Chapter 8: Mastering Image Segmentation with CNNs

Data Mining

Instructor:

Debesh Jha, Ph.D.,

Visiting Assistant Professor,

University of South Dakota,

Vermillion, SD

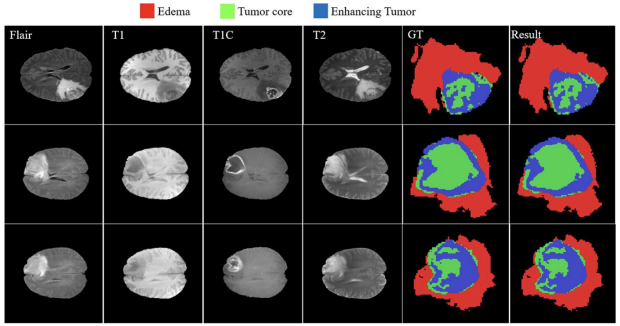
December 2, 2024

Outline

- Foundations of Image segmentation
- Traditional segmentation techniques
- Types of segmentation
- UNet and its variants
- Lightweight architectures
- Challenges in Image segmentation
- Solution to the challenge
- Explainability and visualization
- Evaluation and Metrics
- Future research directions

Why Image Segmentation?

- Critical for accurate decision-making in various fields.
  - Example: A missed tumor in radiology due to poor segmentation can lead to dire consequences.
- **Precise Localization**: Enables clinicians to identify tumor-affected areas and surrounding edema.
- **Tumor Diagnosis:** Assists in identifying tumor type and aggressiveness.
- **Personalized Treatment:** Guides surgeons in resecting tumors while preserving healthy tissues.
- **Quantitative Insights:** Provides metrics like tumor volume, surface area, and shape.



"Every pixel tells a story; segmentation ensures it's heard."

Challenges in Image Segmentation

- Data related challenges
  - Class imbalance
  - Annotation effort and quality
  - Noisy or low quality data
- Technical Challenges
  - Variability of object size
  - Computational costs
  - Boundary precision
  - Temporal consistency in videos
- Emerging challenges
  - LVM (e.g., SAM) requires fine tuning and has high computational requirement.

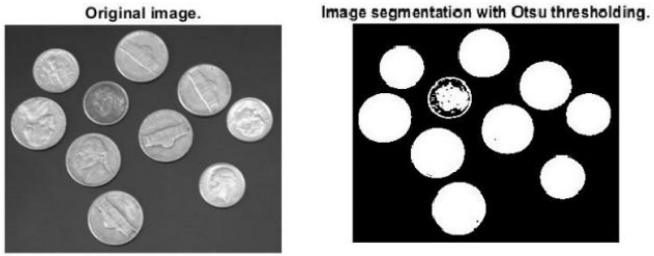
- Adversarial vulnerabilities
- Overdependence on pre-trained models

Importance of Image Segmentation

- **Medical Imaging:** Tumor detection, organ segmentation.
  a. **Diagnostic and Treatment:** Early detection, treatment planning
  b. **Clinical workflow:** Improving efficiency and effectiveness
  c. **Addressing challenges in Medicine:** Addressing variability in anatomical structures, contrast difference and noise.
- **Autonomous Vehicles:** Road and pedestrian detection.
- **Agriculture:** Crop health monitoring via satellite imagery.
- **Satellite Imaging:** Land-use classification.

Pre-deep Learning Image Segmentation Techniques - Broadly

- Thresholding (binary separation on intensity)
- Edge based segmentation(sobel, canny)
- Region based (watershed segmentation)
- Clustering (Graph based segmentation, k-means)
- Active contours
- Superpixel based segmentation
- Deep learning-based segmentation



Thresholding (binary separation on intensity)

- **Definition:** Thresholding is one of the simplest image segmentation techniques. It separates pixels into two or more classes based on their **intensity values** relative to a threshold.

- For binary thresholding, a single threshold value **T** is used. Pixels with intensities above T are classified as foreground, while those below are classified as background.

- For multi-level thresholding, multiple thresholds are used to separate the image into more than two classes.

Original Image

Thresholded and segmented Image

Edge based segmentation

- **Edge-based segmentation** identifies object boundaries in an image by detecting **sharp changes in intensity** (called "edges").

- It is useful for delineating **distinct regions** where there is a noticeable difference in intensity between adjacent areas (e.g., the boundary between an object and its background).

Image segmentation using the Sobel method.

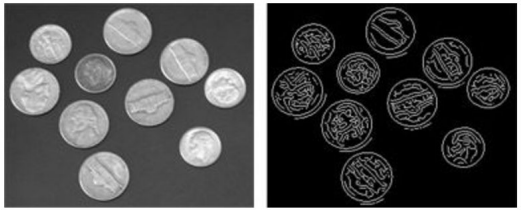
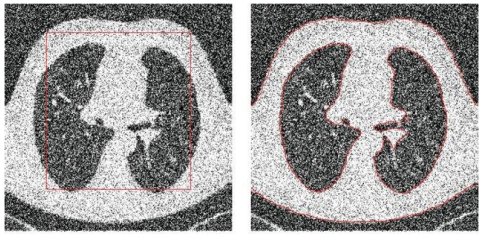


Image segmentation using the Canny method.

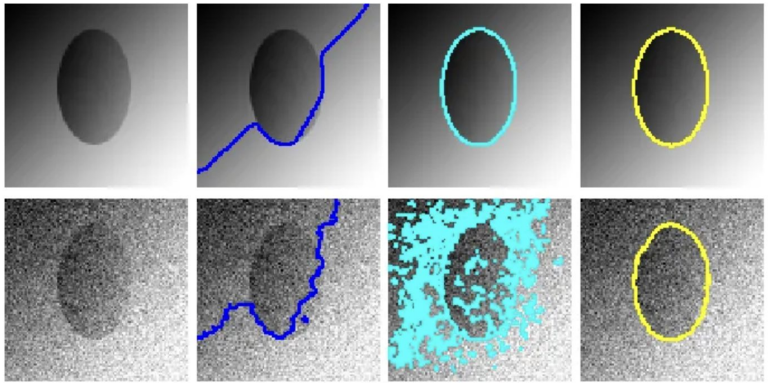


(a) (b)

Region-based segmentation

- Identifies regions of an image by grouping neighboring pixels with similar intensity, texture, or other properties.

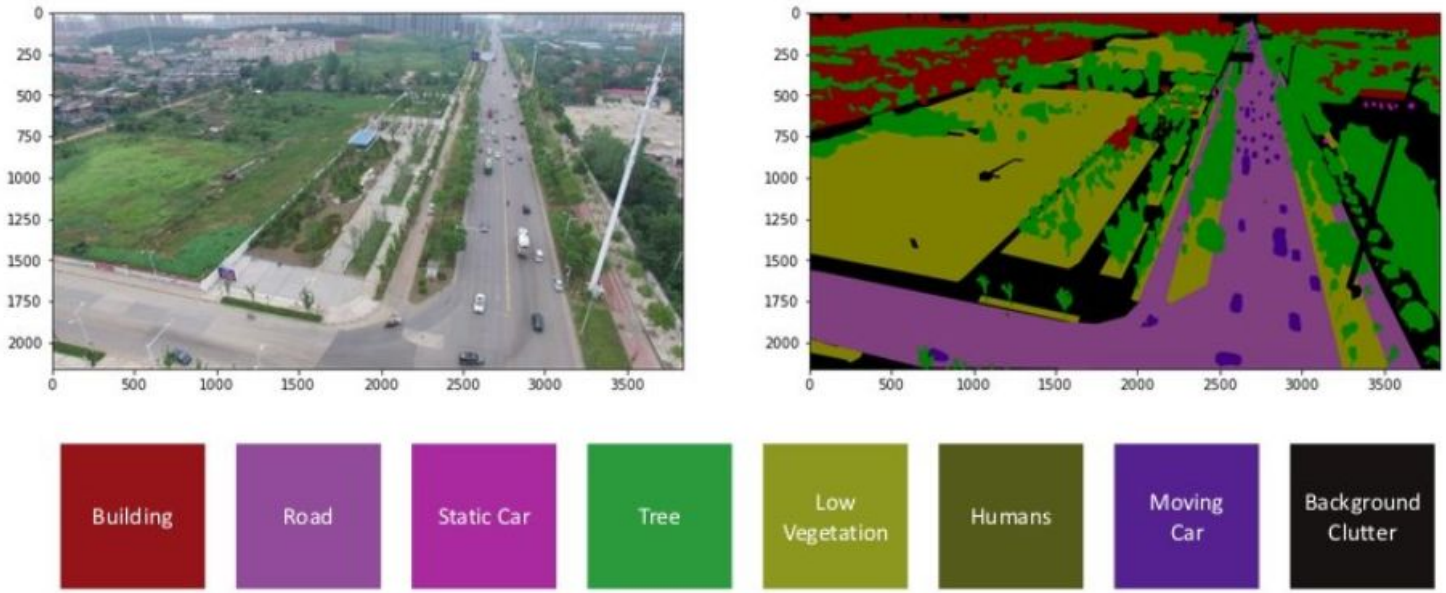
- This method assumes that regions belonging to the same object have consistent properties, such as grayscale values.



(a) Original image (b) Chan-Vese [3] (c) Li et al. [9] (d) L2S

Image segmentation

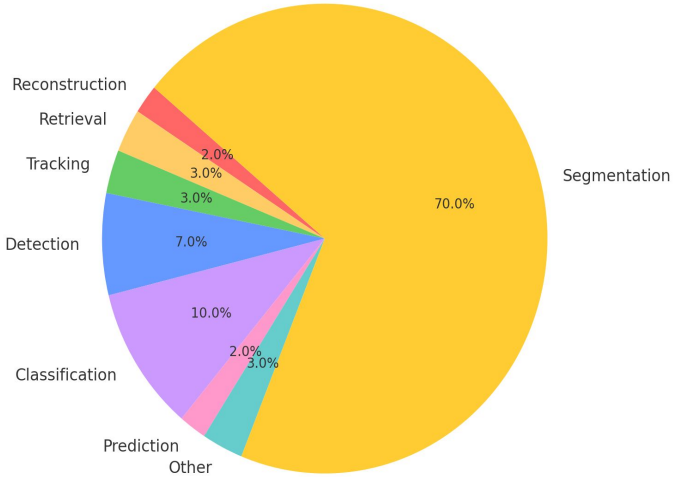
- Partitioning an image into **meaningful regions**.
- Each image is divided into **parts or segment** to simplify its representation and make it more meaningful and easier to analyze.



Importance of semantic segmentation

Comparison of Importance

1. **Granularity**:
   - **Segmentation > Detection > Classification**
   - Segmentation provides the most **detailed understanding**, followed by detection (bounding boxes), and classification (image-level labels).



Maier-Hein, L., Eisenmann, M., Reinke, A., Onogur, S., Stankovic, M., Scholz, P., ... & Kopp-Schneider, A. (2018). Why rankings of biomedical image analysis competitions should be interpreted with care. Nature communications, 9(1), 5217.

Type of Deep Learning Based Segmentation

Semantic Segmentation

- Classifies all pixels of an image into a **set of categories**, without **differentiating object instances**.

Instance segmentation

- Identifies **each instance** of each object present in an image separately, along with its **boundaries**.

Panoptic segmentation

- Combines semantic and instance segmentation, **classifying all pixels** and **differentiating object instances**.

Semantic and Instance Segmentation

1. Semantic Segmentation

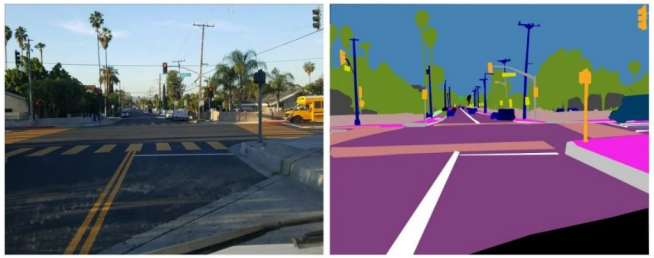
- Pixels are labeled with broad categories like road, sidewalk, sky, trees.
- No differentiation between individual objects of the same category (e.g., separate poles or trees).
- Example (Top Image): Entire road, buildings, and sky segmented into regions.

2. Instance Segmentation

- Each object of the same class is segmented individually (e.g., separate umbrellas and people).
- Example (Bottom Image): Different colors indicate individual persons and umbrellas.

3. Panoptic Segmentation

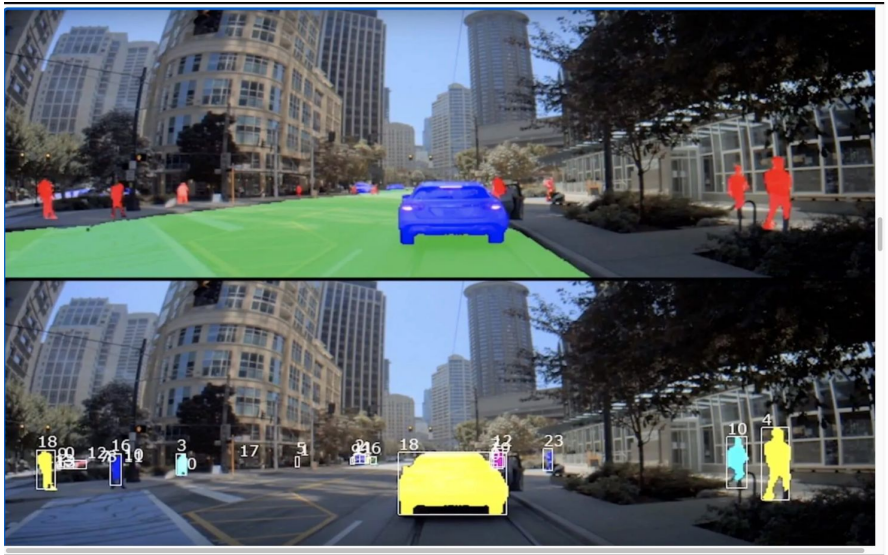
- Combines both semantic and instance segmentation.
- Example:
  - Roads and buildings as regions (semantic).
  - Poles and trees as distinct objects (instance).



