Bios 740- Chapter 9. Image Segmentation

Acknowledgement: Many thanks to Mr. Mingchen Hu for preparing some of these slides. I also drew on material from the lecture presentations of StanfordCS231n as well as content generated by ChatGPT.

Content

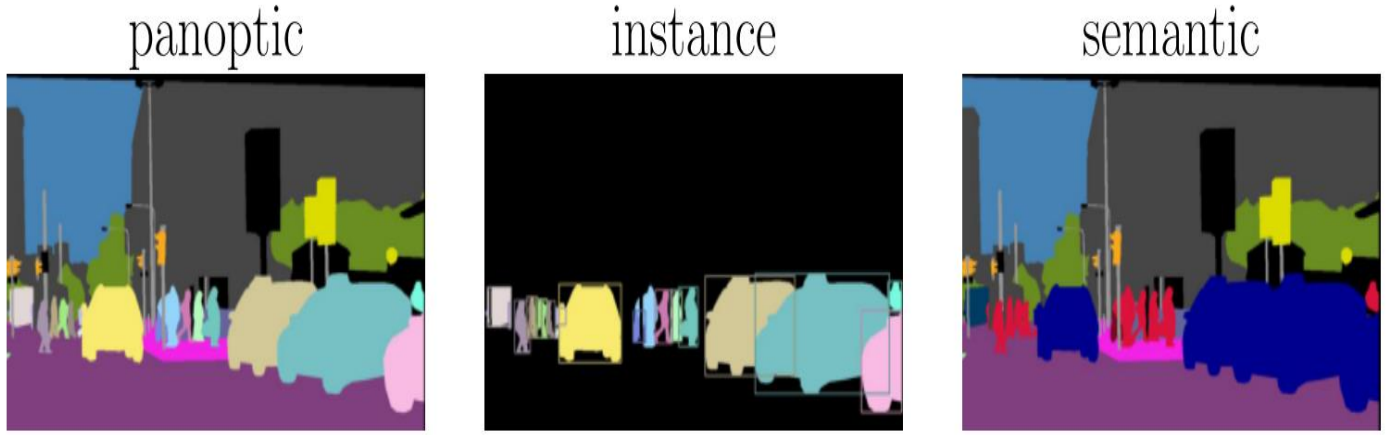
1. Introduction to Image Segmentation
2. Introduction to U-Net
3. U-Net Extensions
4. Foundation Models for Image Segmentation
5. Theoretical Properties

Content

Image Segmentation

Image segmentation, defined as the partition of the entire image into a set of regions, aims to make anatomical or pathological structures changes clearer in images. Image segmentation tasks can be classified into three categories: semantic segmentation, instance segmentation and panoptic segmentation.

- **Semantic segmentation** is a pixel-level classification that assigns corresponding categories to all the pixels in an image.
- **Instance segmentation** needs to identify different objects within the same category.
- **Panoptic segmentation** presents a unified image segmentation approach where each pixel in a scene is assigned a semantic label and a unique instance identifier.



Task Type	Goal	Pixel-wise Label	Object Instances
Semantic Segmentation	Classify all pixels	✅	❌
Instance Segmentation	Segment individual objects	✅	✅
Panoptic Segmentation	Combine both tasks	✅	✅

Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., & Girdhar, R. (2022). *Masked-attention Mask Transformer for Universal Image Segmentation* (No. arXiv:2112.01527). arXiv. https://doi.org/10.48550/arXiv.2112.01527

Mathematical Formulation

Image segmentation aims to assign semantic labels to image regions.

- Formally, this is a mapping problem. $f : \mathcal{X} \to \mathcal{Y}$
- f is typically implemented as a deep neural network.

Input Space $\mathcal{X} = \mathcal{I} \times \mathcal{P}$

- $\mathcal{I}$: Image domain (e.g., RGB images, CT scans)
- $\mathcal{P}$: Prompt space (used in interactive/prompt-based segmentation)
- Each input $x \in \mathcal{X}$ is a pair (i, p), where $i \in \mathcal{I}, p \in \mathcal{P}$

Prompts are used in special settings:

- **Interactive segmentation:** user clicks or bounding boxes
- **Vision-language models:** textual prompts
- Prompts guide the model to focus on relevant regions or semantics

Output Space $\mathcal{Y} = \mathcal{M} \times \mathcal{C}$

- $\mathcal{M}$: Segmentation mask (pixel-wise label matrix)
- $\mathcal{C}$: Set of semantic categories
- Output $y = (m, c)$ links each mask to its semantic class

- f : X → Y models the relationship between input and segmentation output
- Typically implemented using deep models like U-Net, DeepLab, or Vision Transformers

Medical Image Segmentation

In biomedical field, image segmentation often plays a key role in computer-aided diagnosis and smart medicine. There are three main challenges in **medical imaging segmentation**:

- **Limited image samples per specific disease** – The scarcity of annotated medical images for certain diseases restricts the performance of segmentation methods.
- **Complex lesion characteristics** – Similar intensity, variable shapes, and dynamic positions of lesions make accurate segmentation difficult.
- **Image acquisition artifacts** – Noise, spatial aliasing, and sampling artifacts lead to unclear or disconnected boundaries in structures of interest.

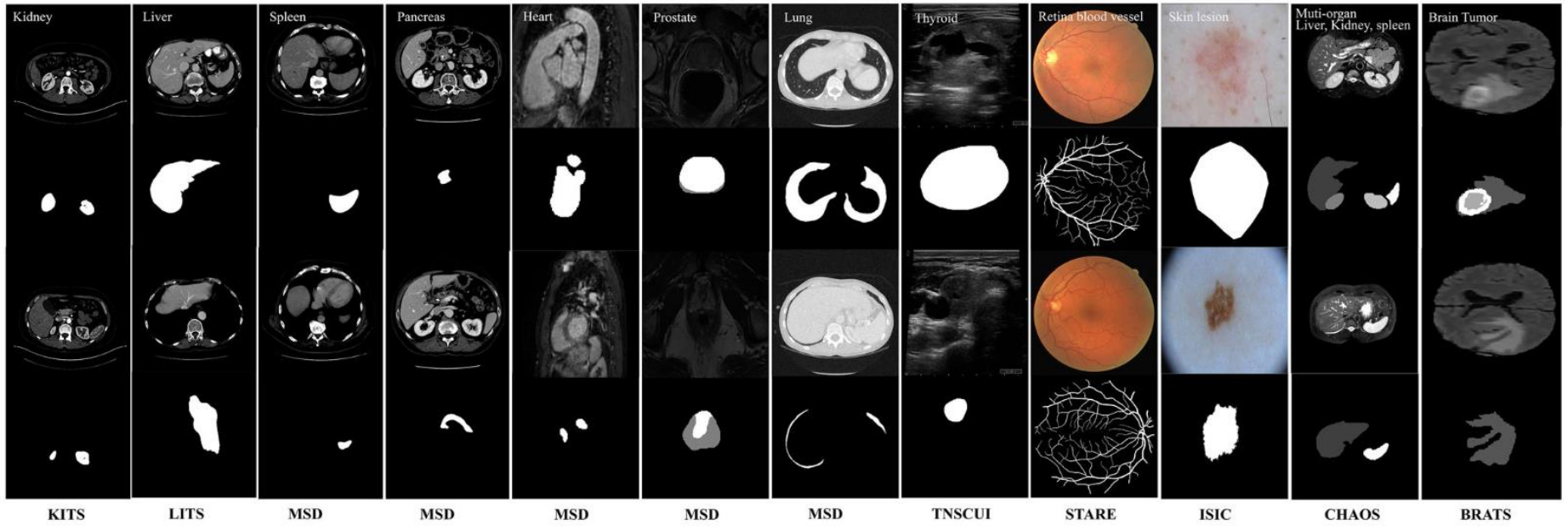
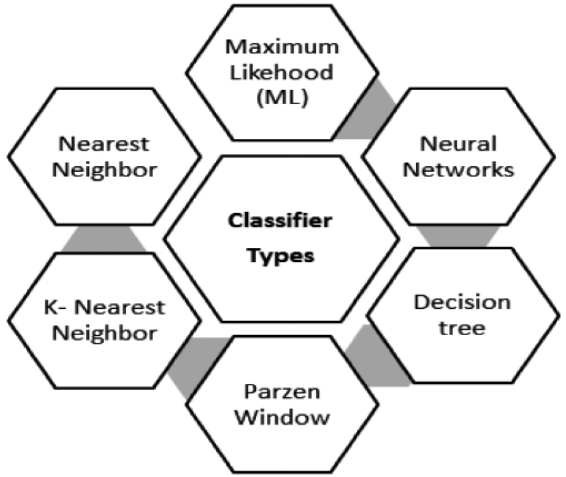


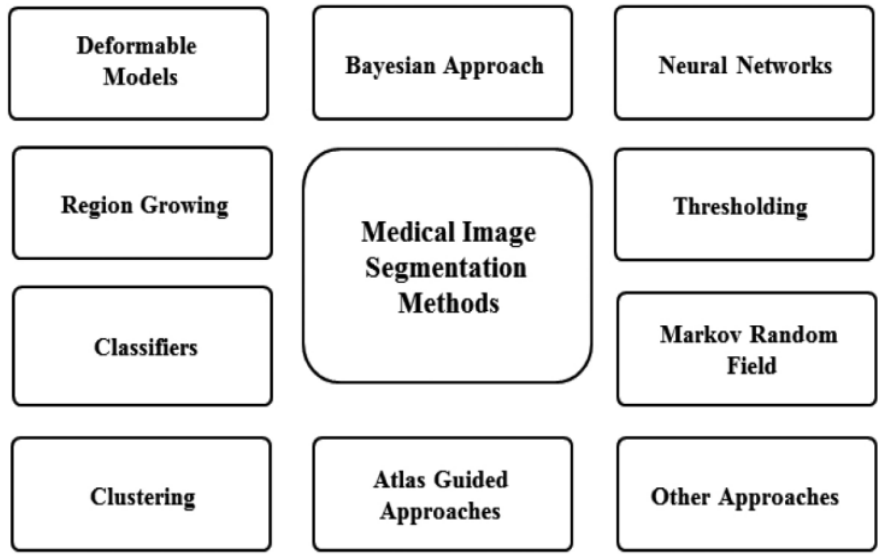
Image Segmentation before Deep Learning

❖ **Early segmentation methods were model-driven.**

❖ **Common techniques included:**

- Thresholding
- Histogram mode seeking
- Region growing and merging
- Spatial clustering
- Energy diffusion
- Super-pixel representation
- Conditional and Markov Random Fields (CRFs & MRFs)





❖ **Relied on prior knowledge and handcrafted features.**

Advantages:

- Intuitive and interpretable
- Computationally efficient

Limitations:

- Poor generalization to complex structures
- Sensitive to noise and intensity variation
- Struggled with diverse anatomical variability

Masood, S., Sharif, M., Masood, A., Yasmin, M., & Raza, M. (2015). A Survey on Medical Image Segmentation. *Current Medical Imaging Reviews*, *11*(1), 3–14. https://doi.org/10.2174/1573405611011504231034 41

Rise of Deep Learning in Segmentation

In recent years, deep learning has become the dominant paradigm in medical image segmentation.

Core architecture types:

- **Convolutional Neural Networks (CNNs)** – foundational for early breakthroughs
- **U-Net and its variants** – encoder-decoder architecture specifically designed for biomedical segmentation
- **Vision Transformers (ViT, Swin)** – exploit long-range dependencies for better global context

Benefits of Deep Learning:

- Learns hierarchical and task-specific features automatically
- Capable of capturing complex anatomical structures
- Handles multimodal inputs (e.g., MRI + CT) and 3D volumetric data
- Integrates with attention mechanisms and prior knowledge via hybrid models

Deep learning methods typically outperform traditional approaches by 10–20 percentage points in Dice coefficient, especially on complex regions and harder substructures.

Dataset	Target Region	Traditional	Deep Learning
BraTS	Whole Tumor	65–75%	90–95%
	Tumor Core	60–70%	85–90%
	Enhancing Tumor	55–65%	75–85%
LiTS	Liver	70–80%	96–98%
	Tumor	50–60%	65–75%
PROMISE12	Prostate Gland	75–80%	88–92%

Table: Dice Similarity Coefficient (%) comparison on benchmark datasets

Content

1. Introduction to Image Segmentation

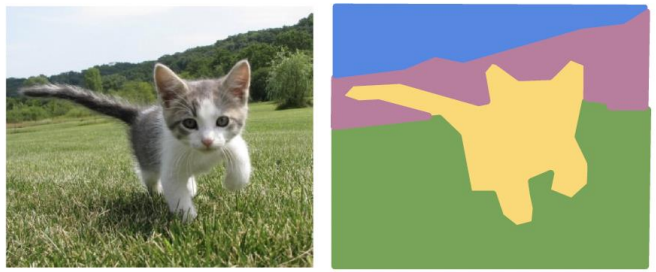
2. Introduction to U-Net

3. U-Net Extensions

4. Foundational Models for Image Segmentation

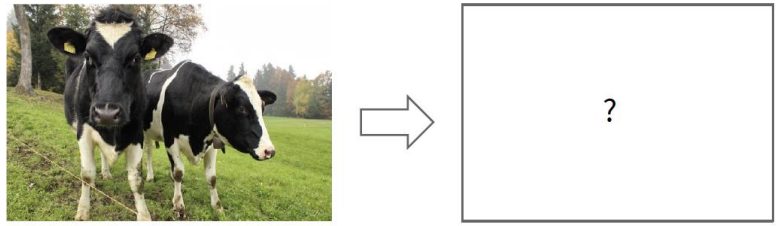
5. Theoretical Properties

Semantic Segmentation: The Problem

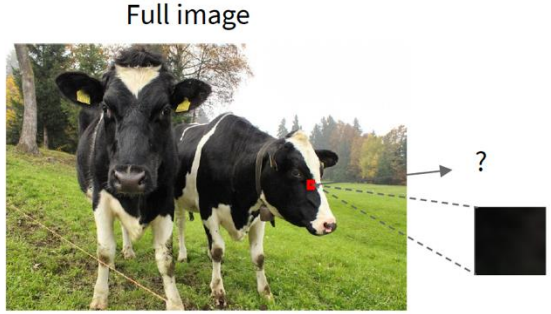


GRASS, CAT, TREE,
SKY, ...

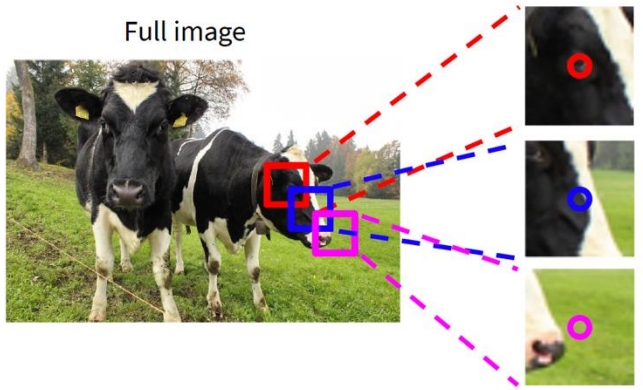
Paired training data: for each training image, each pixel is labeled with a semantic category.



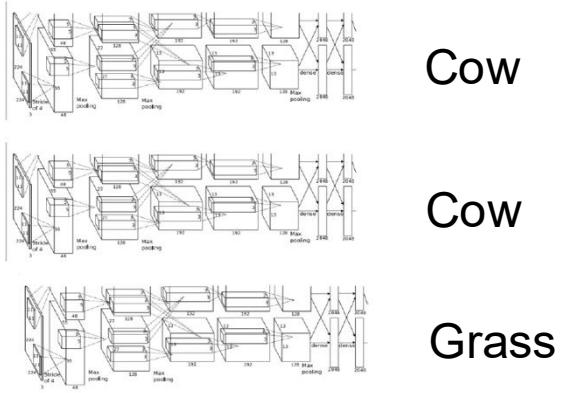
At test time, classify each pixel of a new image.



Impossible to classify without context
Q: how do we include context?



Classify center pixel with CNN



Q: how do we model this?

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

U-Net: Motivation

In CNNs, different layers learn different feature levels:

- **Lower layers:** Learn low-level, fine-grained details (e.g., edges, textures)
- **Higher layers:** Capture high-level, coarse-grained semantic features (e.g., shape, structure)

•This hierarchy is ideal for classification tasks but introduces limitations for pixel-level tasks like segmentation

Challenges in Medical Image Segmentation

•Medical images often suffer from:

- Noise
- Low contrast
- Blurred or unclear boundaries

•Relying only on low-level features results in poor object recognition

•Relying only on high-level semantic features leads to inaccurate boundary detection

Need for Multi-Level Feature Integration

•Effective segmentation requires a combination of:

- High-level semantic understanding (context)
- Low-level spatial precision (details)

•General CNNs lack explicit mechanisms to combine both effectively

Encoder-Decoder Architectures

•Designed to combine high-level and low-level features

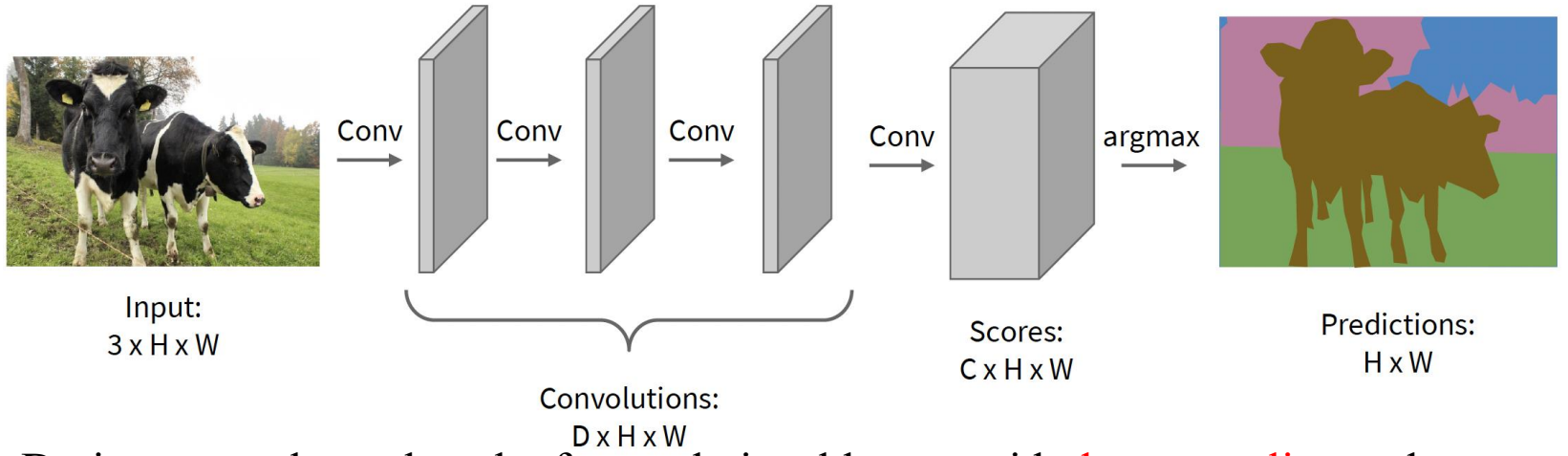
•Consist of:

- **Encoder:** Downsamples and extracts abstract features
- **Decoder:** Upsamples to recover spatial resolution and integrates detail

•Enables pixel-level prediction with semantic awareness

Semantic Segmentation Idea

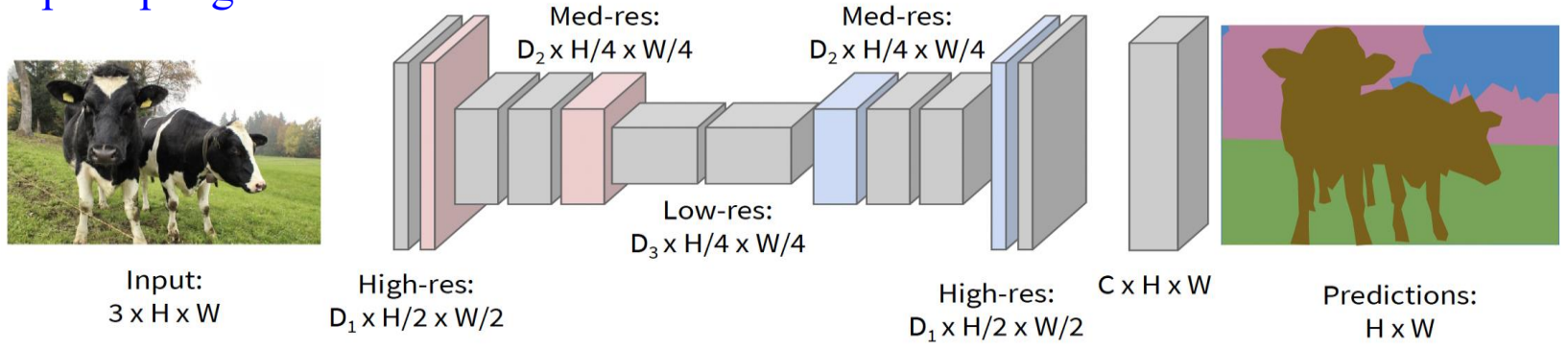
Design a network with only convolutional layers without downsampling operators to make predictions for pixels all at once!



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

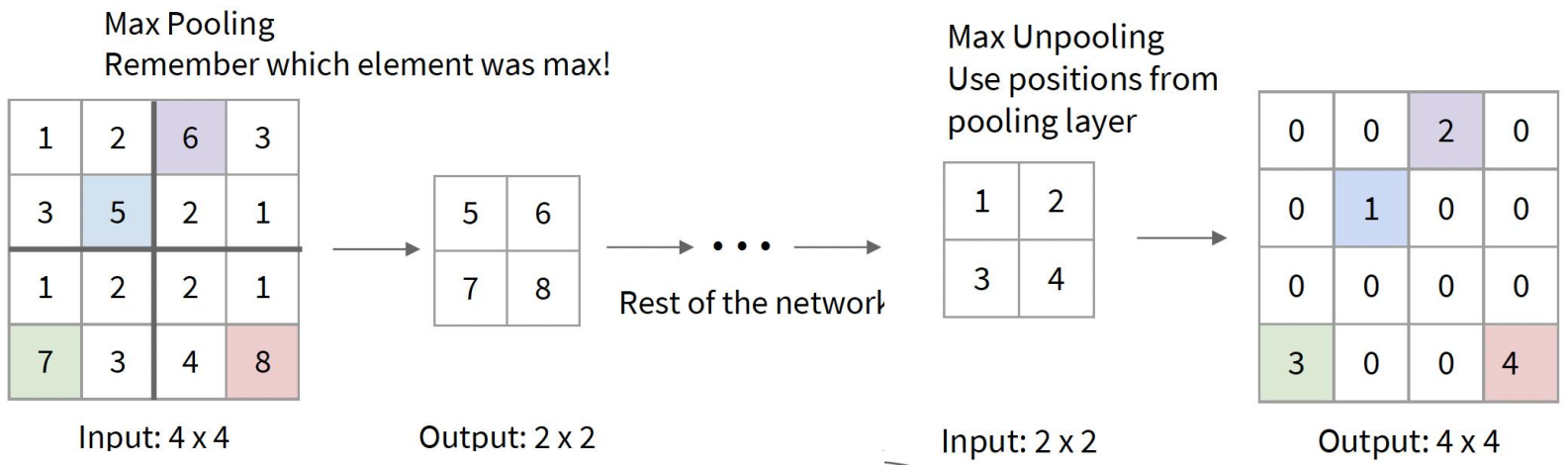
Design network as a bunch of convolutional layers, with downsampling and upsampling inside the network!

Downsampling: Pooling, strided convolution

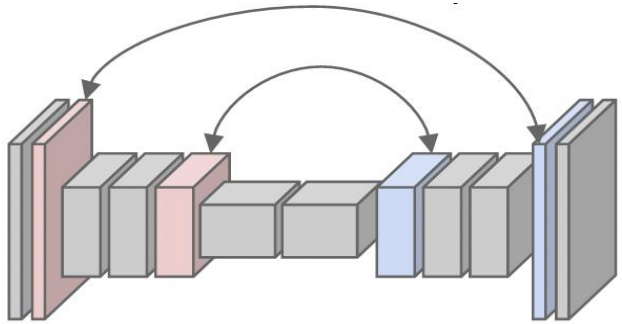


Upsampling: Unpooling or strided transposed convolution

Downsampling and Upsampling



Corresponding pairs of downsampling and upsampling layers



Common Downsampling types:

- **Max pooling:** Takes the maximum value in each window
- **Average pooling:** Computes the average value
- **Stochastic pooling:** Randomly selects an activation based on a probability distribution
- **LP-pooling:** Generalized pooling that uses the p-norm over each region
- **Global pooling:** Applies pooling over the entire feature map to reduce to a single value per channel

•**Purpose:** (i) Reduce computation; (ii) Increase receptive field; (iii) Achieve spatial invariance; (iv) Introduce regularization

Common unpooling strategies:

- **Max-unpooling with indices**:
- **Fixed-position unpooling**: inserts values at top-left corner of window
- **Interpolation-based unpooling**: uses nearest-neighbor or bilinear interpolation to expand feature maps
- **Learnable unpooling**: introduces parameters to learn where and how to upsample

Often followed by convolutional layers to refine outputs

Learnable upsampling

Learnable upsampling replaces heuristic upsampling with trainable layers

- **Common types:**
  - **Transposed convolution (deconvolution):** applies learnable filters to increase spatial resolution
  - **Sub-pixel convolution:** reshapes feature maps using depth-to-space operations
  - **Resize-convolution:** resizes feature map first, then applies standard convolution
- Learnable upsampling is adaptive to data and helps with fine-grained localization

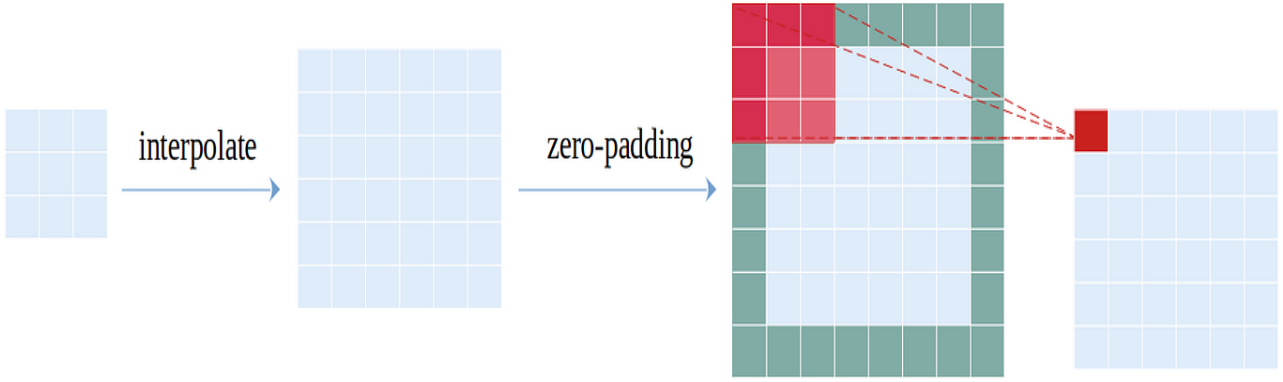
- Resize step: interpolate input $x \in \mathbb{R}^{H \times W}$ to size $rH \times rW$

$$x'_{i,j} = \sum_{(m,n)} x_{m,n} \cdot k(i-m, j-n)$$

where k is the interpolation kernel (e.g., bilinear, bicubic)

- Convolve: apply learnable filters

$$y = \text{Conv}(x')$$



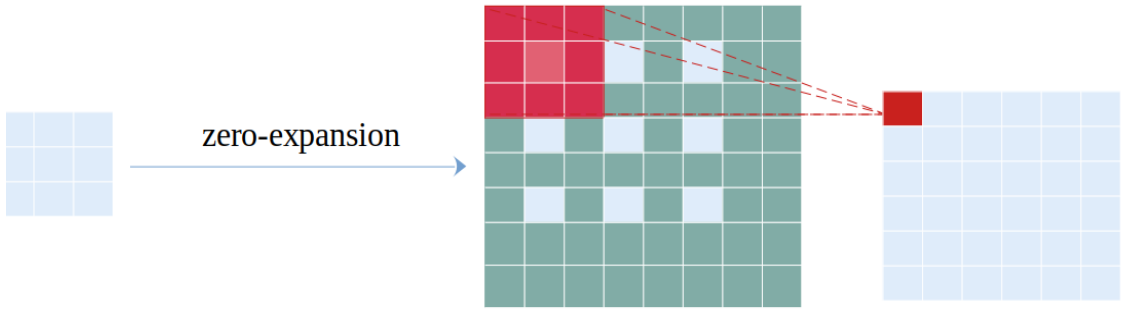
The resize convolution. Dark green cells are zero-valued, red lines indicate a traditional convolution operation.

https://medium.com/@paren8esis/introduction-to-super-resolution-with-deep-learning-pt-2-ced99297a483

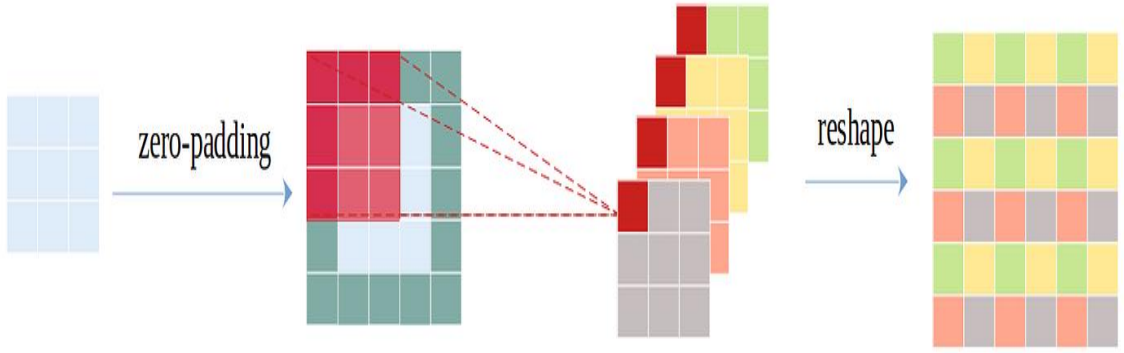
Learnable upsampling

Let $x \in \mathbb{R}^{H_{\text{in}} \times W_{\text{in}}}$, kernel $w \in \mathbb{R}^{k \times k}$, stride s:

$$y_{i,j} = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} x_{\left\lfloor \frac{i-m}{s} \right\rfloor, \left\lfloor \frac{j-n}{s} \right\rfloor} \cdot w_{m,n}$$



The transposed convolution. Dark green cells are zero-valued, red lines indicate a traditional convolution operation.



The sub-pixel convolution. Dark green cells are zero-valued, red lines indicate a traditional convolution operation.

