

## U-Net Models

3 models for the segmentation task:

- **Classic U-Net:** *baseline U-Net model architecture*

## U-Net Models

3 models for the segmentation task:

- **Classic U-Net**: *baseline U-Net model architecture*
- **Improved U-Net**: *small improvements, fewer parameters*

## U-Net Models

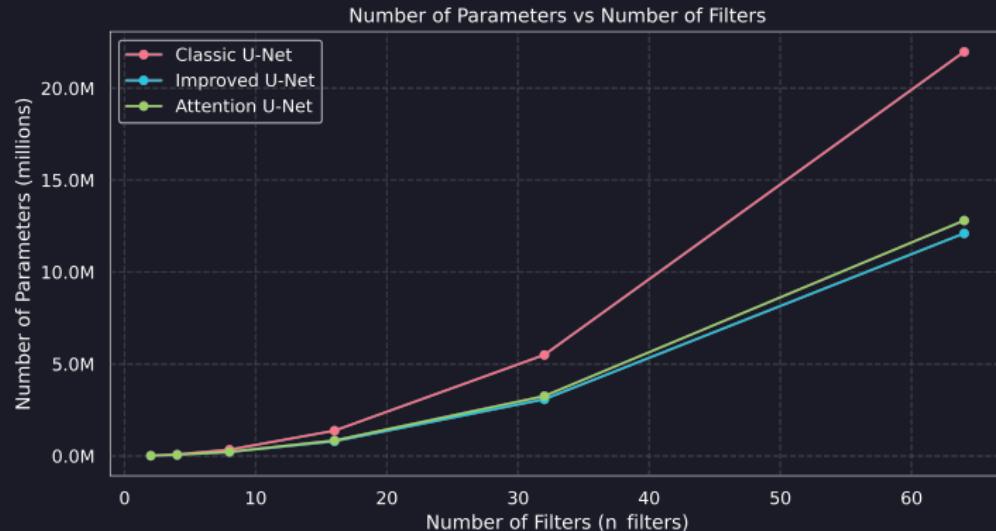
3 models for the segmentation task:

- **Classic U-Net**: *baseline U-Net model architecture*
- **Improved U-Net**: *small improvements, fewer parameters*
- **Attention U-Net**: *attention mechanism added*

### U-Net Models

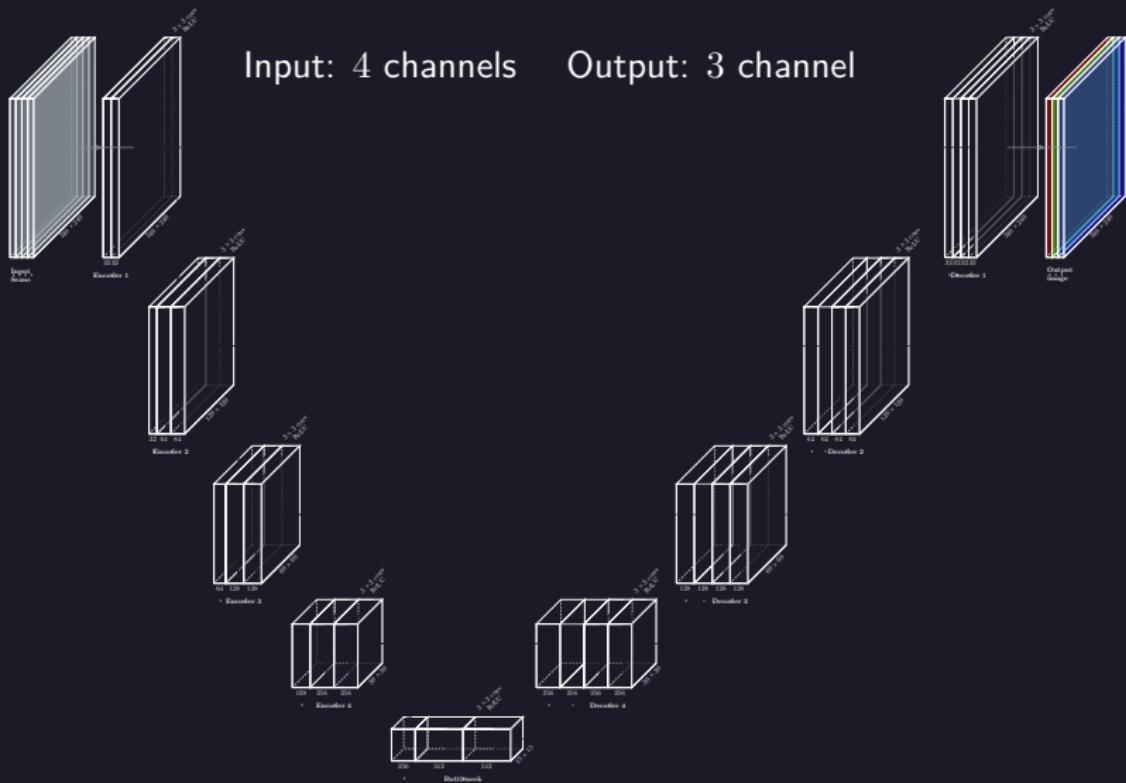
3 models for the segmentation task:

