Computer Vision - 2025

Lecture #05. Segmentation and Object Detection

Lectures by Alexei Kornaev [1,2,3]

Practical sessions by Kirill Yakovlev [2]

[1]AI Institute, Innopolis University (IU), Innopolis

[2]Robotics & CV Master's Program, IU, Innopolis

[3]RC for AI, National RC for Oncology, Moscow

February 17, 2025

Agenda

1. Outcomes
2. Introduction
3. U-Net: Semantic Segmentation
4. Segment Anything Model (SAM)
5. Conclusion

Section 1. Outcomes

Outcomes

This week's lecture and seminar on Segmentation and Object Detection aims to provide an understanding of deep learning-based segmentation methods and object detection techniques. By the end of this week, students will be able to:

1. Understand semantic segmentation and object detection concepts.
2. Describe models such as U-Net, DeepLab, Mask R-CNN, YOLO, Faster R-CNN, and SAM.
3. Implement segmentation models using PyTorch or TensorFlow.

Key Takeaway: Deep learning-based segmentation and object detection methods enable precise image analysis for various applications, from medical imaging to autonomous driving.

Section 2. Introduction

Types of Image Segmentation

Segmentation Overview

Image segmentation is the process of partitioning an image into meaningful regions. It plays a key role in medical imaging, autonomous driving, and scene understanding.

- **Semantic Segmentation:** Classifies each pixel into predefined object categories.
- **Instance Segmentation:** Identifies individual object instances within categories.
- **Panoptic Segmentation:** Unifies semantic and instance segmentation by labeling both things (objects) and stuff (background regions).

Figure: Types of segmentation problems.

Principles of Semantic Segmentation

Key Idea

Semantic segmentation assigns a class label to each pixel without distinguishing between object instances.

- Uses fully convolutional networks (FCNs) to generate dense predictions.
- Common architectures: U-Net [Ronneberger et al., 2015], DeepLab [Chen et al., 2018].
- Loss functions: Cross-entropy loss, Dice loss, IoU loss.

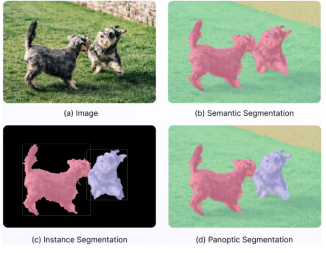


Figure: Types of segmentation problems.

Principles of Instance Segmentation

Key Idea

Instance segmentation detects and segments each object instance separately, unlike semantic segmentation.

- Combines object detection and segmentation.
- Example models: Mask R-CNN [He et al., 2018], SOLOv2 [Wang et al., 2020].
- Applications: Robotics, medical imaging, autonomous driving.

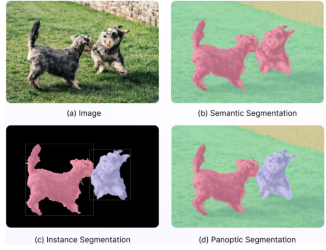


Figure: Types of segmentation problems.

Principles of Panoptic Segmentation

Key Idea

Panoptic segmentation unifies semantic and instance segmentation by labeling both objects and background regions.

- Combines "stuff" segmentation (background regions) with "thing" segmentation (object instances).
- Example models: Panoptic FPN [Kirillov et al., 2019], DETR [Carion et al., 2020].
- Applied in urban scene understanding, AR/VR, and satellite imagery.

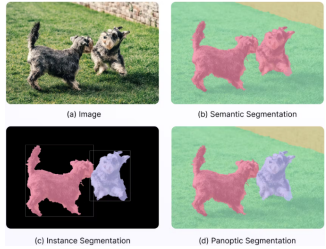


Figure: Types of segmentation problems.

Comparison of Segmentation Types

- **Semantic Segmentation:** Classifies each pixel but does not differentiate object instances.
- **Instance Segmentation:** Separates individual objects within a category.
- **Panoptic Segmentation:** Merges both approaches, distinguishing objects and background.