- Input: $x \in \mathbb{R}^{C \cdot r^2 \times H \times W}$
- Output: $y \in \mathbb{R}^{C \times rH \times rW}$
- Operation:

$$y_{r \cdot i+m, r \cdot j+n, c} = x_{i,j,c \cdot r^2 + r \cdot m + n}$$

for $m, n \in \{0, \ldots, r-1\}$, rearranging depth into spatial resolution

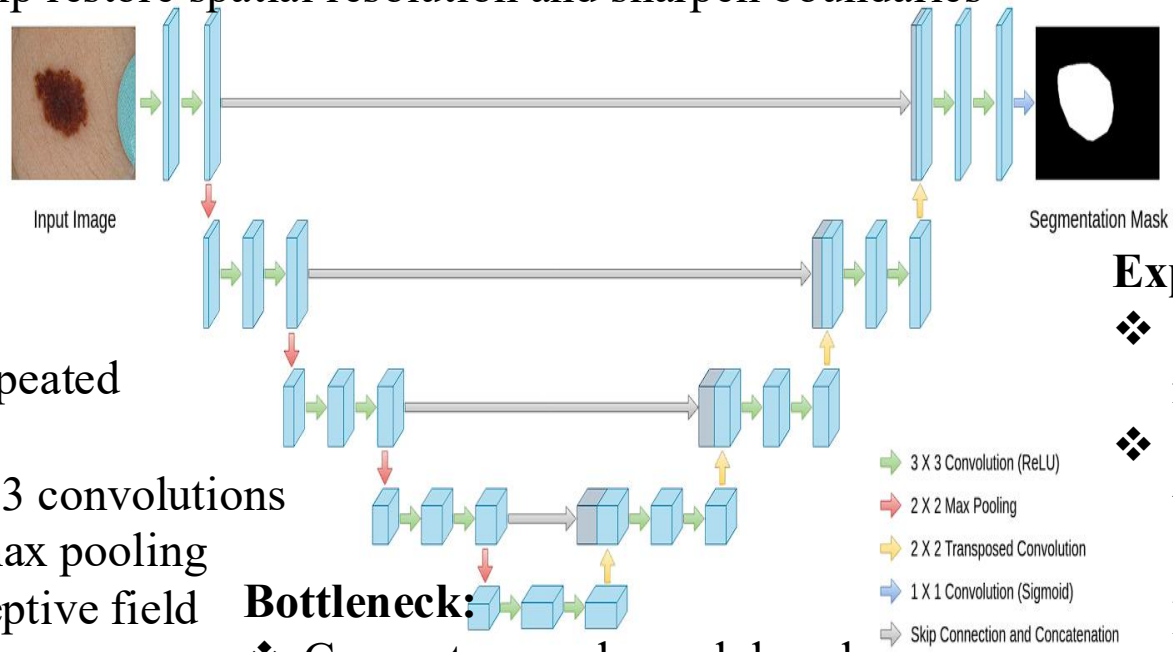
https://medium.com/@paren8esis/introduction-to-super-resolution-with-deep-learning-pt-2-ced99297a483

U-Net: Vanilla Version

❖ **U-Net is a neat end-to-end neural network with a characteristic "U" shape**

Skip Connections:

- ➢ Link encoder and decoder layers at the same depth level
- ➢ Concatenate encoder feature maps with decoder inputs to combine detailed and contextual information
- ➢ Help restore spatial resolution and sharpen boundaries



Final Output:

- ❖ A 1×1 convolution maps the final feature map to the number of target classes
- ❖ Produces a pixel-level classification map (e.g., segmentation mask)

Contracting Path (Encoder):

- ❖ Captures context through repeated downsampling blocks
- ❖ Each block includes two 3×3 convolutions + ReLU, followed by 2×2 max pooling
- ❖ Gradually increases the receptive field without heavy computation

Bottleneck:

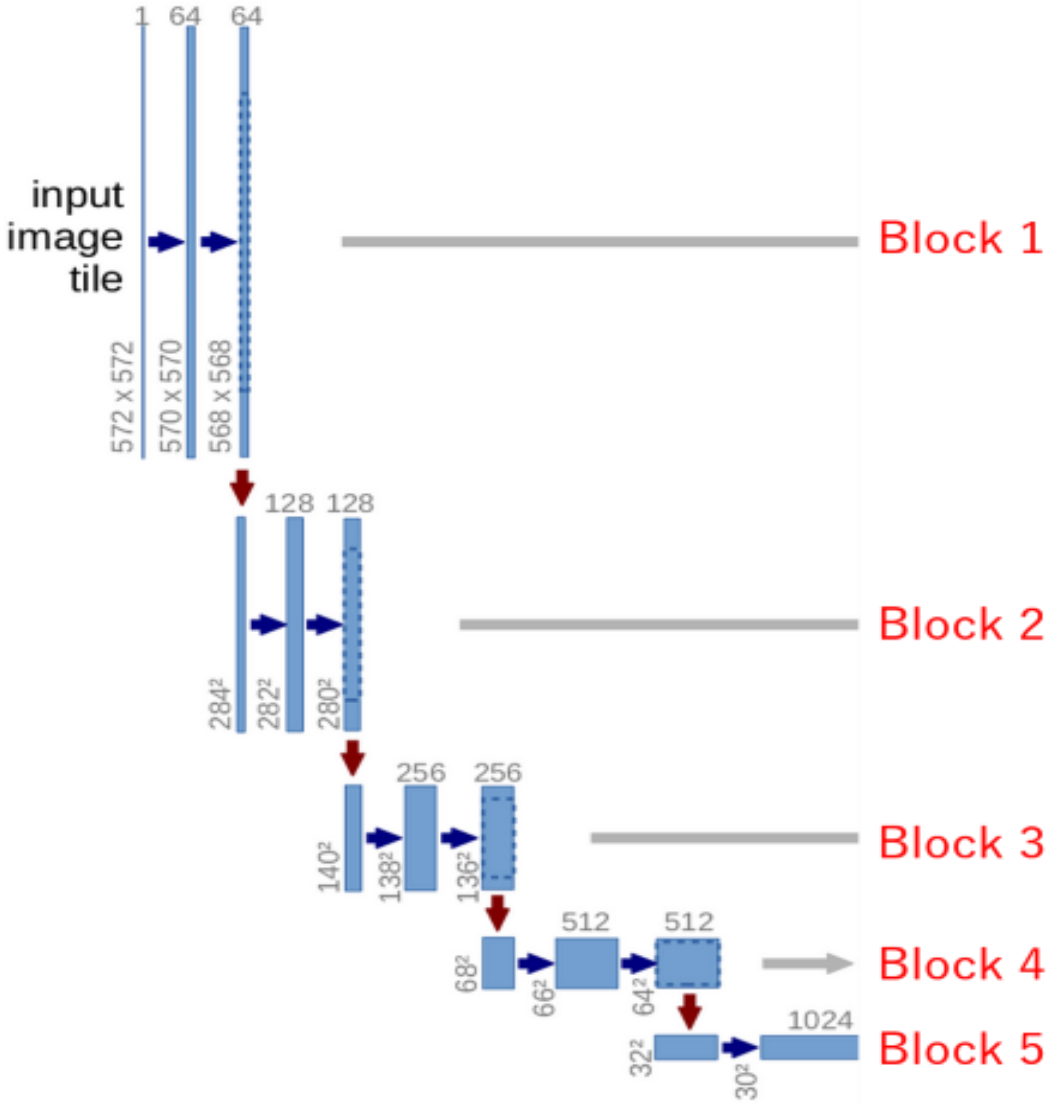
- ❖ Connects encoder and decoder
- ❖ Two 3×3 convolutions + ReLU
- ❖ Reduces spatial resolution and increases depth for high-level abstraction

Expanding Path (Decoder):

- ❖ Upsamples feature maps to match input resolution
- ❖ Each block includes one 2×2 transposed convolution (up-conv), two 3×3 convolutions + ReLU

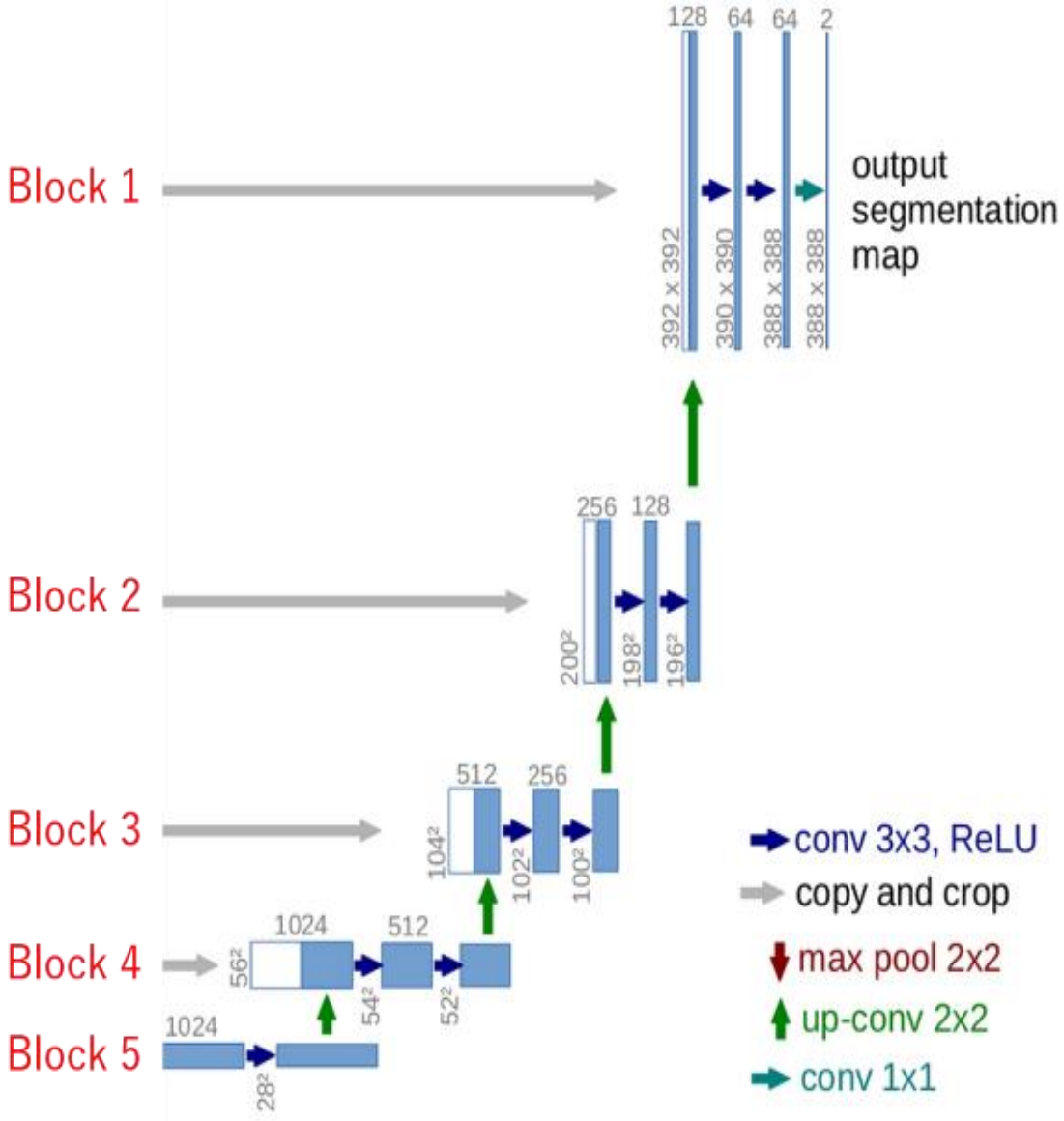
Ronneberger, O., Fischer, P., & Brox, T. (2015, May 18). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv.Org. https://arxiv.org/abs/1505.04597v1

Contracting Path (Encoder)



- **Block 1:**
  - Input: 572×572×1 (grayscale image)
  - Two 3×3 unpadded convolutions + ReLU → 64 channels
  - 2×2 max pooling (stride 2) → downsampled to 284×284
- **Block 2:**
  - Two 3×3 convolutions + ReLU → 128 channels
  - 2×2 max pooling → 140×140
- **Block 3 & Block 4:**
  - Same as previous blocks with doubled channels (256, 512)
  - Max pooling after each block halves spatial dimensions
- **Block 5 (Bottom):**
  - Two 3×3 convolutions + ReLU → 1024 channels
  - First conv in this block included here, second used in expanding path for symmetry

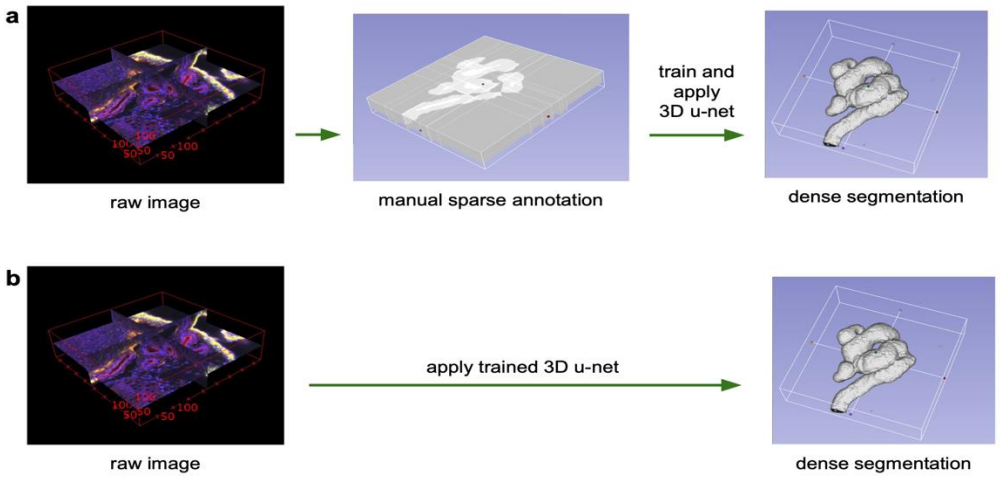
Expanding Path (Decoder)

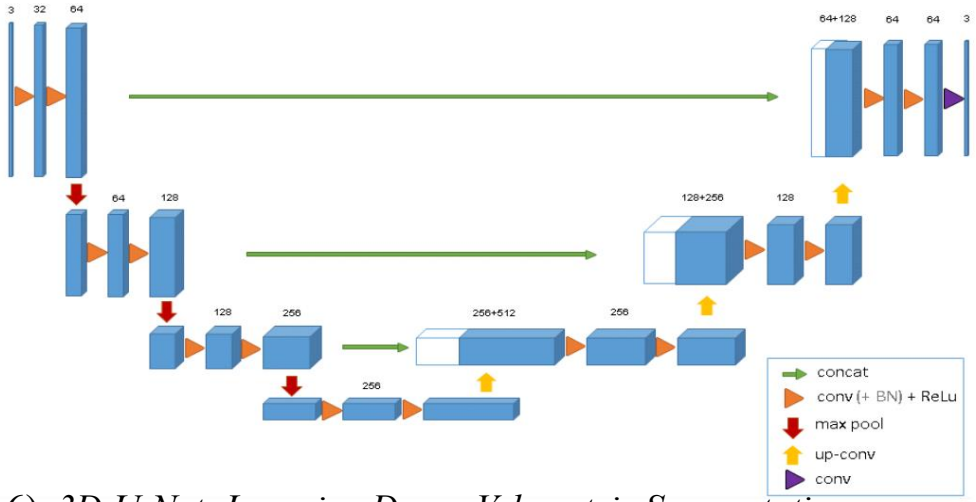


- **Block 5:**
  - Continues from the bottom block with a second 3×3 convolution + ReLU
  - Followed by a 2×2 up-convolution → doubles spatial resolution, reduces channels to 512
- **Block 4:**
  - Skip connection: concatenate encoder feature map (cropped to match size) → 1024 channels
  - Two 3×3 convolutions + ReLU → reduce to 512 channels
  - 2×2 up-convolution → upsample and reduce channels to 256
- **Block 3 & Block 2:**
  - Same as Block 4, with halved channels: 256→128→64
- **Block 1 (Final Block):**
  - After skip connection: 128 channels
  - Two 3×3 convolutions + ReLU → reduce to 64 channels
  - Final 1×1 convolution → maps to number of classes (e.g., 2 for binary)
  - Followed by activation function (e.g., sigmoid for binary classification)

3D U-Net

- Due to the abundance and representation power of volumetric data, most medical image modalities are three-dimensional. 3D U-Net was commonly used in Brain tumor segmentation (e.g., BraTS dataset), Lung nodule detection, and liver and pancreas segmentation.
- 3D U-Net is proposed to deal with 3D medical data directly. It replaces all 2D operations with their 3D counterparts. The users can annotate some slices in the volume to be segmented. The model then learns from these sparse annotations and provides a dense 3D segmentation.
- However, due to the limitation of computational resources, it only includes three down-sampling, which cannot effectively extract deep-layer image features, leading to limited segmentation accuracy for medical images.





Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation* (No. arXiv:1606.06650). arXiv. https://doi.org/10.48550/arXiv.1606.06650

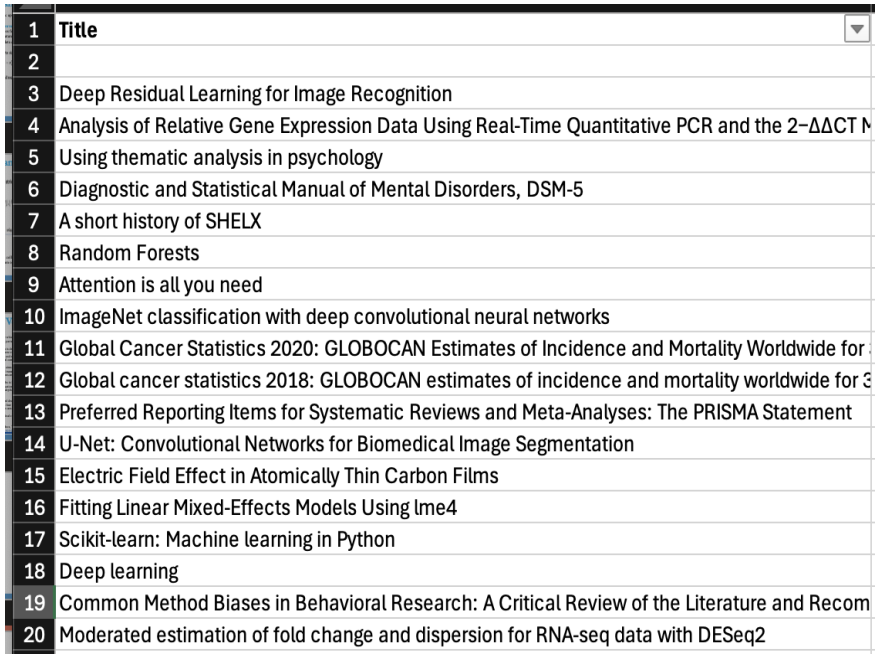
U-Net: Impact

- Since its introduction in 2015, U-Net has become probably the most well-known architecture for segmenting medical images, being cited over 100,000 times so far.
- A lot of variants of the model have been derived to progress the state-of-the-art (SOTA) based on it.

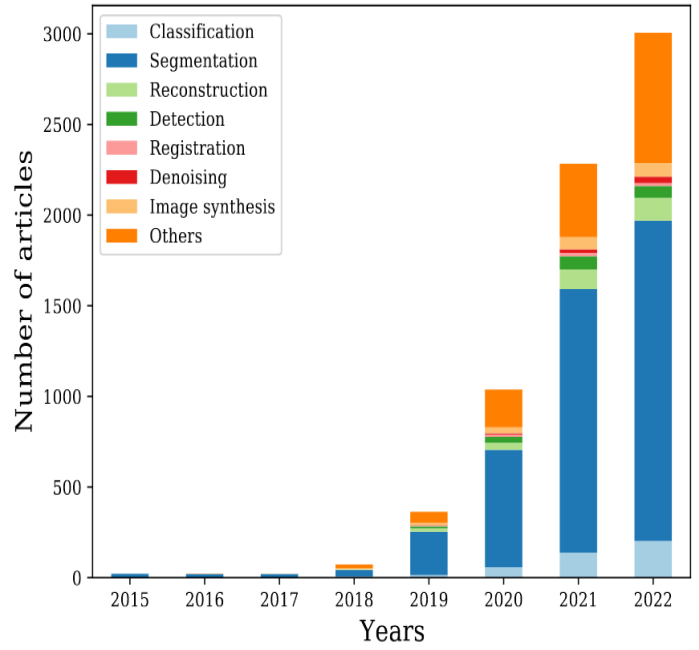
Feature

THE MOST-CITED PAPERS OF THE TWENTY-FIRST CENTURY

An exclusive *Nature* analysis reveals the 25 highest-cited papers published this century and explores why they are breaking records. **By Helen Pearson, Heidi Ledford, Matthew Hutson and Richard Van Noorden**



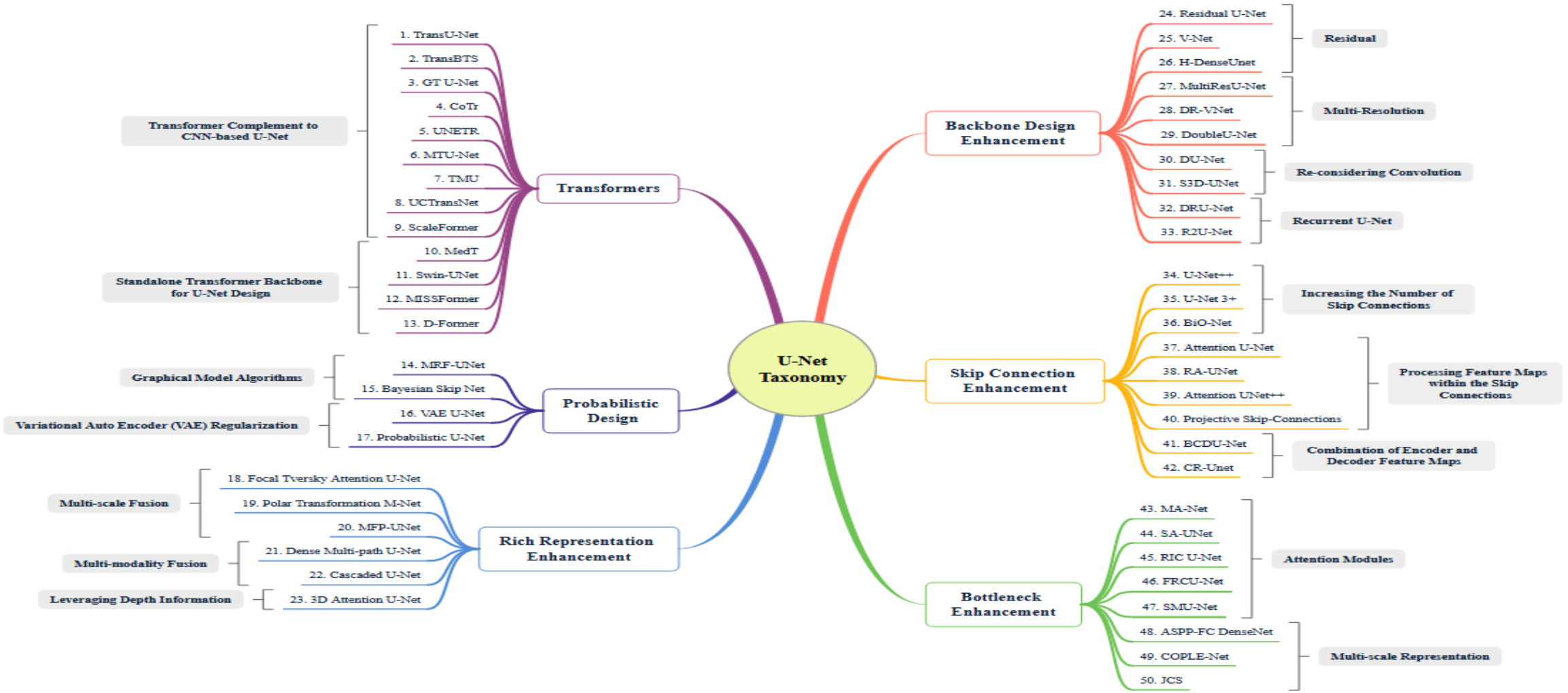
	Title
1	
2	
3	Deep Residual Learning for Image Recognition
4	Analysis of Relative Gene Expression Data Using Real-Time Quantitative PCR and the 2−ΔΔCT M
5	Using thematic analysis in psychology
6	Diagnostic and Statistical Manual of Mental Disorders, DSM-5
7	A short history of SHELX
8	Random Forests
9	Attention is all you need
10	ImageNet classification with deep convolutional neural networks
11	Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for
12	Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 3
13	Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement
14	U-Net: Convolutional Networks for Biomedical Image Segmentation
15	Electric Field Effect in Atomically Thin Carbon Films
16	Fitting Linear Mixed-Effects Models Using lme4
17	Scikit-learn: Machine learning in Python
18	Deep learning
19	Common Method Biases in Behavioral Research: A Critical Review of the Literature and Recom
20	Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2



Pearson, H., Ledford, H., Hutson, M., and Van Noorden, R. (2025) Exclusive: the most-cited papers of the twenty-first century, *Nature*. 588 |, Vol 640.

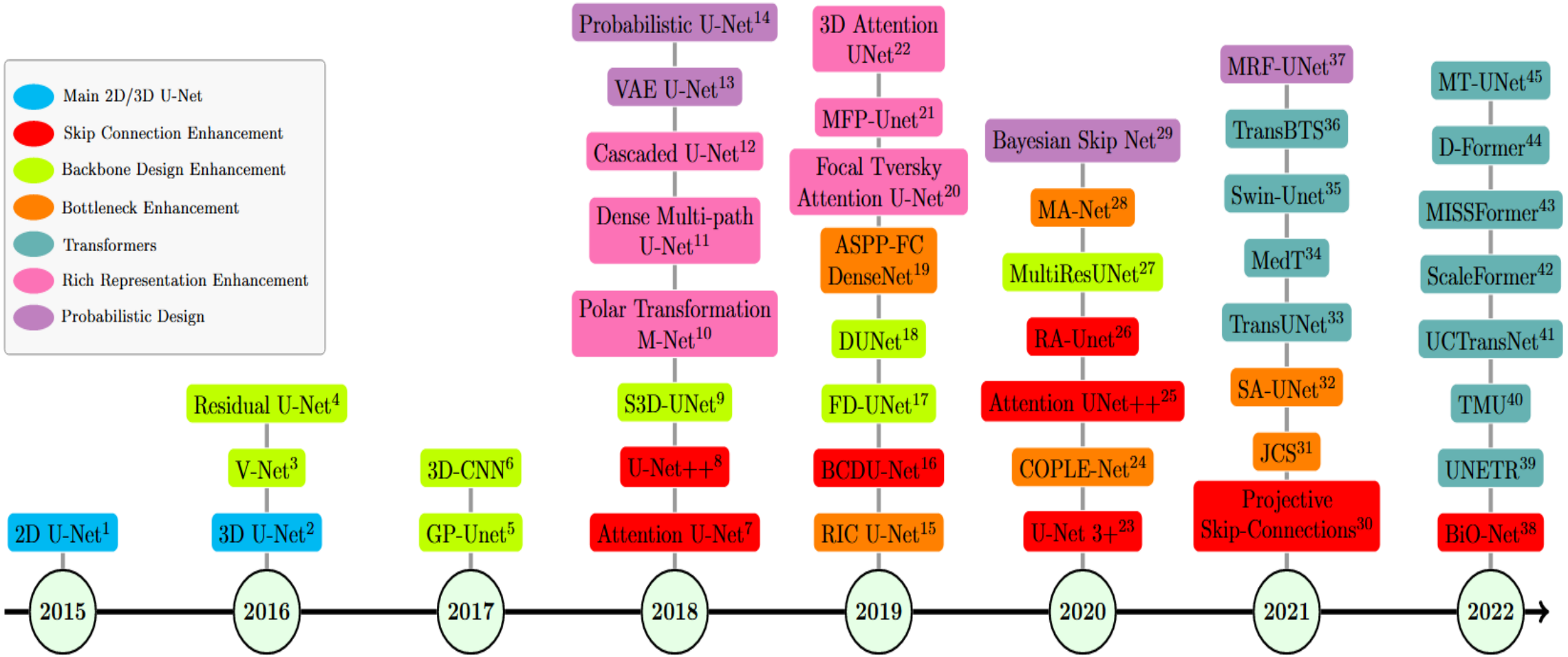
Azad, R., Aghdam, E. K., Rauland, A., Jia, Y., Avval, A. H., Bozorgpour, A., Karimijafarbigloo, S., Cohen, J. P., Adeli, E., & Merhof, D. (2022). *Medical Image Segmentation Review: The success of U-Net* (No. arXiv:2211.14830). arXiv. http://arxiv.org/abs/2211.14830

U-Net Taxonomy based on Design Ideas



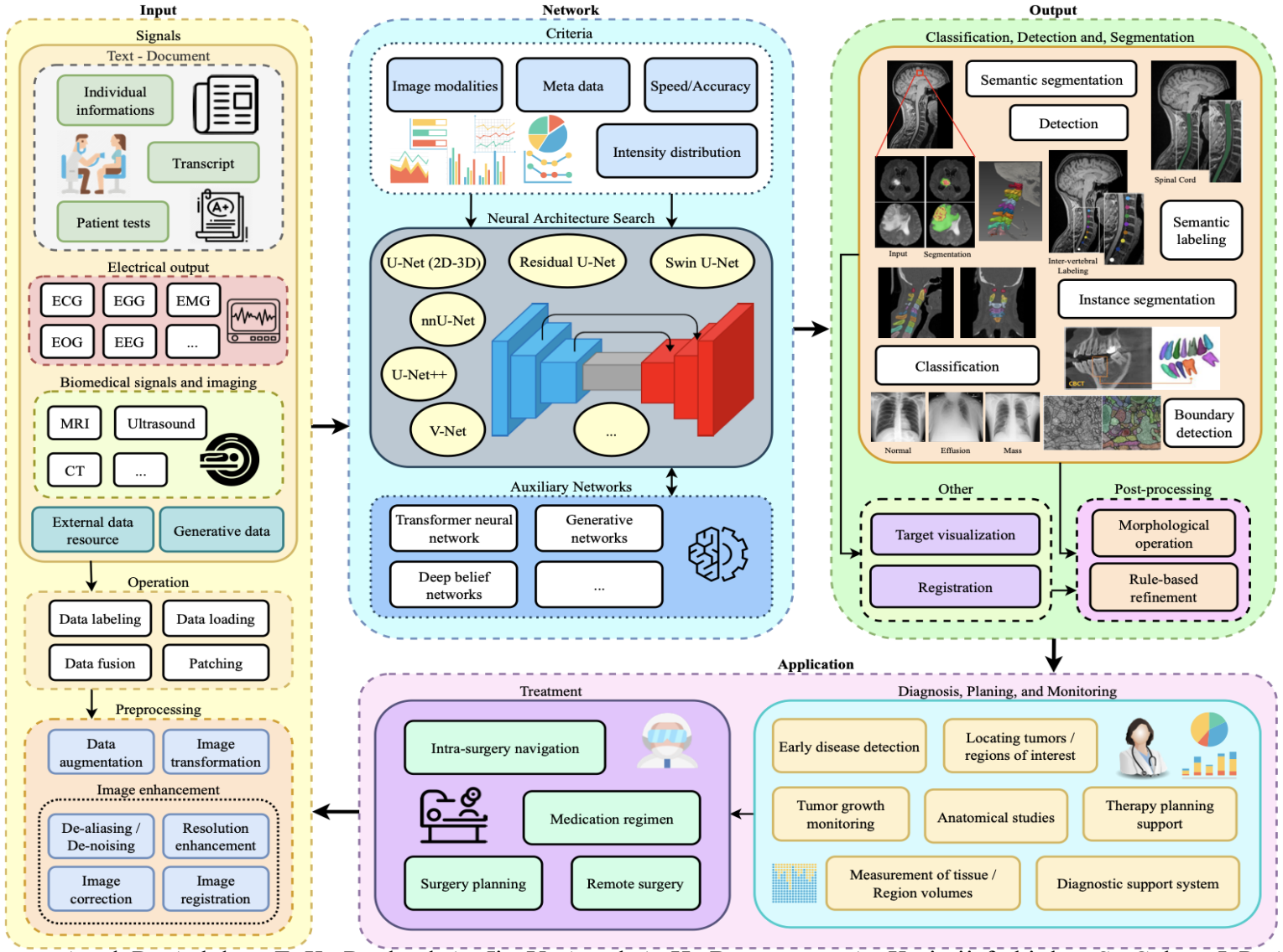
Azad, R., Aghdam, E. K., Rauland, A., Jia, Y., Avval, A. H., Bozorgpour, A., Karimijafarbigloo, S., Cohen, J. P., Adeli, E., & Merhof, D. (2022). *Medical Image Segmentation Review: The success of U-Net* (No. arXiv:2211.14830). arXiv. http://arxiv.org/abs/2211.14830

Timeline of Prominent U-Net Variants



Azad, R., Aghdam, E. K., Rauland, A., Jia, Y., Avval, A. H., Bozorgpour, A., Karimijafarbigloo, S., Cohen, J. P., Adeli, E., & Merhof, D. (2022). *Medical Image Segmentation Review: The success of U-Net* (No. arXiv:2211.14830). arXiv. http://arxiv.org/abs/2211.14830

U-Net in Clinical Image Analysis Pipelines



U-Net plays a central role in clinical image analysis pipelines

Overview of key stages:

- **Input Preparation:** Image acquisition, normalization, and preprocessing for consistent input format
- **Architecture Search:** Automatic selection of the most efficient U-Net variant via neural architecture search
- **Postprocessing:** Refinement of segmentation masks (e.g., morphological operations)
- **Clinical Application:** Supports decisions such as tumor growth tracking or treatment planning

Azad, R., Aghdam, E. K., Rauland, A., Jia, Y., Avval, A. H., Bozorgpour, A., Karimijafarbigloo, S., Cohen, J. P., Adeli, E., & Merhof, D. (2022). *Medical Image Segmentation Review: The success of U-Net* (No. arXiv:2211.14830). arXiv. http://arxiv.org/abs/2211.14830

Content

Improving U-Net

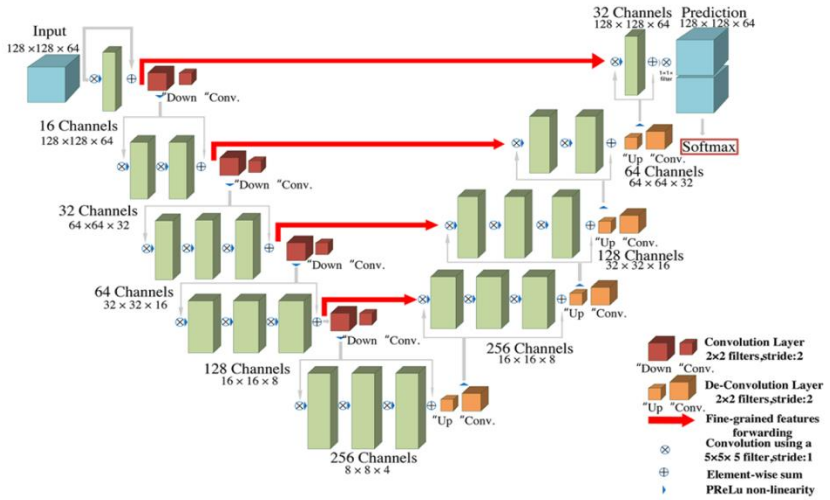
Numerous U-Net variants have emerged to address limitations in specific medical imaging tasks.

