Image Segmentation

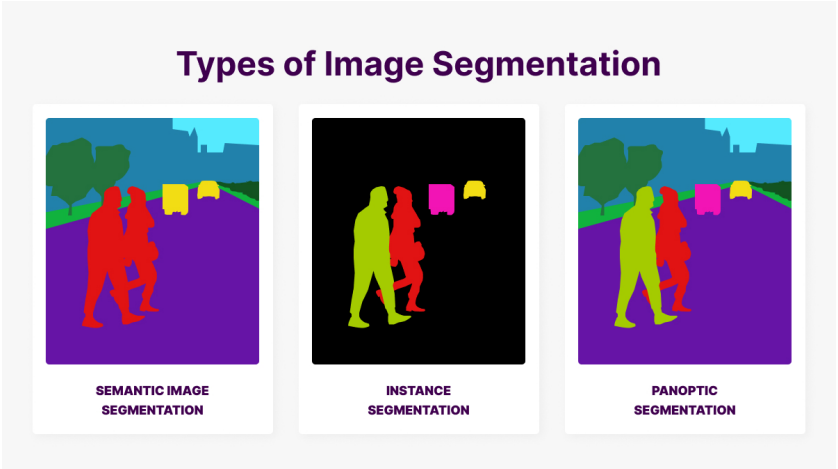
Naeemullah Khan
naeemullah.khan@kaust.edu.sa



جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

KAUST Academy
King Abdullah University of Science and Technology

August 9, 2026



Source: Mindy Support, "What is Image Segmentation: The Basics and Key Techniques", mindy-support.com.

Table of Contents

1. Motivation
2. Learning Outcomes
3. Introduction
4. Things and Stuff
5. Semantic Segmentation
    1. Sliding Window
    2. Convolution
6. Upsampling
7. Transposed Convolution
8. Fully Convolutional Networks (FCNs)
9. Skip Connections and U-Net
10. Instance Segmentation
    1. Mask R-CNN

Table of Contents (cont.)

11. Panoptic Segmentation

12. Human Keypoints Estimation

13. Pose Estimation

14. Captioning

15. 3D Shape Prediction

16. Object Tracking

17. Limitations and Future Directions

18. References

Image Segmentation: **Motivation**

Why Segment Images?

- **Classification**: Tells *what* is in the image.
- **Detection**: Tells *where* objects are.
- **Segmentation**: Tells *exactly which pixels* belong to each object.

Crucial for:

- Medical imaging
- Autonomous driving
- Robotics
- AR/VR
- Satellite imagery

Real-World Impact

- Tumor localization in MRIs
- Road boundary detection for self-driving cars
- Crop field segmentation from drone images

Image Segmentation: **Learning Outcomes**

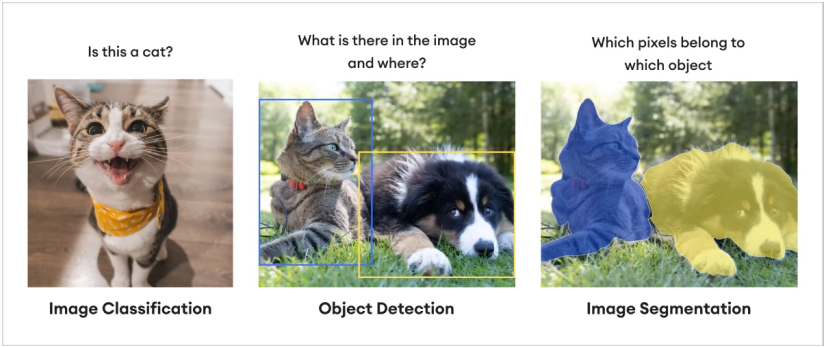
Learning Outcomes

- Understand the need and core idea behind image segmentation
- Explain transposed convolution, Fully Convolutional Networks (FCNs), and U-Net
- Differentiate between semantic, instance, and panoptic segmentation
- Survey tasks built on top of instance segmentation: human keypoints, pose estimation, dense captioning, 3D shape prediction, and object tracking
- Identify key challenges and limitations in image segmentation
- Explore future directions and advanced research topics

Image Segmentation: **Introduction**

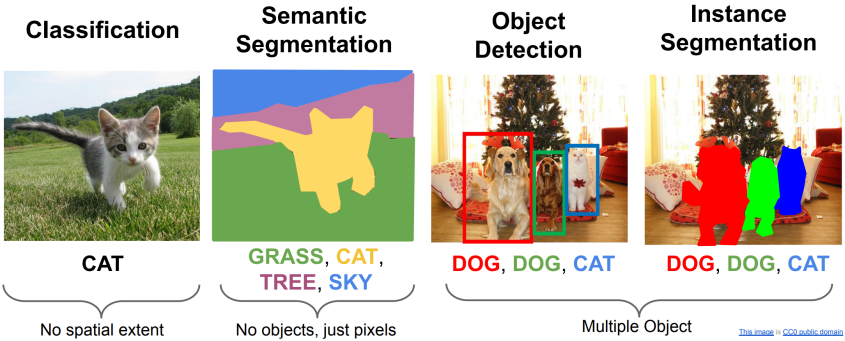
What is Image Segmentation?

- **Definition:** Partitioning an image into multiple meaningful segments.
- **Goal:** Assign a label to every pixel such that pixels with the same label share visual characteristics.



Source: SuperAnnotate, "Image segmentation detailed overview", superannotate.com.

Computer Vision Tasks



Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 3

- **Classification:** Identify the main object in an image.
- **Detection:** Locate objects with bounding boxes.
- **Segmentation:** Classify each pixel to delineate object boundaries.

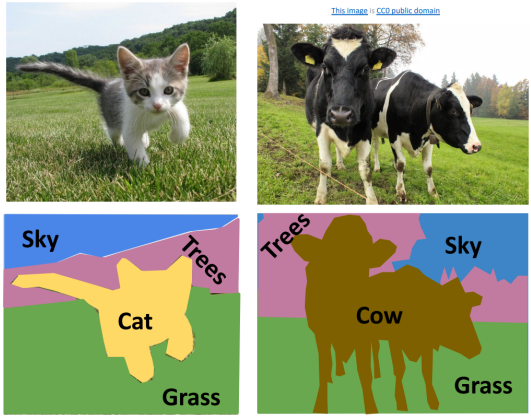
Challenges in Segmentation

- **Pixel-level labeling is labor-intensive**
- Varying object size and shape
- Occlusion and overlapping objects
- Class imbalance (background vs. object)

Image Segmentation: **Things and Stuff**

Things and Stuff

- **Things:** Object categories that can be separated into object instances (e.g. cats, cars, person)
- **Stuff:** Object categories that cannot be separated into instances (e.g. sky, grass, water, trees)



Source: UMich EECS 498 (2019), Lecture 16: Detection and Segmentation, slide 68

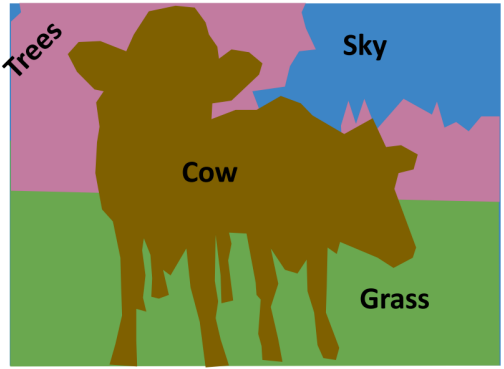
Computer Vision Tasks

- **Object Detection:** Detects individual object instances, but only gives boxes (Only things!)