- **Classic U-Net**: *baseline U-Net model architecture*
- **Improved U-Net**: *small improvements, fewer parameters*
- **Attention U-Net**: *attention mechanism added*



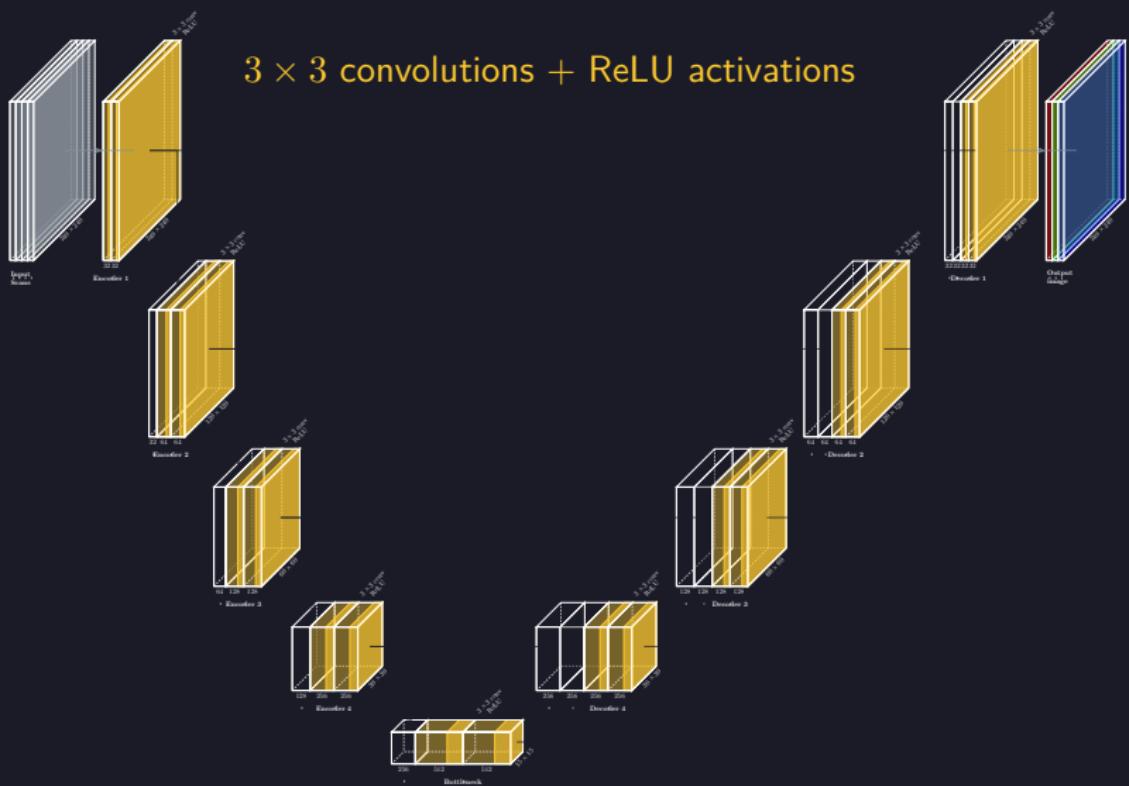
