DEEP LEARNING FELLOWSHIP

UCTRANSNET: RETHINKING THE SKIP CONNECTIONS IN U-NET FROM A CHANNEL-WISE PERSPECTIVE WITH TRANSFORMER

WANG, ET AL.

ARXIV 2109.04335

PRESENTED BY TED EDMONDS

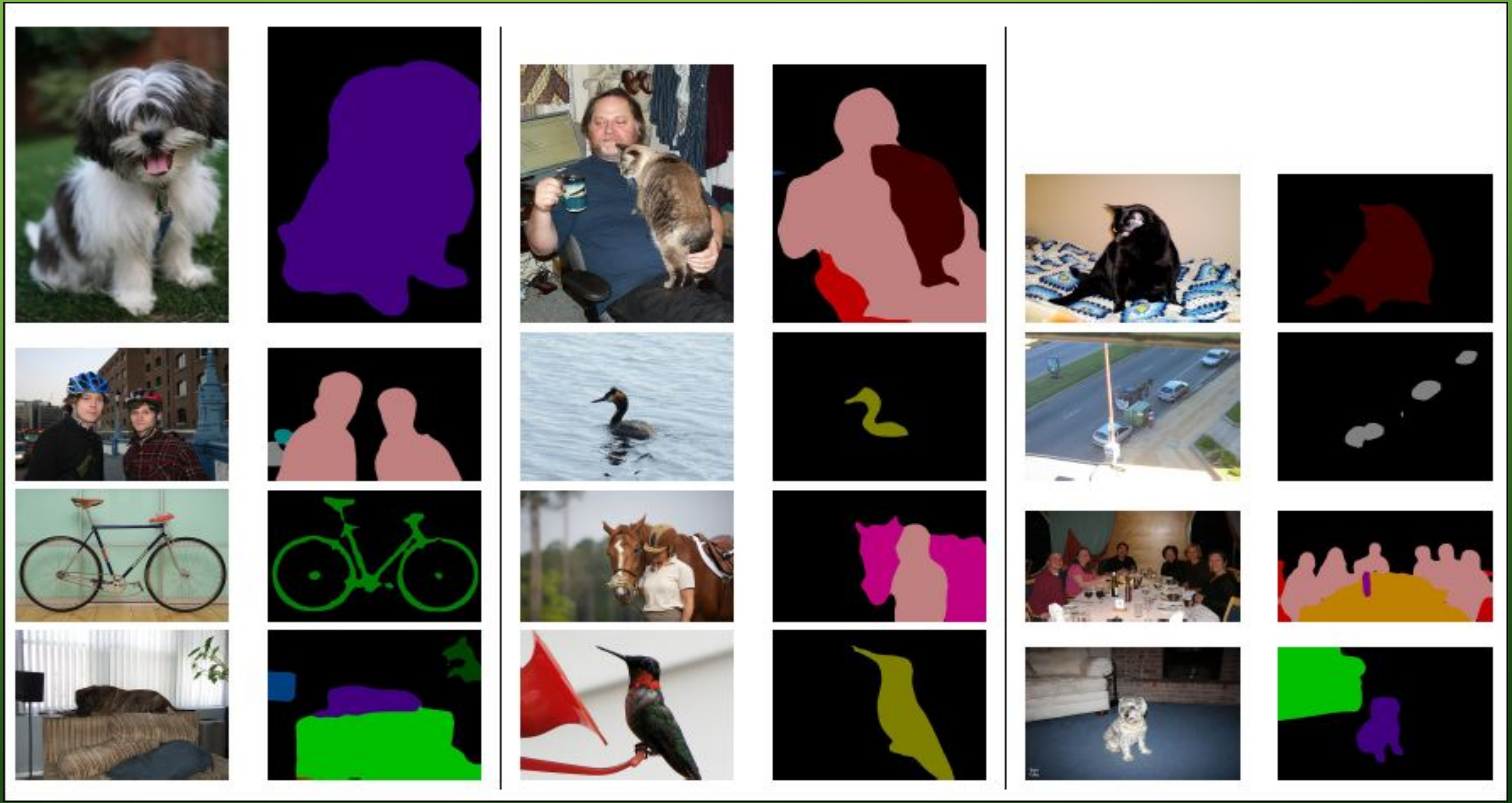
DEEP LEARNING FELLOWSHIP

- Objectives
  - Build community of machine learning enthusiasts
  - Connect and share ideas
  - Learn together
- Built by volunteers
- Work in progress

AGENDA

- Semantic segmentation
- U-Net model (with variations)
- Loss functions and metrics for semantic segmentation
- Semantic gaps
- UCTransNet model
- Code from the authors

WHAT IS SEMANTIC SEGMENTATION



WHY USE SEMANTIC SEGMENTATION

- “Segmentation and the subsequent quantitative assessment of target object in medical images provide valuable information for the analysis of pathologies and are important for planning of treatment strategies, monitoring of disease progression and prediction of patient outcome.”
- “Accurate and automated segmentation of medical images is a crucial step for clinical diagnosis and analysis.”
- Potentially faster, more accurate and more consistent.

