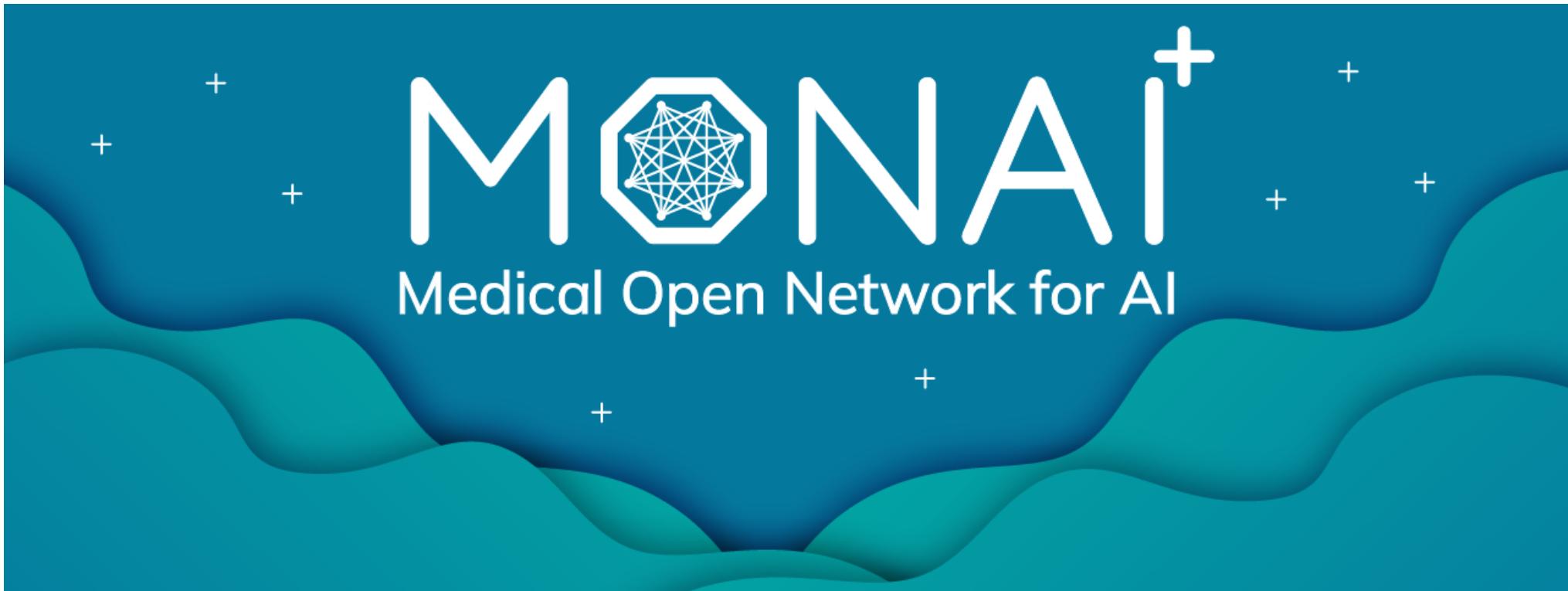


## Medical/Bio Research Topics Ⅱ: Week 07 (17.10.2025)

# **Hands-on AI Segmentation Model Development (2): Model Architecture**

**인공지능 분할 모델 개발 실습 (2): 모델 구조**

# Medical Open Network for Artificial Intelligence (MONAI)



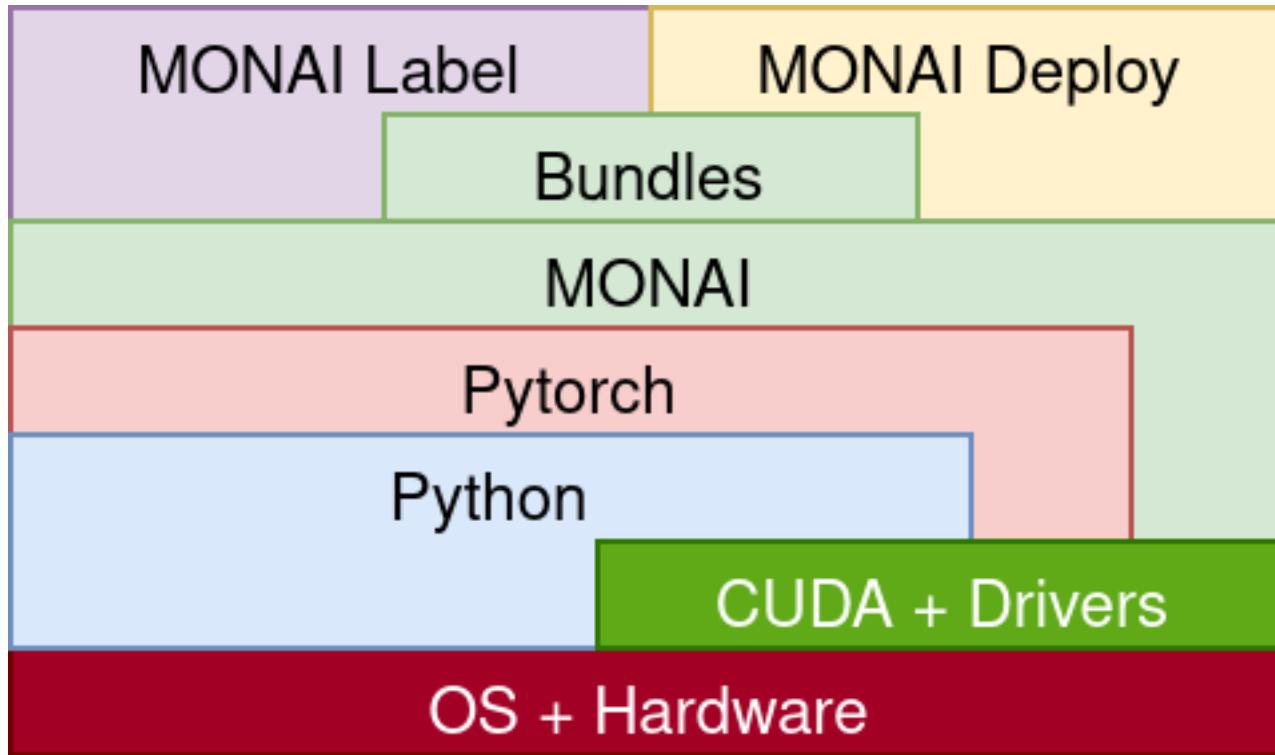
[<https://github.com/Project-MONAI>]

- Project MONAI
  - Set of open-source, freely available collaborative frameworks
  - For accelerating research and clinical collaboration in medical imaging
  - Originally started by NVIDIA and King's College London and expanded to a consortium of 16 institutions aiming to advance healthcare through medical imaging
  - Has gained widespread recognition as a standard within the medical imaging AI development community

- Key features
  - Open source: Built on PyTorch with Apache 2.0 license
  - Standardized: Best practices for healthcare AI with focus on medical imaging
  - User-friendly: Clear error messages and intuitive API
  - Reproducible: Facilitates replication of research experiments
  - Easy integration: Compatible with existing tools and third-party components
  - High quality: Enterprise-grade development with tutorials and robust documentation
- Latest version
  - Stable: 1.5.0 released in June 2025
  - Developing: 1.6.dev2541 released in October 2025

- Released multiple open-source PyTorch-based frameworks for annotating, building, training, deploying, and optimizing AI workflows in healthcare
  - MONAI Core: Foundation framework for medical AI development
    - Domain-specific capabilities for training AI models for medical imaging
    - Built-in medical transforms, networks, losses, and evaluation metrics
    - Integration with PyTorch ecosystem while adding medical-specific functionality
  - MONAI Bundle: Standardized model packaging and sharing system
    - Self-contained model definitions with configs, weights, and metadata
    - Enables reproducible research and easy model distribution
    - Integration with Model Zoo for community collaboration

- MONAI Label: Interactive annotation and active learning platform
  - AI-assisted labeling with DeepEdit and DeepGrow interactive models
  - Active learning strategies to reduce annotation time
  - Integration with 3D Slicer and OHIF viewers
- MONAI Deploy: Clinical deployment and integration framework
  - Robust framework for deploying AI models in clinical settings
  - DICOM integration and clinical workflow support
  - Production-ready containerization and orchestration
- MONAI Model Zoo: Community model repository
  - Collection of pre-trained medical imaging models in Bundle format
  - Easy access to state-of-the-art models for various medical tasks
  - Standardized evaluation and benchmarking



[<https://docs.monai.io/en/stable/>]

## Stack Architecture of Project MONAI

# Why Medical-Specific Frameworks Like MONAI?

- Unique challenges in medical imaging AI
  - 3D volumetric complexity
  - Spatial heterogeneity
  - Domain-specific data formats
  - High-dimensional complexity

- Limitations of general AI frameworks for medical applications
  - Missing medical transforms
  - Inadequate 3D support
  - Lack of domain-specific metrics
  - Format incompatibility
- Development challenges with general AI frameworks
  - Extended R & D lifecycle
  - Increased project risks
  - Reduced reproducibility

- MONAI's domain-specific solutions
  - Medical-optimized pipelines: Pre-built transforms and preprocessing workflows
  - 3D-native architectures: Optimized deep learning designs for volumetric medical data
  - Medical-specific metrics: Built-in medical loss functions and evaluation metrics
  - Seamless format integration: Native support for medical file formats
  - Standardized best practices: Proven methodologies for medical AI development

# MONAI Core

- Flagship framework created by Project MONAI
- Provides domain-specific capabilities for training AI models for medical imaging
- Supports wrappers and adaptors that allow popular healthcare AI tools to be used from within MONAI
  - Developed with minimal required dependencies, namely PyTorch and NumPy

- Key design principles
  - Looks and feels like PyTorch
  - Opt-in and incremental over PyTorch
  - Fully integrates with the PyTorch ecosystem
- Installation
  - `$ pip install monai`

- Key modules
  - **monai.data**: Datasets, image readers/writers, and synthetic data generation
  - **monai.transforms**: Medical image transforms for preprocessing and postprocessing
  - **monai.networks**: Network architectures, building blocks, and PyTorch utilities
  - **monai.metrics**: Medical evaluation metrics (Dice, IoU, ROC-AUC, etc.)
  - **monai.losses**: Medical-specific loss functions (Dice, Hausdorff, SSIM, etc.)

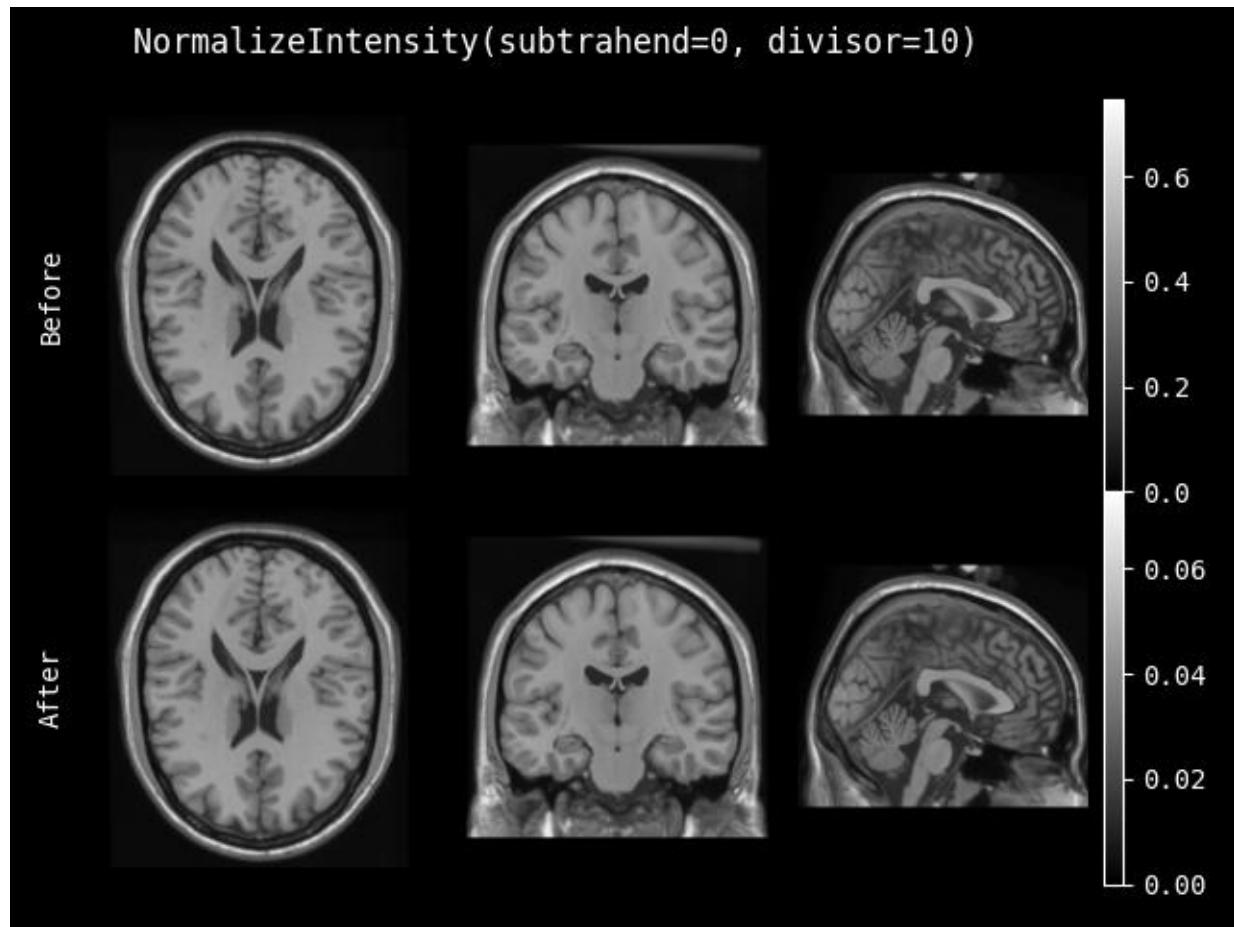
- **monai.optimizers**: Learning rate schedulers and optimization utilities
- **monai.visualize**: Visualization tools including class activation mapping and occlusion sensitivity analysis
- **monai.engines**: Workflow engines for training and evaluation pipelines
- **monai.apps**: Ready-to-use applications
  - Auto3DSeg: Automated 3D segmentation pipeline
- **monai.bundle**: Model packaging and configuration management
- **monai.utils**: Utility functions and helper tools
- **monai.f1**: Federated learning client support

- **monai .data** module
  - Generic interfaces
    - **Dataset**: Loads data samples
    - **ArrayDataset**: Loads array format input data
    - **ImageDataset**: Loads image/segmentation pairs
  - Image readers/writers
    - **ImageReader**: Loads medical image files
    - **ImageWriter**: Writes images to files
  - Key features
    - Native support for medical file formats
    - Efficient data loading pipelines
    - Synthetic data generation capabilities

- **monai.transforms** module
  - Input/output transforms
    - **LoadImage**: Loads images from specified paths with format-specific readers
      - nii, nii.gz: NibabelReader
      - png, jpg, bmp: PILReader
      - npz, npy: NumpyReader
      - nrrd: NrrdReader
      - DICOM file: ITKReader
    - **SaveImage**: Saves images and metadata to files

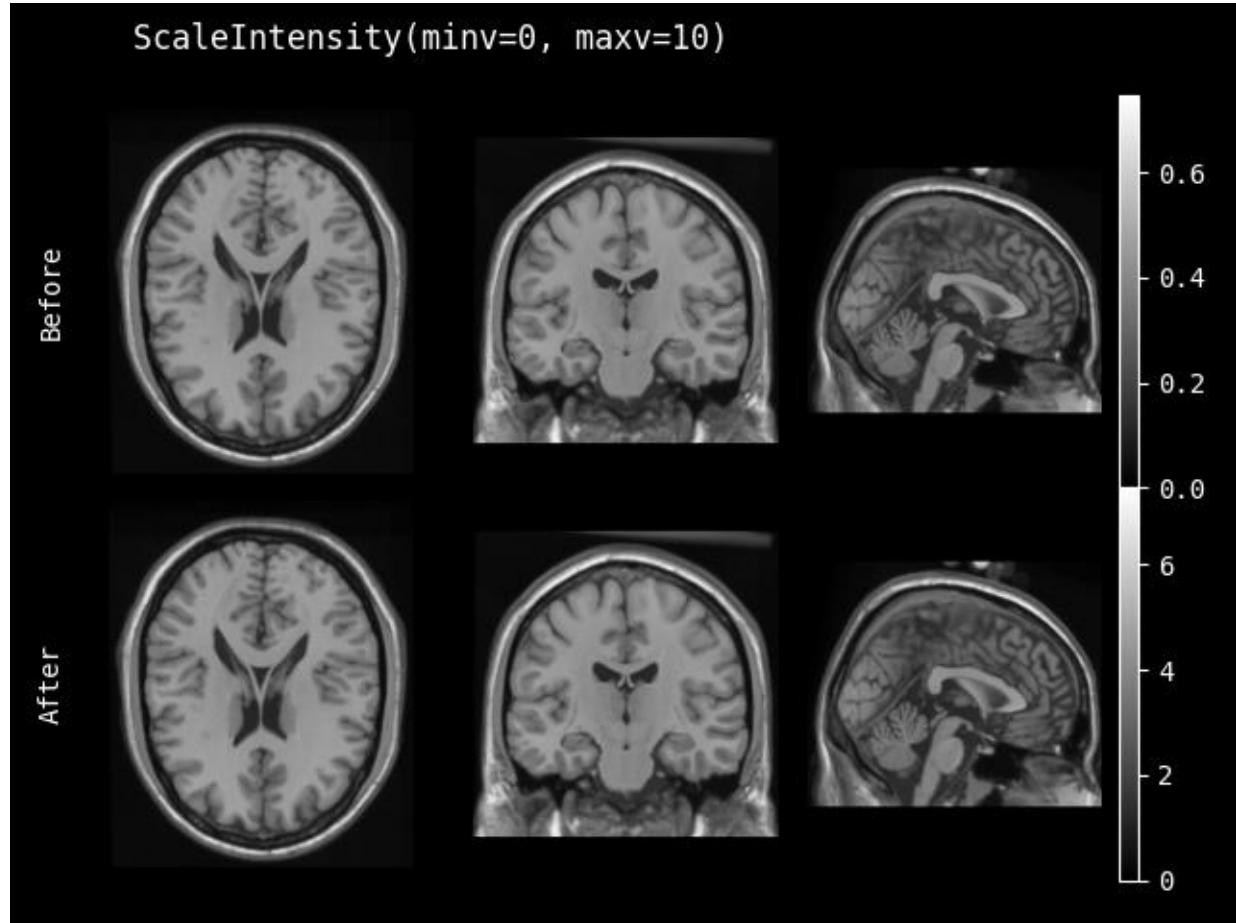
- Intensity transforms

- **NormalizeIntensity**: Normalizes images based on mean and standard deviation
- **ScaleIntensity**: Scales intensity to specified value range
- **RandGaussianNoise**: Adds Gaussian noise for augmentation