- **Backbone Design Enhancements:** Replace standard U-Net encoder with deeper/more powerful backbones (e.g., ResNet, EfficientNet, DenseNet); Improve feature extraction and convergence
- **Skip Connection Enhancements:** Use attention gates (Attention U-Net), dense connections (UNet++), or residual connections. Improve feature fusion and gradient flow
- **Bottleneck Enhancements:** Incorporate dilated convolutions, squeeze-and-excitation blocks, or atrous spatial pyramid pooling. Capture multi-scale context and expand receptive field
- **Transformer Integration:** Embed self-attention mechanisms in encoder, decoder, or bottleneck (TransUNet, UNETR). Improve global context modeling.
- **Rich Representation Enhancements:** Introduce multi-branch or multi-scale input/output streams (e.g., MultiResUNet). Enable robust learning from varying spatial scales and features
- **Probabilistic U-Nets:** Bayesian U-Net introduces variational inference for uncertainty estimation. Useful for detecting ambiguous or low-confidence regions in medical images.

Backbone Design Enhancement: V-Net

Feature	U-Net	V-Net
Dimensionality	2D (originally)	3D volumetric convolutions
Residual Connections	❌	✅ Improves training speed and convergence
Loss Function	Cross-entropy (class imbalance sensitive)	Dice loss (robust to imbalance)
Pooling Method	Max pooling	Strided convolutions (no pooling layers)
Augmentation	Limited	Advanced: B-spline deformations, histogram matching
Efficiency	Slice-by-slice, slower inference	Entire volume processed in 1 pass (~1s)
Clinical Relevance	Strong for 2D slices	Better for volumetric segmentation (MRI, CT)

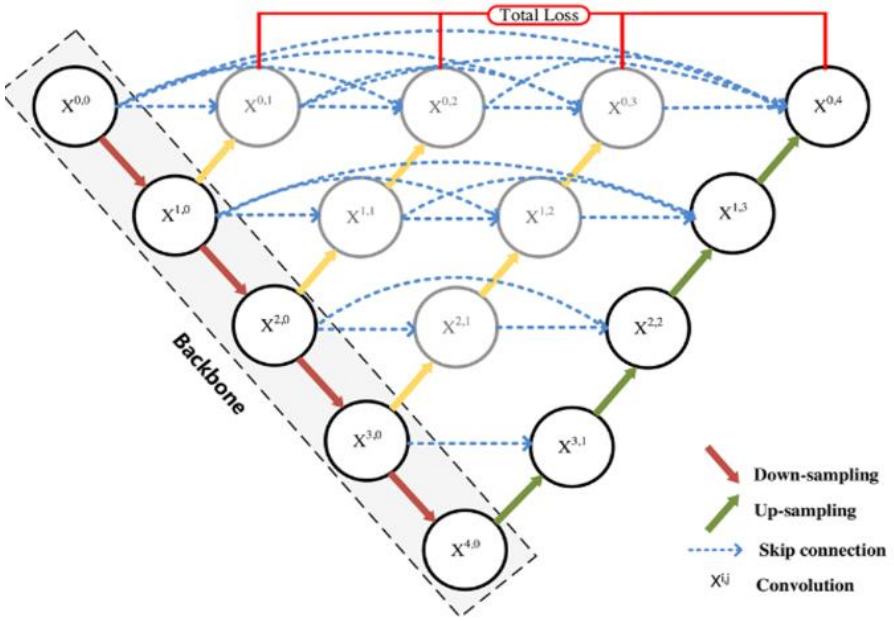
V-Net is a volumetric and residual version of U-Net, specifically designed for 3D medical images. It improves segmentation quality and training efficiency using: Fully 3D convolutions, Dice-based loss, and Residual learning. It is a **strong alternative to U-Net**, particularly for 3D volumetric segmentation tasks like prostate, liver, or brain scans.

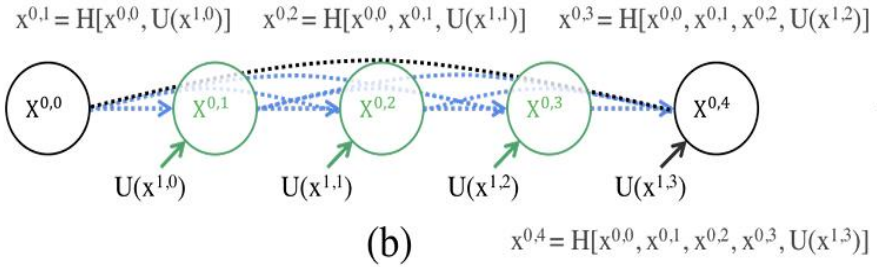


Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). *V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation* (No. arXiv:1606.04797). arXiv. https://doi.org/10.48550/arXiv.1606.04797

Skip Connection Enhancement: UNet++

Feature	U-Net	UNet++
Skip Connections	Direct skip from encoder to decoder	**Dense + Nested skip paths** to narrow semantic gap
Decoder Optimization	Jointly optimized	**Deep supervision** improves gradient flow and allows for pruning
Feature Fusion	Concatenation only	**Progressive fusion** of features with increasing semantic richness
Loss Function	Binary Cross Entropy	Combined **BCE + Dice** at multiple levels
Accuracy (IoU Gain)	Baseline	**+3.9 IoU** over U-Net and **+3.4 IoU** over Wide U-Net
Flexibility	Single-output	**Multi-output + fast inference mode via pruning**

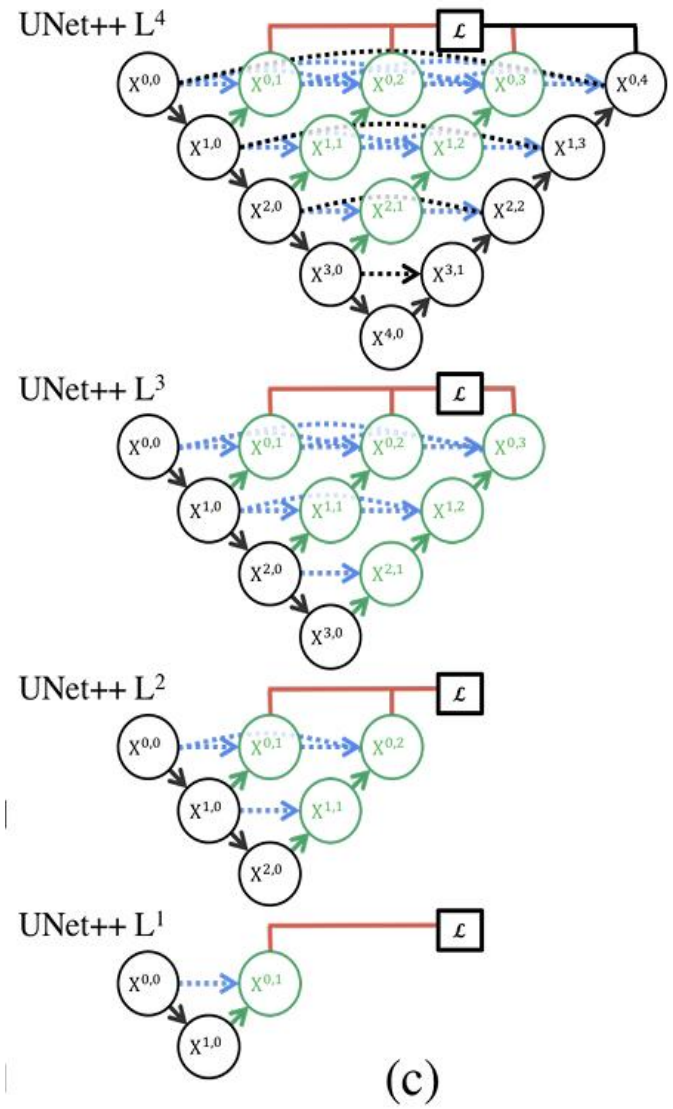




$$x^{i,j} = \begin{cases} \mathcal{H}\left(x^{i-1,j}\right), & j = 0 \\ \mathcal{H}\left(\left[\left[x^{i,k}\right]_{k=0}^{j-1}, \mathcal{U}(x^{i+1,j-1})\right]\right), & j > 0 \end{cases}$$

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2018). *UNet++: A Nested U-Net Architecture for Medical Image Segmentation* (No. arXiv:1807.10165). arXiv. https://doi.org/10.48550/arXiv.1807.10165

UNet++


UNet++ L4
UNet++ L3
UNet++ L2
UNet++ L1
(c)

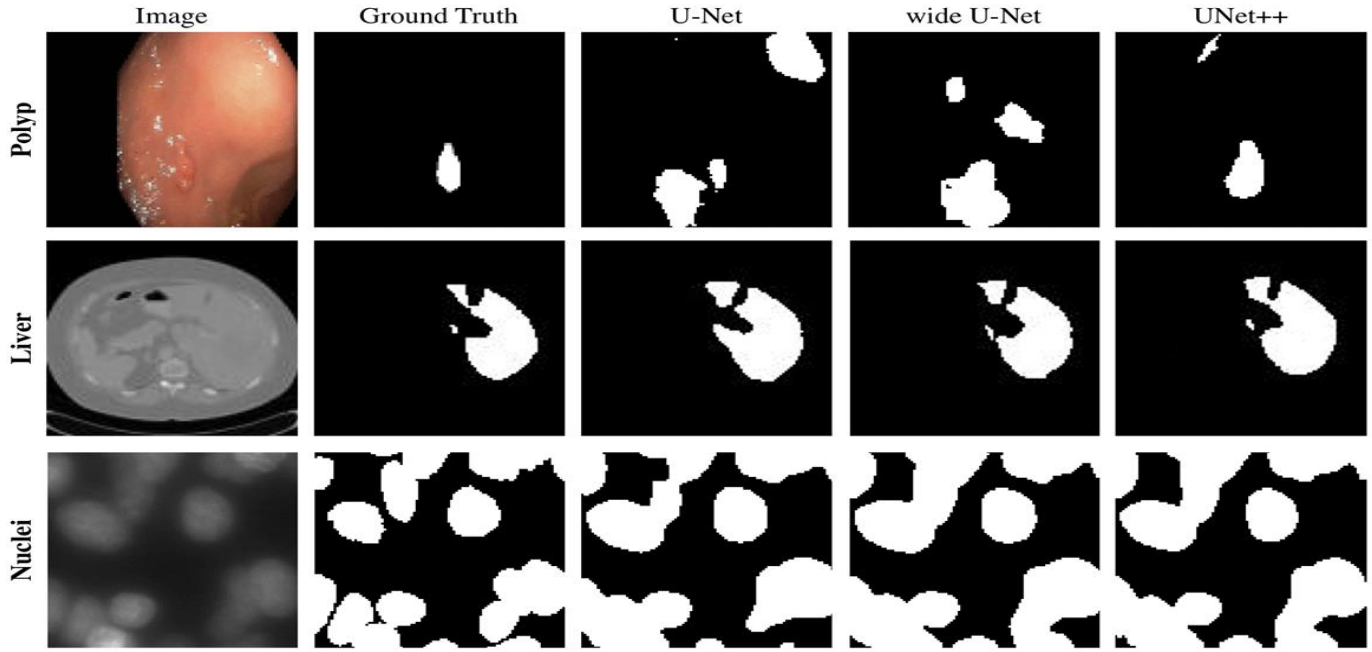
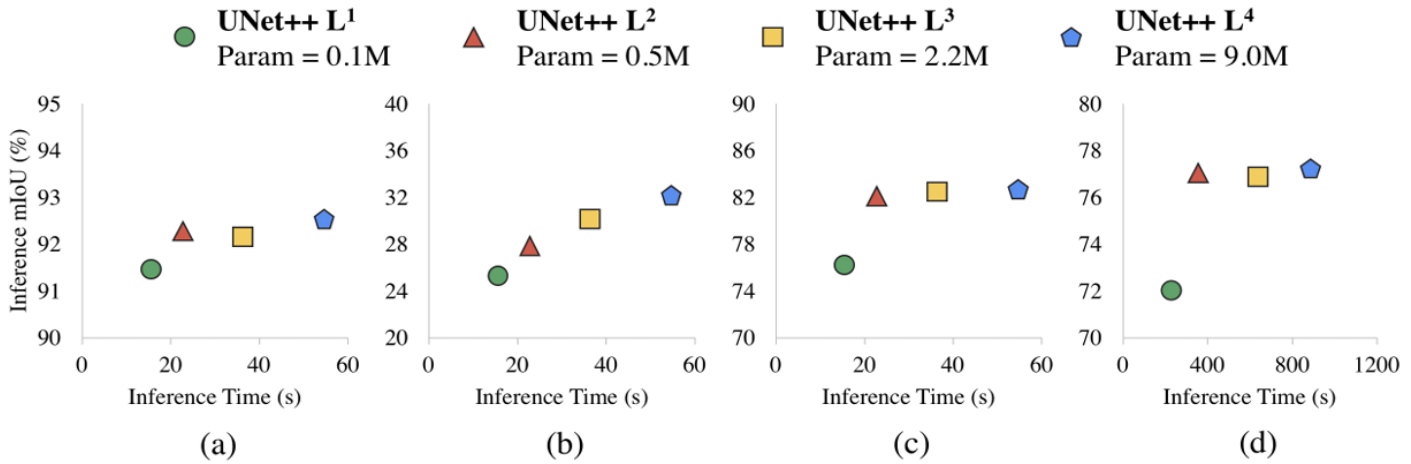
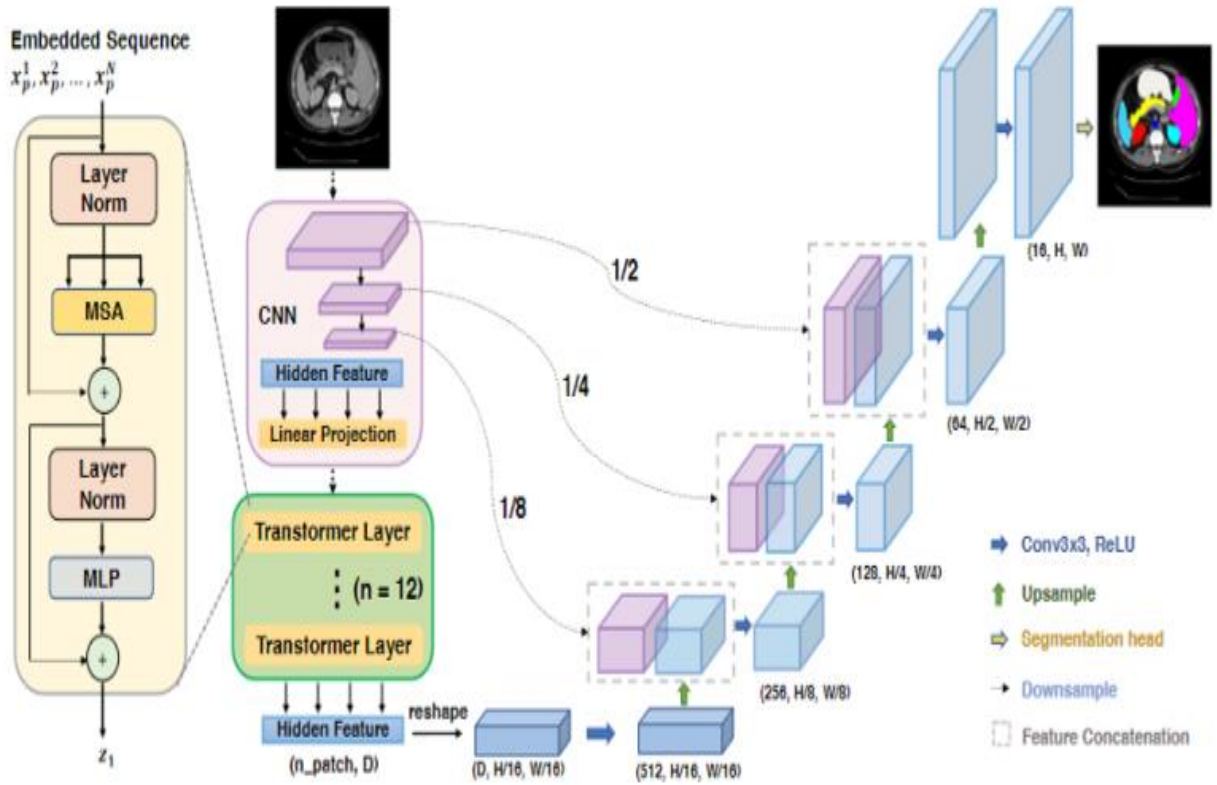


Image
Ground Truth
U-Net
wide U-Net
UNet++
Polyp
Liver
Nuclei


UNet++ L1
Param = 0.1M
UNet++ L2
Param = 0.5M
UNet++ L3
Param = 2.2M
UNet++ L4
Param = 9.0M
Inference mIoU (%)
Inference Time (s)
(a)
(b)
(c)
(d)

Transformer Complement: TransU-Net

- 🔁 **CNN + Transformer Hybrid Encoder**:
  - CNN backbone (e.g., ResNet-50) captures low-level spatial features.
  - Transformer layers model long-range dependencies on patch sequences.
- 🔄 **Cascaded Upsampler (CUP) Decoder**:
  - Multistage upsampling path with skip connections.
  - Precise boundary localization and semantic restoration.
- 🔧 **Skip Connections Enhanced**:
  - Multi-resolution skip connections from encoder to decoder.
  - U-Net-like structure helps retain fine spatial detail lost in transformer layers.
- 🎯 **Superior Accuracy on Benchmarks**:
  - Outperforms U-Net, V-Net, and attention U-Net on Synapse and ACDC datasets.



- **Avg. Dice Score** (Synapse):
  - U-Net: 74.68%,
  - TransUNet: **77.48%**
- **Avg. Dice Score** (ACDC):
  - U-Net: 87.55%,
  - TransUNet: **89.71%**

Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A. L., & Zhou, Y. (2021). *TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation* (No. arXiv:2102.04306). arXiv. https://doi.org/10.48550/arXiv.2102.04306

Swin-UNet

🧠 Motivations

- CNN-based U-Nets perform well but struggle to model **global and long-range dependencies** due to locality of convolutions.
- Transformers offer better global context modeling but lack spatial detail recovery.

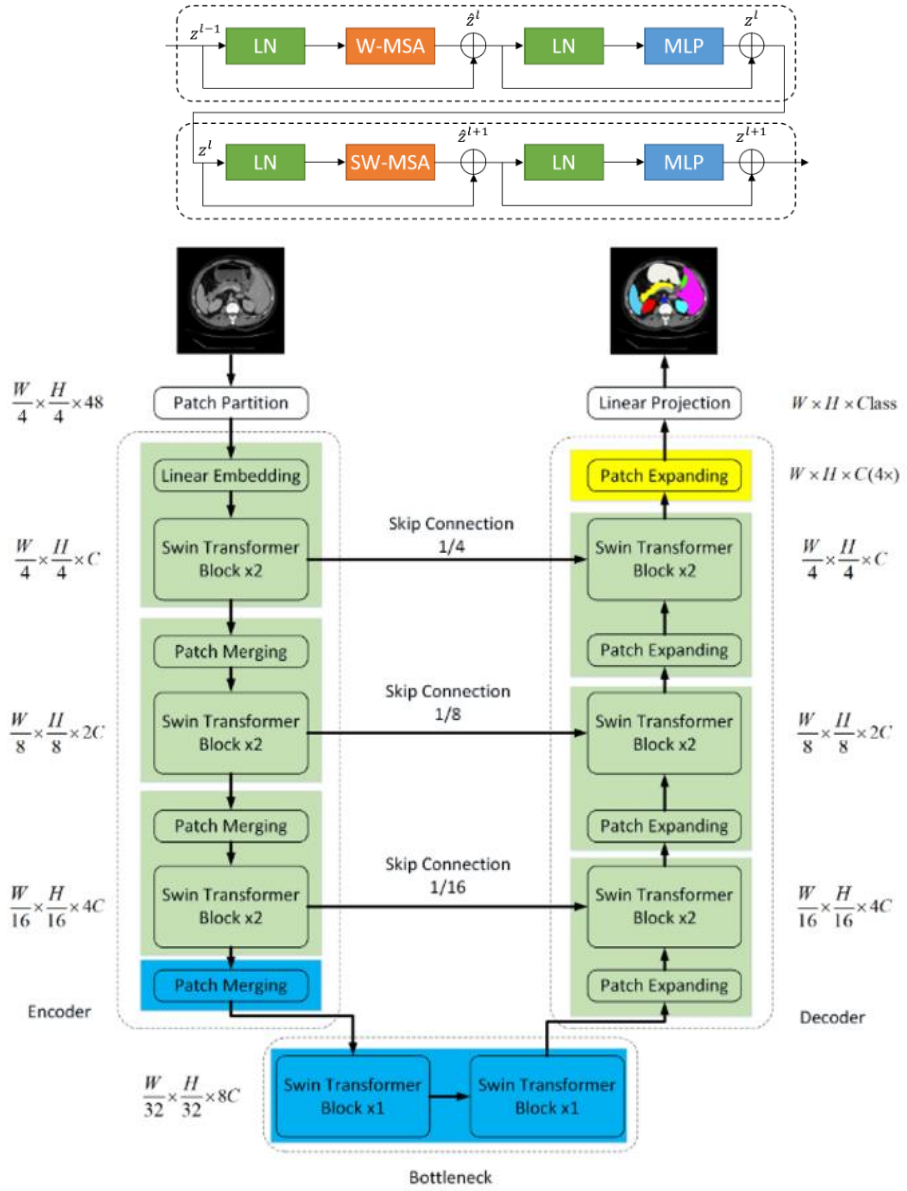
🧠 Proposed Architecture – Swin-Unet

- Pure Transformer-based U-Net-style architecture:
  - **Encoder, bottleneck, decoder, skip connections** — all built from **Swin Transformer blocks**.
  - Uses **shifted window attention** for local-global interaction.
- **Patch Merging (Downsampling)** and **Patch Expanding (Upsampling)** layers replace pooling and deconvolution.
- Skip connections preserve spatial resolution and complement global features.

📊 On **Synapse dataset**:

- Dice Score (DSC): **79.13%**, HD: **21.55**
- Outperforms U-Net, TransUNet, and Att-UNet in boundary accuracy (HD).

- On **ACDC dataset**:
  - Dice Score: **90.00%** — better than U-Net (87.55%) and TransUNet (89.71%)

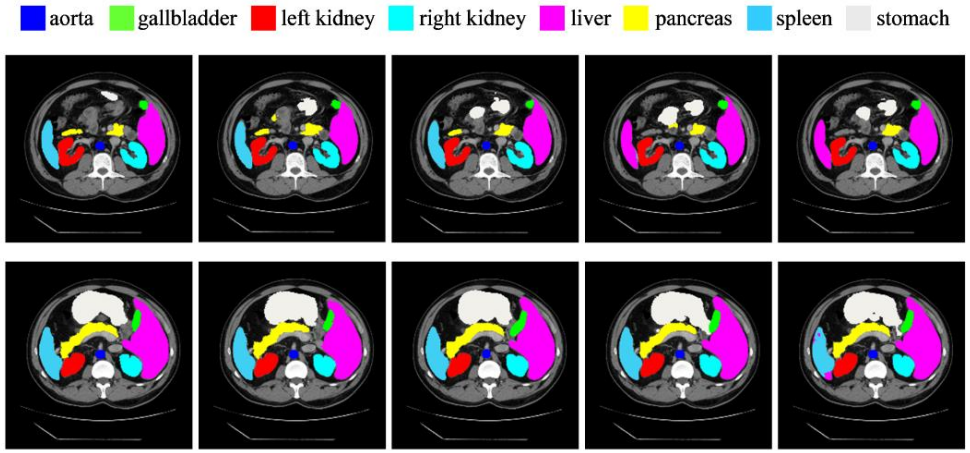


Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., & Wang, M. (2021). *Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation* (No. arXiv:2105.05537). arXiv. https://doi.org/10.48550/arXiv.2105.05537

Swin-U-Net

- Swin-UNet presented SOTA results over the CNN-Transformer hybrid structures like TransUNet and demonstrated the robust generalization ability with the help of two multiorgan (Synapse) and cardiac (ACDC) segmentation datasets.

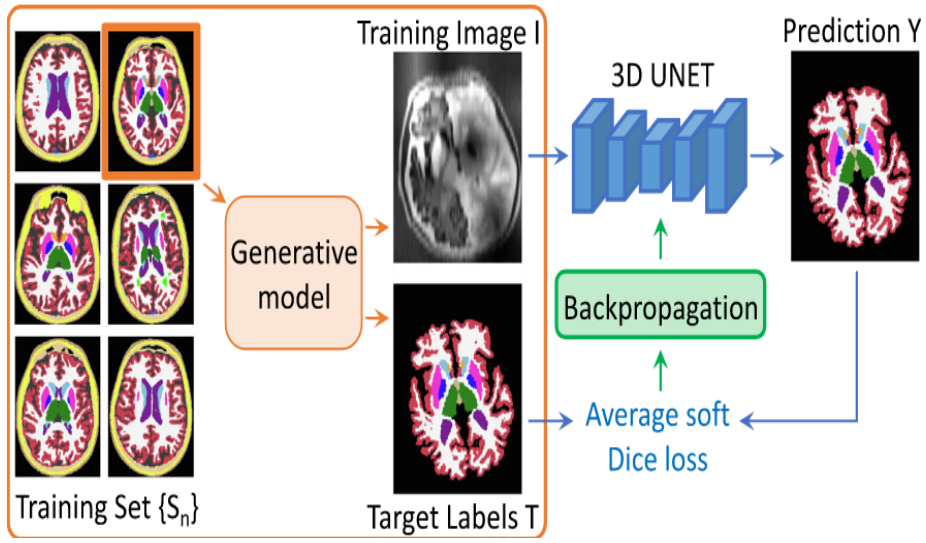
Methods	DSC↑	HD↓	Aorta	Gallbladder	Kidney(L)	Kidney(R)	Liver	Pancreas	Spleen	Stomach
V-Net [35]	68.81	-	75.34	51.87	77.10	**80.75**	87.84	40.05	80.56	56.98
DARR [36]	69.77	-	74.74	53.77	72.31	73.24	94.08	54.18	89.90	45.96
R50 U-Net [2]	74.68	36.87	87.74	63.66	80.60	78.19	93.74	56.90	85.87	74.16
U-Net [3]	76.85	39.70	89.07	**69.72**	77.77	68.60	93.43	53.98	86.67	75.58
R50 Att-UNet [2]	75.57	36.97	55.92	63.91	79.20	72.71	93.56	49.37	87.19	74.95
Att-UNet [37]	77.77	36.02	**89.55**	68.88	77.98	71.11	93.57	**58.04**	87.30	75.75
R50 ViT [2]	71.29	32.87	73.73	55.13	75.80	72.20	91.51	45.99	81.99	73.95
TransUnet [2]	77.48	31.69	87.23	63.13	81.87	77.02	94.08	55.86	85.08	75.62
SwinUnet	**79.13**	**21.55**	85.47	66.53	**83.28**	79.61	**94.29**	56.58	**90.66**	**76.60**



Methods	DSC	RV	Myo	LV
R50 U-Net	87.55	87.10	80.63	94.92
R50 Att-UNet	86.75	87.58	79.20	93.47
R50 ViT	87.57	86.07	81.88	94.75
TransUnet	89.71	88.86	84.53	95.73
SwinUnet	**90.00**	88.55	**85.62**	**95.83**

SynthSeg

- **Problem:** CNNs like U-Net fail to generalize across different MRI contrasts and resolutions.
- **Solution:** **SynthSeg**, a 3D U-Net trained entirely on *synthetic* images with **domain randomization**.
- **Training Pipeline:**
  - Generates synthetic 3D brain scans by sampling from a generative model conditioned on label maps.
  - Applies **random transformations**: spatial deformation, resolution changes, contrast variation, artifacts.
  - Learns to segment across diverse modalities (T1, T2, PD, FLAIR, CT) and resolutions (1–7mm).
- **No need for retraining:** Once trained, SynthSeg generalizes to unseen domains directly.