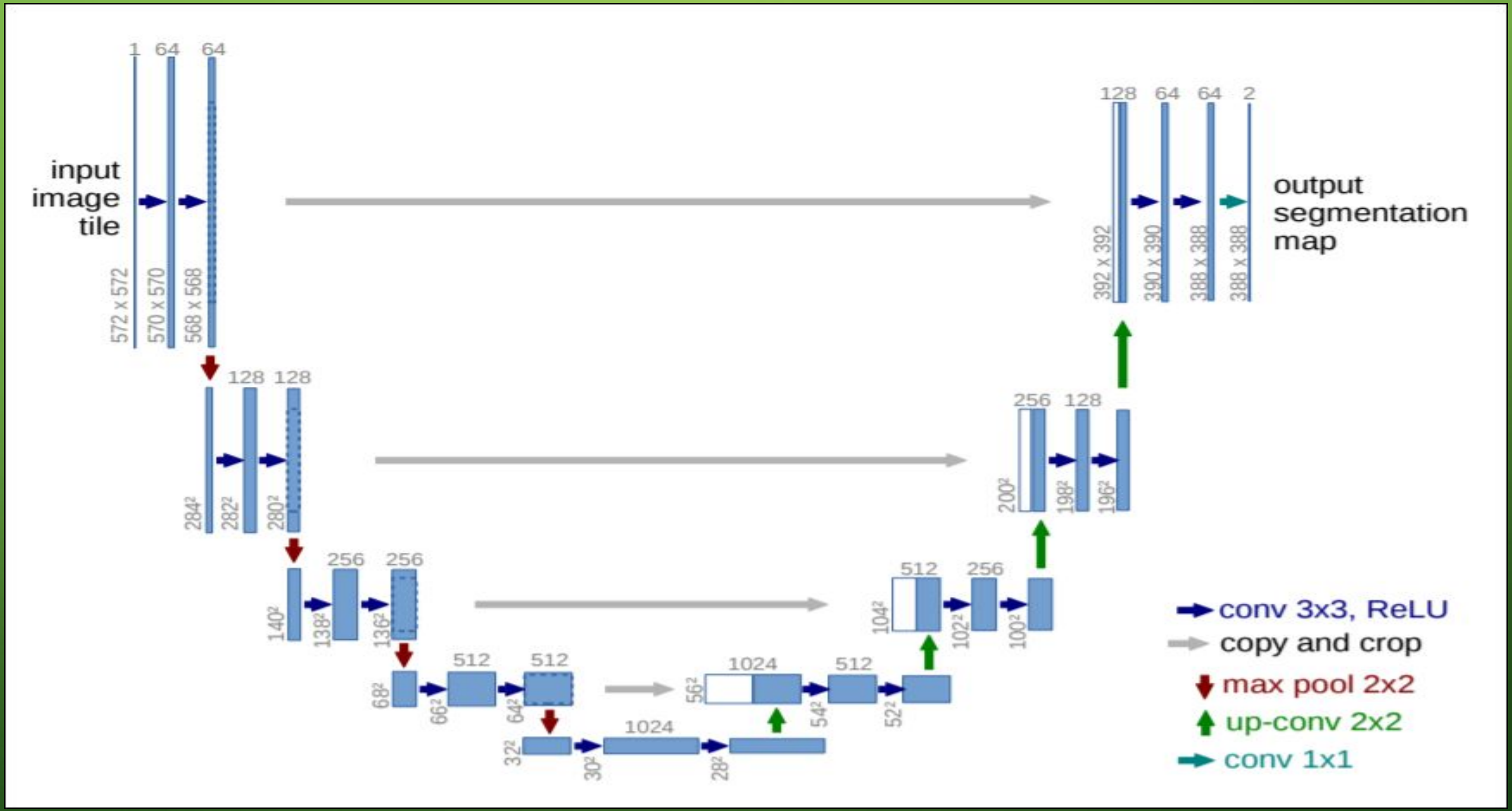
WHAT MAKES SEMANTIC SEGMENTATION DIFFICULT

- Pixel level classification problem
- Image classification algorithms consolidate information until the final output is a single label.
- Semantic segmentation must output a full-resolution map of labels.
- Must learn long distance contextual information while at the same time retaining high spatial resolution at the output for identifying small objects and sharp boundaries.
- More challenging to get large labeled datasets – transfer learning becomes more important

U-NET MODEL ORIGINS

- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. (2015) Arxiv 1505.04597
- Applied the u-net to a cell segmentation task in light microscope images.
- mIoU results:
  - PhC-U373 dataset (35 training images) – 0.9203 mIoU (next best 2015 – 0.83)
  - DIC-HeLa dataset (20 training images) – 0.7756 mIoU (next best 2015 – 0.46)

ORIGINAL U-NET MODEL



Ronneberger, et. al. U-net: Convolutional networks for biomedical image segmentation. Arxiv 1505.04597

ORIGINAL U-NET MODEL

- Combining high resolution features from the contracting path with upsampled output helps with localization.
- Large number of feature channels in upsampling path allows network to propagate context information to higher resolution layers.
- Heavy use of data augmentation / Weighted loss function
- Important to start with correct input size so all splits are even.

SEMANTIC SEGMENTATION LOSS FUNCTIONS & METRICS

- Error used in UCTransNet paper are:
  - Combined cross-entropy loss and dice loss
- Evaluation metrics used in UCTransNet paper:
  - Dice coefficient
  - Intersection over union (IoU)
  - Hausdorff Distance (HD)

SEMANTIC SEGMENTATION LOSS FUNCTIONS & METRICS

- Cross-entropy loss

Binary Cross-Entropy is defined as:

$$L_{BCE}(y, \hat{y}) = -(y log(\hat{y}) + (1 - y) log(1 - \hat{y}))$$

Here, $\hat{y}$ is the predicted value by the prediction model.

- Dice loss

$$DL(y, \hat{p}) = 1 - \frac{2y\hat{p} + 1}{y + \hat{p} + 1}$$

- Combined Cross-entropy and Dice loss attempts to leverage the flexibility of Dice loss for class imbalance at the same time use cross-entropy for curve smoothing

A Survey of loss functions for semantic segmentation.

By Shruti Jadon (arXiv:2006.14822v4)

SEMANTIC SEGMENTATION LOSS FUNCTIONS & METRICS

- Intersection over Union

$$IOU = \frac{Area\ of\ Intersection\ of\ two\ boxes}{Area\ of\ Union\ of\ two\ boxes}$$

- Hausdorff Distance

$$d(X,Y) = max_{x \epsilon X} min_{y \epsilon Y} ||x - y||_2$$

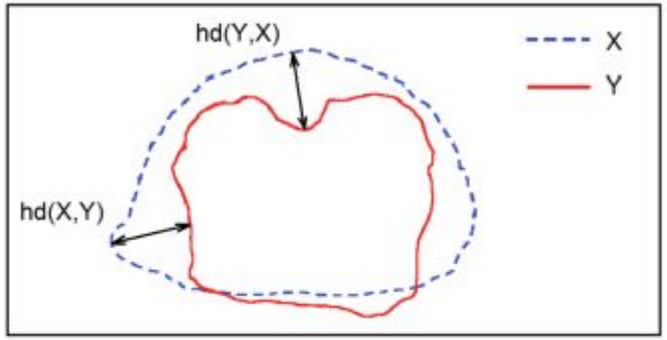


Fig. 3. Hausdorff Distance between point sets X and Y [18]

SEMANTIC SEGMENTATION LOSS FUNCTIONS

TABLE I
TYPES OF SEMANTIC SEGMENTATION LOSS FUNCTIONS [3]

Type	Loss Function
Distribution-based Loss	Binary Cross-Entropy Weighted Cross-Entropy Balanced Cross-Entropy Focal Loss Distance map derived loss penalty term
Region-based Loss	Dice Loss Sensitivity-Specificity Loss Tversky Loss Focal Tversky Loss **Log-Cosh Dice Loss**(ours)
Boundary-based Loss	Hausdorff Distance loss Shape aware loss
Compounded Loss	Combo Loss Exponential Logarithmic Loss