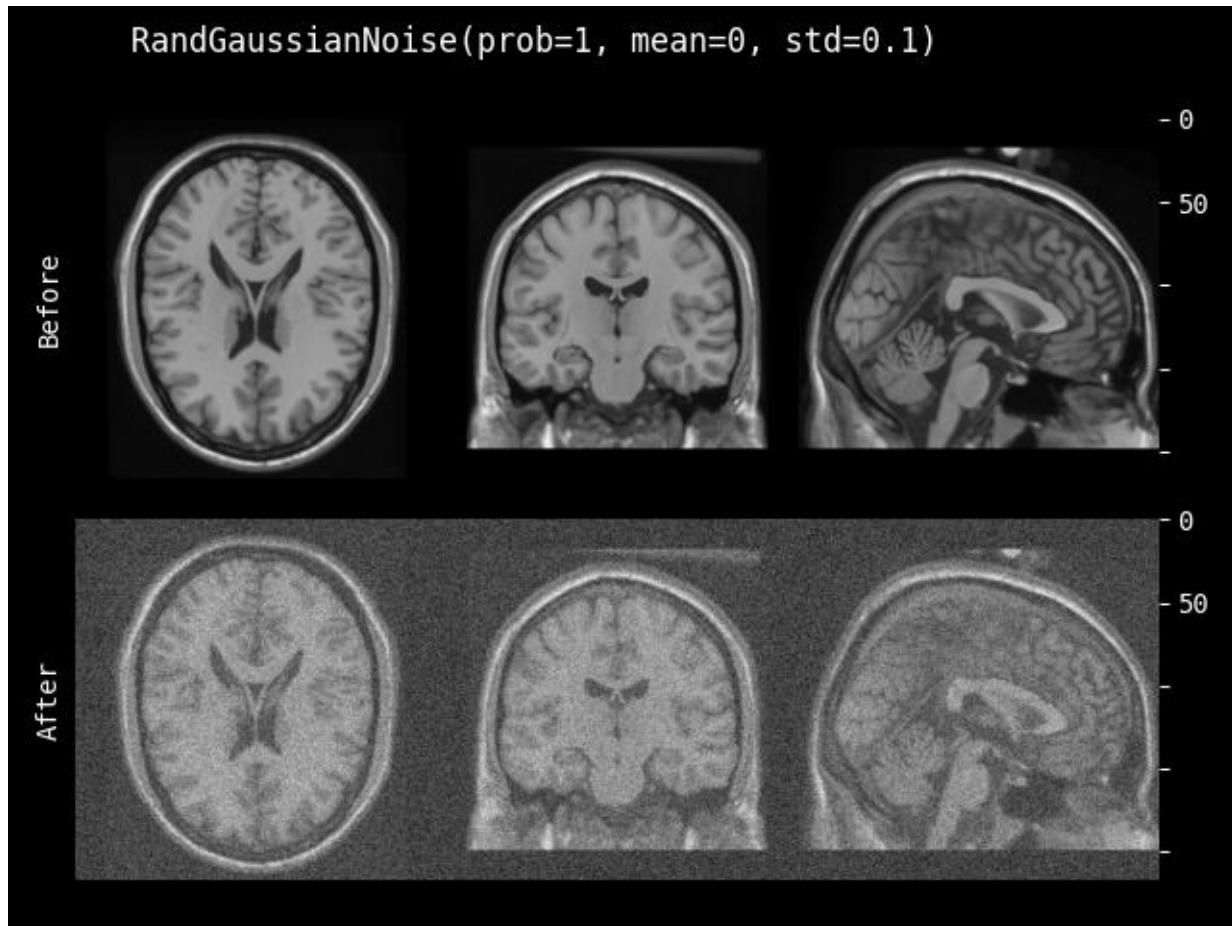
[<https://docs.monai.io/en/stable/transforms.html>]

**monai.transforms.NormalizeIntensity**



[<https://docs.monai.io/en/stable/transforms.html>]

**monai.transforms.ScaleIntensity**



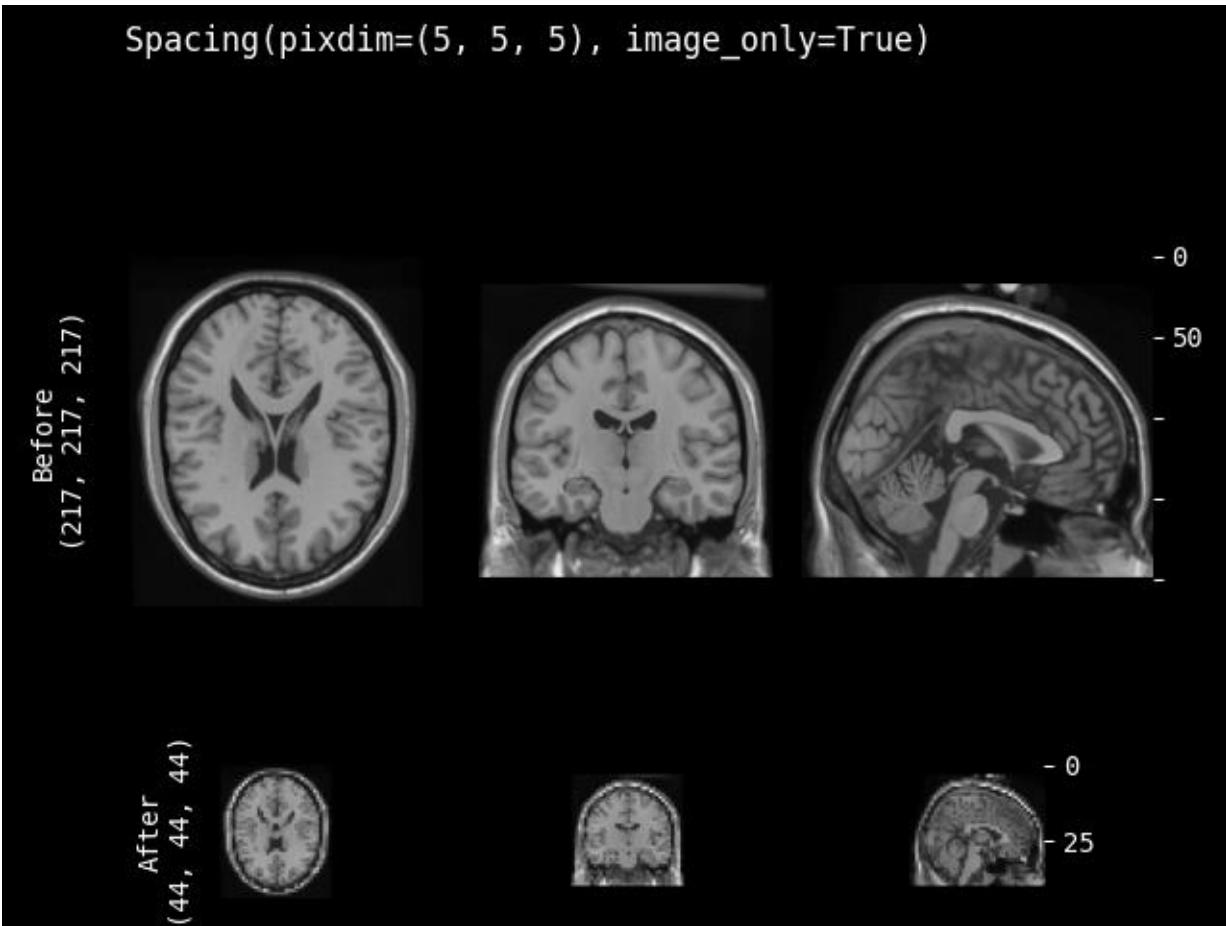
[<https://docs.monai.io/en/stable/transforms.html>]

**monai.transforms.RandGaussianNoise**

- Spatial transforms

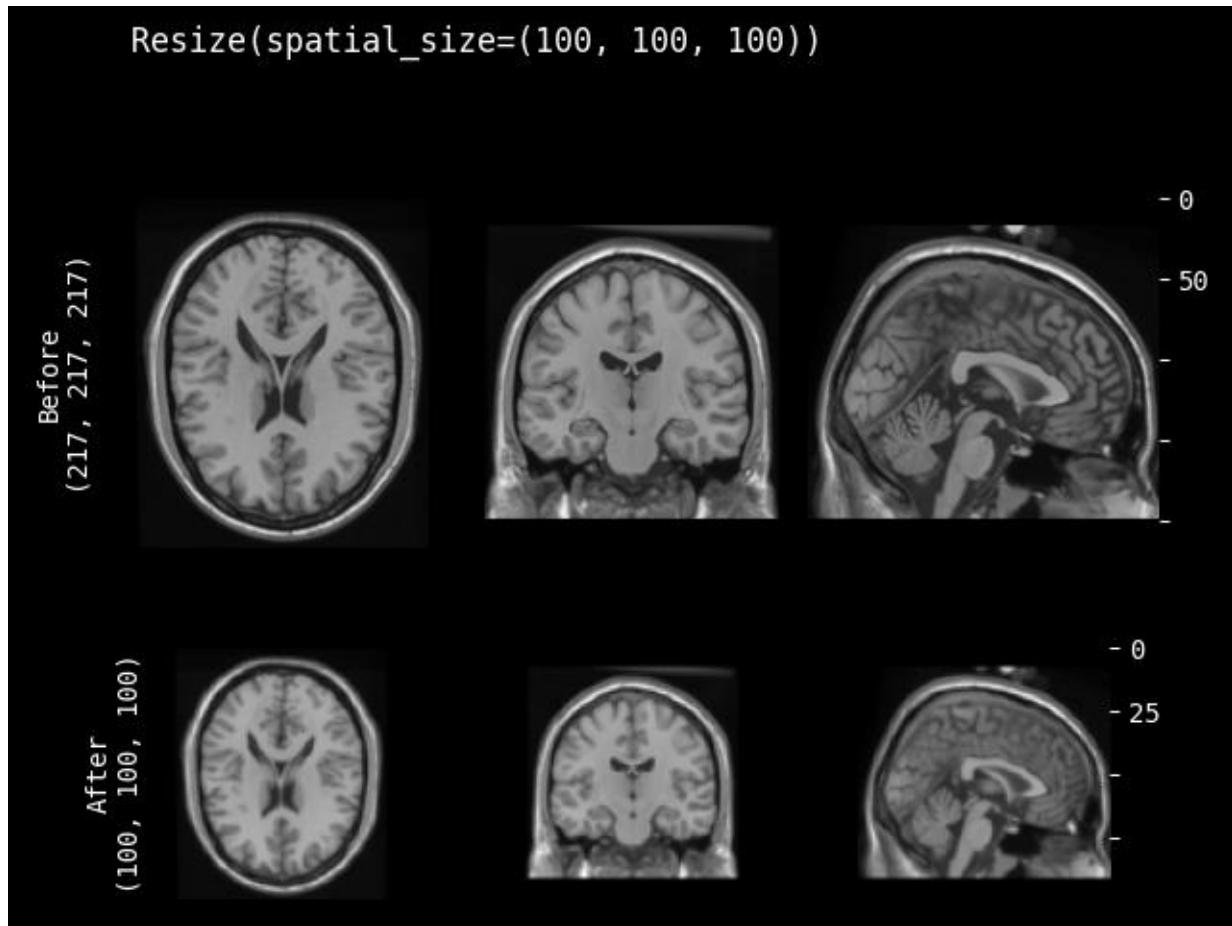
- **Spacing:** Resamples images to specified voxel spacing
- **Resize:** Resizes images to specified spatial dimensions
- **Orientation:** Changes image orientation (e.g., 'RAS')
- **Flip, Rotate, Zoom:** Basic geometric augmentations

```
Spacing(pixdim=(5, 5, 5), image_only=True)
```



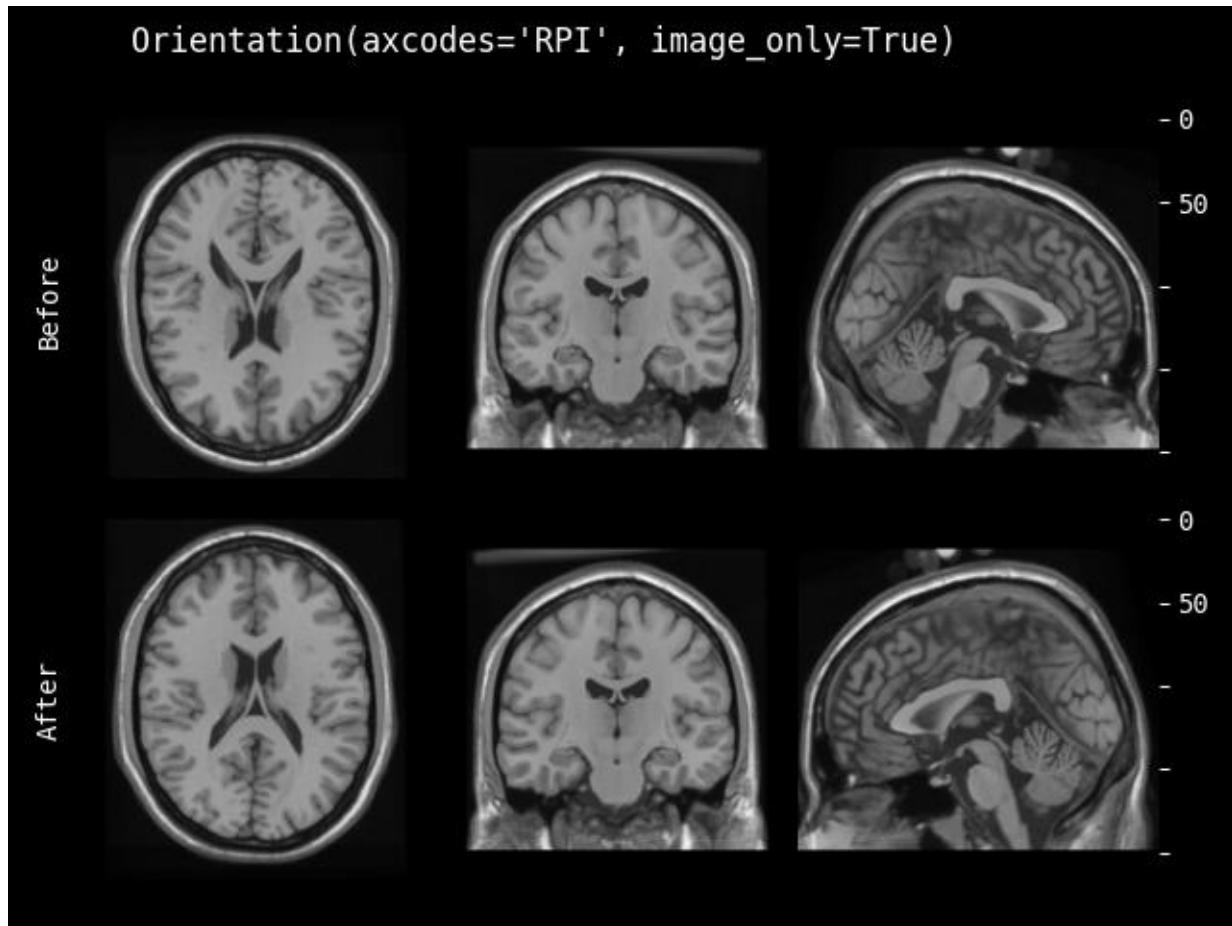
[<https://docs.monai.io/en/stable/transforms.html>]

**monai.transforms.Spacing**



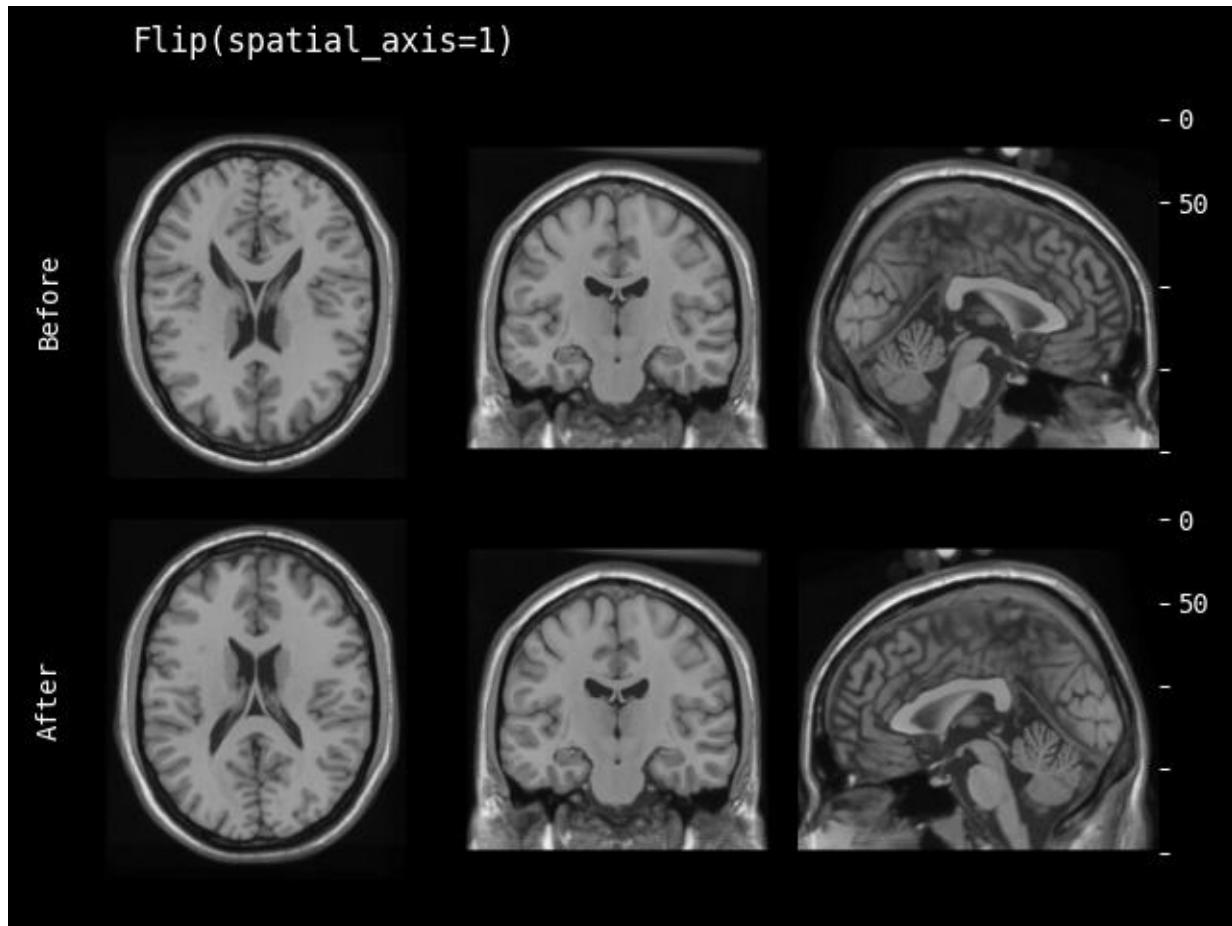
[<https://docs.monai.io/en/stable/transforms.html>]