# Segmentation Models

## Classic U-Net



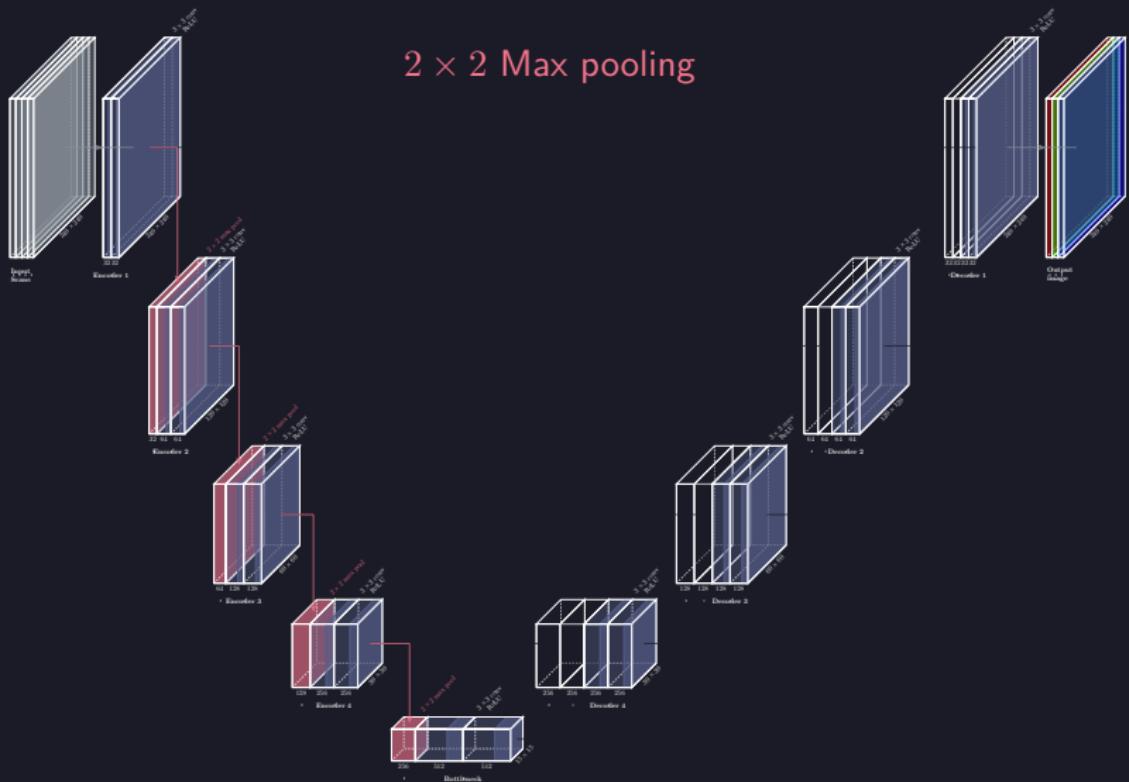
## Segmentation Models

# Classic U-Net



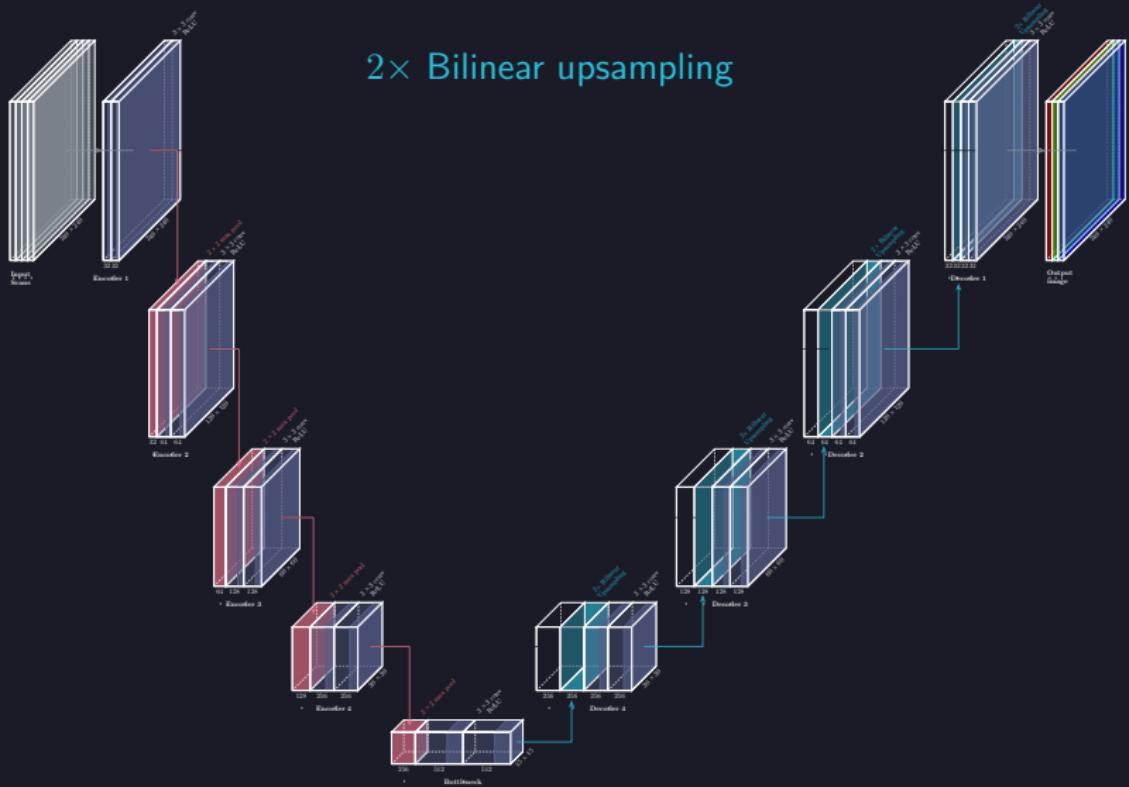
# Segmentation Models

## Classic U-Net



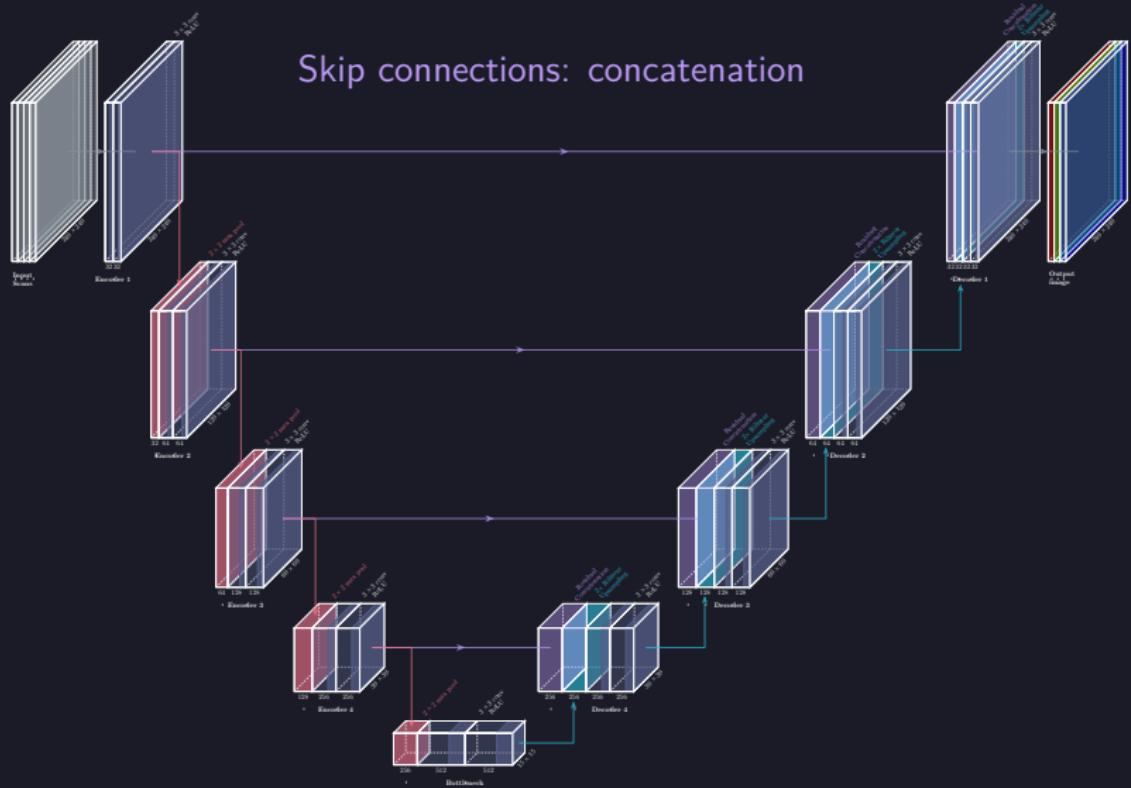