Source: UMich EECS 498 (2019), Lecture 16: Detection and Segmentation, slide 69

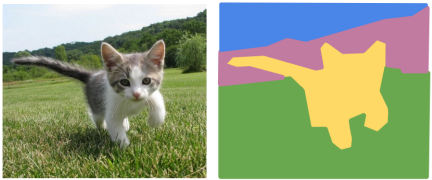
- **Semantic Segmentation:** Gives per-pixel labels, but merges instances (Both things and stuff)



Source: UMich EECS 498 (2019), Lecture 16: Detection and Segmentation, slide 69

Image Segmentation: **Semantic Segmentation**

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.

Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 5

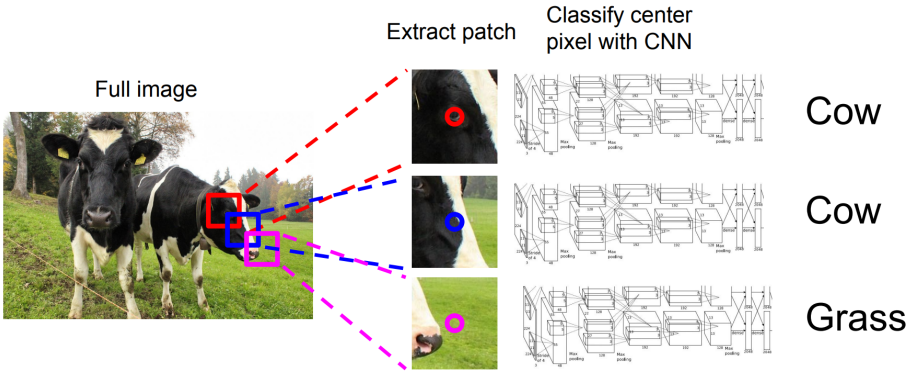
Semantic Segmentation (cont.)



Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 6

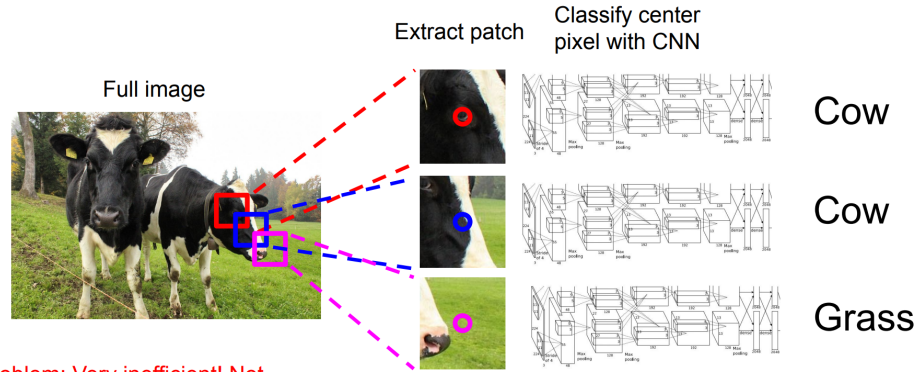
- Impossible to classify without context
- How do we include context?

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

Semantic Segmentation Idea: Sliding Window (cont.)



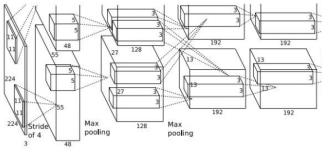
Problem: Very inefficient! Not reusing shared features between overlapping patches

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







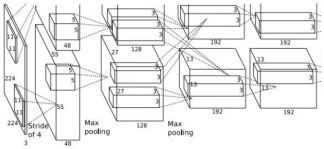
An intuitive idea: encode the entire image with conv net, and do semantic segmentation on top.

Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 11

Semantic Segmentation Idea: Convolution (cont.)

Full image







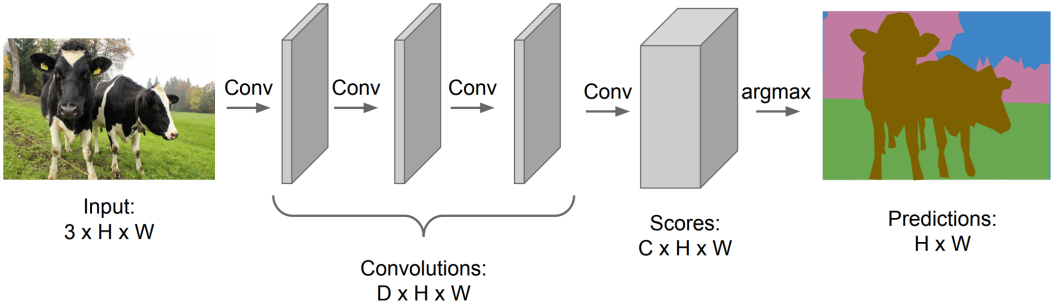
An intuitive idea: encode the entire image with conv net, and do semantic segmentation on top.

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.

Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 11

Semantic Segmentation Idea: Fully Convolutional

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



Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 12

Semantic Segmentation Idea: Fully Convolutional (cont.)

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



Input:
3 x H x W

Conv

Conv

Conv

Conv

argmax



Predictions:
H x W

Scores:
C x H x W

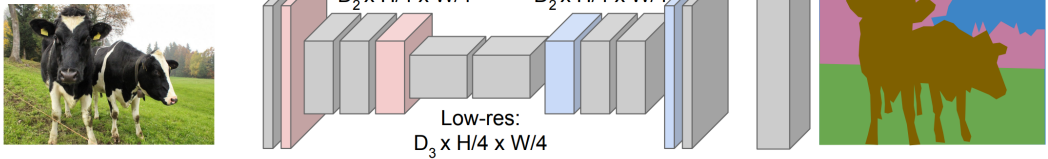
Convolutions:
D x H x W

Problem: convolutions at original image resolution will be very expensive ...

Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 13

Semantic Segmentation Idea: Fully Convolutional (cont.)

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.)

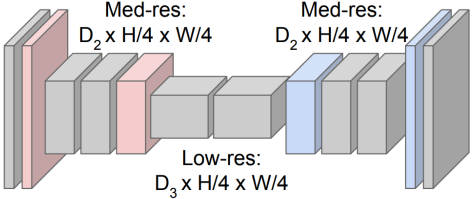
Downsampling: Pooling, strided convolution

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

C x H x W

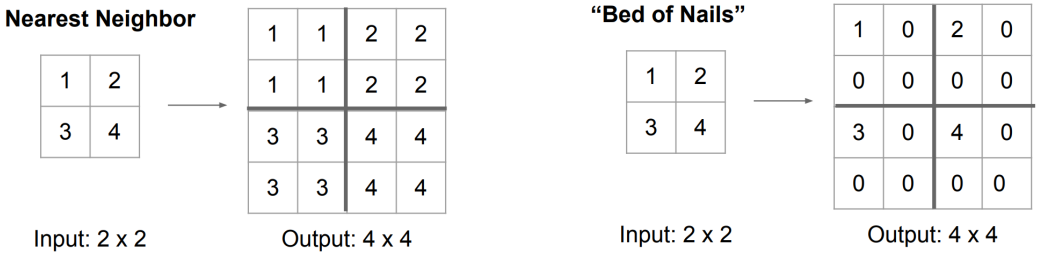


Predictions: H x W

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

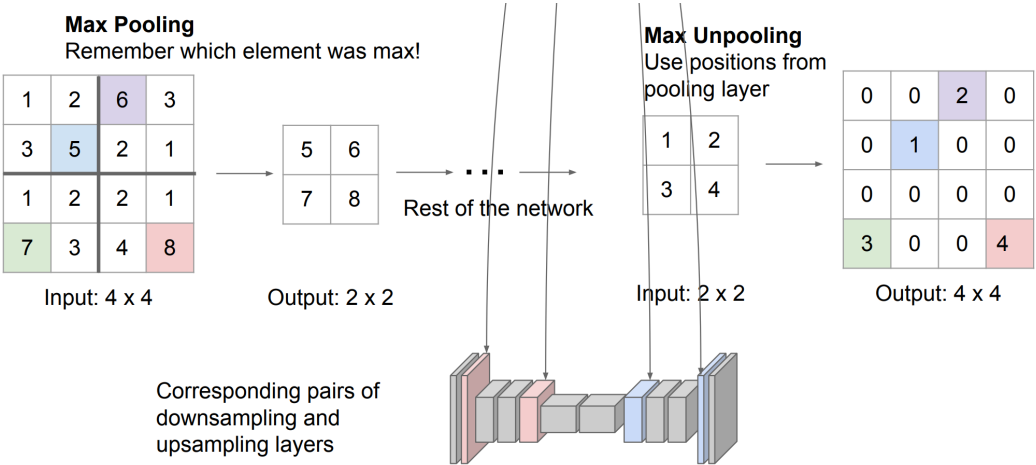
Image Segmentation: **In-Network Upsampling**

In-Network Upsampling: Unpooling



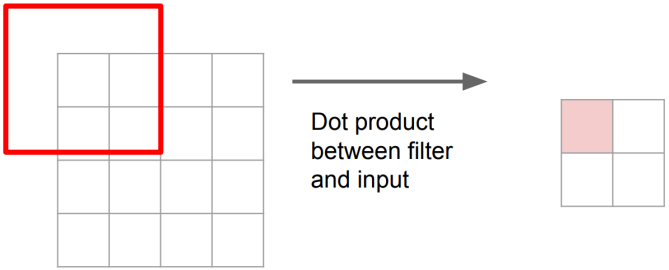
Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 16

In-Network Upsampling: Max Unpooling



Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 17

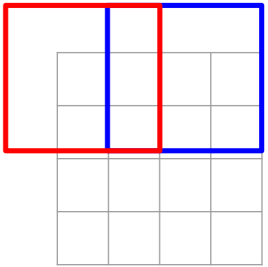
Recall: Normal 3 x 3 convolution, stride 2 pad 1



Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 22

Learnable Upsampling: Transposed Convolution (cont.)

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

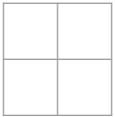
Stride gives ratio between movement in input and output

We can interpret strided convolution as “learnable downsampling”.

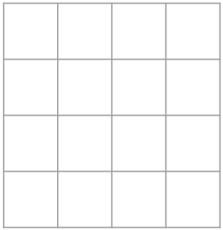
Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 23

Learnable Upsampling: Transposed Convolution (cont.)

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



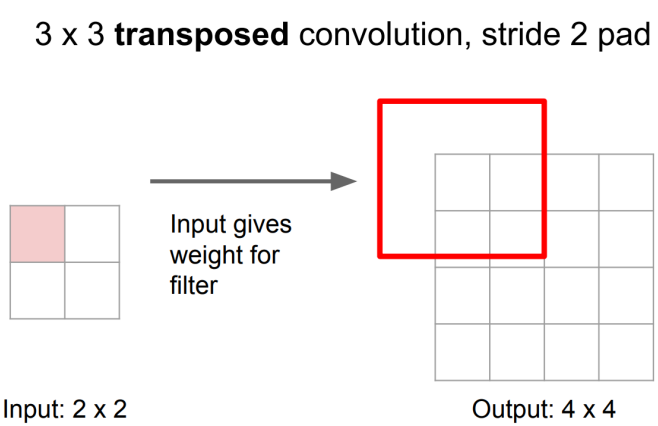
Input: 2 x 2



Output: 4 x 4

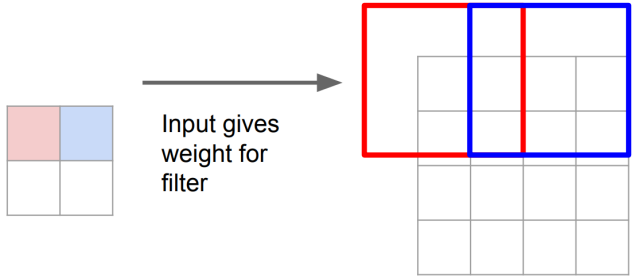
Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 24

Learnable Upsampling: Transposed Convolution (cont.)



Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 25

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

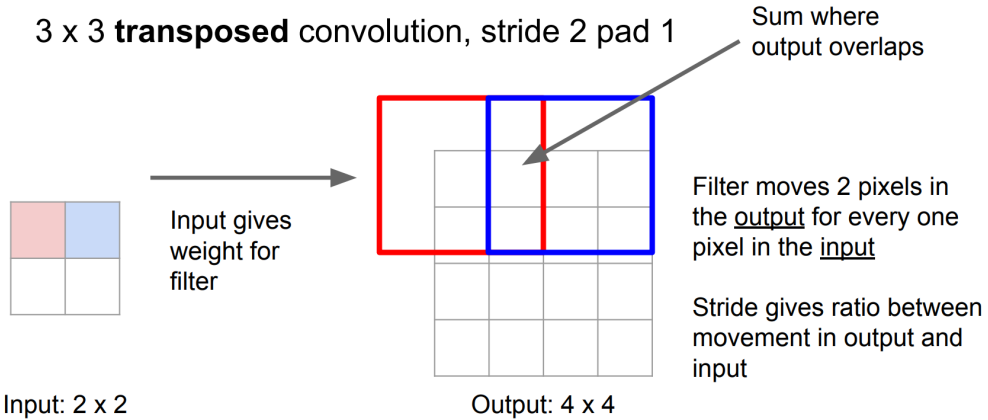


Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 26

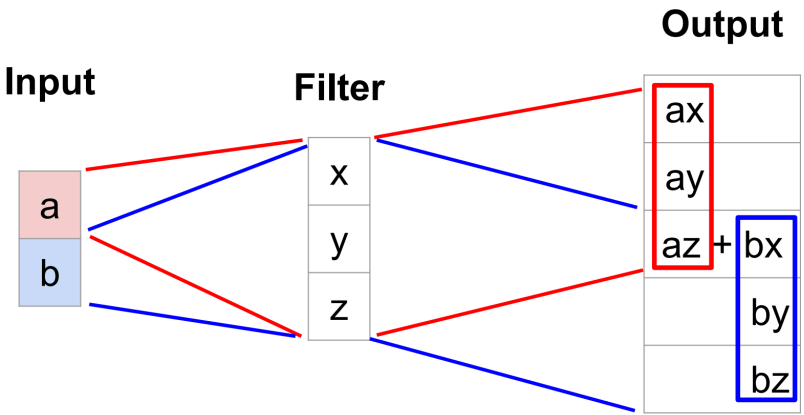
Learnable Upsampling: Transposed Convolution (cont.)



Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 27

Image Segmentation: **Transposed Convolution**

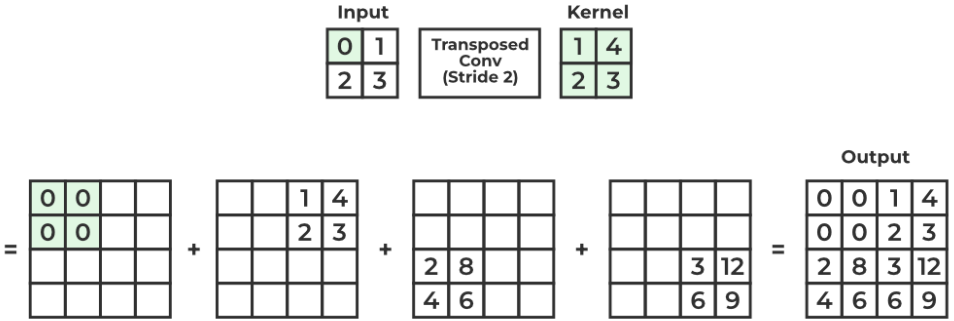
Transposed Convolution: 1D Example



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

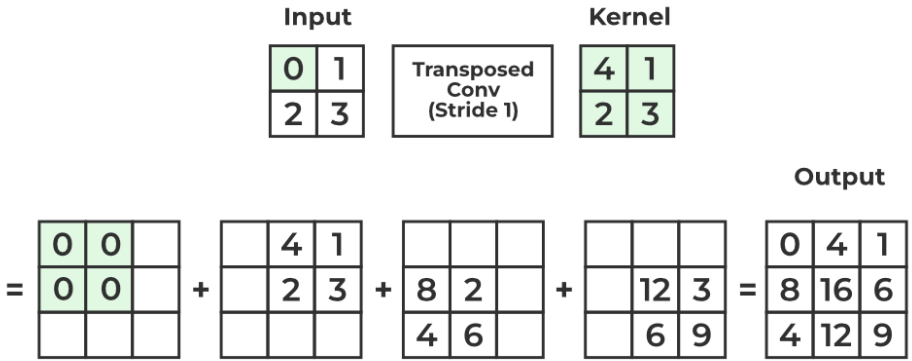
Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 29

Transposed Convolution: How It Works



Stride 2. Source: "What is Transposed Convolutional Layer?", geeksforgeeks.org.

Transposed Convolution: How It Works (cont.)



Stride 1. Source: "What is Transposed Convolutional Layer?", geeksforgeeks.org.

Mathematical Definition: Transposed Convolution

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.

Transposed Convolution Output Size Formula

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

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

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.

Transposed Convolution Output Size Formula (cont.)

- Dilation - Spacing between kernel elements.
- Kernel size - Size of the convolution filter.
- Output padding - Additional padding applied to the output.

Image Segmentation: **Fully Convolutional Networks (FCNs)**

Semantic Segmentation Idea: Fully Convolutional

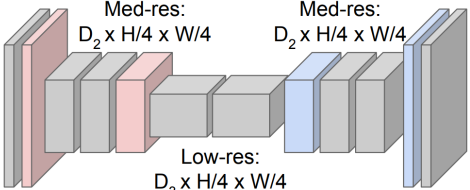
Downsampling:
Pooling, strided convolution

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



High-res:
D1 x H/2 x W/2

High-res:
D1 x H/2 x W/2

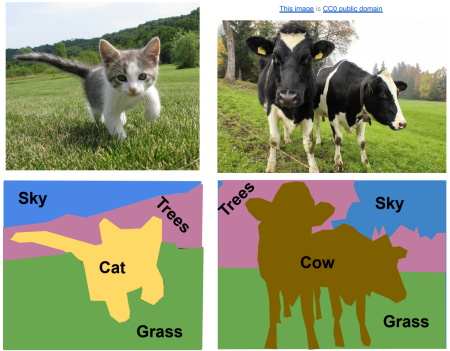


Predictions:
H x W

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

Semantic Segmentation

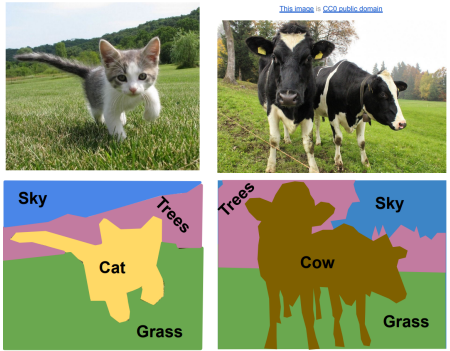
- Label each pixel in the image with a category label



Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 34

Semantic Segmentation

- Label each pixel in the image with a category label



Source: Stanford CS231n (2022), Lecture 9: Object Detection and Image Segmentation, slide 34

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

Image Segmentation: **Skip Connections and U-Net**

Can We Do Better?