		T1–39	ADNI	T1mix	FSM-T1	MSp-T1	FSM-T2	FSM-DBS	MSp-PD	FLAIR	CT
T1 baseline	Dice	0.91	0.83	0.86	0.84	0.82	–	–	–	–	–
	SD95	1.31	2.63	2.14	2.09	3.55	–	–	–	–	–
nnUNet (Isensee et al., 2021)	Dice	0.91	0.82	0.84	0.84	0.81	–	–	–	–	–
	SD95	1.31	2.8	2.32	2.11	3.71	–	–	–	–	–
TTA (Karani et al., 2021)	Dice	–	0.83	0.87	0.87	0.85	0.82	0.71	0.8	0.71	0.46
	SD95	–	2.26	1.73	1.72	2.14	2.35	4.48	3.71	3.95	19.43
SIFA (Chen et al., 2019)	Dice	–	0.8	0.82	0.84	0.84	0.82	0.82	0.74	0.73	0.62
	SD95	–	3.03	2.24	2.21	2.57	2.32	2.09	4.41	3.30	4.51
SAMSEG (Puonti et al., 2016)	Dice	0.85	0.81	0.86	0.86	0.83	0.82	0.81	0.81	0.64	0.71
	SD95	1.85	3.09	1.77	1.81	2.47	2.21	2.34	2.99	3.67	3.36
SynthSeg (ours)	Dice	0.88	**0.84**	0.87	**0.88**	**0.86***	**0.86***	**0.86***	**0.84***	**0.78***	**0.76***
	SD95	1.5	**2.18***	**1.69***	**1.59***	**1.89***	**1.83***	**1.81***	**2.06***	**2.35***	**3.29***

Fig. 4. Box plots showing Dice scores obtained by all methods for every dataset. For each box, the central mark is the median; edges are the first and third quartiles; and outliers are marked with ♦.

Billot, B., Greve, D. N., Puonti, O., Thielscher, A., Van Leemput, K., Fischl, B., Dalca, A. V., & Iglesias, J. E. (2023). SynthSeg: Segmentation of brain MRI scans of any contrast and resolution without retraining. *Medical Image Analysis*, *86*, 102789. https://doi.org/10.1016/j.media.2023.102789

No New U-Net (nnU-Net)

Motivation: U-Net requires manual tuning, which is error-prone and hard to generalize across datasets.

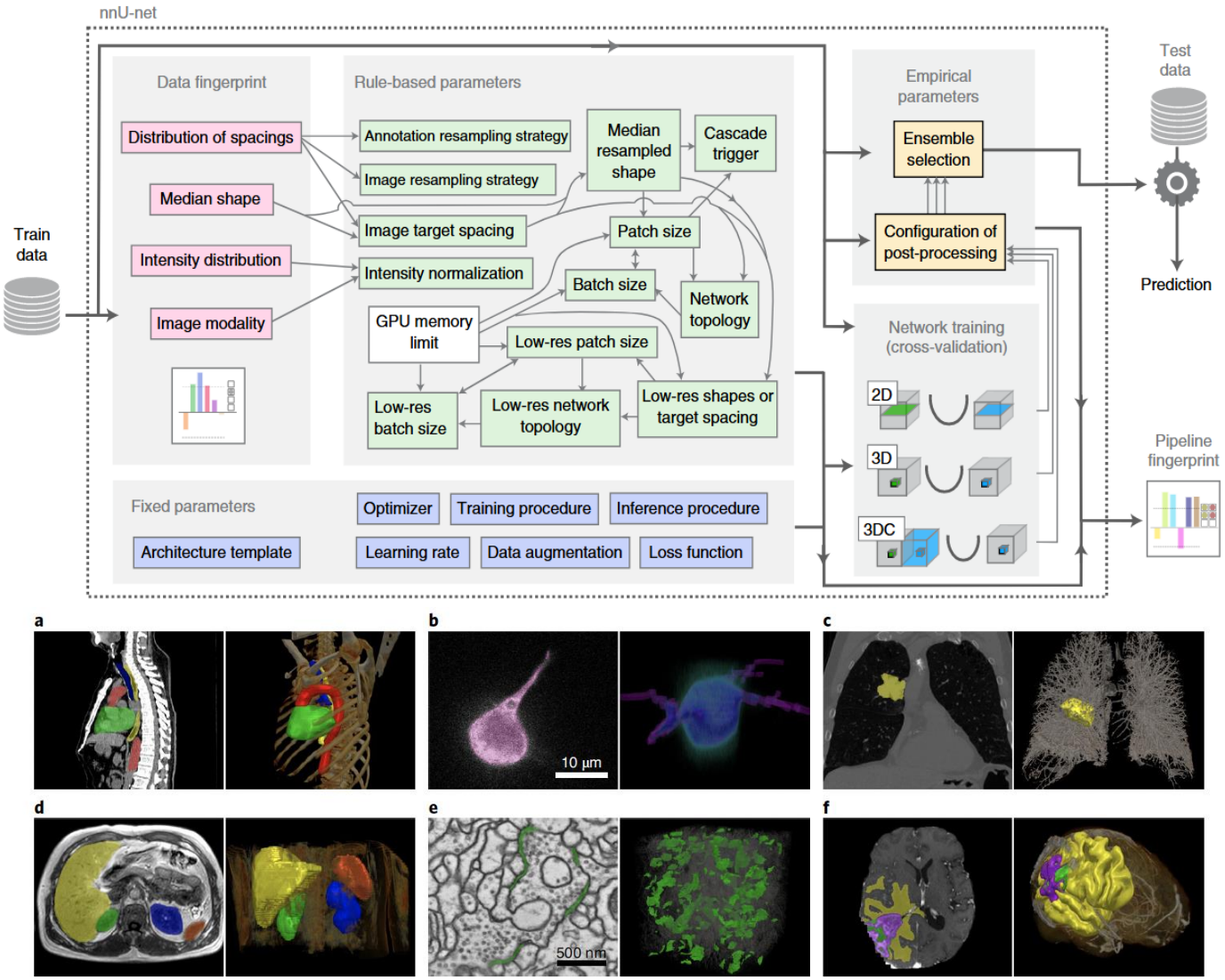
Core Idea: Systematize the pipeline design using **domain knowledge and rules**, allowing automatic adaptation to new datasets.

Pipeline Automation:

- **Fixed parameters:** Common across tasks (e.g., U-Net backbone, leaky ReLU, instance norm).
- **Rule-based parameters:** Heuristic functions linking dataset properties to pipeline choices.
- **Empirical parameters:** Limited optimization via cross-validation (e.g., post-processing).

Framework Includes: Intensity normalization; Resampling and patch-size selection; Network topology and training schedule; Sliding window inference + Gaussian weighting.

- **Results:** Achieved state-of-the-art (SOTA) in **33/53 tasks** across 23 datasets without manual tuning.



Isensee, F., Jaeger, P. F., Kohl, S. A. A., Petersen, J., & Maier-Hein, K. H. (2021). nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, *18*(2), 203–211. https://doi.org/10.1038/s41592-020-01008-z

nnU-Net

1. Dataset Fingerprints

- **Definition:** A compact summary of dataset properties automatically extracted before training.
- **Examples of fingerprint features:**
  - Image spacing and resolution;
  - Image sizes and aspect ratios;
  - Number of classes; Modality (e.g., CT, MRI)
- **Purpose:** Provides the basis for automatic pipeline configuration, replacing manual inspection.

2. Fixed Parameters

- **Definition:** Parameters that remain constant across all tasks and datasets.
- **Examples:** U-Net architecture template;
  - Instance normalization; Leaky ReLU activation;
  - Data augmentation pipeline structure
- **Purpose:** Leverages prior knowledge from segmentation tasks to avoid unnecessary tuning.

3. Rule-Based Parameters

- **Definition:** Parameters dynamically chosen using
- **heuristics** derived from the dataset fingerprint.
- **Examples:**
  - Patch size and batch size (based on image size and GPU memory)
  - Number of downsampling steps
  - Choice between 2D, 3D full-resolution, or 3D low-resolution U-Net
- **Purpose:** Adapt the pipeline design automatically to suit dataset characteristics.

4. Empirical Parameters

- **Definition:** Parameters that are fine-tuned **empirically**, typically through cross-validation.
- **Examples:**
  - Use of post-processing (e.g., removing small false-positive regions)
  - Ensembling strategies
- **Purpose:** Introduce minor empirical refinements for performance boost without extensive manual search.

nnU-Net: Impact

- **Standardized Benchmarking:**
  - nnU-Net became the **de facto baseline** for biomedical segmentation tasks cited over 5500 times.
  - Automatically configured pipelines have redefined expectations in medical imaging competitions.
- **Performance Across Modalities:**
  - Achieved **state-of-the-art results** in 33 of 53 tasks across 23 public datasets (e.g., KiTS, BraTS, ACDC).
  - Demonstrated **cross-modality generalization** (CT, MRI, PET) without manual tuning.
- **Reduced Entry Barrier:**
  - Enabled non-experts to build highly competitive segmentation models without deep learning expertise.
  - Popular in clinical, academic, and industrial settings.
- **Reproducibility and Open Science:**
  - Fully open-source with reproducible configurations and modular design.
  - Inspired a generation of autoML tools and plug-and-play segmentation pipelines.
- **Long-Term Influence:**
  - nnU-Net's automated approach has shaped the development of follow-up frameworks (e.g., nnFormer, nnSAM).
  - Highlights the power of **rule-based heuristics** and data-driven design in practical deep learning systems.

U-Mamba

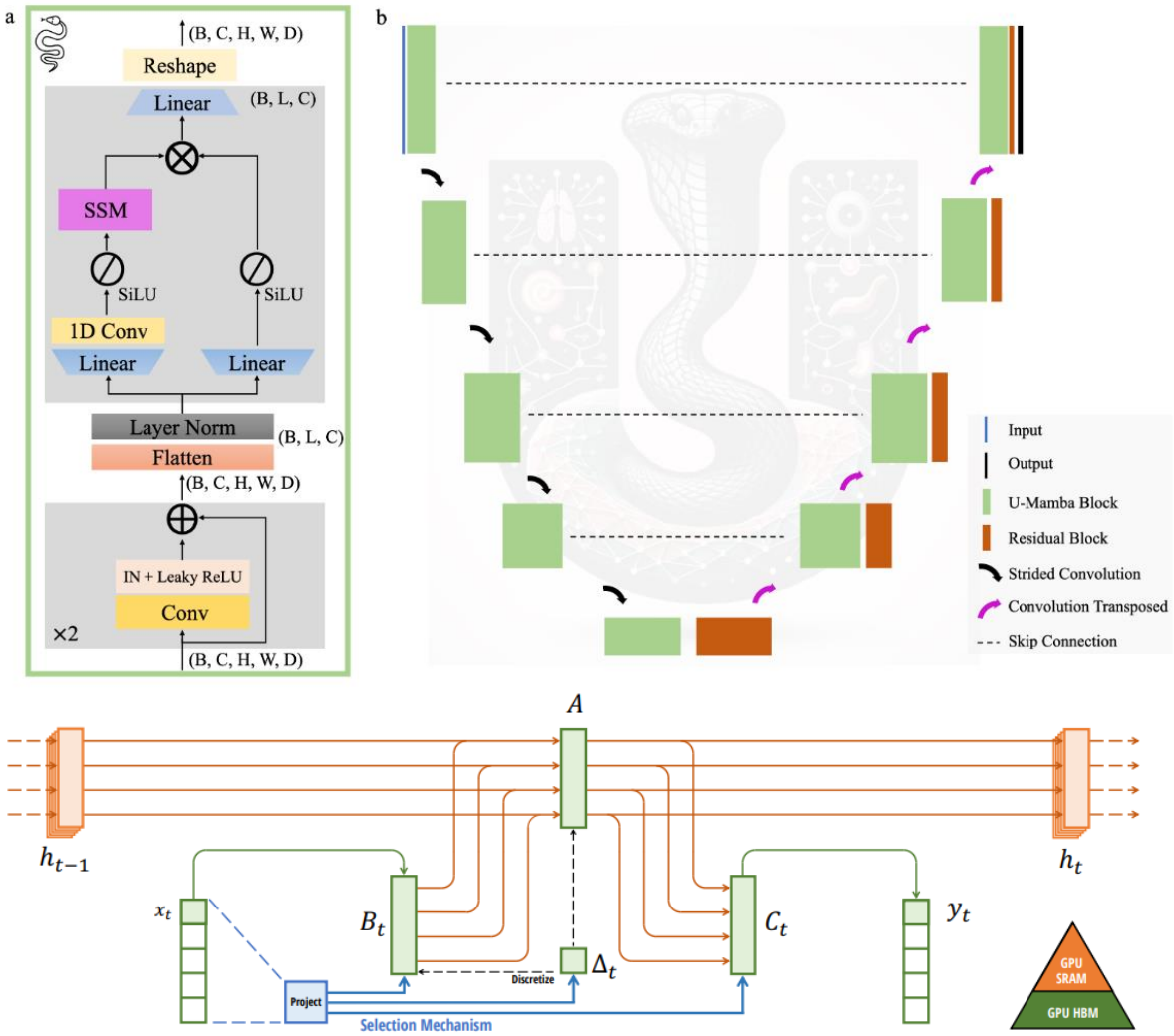
Motivation: CNNs (e.g., U-Net) suffer from **limited receptive fields**. Transformers offer global context but are **computationally expensive**. There is a need for models that **efficiently capture long-range dependencies**.

Key Contributions:

- **U-Mamba** combines:
  - **CNN Residual blocks**: for local feature extraction.
  - **Mamba blocks (SSMs)**: for scalable long-range dependency modeling.
- Employs a **hybrid encoder-decoder** architecture with skip connections.
- **Self-configuring** like nnU-Net: adapts automatically to various datasets.

Architecture Overview:

- Encoder: Residual + Mamba blocks.
- Decoder: Residual blocks with transposed convolution.
- Variants: U-Mamba_Bot (bottleneck only) and U-Mamba_Enc (full encoder).

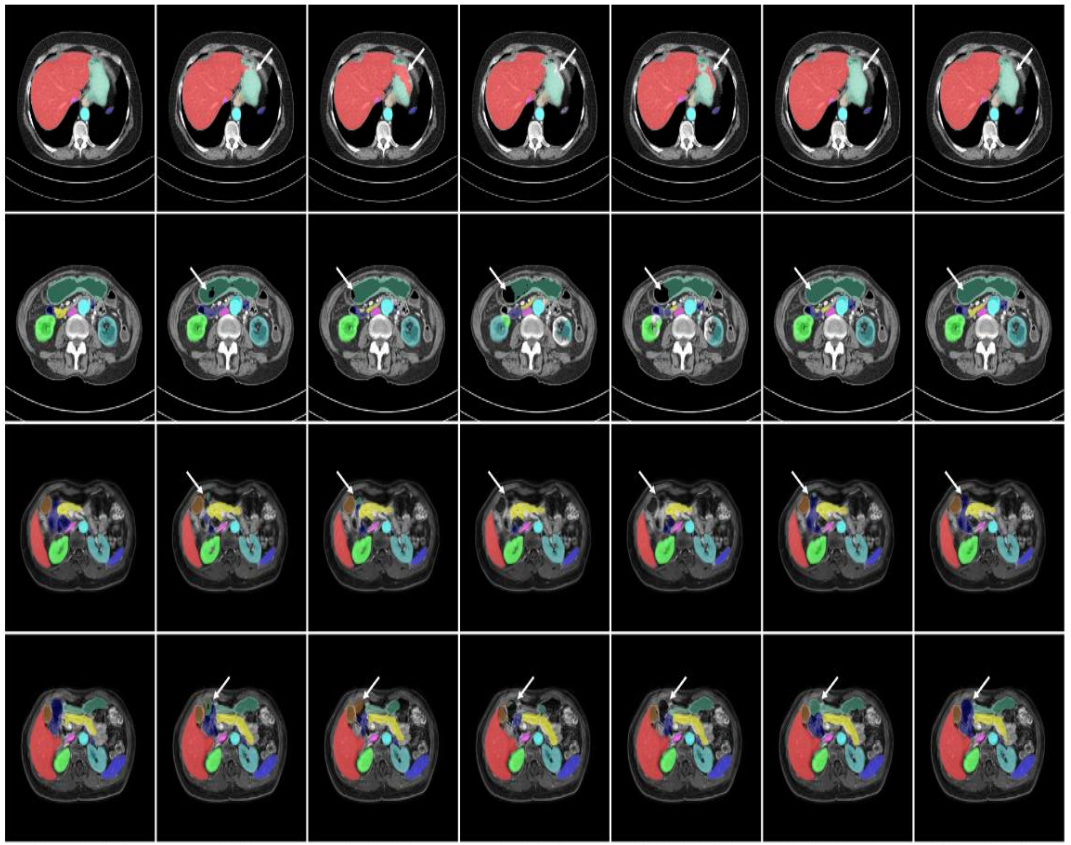


Gu, A., & Dao, T. (2024). *Mamba: Linear-Time Sequence Modeling with Selective State Spaces* (No. arXiv:2312.00752). arXiv. https://doi.org/10.48550/arXiv.2312.00752

Ma, J., Li, F., & Wang, B. (2024). *U-Mamba: Enhancing Long-range Dependency for Biomedical Image Segmentation* (No. arXiv:2401.04722). arXiv. http://arxiv.org/abs/2401.04722

U-Mamba

- U-Mamba also enjoys a self-configuration mechanism, as it is implemented within the nnU-Net framework.
- It outperforms STOA CNN-based and Transformer-based segmentation networks across all tasks.



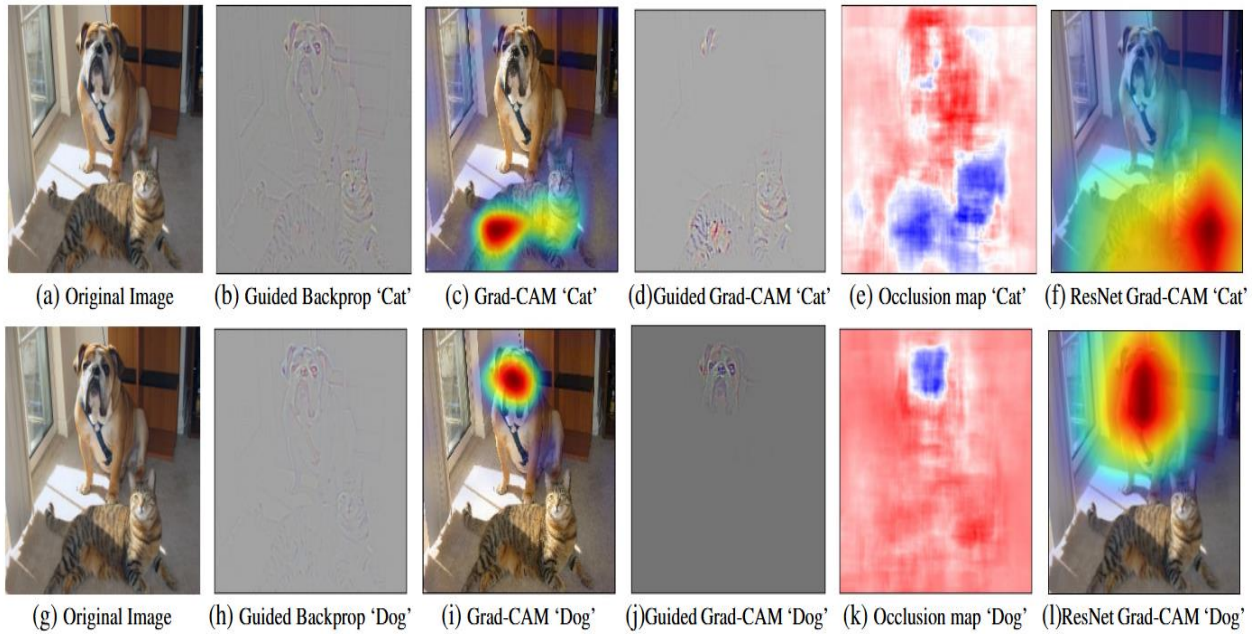
Methods	Organs in Abdomen CT		Organs in Abdomen MRI	
	DSC	NSD	DSC	NSD
nnU-Net	0.8615±0.0790	0.8972±0.0824	0.8309±0.0769	0.8996±0.0729
SegResNet	0.7927±0.1162	0.8257±0.1194	0.8146±0.0959	0.8841±0.0917
UNETR	0.6824±0.1506	0.7004±0.1577	0.6867±0.1488	0.7440±0.1627
SwinUNETR	0.7594±0.1095	0.7663±0.1190	0.7565±0.1394	0.8218±0.1409
U-Mamba_Bot	**0.8683±0.0808**	**0.9049±0.0821**	**0.8453±0.0673**	**0.9121±0.0634**
U-Mamba_Enc	**0.8638±0.0908**	**0.8980±0.0921**	**0.8501±0.0732**	**0.9171±0.0689**

Methods	Organs in Abdomem MRI		Instruments in Endoscopy		Cells in Microscopy
	DSC	NSD	DSC	NSD	F1
nnU-Net	0.7450±0.1117	0.8153±0.1145	0.6264±0.3024	0.6412±0.3074	0.5383±0.2657
SegResNet	0.7317±0.1379	0.8034±0.1386	0.5820±0.3268	0.5968±0.3303	0.5411±0.2633
UNETR	0.5747±0.1672	0.6309±0.1858	0.5017±0.3201	0.5168±0.3235	0.4357±0.2572
SwinUNETR	0.7028±0.1348	0.7669±0.1442	0.5528±0.3089	0.5683±0.3123	0.3967±0.2621
U-Mamba_Bot	**0.7588±0.1051**	**0.8285±0.1074**	**0.6540±0.3008**	**0.6692±0.3050**	**0.5389±0.2817**
U-Mamba_Enc	**0.7625±0.1082**	**0.8327±0.1087**	**0.6303±0.3067**	**0.6451±0.3104**	**0.5607±0.2784**

Interpretability

- **Medical imaging**: Interpret U-Net predictions for lesion/tumor segmentation using Grad-CAM heatmaps.
- **Weak supervision**: Use Grad-CAM as a proxy for segmentation maps when pixel-level labels are missing (e.g., CAM-based segmentation).
- **Training verification**: Check if the model is focusing on anatomically relevant regions.

Methods such as occlusion sensitivity, Grad-CAM and Smoothgrad help them address the question: "Which parts of the image were important in arriving to a given decision of a classification?"



(a) Original image with a cat and a dog.
(b) Guided Backpropagation highlights all contributing features.
(c, f) Grad-CAM localizes class-discriminative regions for VGG-16 and ResNet-18.
(d) Guided Grad-CAM (b × c) provides high-resolution, class-specific visualizations.
(e) Occlusion sensitivity yields similar results to Grad-CAM but is computationally more expensive.
Red in (c, f, i, l) and blue in (e, k) indicate strong evidence for the target class. (Best viewed in color.)

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision*, *128*(2), 336–359. https://doi.org/10.1007/s11263-019-01228-7

Masked Autoencoder (MAE)

MAE = Self-Supervised Vision Learner

Learns by **reconstructing missing image patches**

Key idea: **Mask random patches (e.g., 75%)**, and reconstruct them from the visible ones

Architecture: **Asymmetric** encoder-decoder; **Encoder:** processes only **visible patches;**

Decoder: **lightweight**, reconstructs **masked** patches

MAE Training Workflow

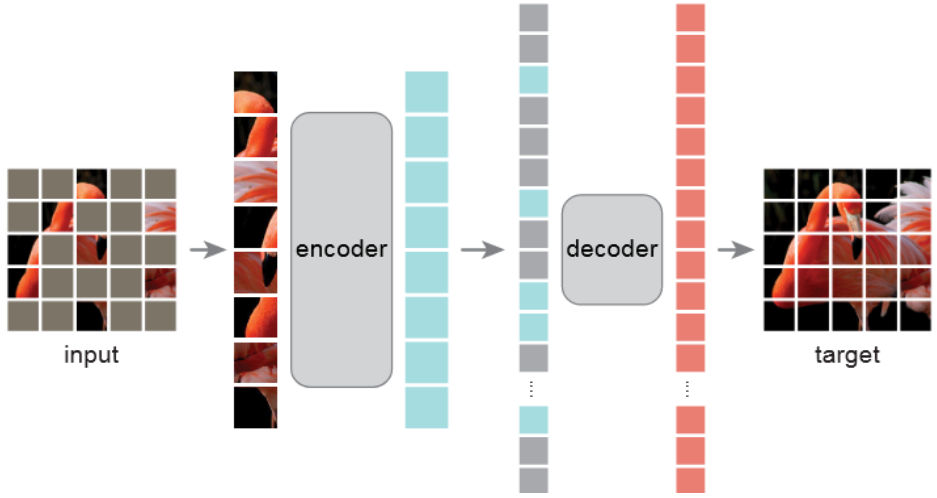
1. **Patchify** the input image into non-overlapping patches
2. **Randomly mask** 75% of the patches
3. **Encoder** processes only the remaining 25%
4. **Mask tokens** + visible patch embeddings go to the **decoder**
5. **Loss:** Mean squared error (MSE) on masked patches

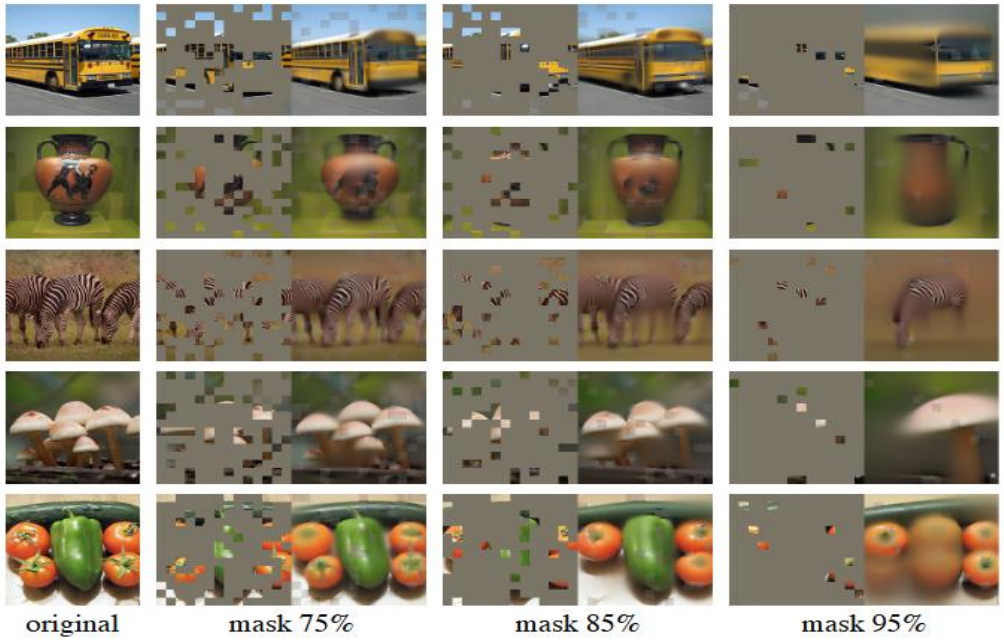
MAE Decoder ≈ U-Net Decoder. Both aim to **restore missing spatial resolution**

U-Net learns **semantic segmentation**, MAE learns **image structure**

Potential integration:

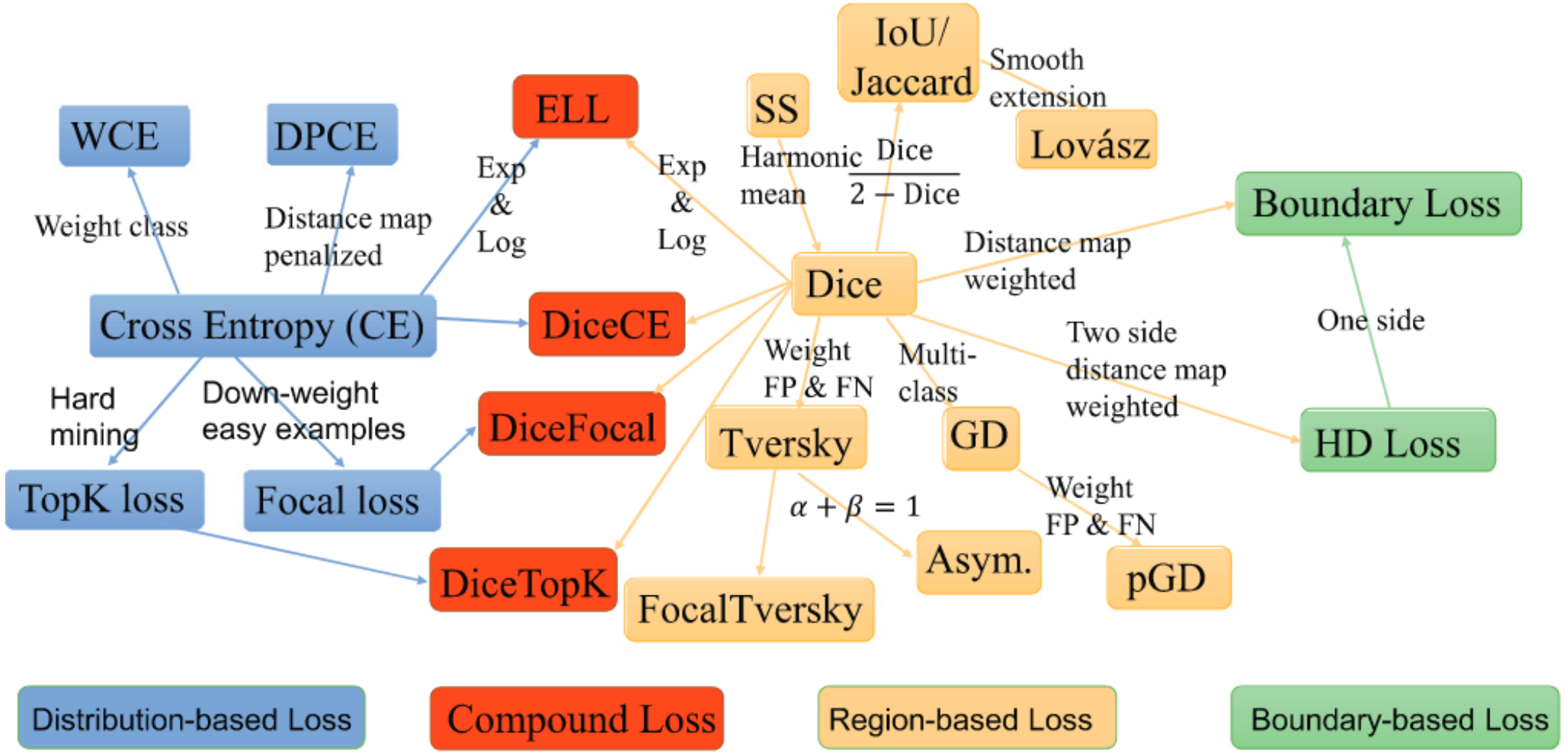
- Use **MAE pretraining** to initialize U-Net encoder (transfer learning)
- Apply MAE-style **reconstruction loss** to regularize segmentation learning





He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2021). *Masked Autoencoders Are Scalable Vision Learners* (No. arXiv:2111.06377). arXiv. https://doi.org/10.48550/arXiv.2111.06377

Overview of Loss Functions

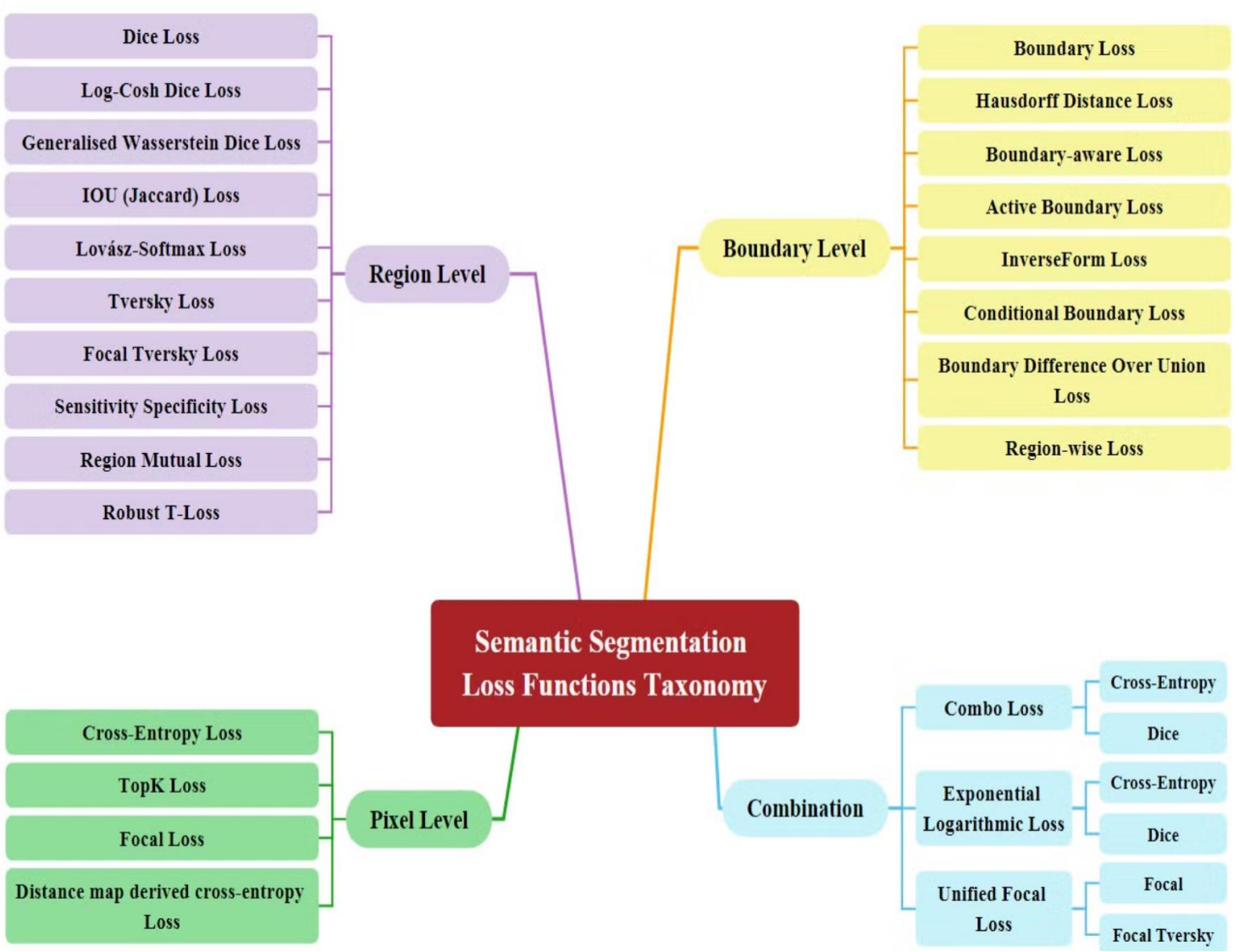


- Ma, J., Chen, J., Ng, M., Huang, R., Li, Y., Li, C., Yang, X., & Martel, A. L. (2021). Loss odyssey in medical image segmentation. *Medical Image Analysis*, *71*, 102035. https://doi.org/10.1016/j.media.2021.102035

Loss Functions: Taxonomy

Loss functions are categorized into four levels:

- **Pixel-Level:**
  Focus on per-pixel accuracy.
  Includes: Cross-Entropy, Focal, TopK, Distance-map Cross-Entropy.
- **Region-Level:**
  Capture global region consistency and handle class imbalance.
  Includes: Dice, IOU, Tversky, Lovász-Softmax, Region Mutual, Robust T-Loss.
- **Boundary-Level:**
  Emphasize precise boundary delineation.
  Includes: Boundary Loss, Hausdorff, Boundary-aware, InverseForm, Conditional Boundary Loss.
- **Combination:**
  Blend multiple losses for complementary benefits.
  Examples: Combo Loss (e.g., Cross-Entropy + Dice), Unified Focal Loss.