1. **Top Image**
   Class identified: Green, blue, red
2. **Bottom Image**

Classes with Instance Differentiation:

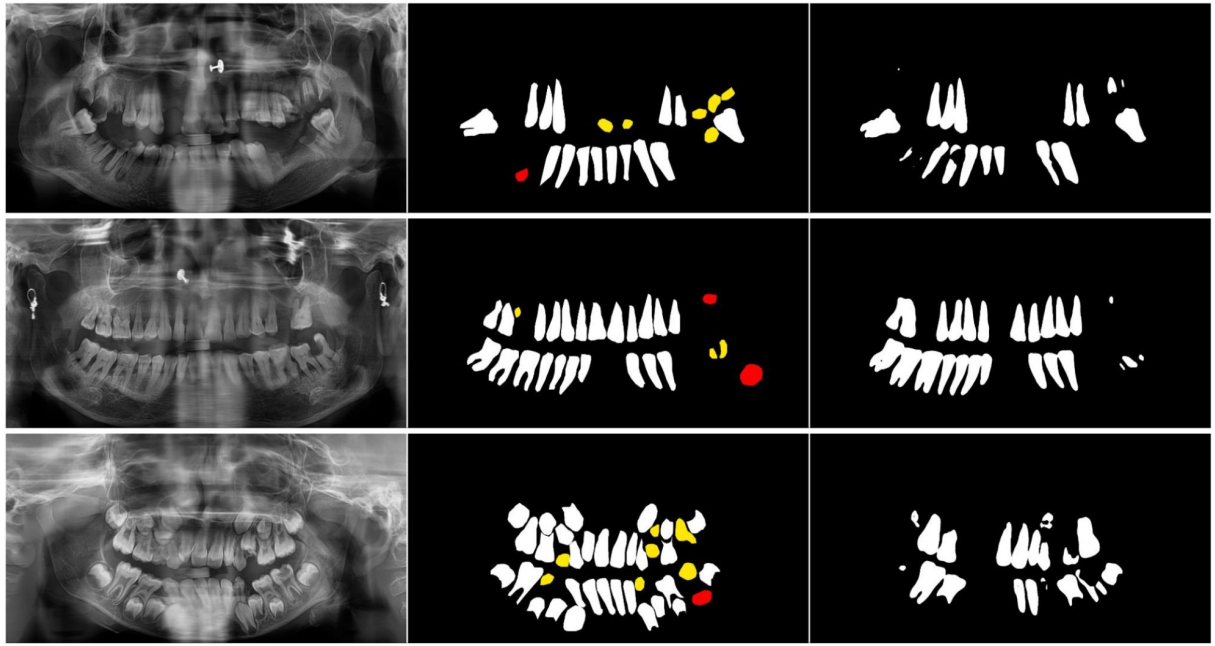
- Yellow: Specific car instance (Car ID: 18).
- Cyan/Red: Specific pedestrian instances (IDs 4, 10, etc.).



Pantopic segmentation:

a. **Background:** Labeled in semantic segmentation.
b. **Foreground:** Labeled in instance segmentation.

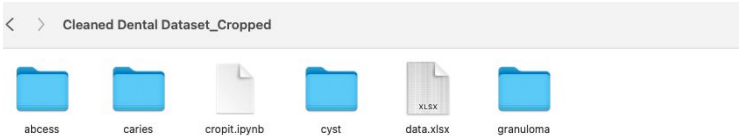
Class work (how many classes and what problem is this)?



- Three distinct foreground classes (teeth, yellow areas, red areas) and a background class. **Total number of class is 4**.
- It is semantic segmentation problem.
  It would be instance segmentation problem, if the goal is to classify individual teeth.

Overview of the problem

- Multi-class semantic segmentation of dental X-rays
- Severe class imbalance:
- Class 1: 800 images
- Class 2: 200 images
- Class 3: 200 images
- Class 4: 25 images
- Requires specialized strategies to handle imbalance.



Solution1: Data-level

- **Oversampling:**
  - Augment minority classes (e.g., rotation, flipping, elastic deformation).
  - Duplicate images from minority classes.
  - Color jittering or intensity adjustments (for dental x-ray, use **brightness/contrast scaling**)
- **Synthetic Data Generation:**
  - Use **GANs or diffusion models** to create new samples for underrepresented classes.
- **Class-Balanced Sampling:**
  - Ensure **each batch has balanced representation (balanced number of samples from each class)** of all classes.

Solution 2: Algorithm-level Solution

- **Weighted Loss Functions**:
  - Assign **higher weights** to **minority classes**.
- **Hybrid Loss Functions:**
  - Combine **Dice Loss** and **Weighted Cross-Entropy Loss**.
- **Focal Loss:**
  - Focus on **hard-to-classify** examples.
  - Use **attention mechanisms** to focus on **minority regions**.

Model training strategies

- **Pretraining:**
  - Use a **pre-trained model** to fine-tune on the dataset.
- **Curriculum Learning:**
  - Train on **majority classes first**, then introduce minority classes.
- **Batch Balancing:**
  - Ensure batches contain samples from all classes.
- Monitor **per-class metrics** during training.

Model Architecture

- Use **segmentation models** designed for medical imaging:
  - UNet, UNet++, DeepLabv3+, Attention UNet
- Ensure input images are **resized** and **normalized**:
  - Example: 256x256 resolution, pixel values scaled to [0, 1].

General Strategies

- Addressing class imbalance requires:
  - Data augmentation and synthetic generation.
  - Weighted and hybrid loss functions.
  - Balanced batches and pretraining.
  - Monitor per-class metrics and fine-tune strategies for best results.

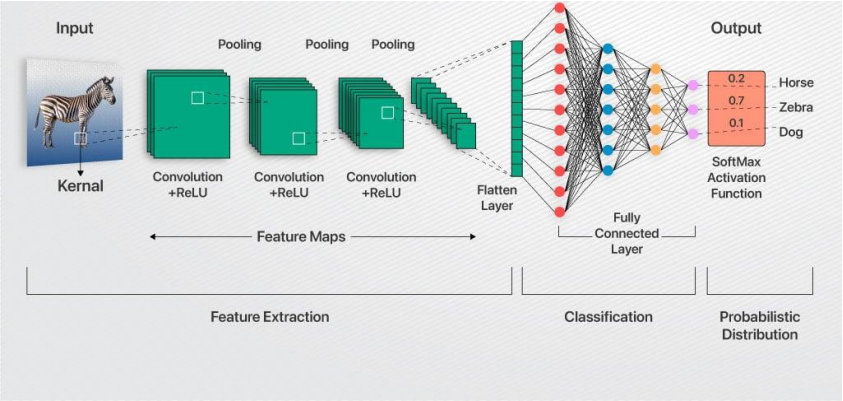
Why CNNs for Image Segmentation

- **Strengths of CNNs:**
  - Automatically learn features (edges, textures, shapes).
  - Handle complex images with high accuracy.
  - Enable pixel-wise classification.
- **Advantages Over Traditional Methods:**
  - Traditional Methods: Manual feature extraction (thresholding, edges).
  - CNN-Based Methods: End-to-end learning, robustness to noise, and better scalability.

Traditional vs CNN based approach

Aspect	Traditional approach	CNN-based approach	Real-world example
Feature extraction	Handcrafted features (edges, textures, requires domain expertise)	Learns features automatically during training	Edge based method struggle in medical imaging to identify boundaries
Robustness	Struggles with noisy or low-contrast images	Robust to noise, occlusions and variations in lighting	Thresholding fails in foggy conditions; CNN can detect lanes and obstacles
Precision	Region or boundary based, lacks pixel-wise accuracy	Pixel-level segmentation ensures detailed and accurate results	CNN can precisely identify tumor regions for surgical planning
Adaptability	Requires manual tuning for each dataset or domain	End-to-end traininable and adaptable via transfer learning	CNNs can identify forests and water bodies.

Fully Connected Layer



- A **fully connected layer** connects every input node to every output node.
- They take high-level feature representations (e.g., extracted edges or patterns) and produce a final prediction (e.g., "cat" or "dog").

Segmentation with CNNs: Fully Convolutional Network (FCN)

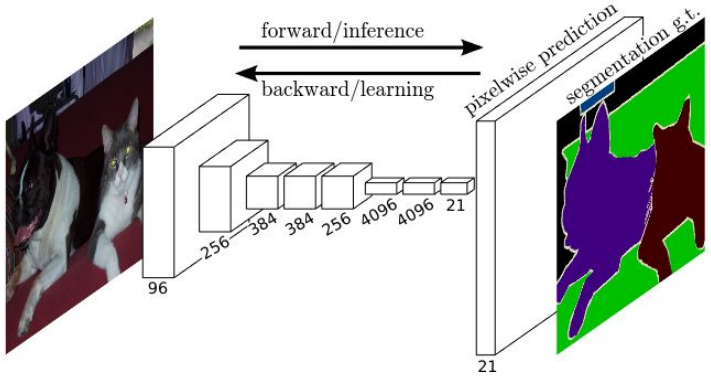
- A fully **convolutional** network is composed of **convolutional layers** without any **fully-connected** layers

- FCN preserves spatial information and it is useful for **segmentation** applications

- In contrast to fully connected network, the fully convolutional network can be applied to an input image of any size.
- Unlike networks with fully connected layers that flatten the input into a 1D vector, FCN retains the 2D spatial layout of the input image.
- This is critical for tasks like image segmentation, where each pixel needs to be assigned a specific label (e.g., "cat," "dog," or "background").

Fully Convolutional Network (FCN)

- **Input Image:** A 2D image of a **dog and a cat** is fed into the FCN.
- **Feature Extraction:** **Convolutional layers** extract hierarchical features like **edges, textures, and objects** while downsampling the image.
- **Upsampling (Deconvolution):** The extracted features are upsampled to the original image size using **transposed convolutions**, restoring resolution.
- **Output Segmentation Map:** The output is a **pixel-wise segmentation map** where each pixel is assigned a label (e.g., green for background, purple for the cat, and brown for the dog).



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* (pp. 3431-3440).

Why FCN become popular?

1. **End-to-End Learning for Pixel-Wise Predictions**

- **What it Solves:** Traditional methods relied on hand-crafted features or region-based approaches that lacked pixel-level precision.
- **FCN Contribution:** FCN introduced an architecture that predicts a label for every pixel in an image, enabling dense predictions (e.g., semantic segmentation).
- **Impact:** **Opened door for DL** to replace traditional handcrafted methods for segmentation tasks.

2. **Fully Convolutional Architecture**

- **What it Solves:** Previous NNs had fully connected layers that flattened image data, losing spatial information.
- **FCN Contribution:** Eliminated fully connected layers, using only convolutional layers, which preserved the spatial structure of the input image.
- Impact: Allowed FCNs to **process images of arbitrary size** and output **segmentation maps with the same aspect ratio**.

Why FCN become popular?

3. Upsampling with Deconvolution

- **What it Solves:** Pooling layers reduced resolution, making precise localization of objects challenging.
- **FCN Contribution:** Introduced **deconvolution (transposed convolution)** to upsample low-resolution feature maps back to the original resolution.
    - **Impact:** Enabled accurate localization of objects while retaining high-level semantic information.

4. Skip Connections

- **What it Solves:** Pooling layers discarded fine-grained spatial details.
- **FCN Contribution:** Introduced **skip connections** from earlier layers to the upsampled layers, which combined high-resolution spatial features with high-level semantic features.
    - **Impact:** Improved the accuracy of segmentation, particularly for small object.

Why FCN become popular?

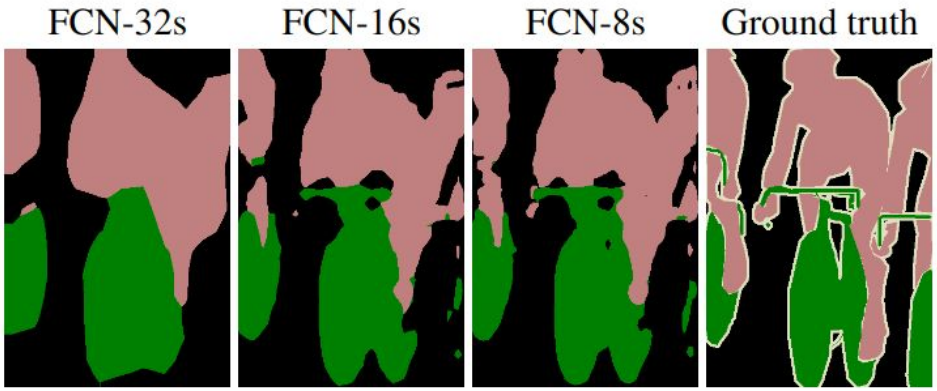
5. **Generalization Across Domains**

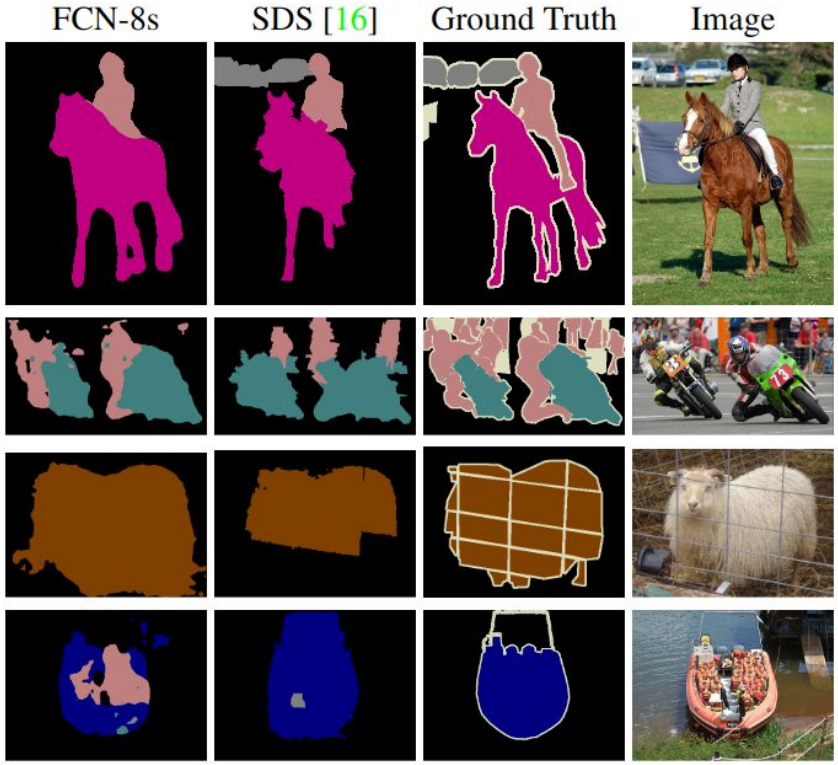
- **What it Solves:** Traditional segmentation methods often failed when applied to diverse datasets.
- **FCN Contribution:** Provided a general framework for segmentation tasks across various domains.
- Impact: Significantly reduced the need for task-specific designs and hand-tuned features.

6. Multi-Scale Contextual Learning

- **What it Solves:** Earlier models lacked the ability to integrate both local (fine details) and global (contextual) information.
- **FCN Contribution:** Learned hierarchical features, from edges in shallow layers to complex patterns in deeper layers, integrating context for better segmentation.
    - Impact: Helped in segmenting objects of varying sizes effectively.