**monai.transforms.Resize**



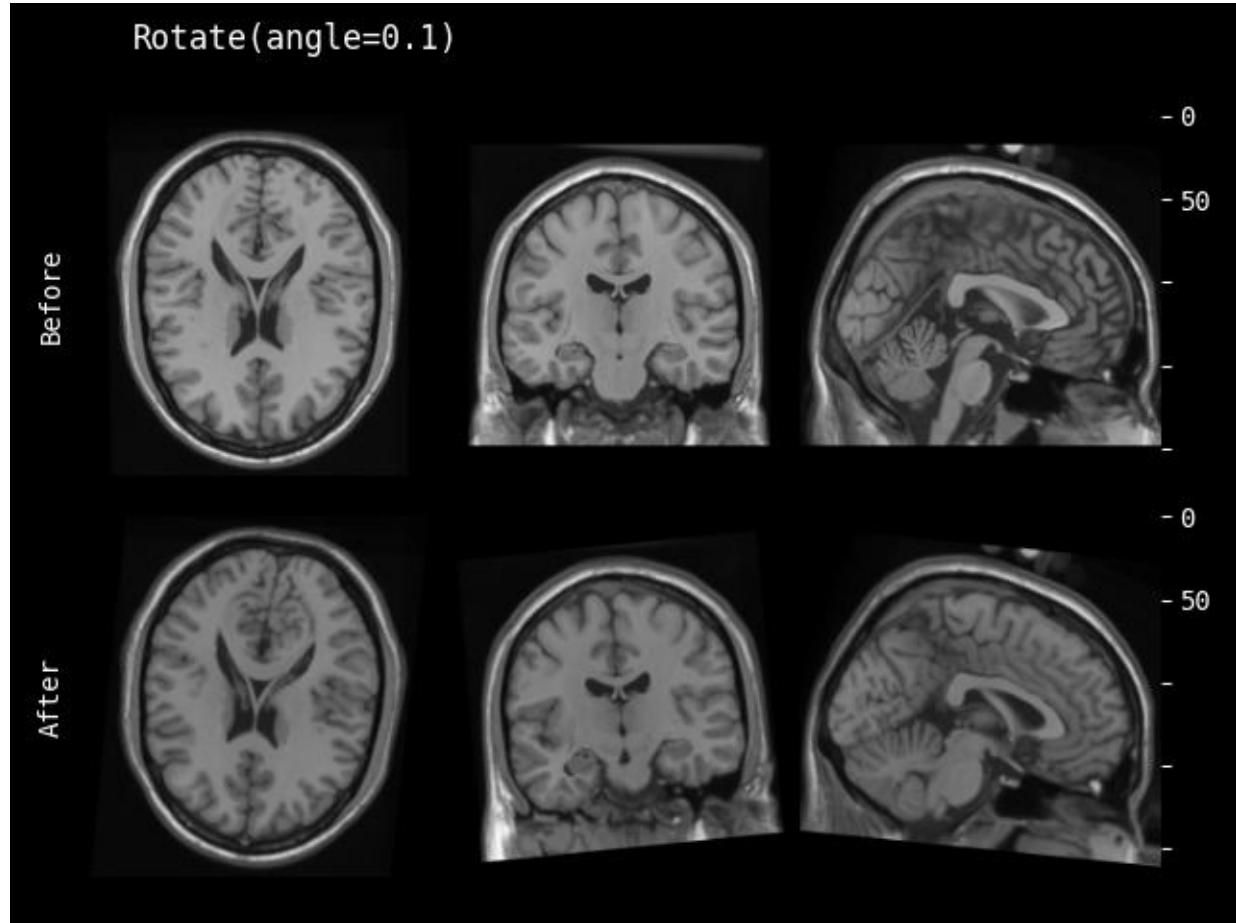
[<https://docs.monai.io/en/stable/transforms.html>]

**monai.transforms.Orientation**



[<https://docs.monai.io/en/stable/transforms.html>]

**monai.transforms.Flip**



[<https://docs.monai.io/en/stable/transforms.html>]

**monai.transforms.Rotate**



[<https://docs.monai.io/en/stable/transforms.html>]

`monai.transforms.Zoom`

- **monai.networks** module
  - Basic building blocks
    - Layers: **Conv**, **Norm**, **Dropout**, **Act**, **Pool**, **Pad**, **Flatten**
    - Blocks: **Convolution**, **DenseBlock**, **TransformerBlock**, **UnetrBasicBlock**
  - Complete architectures
    - Traditional: **UNet**, **VNet**, **DynUNet**, **AttentionUNet**
    - ResNet-based: **ResNet**, **SegResNet**
    - Transformer-based: **UNETR**, **SwinUNETR**
    - General purpose: **DenseNet**, **FullyConnectedNet**

## – Utilities

- **convert\_to\_onnx**: Converts models to ONNX (Open Neural Network Exchange) format
- **convert\_to\_torchscript**: Converts models to TorchScript

- **monai.metrics** module
  - Segmentation metrics
    - **DiceMetric**
    - **compute\_iou**
  - Classification metrics
    - **ROCAUCMetric**
    - **ConfusionMatrixMetric**
  - Regression metrics
    - **RMSEMetric**
    - **MSEMetric**
    - **MAEMetric**

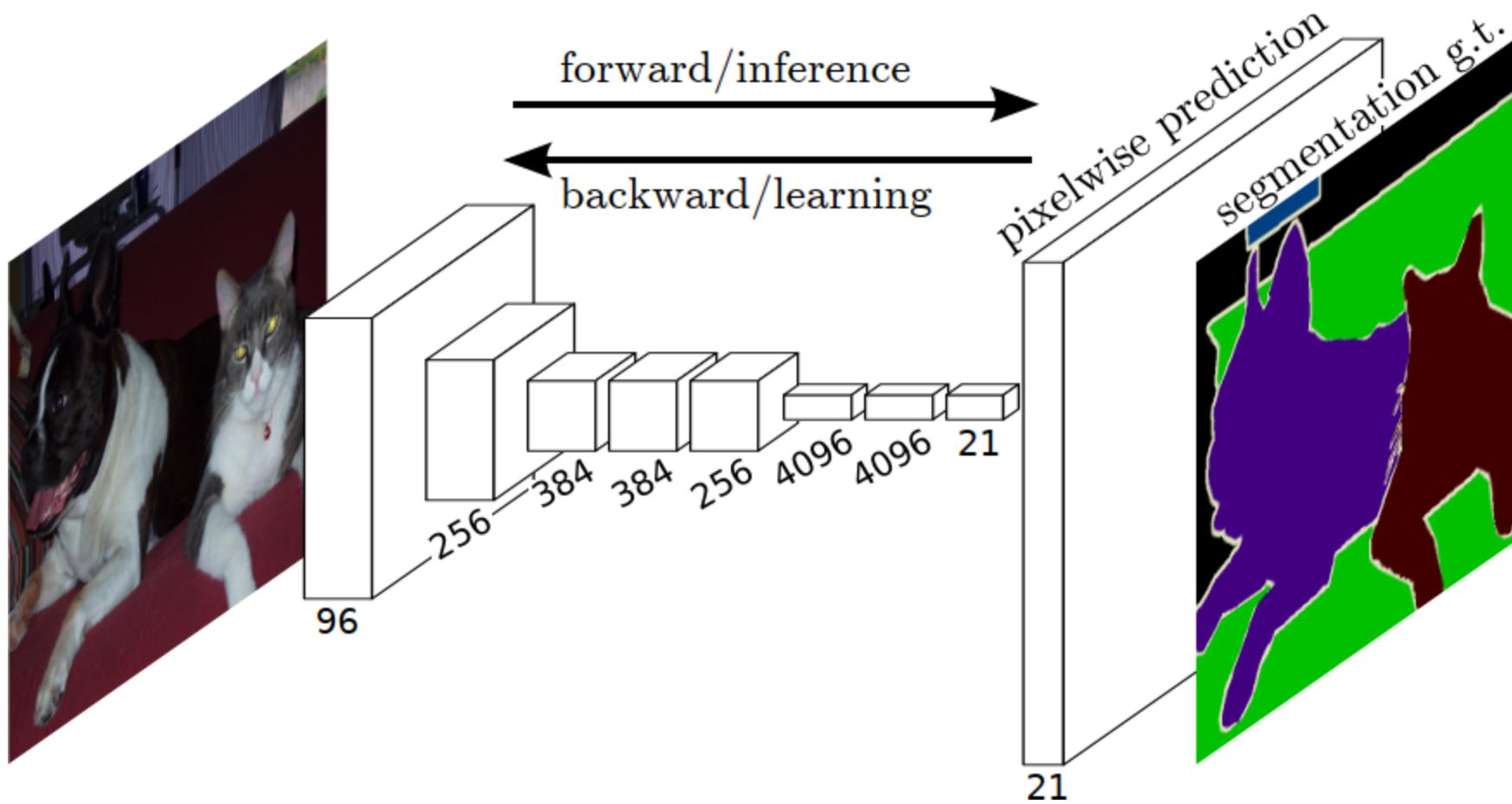
- `monai.losses` module
  - Segmentation losses
    - `DiceLoss`
    - `HausdorffDTLoss`
  - Registration losses
    - `LocalNormalizedCrossCorrelationLoss`
    - `GlobalMutualInformationLoss`
  - Reconstruction losses
    - `SSIMLoss`

- **monai.apps** module
  - **Auto3DSeg**: Automated 3D segmentation pipeline
    - **AutoRunner**: Core automated segmentation workflow

- **monai.visualize** module
  - Class activation mapping
    - CAM, GradCAM, GradCAMpp
  - Occlusion sensitivity analysis
    - OcclusionSensitivity

# Deep Learning Architectures for Lesion Segmentation

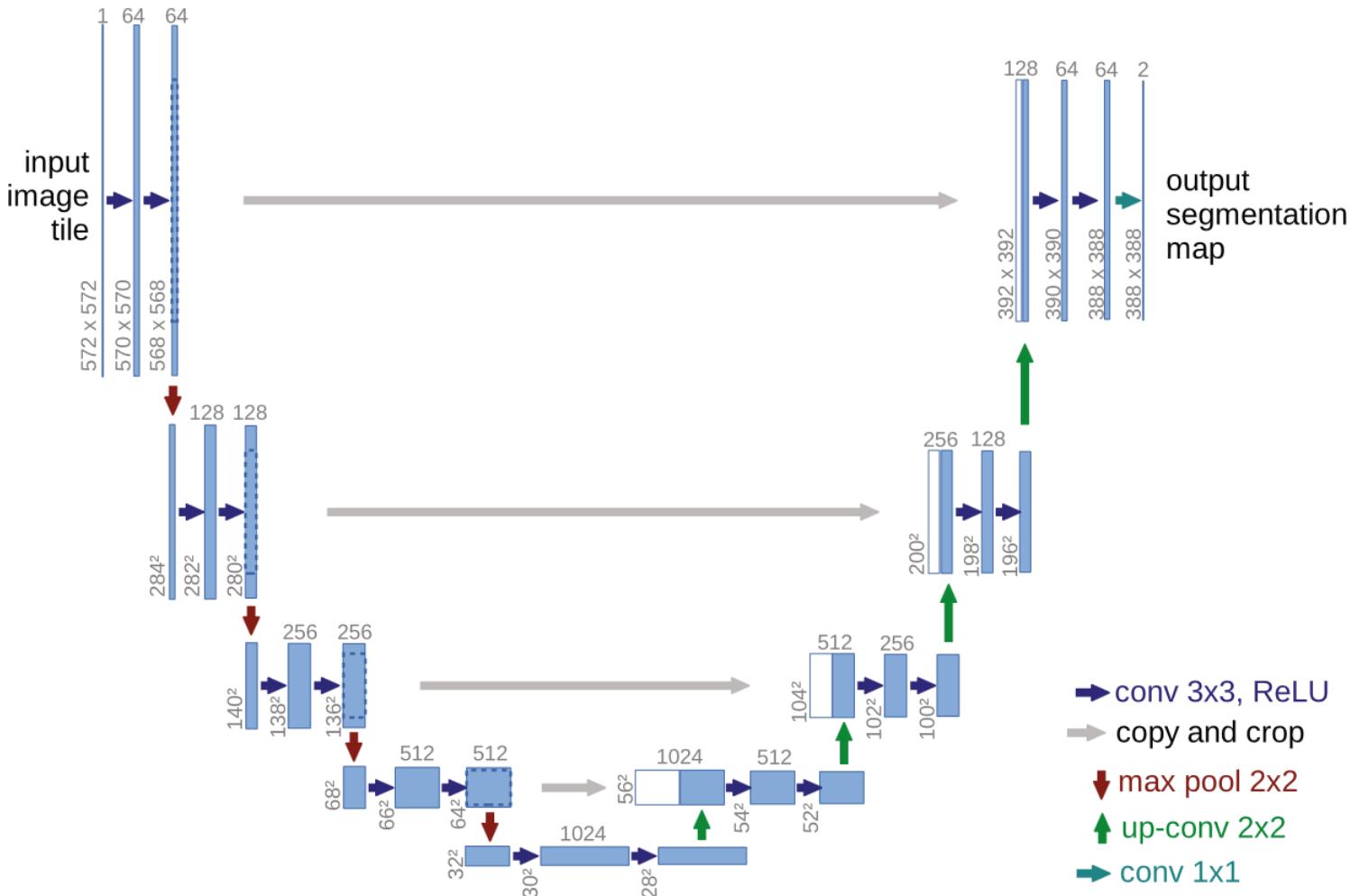
- Fully convolutional network (FCN)
  - "Fully Convolutional Networks for Semantic Segmentation"  
[Long et al., 2015]
  - As a type of convolutional neural network (CNN), emphasizes full convolution throughout the network, preserving spatial information and enabling pixel/voxel-wise predictions on variable-sized inputs
  - Serves as a foundation for many subsequent architectures



[Long et al., 2015])

## FCN Architecture

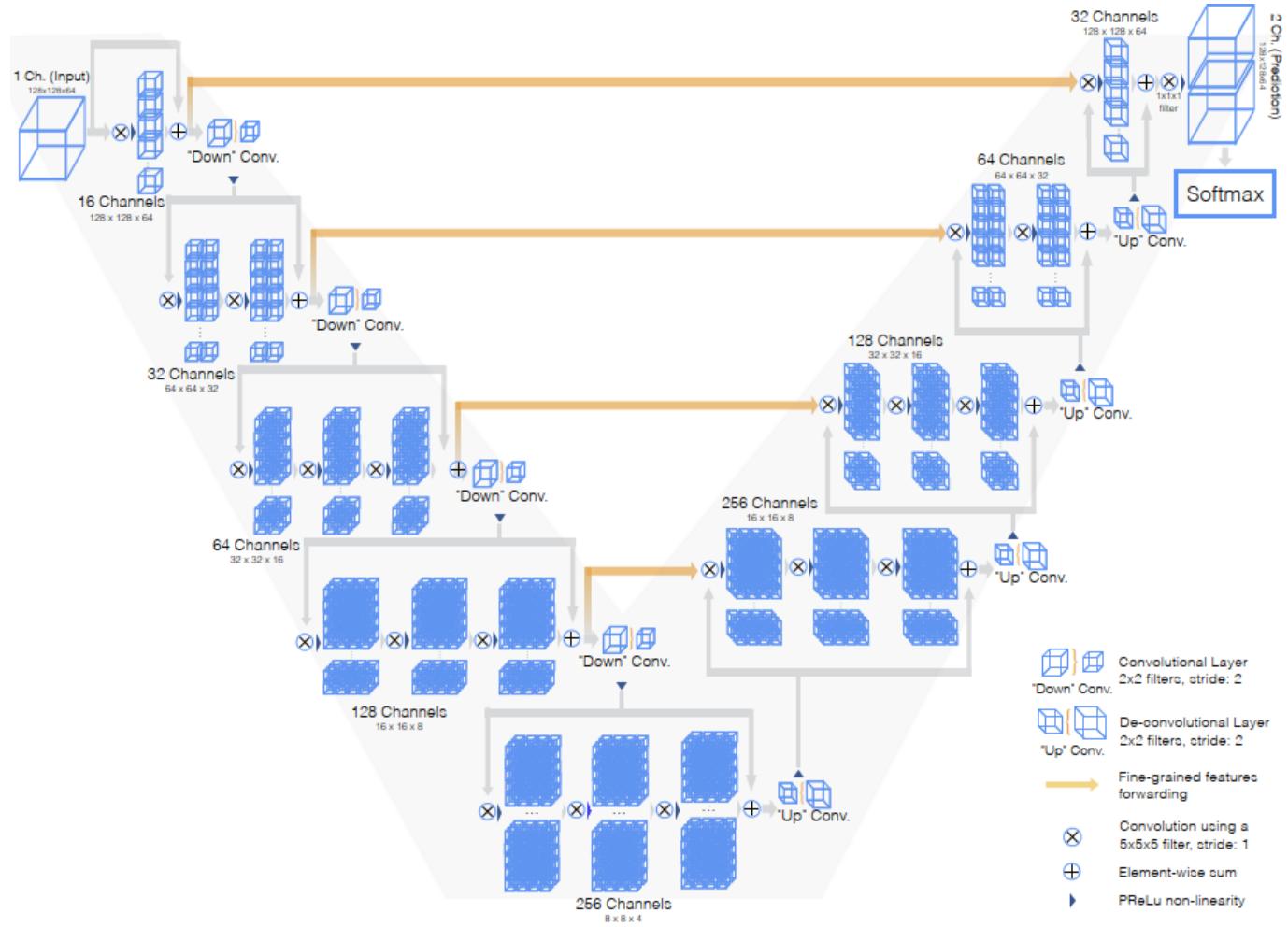
- U-Net
  - "U-Net: Convolutional Networks for Biomedical Image Segmentation" [\[Ronneberger et al., 2015\]](#)
  - Introduces a symmetric encoder-decoder structure with skip connections between corresponding layers
  - Particularly effective in capturing fine-grained details in medical images
  - Remains one of the most commonly used architectures in medical image segmentation



[Ronneberger et al., 2015])

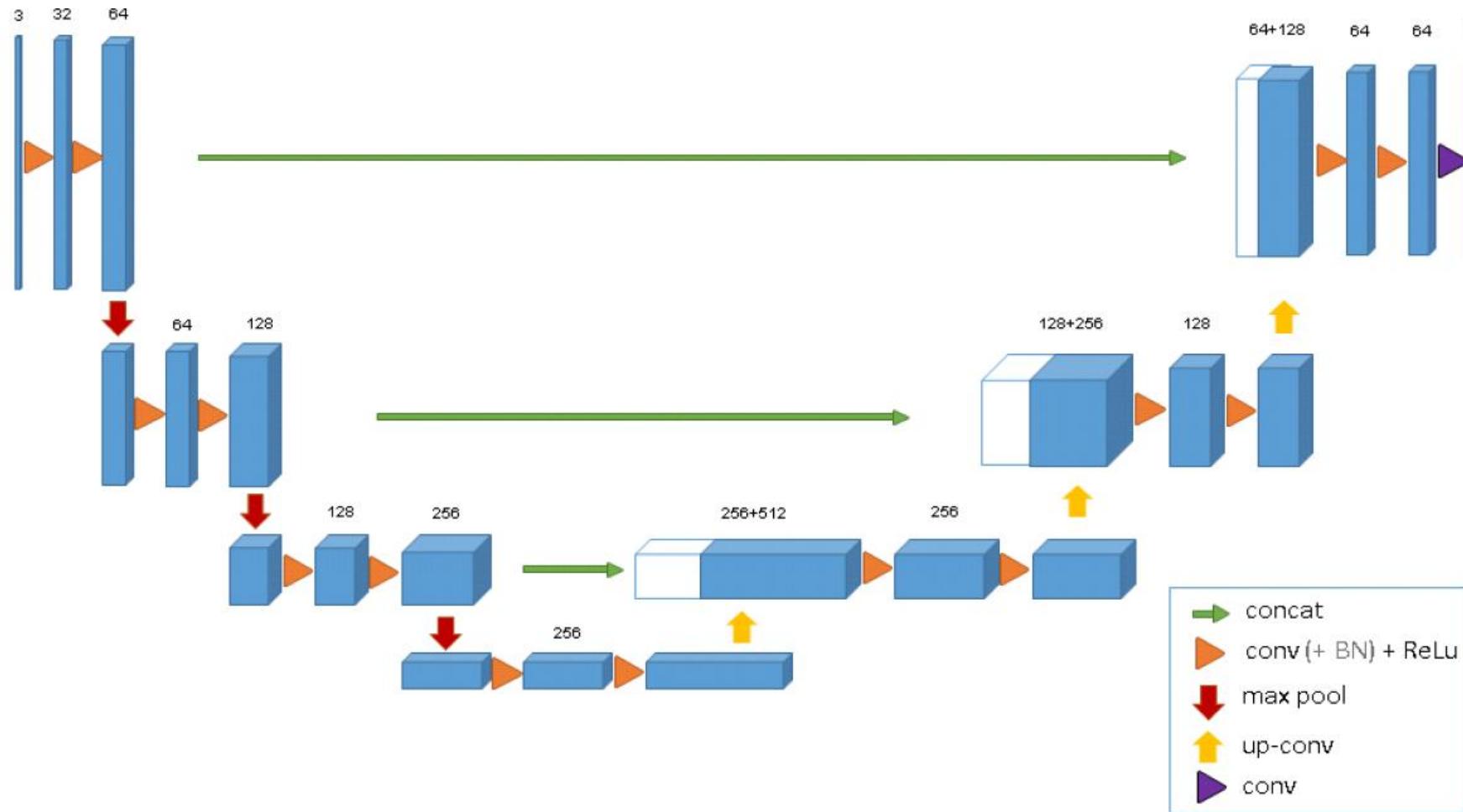
## U-Net Architecture

- V-Net and 3D U-Net
  - V-Net: "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation" [\[Milletari et al., 2016\]](#)
  - 3D U-Net: "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation" [\[Çiçek et al., 2016\]](#)
  - Extend the original 2D U-Net based on the core encoder-decoder structure with skip connections to 3D, but with significant modifications, including the use of a Dice loss function for effectively addressing class imbalance and particularly improving segmentation performance for small structures, in V-Net