# Segmentation Models

## Classic U-Net



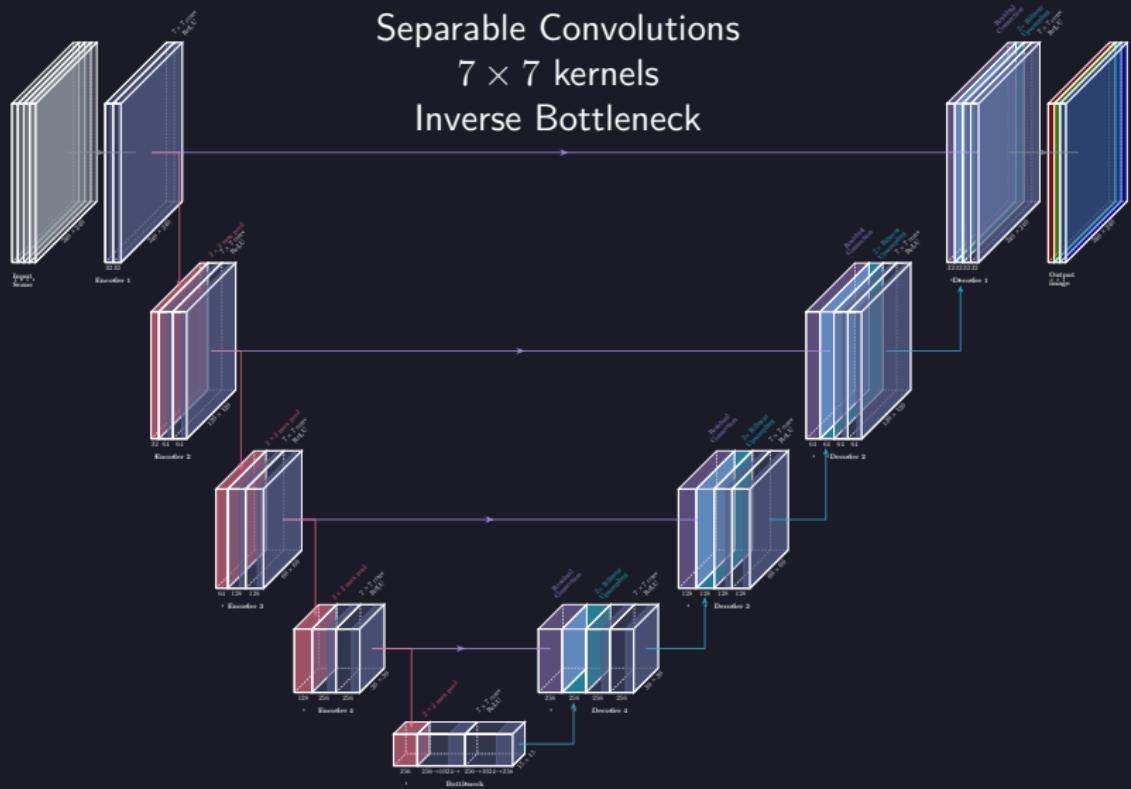
# Segmentation Models

## Classic U-Net



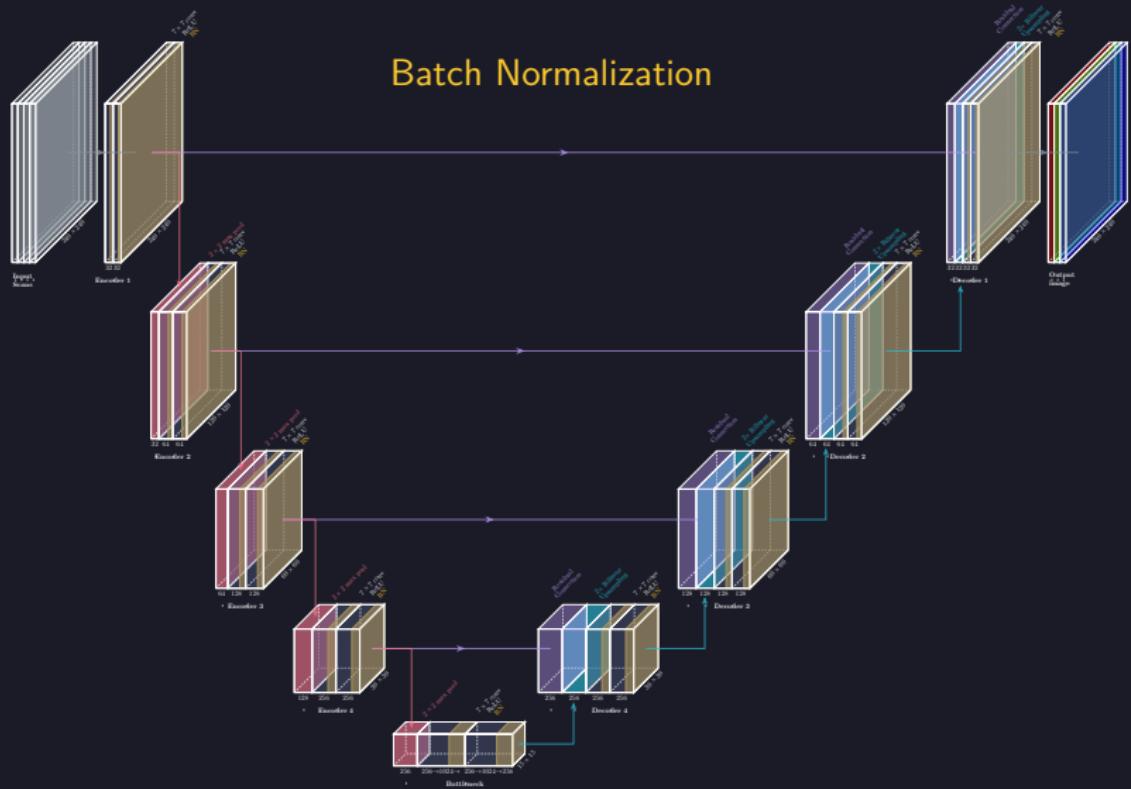
# Segmentation Models

## Improved U-Net



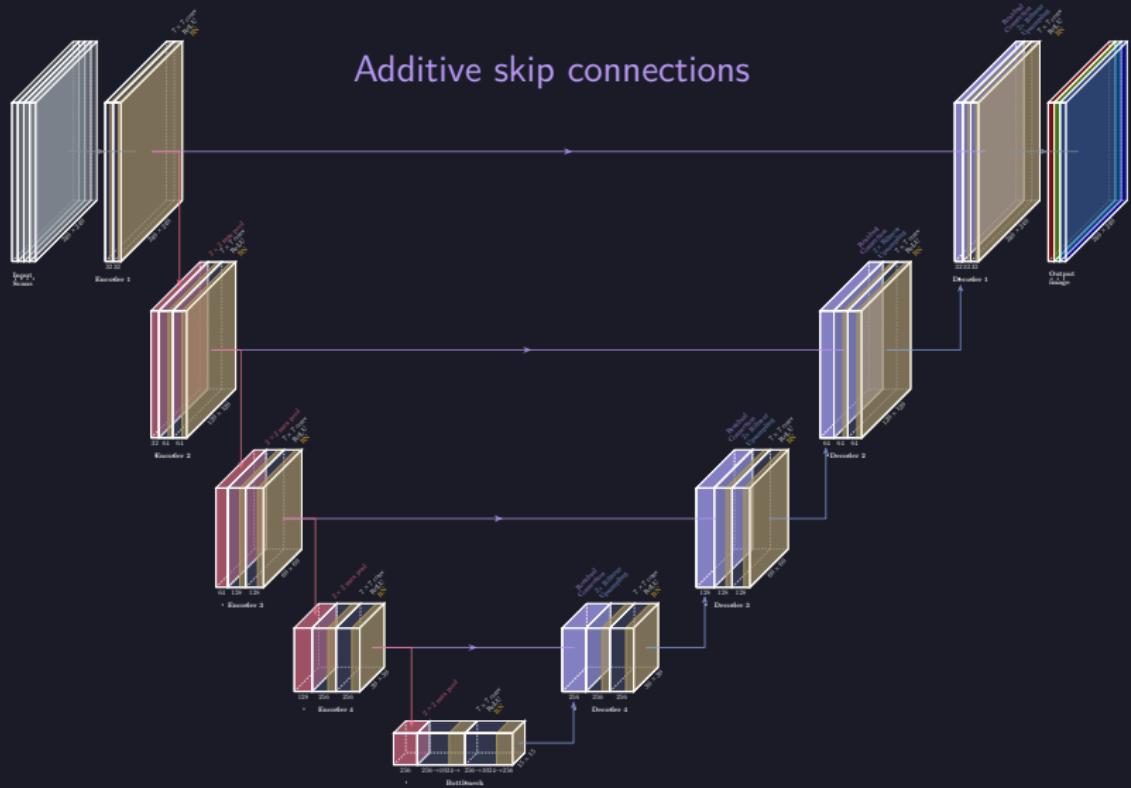
