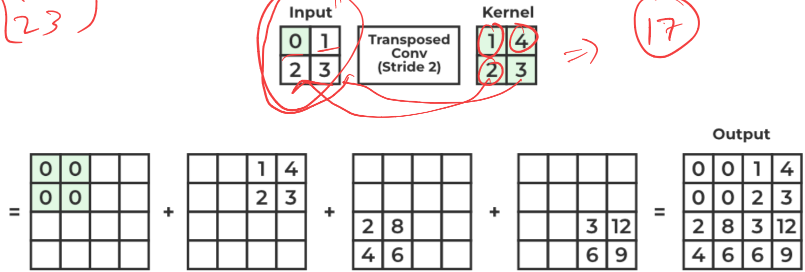
Sum where output overlaps

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

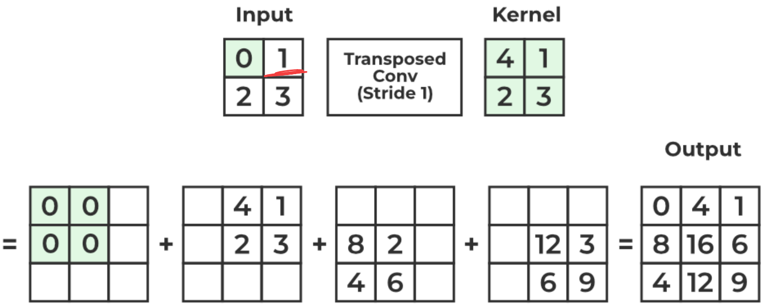
Stride gives ratio between movement in output and input

Learnable Upsampling: Transposed Convolution (cont.)

KAUST ACADEMY



Learnable Upsampling: Transposed Convolution (cont.)



Mathematical Definition: Transposed Convolution

أكاديمية كاوست
KAUST ACADEMY

The output $O(x, y)$ of a transposed convolution is computed as:

$$O(x, y) = \sum_{i,j} I(i, j) \cdot K(x - i \cdot s, y - j \cdot s)$$

where:

- $O(x, y)$ is the output at position (x, y),
- $I(i, j)$ is the input value at (i, j),
- $K(x', y')$ is the kernel value at (x', y'),
- s is the stride.

$\curvearrowright W_{out}$

$$H_{out} = (H_{in} - 1)S - 2P + K + \text{outputpadding}$$

$$= (2-1)2 - 2(1) + 3 + 1$$

$$= 2 - 2 + 3 + 1$$

$$= 4$$

$\rightarrow 128$

2×2 → 4×4

output resolution = 256, $S_{inp} = ?$

input resolution = 64×64 → inp = 1, K = 3×3, S = 2

output padding ← Outp = 2

Transpose Convolution Output Size Formula

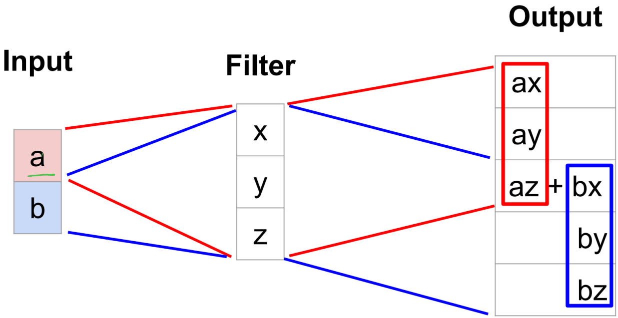
$$H_{\text{out}} = (H_{\text{in}} - 1) \times \texttt{stride}[0] - 2 \times \texttt{padding}[0] + \texttt{dilation}[0] \times (\texttt{kernel\_size}[0] - 1) + \texttt{output\_padding}[0] + 1 \quad (1)$$

$$W_{\text{out}} = (W_{\text{in}} - 1) \times \texttt{stride}[1] - 2 \times \texttt{padding}[1] + \texttt{dilation}[1] \times (\texttt{kernel\_size}[1] - 1) + \texttt{output\_padding}[1] + 1 \quad (2)$$

where:

- $H_{\text{out}}, W_{\text{out}}$ - Output height and width.
- $H_{\text{in}}, W_{\text{in}}$ - Input height and width.
- Stride - Step size of the filter movement.
- Padding - Number of pixels added around the input.
- Dilation - Spacing between kernel elements.
- Kernel size - Size of the convolution filter.
- Output padding - Additional padding applied to the output.