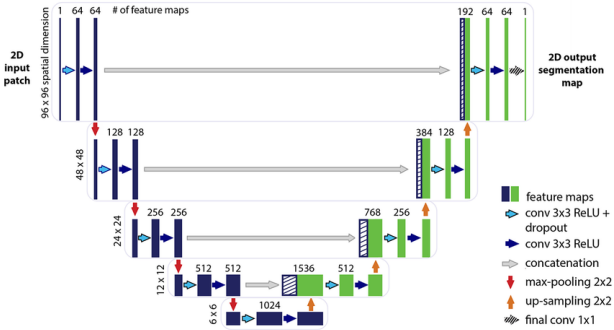
- The downsampling-then-upsampling approach works well for semantic segmentation.
- **But... can we do better?**
- **Problem**: Important details and spatial information may be lost during downsampling.

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 **skip connections** between encoder and decoder to preserve spatial information.

Skip Connections in Segmentation

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



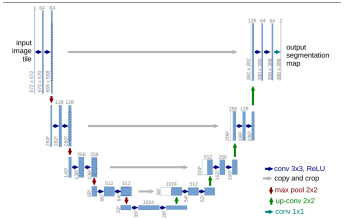
Encoder-decoder with concatenation skip connections. Source: Livne et al., *Front. Neurosci.* 13:97, 2019, Fig. 2 (CC BY).

Skip Connections in Segmentation

- There are two types of skip connection (additive skips are what "residual" specifically means):
  - **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**: addition requires the encoder and decoder features to match exactly in both spatial size and channel count; concatenation only needs spatial alignment, but increases the decoder's channel count.

U-Net

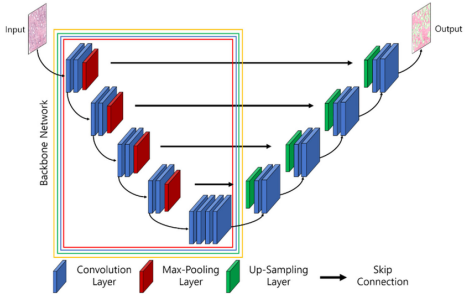
- This architecture, with concatenation skip connections, is called **U-Net**.
- **Why the name?**
  - The architecture resembles the shape of the letter "U".
  - Features are downsampled in the encoder and upsampled in the decoder, with skip connections in between.
- U-Net is widely used for segmentation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



Source: Ronneberger, Fischer & Brox, MICCAI 2015, Fig. 1.

Using a Pretrained U-Net

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

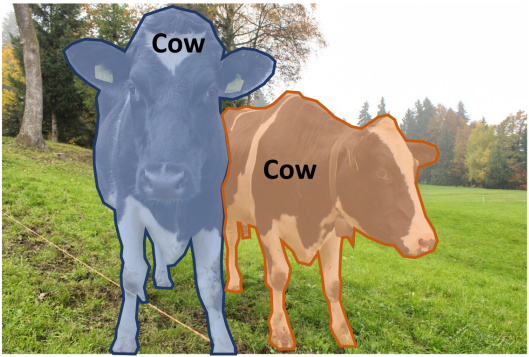


Source: Ikromjanov et al., *Cancers* 15(3):762, 2023, Fig. 5 (CC BY).

Image Segmentation: **Instance Segmentation**

Instance Segmentation

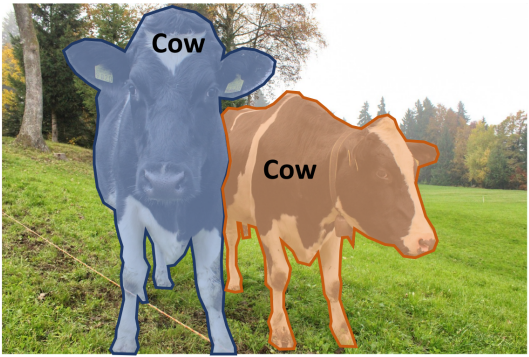
- Detect all objects in the image, and identify the pixels that belong to each object (Only things!)



Source: UMich EECS 498 (2019), Lecture 16: Detection and Segmentation, slide 72

Instance Segmentation

- Detect all objects in the image, and identify the pixels that belong to each object (Only things!)
- **Approach:** Perform object detection, then predict a segmentation mask for each object!



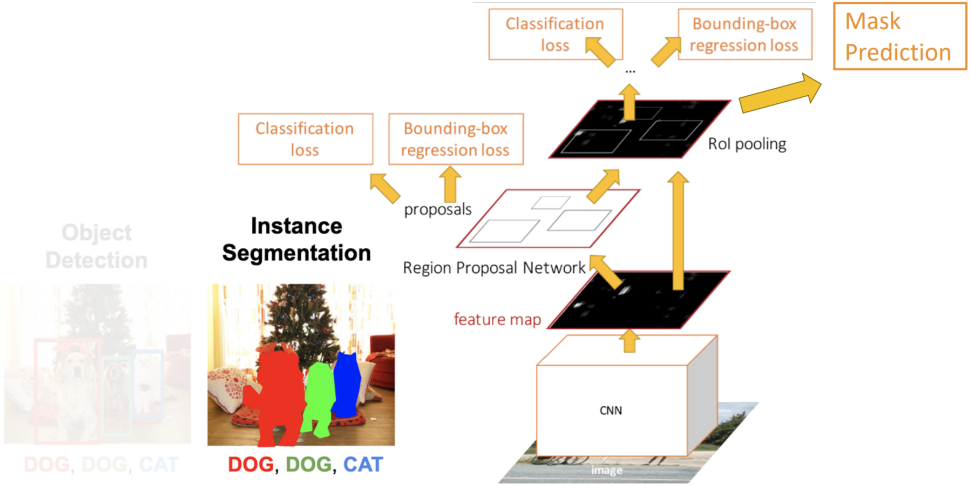
Source: UMich EECS 498 (2019), Lecture 16: Detection and Segmentation, slide 72