[Milletari et al., 2016])

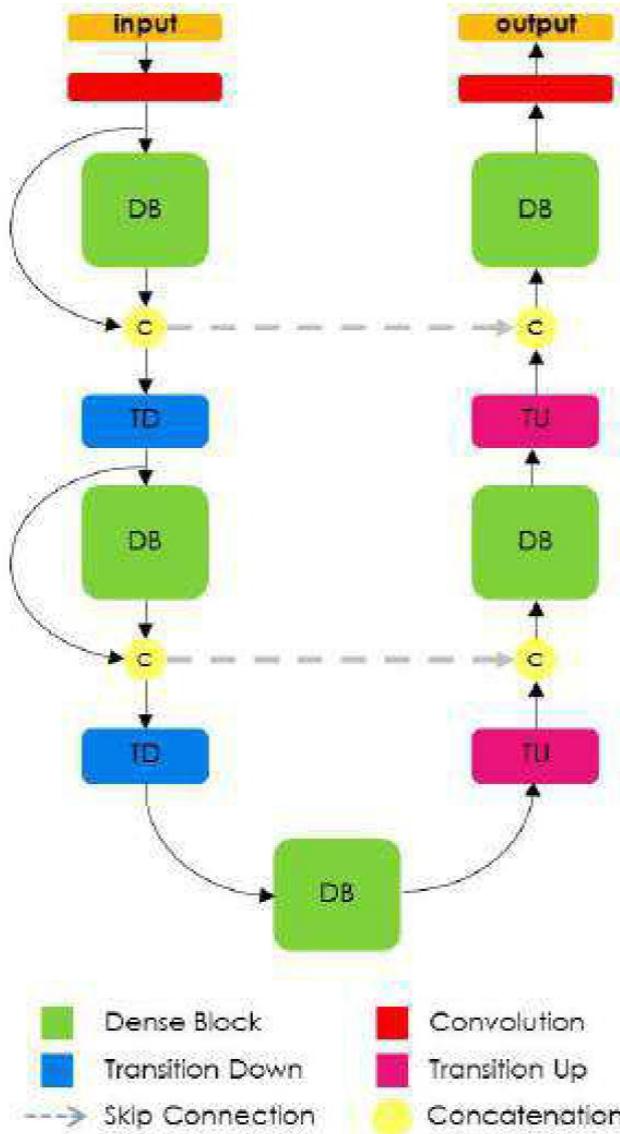
## V-Net Architecture



[Çiçek et al., 2016]

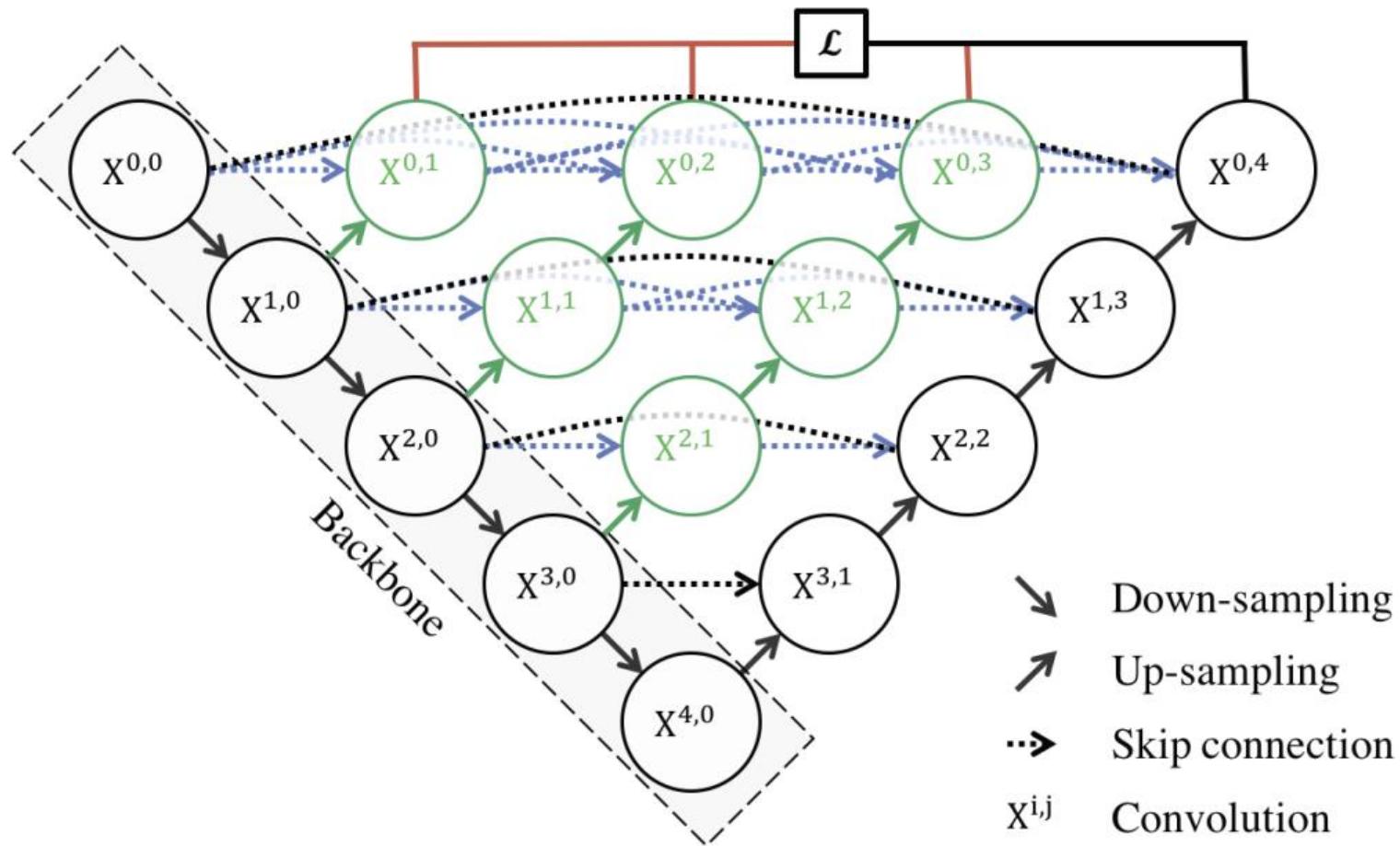
## 3D U-Net Architecture

- Various U-Net variants
  - Dense U-Net: "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation" [Jégou et al., 2017]
    - Incorporates dense connections within each encoder and decoder block
  - Nested U-Net (UNet++): "UNet++: A Nested U-Net Architecture for Medical Image Segmentation" [Zhou et al., 2018]
    - Uses dense skip connections between encoder and decoder and introduces intermediate supervision to reduce the semantic gap between encoder and decoder features



[Jégou et al., 2017]

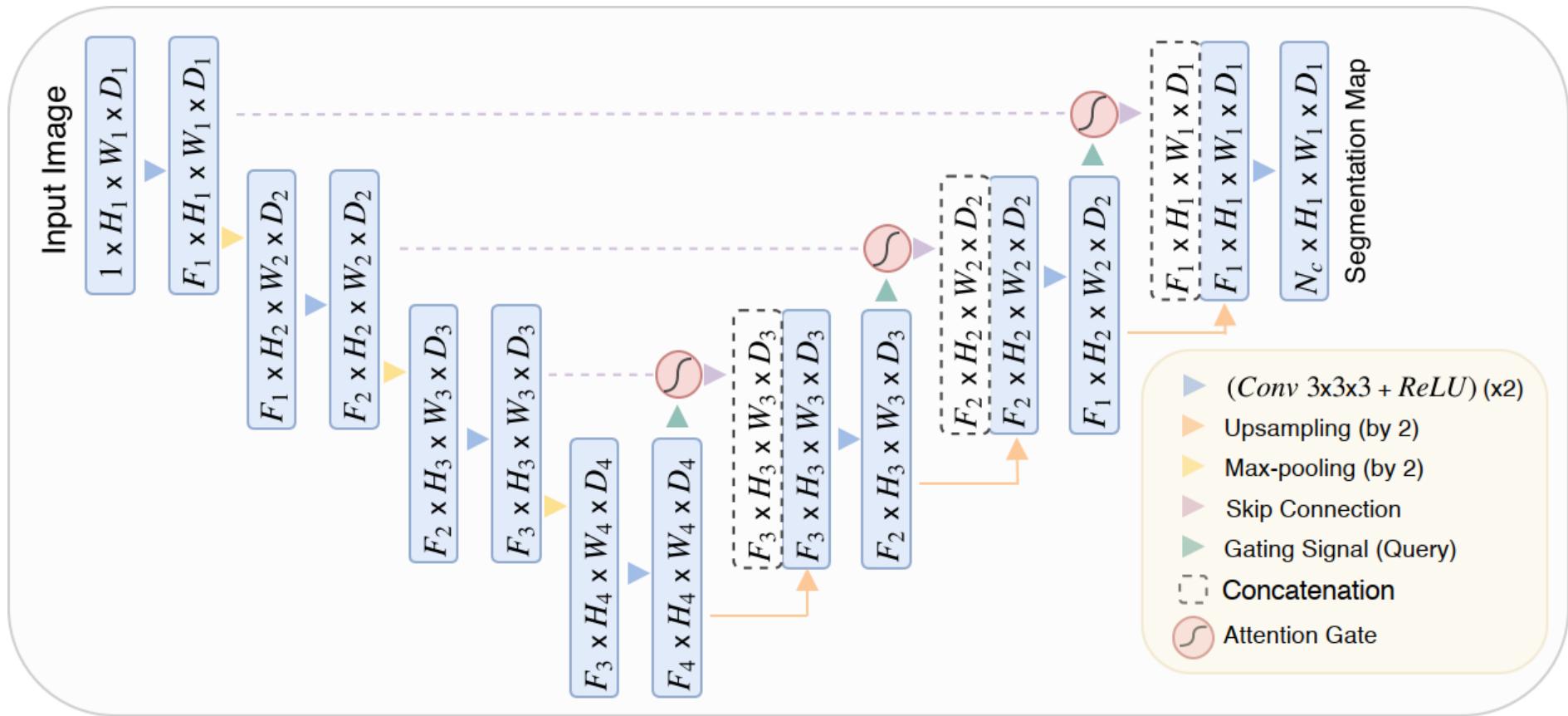
## Dense U-Net Architecture



[Zhou et al., 2018]

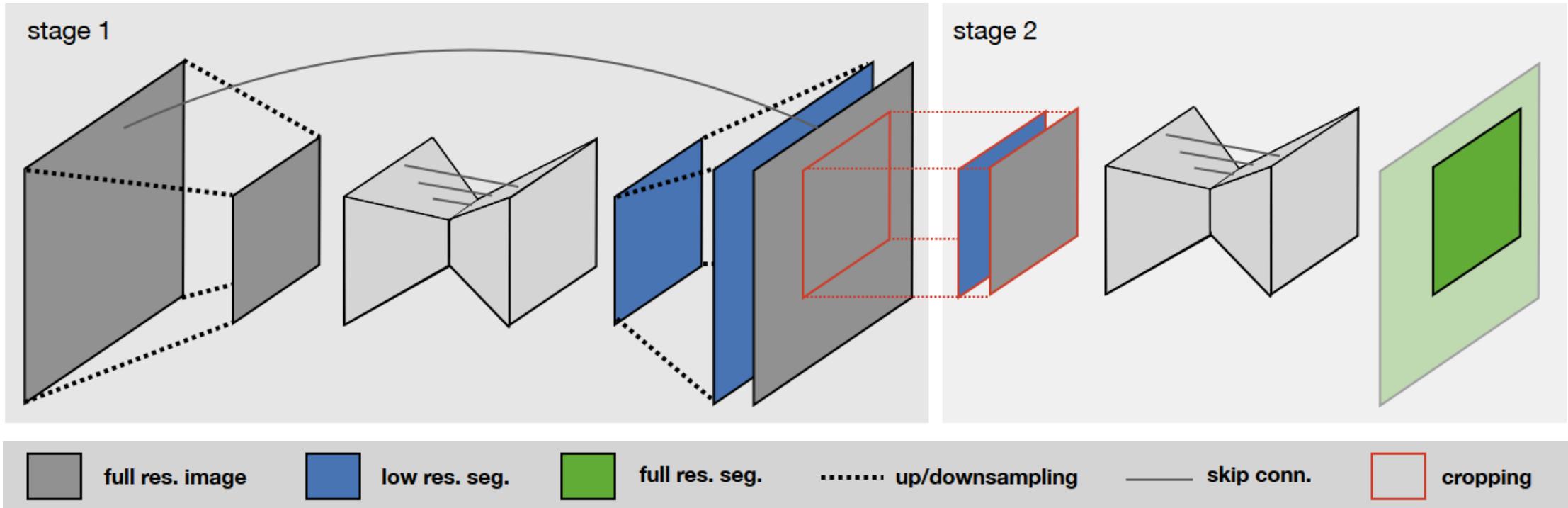
## UNet++ Architecture

- Attention U-Net: "Attention U-Net: Learning Where to Look for the Pancreas" [Oktay et al., 2018]
  - Adds attention gates to the skip connections, helping focus on relevant features and suppress irrelevant ones
- nnU-Net (no-new-Net): "nnU-Net: Self-adapting Framework for U-Net-Based Medical Image Segmentation" [Isensee et al., 2018]
  - Automatically configures and optimizes U-Net-based models for various datasets and segmentation tasks



[Oktay et al., 2018]

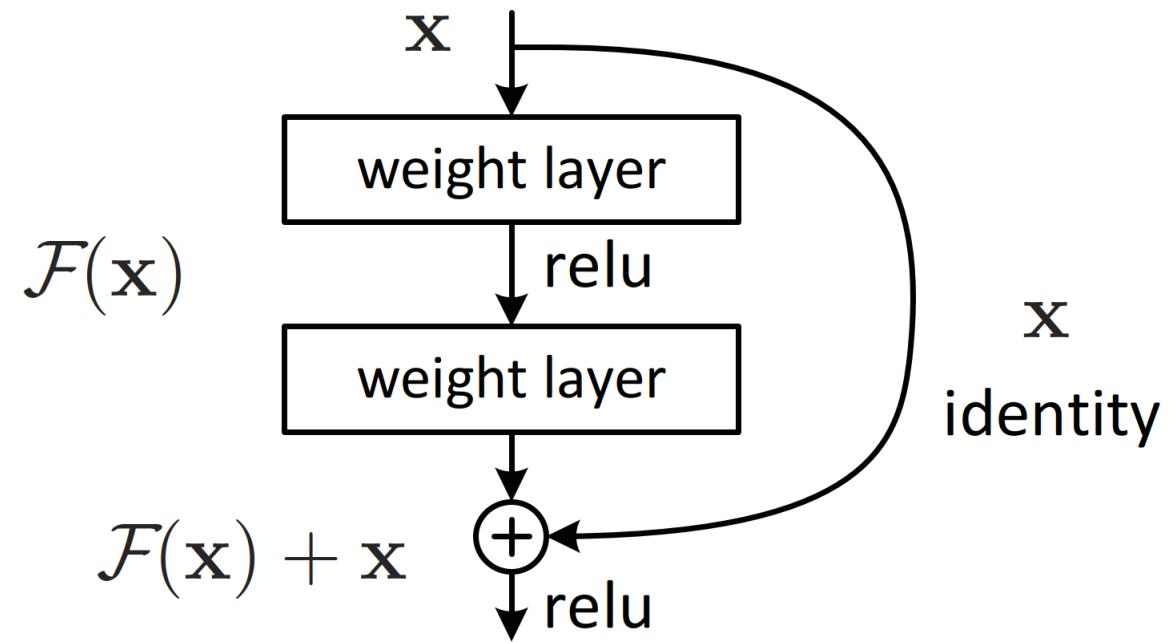
## Attention U-Net Architecture



[Isensee et al., 2018]

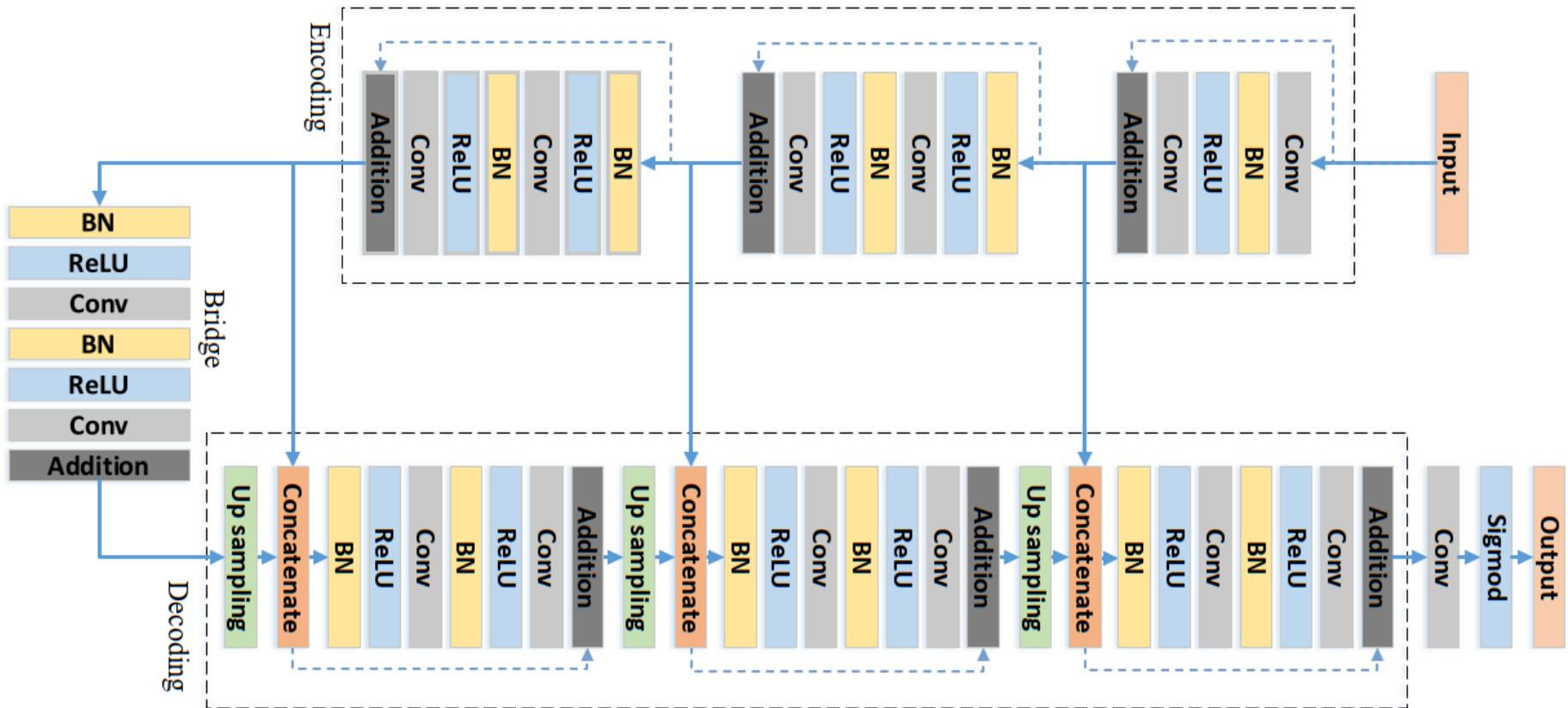
## Cascaded U-Nets as One Possible Configuration Within nnU-Net