Qualitative Results

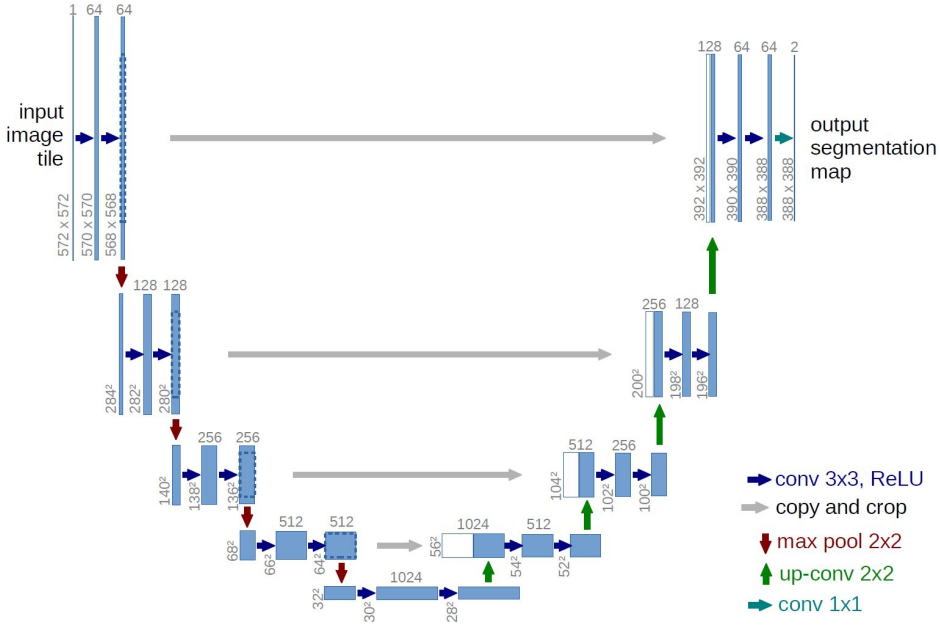




- **FCN-32s:** Coarsest resolution. Upsamples feature maps by a factor of 32.
- **FCN-16s:** Medium resolution. Uses finer upsampling by a factor of 16, improving spatial accuracy.
- **FCN-8s:** Finest resolution.

U-Net: A Landmark Architecture

- The architecture comprises a **contracting path** that **captures context** and an **expanding path** that **helps localization.**
- The **contracting path** consists of two 3x3 convolutions, each followed by a rectified linear unit (ReLU) and a max pooling operation.
- The **expansive path** is more or less symmetric to the contracting path and consists of **upsampling** of the feature map.



Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18* (pp. 234-241). Springer International Publishing.

FCN vs U-Net Comparison

Feature	FCN	U-Net
Localization	Limited due to coarse upsampling.	Enhanced with skip connections that retain spatial details.
Architecture	Asymmetrical, focusing more on downsampling than upsampling.	Symmetrical encoder-decoder structure for balanced feature learning
Small dataset support	Requires large datasets to generalize well.	Performs effectively with small datasets using data augmentation.
Fine structures	Struggles with small and thin structures due to information loss.	Captures small details effectively, such as blood vessels or thin edges.
Applications	General purpose segmentation.	Highly effective in biomedical image segmentation

Why U-Net become so popular?

- **Problem Solved:**
  - In traditional architectures like FCNs, upsampling (or deconvolution) often failed to recover fine-grained spatial details, especially for small objects or boundaries.
- **U-Net Solution:**
  - U-Net introduced skip connections that directly transfer feature maps from the contracting (downsampling) path to the expanding (upsampling) path.
  - This allowed U-Net to combine low-level spatial features (from earlier layers) with high-level semantic features (from deeper layers), improving segmentation accuracy.
- **Impact:**
  - Enhanced localization and made it possible to accurately segment small structures like tumors, blood vessels, or cell membranes in medical images.

Symmetric Encoder-Decoder Architecture

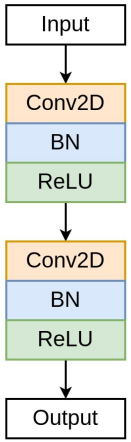
- **Problem Solved:**
  - Existing models like FCN lacked balance between downsampling (context extraction) and upsampling (spatial reconstruction), leading to coarse results.
- **U-Net Solution:**
  - U-Net introduced a symmetrical architecture, where:
  - The encoder path (contracting path) extracts semantic features by downsampling.
  - The decoder path (expanding path) reconstructs the spatial resolution while refining predictions.
- **Impact:**
  - Provided a structured framework that systematically recovered spatial information, improving the quality of segmentation maps.

Effective with Small Datasets

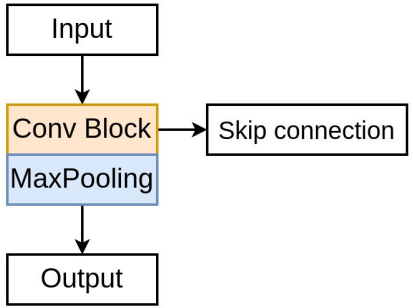
- **Problem Solved:**
  - Many segmentation tasks, especially in medical imaging, have limited labeled data, making it difficult for deep learning models to perform well.
- **U-Net Solution:**
  - U-Net used extensive data augmentation strategies to increase the diversity of the training set artificially, making it robust to small datasets.
- **Impact:**
  - Made U-Net suitable for fields like biomedical imaging, where annotating large datasets is challenging and expensive.

- The main contribution of U-Net lies in its skip connections, symmetrical encoder-decoder architecture, and ability to work with small datasets. These innovations made U-Net a gold standard for image segmentation tasks, particularly in medical imaging, and a foundational model that inspired many subsequent architectures.

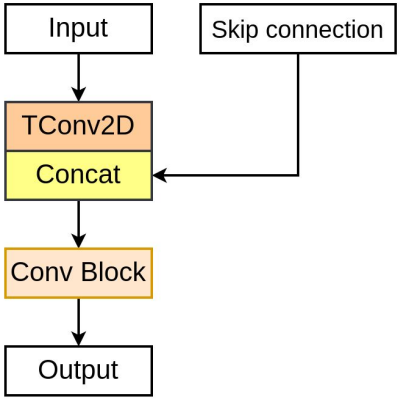
Convolutional, Encoder and Decoder Block



Fundamental building block



Downsampling path of UNet



Upsampling path of U-Net

Fundamental Building Block

- **What It Represents:**
  - This is the basic unit of the U-Net architecture, used in both the encoder and decoder.
- **Components:**
  - Conv2D:
  - Applies convolutional filters to the input to extract features (e.g., edges, textures, shapes).
- **Batch Normalization (BN):**
  - Normalizes the feature maps to stabilize learning and speed up convergence.
- **ReLU (Rectified Linear Unit):**
  - Introduces non-linearity, allowing the network to learn complex patterns.
- **Workflow:**
- Input → Conv2D → BN → ReLU → Conv2D → BN → ReLU → Output Feature Map.
- **Key Point:**
  - This block learns hierarchical features at every stage of the U-Net, from low-level edges to high-level object shapes.

Downsampling Path (Encoder Block)

- What It Represents:
  - The encoder block in U-Net is part of the downsampling path. It reduces the spatial dimensions while extracting high-level semantic features.
- Components:
  - Conv Block:
    - A stack of convolutional layers that processes the input and extracts meaningful features.
  - MaxPooling:
    - Reduces the spatial resolution (e.g., by taking the maximum value in a 2x2 region). This focuses on key features and reduces computation.
  - Skip Connection:
    - The output feature map is saved and passed directly to the decoder for reconstruction, ensuring spatial details are preserved.
- Workflow:
  - Input → Conv Block → MaxPooling → Feature Map + Skip Connection.
- Key Point:
  - The downsampling path captures contextual information (semantic understanding of the image).

Upsampling Path (Decoder Block)

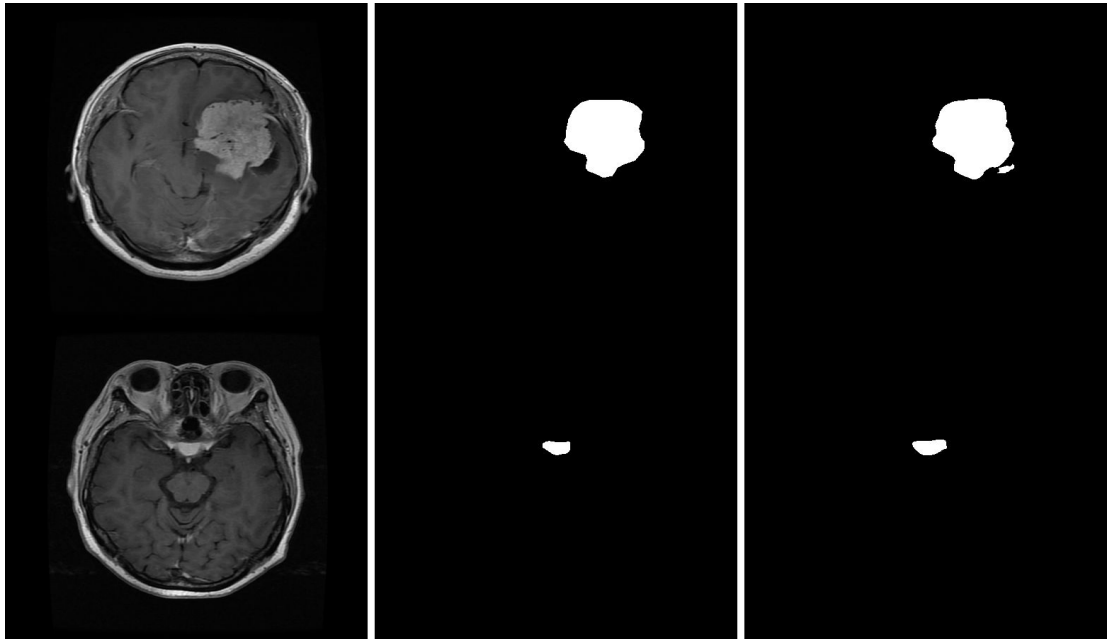
- **What It Represents:**
  - The **decoder block** in U-Net is part of the upsampling path. It restores the original resolution of the input while combining semantic and spatial features.
- **Components:**
  - **TConv2D (Transposed Convolution):**
    - Upsamples the feature map, increasing spatial resolution by reversing the effect of MaxPooling.
  - **Concatenate:**
    - Combines the upsampled feature map with the corresponding feature map from the encoder (via skip connections). This merges spatial details with semantic information.
  - **Conv Block:**
    - Further processes the combined feature map to refine details and improve prediction accuracy.
- **Workflow:**
  - Input → TConv2D → Concatenate (with skip connection) → Conv Block → Output Feature Map.
- **Key Point:**
  - The upsampling path reconstructs the image resolution and **localizes fine-grained details**.

Importance of Skip Connections

- Skip connections directly transfer spatial features from the encoder to the decoder, addressing:
    1. **Information Loss:** Prevents loss of spatial details during pooling.
    2. **Better Localization:** Enhances segmentation accuracy, especially for small or thin structures (e.g., blood vessels, tumor boundaries).
- **Downsampling Path:** Captures context and reduces spatial dimensions.
- **Upsampling Path:** Restores resolution while localizing objects using skip connections.
- **Fundamental Building Block:** Learns meaningful features at every stage.

Results on Brain Tumor Segmentation on MRI

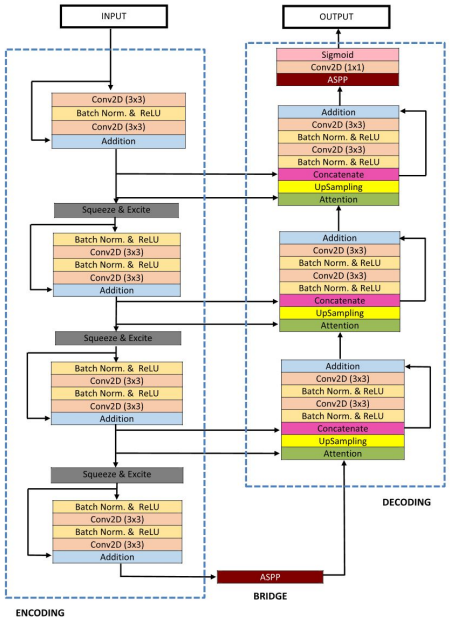
Input Image | Ground Truth | Prediction



FCN vs UNet Comparison

S.N	Feature	FCN advantage	FCN disadvantages	UNet Advantages	UNet disadvantage
1	Training Data	**Can be trained end-to-end with patches**	**Requires more data** and **can be slow** due to processing each patch separately.	Requires **fewer images for training**, can be trained end-to-end with entire images.	-
2	Localization	Capable of **localizing features in images**.	Larger patches may **reduce localization accuracy** due to more max-pooling layers.	Better localization due to **skip connections** that provide local information during upsampling.	-
3	Context	Can **utilize context** from larger patches.	**Trade-off between localization and context**; smaller patches provide less context.	**Symmetric architecture** helps in capturing **better context** from images.	-
5	Computational Efficiency	-	May be **computationally intensive** due to patch-wise processing.	Tiling strategy allows **handling large images efficiently.**	-

ResUNet++

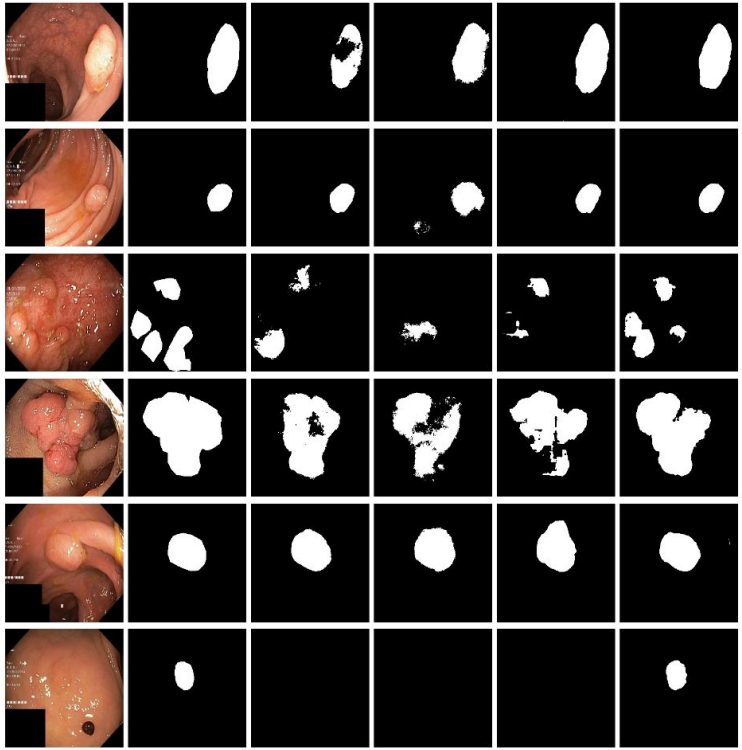


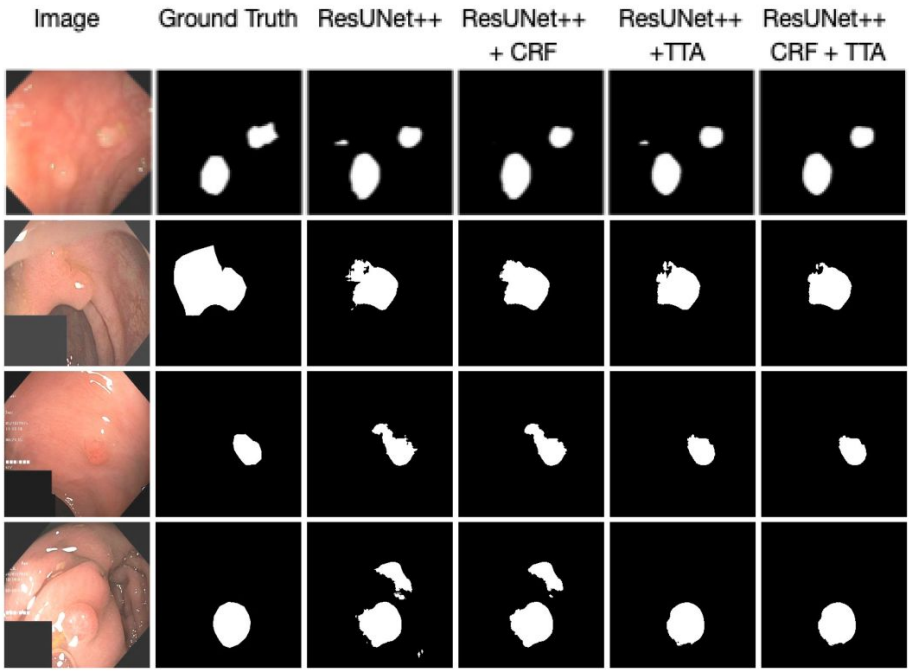
Method	Dice	mIoU	Recall	Precision
ResUNet++	**0.8133**	**0.7927**	**0.7064**	0.8774
ResUNet-mod	0.7909	0.4287	0.6909	0.8713
ResUNet	0.5144	0.4364	0.5041	0.7292
U-Net	0.7147	0.4334	0.6306	**0.9222**

Method	Dice	mIoU	Recall	Precision
ResUNet++	**0.7955**	**0.7962**	**0.7022**	0.8785
ResUNet-mod	0.7788	0.4545	0.6683	**0.8877**
ResUNet	0.4510	0.4570	0.5775	0.5614
U-Net	0.6419	0.4711	0.6756	0.6868

Jha, D., Smedsrud, P. H., Riegler, M. A., Johansen, D., De Lange, T., Halvorsen, P., & Johansen, H. D. (2019, December). Resunet++: An advanced architecture for medical image segmentation. In *2019 IEEE international symposium on multimedia (ISM)* (pp. 225-2255). IEEE.