Aspect	Semantic	Instance	Panoptic
Pixel-wise classification	Yes	Yes	Yes
Object instance distinction	No	Yes	Yes
Stuff vs. things distinction	No	No	Yes
Example models	U-Net, DeepLab	Mask R-CNN, SOLO	Panoptic FPN, DETR

Table: Comparison of segmentation approaches.

Principles of Object Detection

Key Idea

Object detection identifies and localizes objects in an image by predicting bounding boxes and class labels.

- Combines classification and localization to detect multiple objects.
- Two-stage detectors: Region proposal + classification (e.g., Faster R-CNN [Ren et al., 2016]).
- One-stage detectors: Direct prediction (e.g., YOLO [Redmon et al., 2016]).
- Applications: Autonomous vehicles, surveillance, medical imaging.

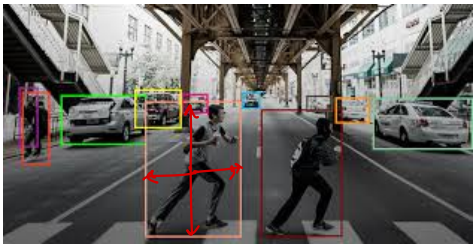


Figure: Example of object detection.

Section 3. U-Net: Semantic Segmentation

Key Ideas of U-Net

Key Idea

U-Net is a fully convolutional neural network designed for biomedical image segmentation. It follows a U-shaped architecture with an encoder-decoder structure.

- Encoder: Uses convolutional and pooling layers to extract feature maps.
- Bottleneck: Captures deep features with high-level abstractions.
- Decoder: Uses upsampling and skip connections to reconstruct segmentation masks.

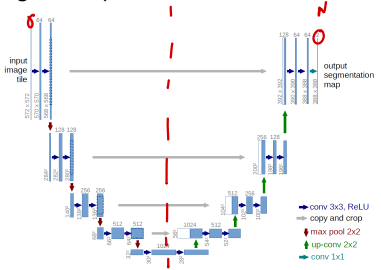


Figure: U-Net architecture [Ronneberger et al., 2015].

U-Net Architecture

- U-Net consists of a contracting path (encoder) and an expansive path (decoder).
- Skip connections concatenate corresponding encoder and decoder feature maps to retain spatial details.
- Final layer applies a 1x1 convolution to obtain pixel-wise classification.

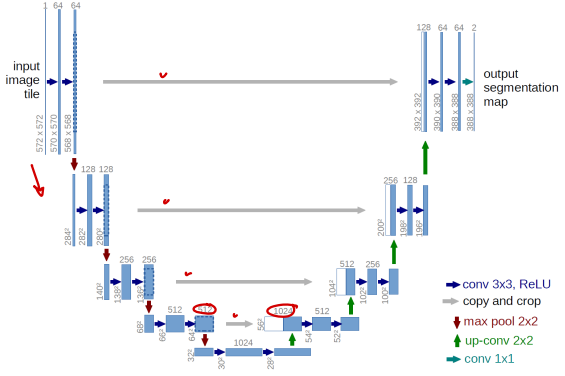


Figure: Detailed U-Net structure [Ronneberger et al., 2015].

Residual Connections in U-Net

Key Idea (He et al., 2015)

Residual connections help mitigate the vanishing gradient problem and improve gradient flow in deep networks.

- A residual connection allows the gradient to bypass some layers, making training easier.
- Instead of learning the full transformation $F(x)$, the network learns the residual function $F(x) + x$.
- Some U-Net variants, such as ResUNet, integrate residual blocks for better performance.

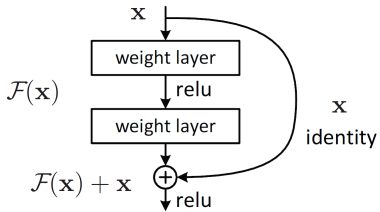


Figure: Residual connections in a CNN block.

Upsampling in U-Net

- U-Net uses upsampling to restore spatial resolution in the decoder.
- Two main types of upsampling:
  - **Interpolation-based Upsampling:** Bilinear or nearest neighbor interpolation followed by convolution.
  - **Transposed Convolution (Deconvolution):** Learnable filters upscale feature maps.