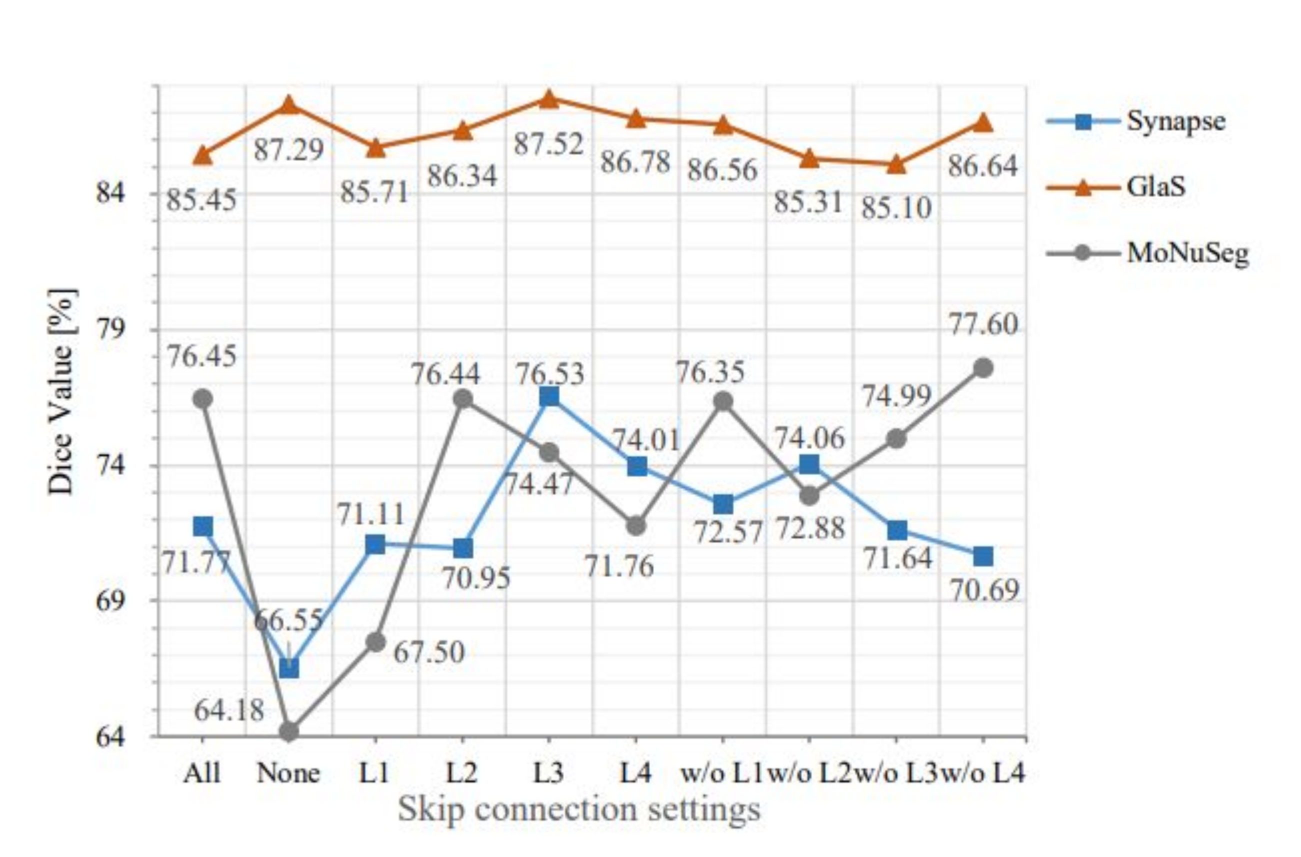
A Survey of loss functions for semantic segmentation.

By Shruti Jadon (arXiv:2006.14822v4)

TABLE II
TABULAR SUMMARY OF SEMANTIC SEGMENTATION LOSS FUNCTIONS

Loss Function	Use cases
Binary Cross-Entropy	Works best in equal data distribution among classes scenarios Bernoulli distribution based loss function
Weighted Cross-Entropy	Widely used with skewed dataset Weighs positive examples by β coefficient
Balanced Cross-Entropy	Similar to weighted-cross entropy, used widely with skewed dataset weighs both positive as well as negative examples by β and $1-\beta$ respectively
Focal Loss	works best with highly-imbalanced dataset down-weight the contribution of easy examples, enabling model to learn hard examples
Distance map derived loss penalty term	Variant of Cross-Entropy Used for hard-to-segment boundaries
Dice Loss	Inspired from Dice Coefficient, a metric to evaluate segmentation results. As Dice Coefficient is non-convex in nature, it has been modified to make it more tractable.
Sensitivity-Specificity Loss	Inspired from Sensitivity and Specificity metrics Used for cases where there is more focus on True Positives.
Tversky Loss	Variant of Dice Coefficient Add weight to False positives and False negatives.
Focal Tversky Loss	Variant of Tversky loss with focus on hard examples
Log-Cosh Dice Loss(ours)	Variant of Dice Loss and inspired regression log-cosh approach for smoothing Variations can be used for skewed dataset
Hausdorff Distance loss	Inspired by Hausdorff Distance metric used for evaluation of segmentation Loss tackle the non-convex nature of Distance metric by adding some variations
Shape aware loss	Variation of cross-entropy loss by adding a shape based coefficient used in cases of hard-to-segment boundaries.
Combo Loss	Combination of Dice Loss and Binary Cross-Entropy used for lightly class imbalanced by leveraging benefits of BCE and Dice Loss
Exponential Logarithmic Loss	Combined function of Dice Loss and Binary Cross-Entropy Focuses on less accurately predicted cases
Correlation Maximized Structural Similarity Loss	Focuses on Segmentation Structure. Used in cases of structural importance such as medical images.

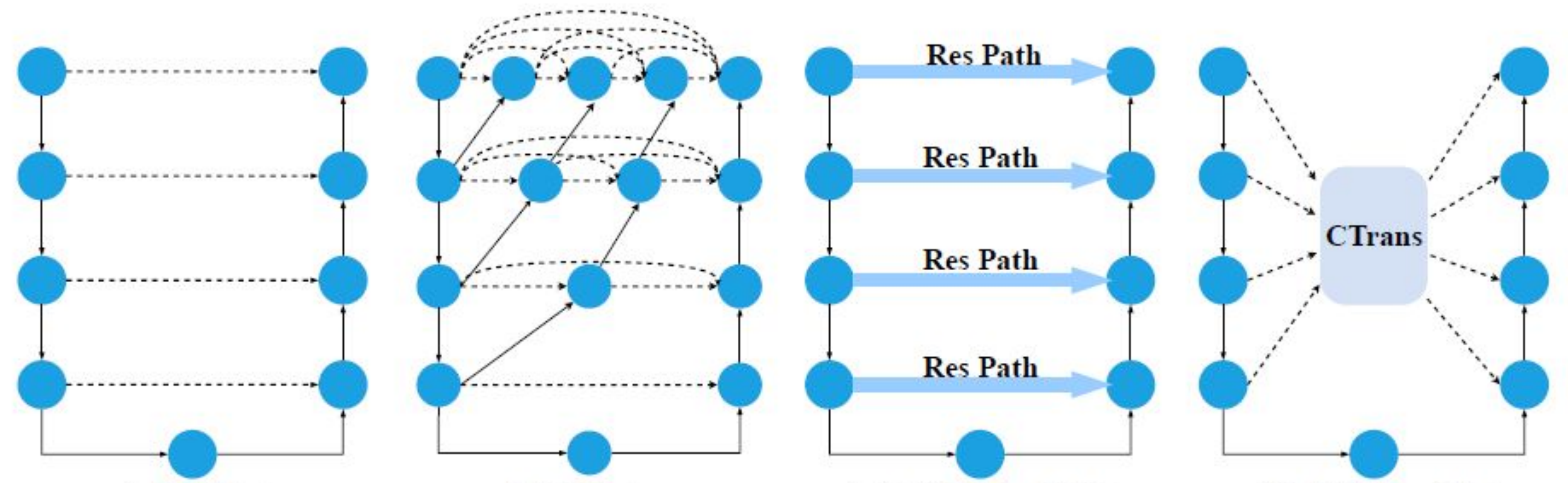
U-NET MODEL TESTING



- UNet model tested on various datasets
- Dice Values – higher is better
- Having no skip connect can make results better
- Including certain connections can make model worse
- Optimal combination is different for different datasets