Pixel Level (Distribution-based)

Pixel-level loss functions dive deep into the individual pixels to achieve high accuracy in classifying each pixel within segmented regions. These loss functions compute the dissimilarity or error between the predicted pixel values and their corresponding ground truth labels independently for each pixel.

- **Cross-Entropy (CE) Loss** measures the difference between two probability distributions for a given random variable, measuring how well the model's predictions match the target labels.

- $L_{CE}(y,t) = -\sum_{n=1}^{N} \log(t_n \cdot y_n)$, when dealing with imbalanced datasets, one approach is to assign different ways to each class to help to balance the influence of each class on the overall loss and improve the performance of the model on the under-represented classes. Specifically, one may set the weight for each class to be inversely proportional to number of sample in that class.

$$L_{WCE}(y,t) = -\sum_{n=1}^{N} \mathrm{t_n} \cdot w\log(t_n \cdot y_n)$$

- **Focal Loss** is a modified version of the CE loss that assigns different weights to easy and hard samples. Here, hard samples are sample that are mis-classified with a high probability, while easy samples are those correctly classified with a high probability. $L_{focal}(y,t,\gamma) = -\sum_{n=1}^{N}(1 - t_n \cdot y_n)^{\gamma} \log(t_n \cdot y_n)$, where γ is a non-negative tunable hyperparameter. When γ is set to 0 for all samples, it reduces to the plain CE loss.

Symbol	Description
N	Number of pixels
C	Number of target classes
t_n	One-hot encoding vector representing the target class of the n^{th} pixel.
t_n^c	Binary indicator: 1 if the n^{th} pixel belongs to class c, otherwise 0.
y_n	Predicted class probabilities for n^{th} pixel.
y_n^c	Predicted probability of n^{th} pixel belonging to class c.
$t_n \cdot y_n$	Predicted probability for the target class of n^{th} pixel.
w	Weights assigned to target classes.

Region Level

- Region-level loss functions take a broader view in semantic segmentation tasks. Instead of focusing on each pixel, these methods prioritize the overall accuracy of object segmentation.

- **Dice Loss** originates from Dice Coefficient $\frac{2|Y \cap T|}{|Y|+|T|}$ where Y is the binary segmentation prediction mask and T is the binary segmentation target mask for a single class. It is computed separately for each target class and the average over all classes is used.

$$L_{dice} = 1 - \frac{1}{C}\sum_{c=0}^{C-1}\frac{2\sum_{n=1}^{N} t_n^c y_n^c}{\sum_{n=1}^{N}(t_n^c + y_n^c)}$$

- **IOU(Jaccard) Loss** originates from Jaccard index $\frac{|Y \cap T|}{|Y \cup T|}$, and is calculated for each class as well.

$$L_{IoU} = 1 - \frac{1}{C}\sum_{c=0}^{C-1}\frac{\sum_{n=1}^{N} t_n^c y_n^c}{\sum_{n=1}^{N}(t_n^c + y_n^c - t_n^c y_n^c)}$$

- **Tversky Loss** originates from the Tversky index $\frac{|Y \cap T|}{|Y \cap T| + \alpha|Y \backslash T| + \beta|Y \backslash T|}$ where α and β control the weights for false negatives and false positives. When $\alpha = \beta = 0.5$, it reduces to Dice coefficient. When $\alpha = \beta = 1$, it reduces to IoU.

$$L_{Tversky} = 1 - \frac{1}{C}\sum_{c=0}^{C-1}\frac{\sum_{n=1}^{N} t_n^c y_n^c}{\sum_{n=1}^{N}(t_n^c y_n^c + \alpha t_n^c(1 - y_n^c) + \beta y_n^c(t - t_n^c))}$$

Boundary Level

- Boundary-level loss functions specialize in the precision of object boundaries within the segmentation task. Their primary objective is to sharpen object boundaries and effectively separate overlapping objects.
- **Boundary Loss** aims to minimize the distance between ground truth and predicted segmentation. It computes the distance between two boundaries in the integral framework:

$$Dist(\partial G, \partial S) \approx 2\left(\int_{\Omega} \phi_G(q)s(q)dq - \int_{\Omega} \phi_G(q)g(q)dq\right)$$

$$L_B = \int_{\Omega} \phi_G(q)s(q)dq$$

where $g(q)$ is a binary indicator function that indicates whether q is on the boundary of ground truth; $\phi_G(q)$ represents the distance term and is defined as $\phi_G = -D_G(q)$ if $q \in G$ and $D_G(q)$ otherwise for the distance map of ground truth $D_G(q)$; and $s(q)$ denotes the probability predictions generated by the model.

- **Hausdorff Distance (HD) Loss** is based on Hausdorff distance $HD(X,Y) = \max(d_h(X,Y), d_h(Y,X))$ where $d_h(X,Y) = \max_{x \in X} \min_{y \in Y} d(x,y)$ and $d_h(Y,X) = \max_{y \in Y} \min_{x \in X} d(y,x)$, which is calculated between the boundaries of the predicted and ground truth masks. However, it solely depends on the largest error and is overly sensitive to outliers, leading to algorithm instability and unreliable results.

Combination

- By integrating multiple loss functions, a combination of them seeks equilibrium between pixel-wise precision, overall object segmentation quality and boundary delineation accuracy.
- **Combo Loss** is the most commonly used loss function in practice, combining Dice loss and weighted CE loss to overcome the class imbalance problem.

$$L_{combo} = \alpha L_{WCE} + (1 - \alpha) L_{dice}$$

- **Exponential Logarithmic Loss** is similar to the Combo loss in terms of combination. The difference is that it takes the logarithmic and exponential of both the loss functions before combining them, giving flexibility to control how much the model focuses on easy/hard pixels.

$$L_{Exp-Log} = \alpha L_{Exp-Log-WCE} + \beta L_{Exp-Log-Dice}$$

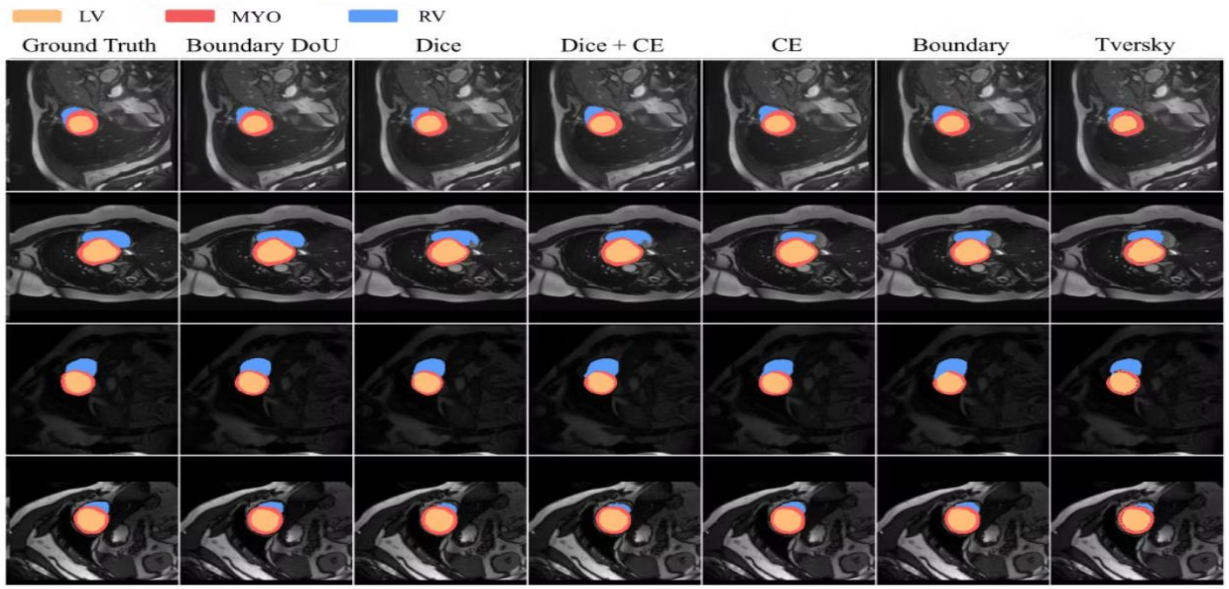
where $L_{Exp-Log-WCE} = \left(-\log(L_{WCE})\right)^{\gamma_{WCE}}$ and $L_{Exp-Log-Dice} = \left(-\log(L_{dice})\right)^{\gamma_{dice}}$, and $\gamma_{WCE}, \gamma_{dice}$ can be used to control the focus of the loss function, with $\gamma > 1$ focusing more on hard-to-classify pixels.

- **Dice Loss with Focal Loss** is used to alleviate the imbalanced organ segmentation problem and force the model to learn from poorly segmented voxels better.

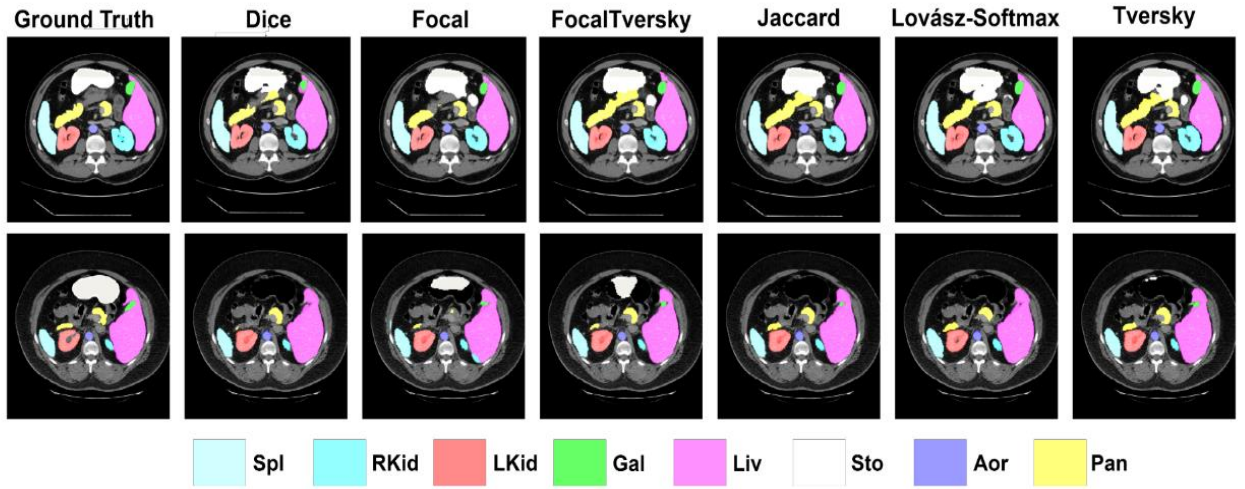
$$L_{Dice-Focal} = \alpha L_{dice} + (1 - \alpha) L_{focal}$$

Visual Comparison

On the ACDC dataset, we can observe a more accurate localization and segmentation for boundary regions in boundary-level loss functions. The boundary DoU loss function effectively address the challenge caused by the significant shape variations of the right ventricle region compared to the alternative loss functions.



On the Synapse dataset, Dice loss shows varying performance, performing quite good in the top example while completely failing to identify the stomach and gallbladder in the lower example. The Focal Tversky loss, on the other hand, presents the most promising segmentation map, correctly identifying all the organs.



Content

Foundation Models and Image Segmentation

- **Definition:** Foundation Models (FMs) are large-scale pre-trained models designed to adapt across diverse downstream tasks.
- **Paradigm Shift:**
  - Move from narrow task-specific models to **generic, task-agnostic systems**.
  - Enabled by advances in **neural networks**, **self-supervised learning**, and **scaling laws**.
- **Impact on Segmentation:**
  - FMs give rise to **segmentation generalists**, capable of handling a wide range of tasks.
  - They are **promptable**, similar to LLMs, allowing dynamic conditioning and flexible task specifications.
  - Support **zero-shot** and **few-shot** segmentation across domains without retraining.
- **Key Benefits:**
  - Unified models across modalities (e.g., CT, MRI, X-ray)
  - Adaptability to novel or underrepresented tasks
  - Reduced dependency on large annotated datasets
- **Conclusion:** FMs are transforming the landscape of medical image segmentation by introducing generalizable, interactive, and versatile frameworks.

R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill et al., "On the opportunities and risks of foundation models," arXiv preprint arXiv:2108.07258, 2021.

Foundation Models

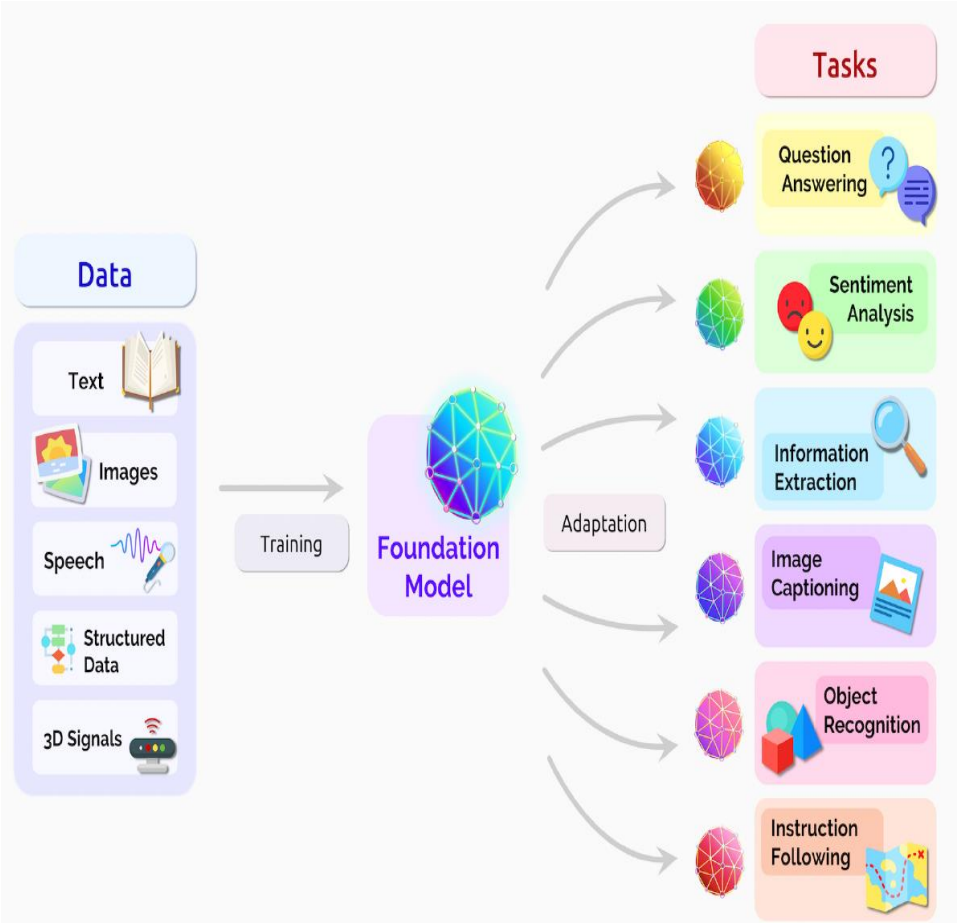
- **Foundation Models (FMs)** can be broadly categorized into:
  - **Language Foundation Models**
  - **Vision Foundation Models**

◆ Language Foundation Models

- **Large Language Models (LLMs):**
  - Core approach for machine language intelligence
  - Trained to predict the next token by modeling the generative likelihood of word sequences
  - Enable applications like text generation, summarization, and translation
- **Multimodal Large Language Models (MLLMs):**
  - Extend LLMs to incorporate non-textual inputs (e.g., audio)
  - Combine **language reasoning** with **vision/audio perception**
  - Facilitate complex tasks such as visual question answering, image captioning, and multimodal dialogue

◆ Vision Foundation Models

- Learn generic visual representations across a wide range of domains
- Examples: Vision Transformers (ViTs), SAM, CLIP, DINO
- Serve as backbones for tasks such as classification, detection, segmentation, and generation



A foundation model can centralize the information from all the data from various modalities.

R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill et al., "On the opportunities and risks of foundation models," arXiv preprint arXiv:2108.07258, 2021.

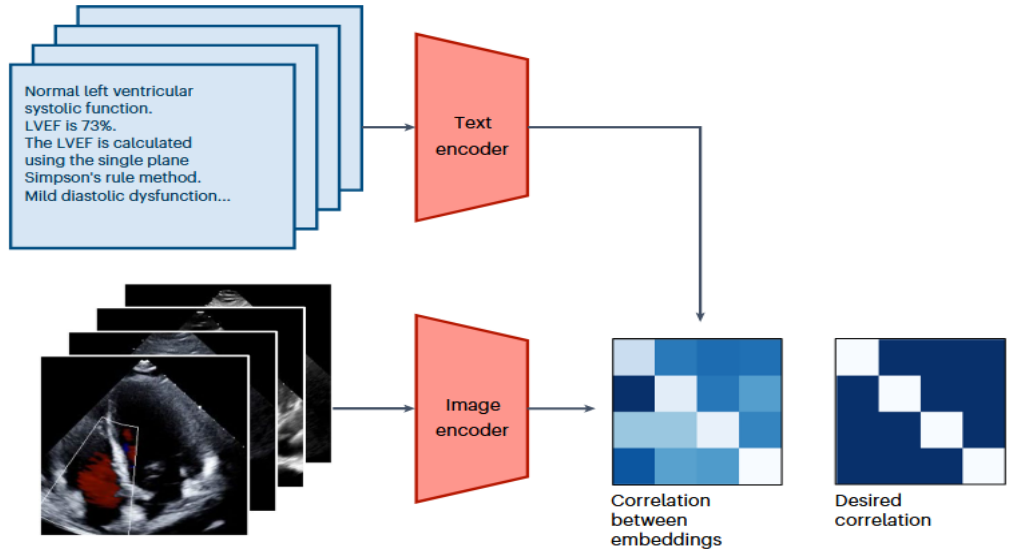
Categories of Vision Foundation Models

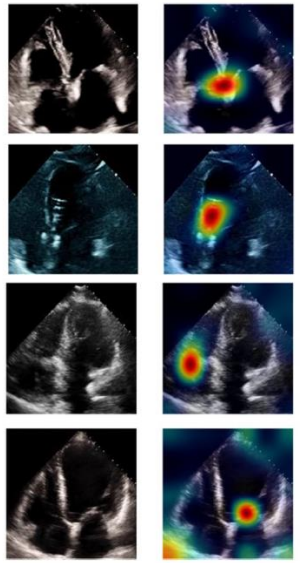
- **1. Contrastive Language-Image Pre-training (CLIP):**
  - Encoder-only architecture with separate image and text encoders
  - Learns by maximizing agreement between matched image-text pairs
  - Enables zero-shot classification and retrieval
- **2. Self-Distillation with No Labels (DINO):**
  - Self-supervised learning using Vision Transformers (ViTs)
  - Trains by aligning student-teacher models without labels
  - Captures strong visual representations despite compact size
- **3. Diffusion Models (DMs):**
  - Generative models trained via denoising and variational inference
  - Generate high-quality, realistic images from noise
  - Used in creative applications and medical synthesis
- **4. Segment Anything Model (SAM):**
  - General-purpose segmentation model
  - Enables promptable, zero-shot segmentation on diverse domains
  - A major advance in image registration and spatial understanding

R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill et al., "On the opportunities and risks of foundation models," arXiv preprint arXiv:2108.07258, 2021.

EchoCLIP

- Visual-Language Foundation Model for Echocardiogram (EchoCLIP) is trained on more than 1 million cardiac ultrasound videos and corresponding expert text.
- It is built on OpenCLIP and composed of an image encoder using ConvNeXt architecture for processing video frames and a text encoder using decoder-only Transformer for processing the corresponding physician interpretations. These two encoders project the images and interpretations onto a joint embedding space.
- A long-context variant using a custom tokenizer based on common echocardiography concepts is developed.
- EchoCLIP can be adapted to perform both classification and regression tasks.
  - For classification task, we can construct text prompts describing a positive case, obtain an embedding of those prompts using EchoCLIP's text encoder and compute the cosine similarity between them.
  - For a regression task, we can generate a collection of variations on a base text prompt by only changing the relevant value in the text. Then cosine similarity between the generated prompt embeddings and the embeddings of each of the first 20 frames of videos is then computed.





The figure demonstrates Grad-CAM visualizations over ultrasound images for identifying different cardiac implantable devices:

- **TAVR (Transcatheter Aortic Valve Replacement)**: Clearly highlighted central aortic valve region
- **Impella**: High-intensity focus along the catheter path
- **Pacemaker**: Lateral region marked, indicating lead presence
- **MitraClip**: Strong activation in the mitral valve area

Christensen, M., Vukadinovic, M., Yuan, N., & Ouyang, D. (2024). Vision–language foundation model for echocardiogram interpretation. *Nature Medicine*, *30*(5), 1481–1488. https://doi.org/10.1038/s41591-024-02959-y

Segment Anything (SAM)

- **SAM** has rapidly gained traction in medical image analysis.
- **Key Capability:**
  - Segments objects without prior knowledge of the object type or imaging modality
  - Mimics the flexibility of human visual perception
- **Promptable Interaction:**
  - Inspired by NLP, users can input prompts (points, bounding boxes)
  - Adjusts segmentation results based on resolution scale or area of interest
- **Zero-Shot and Few-Shot Learning:**
  - Requires little to no additional training to adapt to new segmentation tasks
  - Supports highly generalizable medical imaging workflows across modalities

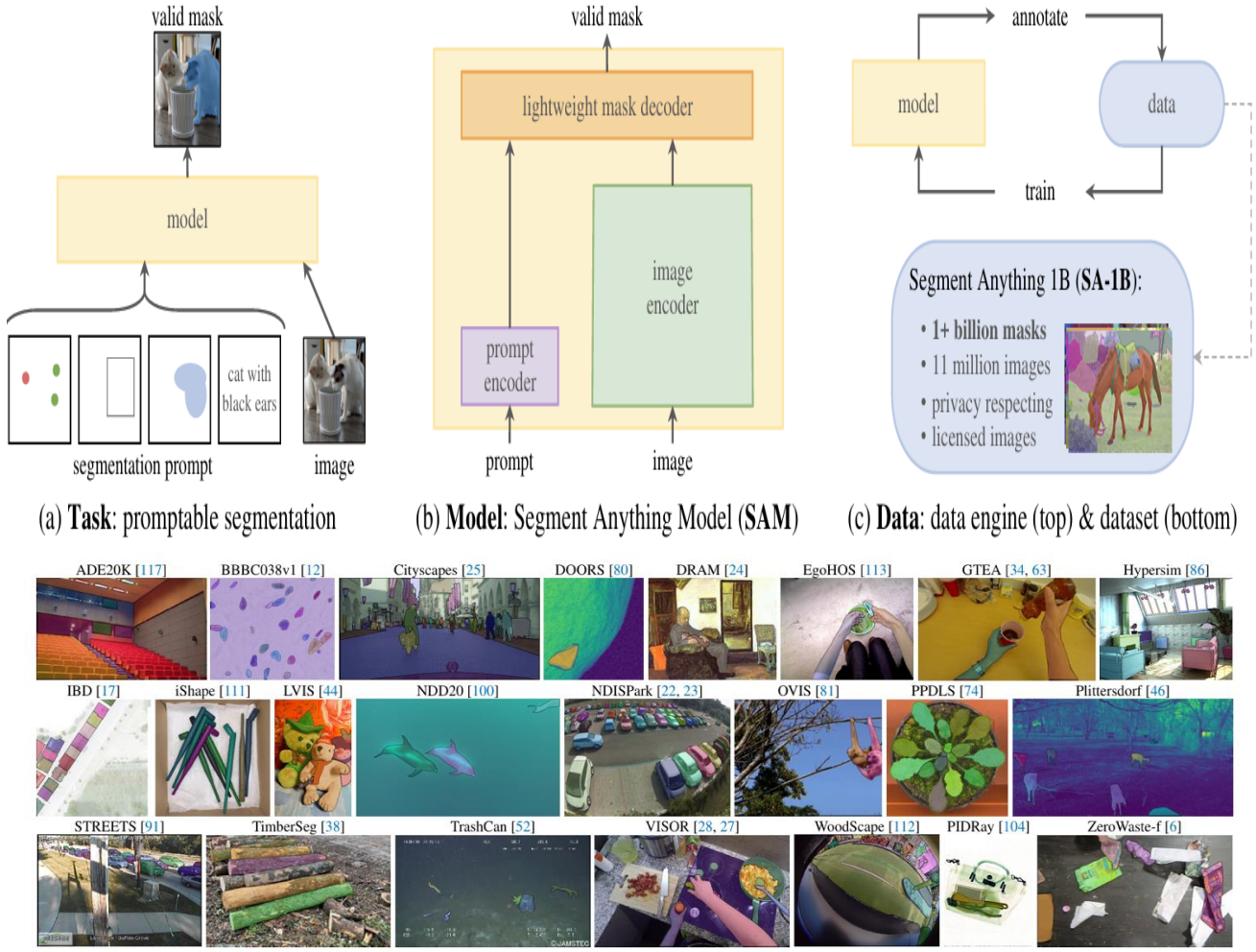
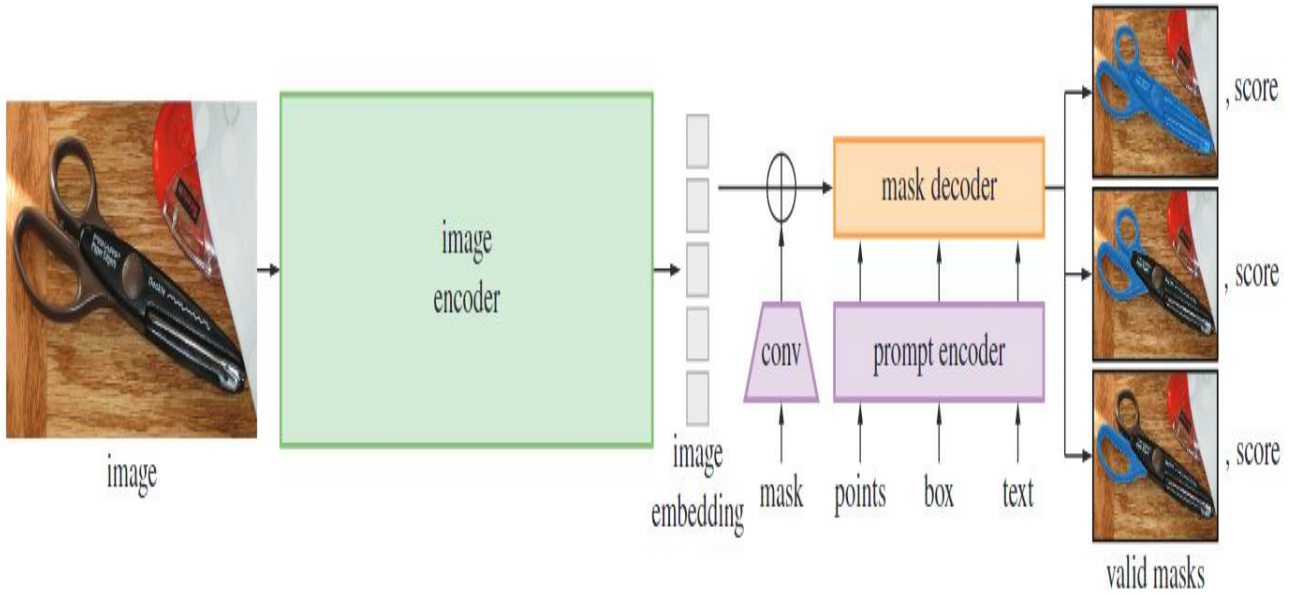


Figure 8: Samples from the 23 diverse segmentation datasets used to evaluate SAM's zero-shot transfer capabilities.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment Anything. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3992–4003. https://doi.org/10.1109/ICCV51070.2023.00371

SAM Architecture Overview

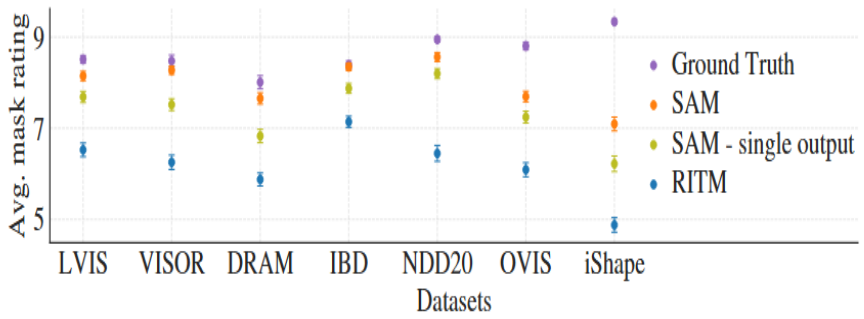
- **1. Image Encoder:**
  - Based on a **ViT pre-trained using Masked Autoencoders (MAE)**
  - Produces rich multi-scale image embeddings
- **2. Prompt Encoder:**
  - Handles **sparse prompts** (points, boxes) and **dense prompts** (masks)
  - Encodes spatial and semantic information from user input
- **3. Mask Decoder:**
  - A lightweight, efficient module
  - Combines image and prompt embeddings
  - Outputs accurate segmentation masks guided by prompts



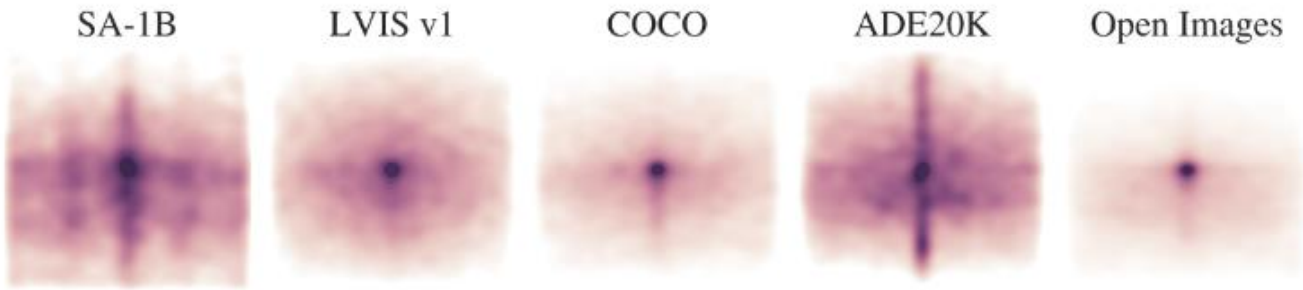
Segment Anything Model (SAM) overview. A heavyweight image encoder outputs an image embedding that can then be efficiently queried by a variety of input prompts to produce object masks at amortized real-time speed. For ambiguous prompts corresponding to more than one object, SAM can output multiple valid masks and associated confidence scores.