- ResNet-based architectures
  - ResNet: "Deep Residual Learning for Image Recognition" [He et al., 2015]
    - Introduces residual learning with skip connections to train very deep networks, primarily for image classification tasks
  - ResUNet / Residual U-Net: "Road Extraction by Deep Residual U-Net" [Zhang et al., 2018]
    - Combines the U-Net architecture with residual learning, incorporating both U-Net style skip connections and ResNet style residual connections
  - ResUNet++: "ResUNet++: An Advanced Architecture for Medical Image Segmentation" [Jha et al., 2019]
    - Enhances ResUNet by adding squeeze and excitation blocks, attention blocks, and a feature fusion mechanism to capture more contextual information



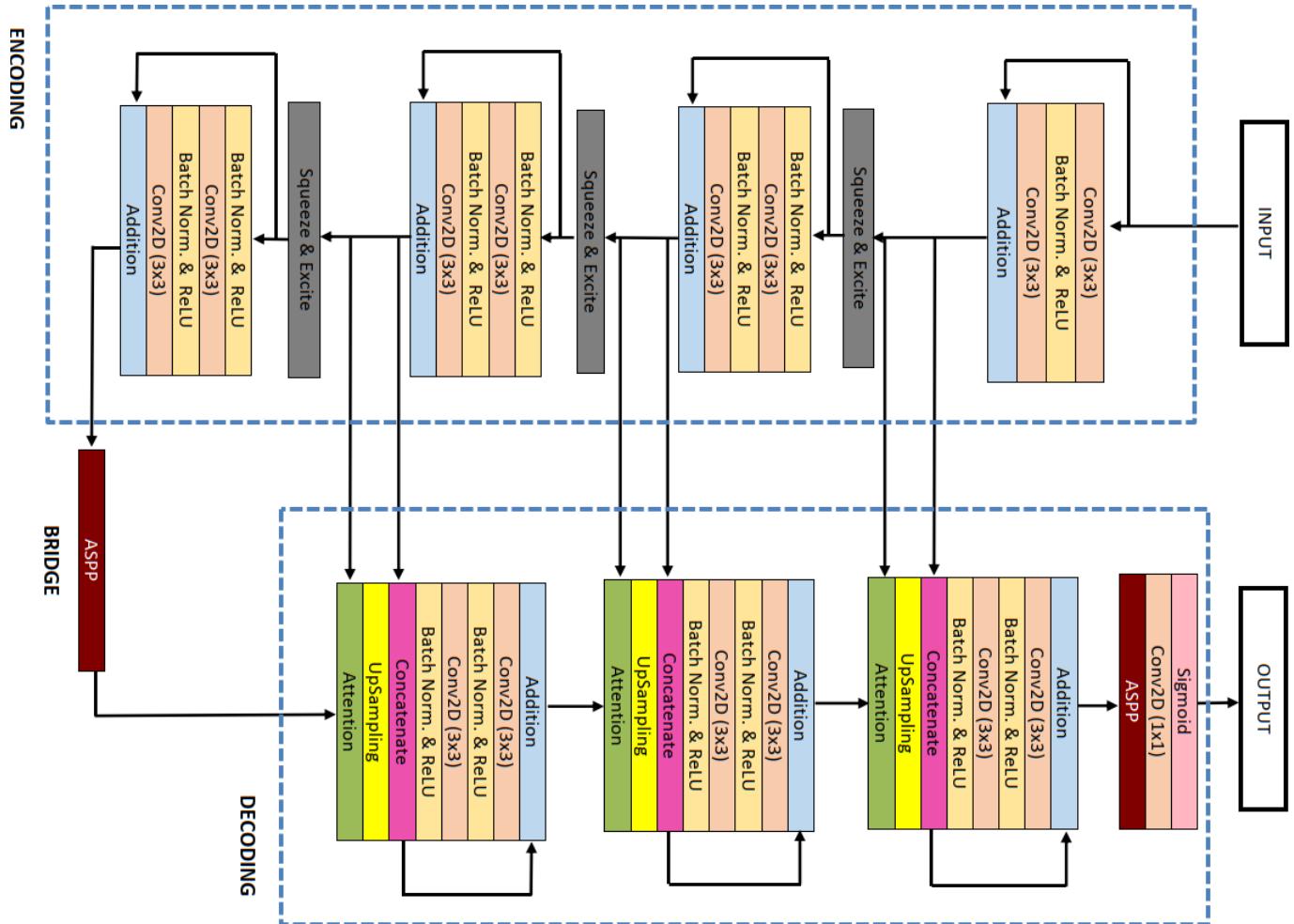
[He et al., 2015]

## Building Block for Residual Learning in ResNet



[Zhang et al., 2018]

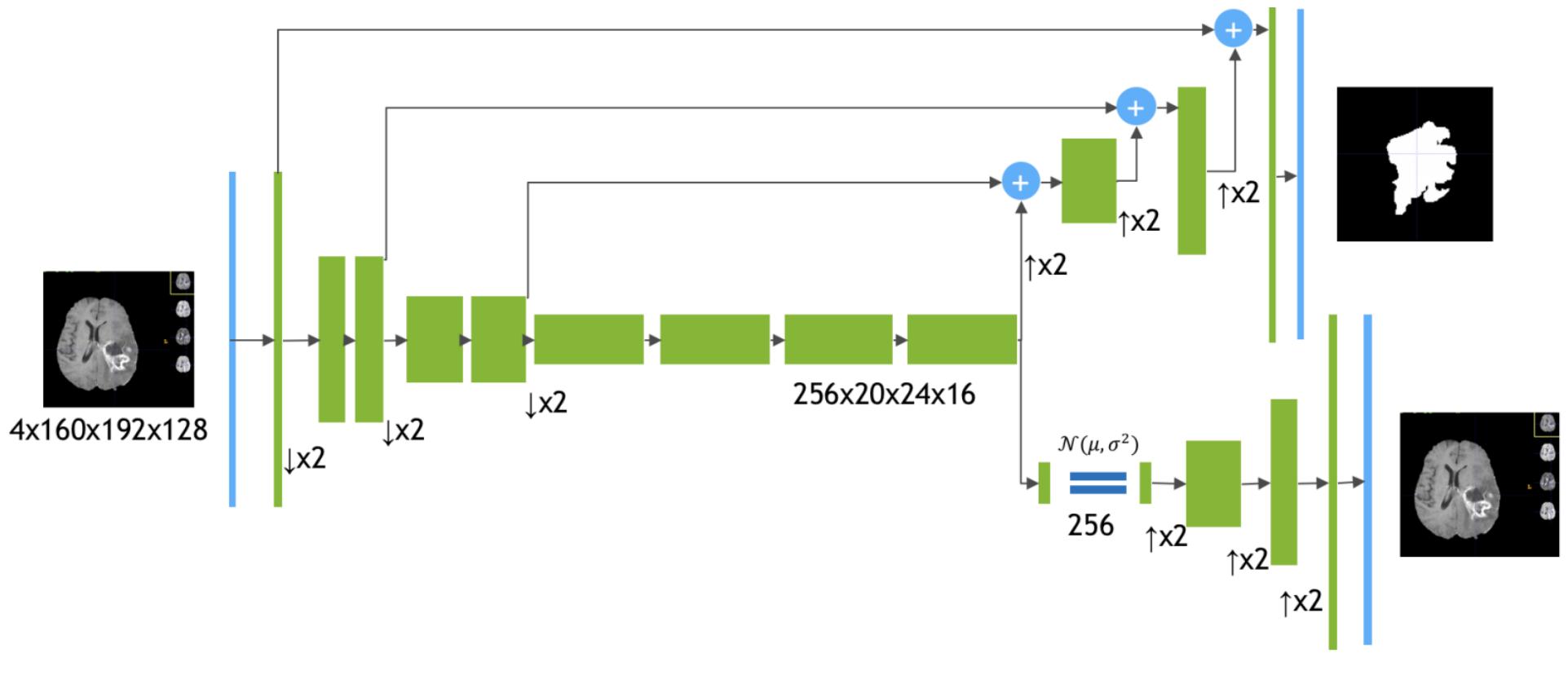
## ResUNet Architecture



[Jha et al., 2019]

## ResUNet++ Architecture

- SegResNet: "3D MRI Brain Tumor Segmentation Using Autoencoder Regularization" [\[Myronenko, 2018\]](#)
  - Adapts the ResNet architecture for segmentation tasks, incorporating both down-sampling and up-sampling paths with residual connections



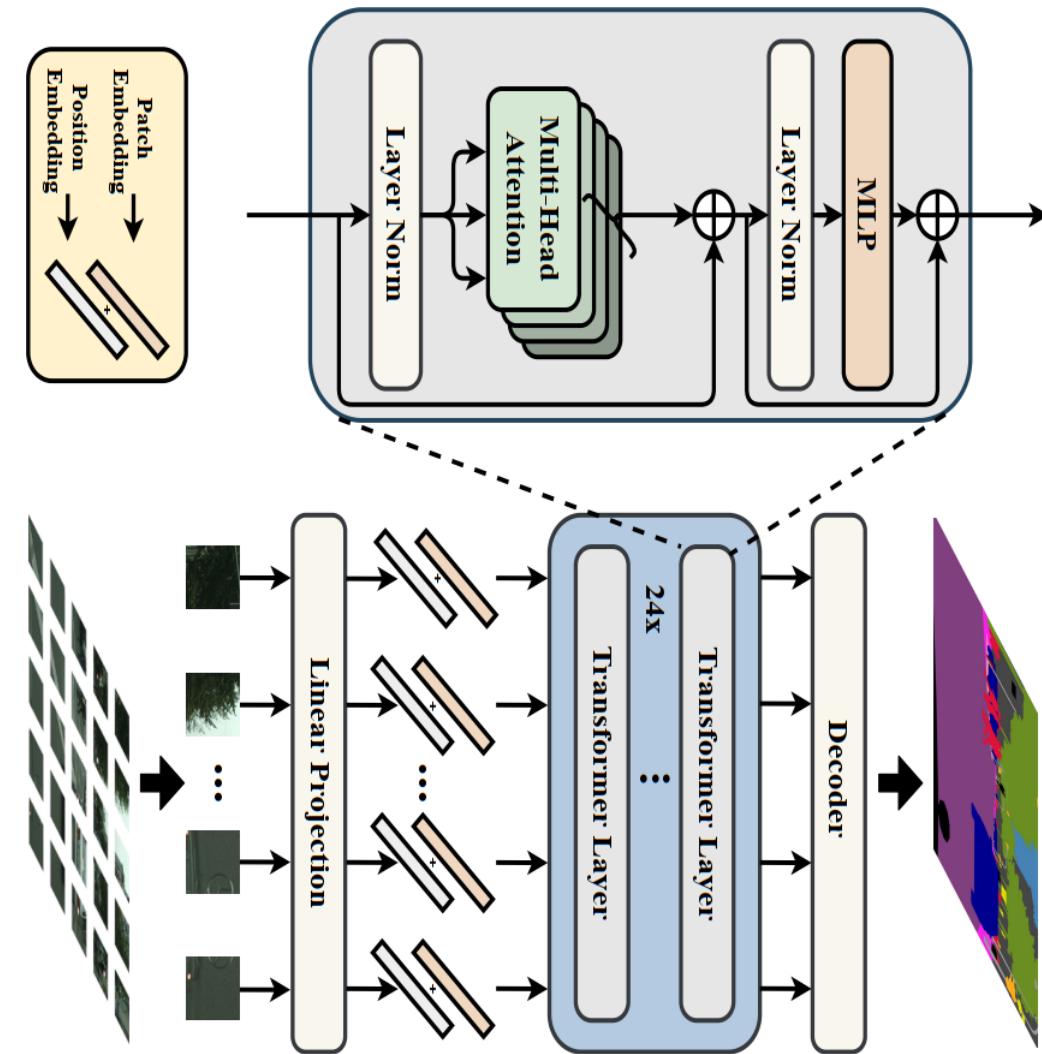
The diagram illustrates the dual-path residual block architecture. It consists of two parallel paths. The left path starts with a green square input, followed by a 'Group Norm' layer, a 'ReLU' activation, a 'Conv3x3x3' layer, another 'Group Norm' layer, a second 'ReLU' activation, and a final 'Conv3x3x3' layer. The right path is a residual connection that adds the output of the first 'Conv3x3x3' layer to the output of the last 'Conv3x3x3' layer via a blue circle labeled '+'. A large bracket groups the first five layers of each path.

$\downarrow x_2 = \text{conv}3x3x3$  stride 2  
 $\uparrow x_2 = \text{conv}1x1x1$ , 3D bilinear upsizing

[Myronenko, 2018]

## SegResNet Architecture

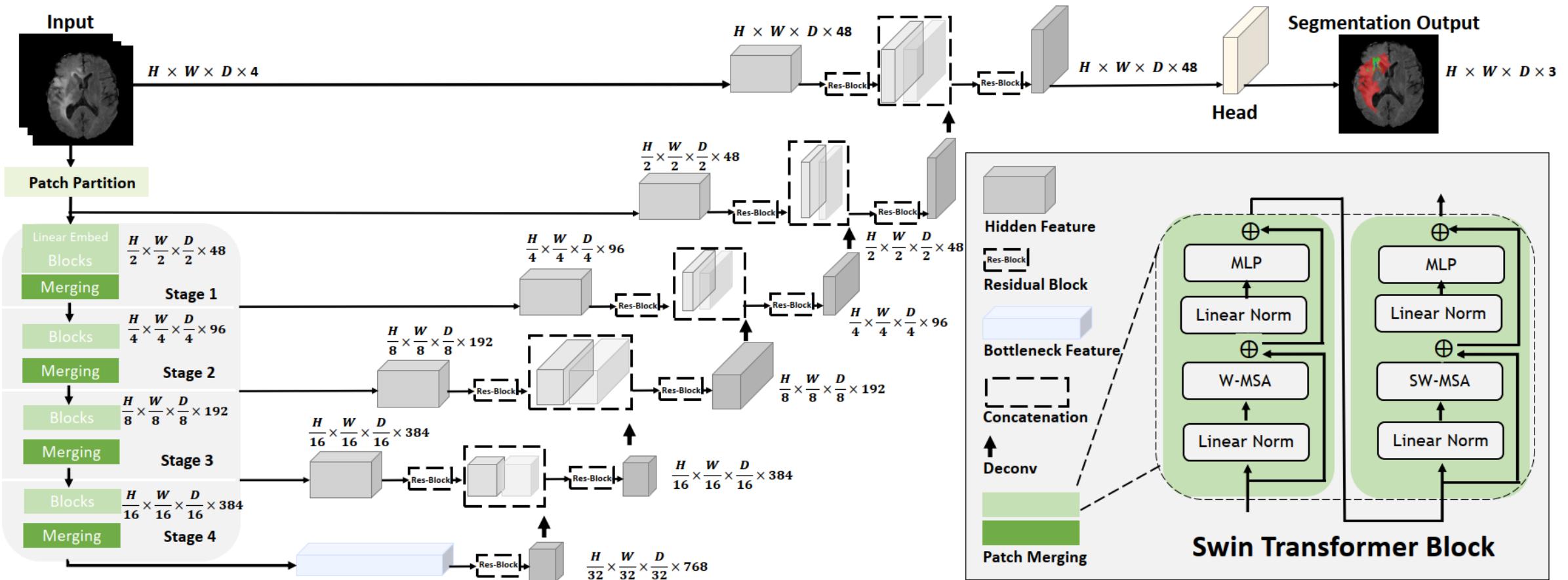
- Transformer-based architectures: Transformer-dominant architectures
  - Transformer as primary component (minimal CNN usage)
  - SETR (Segmentation Transformer): "Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers" [\[Zheng et al., 2021\]](#)
    - First transformer-dominant approach for dense prediction
    - Uses a pure transformer-based architecture by treating semantic segmentation as sequence-to-sequence prediction
    - Encoder: ViT (Vision Transformer) | Decoder: Minimal CNN



**SETR Architecture**

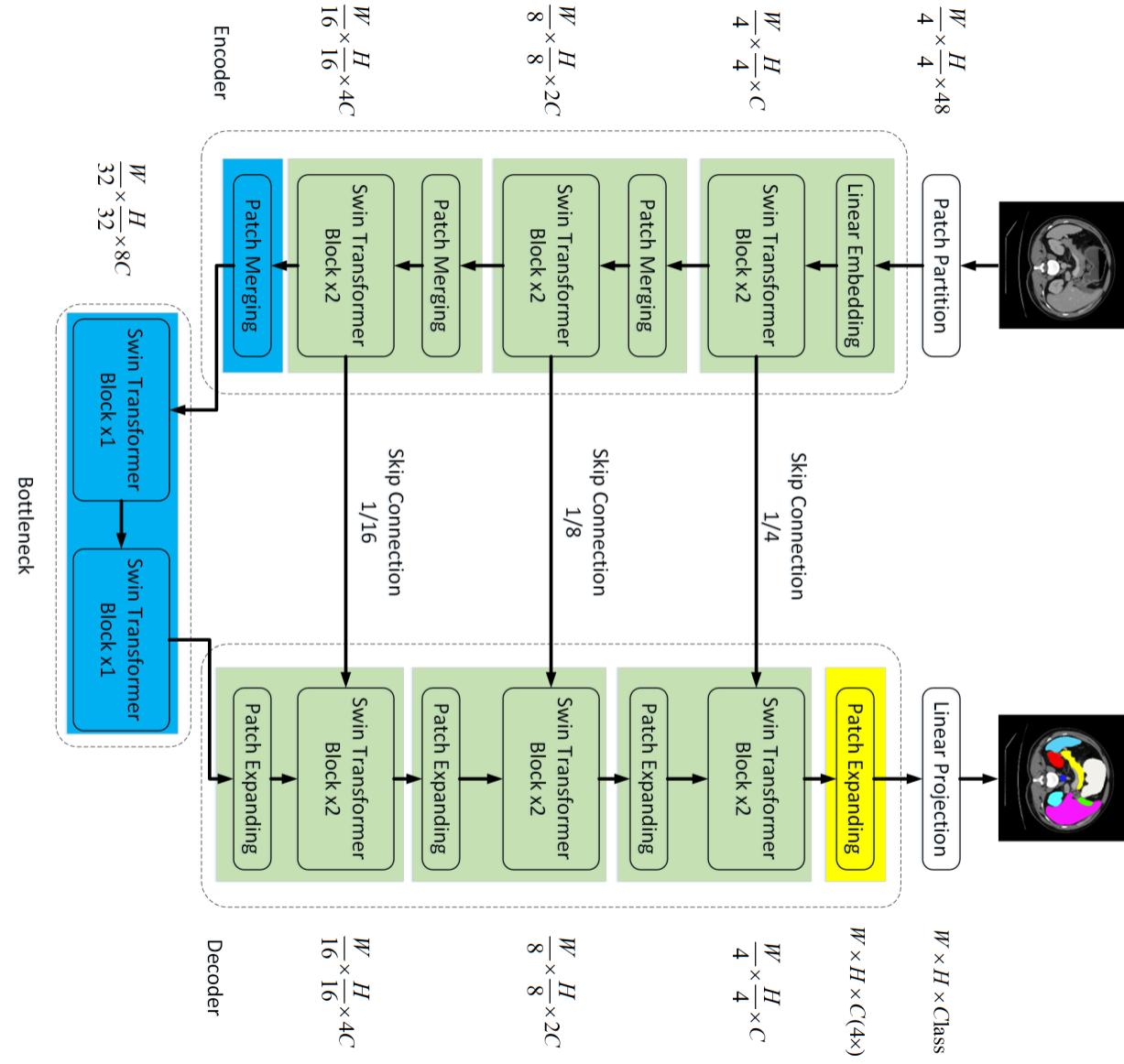
[Zheng et al., 2021]