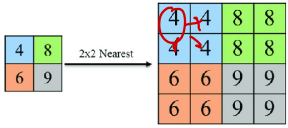


Figure: Nearest neighbor interpolation [Prince, 2023].

Transposed Convolution (Deconvolution)

- Transposed convolution is a learnable upsampling method.
- Instead of direct interpolation, it applies convolution on an expanded input.
- Steps:
  1. Insert zeros between pixels (if stride > 1).
  2. Convolve with a learnable kernel.
  3. Output has a larger spatial resolution.

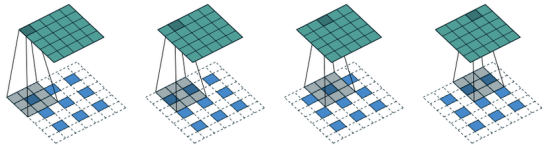
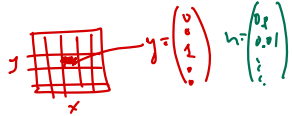


Figure: Example of transposed convolution operation.

Loss Functions for U-Net



- **Cross-Entropy Loss:** Pixel-wise loss for multi-class segmentation.

$$\mathcal{L}_{CE} = -\frac{1}{m \cdot H \cdot W} \sum_{i=1}^{m} \sum_{x=1}^{H} \sum_{y=1}^{W} \sum_{k=1}^{N} y_k^{(i)}(x,y) \log(h_k^{(i)}(x,y)) \tag{1}$$

- **Dice Loss:** Optimized for imbalanced datasets.

$$\mathcal{L}_{Dice} = 1 - \frac{2\sum_{i=1}^{m}\sum_{x=1}^{H}\sum_{y=1}^{W}\sum_{k=1}^{N} y_k^{(i)}(x,y) h_k^{(i)}(x,y)}{\sum_{i=1}^{m}\sum_{x=1}^{H}\sum_{y=1}^{W}\sum_{k=1}^{N} y_k^{(i)}(x,y) + \sum_{i=1}^{m}\sum_{x=1}^{H}\sum_{y=1}^{W}\sum_{k=1}^{N} h_k^{(i)}(x,y)} \tag{2}$$

- **IoU Loss:** Measures the intersection-over-union between prediction and ground truth.

$$\mathcal{L}_{IoU} = 1 - \frac{\sum_{i=1}^{m}\sum_{x=1}^{H}\sum_{y=1}^{W}\sum_{k=1}^{N} y_k^{(i)}(x,y) h_k^{(i)}(x,y)}{\sum_{i=1}^{m}\sum_{x=1}^{H}\sum_{y=1}^{W}\sum_{k=1}^{N} \left(y_k^{(i)}(x,y) + h_k^{(i)}(x,y) - y_k^{(i)}(x,y) h_k^{(i)}(x,y) \right)} \tag{3}$$

where m is the number of images of size $H \times W$, N is the number of classes, $y_k^{(i)}(x,y)$ is the label for pixel (x,y) in image i (one-hot encoded), $h_k^{(i)}(x,y)$ is the prediction.

Metrics for U-Net Evaluation

- **Dice Coefficient:** Measures segmentation accuracy.
- **Intersection over Union (IoU):** Evaluates overlap between prediction and ground truth.
- **Pixel Accuracy:** Computes correctly classified pixels.
- **Mean IoU:** Averages IoU across all classes.

$$\text{IoU} = \frac{|\mathbf{y} \cap \mathbf{h}|}{|\mathbf{y} \cup \mathbf{h}|} \tag{4}$$

Applications of U-Net

- **Medical Imaging:** Segmentation of tumors, organs, and tissues.
- **Satellite Image Analysis:** Land cover classification and urban planning.
- **Autonomous Driving:** Road scene understanding.
- **Agriculture:** Crop and plant segmentation from aerial images.

Section 4. Segment Anything Model (SAM)

What is SAM?

Key Idea

The Segment Anything Model (SAM) is a foundation model for segmentation developed by Meta AI. It can segment any object in an image using different input prompts, making it highly versatile.

- Works with bounding boxes, clicks, or freeform masks as prompts.
- Trained on SA-1B dataset with over 1 billion masks.
- Enables zero-shot segmentation on unseen images.