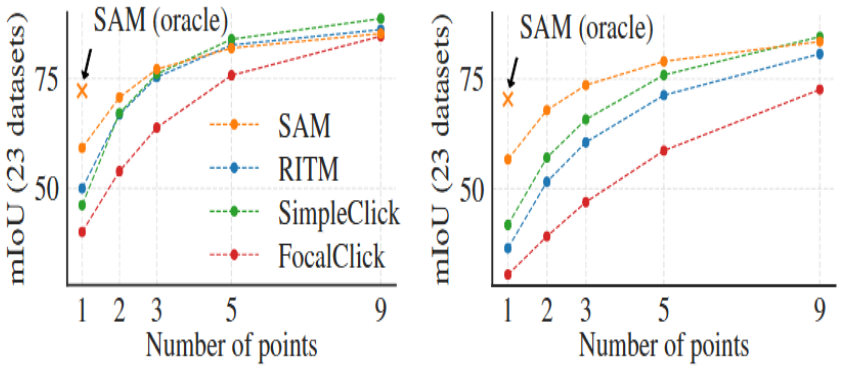
Segment Anything (SAM)

- SAM is trained on the large-scale dataset SA-1B, which consists of 11M high-resolution images with 1.1B high-quality segmentation masks, which is 400 times more masks than any existing segmentation dataset.
- The dataset has three stages: manual annotation stage, semi-automatic stage, and fully automatic stage.



(c) Mask quality ratings by human annotators





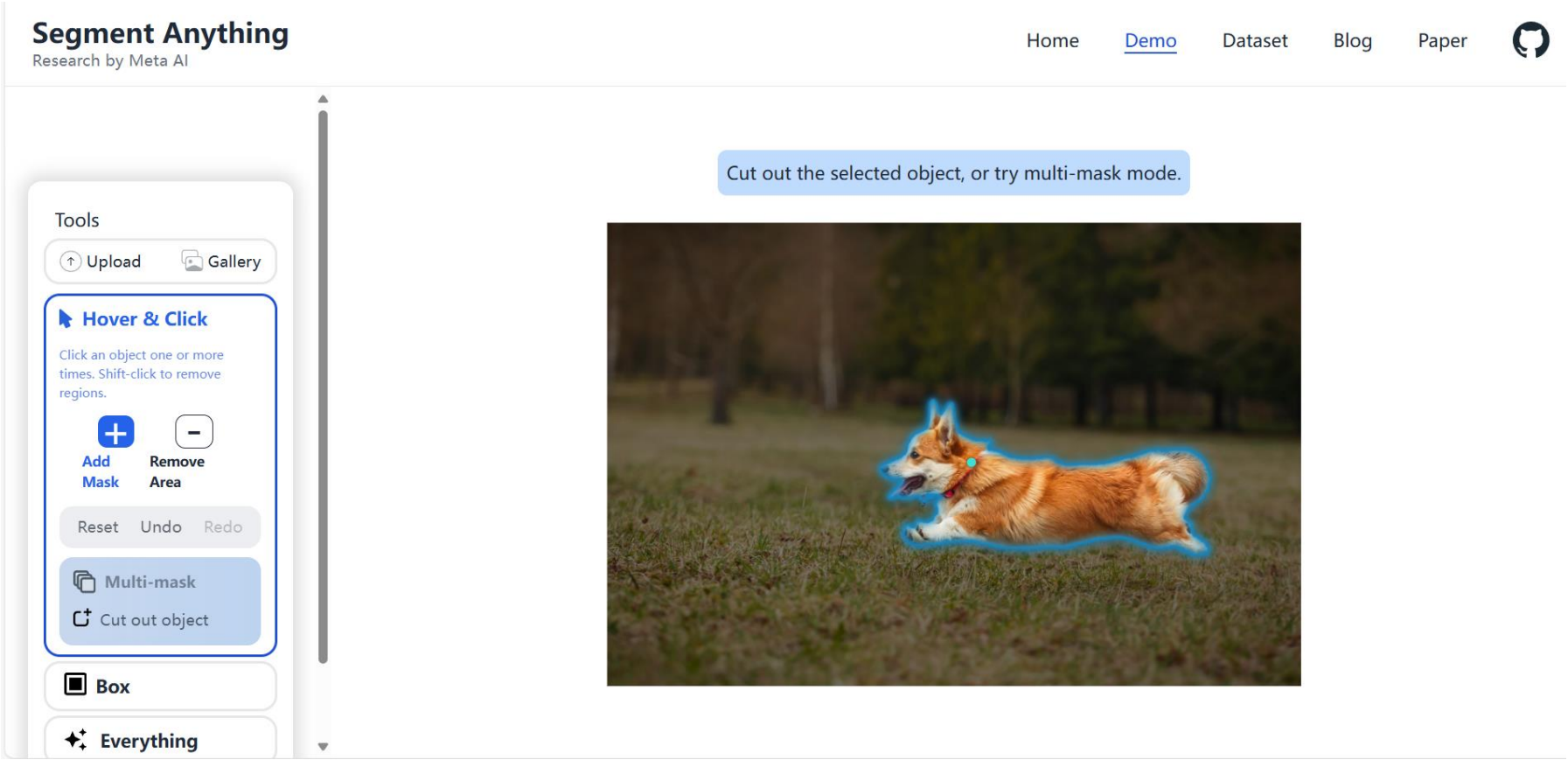
(d) Center points (default)

(e) Random points

- In zero-shot single point valid mask evaluation, annotators consistently rate the quality of SAM's masks substantially higher than the strongest baseline, RITM.
- As number of points increases from 1 to 9, the gap between methods decreases, as the task of segmentation becomes easier.

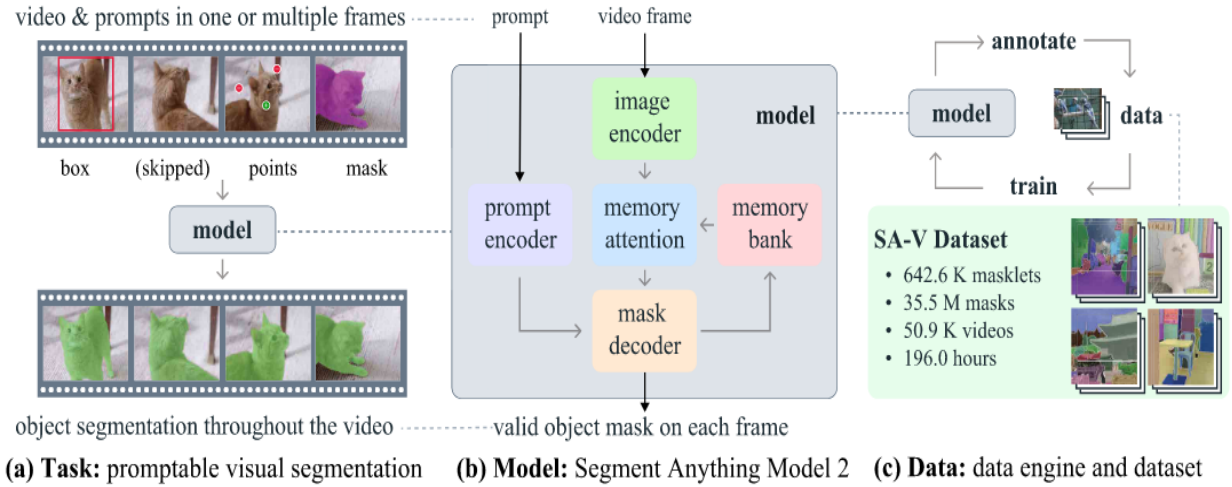
SAM: Play Around with it Online!

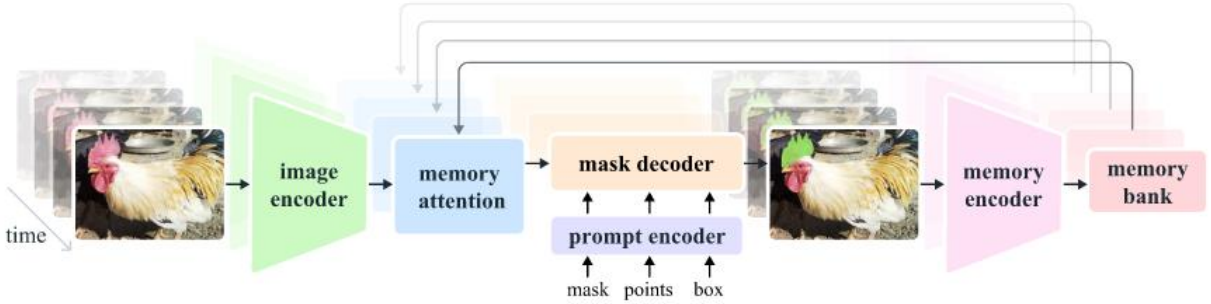
[Segment Anything | Meta AI (segment-anything.com)]



Segment Anything 2 (SAM 2)

- However, SAM only works for images. In order to be applicable to both images and videos, Segment Anything 2 is proposed as a unified model for video and image segmentation, considering image as a single-frame video.
- It is a natural generalization of SAM to the video domain, processing video frames one at a time, equipped with a memory attention module to attend to the previous memories of the target object. When applied to images, the memory is empty and the model behaves like SAM.
- A geographically diverse dataset SA-V is constructed, consisting of 35.5M masks across 50.9K videos, 53 times more masks than any existing video segmentation dataset.
- SAM 2 behaves spatially similar to SAM. However, the frame embedding used by it is not directly from an image encoder, but instead conditioned on memories of past predictions and prompted frames.
- SAM 2 can produce better segmentation accuracy while using 3 times fewer interactions than prior approaches, and deliver better performance compared to SAM on image segmentation benchmarks, while being 6 times faster.

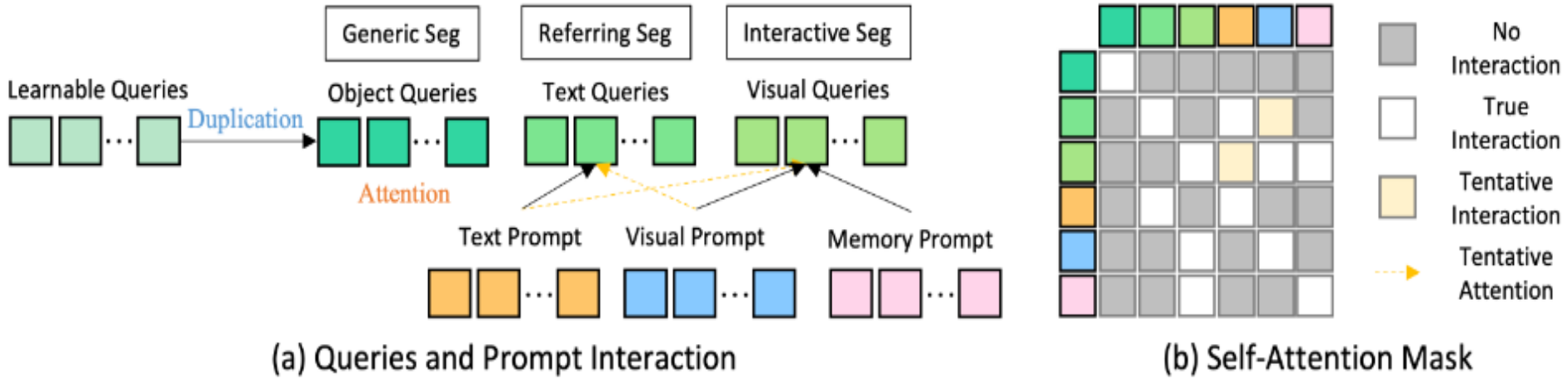




Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., & Feichtenhofer, C. (2024). *SAM 2: Segment Anything in Images and Videos* (No. arXiv:2408.00714). arXiv. https://doi.org/10.48550/arXiv.2408.00714

SEEM

- Although SAM demonstrates strong zero-shot performance, it produces segmentations without semantic meaning. In addition, its prompt types are limited to points, boxes and text.
- Segment Everything Everywhere All at Once (SEEM) proposes a novel decoding mechanism that enables diverse prompting including a referred region from another image, aiming at a universal segmentation interface that behaves like LLMs.
- It not only employs a generic encoder-decoder architecture, but also employs a sophisticated interaction scheme between queries and prompts.
- Given an image $I \in R^{H \times W \times 3}$, an image encoder is first used to extract image features Z. Then based on the text, visual and memory prompts $\langle P_t, P_v, P_m \rangle$, the decoder guides the learnable queries Q_h to predict the mask embeddings O_h^m and class embeddings O_h^c to generate masks M and semantic concepts C.



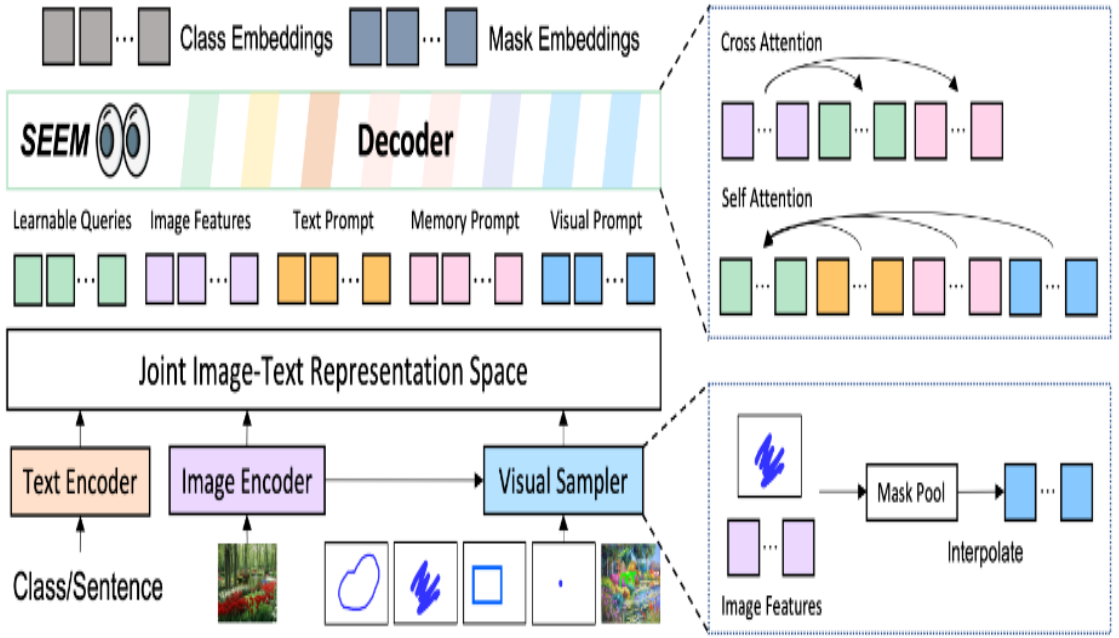
Zou, X., Yang, J., Zhang, H., Li, F., Li, L., Wang, J., Wang, L., Gao, J., & Lee, Y. J. (2023). *Segment Everything Everywhere All at Once* (No. arXiv:2304.06718). arXiv. https://doi.org/10.48550/arXiv.2304.06718

SEEM

Algorithm 1: Pseudo code for SEEM.

```
# Inputs: Image(img)[B,3,H,W]; Pos_Mask(pm), Neg_Mask(nm)[B,1,H,W]; Text(txt)[abc...];
# Variables: Learnable Queries(Q_h); Attention Masks between Q and P(qpm)
# Functions: Img_Encoder(),Text_Encoder(),Visual_Sampler(),feature_attn(),prompt_attn(),output();

def init( ):
    Q_o,Q_t,Q_v = Q_h.copy();# Initialize object, text and visual queries.
    F_v,P_t = Img_Encoder(img), Text_Encoder(txt);# F_v and P_t denote image feature, text prompt.
    P_v = Visual_Sampler(F_v, pm, nm);# Sample visual prompt from image feature, pos/neg mask.
def SEEM_Decoder(F_v,Q_o,Q_t,Q_v,P_v,P_t,P_m):
    Q_o,Q_t,Q_v = feature_attn(F_v,Q_o,Q_t,Q_v);# Cross attend queries with image features.
    Q_o,Q_t,Q_v = prompt_attn(qpm,Q_o,Q_t,Q_v,P_v,P_t,P_m);# Self attend queries and prompts.
    O_m,O_c,P_m = output(F_v,Q_o,Q_t,Q_v);# Compute mask and class outputs.
def forward(img,pm,nm,txt):
    F_v,Q_o,Q_t,Q_v,P_v,P_t = init(); P_m = None;# Initialize variables.
    for i in range(max_iter):
        O_m,O_c,P_m = SEEM_Decoder(F_v,Q_o,Q_t,Q_v,P_v,P_t,P_m)
```

$$\mathcal{L} = \alpha\mathcal{L}_{\text{c\_CE\_pano}} + \beta\mathcal{L}_{\text{m\_BCE\_pano}} + \gamma\mathcal{L}_{\text{m\_DICE\_pano}} + a\mathcal{L}_{\text{c\_CE\_ref}} + b\mathcal{L}_{\text{m\_BCE\_ref}} + c\mathcal{L}_{\text{m\_DICE\_ref}} + a\mathcal{L}_{\text{c\_CE\_iseg}} + b\mathcal{L}_{\text{m\_BCE\_iseg}} + c\mathcal{L}_{\text{m\_DICE\_iseg}}$$



Compare with other strong baselines SimpleClick and SAM with 5 common types of prompts, the SEEM achieves the best performance in the extremely limited number of clicks over all three datasets.

Method	COCO					Open Image					ADE				
	Point 1-IoU	Stroke 1-IoU	Scribble 1-IoU	Polygon 1-IoU	Box 1-IoU	Point 1-IoU	Stroke 1-IoU	Scribble 1-IoU	Polygon 1-IoU	BoX 1-IoU	Point 1-IoU	Stroke 1-IoU	Scribble 1-IoU	Polygon 1-IoU	BoX 1-IoU
SimpleClick (B)	49.0	33.1	65.1	48.6	42.5	48.6	29.5	54.2	49.5	42.7	47.0	19.0	52.1	48.3	37.2
SimpleClick (L)	38.9	33.9	68.8	39.2	34.7	37.5	29.1	59.8	35.2	31.2	36.8	16.4	56.4	41.7	29.5
SimpleClick (H)	59.0	37.3	71.5	45.3	52.4	54.1	32.6	64.7	39.9	49.3	52.8	18.4	58.3	46.8	41.8
SAM (B)	58.6	22.8	34.2	44.5	50.7	62.3	28.4	39.2	45.8	53.6	51.0	21.9	31.1	31.0	**58.8**
SAM (L)	64.7	44.4	57.1	60.7	50.9	65.3	45.9	55.7	57.8	52.4	57.4	45.8	53.1	45.8	58.7
SAM (H)	65.0	27.7	30.6	37.8	50.4	67.7	26.5	29.9	41.9	52.1	58.4	20.4	22.2	28.3	58.5
SEEM (T)	78.9	81.0	81.2	72.2	73.7	67.1	**69.4**	**69.5**	63.1	**60.9**	65.4	67.3	67.3	59.0	53.4
SEEM (B)	81.7	82.8	83.5	76.0	75.7	**67.6**	69.0	68.7	**64.2**	60.3	**66.4**	**68.6**	**67.7**	**60.5**	53.6
SEEM (L)	**83.4**	**84.6**	**84.1**	**76.5**	**76.9**	66.8	67.8	67.6	62.4	60.1	65.5	66.6	66.3	58.1	54.1

FMs for Biomedical Image Segmentation

- While **SAM** achieves or approaches state-of-the-art performance in many general vision tasks, its performance in medical image segmentation presents certain limitations:

Application Gaps:

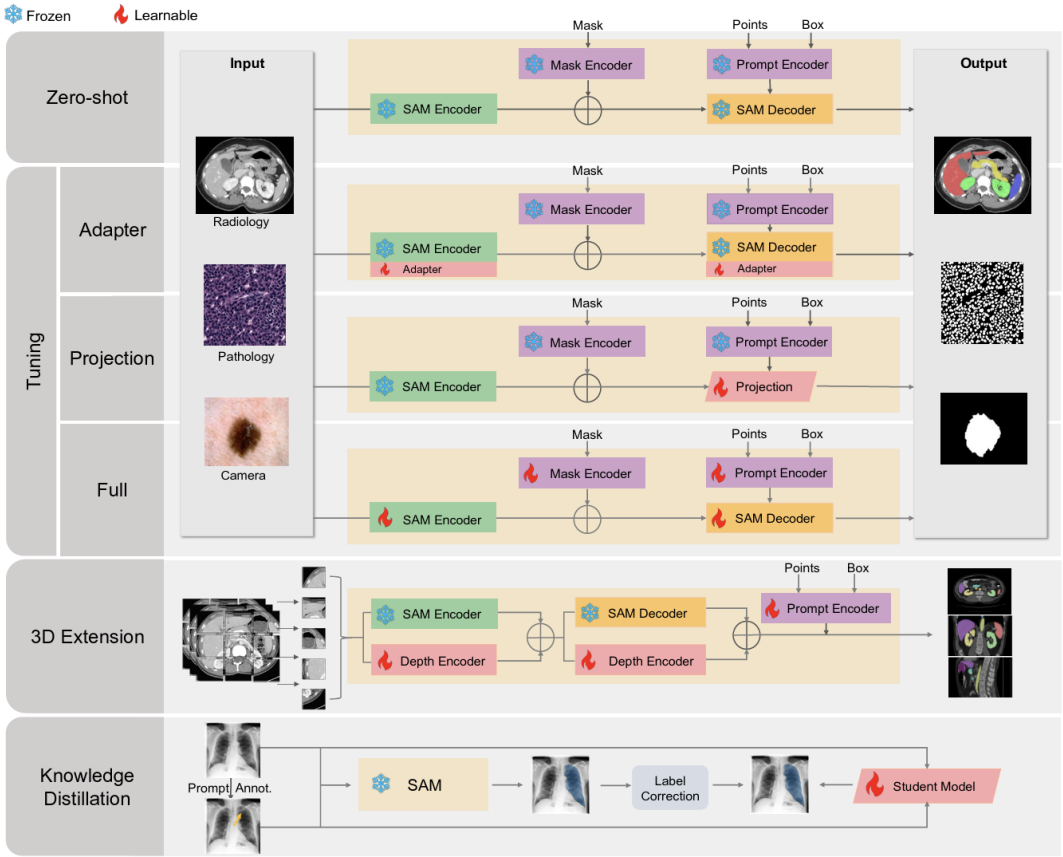
- **Difficult Anatomical Structures:**
  - Struggles with segmentation of small or complex organs such as the **carotid artery**, **adrenal glands**, **optic nerve**, and **mandible bone**.

Key Challenges:

- **Data Specificity:**
  - SAM excels in general-domain images, but medical images often exhibit features and artifacts not typically present in everyday visual data.
- **Dimension Mismatch:**
  - Most medical imaging modalities (e.g., **MRI**, **CT**) produce **volumetric (3D) data**, while SAM is inherently 2D-based.
  - Lack of native support for 3D spatial context limits its performance in full-volume analysis.
- **Data Scarcity and Annotation Quality:**
  - High-quality annotations in medicine require expert knowledge, are time-consuming to produce, and face **privacy constraints**.
  - Limits the availability of large-scale datasets to fine-tune or evaluate SAM reliably for clinical applications.

FMs for Biomedical Image Segmentation

- Multiple method have been proposed for the adaption of SAM to the medical domain.
- **Zero-shot segmentation capabilities evaluation:** Medical imaging presents unique challenges, distinguished by factors like varied imaging protocols and a wider range of patient demographics. These complexities are not as predominant in standard domain images, making SAM's adaptability in this context particularly intriguing.
- **Domain-specific tuning:** To address the varying results across different contrast appearances and organ morphologies, researchers have explored several domain-specific tuning strategies:
  - **Projection tuning:** Replacing the pretrained decoder with a new, task-specific projection head, aiming to harness generalized features
  - **Adapter tuning:** Incorporating adapters designed to fine-tune the model's response to the specific challenges presented by medical imaging.
  - **Full tuning:** A substantial reconfiguration, finetuning both the encoder and decoder of SAM to transition its generalized knowledge base.

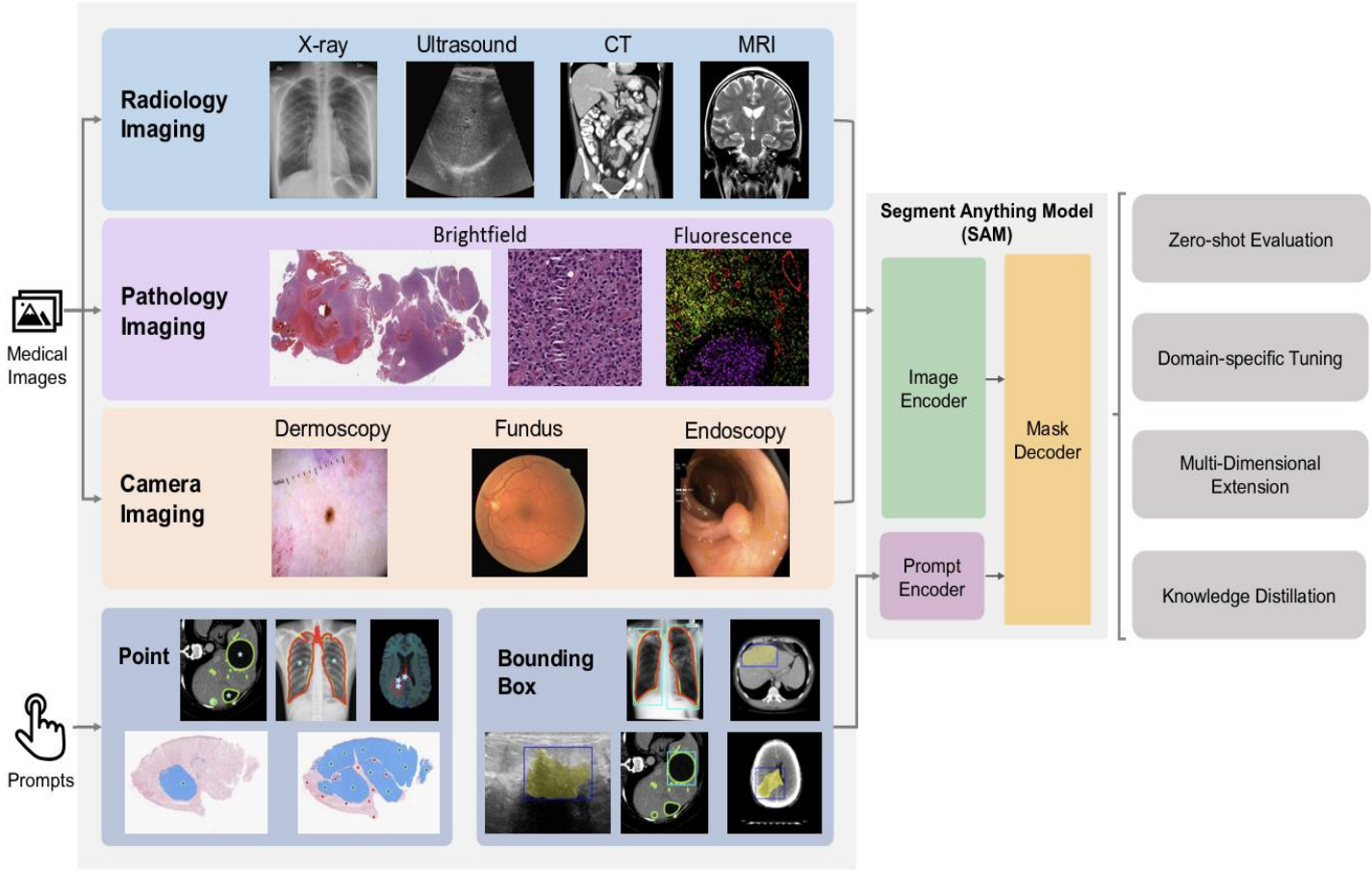


SAM adaptation in medical imaging includes **zero-shot evaluation, varying degrees of model fine-tuning (adapter, projection, full tuning), 3D extension for volumetric data, and knowledge distillation** to transfer expertise to lighter models, each enhancing domain-specific performance through tailored pipelines.

Lee, H. H., Gu, Y., Zhao, T., Xu, Y., Yang, J., Usuyama, N., Wong, C., Wei, M., Landman, B. A., Huo, Y., Santamaria-Pang, A., & Poon, H. (2024). *Foundation Models for Biomedical Image Segmentation: A Survey* (No. arXiv:2401.07654). arXiv. http://arxiv.org/abs/2401.07654

FMs for Biomedical Image Segmentation

- **3D Imaging Modalities Extension:**
  - To align with SAM's 2D framework, 3D images are processed as axial slices for slice-by-slice predictions.
  - All slice-level outputs are then fused into a comprehensive volumetric map to restore spatial continuity and capture 3D anatomical relationships.
- **Knowledge Distillation:**
  - A **label refinement network** improves the coarse masks generated by SAM.
  - These refined annotations are used to train a task-specific **student model** for enhanced segmentation accuracy tailored to medical tasks.



Application of SAM Across Medical Imaging Modalities. The figure showcases Radiology, Pathology, and Camera Imaging examples. Central components of SAM, including the Image Encoder, Mask Decoder, and Prompt Encoder, are delineated. Methods ranging from Zero-shot Evaluation to Knowledge Distillation are accentuated within tan boxes.