Qualitative Results Comparison



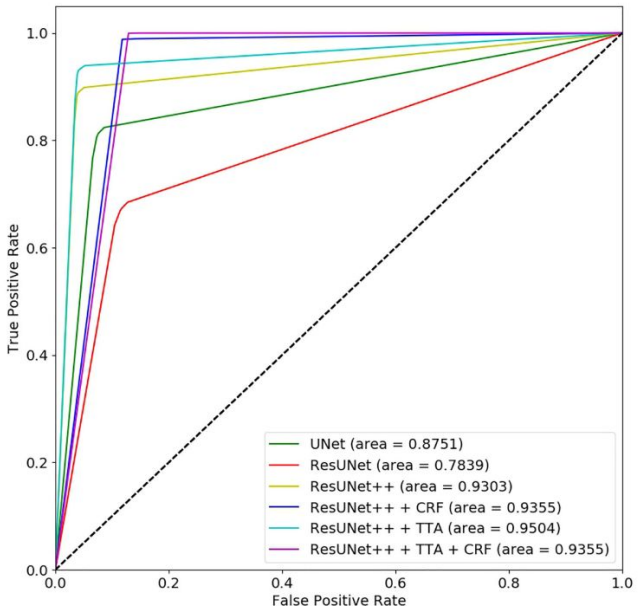


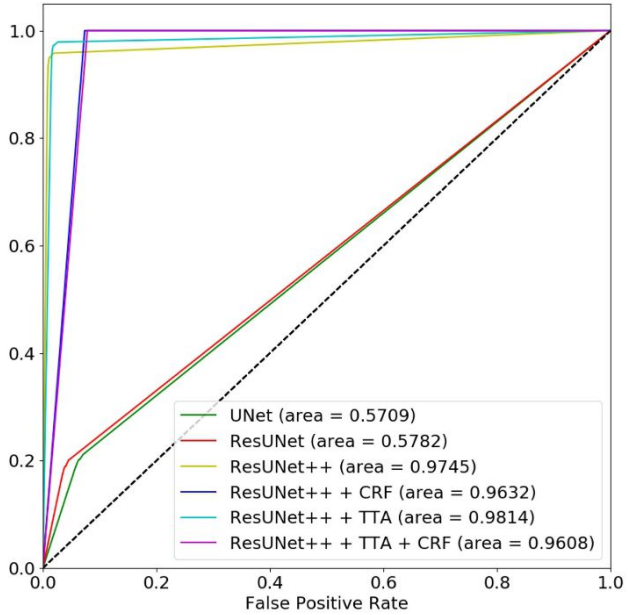
Results on CVC-VideoClinicDB and MayoClinic Dataset

Method	DSC	mIoU	Recall	Precision
ResUNet++ [1]	0.8798	0.8730	**0.7749**	0.6702
ResUNet++ + CRF	**0.8811**	**0.8739**	0.7743	**0.6706**
ResUNet++ + TTA	0.8125	0.8467	0.6896	0.6421
ResUNet++ + TTA + CRF	0.8130	0.8477	0.6875	0.6276

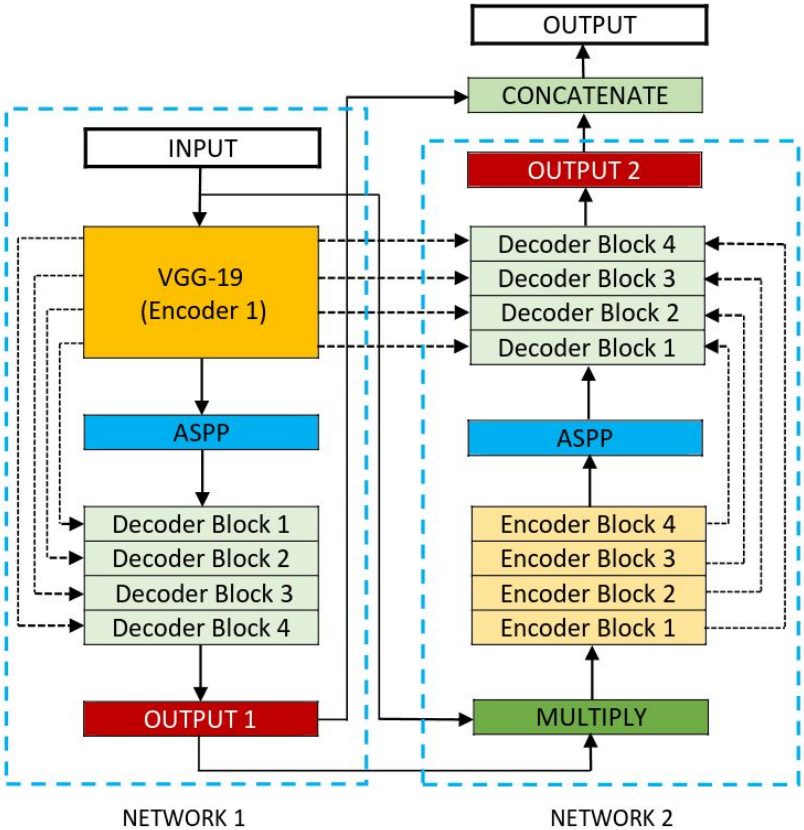
Method	DSC	mIoU	Recall	Precision
ResUNet++ [1]	0.8743	0.8569	**0.6534**	0.4896
ResUNet++ + CRF	**0.8850**	**0.8635**	0.6504	0.4858
ResUNet++ + TTA	0.8553	0.8535	0.6162	**0.4912**
ResUNet++ + TTA + CRF	0.8550	0.8551	0.6107	0.4743

AUC Curve


True Positive Rate
False Positive Rate
UNet (area = 0.8751)
ResUNet (area = 0.7839)
ResUNet++ (area = 0.9303)
ResUNet++ + CRF (area = 0.9355)
ResUNet++ + TTA (area = 0.9504)
ResUNet++ + TTA + CRF (area = 0.9355)

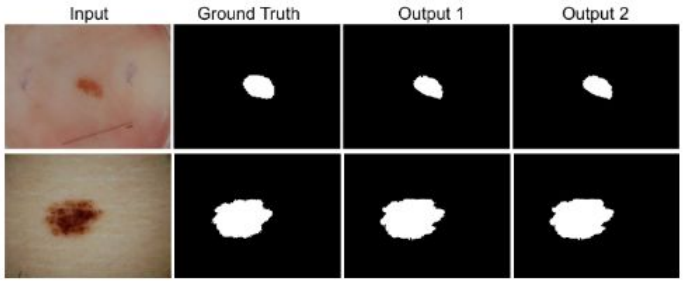

False Positive Rate
UNet (area = 0.5709)
ResUNet (area = 0.5782)
ResUNet++ (area = 0.9745)
ResUNet++ + CRF (area = 0.9632)
ResUNet++ + TTA (area = 0.9814)
ResUNet++ + TTA + CRF (area = 0.9608)

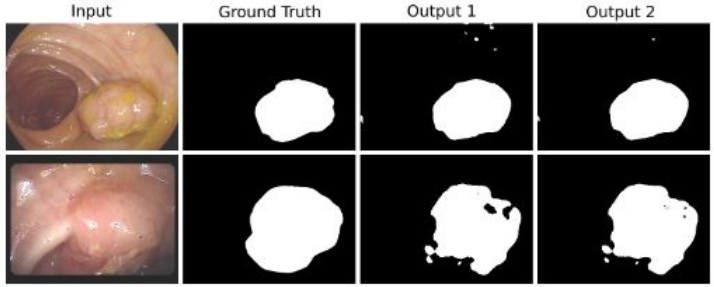
DoubleUNet

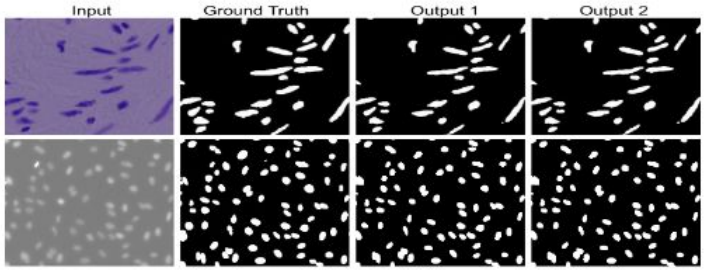


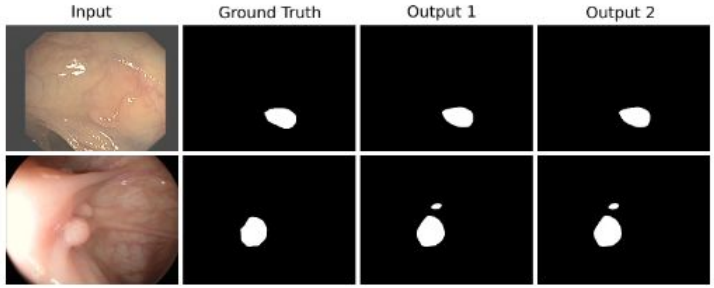
Jha, D., Riegler, M. A., Johansen, D., Halvorsen, P., & Johansen, H. D. (2020, July). Doubleu-net: A deep convolutional neural network for medical image segmentation. In *2020 IEEE 33rd International symposium on computer-based medical systems (CBMS)* (pp. 558-564). IEEE.

Qualitative Results









Quantitative Results

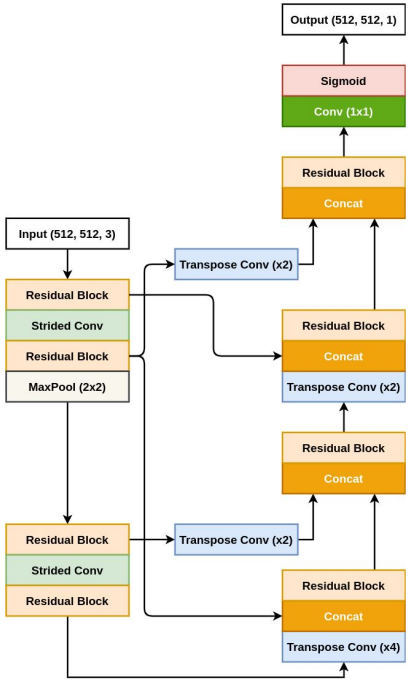
Method	Pre-trained network	DSC	mIoU	Recall	Precision
U-Net	Resnet101	0.7573	0.9103	-	-
UNet++	Resnet101	0.8974	**0.9255**	-	-
DoubleU-Net	VGG-19	**0.9133**	0.8407	**0.6407**	**0.9496**

Dataset	UNet DSC (%)	DoubleUNet DSC (%)	Improvement
MICCAI 2015	29.20	76.49	**47.29%**
CVC-ClinicDB	87.81	92.39	**4.58%**
ISIC-2018	85.54	89.38	**3.84%**
2018 Data Science Bowl	0.7573	0.9133	**15.60%**

ResUNet++ vs DoubleUNet

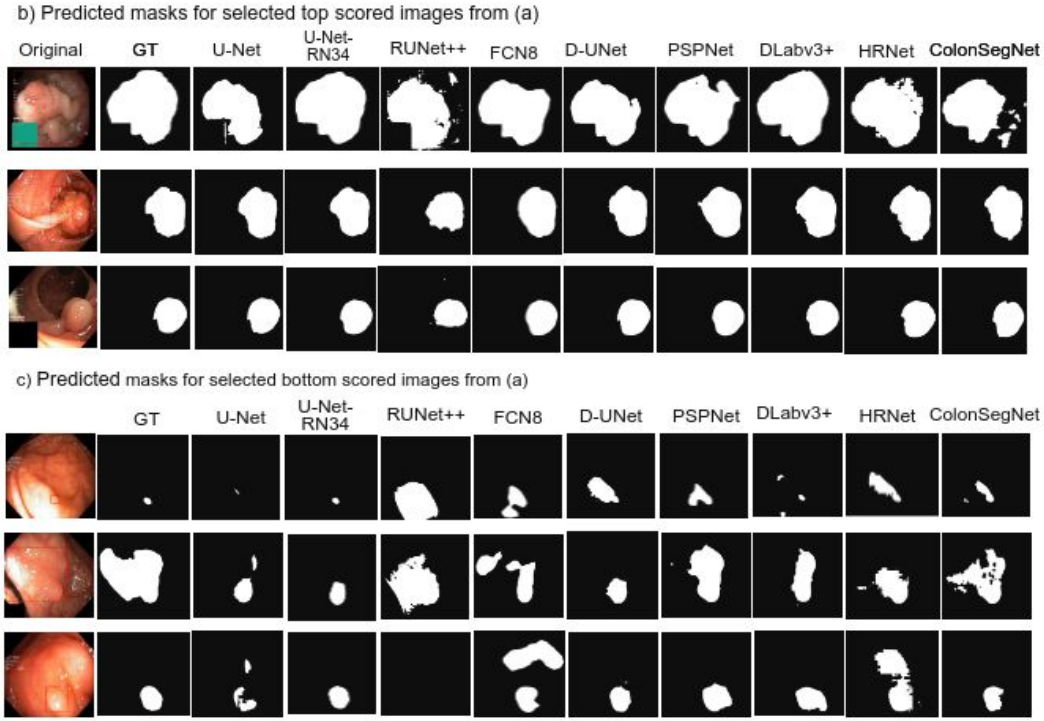
Features	ResUNet++ advantages	ResUNet++ disadvantages	DoubleUNet advantages	DoubleUNet disadvantages
Architecture	Incorporates residual connections	More complex due to advanced component	Stacks two U-Net with a VGG-19 backbone	Increased model complexity
Performance	Achieves high DSC and mIoU	Requires careful tuning	Demonstrates superior performance	May overfit on smaller datasets without data augmentation or regularization.
Usability	Offers strong solution for medical image segmentation	Sophisticated architecture	Provides robust baseline for medical image segmentation tasks	Requires high computation resources.