# Segmentation Models

## Improved U-Net



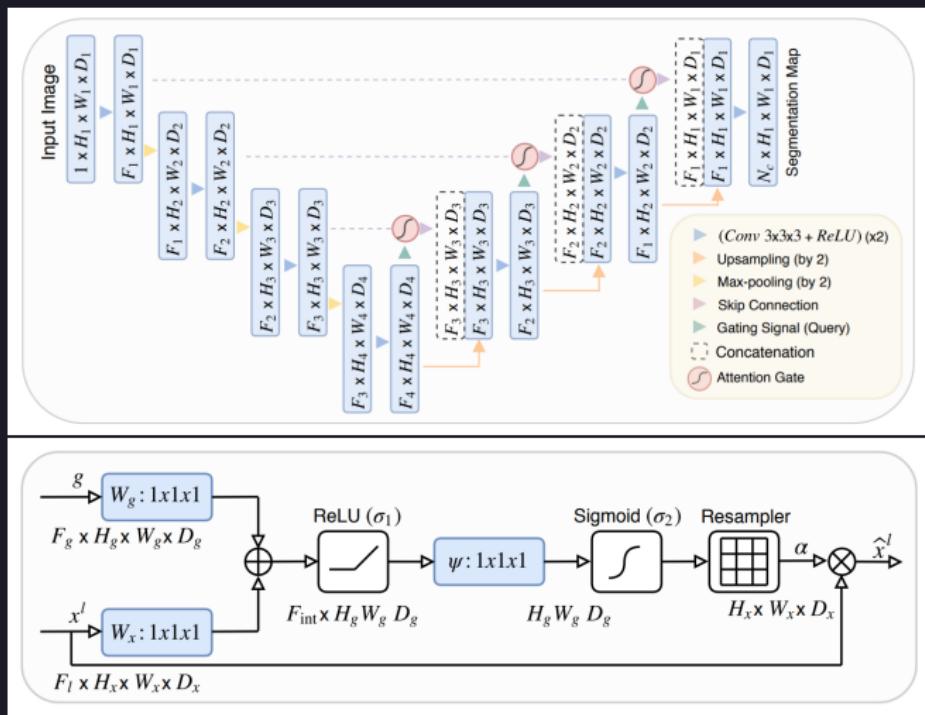
# Segmentation Models

## Improved U-Net



## Segmentation Models

## Attention U-Net



## Training Details

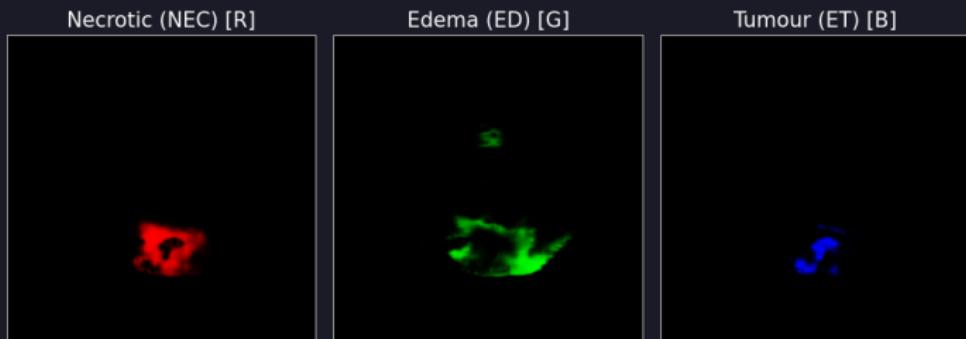