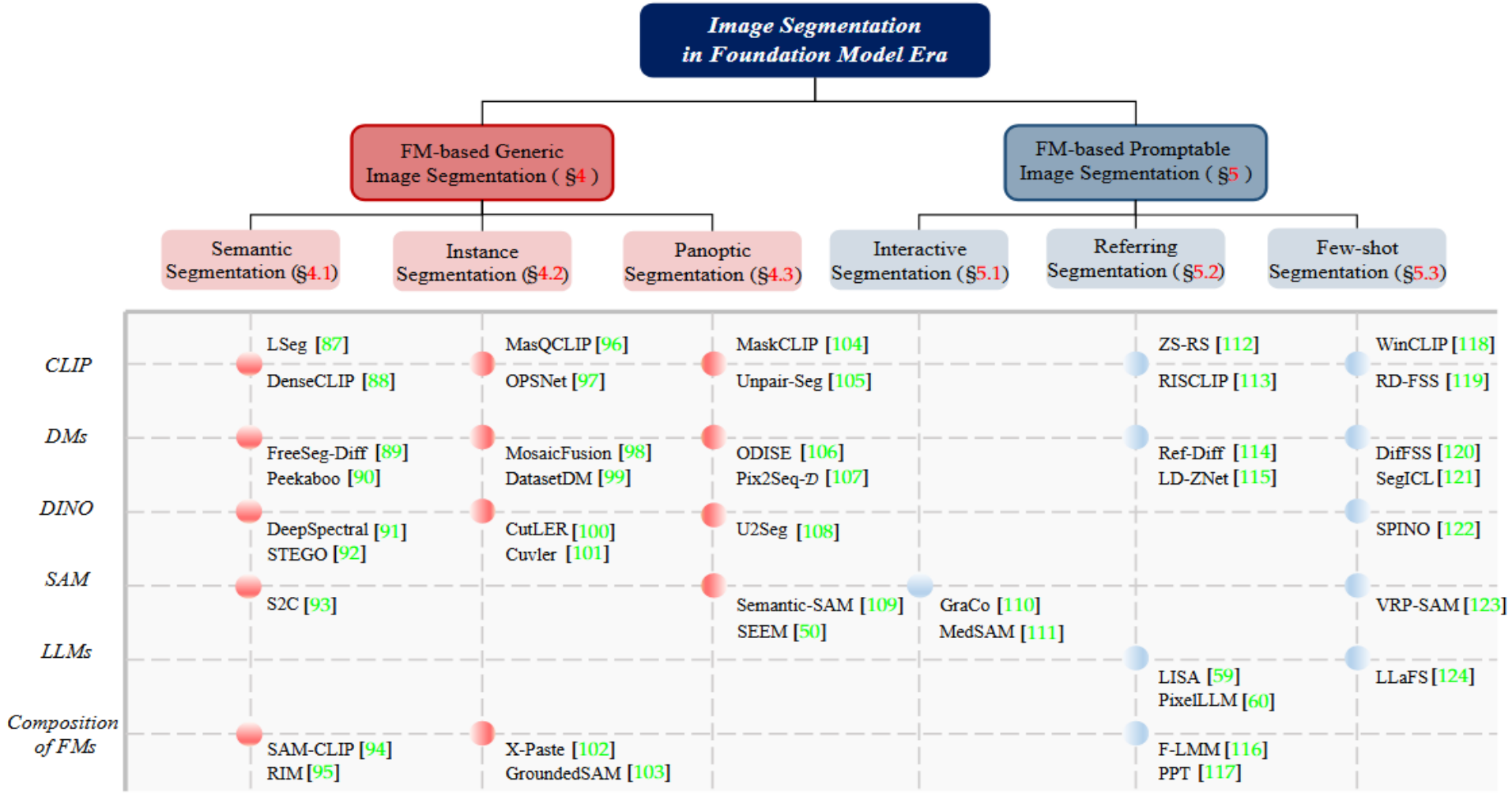
Lee, H. H., Gu, Y., Zhao, T., Xu, Y., Yang, J., Usuyama, N., Wong, C., Wei, M., Landman, B. A., Huo, Y., Santamaria-Pang, A., & Poon, H. (2024). *Foundation Models for Biomedical Image Segmentation: A Survey* (No. arXiv:2401.07654). arXiv. http://arxiv.org/abs/2401.07654

Chronological Timeline of Medical IS Datasets

Year	Dataset	Public	Details					
			Modality	Anatomy	Data Size	Label Quality	# Targets	Seg. Target Type
1998	JSRT [134]	✓	X-Ray	Chest	307	Manual	2	Multi-Organ
2012	VESSEL12 [124]	✓	CT	Lung	20	Manual	1	Organ Parts
2012	PROMISE12 [96]	✓	MRI	Prostate	100	Manual	1	Single Organ
2013	NCI-ISBI [21]	✓	MRI	Prostate	80	Manual	2	Organ Parts
2015	BTCV [82]	✓	CT	Abdomen	50	Manual	13	Multi-Organ
2015	CT-Lymph Nodes [34, 119, 122]	✓	CT	Mediastinum	176	Manual	1	Single Organ
2015	GlaS [135, 136]	✓	Pathology	Colon	165	Manual	1	Cells
2016	Pancreas-CT [34, 120, 121]	✓	CT	Pancreas	80	Manual	1	Single Organ
2017	LiTS [18]	✓	CT	Liver	131	Manual	2	Tumor
2017	ACDC [17]	✓	MRI	Heart	150	Manual	3	Organ Parts
2018	FUMPE [103]	✓	CT	Lung	35	Exp.+Mdl.	1	Lesion
2018	MSD [10]	✓	CT, MRI	Multiple	1411 CT, 1222 MRI	Manual	18	Multi-Task
2018	DRIVE [138]	✓	Fundus	Retina	40	Manual	1	Organ Parts
2018	REFUGE [111]	✓	Fundus	Retina	1200	Manual	2	Organ Parts
2019	CHAOS [74–76]	✓	CT, MRI	Abdomen	40 CT, 40 MRI	Manual	4	Multi-Organ
2019	SIIM-ACR Pneumothorax [160]	✓	X-Ray	Chest	12047	Manual	1	Lesion
2019	AbdomenUS [146]	✓	Ultrasound	Abdomen	61 Real, 926 Synth.	Real+Synth.	8	Multi-Organ
2019	Breast Ultrasound Images [4]	✓	Ultrasound	Breast	780	Manual	3	Tumor
2019	CAMUS [83]	✓	Ultrasound	Heart	500	Manual	3	Organ Parts
2020	M&Ms [24]	✓	MRI	Heart	375	Manual	3	Organ Parts
2020	MosMed COVID-19 [109]	✓	CT	Lung	50	Manual	1	Infection
2020	COVID-19 Radiography [32, 116]	✓	X-Ray	Chest	21165	Manual	1	Single Organ
2021	COVID-QU-Ex [32, 37, 116, 141, 142]	✓	X-Ray	Chest	33920	Manual	2	Infection
2021	QaTa-COV19 [38]	✓	X-Ray	Chest	9258	Manual	1	Infection
2021	CT2US [137]	✓	Ultrasound	Abdomen	4586	Synth.	1	Single Organ
2021	PolypGen [5–7]	✓	Endoscope	Colon	8037	Manual	1	Polyp
2022	AbdomenCT-1K [102]	✓	CT	Abdomen	1112	Exp.+Mdl.	4	Multi-Organ
2022	AMOS [72]	✓	CT, MRI	Abdomen	500 CT, 100 MRI	Exp.+Mdl.	15	Multi-Organ
2023	KiTS [57]	✓	CT	Kidney	599	Exp.+Mdl.	3	Organ, Tumor
2023	TotalSegmentator [153]	✓	CT	Full Body	1228	Manual	117	Multi-Organ
2023	BraTS [2, 13–16, 73, 77, 81, 106, 107]	✓	MRI	Brain	4500	Manual	3	Tumor
2023	HaN-Seg [113]	✓	CT, MRI	Head & Neck	56 CT, 56 MRI	Manual	30	Multi-Organ
2023	FH-PS-AOP [100]	✓	Ultrasound	Transperineal	6224	Exp.+Mdl.	2	Multi-Organ

Lee, H. H., Gu, Y., Zhao, T., Xu, Y., Yang, J., Usuyama, N., Wong, C., Wei, M., Landman, B. A., Huo, Y., Santamaria-Pang, A., & Poon, H. (2024). *Foundation Models for Biomedical Image Segmentation: A Survey* (No. arXiv:2401.07654). arXiv. http://arxiv.org/abs/2401.07654

Taxonomy based on IS Tasks and FMs



- Zhou, T., Zhang, F., Chang, B., Wang, W., Yuan, Y., Konukoglu, E., & Cremers, D. (2024). *Image Segmentation in Foundation Model Era: A Survey* (No. arXiv:2408.12957). arXiv. http://arxiv.org/abs/2408.12957

Various Existing Works Build Upon SAM

Year-Month	Method \ Tasks	2D	3D	A.P.P	P.A	E.Frozen	E.Finetune	R.N.M	T.P.H	T.P.E	T.A	Downstream Tasks
2023-April	SAM-Adaptor [28]	✔	-	-	-	✔	-	-	-	-	✔	Polyp
2023-April	SAMAug [166]	✔	-	-	✔	✔	-	✔	-	-	-	H&E, Polyp
2023-April	MedSAM Adaptor [154]	✔	✔	-	-	-	-	✔	-	-	✔	Abd, Opt, B.T, T.N
2023-April	LOSAM [166]	✔	-	-	-	✔	-	-	-	✔	-	Vessel & Lesion
2023-April	SAMed [162]	✔	-	-	-	✔	-	-	✔	✔	✔	Abd
2023-April	GazeSAM [148]	✔	-	✔	-	✔	-	-	-	-	-	Abd
2023-April	SkinSAM [63]	✔	-	-	✔	-	✔	-	-	-	-	S.L
2023-April	PiClick [158]	✔	-	-	-	✔	-	-	-	-	-	Neural Tissue
2023-May	Polyp-SAM [94]	✔	-	-	-	-	✔	-	-	✔	-	Polyp
2023-May	SAM-Track [31]	✔	-	-	✔	✔	-	-	-	-	-	Brain
2023-May	WS-SAM [54]	✔	-	✔	-	✔	-	✔	-	-	-	Polyp
2023-May	BreastSAM [62]	✔	-	-	✔	✔	-	-	-	-	-	Breast C.
2023-May	LuSAM [69]	✔	-	-	-	✔	-	-	-	-	-	Lung
2023-May	IAMSAM [84]	✔	-	-	-	✔	-	-	-	-	-	H & E
2023-June	DeSAM [46]	✔	-	-	-	✔	-	-	-	✔	✔	Prostate
2023-June	AutoSAM(1) [128]	✔	-	-	-	✔	-	-	-	✔	-	H & E, Polyp
2023-June	TEPO [129]	✔	-	✔	-	✔	-	✔	-	-	-	Brain
2023-June	RASAM [163]	✔	-	-	-	✔	-	-	-	-	-	Organ-at-risk
2023-June	3DSAM-adaptor [48]	-	✔	-	✔	✔	-	-	-	✔	✔	Parts Tumor
2023-June	AutoSAM(2) [64]	✔	-	-	-	✔	-	-	✔	-	-	Cardiac Structure
2023-June	MedLSAM [89]	✔	-	✔	-	✔	-	✔	-	-	-	H & N, Abd, Lung
2023-June	CellViT [59]	✔	-	-	-	-	✔	-	✔	-	-	H & E
2023-July	SAM-U [39]	✔	-	-	✔	✔	-	-	-	-	-	Opt
2023-July	SAM^{Med} [149]	✔	-	✔	✔	✔	-	-	-	-	✔	Abd, Prostate
2023-July	SAMAug [36]	✔	-	-	✔	✔	-	-	-	-	-	Polyp, Lung
2023-July	All-in-SAM [35]	✔	-	✔	✔	✔	-	-	✔	✔	-	H & E
2023-July	SAM-Path [161]	✔	-	-	✔	✔	-	-	✔	✔	-	H & E
2023-July	CmAA [132]	✔	-	-	-	✔	-	-	✔	-	-	Glioma
2023-July	MedSAM [101]	✔	-	-	-	-	-	✔	-	-	-	15 I.M, >30 C.T
2023-August	SAM-MLC [66]	✔	-	✔	-	✔	-	✔	-	-	-	Lung
2023-August	AdaptiveSAM [112]	✔	-	-	✔	-	✔	-	✔	✔	-	S.S
2023-August	Poly-SAM++ [20]	✔	-	-	✔	✔	-	-	-	-	-	Polyp
2023-August	SPSAM [155]	✔	-	✔	✔	✔	-	-	-	✔	-	Polyp, S.L
2023-August	SamDSK [167]	✔	-	-	-	✔	-	✔	-	-	-	Polyp, S.L, Breast C.
2023-August	AutoSAM Adaptor [90]	-	✔	-	✔	✔	-	-	-	✔	✔	Abd
2023-August	SAM-Med2D [30]	✔	-	-	-	✔	-	-	-	✔	✔	9 MICCAI2023
2023-August	SAMedOCT [43]	✔	-	-	-	✔	-	✔	-	-	-	OCT
2023-September	SAM3D [23]	✔	-	-	-	✔	-	-	✔	-	-	Brain Lung,, Abd
2023-September	SAMUS [95]	✔	-	-	-	✔	-	✔	-	-	-	Ultrasound
2023-September	MA-SAM [25]	-	✔	-	-	✔	-	-	-	-	✔	Abd, Prostate, S.S
2023-September	MedVISTA-SAM [25]	✔	✔	-	-	✔	-	-	-	-	✔	Echocardiography

<Acronym: Meaning> **A.P.P**:Adapt Psuedo Prior; **PA**:Prompt Augmentation; **E.**:Encoder; **R.N.M**: Retrain New Model; **T.P.H**:Train Projection Head; **T.P.E**:Train Prompt Encoder; **T.A**:Train Adaptor; **Abd**:Abdomen; **Opt**:Optic; **B.T**:Brain Tumor; **T.N**:Thyroid Nodule; **I.M**: Imaging Modalities; **C.T**: Cancer Types; **Per.**: Peripheral; **C.**: Cancer

Lee, H. H., Gu, Y., Zhao, T., Xu, Y., Yang, J., Usuyama, N., Wong, C., Wei, M., Landman, B. A., Huo, Y., Santamaria-Pang, A., & Poon, H. (2024). *Foundation Models for Biomedical Image Segmentation: A Survey* (No. arXiv:2401.07654). arXiv. http://arxiv.org/abs/2401.07654

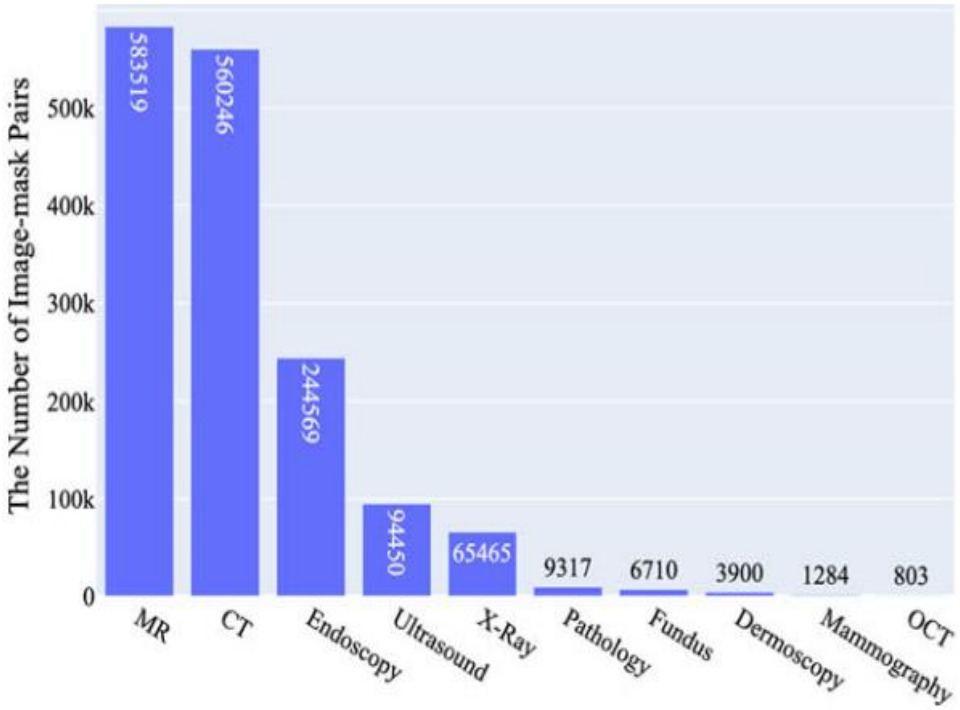
Comparisons between SOTA and SAM

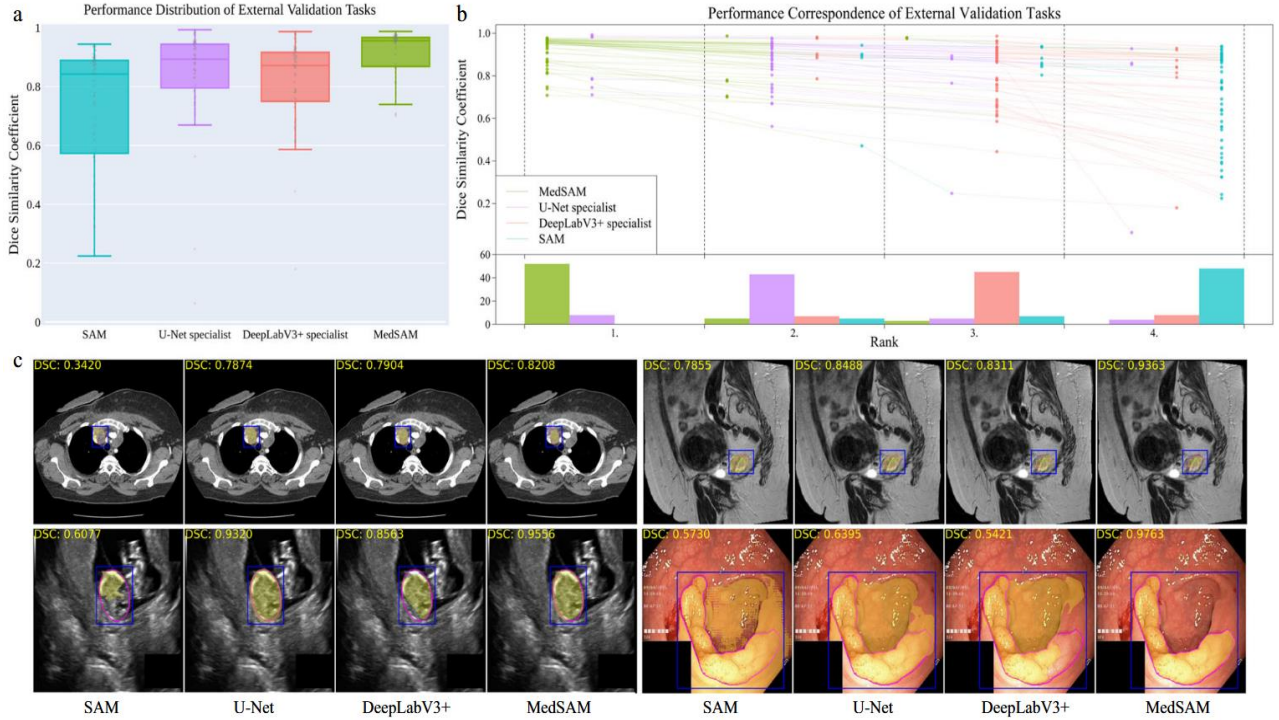
Dim.	Modality	Region	Targets	Performance SOTAs	Performance MedSAM	Performance SAM	Prompt Mode
3D	CT	Brain	Intracranial Hemorrhage	0.795 [92]	0.940	0.867 [101]	1 p.f, 2 p.b
			Glioblastoma	0.913 [52]	0.943	0.744 [101]	1 p.f, 2 p.b
		Head & Neck	Head-Neck Cancer	0.788 [9]	0.794	0.614 [101]	1 p.f, 2 p.b
			Lymph Nodes	0.742 [50]	0.821	0.771 [101]	1 p.f, 2 p.b
			Throat Cancer	0.667 [12]	0.803	0.281 [101]	1 p.f, 2 p.b
		Abdomen	Pancreas & Tumor	0.828, 0.623 [97]	0.872, 0.791	0.731, 0.741 [101]	1 p.f, 2 p.b
			Liver & Tumor	0.950, 0.790 [97]	0.980, 0.887	0.916, 0.766 [101]	1 p.f, 2 p.b
			Spleen	0.974 [67]	0.976	0.938 [65]	1 box
			Kidney & Tumor	0.948, 0.763 [88]	0.971, 0.902	0.947, 0.867 [101]	1 p.f, 2 p.b
			Aorta	0.956 [87]	0.956	0.912 [101]	1 p.f, 2 p.b
			Esophagus	0.861 [87]	0.737	0.845 [56]	1 box
			Stomach	0.921 [87]	0.962	0.855 [101]	1 p.f, 2 p.b
			Gallblader	0.921 [87]	0.918	0.872 [56]	1 box
			IVC	0.924 [87]	0.918	0.897 [56]	1 box
			Adrenal Gland	0.798 [87]	0.661	0.742 [56]	1 box
	MR	Brain	Brainstem	0.860 [113]	0.971	0.692 [101]	1 p.f, 2 p.b
			Cerebellum	0.915 [139]	0.968	0.765 [101]	1 p.f, 2 p.b
			Deep Grey Matter	0.974 [67]	0.956	0.496 [65]	1 p.f, 2 p.b
			ventricles	0.872 [86]	0.900	0.639 [101]	1 p.f, 2 p.b
			Glioma	0.878, 0.928 [118]	0.944, 0.962	0.763 (T1), 0.834 (FLAIR) [101]	1 p.f, 2 p.b
			Glioma Enhancing Tumor	0.956 [87]	0.952	0.788 [101]	1 p.f, 2 p.b
			Glioma Tumor Core	0.956 [87]	0.959	0.710 [101]	1 p.f, 2 p.b
			Ischemic Stroke	0.964 [58]	0.923	0.613 [101]	1 p.f, 2 p.b
			Meningioma	0.946, 0.892 [118]	0.979, 0.970	0.921 (T1-CE), 0.792 (T2-FLAIR) [101]	1 p.f, 2 p.b
			Vestibular Schwannoma	0.925 [41]	0.952	0.853 [101]	1 p.f, 2 p.b
		Head & Neck	Eye PL	0.930 [67]	0.941	0.815 [101]	1 p.f, 2 p.b
			Eye PR	0.923 [67]	0.940	0.819 [101]	1 p.f, 2 p.b
			Optic Nerve	0.699, 0.746 [67]	0.613, 0.703	0.395 (L), 0.433 (R) [101]	1 p.f, 2 p.b
			Bone Mandible	0.944 [67]	0.697	0.543 [101]	1 p.f, 2 p.b
			Cricopharyngeus	0.632 [67]	0.902	0.614 [101]	1 p.f, 2 p.b
			Glnd Lacrimal	0.631, 0.621 [67]	0.640, 0.687	0.613 (L), 0.599 (R) [101]	1 p.f, 2 p.b
			Glnd Submand	0.848, 0.840 [67]	0.913, 0.909	0.779 (L), 0.797 (R) [101]	1 p.f, 2 p.b
			Parotid	0.871, 0.856 [67]	0.917, 0.916	0.727 (L), 0.714 (R) [101]	1 p.f, 2 p.b
			Glottis	0.752 [67]	0.850	0.301 [101]	1 p.f, 2 p.b
			Larynx SG	0.814 [67]	0.882	0.540 [101]	1 p.f, 2 p.b
			Lips	0.722 [67]	0.869	0.584 [101]	1 p.f, 2 p.b
		Abdomen	Left Kidney	0.921 [68]	0.948	0.912 [101]	1 p.f, 2 p.b
			Right Kidney	0.927 [68]	0.948	0.921 [101]	1 p.f, 2 p.b
			Liver	0.920 [68]	0.957	0.902 [101]	1 p.f, 2 p.b
			Spleen	0.894 [68]	0.948	0.910 [101]	1 p.f, 2 p.b
		Heart	Left Atrium	0.933 [97]	0.973	0.836 [101]	1 p.f, 2 p.b
			Left Ventricle	0.959 [143]	0.985	0.775 [101]	1 p.f, 2 p.b
			Right Ventricle	0.926 [143]	0.972	0.903 [101]	1 p.f, 2 p.b
			Artery Carotid	0.874, 0.833 [152]	0.620, 0.627	0.578 (L), 0.610 (R) [101]	1 p.f, 2 p.b
			Whole Heart	0.867 [19]	0.963	0.521 [101]	1 p.f, 2 p.b
		Prostate	Prostate	0.831 [67]	0.985	0.872 [101]	1 p.f, 2 p.b
			Prostate Cancer	0.800 [126]	0.969	0.693 [101]	1 p.f, 2 p.b
		Spine	Spine	0.952 [125]	0.918	0.808 [101]	1 p.f, 2 p.b
			SpinalCord	0.891 [165]	0.774	0.559 [101]	1 p.f, 2 p.b
2D	OCT	Eye	Diabetic Macular Edema	0.983 [144]	0.950	0.884 [101]	1 p.f, 2 p.b
	X-Ray	Chest	Heart	0.950 [145]	0.968	0.901 [101]	1 p.f, 2 p.b
			Lung	0.979 [145]	0.991	0.933 [101]	1 p.f, 2 p.b
			Viral Pneumonia	0.992 [51]	0.984	0.892 [101]	1 p.f, 2 p.b
			Pneumothorax	0.891 [1]	0.815	0.502 [101]	1 p.f, 2 p.b
			Tuberculosis	0.978 [98]	0.969	0.939 [101]	1 p.f, 2 p.b
			COVID-19	0.971 [47]	0.989	0.782 [101]	1 p.f, 2 p.b
		Breast	Breast Cancer	0.963 [8]	0.833	0.665 [101]	1 p.f, 2 p.b

Lee, H. H., Gu, Y., Zhao, T., Xu, Y., Yang, J., Usuyama, N., Wong, C., Wei, M., Landman, B. A., Huo, Y., Santamaria-Pang, A., & Poon, H. (2024). *Foundation Models for Biomedical Image Segmentation: A Survey* (No. arXiv:2401.07654). arXiv. http://arxiv.org/abs/2401.07654

MedSAM

- To enhance SAM's performance on medical images, Segment Anything in Medical Images: (MedSAM) curates a large-scale dataset containing over one million medical image-mask pairs of 11 modalities to fine-tune SAM on these medical images.
- MedSAM has demonstrated substantial capabilities in segmenting a diverse array of targets and robust generalization abilities to manage new data and tasks. Its performance not only significantly exceeds that of existing the STOA segmentation foundation model, but also rivals or even surpasses specialist models.

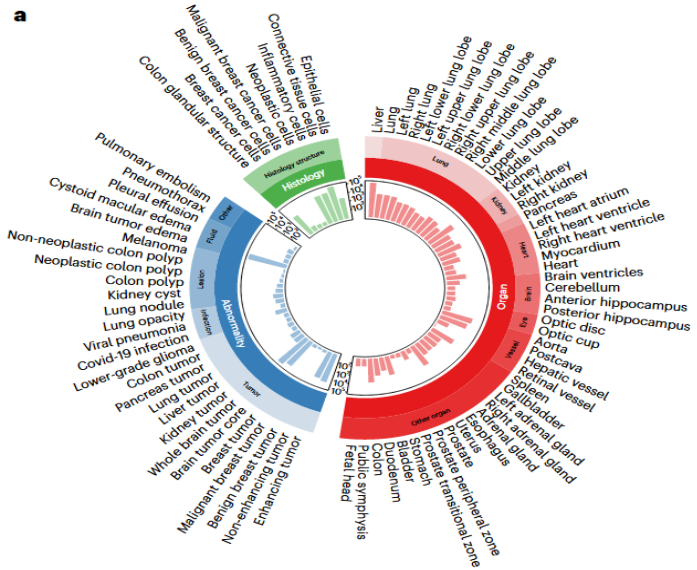


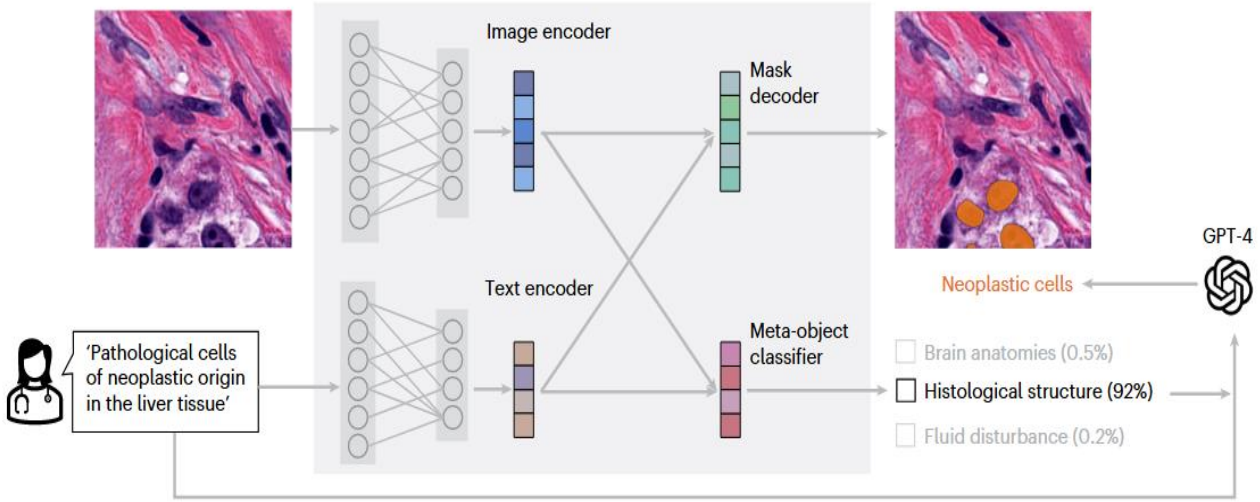


Ma, J., He, Y., Li, F., Han, L., You, C., & Wang, B. (2024). Segment anything in medical images. *Nature Communications*, *15*(1), 654. https://doi.org/10.1038/s41467-024-44824-z

BioMedParse

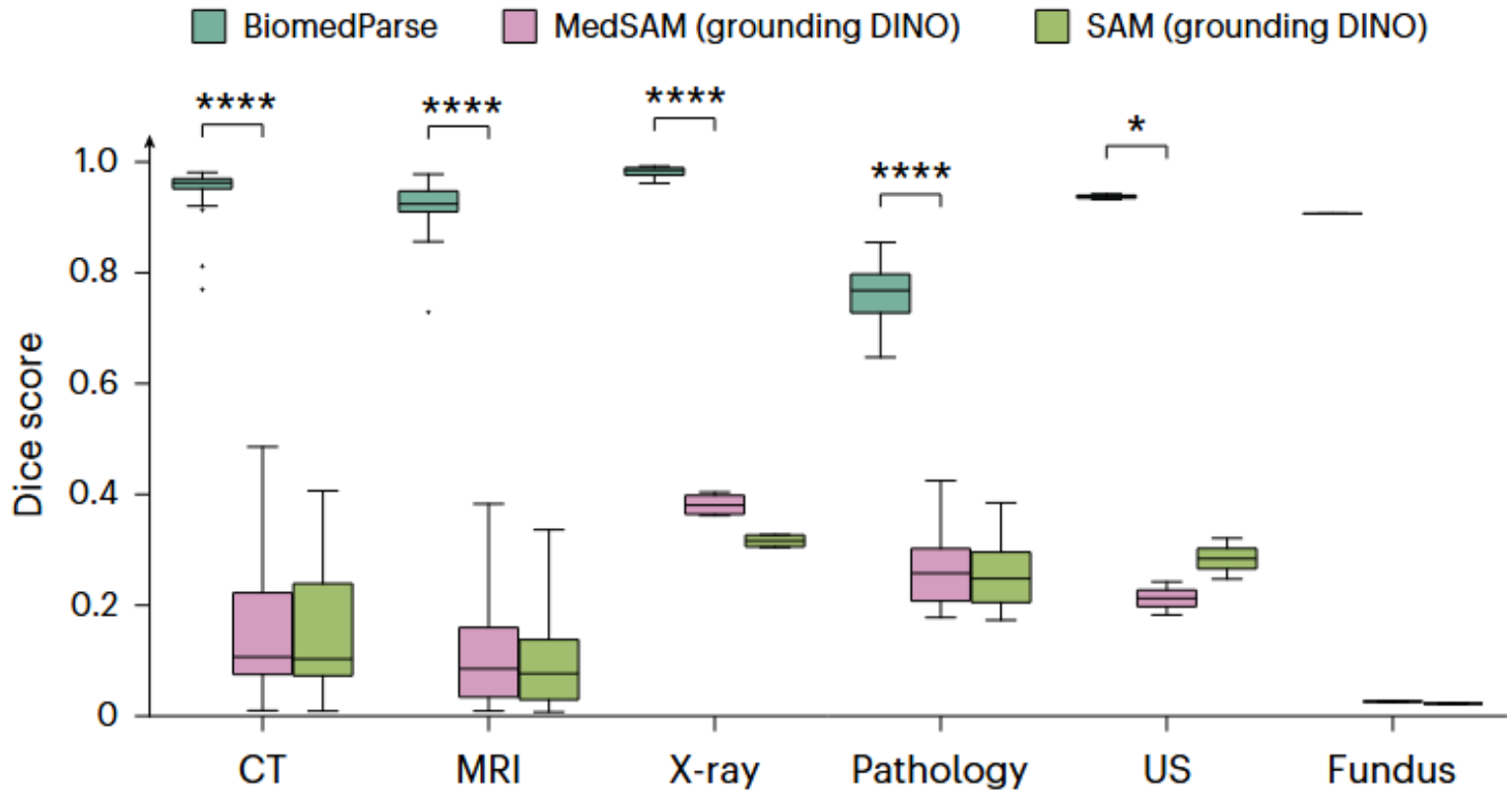
- A large dataset comprising over 6 million triples of image, segmentation mask and textual description is curated. GPT-4 is used to harmonize noisy, unstructured available texts with established biomedical object ontologies.
- Similar to SEEM, BioMedParse focuses on learning text prompts.
  - The input is an image and a text prompt that specifies the object type for segmentation and detection, which are passed along to the image and text encoders, respectively. The image encoder can be SAM-ViT, while the text encoder can be PubMedBERT or a transformer from scratch.
  - The mask decoder outputs a segmentation mask by cross-attending the image and text embeddings and gradually upsample the image features back to high-resolution pixels. At the last layer, the attention dot product on the pixel embeddings delivers the segmentation mask.