- Swin UNETR: "Swin UNETR: Swin Transformers for Semantic Segmentation of Brain Tumors in MRI Images" [Hatamizadeh et al., 2021]
  - First to apply hierarchical Swin (Shifted Window) Transformer to 3D medical imaging
  - Addresses long-range dependency limitations of CNNs using hierarchical Swin Transformer for 3D images
  - Encoder: Swin Transformer | Decoder: U-Net-like
- Swin-Unet: "Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation" [Cao et al., 2022]
  - Adapts Swin Transformer for 2D images in a U-Net-like structure
  - Encoder: Swin Transformer | Decoder: Swin Transformer + minimal CNN



[Hatamizadeh et al., 2022]

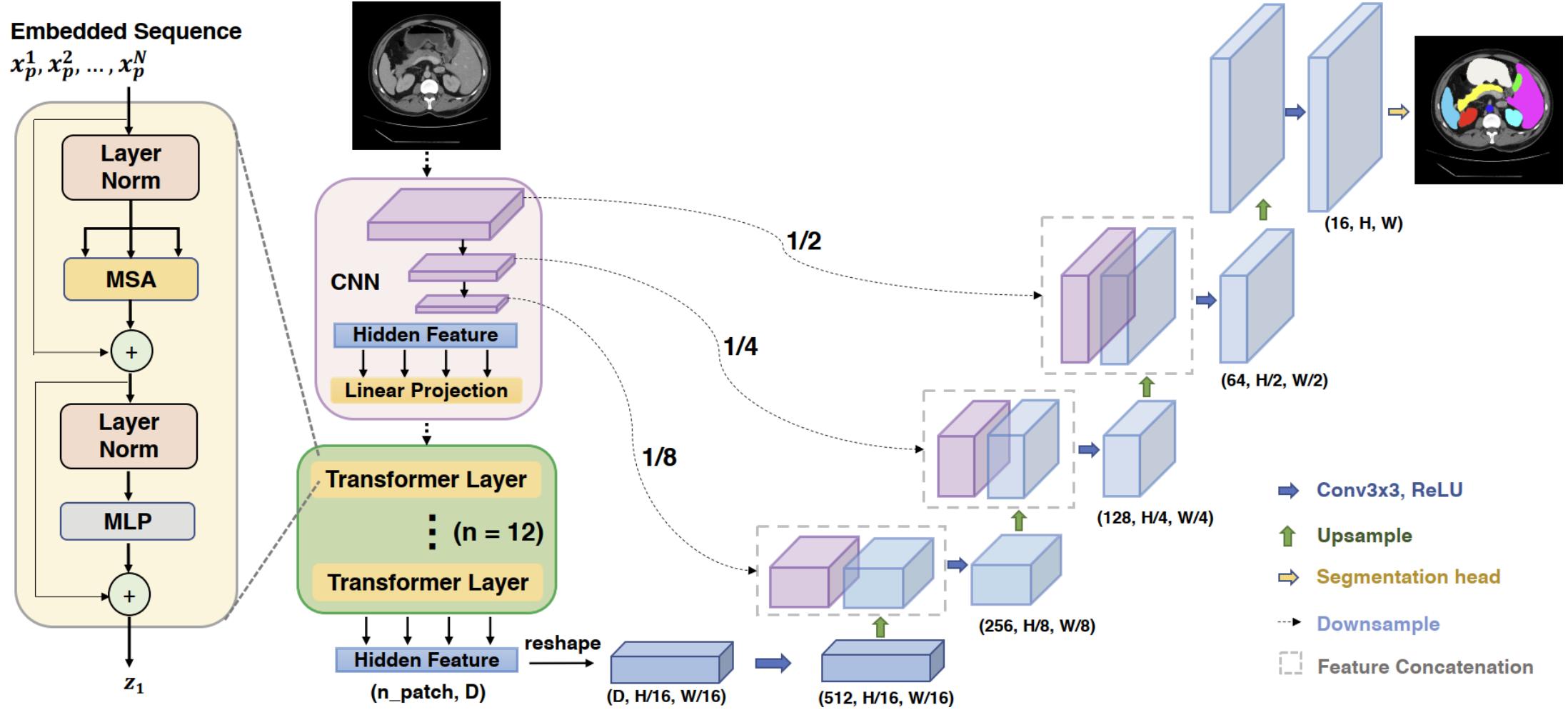
## Swin UNETR Architecture



[Cao et al., 2021]

## Swin-UNet Architecture

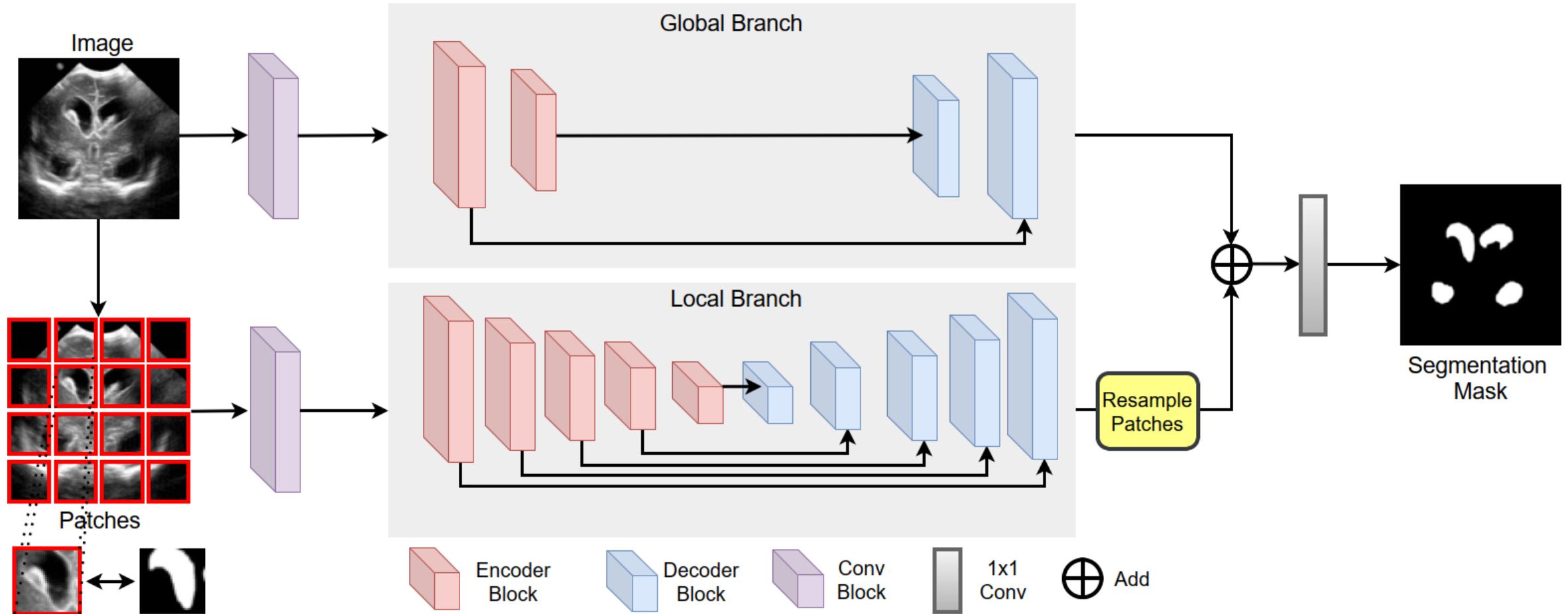
- Transformer-based architectures: CNN-Transformer hybrid architectures
  - Substantial integration of both CNN and Transformer, moving toward efficient hybrids rather than pure architectures
  - TransUNet: "TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation" [\[Chen et al., 2021\]](#)
    - First successful balanced CNN-Transformer hybrid
    - Introduces a symmetric encoder-decoder structure combining CNN and Transformer blocks
    - Encoder: ResNet-like | Decoder: CNN with Transformer bottleneck



[Chen et al., 2021]

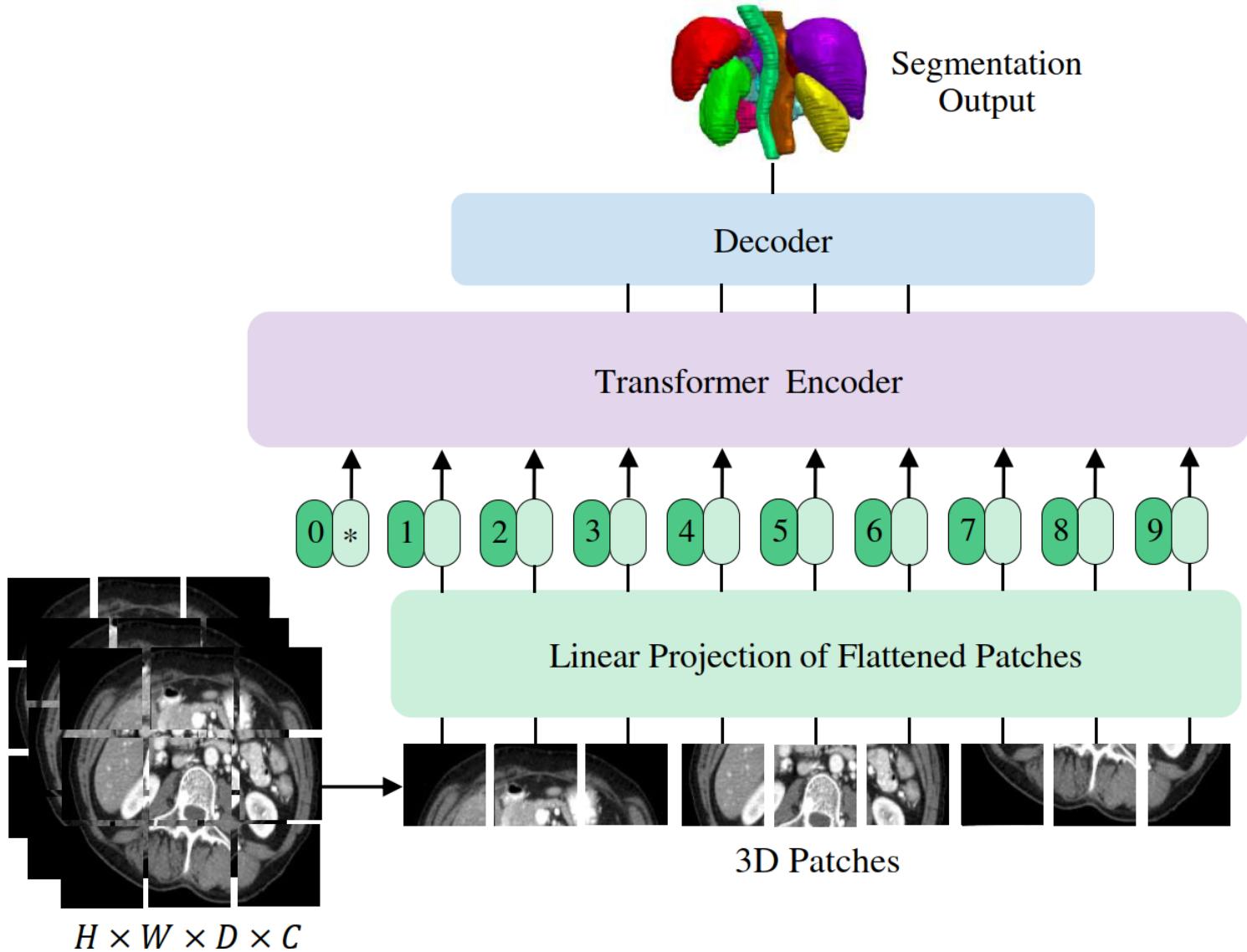
## TransUNet Architecture

- MedT (Medical Transformer): "Medical Transformer: Gated Axial-Attention for Medical Image Segmentation" [Valanarasu et al., 2021]
  - Uses a transformer-based architecture designed specifically for medical image segmentation with gated axial-attention
  - Encoder: Interleaved CNN + Transformer | Decoder: Interleaved CNN + Transformer
- UNETR (Unet Transformers): "UNETR: Transformers for 3D Medical Image Segmentation" [Hatamizadeh et al., 2022]
  - Uses transformer encoder and CNN decoder with skip connections for 3D medical images
  - Encoder: ViT | Decoder: CNN with skip connections



[Valanarasu et al., 2021]

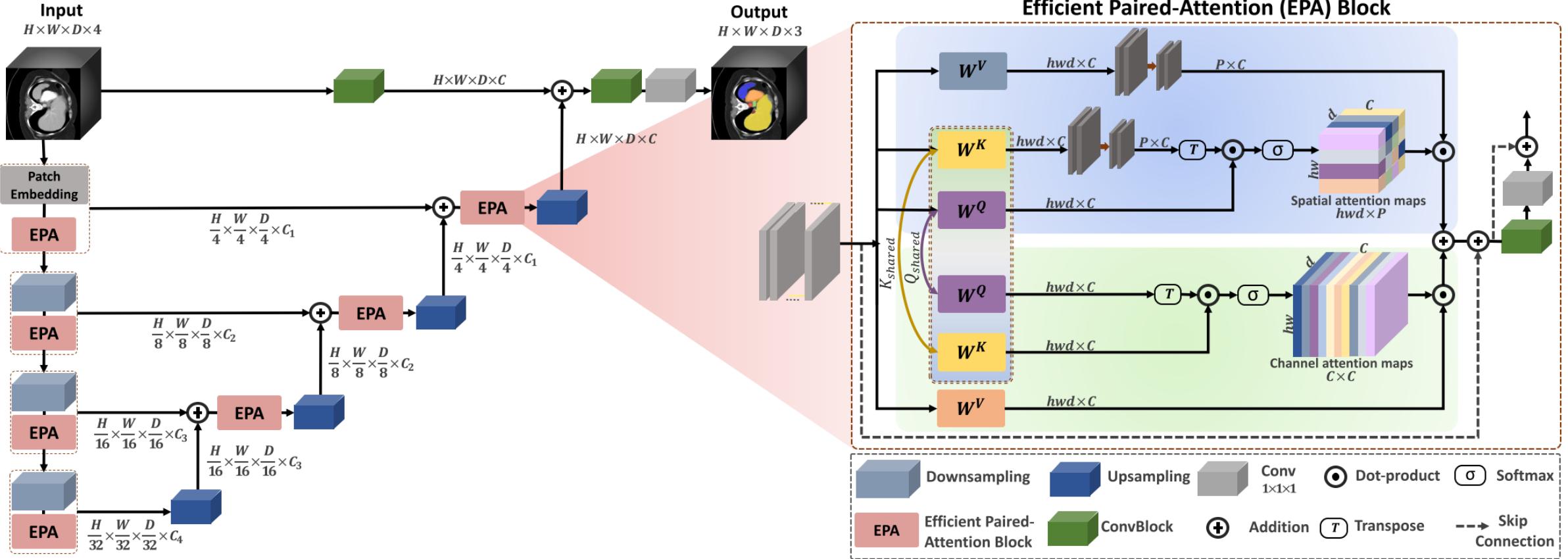
## MedT Architecture



[Hatamizadeh et al., 2022]

## UNETR Architecture

- UNETR++: "UNETR++: Delving Into Efficient and Accurate 3D Medical Image Segmentation" [Shaker et al., 2024]
  - Builds upon UNETR by incorporating efficient paired attention block for reducing computational complexity
  - Encoder: ViT | Decoder: Enhanced CNN with better fusion



[Shaker et al., 2024]]

## UNETR++ Architecture

# Foundation Models

- Large-scale AI models trained on vast amounts of data that serve as a foundation for various downstream tasks
  - First defined in the paper "On the Opportunities and Risks of Foundation Models" [\[Bommasani et al., 2021\]](#)
- Initially emerged from the natural language processing (NLP) field and expanded later to other domains (vision, audio, etc.)

- Why "foundation"?
  - Act as a foundation/basis for various applications
  - Serve as a building block for specialized models
  - Provide fundamental understanding of data patterns
  - Form the base for transfer learning
- Characteristics
  - Large-scale pre-training
  - Self-supervised or semi-supervised learning
  - Adaptability to multiple downstream tasks
  - Transfer learning capabilities
  - Zero/few-shot learning abilities

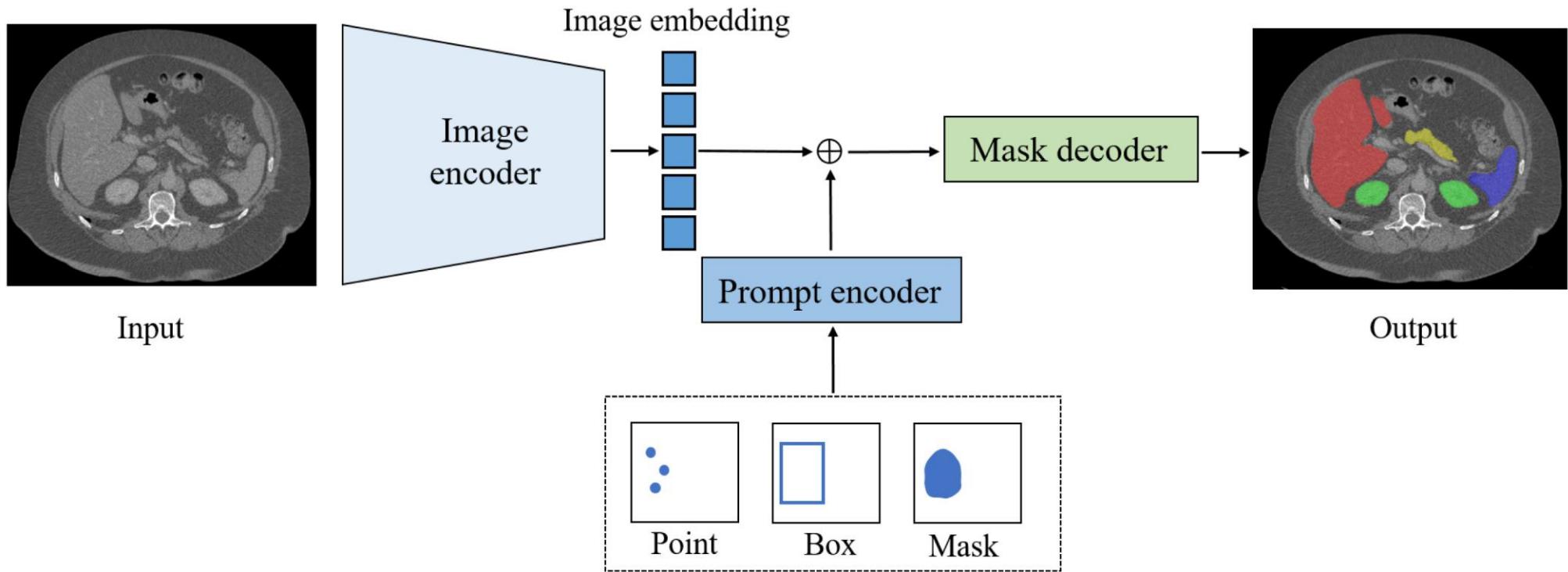
- Impact on AI development
  - Reduced need for task-specific training
  - Improved generalization
  - Cost-effective deployment
- Common examples
  - Language: BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer)
  - Vision: DALL-E, CLIP (Contrastive Language-Image Pre-training)
  - Multimodal: SAM (Segment Anything Model), SEEM (Segment Everything Everywhere Model)
  - Cross-domain: PaLM (Pathways Language Model), Gato