U-Net Models training parameters:

- **epochs:** 20
- **Optimizer:** Adam (with weight decay ( $1 \times 10^{-2}$ ))
- **Scheduler:** Exponential Decay ( $\gamma = 0.9$ )
- **Loss Function:** BCE with Logits Loss
- **learning rate:**  $2 \times 10^{-3}$
- **batch size:** 32 (both training and validation)
- **image size:**  $240 \times 240$
- **first encoder filters:** 32

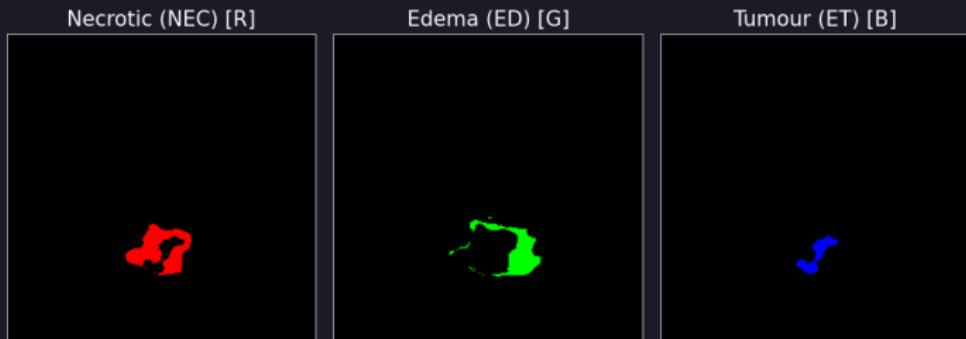