Lightweight Segmentation Architecture: ColonSegNet

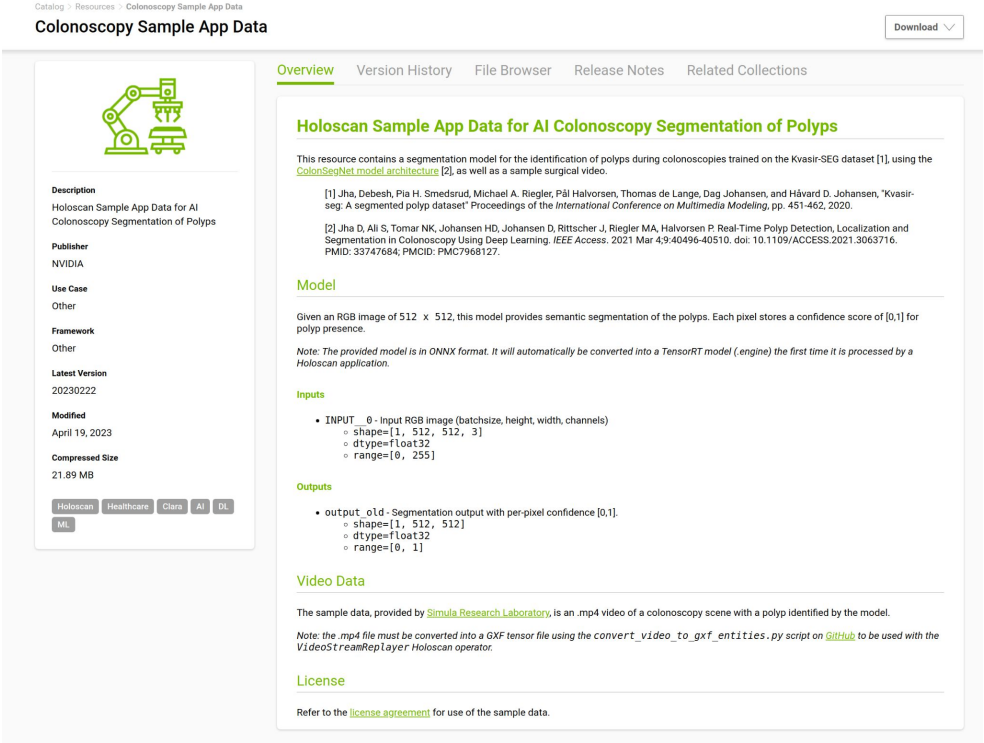


Method	Backbone	Jaccard	DSC	F2-score	Precision	Recall	FPS
UNet	-	0.4713	0.5969	0.5980	0.6722	0.6171	11.01
ResUNet	-	0.5721	0.6902	0.6986	0.7454	0.7248	14.82
ResUNet++	-	0.6126	0.7143	0.7198	0.7836	0.7419	7.01
FCN8	VGG 16	0.7365	0.8310	0.8248	0.8817	0.8346	24.91
HRNet	-	0.7592	0.8446	0.8467	0.8778	0.8588	11.69
DoubleUNet	VGG 19	0.7332	0.8129	0.8207	0.8611	0.8402	7.46
PSPNet	ResNet50	0.7444	0.8406	0.8314	0.8901	0.8357	16.80
DeepLabv3+	ResNet50	0.7759	0.8572	0.8545	0.8907	**0.8616**	27.90
DeepLabv3+	ResNet101	**0.7862**	**0.8643**	**0.8570**	**0.9064**	0.8592	16.75
UNet	ResNet34	**0.8100**	**0.8757**	**0.8622**	**0.9435**	**0.8597**	**35.00**
ColonSegNet	-	0.7239	0.8206	0.8206	0.8435	0.8496	**182.38**

Qualitative Results



Practical Use Case of ColonSegNet



Catalog > Resources > Colonoscopy Sample App Data

Colonoscopy Sample App Data

Download

Overview | Version History | File Browser | Release Notes | Related Collections

Description
Holoscan Sample App Data for AI Colonoscopy Segmentation of Polyps

Publisher
NVIDIA

Use Case
Other

Framework
Other

Latest Version
20230222

Modified
April 19, 2023

Compressed Size
21.89 MB

Holoscan | Healthcare | Clara | AI | DL | ML

Holoscan Sample App Data for AI Colonoscopy Segmentation of Polyps

This resource contains a segmentation model for the identification of polyps during colonoscopies trained on the Kvasir-SEG dataset [1], using the ColonSegNet model architecture [2], as well as a sample surgical video.

[1] Jha, Debesh, Pia H. Smedsrud, Michael A. Riegler, Pål Halvorsen, Thomas de Lange, Dag Johansen, and Håvard D. Johansen, "Kvasir-seg: A segmented polyp dataset" Proceedings of the *International Conference on Multimedia Modeling*, pp. 451-462, 2020.

[2] Jha D, Ali S, Tomar NK, Johansen HD, Johansen D, Rittscher J, Riegler MA, Halvorsen P. Real-Time Polyp Detection, Localization and Segmentation in Colonoscopy Using Deep Learning. *IEEE Access*. 2021 Mar 4;9:40496-40510. doi: 10.1109/ACCESS.2021.3063716. PMID: 33747684; PMCID: PMC7968127.

Model

Given an RGB image of `512 x 512`, this model provides semantic segmentation of the polyps. Each pixel stores a confidence score of [0,1] for polyp presence.

Note: The provided model is in ONNX format. It will automatically be converted into a TensorRT model (.engine) the first time it is processed by a Holoscan application.

Inputs

- `INPUT__0` - Input RGB image (batchsize, height, width, channels)
  - `shape=[1, 512, 512, 3]`
  - `dtype=float32`
  - `range=[0, 255]`

Outputs

- `output_old` - Segmentation output with per-pixel confidence [0,1].
  - `shape=[1, 512, 512]`
  - `dtype=float32`
  - `range=[0, 1]`

Video Data

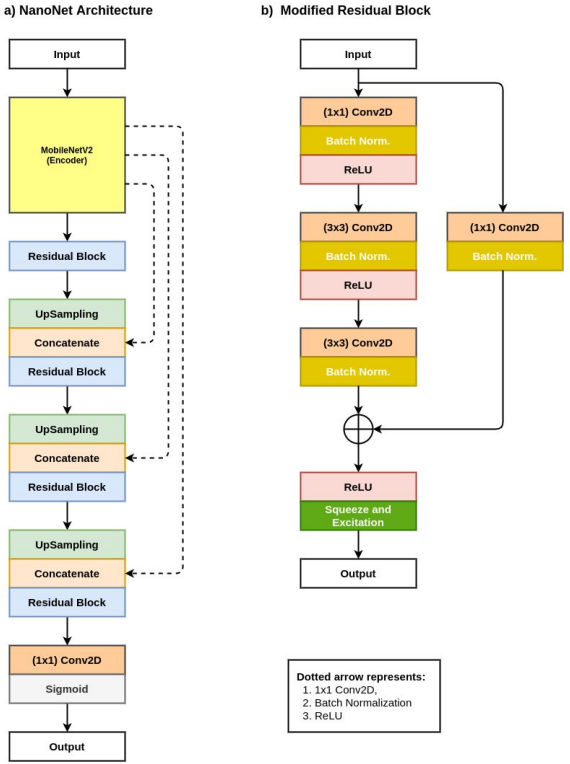
The sample data, provided by Simula Research Laboratory, is an .mp4 video of a colonoscopy scene with a polyp identified by the model.

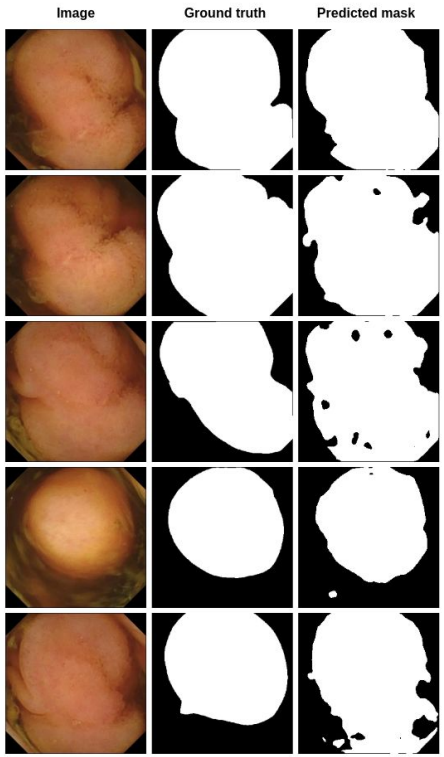
Note: the .mp4 file must be converted into a GXF tensor file using the `convert_video_to_gxf_entities.py` script on GitHub to be used with the `VideoStreamReplayer` Holoscan operator.

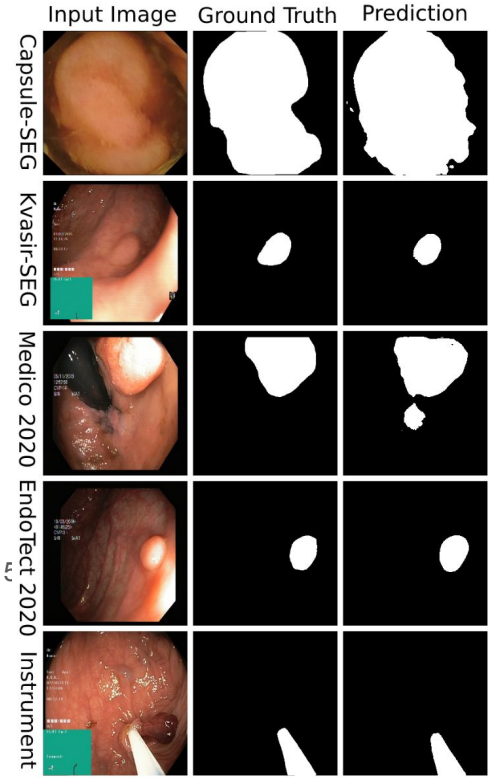
License

Refer to the license agreement for use of the sample data.

Lightweight Segmentation Architecture: NanoNet







Method	Parameters	DSC	mIoU	Recall	Precision	F2	Accuracy	FPS
ResUNet (GRSL'18) [38]	8,227,393	**0.9532**	**0.9137**	**0.9785**	**0.9325**	**0.9677**	**0.9386**	17.96
ResUNet++ (ISM'19) [24]	4,070,385	0.9499	0.9087	0.9762	0.9296	0.9648	0.9334	15.39
NanoNet-A (Ours)	235,425	0.9493	0.9059	0.9693	**0.9325**	0.9609	0.9351	28.35
NanoNet-B (Ours)	132,049	0.9474	0.9028	0.9682	0.9308	0.9593	0.9324	27.39
NanoNet-C (Ours)	36,561	0.9465	0.9021	0.9754	0.9238	0.9629	0.9297	**29.48**

Method	Parameters	DSC	mIoU	Recall	Precision	F2	Accuracy	FPS
ResUNet (GRSL'18) [38]	8,227,393	0.7203	0.6106	0.7602	0.7624	0.7327	0.9251	17.72
ResUNet++ (ISM'19) [24]	4,070,385	0.7310	0.6363	0.7925	0.7932	0.7478	0.9223	19.79
NanoNet-A (Ours)	235,425	**0.8227**	**0.7282**	**0.8588**	**0.8367**	**0.8354**	**0.9456**	26.13
NanoNet-B (Ours)	132,049	0.7860	0.6799	0.8392	0.8004	0.8067	0.9365	29.73
NanoNet-C (Ours)	36,561	0.7494	0.6360	0.8081	0.7738	0.7719	0.9290	**32.17**

Method	Parameters	DSC	mIoU	Recall	Precision	F2	Accuracy	FPS
ResUNet (GRSL'18) [38]	8,227,393	0.6846	0.5599	0.7235	0.7236	0.6961	**0.9231**	18.54
ResUNet++ (ISM'19) [24]	4,070,385	0.6925	0.5849	0.8249	0.6840	0.7434	0.8995	19.47
NanoNet-A (Ours)	235,425	0.7364	**0.6319**	**0.8566**	0.7310	**0.7804**	0.9166	28.07
NanoNet-B (Ours)	132,049	**0.7378**	0.6247	0.8283	**0.7373**	0.7685	0.9223	29.04
NanoNet-C (Ours)	36,651	0.7070	0.5866	0.8095	0.7089	0.7432	0.9148	**32.66**

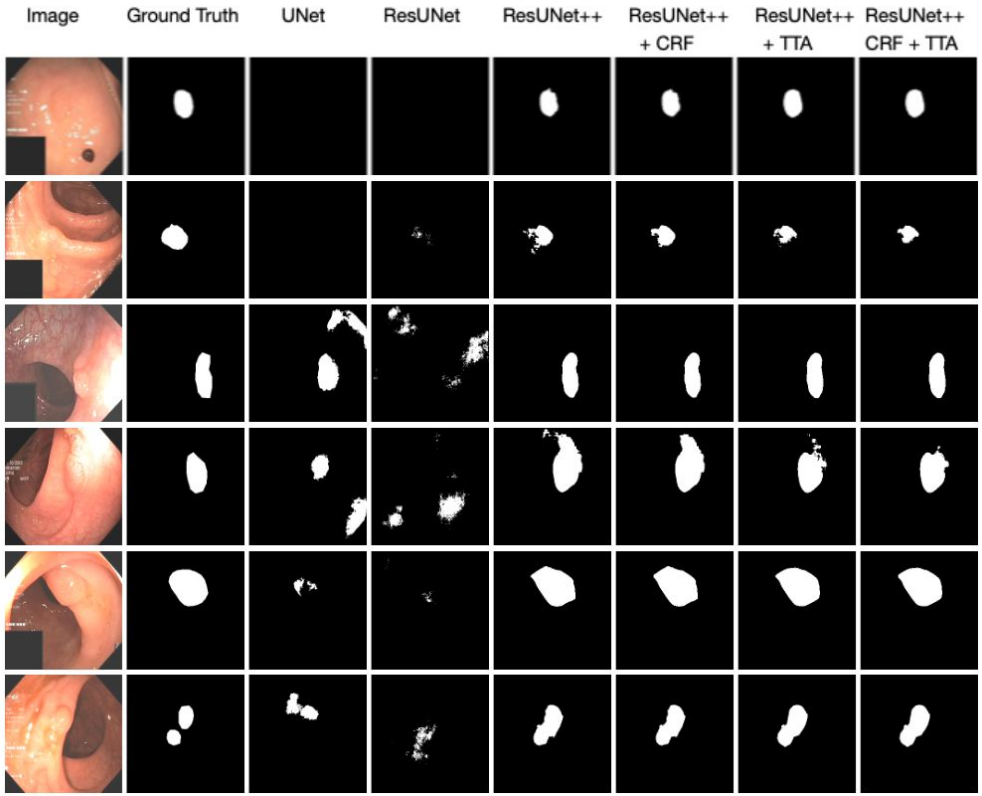
Method	Parameters	DSC	mIoU	Recall	Precision	F2	Accuracy	FPS
ResUNet (GRSL'18) [39]	8,227,393	0.6640	0.5408	0.7510	0.6841	0.6943	0.9075	26.55
ResUNet++ (ISM'19) [24]	4,070,385	0.6940	0.5838	**0.8797**	0.6591	0.7597	0.8841	18.58
NanoNet-A (Ours)	235,425	**0.7508**	**0.6466**	0.8238	**0.7744**	**0.7773**	**0.9255**	27.19
NanoNet-B (Ours)	132,049	0.7362	0.6238	0.8109	0.7532	0.7646	0.9252	29.91
NanoNet-C (Ours)	36,651	0.7001	0.5792	0.8000	0.7159	0.7380	0.9091	**32.98**