U-NET STRUCTURES

- Some of the historical U-Net structures are as follows:



(a) U-Net (b) UNet++ (c) MultiResUNet (d) UCTransNet

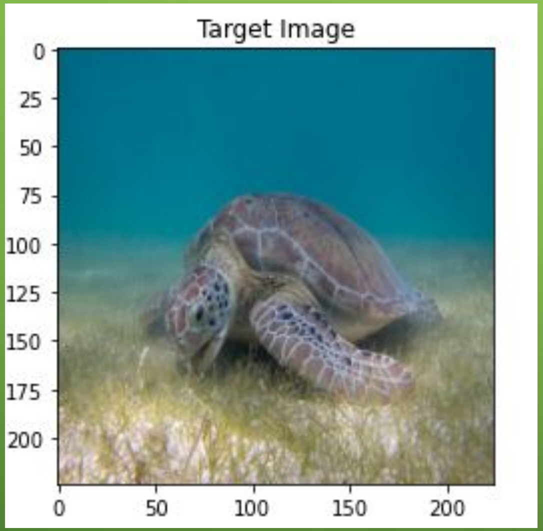
KEY ISSUES IDENTIFIED

- Two key Issues:
  1. Which layers of the encoder are connected to the decoder?
  2. How do you effectively fuse the features with possible semantic gaps instead of simply concatenating?
- Two semantic gaps
  - Semantic gap among multi-scale encoder features
  - Semantic gap between the stages of the encoder and decoder
- These semantic gaps limit the segmentation performance