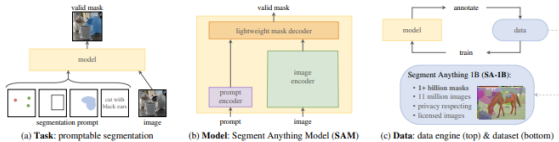


Figure: SAM Model Overview.

SAM Architecture

- **Image Encoder:** A large Vision Transformer (ViT) pre-trained using masked autoencoding.
- **Prompt Encoder:** Processes input prompts (points, boxes, masks).
- **Mask Decoder:** Predicts segmentation masks based on encoded image and prompts.

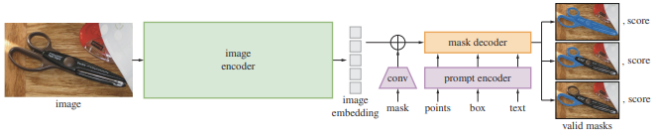


Figure: Segment Anything Model (SAM) Architecture.

Training and Dataset

- **SA-1B Dataset:** The largest segmentation dataset with 1 billion masks.
- **Semi-Automatic Annotation:** Human annotators refined AI-generated masks.
- **Generalization:** Trained to segment unseen objects and domains.

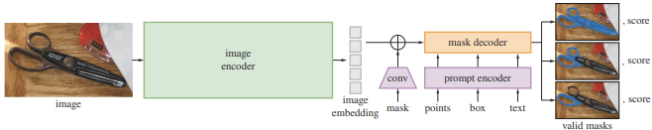


Figure: Segment Anything Model (SAM) Architecture.

How SAM Works?

- The user provides an input prompt (point, bounding box, or mask).
- SAM generates segmentation masks based on prompt and image encoding.
- Output can be refined interactively.

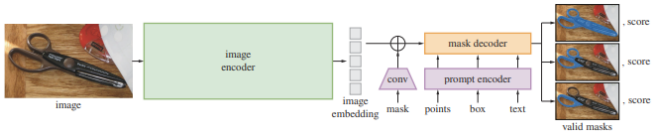


Figure: Segment Anything Model (SAM) Architecture.

Applications of SAM

- **Medical Imaging:** Tumor and organ segmentation.
- **Autonomous Driving:** Road scene segmentation.
- **AR/VR:** Object segmentation for augmented reality.
- **Robotics:** Object recognition and manipulation.

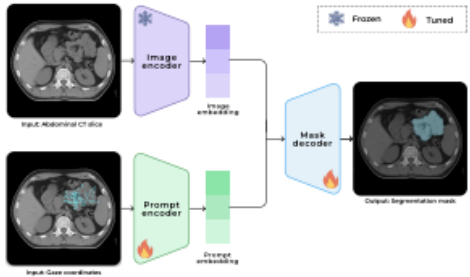


Figure: Medical Application of SAM.

Key Takeaways

- SAM is a powerful, zero-shot segmentation model.
- Works with multiple input prompts for flexible segmentation.
- Trained on the largest segmentation dataset, enabling generalization.
- Open-source and widely applicable across industries.

Section 5. Conclusion

Key Takeaways

- Segmentation and detection are essential for computer vision applications.
- Deep learning-based models have achieved state-of-the-art performance.
- Practical implementation involves dataset preparation, model training, and evaluation.

Bibliography

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020. URL https://arxiv.org/abs/2005.12872.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 40(4):834–848, 2018. doi: 10.1109/TPAMI.2017.2699184.

Kaiming He, Georgia Gkioxari, Piotr DollÃ¡r, and Ross Girshick. Mask r-cnn, 2018. URL https://arxiv.org/abs/1703.06870.

Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr DollÃ¡r. Panoptic segmentation, 2019. URL https://arxiv.org/abs/1801.00868.

Simon J.D. Prince. **Understanding Deep Learning**. The MIT Press, 2023. URL http://udlbook.com.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016. URL https://arxiv.org/abs/1506.02640.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. URL https://arxiv.org/abs/1506.01497.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation, 2020. URL https://arxiv.org/abs/2003.10152.