NanoNet vs ColonSegNet vs DDANet

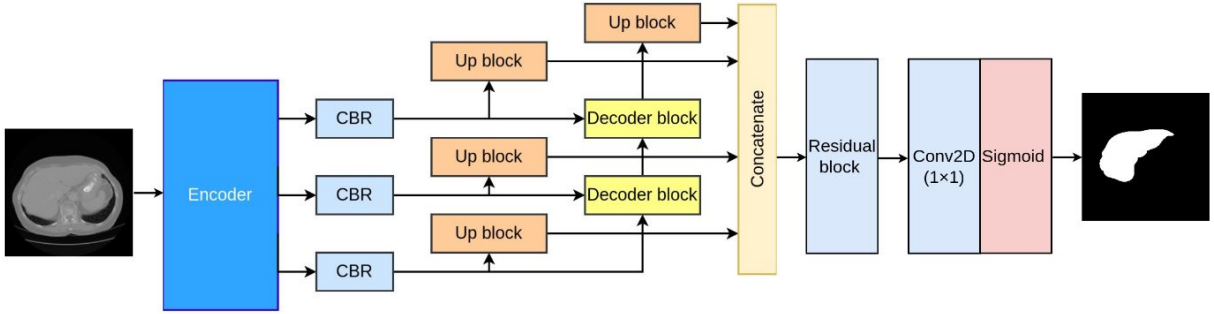
Features	NanoNet Advantages	ColonSegNet Advantages	DDANet Advantages
Architecture	Utilizes pre-trained MobileNetV2 and modified residual block.	Utilizes residual blocks.	Utilizes two decoder network.
Performance	Shows undersegmentation in some scenarios.	Demonstrates high performance	Demonstrates superior performance.
Usability	Offers real-time processing speed	Offers fastest processing speed	Requires high computation resources.

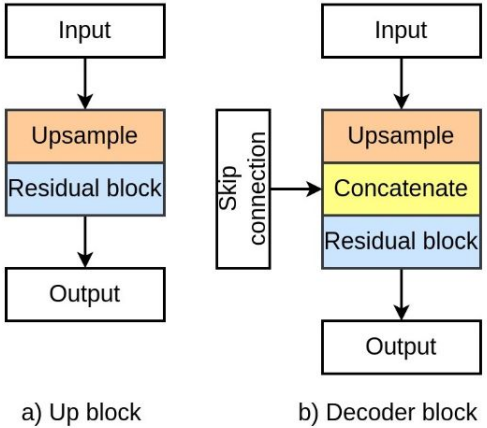
Post-Processing Technique

- Conditional Random fields as post-processing.
- Generative random fields as post processing.
- Test time augmentation



Liver Segmentation



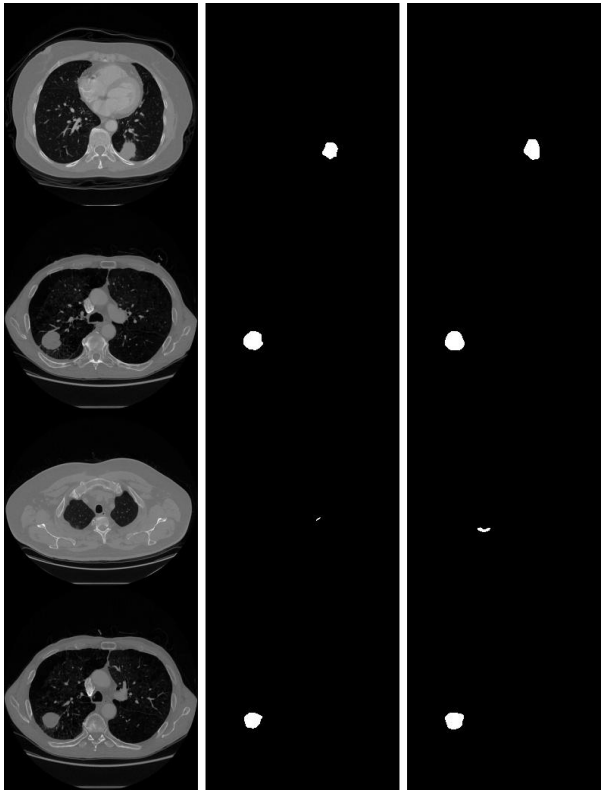


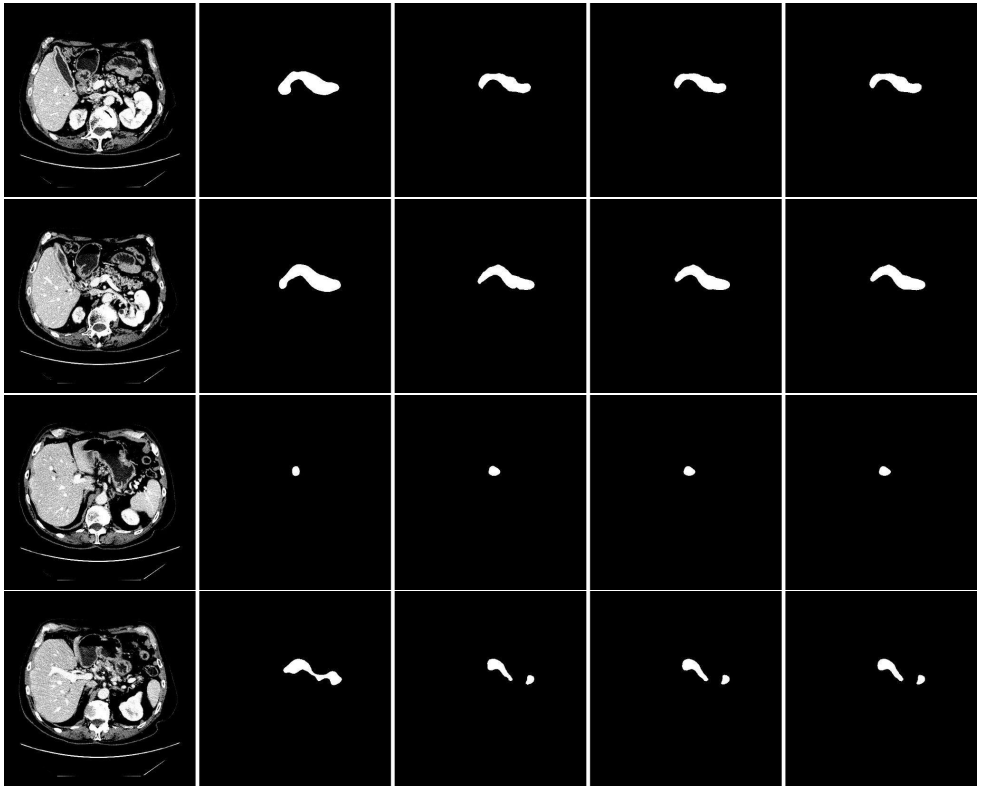
a) Up block b) Decoder block

Quantitative Results

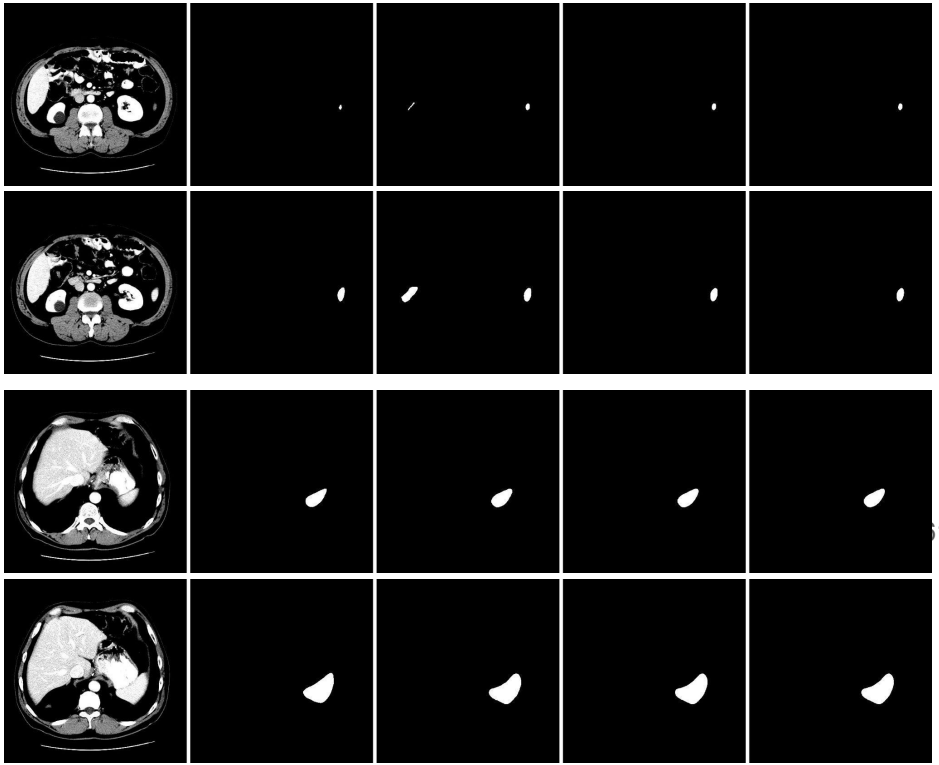
Method	Publication	Dice (%)	mIoU (%)	Recall (%)	Precision (%)	F2 (%)	HD
U-Net [15]	MICCAI 2015	82.06	73.40	77.82	91.10	78.98	3.79
ResUNet++ [16]	IEEE ISM 2019	77.03	68.19	79.90	83.57	76.24	3.96
DoubleU-Net [17]	IEEE CBMS 2020	86.24	77.89	80.68	95.00	81.28	3.69
ColonSegNet [17]	IEEE Access 2021	80.87	71.71	80.07	87.50	79.22	3.84
NanoNet [18]	IEEE CBMS 2021	75.05	66.53	73.33	83.25	73.23	4.01
UNeXt [19]	MICCAI 2022	81.31	72.43	80.73	87.40	79.93	3.74
TransNetR [20]	MIDL 2023	86.11	77.95	80.16	**96.34**	82.30	3.54
TransResUNet [21]	IEEE CMBS 2023	86.38	78.23	**80.85**	95.77	82.83	3.54
PVTFormer	Proposed	**86.78**	**78.46**	80.70	96.11	**82.86**	**3.50**

Lung Tumor and Pancreas Segmentation

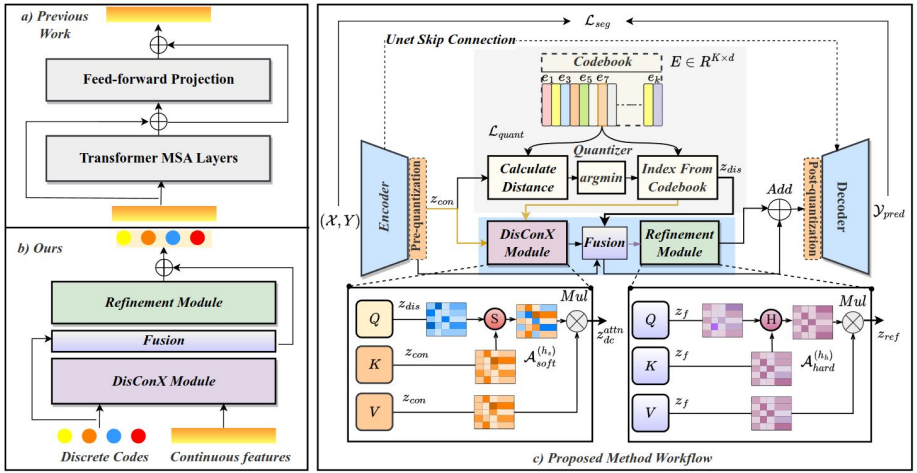


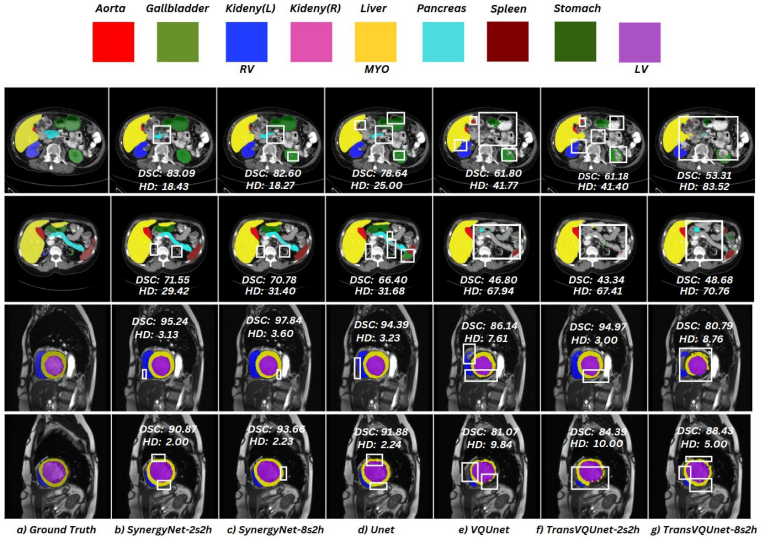


Spleen Segmentation

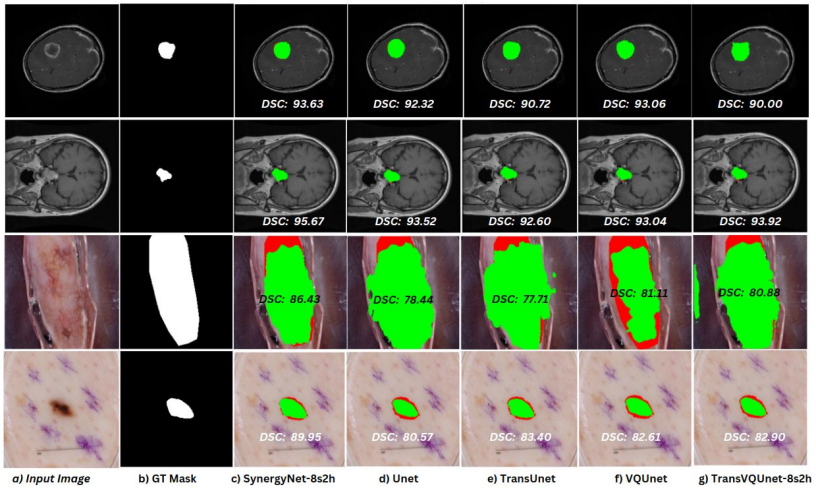


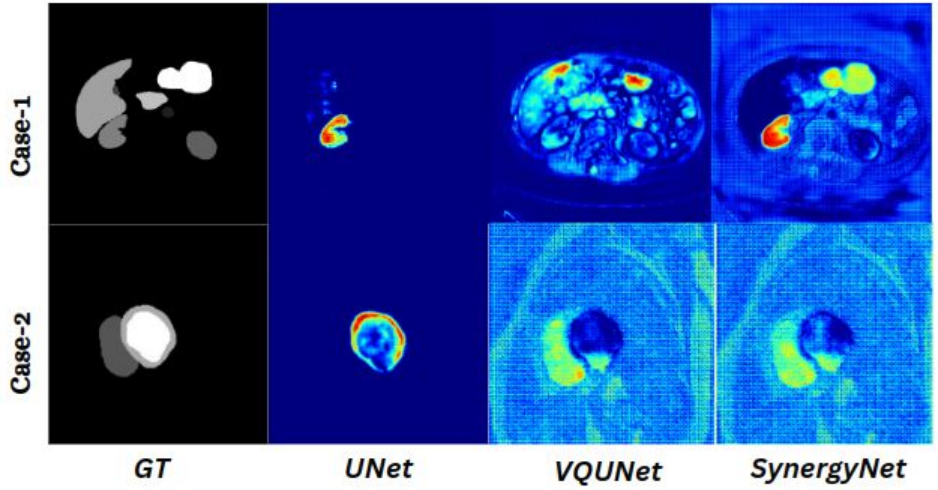
Multi-organ Segmentation (SynergyNet)