Transposed Convolution: 1D Example



Output contains copies of the filter weighted by the input, summing at where at overlaps in the output

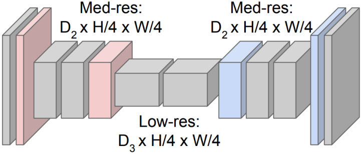
Semantic Segmentation Idea: Fully Convolutional

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

Input: 3 x H x W



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

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

Can We Do Better?

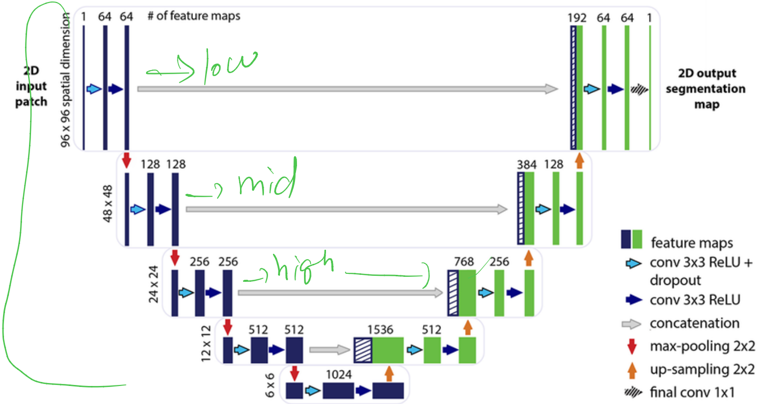
- The downsampling-then-upsampling approach works well for semantic segmentation.
- **But... can we do better?**
- **Problem**: Important details and spatial information may be lost during downsampling.

Can We Do Better?

- The downsampling-then-upsampling approach works well for semantic segmentation.
- **But... can we do better?**
- **Problem**: Important details and spatial information may be lost during downsampling.
- **Solution**: Introduce **residual connections** to preserve spatial information.

أكاديمية كاوست
KAUST ACADEMY

- Directly connect features from downsampling layers to upsampling layers.
- Help recover lost spatial details and improve segmentation accuracy.

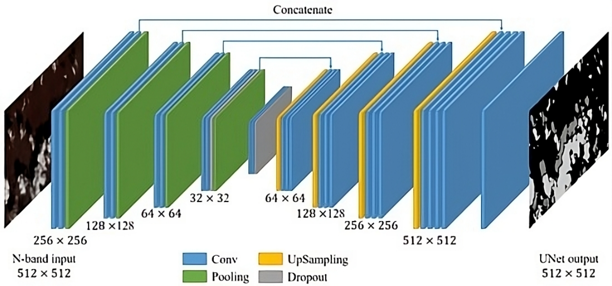


Residual Connections in Segmentation

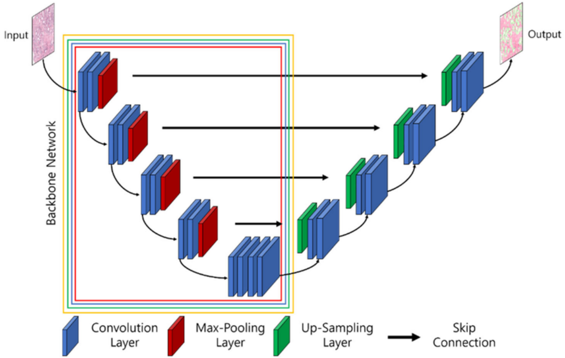
- There are two Types of Residuals:
  - **Addition:** Adds features from the encoder to the decoder **element-wise**.
  - **Concatenation:** Concatenates features from the encoder to the decoder along the **channel dimension**.
- **Which Is Better?**
  - Concatenation is often better because it retains more feature information from the encoder.
  - **Note**: technically, concatenation might be harder to implement because it requires aligning input and output shapes.