Instance Segmentation: **Mask R-CNN**

Mask R-CNN Overview

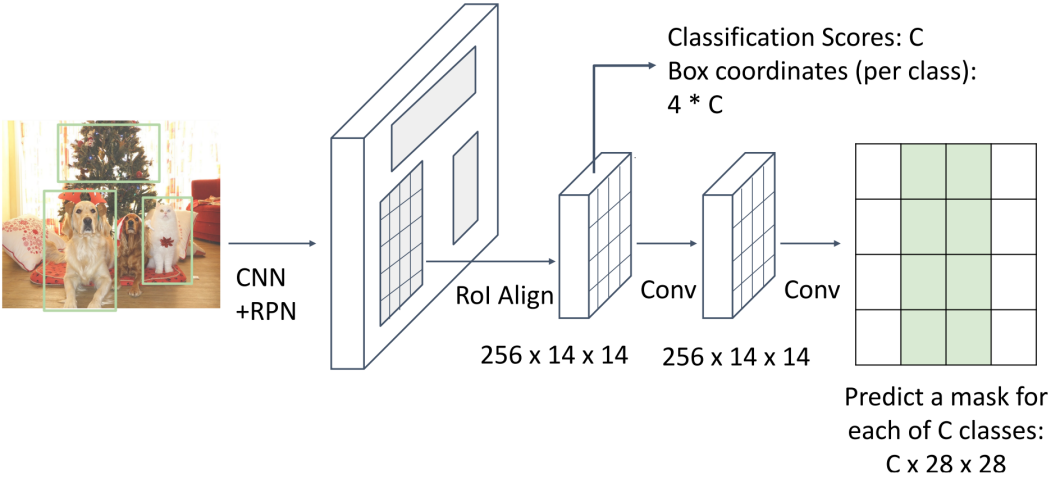
- **Developed on top of Faster R-CNN**
- Faster R-CNN outputs:
  - Class label
  - Bounding-box offset
- Mask R-CNN adds a third branch:
  - Predicts object mask for each candidate
- Performs **Instance Segmentation** (per-object masks; extensible to keypoint/pose estimation)

Mask R-CNN (cont.)



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

Mask R-CNN (cont.)

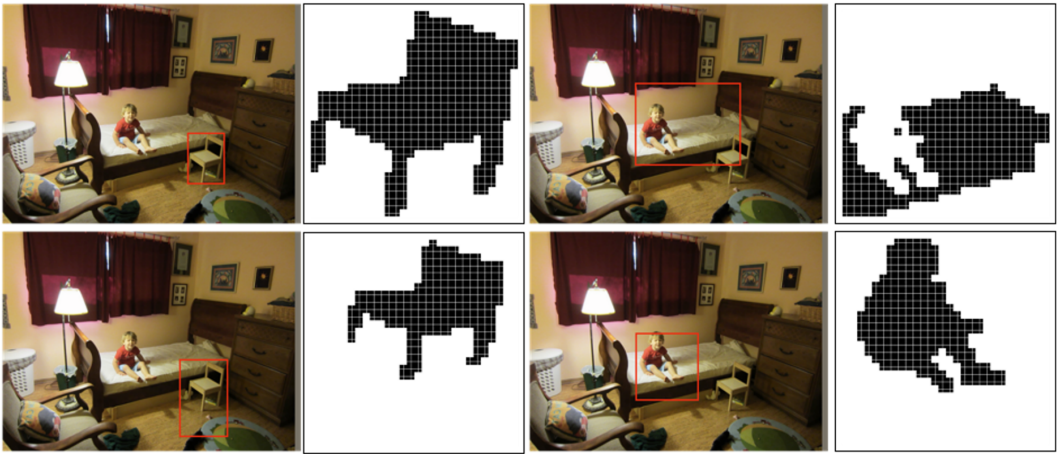


Source: UMich EECS 498 (2022), Lecture 15: Image Segmentation, slide 79 (figure from He et al., "Mask R-CNN", ICCV 2017)

Advantages of Mask R-CNN

- **Simplicity**: Simple to train
- **Performance**: Outperforms previous methods, at 5 fps (195 ms per image on a Tesla M40)
- **Efficiency**: Very efficient, small overhead compared to Faster R-CNN
- **Flexibility**: Can perform detection and estimation tasks simultaneously

Mask R-CNN: Example Training Targets



Source: UMich EECS 498 (2022), Lecture 15: Image Segmentation, slide 83 (figure from He et al., "Mask R-CNN", ICCV 2017)

Mask R-CNN: Very Good Results!

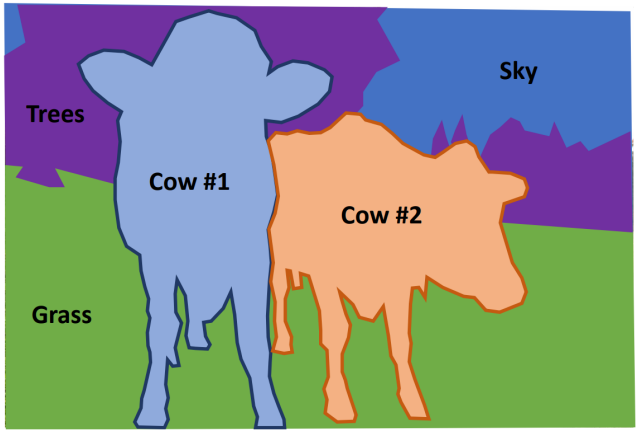


Source: UMich EECS 498 (2022), Lecture 15: Image Segmentation, slide 84 (figure from He et al., "Mask R-CNN", ICCV 2017)

Image Segmentation: **Panoptic Segmentation**

Beyond Instance Segmentation: Panoptic Segmentation

- Label all pixels in the image (both things and stuff)
- For "thing" categories also separate into instances



Source: UMich EECS 498 (2019), Lecture 16: Detection and Segmentation, slide 82

Beyond Instance Segmentation: Panoptic Segmentation

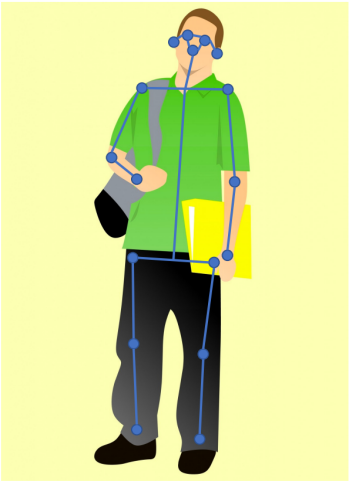


Source: Kirillov, Girshick, He & Dollár, "Panoptic Feature Pyramid Networks", CVPR 2019, Fig. 2.