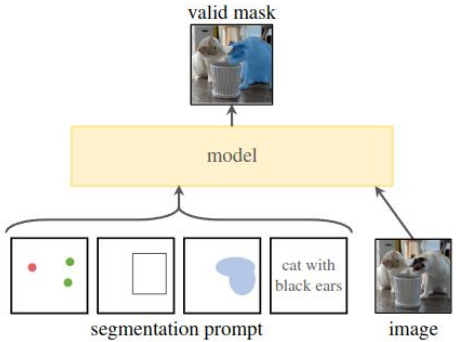


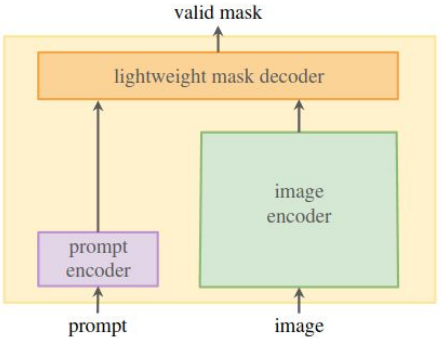
Multi-organ Segmentation

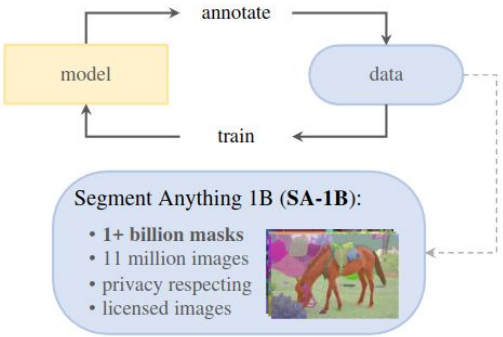


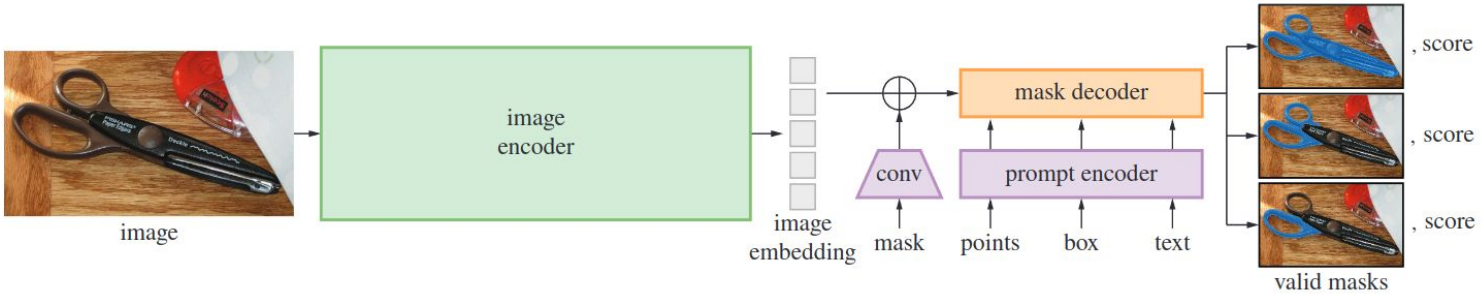


Segment Anything Model

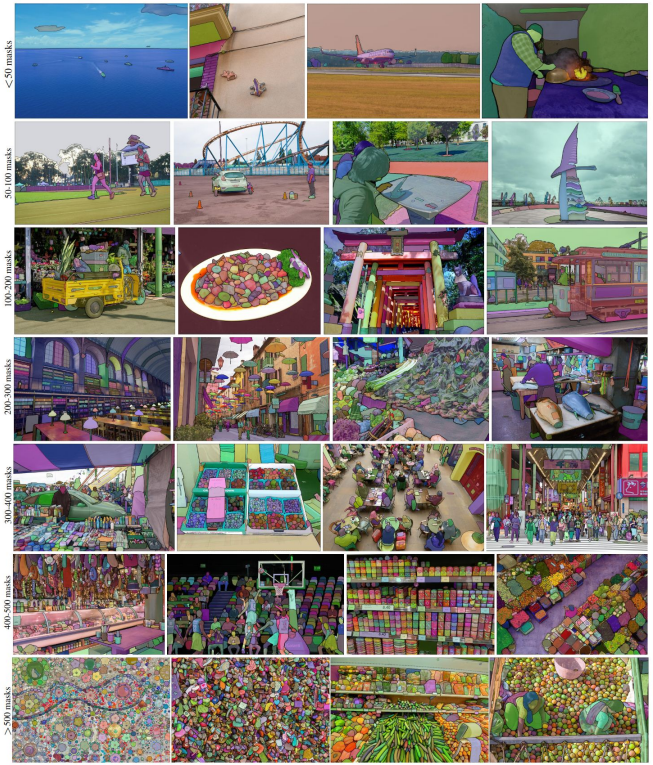


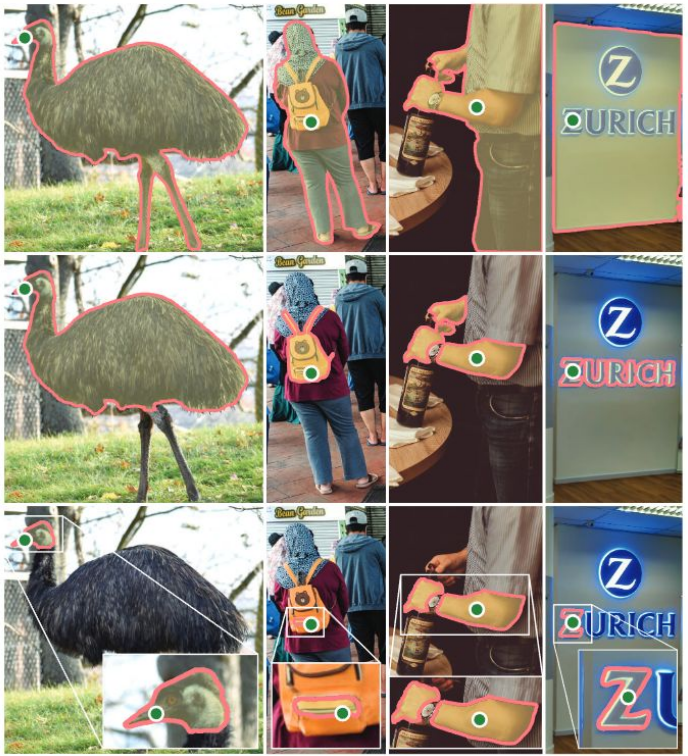




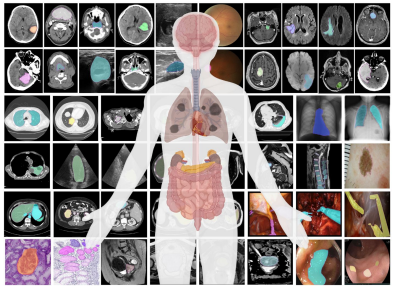


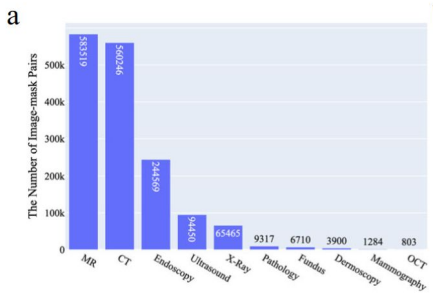
Results of the SAM model

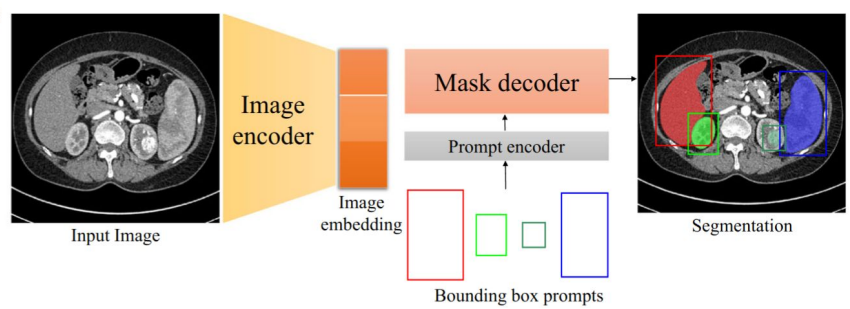




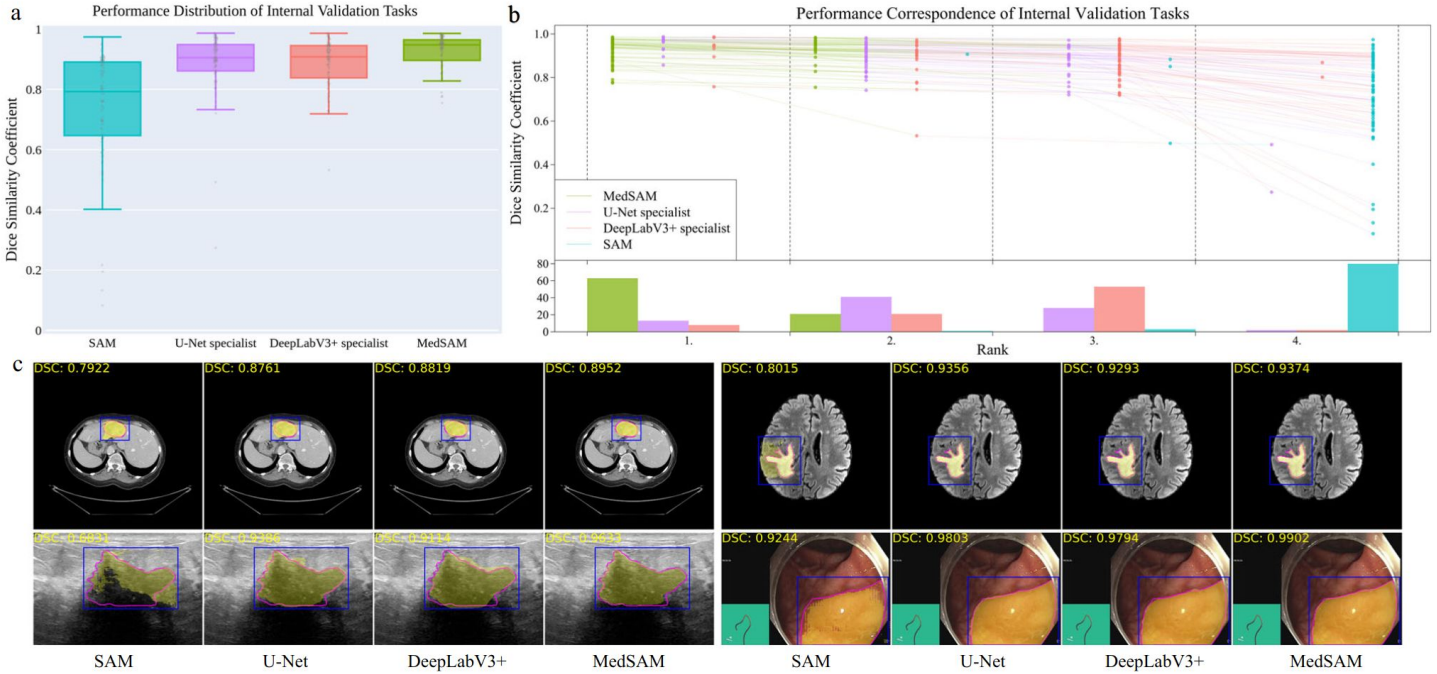
MedicalSAM



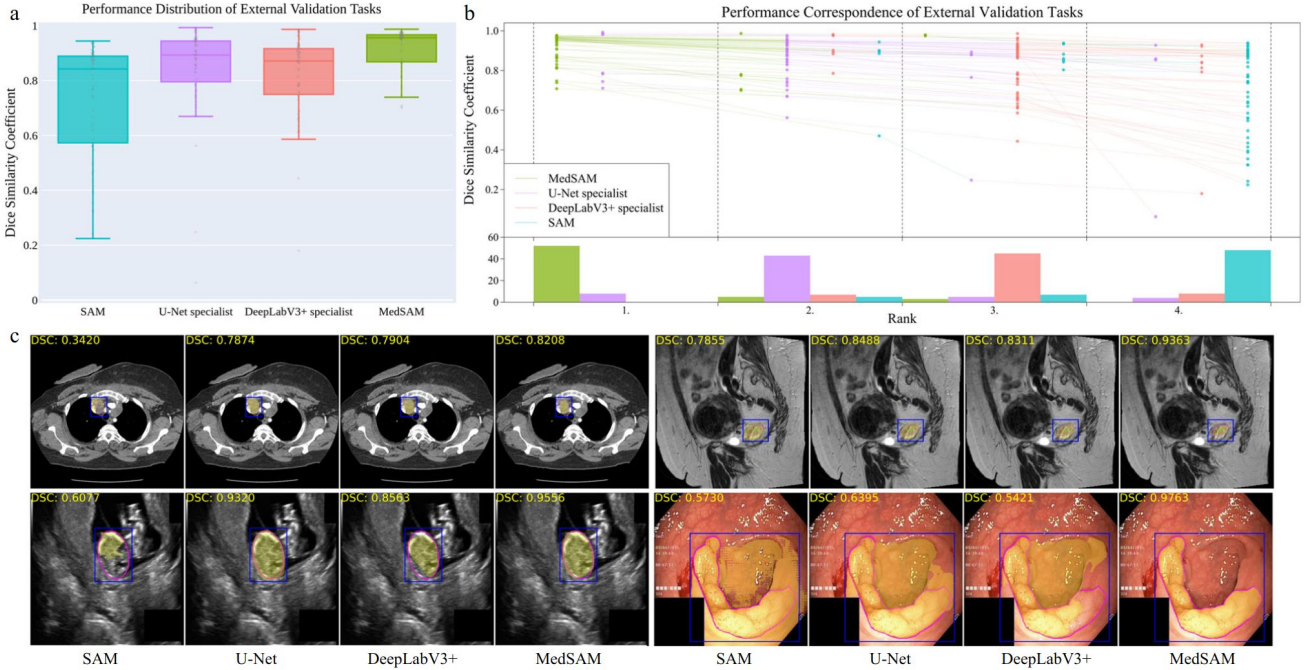

a
The Number of Image-mask Pairs
0
100k
200k
300k
400k
500k
525539
502046
244569
96496
65465
9317
6710
3900
1284
803
MR
CT
Endoscopy
Ultrasound
X-Ray
Pathology
Fundus
Dermoscopy
Mammography
OCT


b
Input Image
Image encoder
Image embedding
Mask decoder
Prompt encoder
Bounding box prompts
Segmentation