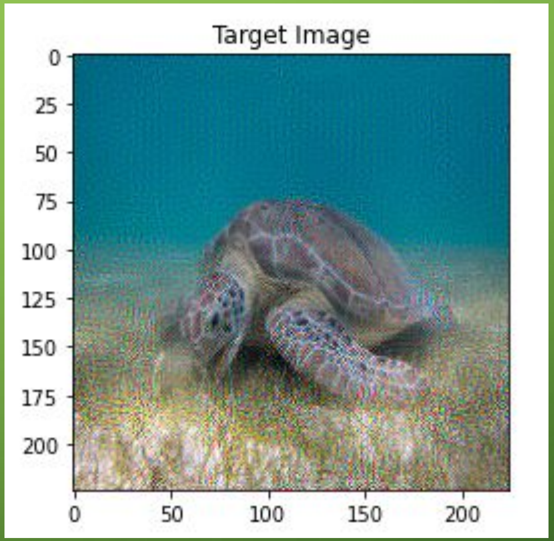
SEMANTIC GAP

Original Image

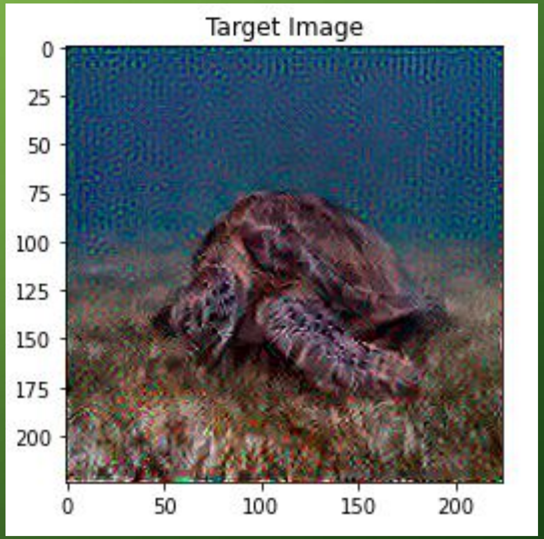
Block 1 Conv 2



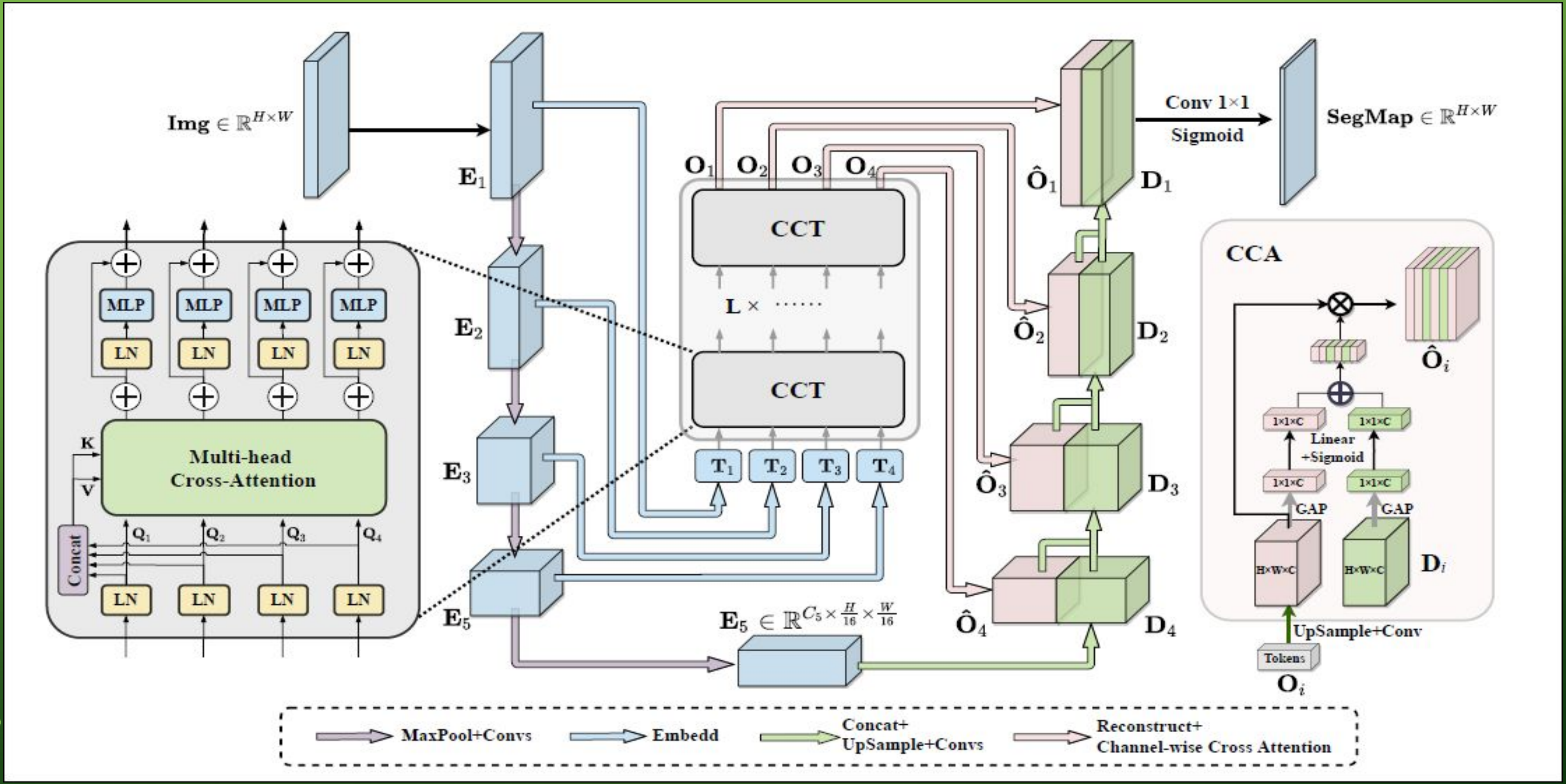
Block 3 Conv 2



Block 5 Conv 2



UCTRANSNET MODEL



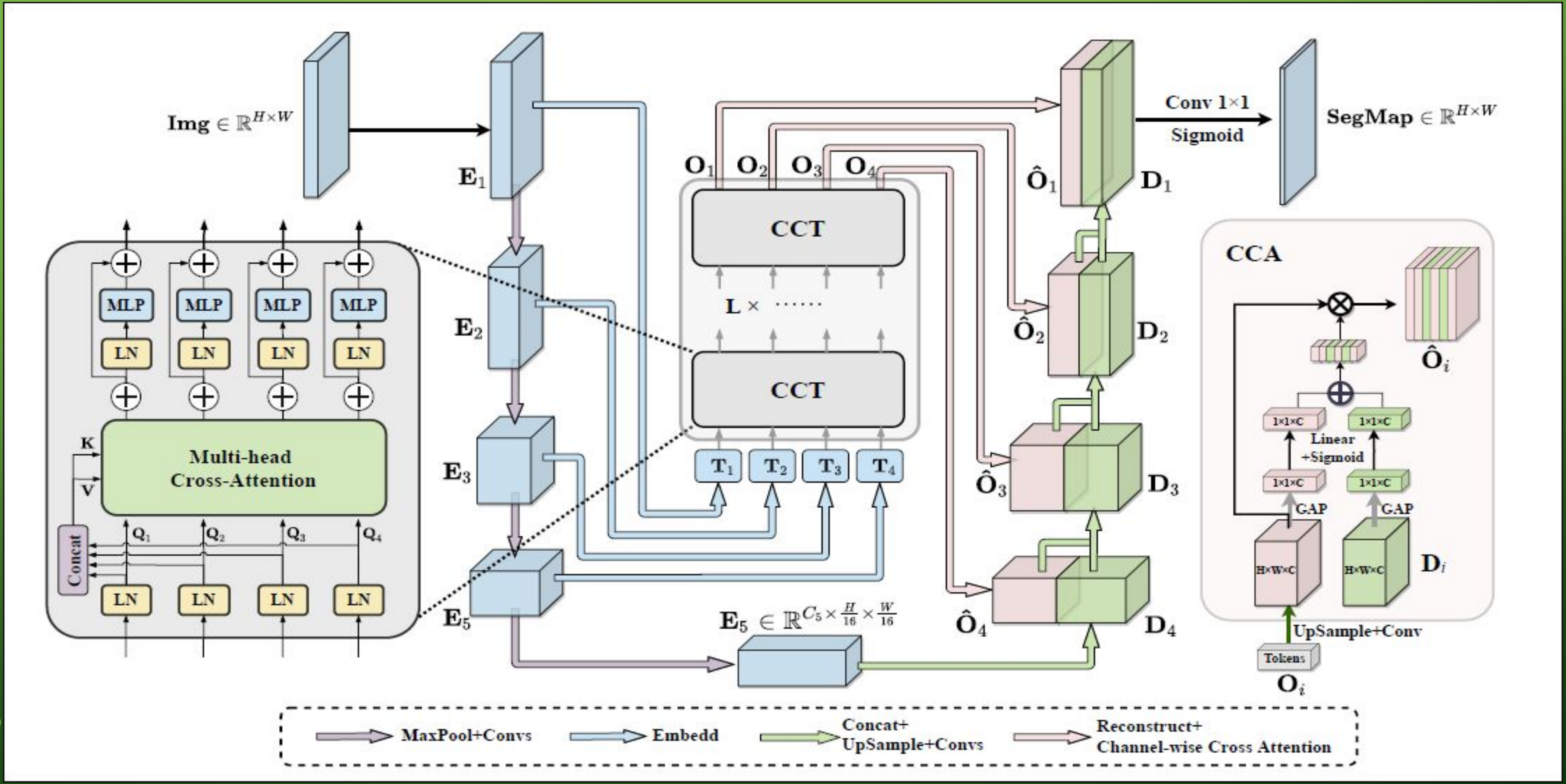
CHANNEL-WISE CROSS FUSION TRANSFORMER (CCT)

- Used for encoder feature transformation
- Consists of three steps:
  1. Multi-scale feature embedding
  2. Multi-head channel-wise cross attention
  3. Multi-layer perceptron (MLP)

MULTI-SCALE FEATURE EMBEDDING

- Convolution run over the encoder feature map at each skip connect level
- Each feature map (channel) processed separately
- Filter size and step size (patch size) and step size set to generate the same sized output for each skip connect (P, P/2, P/4, P/8)
- Flattened output of embedding (labeled T_i) becomes the output of the multi-scale feature embedding
- Serves as a summary of each feature map with spatial information preserved in sequence order

UCTRANSNET MODEL



MULTI-HEAD CROSS-ATTENTION

- Tokens from embedding layer are fed into the multi-head cross-attention module
- Flattened output of embedding at each level (labeled T_i) multiplied by learned weights W to get query (Q_i) for Transformer
- T_i for each level concatenated and used as input to get key (K) and value (V) (after applying learned weights to each
- Results in 4 Queries, 1 Key and 1 Value

$$\mathbf{Q}_i = \mathbf{T}_i W_{\mathbf{Q}_i}, \mathbf{K} = \mathbf{T}_\Sigma W_\mathbf{K}, \mathbf{V} = \mathbf{T}_\Sigma W_\mathbf{V}$$