U-Net

- This architecture, with residual connections, is called **U-Net**.
- **Why the name?**
  - The architecture resembles the shape of the letter "U".
  - Features are downsampled in the encoder and upsampled in the decoder, with skip connections in between.
- U-Net is widely used for segmentation tasks, especially in biomedical imaging.

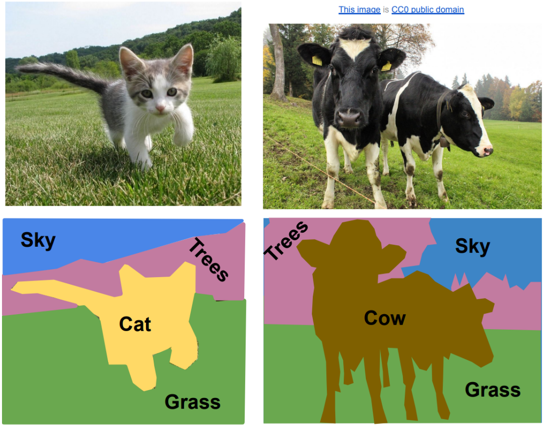


- **Pretrained Encoder:**
    - The encoder can use a pretrained backbone (e.g., ResNet, EfficientNet).
    - This help utilize features learned on large datasets (e.g., ImageNet).
    - Only the decoder is trained from scratch for segmentation-specific tasks.

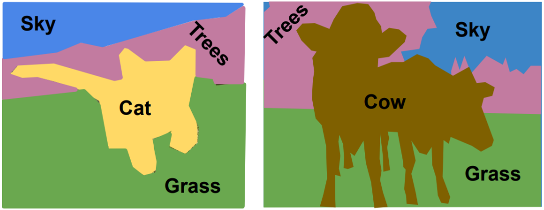


Semantic Segmentation

- Label each pixel in the image with a category label

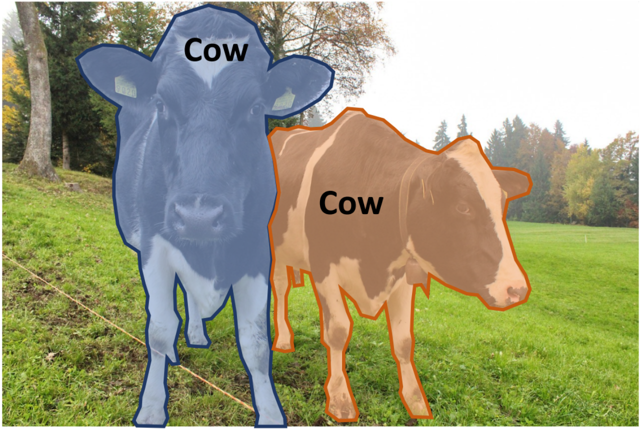


- Label each pixel in the image with a category label



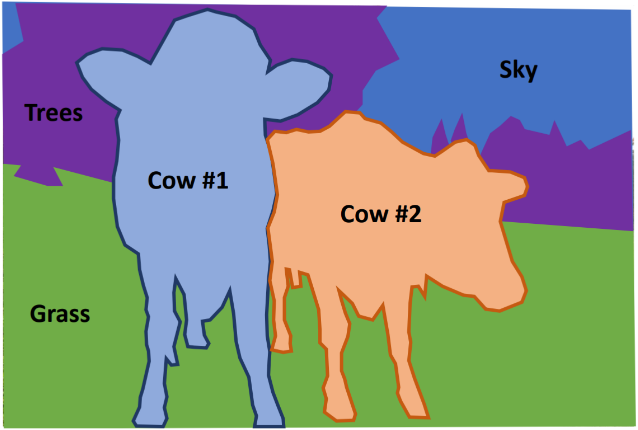
- **Does not differentiate instances, only care about pixels**

- Separate object instances, but only things



Panoptic Segmentation

- Label all pixels in the image (both things and stuff)



References

These slides have been adapted from

- Fei-Fei Li, Yunzhu Li & Ruohan Gao, Stanford CS231n: Deep Learning for Computer Vision
- Assaf Shocher, Shai Bagon, Meirav Galun & Tali Dekel, WAIC DL4CV Deep Learning for Computer Vision: Fundamentals and Applications
- Justin Johnson, UMich EECS 498.008/598.008: Deep Learning for Computer Vision