Image Segmentation: **Human Keypoints Estimation**

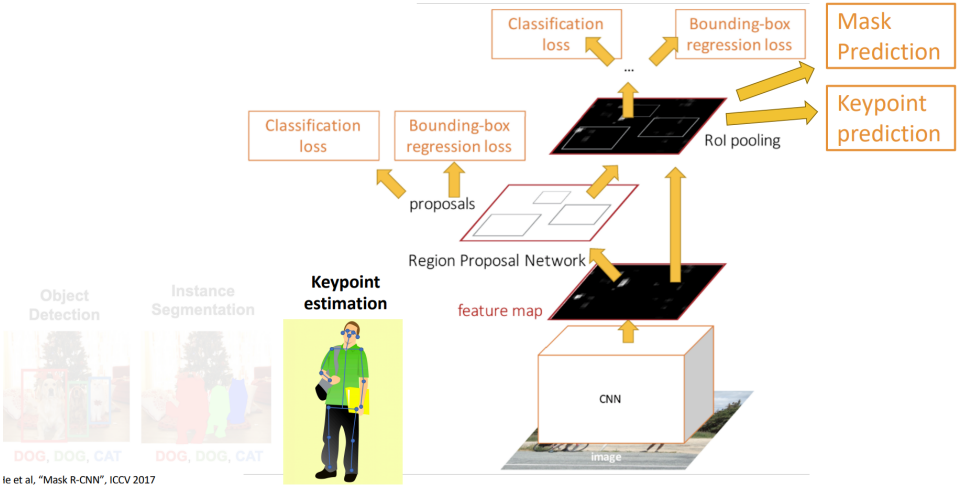
Beyond Instance Segmentation: Human Keypoints

- Represent the pose of a human by locating a set of keypoints, e.g., 17 keypoints:
- Nose
- Left / Right eye
- Left / Right ear
- Left / Right shoulder
- Left / Right elbow
- Left / Right wrist
- Left / Right hip
- Left / Right knee
- Left / Right ankle



Source: UMich EECS 498 (2019), Lecture 16: Detection and Segmentation, slide 84

Mask R-CNN: Keypoint Estimation

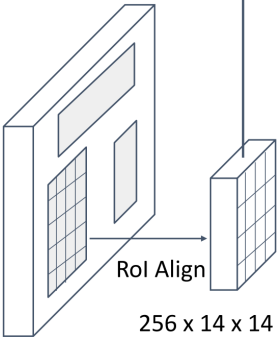


Ie et al, "Mask R-CNN", ICCV 2017

Mask R-CNN: Keypoint Estimation



CNN
+RPN



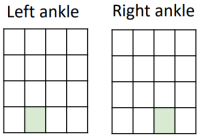
RoI Align

256 x 14 x 14

Conv...

Classification Scores: C
Box coordinates (per class): 4 * C
Segmentation mask: C x 28 x 28

One mask for each of
the K different keypoints



...

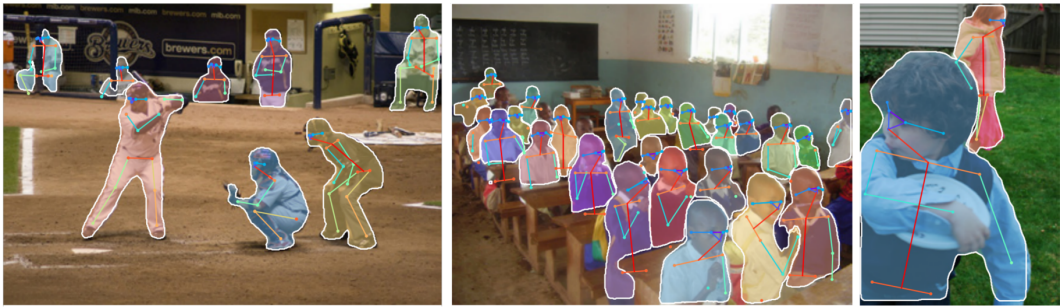
Keypoint masks:
K x 56 x 56

Ground-truth has one “pixel” turned c
per keypoint. Train with softmax loss

He et al. “Mask R-CNN”. ICCV 2017

Image Segmentation: **Pose Estimation**

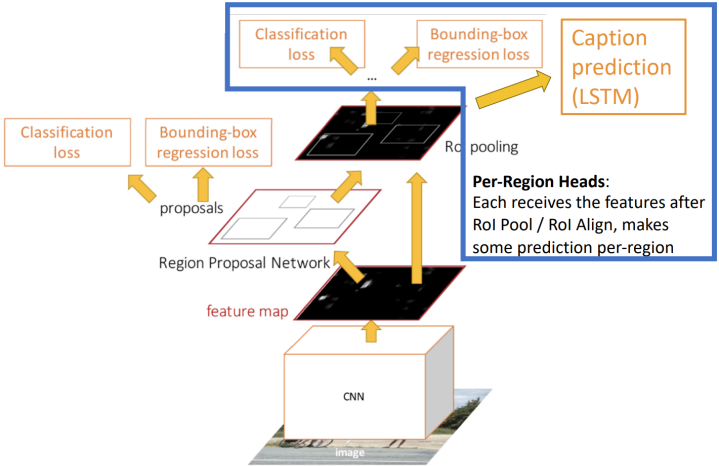
Joint Instance Segmentation and Pose Estimation



Source: He, Gkioxari, Dollár & Girshick, "Mask R-CNN", ICCV 2017.

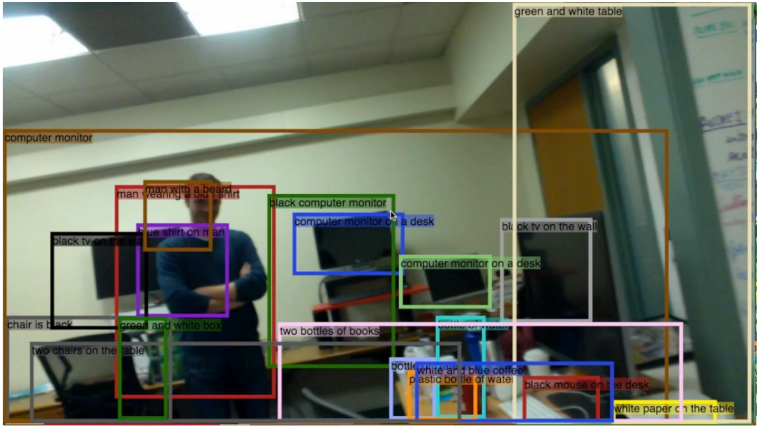
Image Segmentation: **Captioning**

Captioning: Predict a caption per region!