# SAM

- First foundation model for image segmentation
- Initially developed and introduced by Meta AI (formerly Facebook AI Research): "Segment Anything" [\[Kirillov et al., 2023\]](#)
- Released as open source in April 2023
  - Available on GitHub [\[https://github.com/facebookresearch/segment-anything\]](https://github.com/facebookresearch/segment-anything)

- Key features
  - Zero-shot segmentation capabilities
  - Prompt-based interface
  - Trained on Segment Anything 1 Billion masks (SA-1B, 1 billion masks extracted from 11 million images) dataset
  - Versatile input prompts (points, boxes, text)
- Notable developments
  - Medical SAM
  - Fast SAM
  - Mobile SAM
  - Other domain-specific variants

- **Architecture**
  - Based on transformer architecture
  - Main components
    - Image encoder
      - Based on ViT
      - Extracts image embeddings
    - Prompt encoder
      - Processes user interactions
      - Handles different prompt modes
    - Mask decoder
      - Fuses image and prompt embeddings
      - Predicts segmentation masks



[Zhang et al., 2024]

## SAM Architecture

# Medical Segmentation Foundation Models

- SAM-based models
  - MedSAM [Ma et al., 2024]
    - Fine-tunes SAM on 1.57M medical image-mask pairs across 10 modalities for universal medical segmentation
  - SAM-Med2D [Cheng et al., 2023]
    - Adapts SAM for 2D medical images using comprehensive medical datasets with improved cross-modal performance

- SAM-Med3D [Wang et al., 2023]
  - Extends SAM to 3D volumetric medical images, trained from scratch on 131K+ 3D masks across 247 categories
- SAMIHS [Wang et al., 2024]
  - Introduces SAM-based parameter-efficient fine-tuning method specifically for intracranial hemorrhage (ICH) segmentation

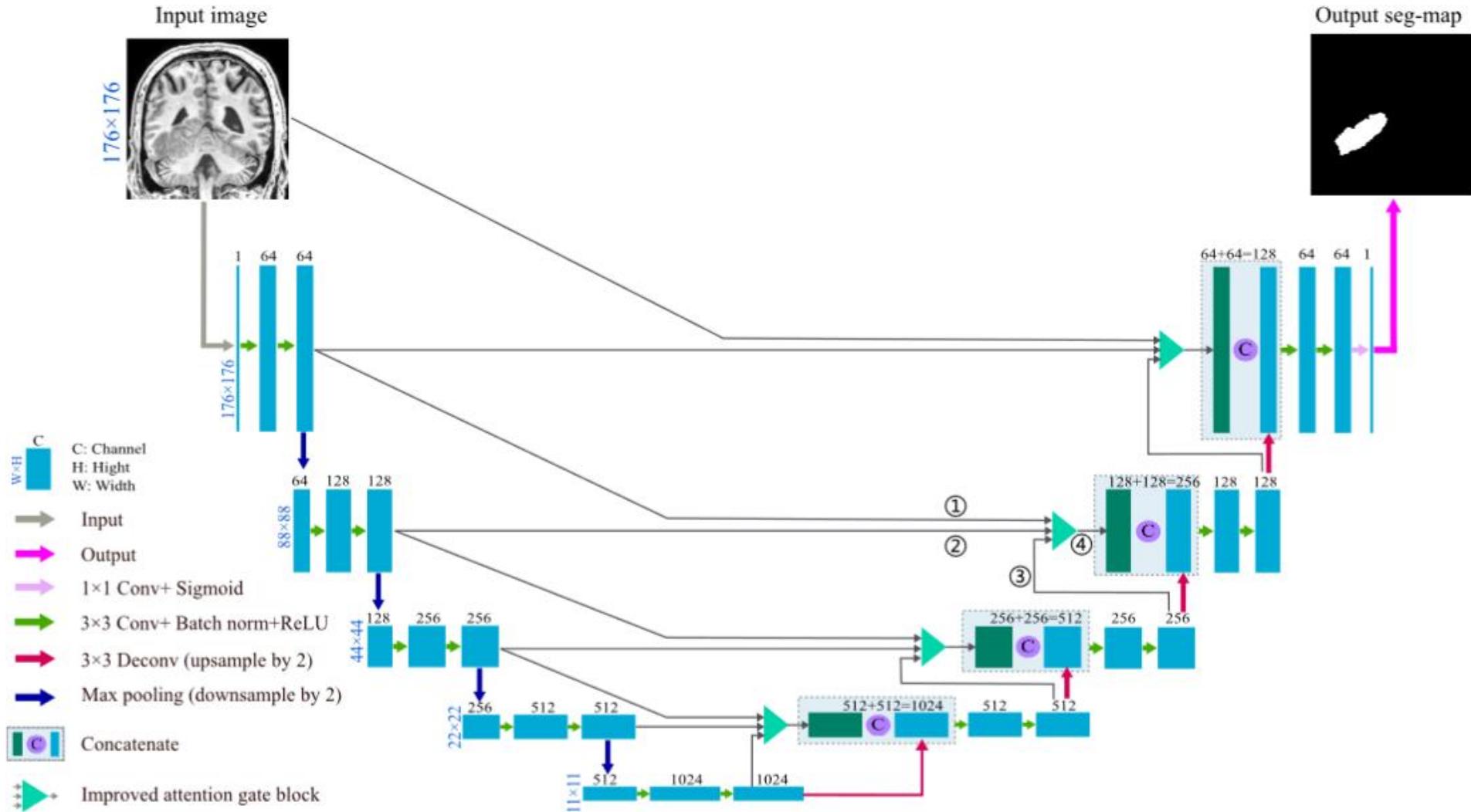
Dataset	BCIHM		Instance		Params(M)	
	Method	Dice	HD95	Dice	HD95	
	U-Net	50.06	3.99	62.07	4.26	7.77
	Att-UNet	54.29	3.89	29.74	7.14	34.88
	U-Net++	53.80	3.92	53.67	4.86	9.16
	TransUNet	46.47	4.05	58.23	4.44	106.17
	TransFuse	52.14	3.86	24.83	7.63	26.57
	H2Former	48.79	4.03	31.12	5.61	33.86
	SAM	49.32	4.29	61.46	5.04	N/A
	MedSAM	51.38	4.51	51.38	4.51	N/A
	SAMed	66.13	3.56	74.99	3.77	3.93
	SAMUS	60.29	3.85	43.85	5.46	42.60
	MSA	67.08	3.53	72.65	3.98	11.17
	SAMIHS	<b>69.77</b>	<b>3.31</b>	<b>76.52</b>	<b>3.71</b>	4.24

- Alternative foundation models
  - VISTA3D [He et al., 2024]
    - NVIDIA's specialized interactive foundation model trained on 11,454 3D CT volumes supporting 127 anatomical classes with state-of-the-art automatic and interactive segmentation
  - MedSegX [Zhang et al., 2025]
    - Open-world medical segmentation foundation model using ConMoAE (Contextual Mixture of Adapter Experts), trained on tree-structured MedSegDB covering 39 organs and tissues with strong out-of-distribution performance

# Lesion Segmentation Performance for ATLAS Dataset

- U-Net [Chen et al., 2018]
  - Dice similarity coefficient (DSC) = 0.50
    - ATLAS R1.2 dataset
    - Slice input ( $128 \times 128$  or  $256 \times 256$ )

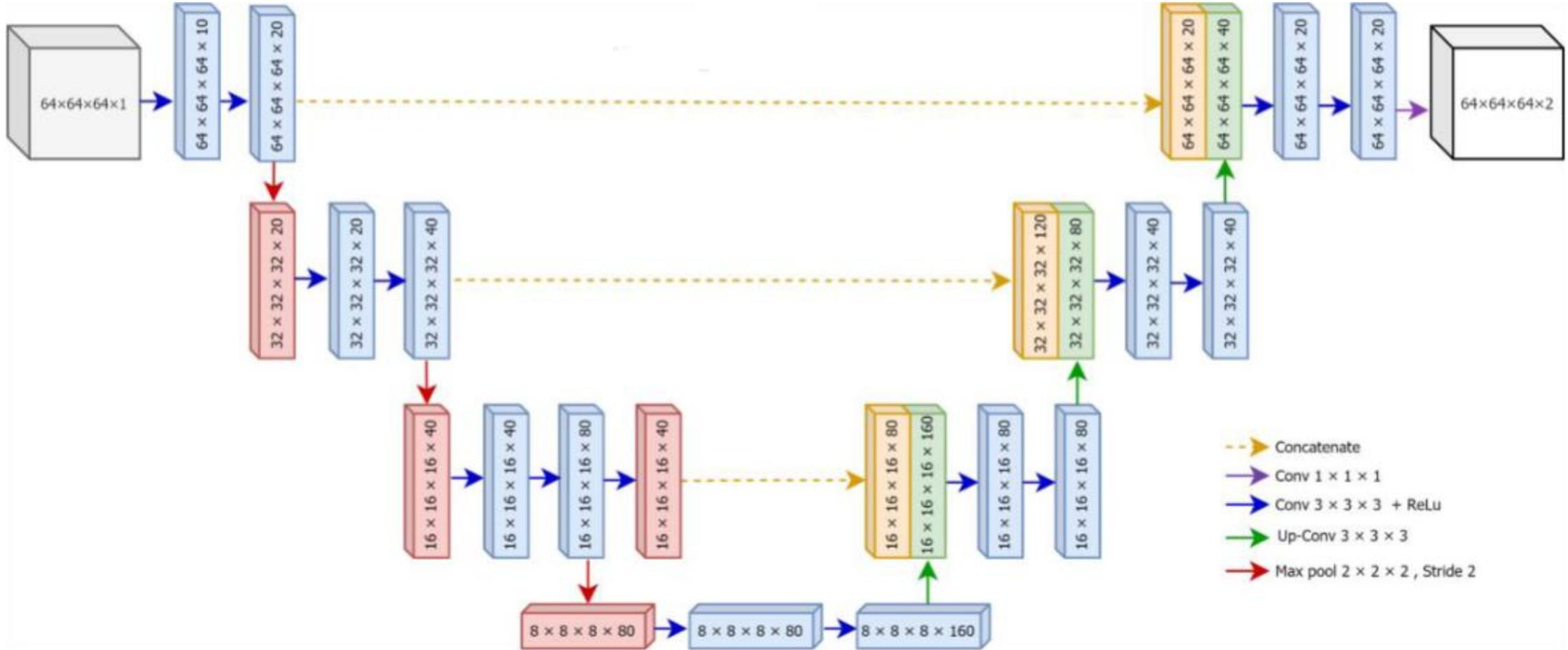
- Attention U-Net [Hui et al., 2020]
  - Partitioning-stacking prediction fusion
  - DSC = 0.593
    - ATLAS R1.2 dataset
    - Slice input ( $176 \times 176$ )



[Hui et al., 202]0

## Proposed Attention U-Net-based Architecture

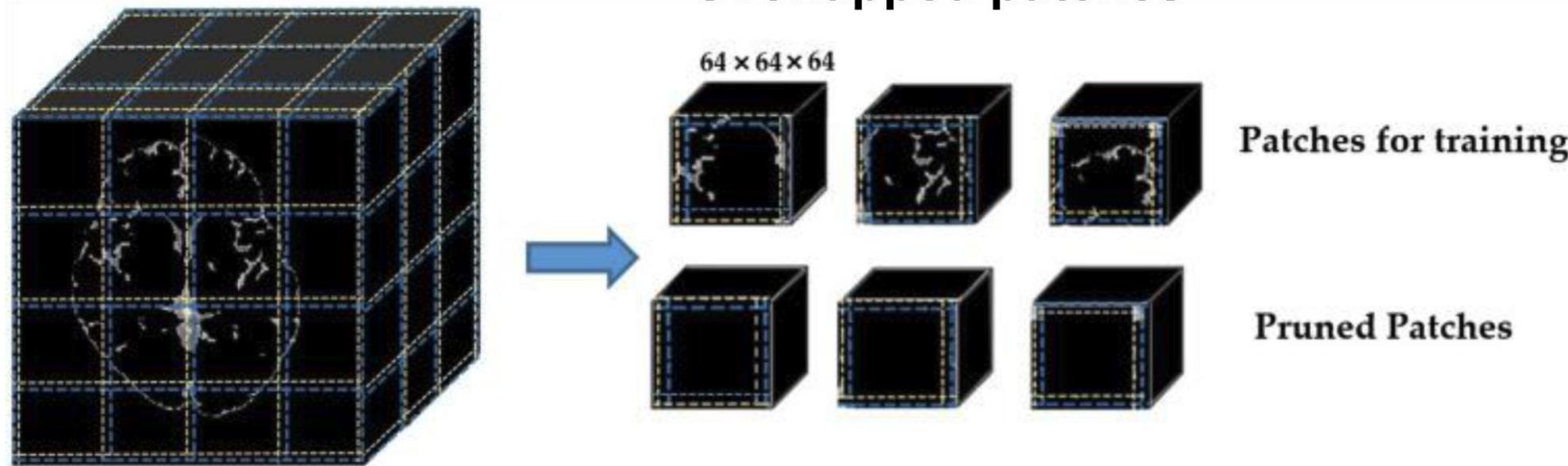
- 3D U-Net [Paing et al., 2021]
  - Variational mode decomposition for preprocessing
  - Overlapped patches strategy
  - DSC = 0.668
    - ATLAS R1.2 dataset
    - Patch input ( $64 \times 64 \times 64$ )



[Paing et al., 2021]

## Proposed 3D U-Net-based Architecture

## Overlapped patches

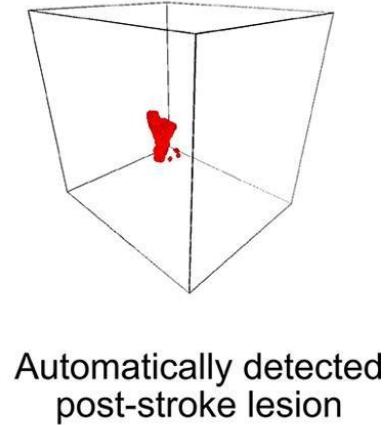
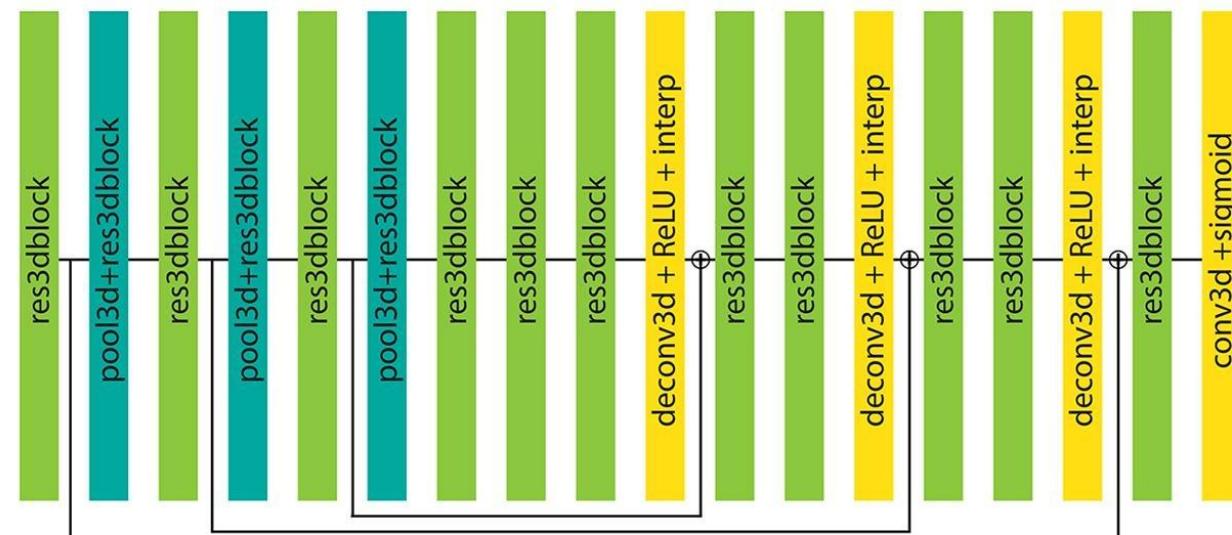
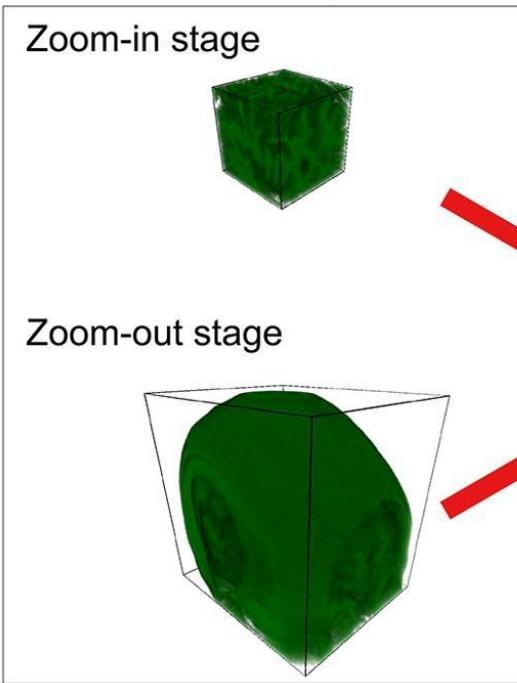


[Paing et al., 2021]

Overlapping Patches Strategy

- Residual U-Net [Tomita et al., 2020]
  - Two-stage zoom-in&out training strategy
  - DSC = 0.64 (0.51 ~ 0.76)
    - ATLAS R1.2 dataset
    - Volume input ( $128 \times 128 \times 128$  for the zoom-in stage and  $144 \times 172 \times 168$  for the zoom-out stage)

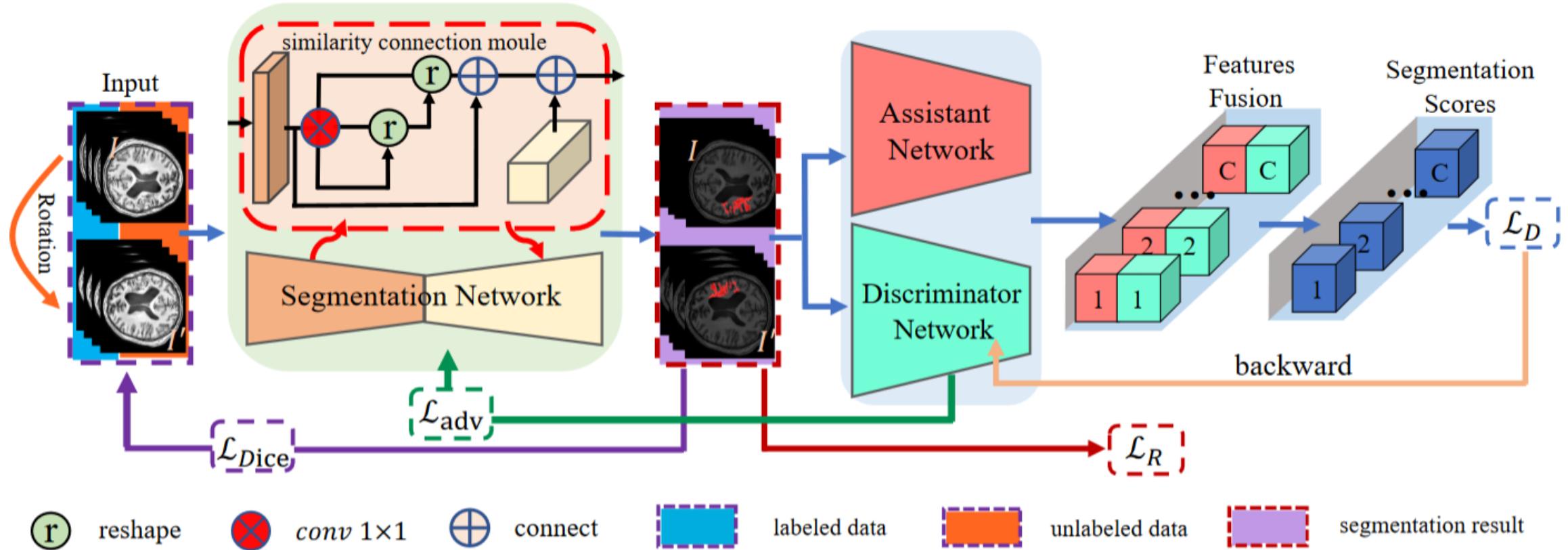
Zoom-in&out training strategy  
for volumetric segmentation