SEMANTIC SEGMENTATION

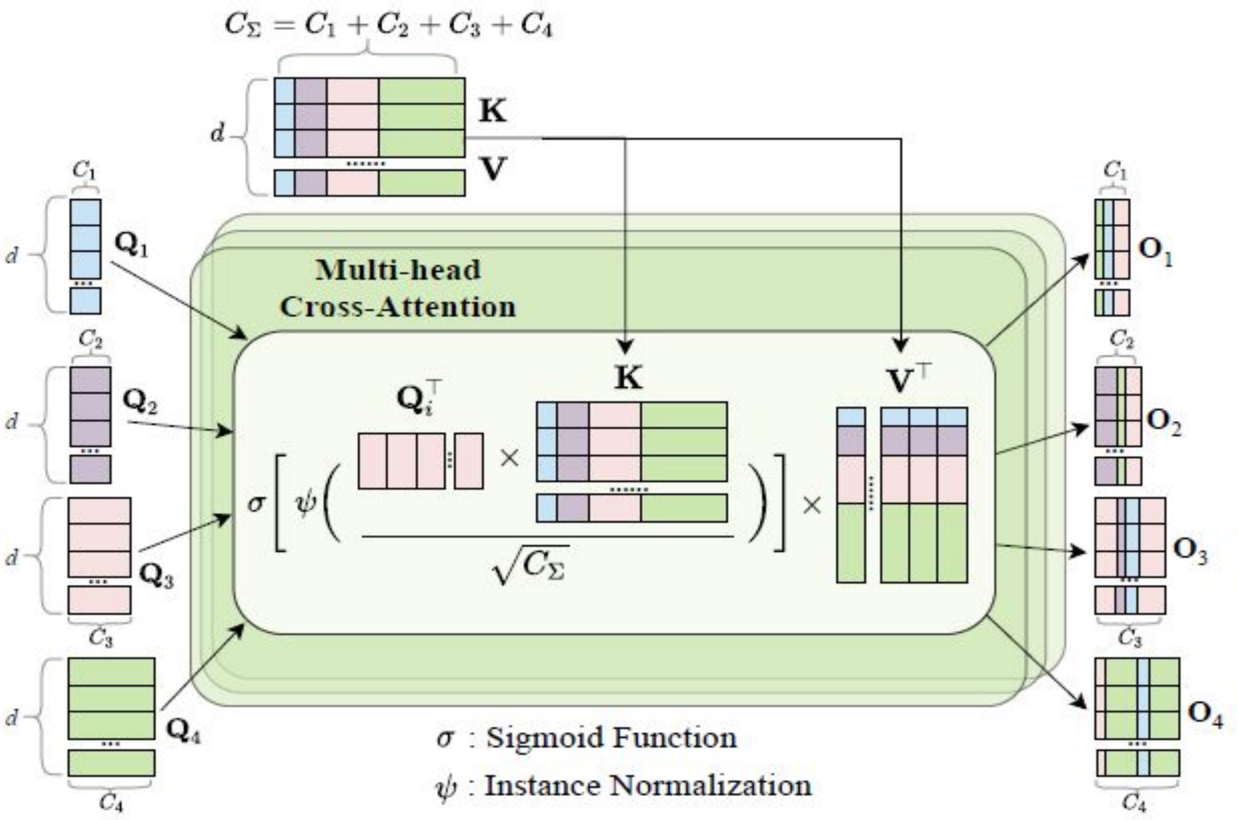


Figure 5: Multi-head Cross-Attention.

MULTI-HEAD CROSS-ATTENTION

- Similarity matrix (M_i) is produced as follows:
  - Get dot product of current Query and Key – provides a proximity measure between the channels in the current query and all channels from the encoder
  - Scale the dot product base on total number of channels
  - Perform instance normalization
  - Put the result through a function that will output a range bound by 0 and 1 (sigmoid or softmax)

$$= \sigma\left[\psi\left(\frac{\mathbf{Q}_i^\top \mathbf{K}}{\sqrt{C_\Sigma}}\right)\right]$$

MULTI-HEAD CROSS-ATTENTION

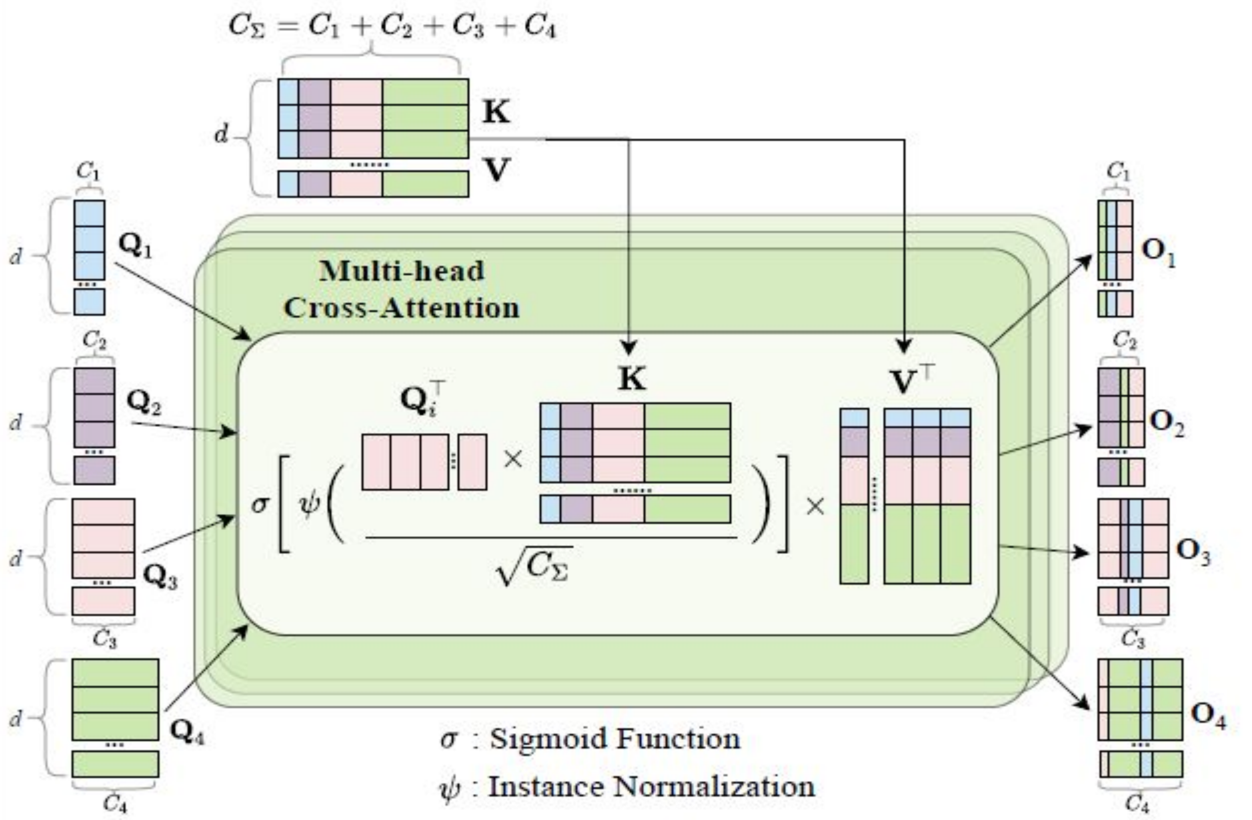


Figure 5: Multi-head Cross-Attention.