Qualitative Results



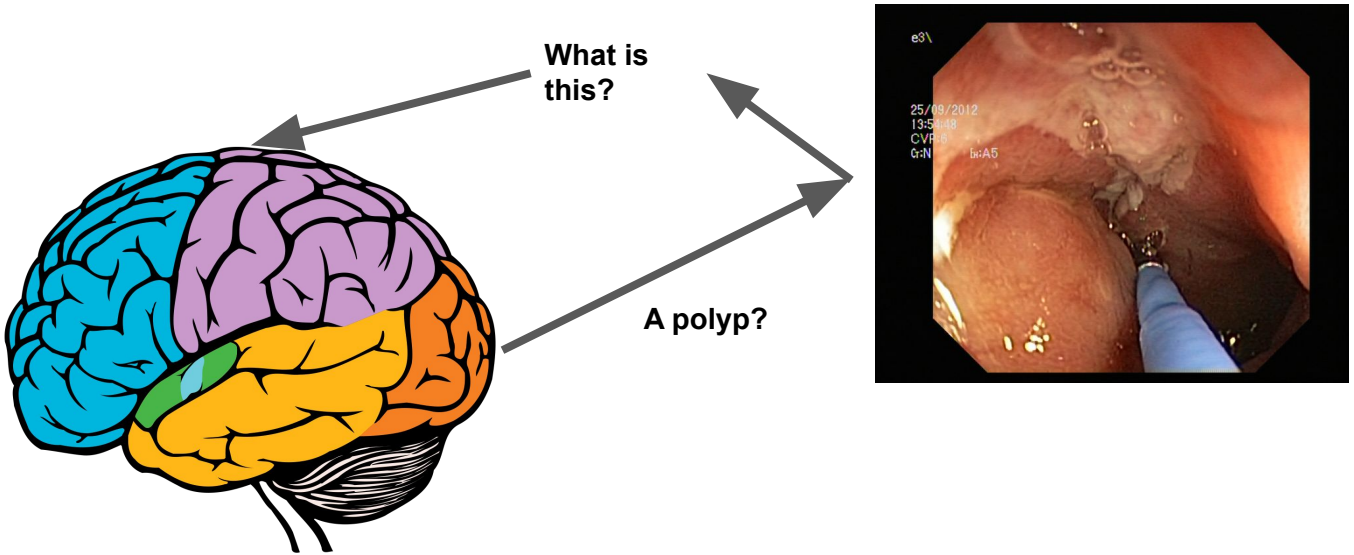
Qualitative Results



Evaluation metrics for Medical image segmentation

Disease	Metrics
Prostate Cancer	Sensitivity, Specificity, PPV, NPV, DSC
Liver Cancer	Sensitivity, Specificity, Volume overlap error, surface distance measure (ASD, RMSD)
Breast Cancer	Sensitivity, Specificity, DSC, Volume Overlap Error
Lung Cancer	Sensitivity, False Positive Per Scan, Volume Precision, DSC
Colorectal Cancer	Sensitivity, Specificity, HD, DSC
Stomach Cancer	Sensitivity, Specificity, Volume Overlap Error, MSD
Cervix Uteri Cancer	Sensitivity, Specificity, DSC, Volume Overlap Error
Pancreatic Cancer	Sensitivity, Specificity, Surface Distance Measures (e.g., ASD), Volume Overlap Error
Oesophageal Cancer	Sensitivity, Specificity, Dice Similarity Coefficient (DSC), Length Accuracy

Interpretability and explainability of segmentation methods



Why?

Are you sure?

Why do we need interpretability/explainability methods?

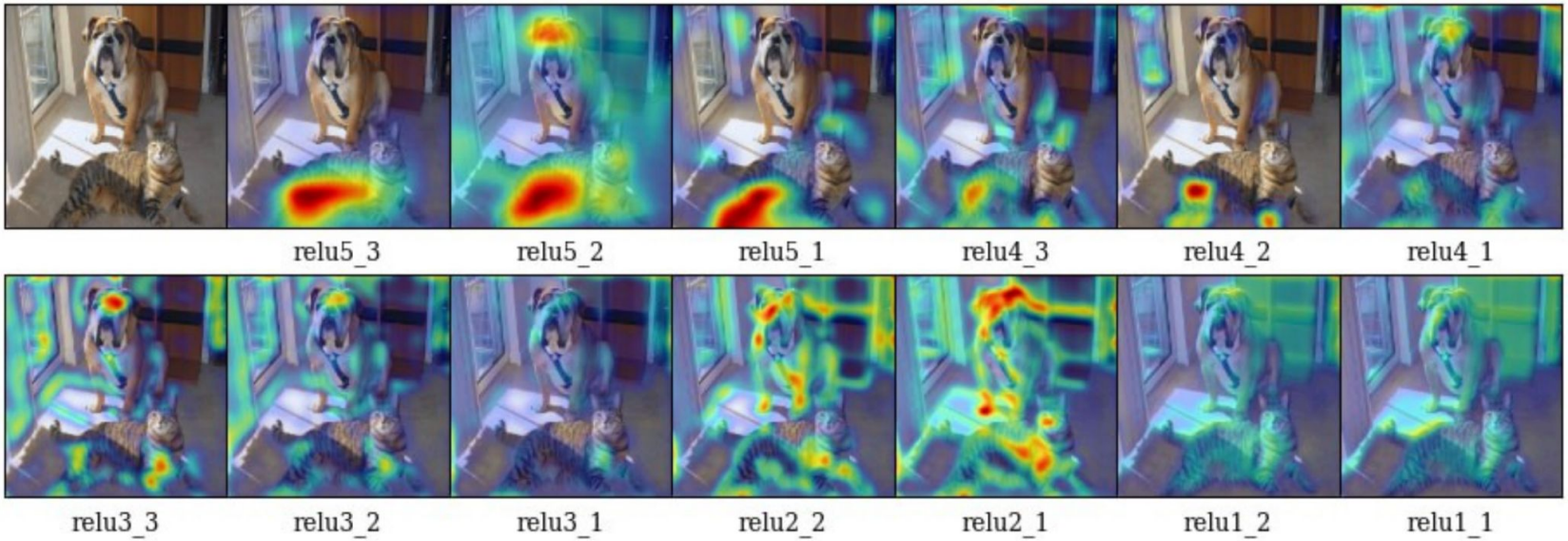
- Who is accountable for wrong decisions? (Ex: in the medical domain)
- How do end-users (doctors) trust decisions?
- How to find most important features to improve the performance and generalizability of DL models?
- How to detect erroneous reasoning?
- How to select the best model among competing models?

Gradcam (Gradient-weighted Class Activation Mapping)

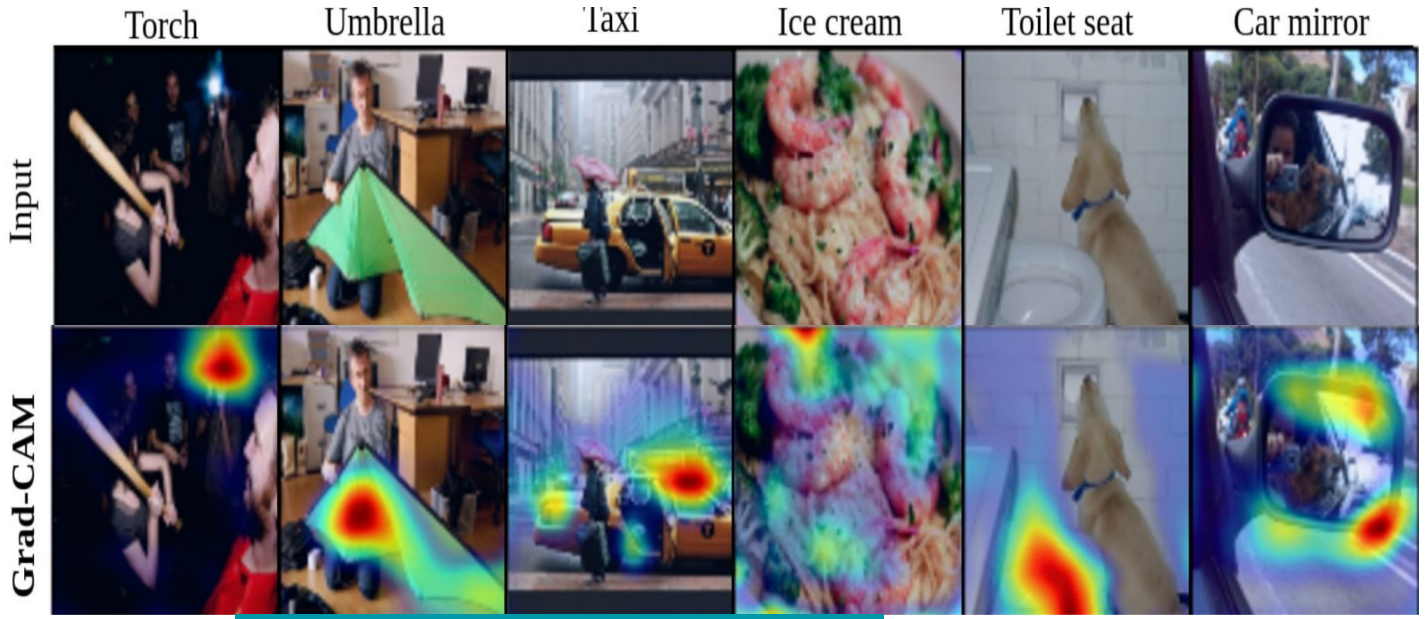
- Deep learning models, especially CNNs, are often considered "black boxes" because it’s hard to understand how they make decisions.
- Grad-CAM provides explainability by answering:
- "**Which parts of the input image influenced the model's decision**?
- For example:
  - In a medical imaging task, it can highlight areas where the model identified a tumor.
  - In an object recognition task, it can show the regions the model focused on to classify a dog or cat.

Grad-CAM : examples 1/2

Grad-CAM with different convolution layers (VGG16)

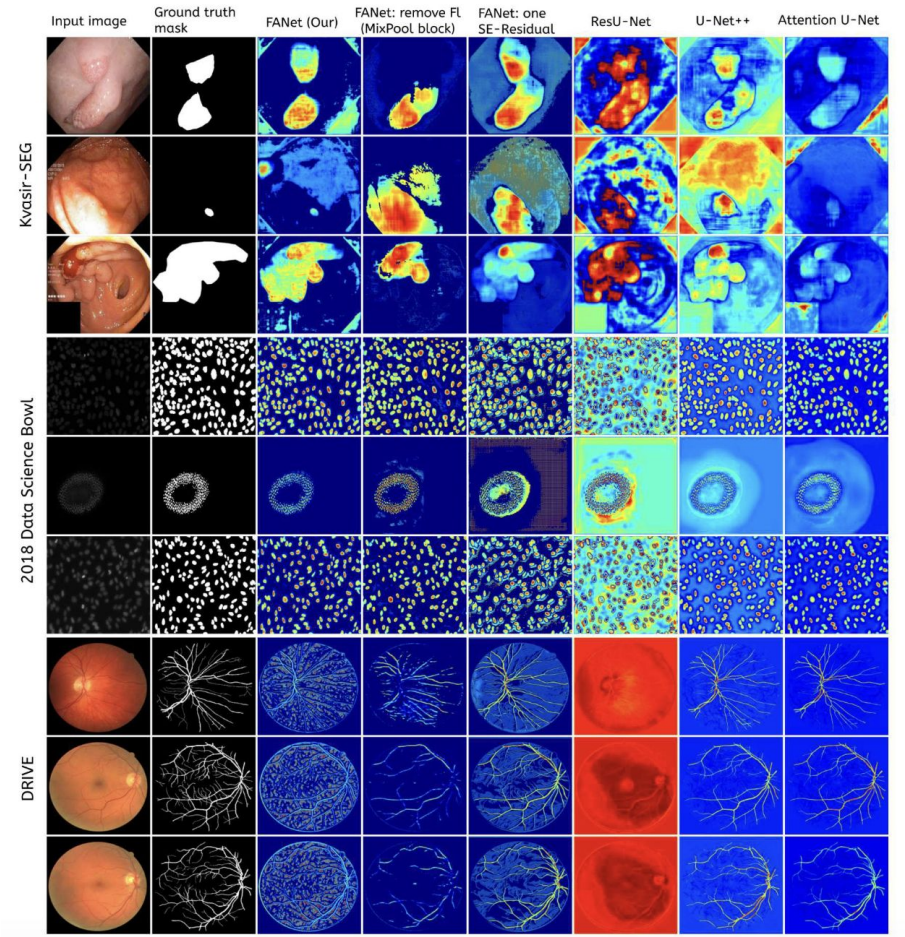


Grad-CAM : examples 2/2



Can we use this method with medical data?

Gradcam for Comparison



Conclusion

Revolutionary Impact of CNNs:

- CNNs, especially architectures like U-Net and ResUNet++, have transformed image segmentation.

Real-World Relevance:

- Applications in medical imaging, autonomous vehicles, and agriculture underline the importance of segmentation.

Emerging Challenges:

- Addressing challenges like class imbalance, computational cost, and ethical concerns is crucial for advancing this field.

Future Directions

- **Collaboration:** Encouraging interdisciplinary work between domain experts and computer scientists.
- **Innovation:** Development of lightweight, explainable, and robust architectures.
- **Ethics:** Tackling issues like fairness, privacy, and accountability in AI-based segmentation.

"The journey of every pixel from noise to meaning is the story of segmentation, and CNNs are our guide."