[Tomita et al., 2020]

## Proposed Residual U-Net-based Architecture

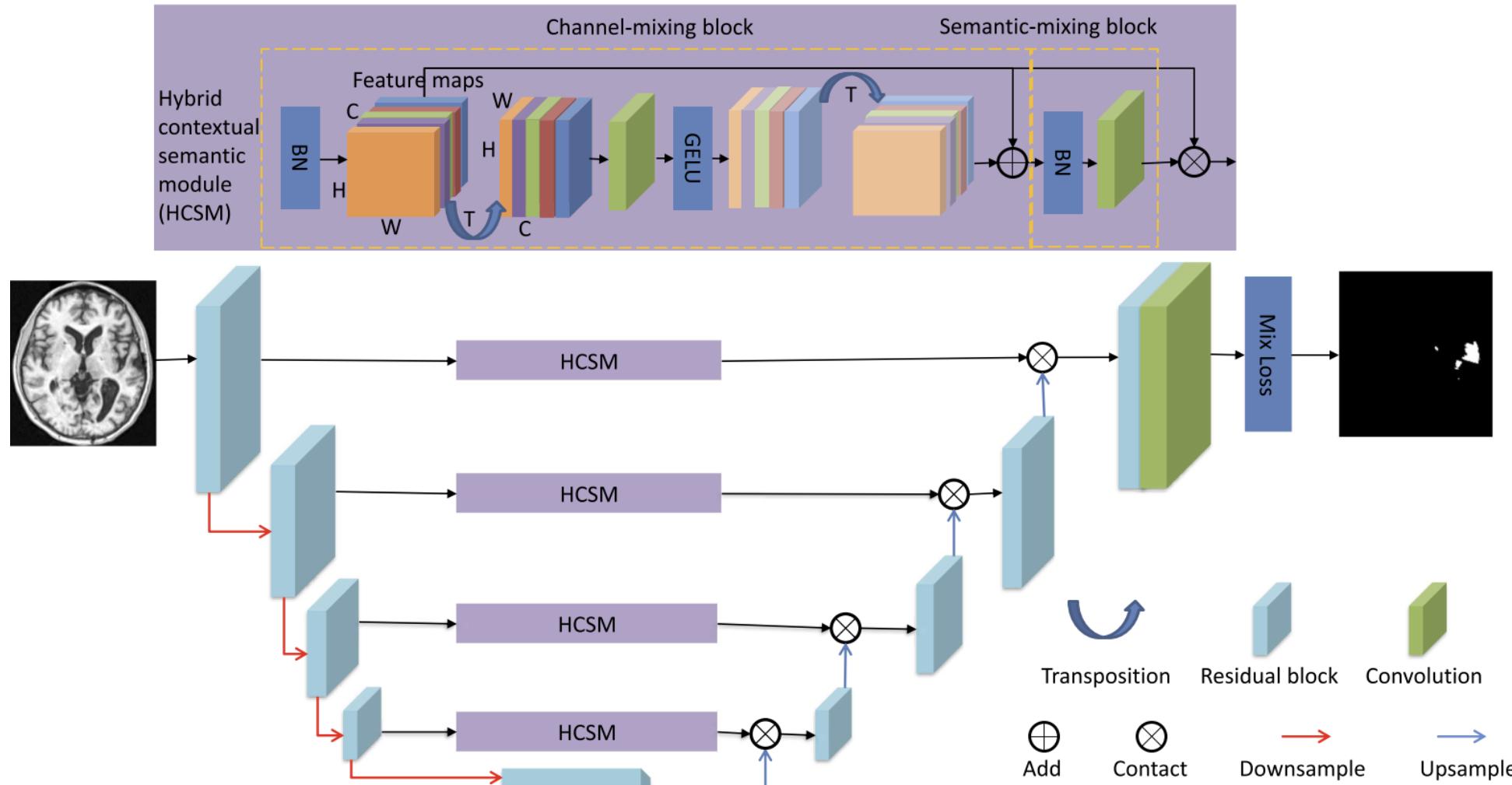
- CPGAN (Consistent Perception Generative Adversarial Network) [Wang et al., 2021]
  - Consistent perception strategy
  - Similarity connection module
  - DSC = 0.617
    - ATLAS R1.2 dataset
    - Slice input ( $256 \times 256$ )



[Wang et al., 2021]

## Proposed CPGAN Architecture

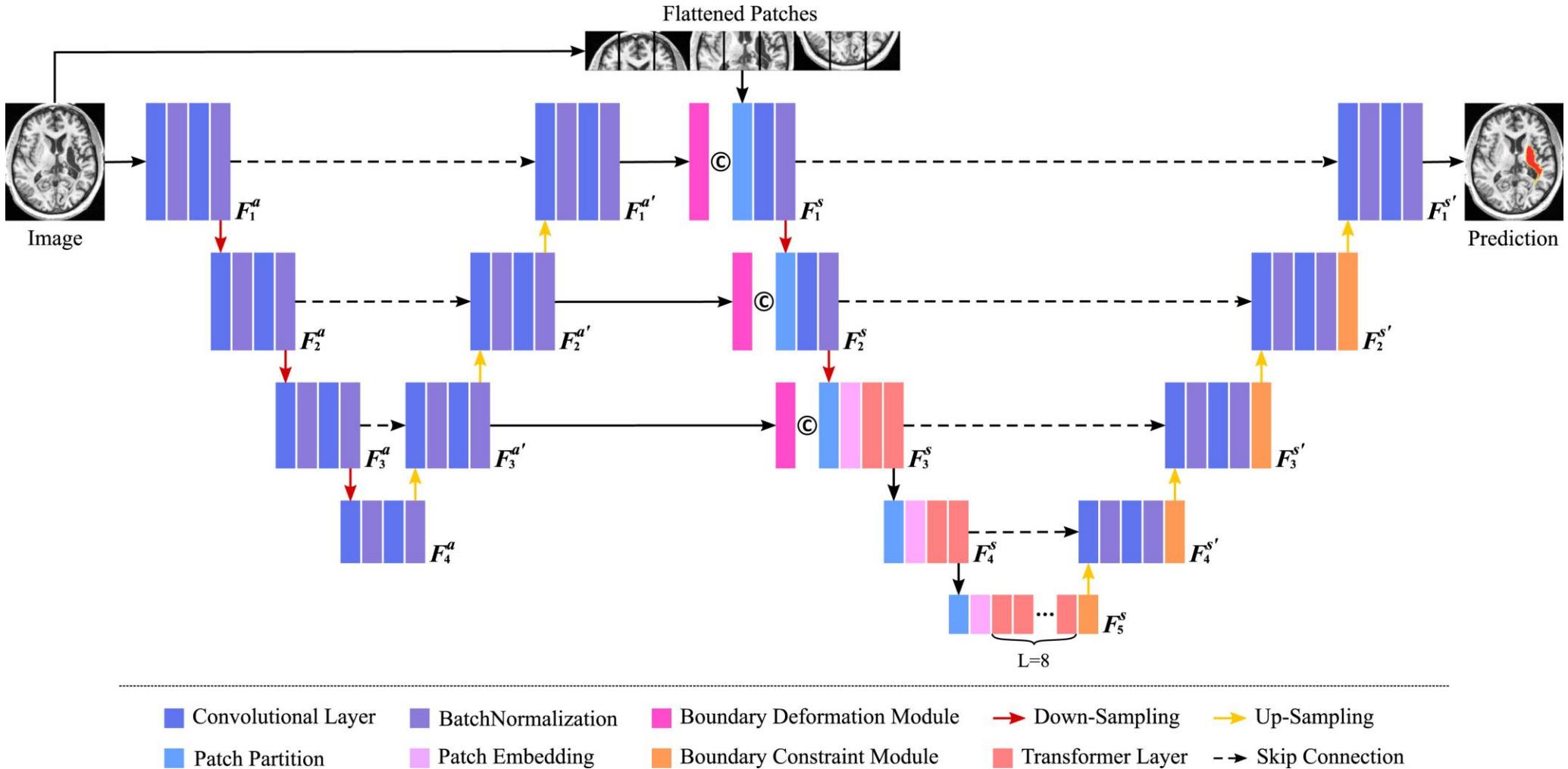
- HCS (Hybrid Contextual Semantic)-Net [Liu et al., 2023]
  - Hybrid contextual semantic module
  - Mixing-loss function for unbalanced small-size lesions
  - DSC = 0.6972
    - ATLAS R2.0 dataset
    - Slice input ( $240 \times 240$ )



[Liu et al., 2023]

## Proposed HCS-Net Architecture

- W-Net [Wu et al., 2023]
  - CNN and Transformer as the backbone network
  - Boundary deformation module
  - Boundary constraint module
  - DSC = 0.6176
    - ATLAS R1.2 dataset
    - Volume input ( $233 \times 197 \times 189$ )



[Wu et al., 2023]

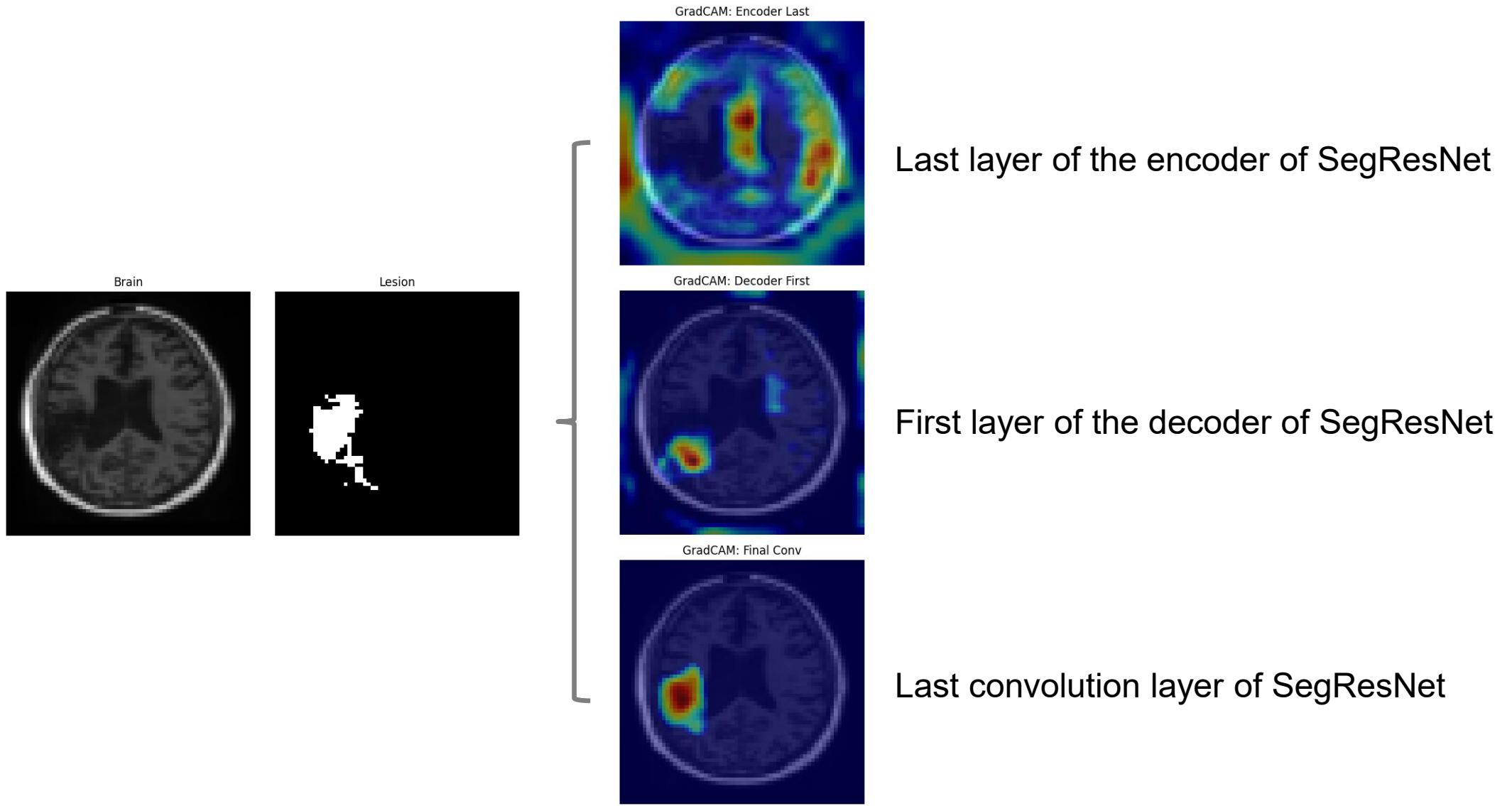
## Proposed W-Net Architecture

# Model Explanations: Class Activation Mapping Techniques

- Class activation mapping (CAM) [Zhou et al., 2016]
  - Highlights discriminative regions for a specific class
  - Requires modification of model architecture (global average pooling (GAP) layer)
  - Each feature map's weight is directly from the final fully connected (FC) layer
  - Only applicable to certain network architectures

- Gradient-weighted CAM (Grad-CAM) [Selvaraju et al., 2017]

- Improvement over CAM
  - No need for model architecture changes
- Uses gradients to determine feature importance
- Each feature map's weight is the global average of gradients of the class score with respect to the feature map
- Applicable to any CNN-based model



## Grad-CAM-derived Feature Maps at Different Layers for Lesion Segmentation

- Grad-CAM++ [Chattopadhyay et al., 2018]
  - Further refinement of Grad-CAM
    - Improved multi-object localization
    - Better for detecting small objects
    - Provides smoother and more complete visualization of target objects
  - Uses gradients to determine feature importance
  - Each feature map's weight is calculated using higher-order derivatives and pixel/voxel-wise weighting, considering the spatial distribution of gradients
  - Applicable to any CNN-based model
  - Higher computational complexity compared to Grad-CAM

- Implementation in MONAI
  - Available classes
    - CAM: `monai.visualize.CAM`
    - Grad-CAM: `monai.visualize.GradCAM`
    - Grad-CAM++: `monai.visualize.GradCAMpp`
  - Supports 2D and 3D medical images
  - Easily integrates with MONAI model pipelines

```
from monai.visualize import CAM, GradCAM, GradCAMpp

cam = GradCAM(nn_module=model, target_layers="target_layer")
result = cam(x=input_tensor)
```

# Demonstration Experiments

- **monai.networks.nets**
  - **UNet**
    - Foundational encoder-decoder architecture with skip connections for medical image segmentation
    - Lesion applicability: Fine detail preservation through skip connections and effectiveness for small lesions
    - Advantages: Simple structure, interpretable design, and baseline performance for most segmentation tasks

## – **VNet**

- 3D-native architecture design with volumetric convolutions and integrated Dice loss
- Lesion applicability: Spatial continuity leverage of 3D lesions in CT/MRI volumes
- Advantages: Dice loss integration for medical segmentation and end-to-end 3D optimization

## – **SegResNet**

- Residual learning integration with segmentation for deep network training
- Lesion applicability: Complex lesion pattern capture through deep structure and contextual relationships
- Advantages: Lightweight yet effective design with vanishing gradient problem resolution

## – **AttentionUNet**

- Attention mechanism introduction for selective feature focusing
- Lesion applicability: Lesion region focus with irrelevant background suppression
- Advantages: Enhanced interpretability through attention visualization and improved feature selection

## – **UNETR**

- ViT encoder applied to 3D medical segmentation with global context modeling
- Lesion applicability: Long-range dependency capture across entire 3D volume (pure global attention)
- Advantages: Comprehensive global modeling and established transformer principles

## – SwinUNETR

- Hierarchical Swin Transformer with shifted window attention for efficient 3D medical segmentation
- Lesion applicability: Efficient processing of lesions with varying sizes and spatial distributions (hierarchical local + global attention)
- Advantages: Computational efficiency through hierarchical feature representation

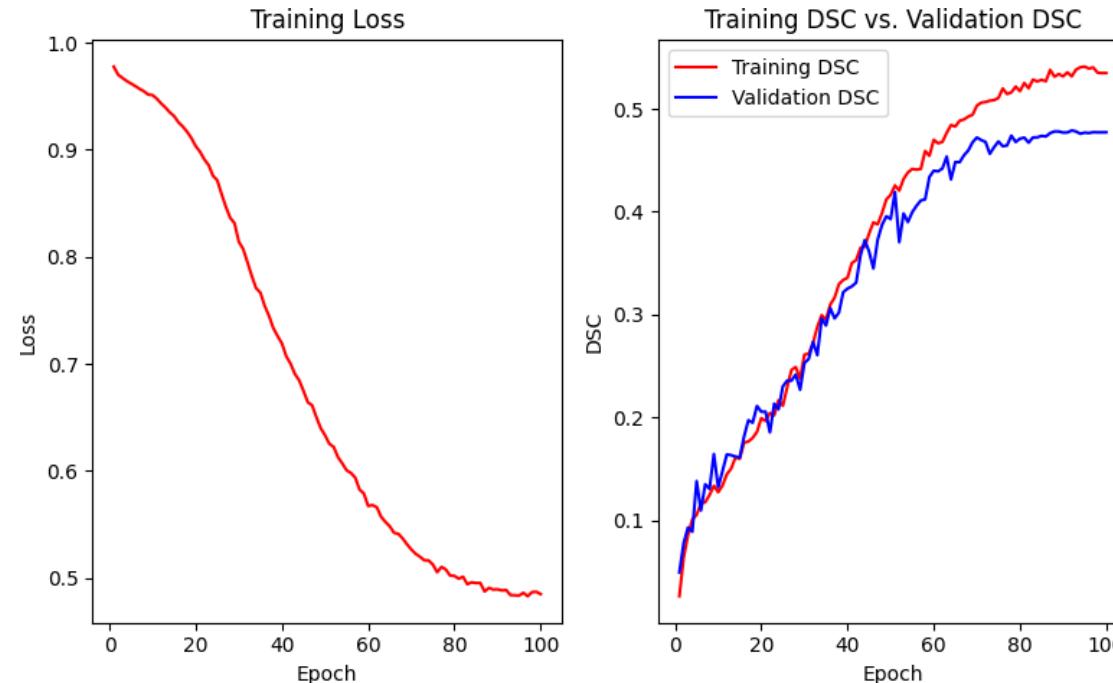
---

<b>Category</b>	<b>Parameters</b>	<b>Models</b>
Small	< 10M	SegResNet, UNet, AttentionUnet,
Medium	10-50M	VNet
Large	50-100M	SwinUNETR
Very large	> 100M	UNETR

---

## Architecture Scale Comparison

- Implementation of SegResNet-based lesion segmentation
  - Input: Brain
  - Number of trainable parameters: 1,176,177
  - Validation set: DSC = 0.479
  - Test set: DSC =  $0.519 \pm 0.239$  (0.000 ~ 0.861)



- Padding-based vs. resizing-based approaches
  - Padding-based approach (SegResNet)
    - Preserves original dimensions through adaptive padding: (98, 116, 94) → (112, 128, 96)
    - Validation set: DSC = 0.479
    - Test set: DSC =  $0.519 \pm 0.239$  (0.000 ~ 0.861)
  - Resizing-based approach (SegResNet)
    - Transforms to fixed dimensions through uniform resizing: (98, 116, 94) → (96, 112, 96)
    - Validation set: DSC = 0.460
    - Test set: DSC =  $0.468 \pm 0.234$  (0.000 ~ 0.841)