$$\mathrm{CA}_i = \mathbf{M}_i \mathbf{V}^\top = \sigma\left[\psi\left(\frac{\mathbf{Q}_i^\top \mathbf{K}}{\sqrt{C_\Sigma}}\right)\right]\mathbf{V}^\top$$
$$= \sigma\left[\psi\left(\frac{W_{\mathbf{Q}_i}^\top \mathbf{T}_i^\top \mathbf{T}_\Sigma W_\mathbf{K}}{\sqrt{C_\Sigma}}\right)\right] W_\mathbf{V}^\top \mathbf{T}_\Sigma^\top$$

MULTI-HEAD CROSS-ATTENTION

- The cross-attention mechanism is determined by multiplying the similarity matrix by the value matrix
- W_Q and W_K should learn to enhance those filters and locations that are important to the segmentation
- The similarity matrix highlights those channels (filters) in all cross connected layers that are similar to the current Q
- The softmax (sigmoid) output allows the model to downscale the unimportant channels and locations when the cross-attention mechanism is multiplied by the Value matrix

UCTRANSNET CHANNEL ATTENTION

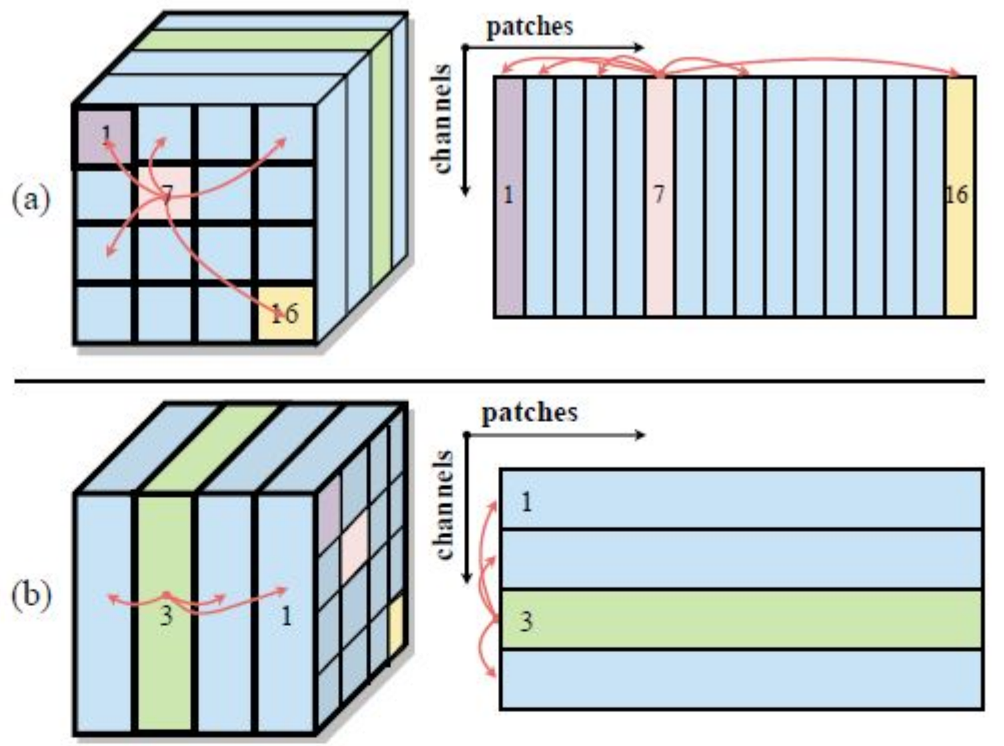
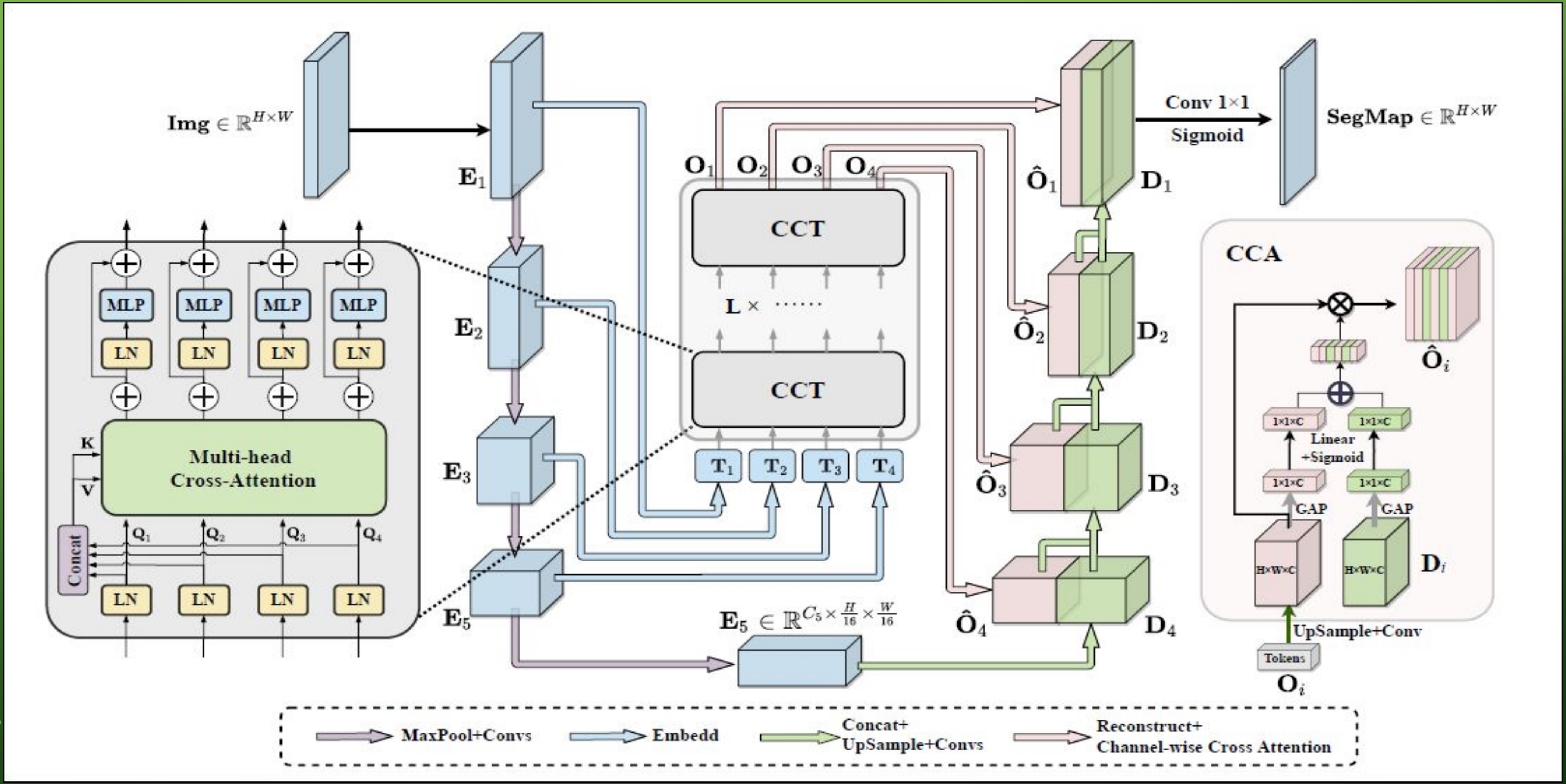


Figure 4: Comparison between the original self-attention (a) and our proposed channel-wise cross-attention (b).