BioMedParse

- The BioMedParse outperforms existing methods such as SAM and MedSAM on image segmentation across nine imaging modalities, with larger improvement on objects with irregular shapes, it can also simultaneously segment and label all objects in an image. Moreover, using text prompts alone, BiomedParse is much more scalable than previous methods, which require orders of magnitude more user operations in specifying bounding boxes.



Content

1. Introduction to Image Segmentation
2. Introduction to U-Net
3. U-Net Extensions
4. Foundational Models for Image Segmentation
5. **Theoretical Properties**

Theoretical Challenges

1. Semantic Gap in Skip Connections: U-Net combines encoder and decoder features at the same spatial scale via skip connections.

- **Challenge:** Encoder features are low-level (e.g., edges), while decoder features are high-level (e.g., semantic). There's no theoretical guarantee that such **feature fusion preserves semantic coherence**.
- **Open Question:** What is the optimal way to bridge this gap while preserving both localization and semantics?

2. Limited Receptive Field and Global Context: U-Net is built with local convolution operations.

- **Challenge:** Its theoretical **receptive field grows linearly with depth**, so modeling long-range dependencies requires deep networks.
- **Consequence:** U-Net lacks formal mechanisms (like self-attention) to **capture global structure**, which is crucial in medical imaging for understanding spatial dependencies.

3. Overparameterization Without Generalization Guarantees: U-Net can contain tens of millions of parameters.

- **Challenge:** There are **no strong generalization bounds** for U-Net specifically. Standard bounds (e.g., VC-dimension or Rademacher complexity) are either too loose or do not reflect real-world performance.
- **Research Gap:** How does overparameterization influence generalization in structured prediction tasks like segmentation?

4. Sensitivity to Input Perturbations

- **Observation:** U-Net's segmentation output can be unstable under small changes (e.g., adversarial noise, brightness shifts).
- **Theoretical Challenge:** U-Net lacks **certified robustness guarantees**—a theoretical framework to ensure stable outputs under bounded input perturbations.
- **Implication:** This limits its safe deployment in clinical decision-making.

5. Lack of Theoretical Justification for Architectural Choices

- **Examples:** Why are two 3×3 convolutions used per block? Why use symmetric architecture between encoder and decoder?
- **Challenge:** These choices are **empirically motivated**, not derived from principled optimization or information-theoretic criteria.

6. No Optimality Guarantee in Segmentation Accuracy

- **Observation:** U-Net minimizes per-pixel cross-entropy or Dice loss.
- **Theoretical Issue:** These **losses are surrogates** and may not correspond to true segmentation performance (e.g., IoU, boundary precision).
- **Open Problem:** How to **design loss functions** that are both theoretically consistent and aligned with segmentation metrics?

Content

Project MONAI

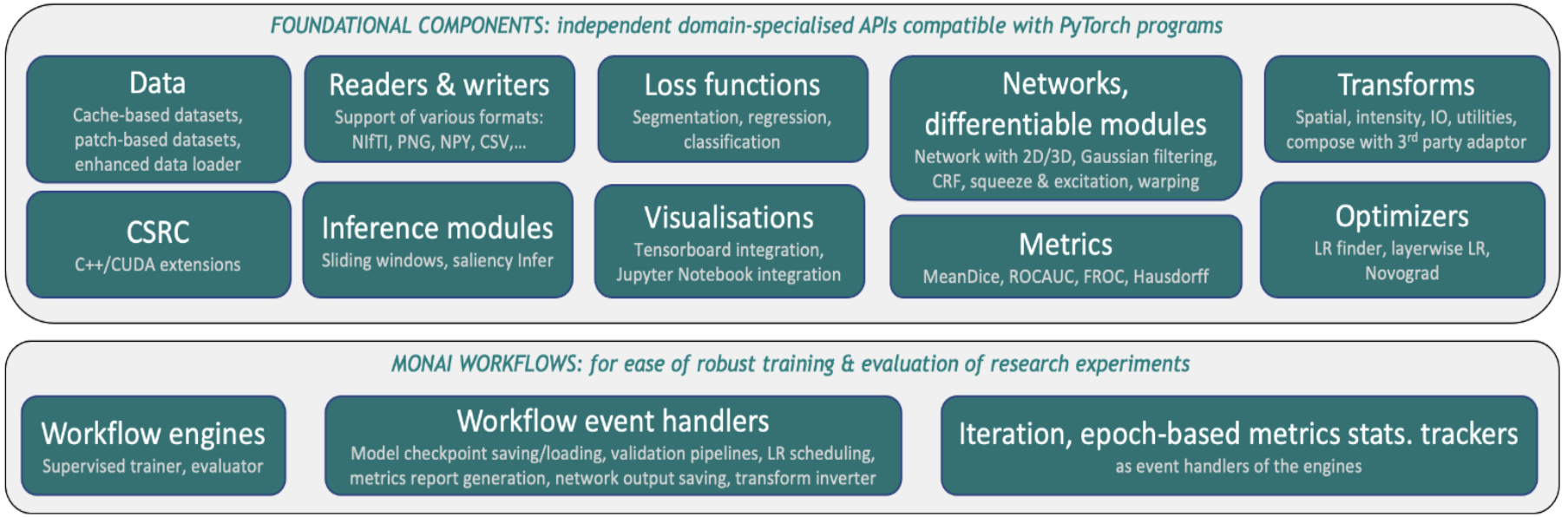
- MONAI is a freely available, community-supported PyTorch-based framework for deep learning in healthcare, providing purpose-specific AI model architectures, transformations and utilities that streamline the development and deployment of medical AI models.
- Multiple aspects are carefully considered for challenges in real data:
  - The design considers the accompanying metadata that indicates the underlying physical interpretations of the data acquisition process and relevant annotations .
  - Low-level data processing components are simple and robust to handle the data variability and highly flexible requirements.
  - High-level workflows are also introduced in addition to the exposed low-level component APIs.

MONAI

Medical Open Network for Artificial Intelligence

Empowering healthcare innovation through open-source AI frameworks that bridge research and clinical deployment

Core → Label → Deploy →



Cardoso, M. J., Li, W., Brown, R., Ma, N., Kerfoot, E., Wang, Y., Murrey, B., Myronenko, A., Zhao, C., Yang, D., Nath, V., He, Y., Xu, Z., Hatamizadeh, A., Myronenko, A., Zhu, W., Liu, Y., Zheng, M., Tang, Y., … Feng, A. (2022). *MONAI: An open-source framework for deep learning in healthcare* (No. arXiv:2211.02701). arXiv. https://doi.org/10.48550/arXiv.2211.02701

Project MONAI

- Transforms
- Loss functions
- Network architectures
- Dataset and IO
- Training, inference engines and event handlers:
- Visualization and utilizes

Reference		General Purpose
AHNet	SegResNet	AutoEncoder
BasicUNet	SegResNetVAE	Regressor
DenseNet	SENet	Classifier
DiNTS	Transchex	Discriminator
DynUNet	UNETR	Critic
EfficientNet	ViT	FullyConnectedNet
HighResNet	ViTAutoEnc	VarFullyConnectedNet
RegUNet	VNet	Generator
ResNet	SwinUNETR	UNet
		VarAutoEncoder

PyTorch Training Loop

```
for epoch in range(max_epochs):
    network.train()
    for inputs, labels in train_loader:
        optimizer.zero_grad()
        outputs = network(inputs)
        loss = loss_function(outputs, labels)
        loss.backward()
        optimizer.step()

    network.eval()
    with torch.no_grad():
        for val_inputs, val_labels in val_loader:
            val_outputs = network(val_inputs)
            metric(y_pred=val_outputs, y=val_labels)

        metric = metric.aggregate().item()
        print("Validation result:", metric)
```

MONAI Training Loop

```
evaluator = SupervisedEvaluator(
    val_data_loader=val_loader,
    network=network,
    key_val_metric={ "metric": metric },
    ...
)

trainer = SupervisedTrainer(
    max_epochs=num_epochs,
    train_data_loader=train_loader,
    network=network,
    optimizer=optimizer,
    loss_function=loss_function,
    train_handlers=[ValidationHandler(1,evaluator)],
    ...
)

trainer.run()  # do the training run for 10 epochs
```

Loss Functions in MONAI

```python
import numpy as np
from matplotlib import pyplot as plt
import torch
from torch.nn import CrossEntropyLoss
from monai.losses import (
    FocalLoss,
    DiceLoss,
    TverskyLoss,
    HausdorffDTLoss,
    DiceCELoss,
    DiceFocalLoss
)
from monai.networks.utils import one_hot

np.random.seed(1234)
torch.manual_seed(1234)
```

```python
# Let B be batch size, N be number of classes, H, W, D be height, width, depth of the image
# Shape of input should be BNH[WD]
# Shape of target should be BNH[WD] or B1H[WD]
B = 3
N = 2  # number of classes
H = 64
W = 64

# Here let mask_pred be a square and mask_true be a circle
x = np.linspace(-1, 1, W)
y = np.linspace(-1, 1, H)
xx, yy = np.meshgrid(x, y)


radius = 0.5
circle = xx**2 + yy**2 <= radius**2
mask_true_idx = torch.from_numpy(circle).long()  # long is used for indexing to avoid overflow
mask_true_idx = mask_true_idx.repeat(B, 1, 1)
mask_true = one_hot(mask_true_idx[:, None, ...], N)  # [:, None, ...] adds a new dimension

square_size = radius * 2
square = (np.abs(xx) <= radius) & (np.abs(yy) <= radius)
mask_pred = torch.from_numpy(square).float()
mask_pred = mask_pred.repeat(B, 1, 1)
mask_pred = torch.stack([1-mask_pred, mask_pred], dim=1)
mask_pred_logits = np.log((mask_pred + 1e-8) / (1 - mask_pred + 1e-8))

print(mask_pred.shape, mask_true_idx[:, None, ...].shape, mask_true.shape)
# torch.Size([3, 2, 64, 64]) torch.Size([3, 1, 64, 64]) torch.Size([3, 2, 64, 64])
```

```python
# PyTorch will apply softmax by default while MONAI won't use softmax by default
# To unify the behavior, we will always use softmax, meaning that we should use logits instead of probabilities
CE_loss = CrossEntropyLoss()
focal_loss = FocalLoss(use_softmax=True)
dice_loss = DiceLoss(softmax=True)
jaccard_loss = DiceLoss(softmax=True, jaccard=True)
tversky_loss = TverskyLoss(softmax=True)
hausdorff_loss = HausdorffDTLoss(softmax=True)
dice_ce_loss = DiceCELoss(softmax=True)
dice_focal_loss = DiceFocalLoss(softmax=True)
```

Loss Functions in MONAI

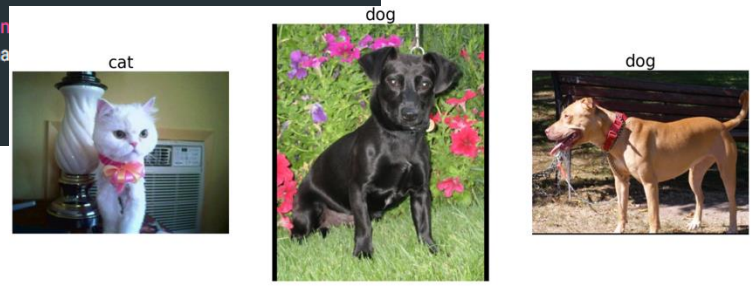
```python
# Input must be the unnormalized logits

loss_ce = CE_loss(mask_pred_logits, mask_true_idx)
loss_focal = focal_loss(mask_pred_logits, mask_true)
loss_dice = dice_loss(mask_pred_logits, mask_true)
loss_jaccard = jaccard_loss(mask_pred_logits, mask_true)
loss_tversky = tversky_loss(mask_pred_logits, mask_true)
loss_hausdorff = hausdorff_loss(mask_pred_logits, mask_true)
loss_dice_ce = dice_ce_loss(mask_pred_logits, mask_true)
loss_dice_focal = dice_focal_loss(mask_pred_logits, mask_true)

print(f"Cross Entropy Loss: {loss_ce}")  # 2.12
print(f"Focal Loss: {loss_focal}")  # 1.06
print(f"Dice Loss: {loss_dice}")  # 0.09
print(f"Jaccard Loss: {loss_jaccard}")  # 0.15
print(f"Tversky Loss: {loss_tversky}")  # 0.84
print(f"Hausdorff Loss: {loss_hausdorff}")  # 0.75
print(f"Dice CE Loss: {loss_dice_ce}")  # 2.21 = 2.12 + 0.09
print(f"Dice Focal Loss: {loss_dice_focal}")  # 1.15 = 1.06 + 0.09
```

```python
mask_true_logits = torch.log((mask_true + 1e-8) / (1 - mask_true + 1e-8))  # convert it to unnormalized logits


loss_ce = CE_loss(mask_true_logits, mask_true_idx)
loss_focal = focal_loss(mask_true_logits, mask_true)
loss_dice = dice_loss(mask_true_logits, mask_true)
loss_jaccard = jaccard_loss(mask_true_logits, mask_true)
loss_tversky = tversky_loss(mask_true_logits, mask_true)
loss_hausdorff = hausdorff_loss(mask_true_logits, mask_true)
loss_dice_ce = dice_ce_loss(mask_true_logits, mask_true)
loss_dice_focal = dice_focal_loss(mask_true_logits, mask_true)

# The loss should be 0 for perfect prediction.
print(f"Cross Entropy Loss: {loss_ce}")
print(f"Focal Loss: {loss_focal}")
print(f"Dice Loss: {loss_dice}")
print(f"Jaccard Loss: {loss_jaccard}")
print(f"Tversky Loss: {loss_tversky}")
print(f"Hausdorff Loss: {loss_hausdorff}")
print(f"Dice CE Loss: {loss_dice_ce}")
print(f"Dice Focal Loss: {loss_dice_focal}")
```

Training Models with MONAI

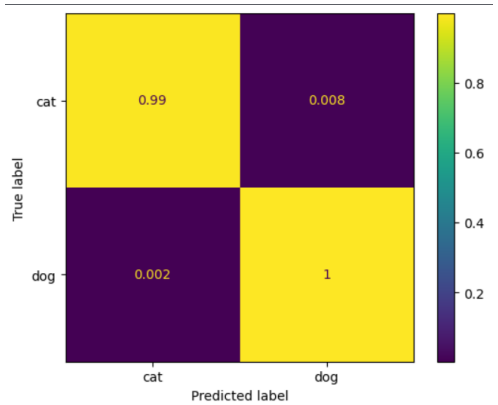
```python
import os
from enum import Enum
from matplotlib import pyplot as plt
from matplotlib import colors
from random import shuffle
import numpy as np
from tqdm.notebook import tqdm, trange
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

import torch
from monai.transforms import (
    EnsureChannelFirstd,
    Compose,
    DivisiblePadd,
    Lambdad,
    LoadImaged,
    Resized,
    Rotate90d,
    ScaleIntensityd,
)
from monai.networks.utils import eval_mode
from monai.data import Dataset, DataLoader
from monai.networks.nets import DenseNet121
from monai.data.utils import pad_list_data_collate
from monai.visualize import (
    GradCAMpp,
    OcclusionSensitivity,
    SmoothGrad,
    GuidedBackpropGrad,
    GuidedBackpropSmoothGrad,
)
from monai.utils import set_determinism
from monai.apps import download_and_extract
```

```python
# Download Kaggle dog and cat dataset https://www.kaggle.com/c/dogs-vs-cats to data_path

class Animals(Enum):
    cat = 0
    dog = 1

def remove_non_rgb(data, max_num=None):
    """Some images are grayscale or rgba. For simplicity, remove them."""
    loader = LoadImaged("image")
    out = []
    for i in data:
        if os.path.getsize(i["image"]) > 100:
            im = loader(i)["image"]
            if im.ndim == 3 and im.shape[-1] == 3:
                out.append(i)
        if max_num is not None and len(out) == max_num:
            return out
    return out


def get_data(animal, max_num=None):
    files = glob(os.path.join(data_path, "PetImages", animal.name.capitalize(), "*.jpg"))
    data = [{"image": i, "label": animal.value} for i in files]
    shuffle(data)
    data = remove_non_rgb(data, max_num)
    return data

# We only need 500 of each class as this is sufficien
cats, dogs = [get_data(i, max_num=500) for i in Anima
all_data = cats + dogs
shuffle(all_data)
```



Training Models with MONAI

```python
batch_size = 8
divisible_factor = 20
transforms = Compose(
    [
        LoadImaged("image"),  # Load image from file path and create a dictionary with 'image' key
        EnsureChannelFirstd("image"),  # Ensure image has channel-first format (C,H,W) or (C,H,W,D)
        ScaleIntensityd("image"),  # Normalize image intensity values to [0,1] range
        Rotate90d("image", k=3),   # Rotate image 270 degrees
        DivisiblePadd("image", k=divisible_factor),  # Pad image dimensions to be divisible by divisible_factor
    ]
)

ds = Dataset(all_data, transforms)
dl = DataLoader(
    ds,
    batch_size=batch_size,
    shuffle=True,
    num_workers=4,  # adjust downwards if memory is limited
    collate_fn=pad_list_data_collate,
    drop_last=True,
)
```

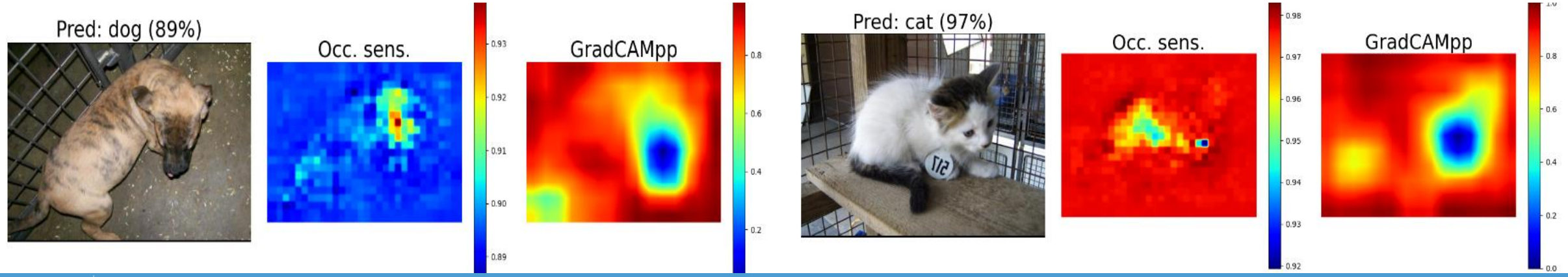



```python
model = DenseNet121(spatial_dims=2, in_channels=3, out_channels=2, pretrained=True).to(device)
optimizer = torch.optim.Adam(model.parameters(), 1e-5)  # Use Adam optimizer with learning rate 1e-5

# Only use cross entropy loss
def criterion(y_pred, y):
    return torch.nn.functional.cross_entropy(y_pred, y, reduction="sum")

def get_num_correct(y_pred, y):
    return (y_pred.argmax(dim=1) == y).sum().item()

max_epochs = 2  # 2 epochs are enough for this small dataset
for epoch in trange(max_epochs, desc="Epoch"):
    epoch_loss = 0
    acc = 0
    for data in dl:
        inputs, labels = data["image"].to(device), data["label"].to(device)
        optimizer.zero_grad()
        outputs = model(inputs)

        train_loss = criterion(outputs, labels)
        acc += get_num_correct(outputs, labels)

        train_loss.backward()
        optimizer.step()
        epoch_loss += train_loss.item()
    epoch_loss /= len(dl) * batch_size
    acc /= len(dl) * batch_size
    print(f"Epoch {epoch+1}, loss: {epoch_loss:.3f}, acc: {acc:.4f}")
```

Interpretability with MONAI

```
target_layer = "class_layers.relu"
gradcampp = GradCAMpp(model, target_layers=target_layer)
occ_sens = OcclusionSensitivity(
    model,
    mask_size=32,
    n_batch=batch_size,
    overlap=0.5,
    verbose=False,
)


def saliency(model, d):
    ims = []
    titles = []
    log_scales = []
    img = torch.as_tensor(d["image"])[None].to(device)
    pred_logits = model(img)
    pred_label = pred_logits.argmax(dim=1).item()
    pred_prob = int(torch.nn.functional.softmax(pred_logits, dim=1)[0, pred_label].item() * 100)
    # Image
    ims.append(torch.moveaxis(img, 1, -1))
    titles.append(f"Pred: {Animals(pred_label).name} ({pred_prob}%)")
    log_scales.append(False)
    # Occlusion sensitivity images
    occ_map, _ = occ_sens(img)
    ims.append(occ_map[0, pred_label][None])
    titles.append("Occ. sens.")
    log_scales.append(False)
    # GradCAM
    res_cam_pp = gradcampp(x=img, class_idx=pred_label)[0]
    ims.append(res_cam_pp)
    titles.append("GradCAMpp")
    log_scales.append(False)
```

```
num_examples = 2
rand_data = np.random.choice(ds, replace=False, size=num_examples)
tr = tqdm(rand_data)
for row, d in enumerate(tr):
    tr.set_description(f"img shape: {d['image'].shape[1:]}")
    ims, titles, log_scales = saliency(model, d)
    if row == 0:
        num_cols = len(ims)
        subplot_shape = [num_examples, num_cols]
        figsize = [i * 5 for i in subplot_shape][::-1]  # 5 is a scale factor
        fig, axes = plt.subplots(*subplot_shape, figsize=figsize, facecolor="white")
    add_row(ims, titles, log_scales, row, axes, num_examples)
plt.tight_layout()
```



References

Azad, R., Aghdam, E. K., Rauland, A., Jia, Y., Avval, A. H., Bozorgpour, A., Karimijafarbigloo, S., Cohen, J. P., Adeli, E., & Merhof, D. (2022). *Medical Image Segmentation Review: The success of U-Net* (No. arXiv:2211.14830). arXiv. http://arxiv.org/abs/2211.14830

Azad, R., Heidary, M., Yilmaz, K., Hüttemann, M., Karimijafarbigloo, S., Wu, Y., Schmeink, A., & Merhof, D. (2023). *Loss Functions in the Era of Semantic Segmentation: A Survey and Outlook* (No. arXiv:2312.05391). arXiv. http://arxiv.org/abs/2312.05391

Billot, B., Greve, D. N., Puonti, O., Thielscher, A., Van Leemput, K., Fischl, B., Dalca, A. V., & Iglesias, J. E. (2023). SynthSeg: Segmentation of brain MRI scans of any contrast and resolution without retraining. *Medical Image Analysis*, *86*, 102789. https://doi.org/10.1016/j.media.2023.102789

R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill et al., "On the opportunities and risks of foundation models," arXiv preprint arXiv:2108.07258, 2021.

Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., & Wang, M. (2021). *Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation* (No. arXiv:2105.05537). arXiv. https://doi.org/10.48550/arXiv.2105.05537

Cardoso, M. J., Li, W., Brown, R., Ma, N., Kerfoot, E., Wang, Y., Murrey, B., Myronenko, A., Zhao, C., Yang, D., Nath, V., He, Y., Xu, Z., Hatamizadeh, A., Myronenko, A., Zhu, W., Liu, Y., Zheng, M., Tang, Y., … Feng, A. (2022). *MONAI: An open-source framework for deep learning in healthcare* (No. arXiv:2211.02701). arXiv. https://doi.org/10.48550/arXiv.2211.02701

Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A. L., & Zhou, Y. (2021). *TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation* (No. arXiv:2102.04306). arXiv. https://doi.org/10.48550/arXiv.2102.04306

Chen, X., Wang, X., Zhang, K., Fung, K.-M., Thai, T. C., Moore, K., Mannel, R. S., Liu, H., Zheng, B., & Qiu, Y. (2022). *Recent advances and clinical applications of deep learning in medical image analysis*. Medical Image Analysis, 79, 102444. https://doi.org/10.1016/j.media.2022.102444

Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., & Girdhar, R. (2022). *Masked-attention Mask Transformer for Universal Image Segmentation* (No. arXiv:2112.01527). arXiv. https://doi.org/10.48550/arXiv.2112.01527

Christensen, M., Vukadinovic, M., Yuan, N., & Ouyang, D. (2024). Vision–language foundation model for echocardiogram interpretation. *Nature Medicine*, *30*(5), 1481–1488. https://doi.org/10.1038/s41591-024-02959-y

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation* (No. arXiv:1606.06650). arXiv. https://doi.org/10.48550/arXiv.1606.06650

Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013

Gu, A., & Dao, T. (2024). *Mamba: Linear-Time Sequence Modeling with Selective State Spaces* (No. arXiv:2312.00752). arXiv. https://doi.org/10.48550/arXiv.2312.00752

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2021). *Masked Autoencoders Are Scalable Vision Learners* (No. arXiv:2111.06377). arXiv. https://doi.org/10.48550/arXiv.2111.06377

Isensee, F., Jaeger, P. F., Kohl, S. A. A., Petersen, J., & Maier-Hein, K. H. (2021). nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, *18*(2), 203–211. https://doi.org/10.1038/s41592-020-01008-z

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment Anything. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3992–4003. https://doi.org/10.1109/ICCV51070.2023.00371

Lee, H. H., Gu, Y., Zhao, T., Xu, Y., Yang, J., Usuyama, N., Wong, C., Wei, M., Landman, B. A., Huo, Y., Santamaria-Pang, A., & Poon, H. (2024). *Foundation Models for Biomedical Image Segmentation: A Survey* (No. arXiv:2401.07654). arXiv. http://arxiv.org/abs/2401.07654

Liu, L., Cheng, J., Quan, Q., Wu, F.-X., Wang, Y.-P., & Wang, J. (2020). *A survey on U-shaped networks in medical image segmentations*. Neurocomputing, 409, 244–258. https://doi.org/10.1016/j.neucom.2020.05.070

Ma, J., Chen, J., Ng, M., Huang, R., Li, Y., Li, C., Yang, X., & Martel, A. L. (2021). *Loss odyssey in medical image segmentation*. Medical Image Analysis, 71, 102035. https://doi.org/10.1016/j.media.2021.102035

Ma, J., He, Y., Li, F., Han, L., You, C., & Wang, B. (2024). Segment anything in medical images. *Nature Communications*, *15*(1), 654. https://doi.org/10.1038/s41467-024-44824-z

Ma, J., Li, F., & Wang, B. (2024). *U-Mamba: Enhancing Long-range Dependency for Biomedical Image Segmentation* (No. arXiv:2401.04722). arXiv. http://arxiv.org/abs/2401.04722

References

Masood, S., Sharif, M., Masood, A., Yasmin, M., & Raza, M. (2015). A Survey on Medical Image Segmentation. *Current Medical Imaging Reviews*, *11*(1), 3–14. https://doi.org/10.2174/1573405611011504231034 41

Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). *V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation* (No. arXiv:1606.04797). arXiv. https://doi.org/10.48550/arXiv.1606.04797

Pearson, H., Ledford, H., Hutson, M., and Van Noorden, R. (2025) Exclusive: the most-cited papers of the twenty-first century, *Nature*. 588 |, Vol 640.

Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., & Feichtenhofer, C. (2024). *SAM 2: Segment Anything in Images and Videos* (No. arXiv:2408.00714). arXiv. https://doi.org/10.48550/arXiv.2408.00714

Ronneberger, O., Fischer, P., & Brox, T. (2015, May 18). *U-Net: Convolutional Networks for Biomedical Image Segmentation.* arXiv.Org. https://arxiv.org/abs/1505.04597v1

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision*, *128*(2), 336–359. https://doi.org/10.1007/s11263-019-01228-7

Suganyadevi, S., Seethalakshmi, V., & Balasamy, K. (2022). *A review on deep learning in medical image analysis*. International Journal of Multimedia Information Retrieval, 11(1), 19–38. https://doi.org/10.1007/s13735-021-00218-1

Wang, R., Lei, T., Cui, R., Zhang, B., Meng, H., & Nandi, A. K. (2022). *Medical image segmentation using deep learning: A survey*. IET Image Processing, 16(5), 1243–1267. https://doi.org/10.1049/ipr2.12419

Zhou, T., Zhang, F., Chang, B., Wang, W., Yuan, Y., Konukoglu, E., & Cremers, D. (2024). *Image Segmentation in Foundation Model Era: A Survey* (No. arXiv:2408.12957). arXiv. http://arxiv.org/abs/2408.12957

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2018). *UNet++: A Nested U-Net Architecture for Medical Image Segmentation* (No. arXiv:1807.10165). arXiv. https://doi.org/10.48550/arXiv.1807.10165

Zou, X., Yang, J., Zhang, H., Li, F., Li, L., Wang, J., Wang, L., Gao, J., & Lee, Y. J. (2023). *Segment Everything Everywhere All at Once* (No. arXiv:2304.06718). arXiv. https://doi.org/10.48550/arXiv.2304.06718