Source: UMich EECS 498 (2019), Lecture 16: Detection and Segmentation, slide 91

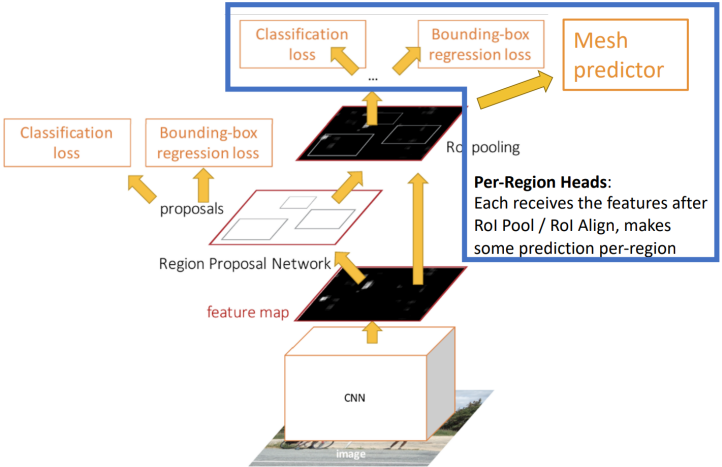
Captioning: Predict a caption per region!



Source: Johnson, Karpathy & Fei-Fei, "DenseCap: Fully Convolutional Localization Networks for Dense Captioning", CVPR 2016.

Image Segmentation: **3D Shape Prediction**

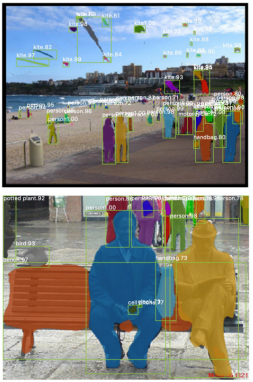
3D Shape Prediction



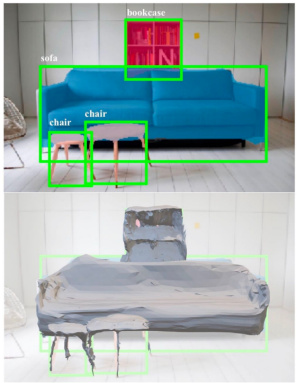
Source: UMich EECS 498 (2019), Lecture 16: Detection and Segmentation, slide 94

3D Shape Prediction

Mask R-CNN:
2D Image -> 2D shapes



Mesh R-CNN:
2D Image -> **3D** shapes



Source: Gkioxari, Malik & Johnson, "Mesh R-CNN", ICCV 2019.

Image Segmentation: **Object Tracking**

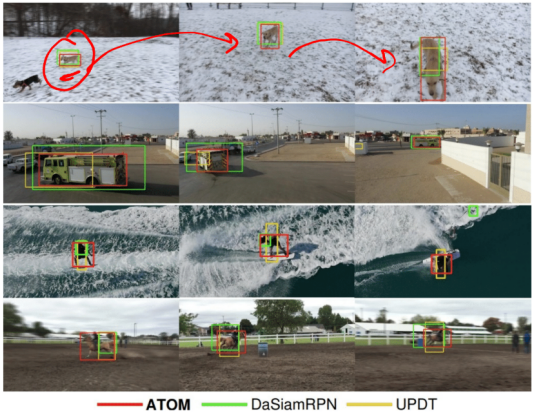
Goal of Object Tracking

- Track objects over a sequence of photos or a video
- Exceedingly challenging in multi-object tracking scenarios
- Need to ensure objects are not mixed up or lost midway

A Common Solution

- Perform object detection in each frame
- Assign unique IDs to each detected object
- Store feature vectors for each object (e.g., appearance, position)
- Track objects across frames based on their IDs and feature vectors

A Common Solution (cont.)



Comparison of three tracking approaches. Source: Danelljan, Bhat, Khan & Felsberg, "ATOM: Accurate Tracking by Overlap Maximization", CVPR 2019, Fig. 1.

Image Segmentation: **Limitations and Future Directions**

Common Limitations

- Requires a large amount of labeled data (pixel-wise annotations are expensive)
- Computationally intensive
- Struggles with:
  - Occlusion
  - Small objects
  - Rare classes
- Poor generalization across domains (e.g., synthetic $\rightarrow$ real)

Where is the Field Going?

- **Self-supervised learning:** Reduce need for labeled data
- **Few-shot segmentation:** Learn from few examples
- **Transformer-based segmentation:** (e.g., Segmenter, Mask2Former)
- **Real-time segmentation:** For AR/VR, autonomous systems
- **Multimodal segmentation:** Combine RGB + depth + text

Image Segmentation: **References**

References & Further Reading

[1] Long, J., Shelhamer, E., & Darrell, T. (2015).

Fully Convolutional Networks for Semantic Segmentation.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

arxiv.org/abs/1411.4038

[2] Ronneberger, O., Fischer, P., & Brox, T. (2015).

U-Net: Convolutional Networks for Biomedical Image Segmentation.

In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.

arxiv.org/abs/1505.04597

References & Further Reading (cont.)

[3] Johnson, J., Karpathy, A., & Fei-Fei, L. (2016).

DenseCap: Fully Convolutional Localization Networks for Dense Captioning.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

arxiv.org/abs/1511.07571

[4] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017).

Mask R-CNN.

In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

arxiv.org/abs/1703.06870

References & Further Reading (cont.)

[5] Gkioxari, G., Malik, J., & Johnson, J. (2019).

Mesh R-CNN.

In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

arxiv.org/abs/1906.02739

[6] Kirillov, A., He, K., Girshick, R., Rother, C., & Dollár, P. (2019).

Panoptic Segmentation.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

arxiv.org/abs/1801.00868

References & Further Reading (cont.)

[7] Kirillov, A., Girshick, R., He, K., & Dollár, P. (2019).

Panoptic Feature Pyramid Networks.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

arxiv.org/abs/1901.02446

[8] Danelljan, M., Bhat, G., Khan, F. S., & Felsberg, M. (2019).

ATOM: Accurate Tracking by Overlap Maximization.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

arxiv.org/abs/1811.07628

References & Further Reading (cont.)

[9] Strudel, R., Garcia, R., Laptev, I., & Schmid, C. (2021).

Segmenter: Transformer for Semantic Segmentation.

In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

arxiv.org/abs/2105.05633

[10] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021).

SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers.

In *Advances in Neural Information Processing Systems (NeurIPS)*.

arxiv.org/abs/2105.15203

Additional Resources:

- Hugging Face Tasks: huggingface.co/tasks/image-segmentation

References & Further Reading (cont.)

- Stanford CS231n Slides (Segmentation Modules): cs231n.stanford.edu/slides/2022/
- Michigan EECS 498/598, Deep Learning for Computer Vision (Justin Johnson): web.eecs.umich.edu/~justincj/teaching/eecs498/
- DeepMind Perceiver IO: arxiv.org/abs/2107.14795
- Meta AI (FAIR) Mask2Former: arxiv.org/abs/2112.01527