UCTRANSNET MODEL



MULTI-LAYER PERCEPTRON (MLP)

- In an N-head attention situation, the output after the multi-head cross-attention is calculated as an average of the head outputs

$$\boxed{\mathrm{MCA}_i = (\mathrm{CA}_i^1 + \mathrm{CA}_i^2 +, \ldots, +\mathrm{CA}_i^N)/N}$$

- The MCA for each level is then put through a layer normalization and a fully connected layer (with a residual connection)

$$\boxed{\mathbf{O}_i = \mathrm{MCA}_i + \mathrm{MLP}(\mathbf{Q}_i + \mathrm{MCA}_i)}$$

CHANNEL-WISE CROSS ATTENTION (CCA)

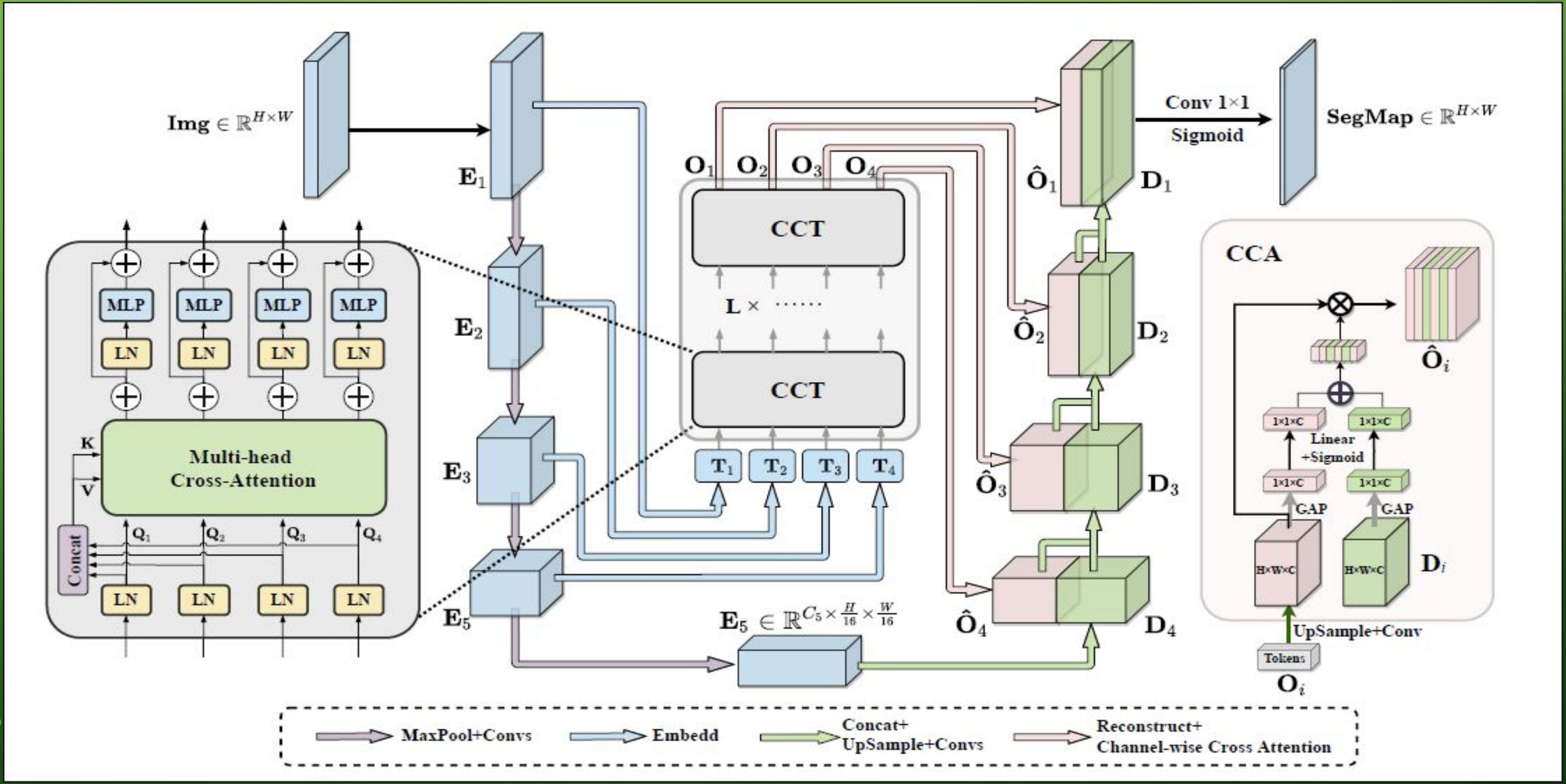
- The CCA is used to better fuse the inconsistent semantics between the Channel Transformer and U-Net decoder
- This will guide the channel and information filtration of the Transformer features and eliminate the ambiguity with the decoder features

$$\mathcal{G}(\mathbf{X}) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{X}^k(i,j)$$

$$\mathbf{M}_i = \mathbf{L}_1 \cdot \mathcal{G}(\mathbf{O_i}) + \mathbf{L}_2 \cdot \mathcal{G}(\mathbf{D_i})$$

$$\hat{\mathbf{O}}_i = \sigma(\mathbf{M}_i) \cdot \mathbf{O_i}$$

UCTRANSNET MODEL



UCTRANSNET MODEL – TEST RESULTS

Method	Param (M)	FlOPs (G)	GlaS Dice (%)	GlaS IoU (%)	MoNuSeg Dice (%)	MoNuSeg IoU (%)
U-Net	14.8	50.3	85.45±1.25	74.78±1.67	76.45±2.62	62.86±3.00
UNet++	74.5	94.6	87.56±1.17	79.13±1.70	77.01±2.10	63.04±2.54
AttUNet	34.9	101.9	88.80±1.07	80.69±1.66	76.67±1.06	63.47±1.16
MRUNet	57.2	78.4	88.73±1.17	80.89±1.67	78.22±2.47	64.83±2.87
TransUNet	105	56.7	88.40±0.74	80.40±1.04	78.53±1.06	65.05±1.28
MedT	98.3	131.5	85.92±2.93	75.47±3.46	77.46±2.38	63.37±3.11
Swin-Unet	82.3	67.3	89.58±0.57	82.06±0.73	77.69±0.94	63.77±1.15
Ours	65.6	63.2	**90.18±0.71***	**82.96±1.06***	**79.08±0.67**	**65.50±0.91**

QUESTIONS?

https://www.meetup.com/fr-FR/meetup-group-optfgvkc/

https://the-deep-learning-fellowship.github.io/