## Segmentation Models

# Visualizing a prediction

Predicted Mask Channels [RGB]



Ground Truth Mask Channels [RGB]



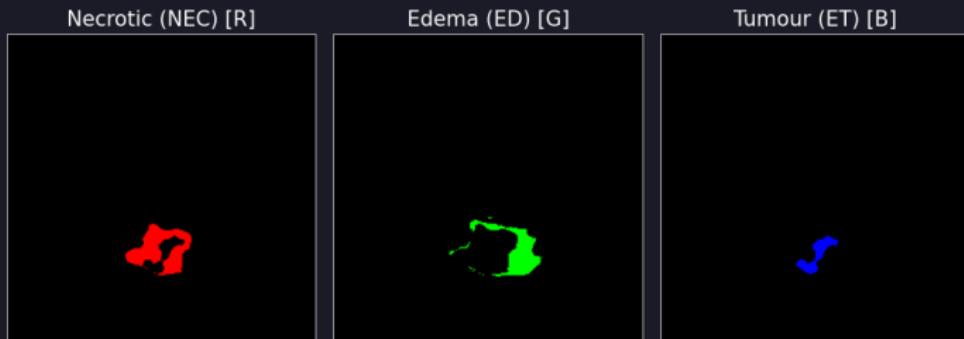
## Segmentation Models

# Visualizing a prediction

Binarized Predicted Mask Channels [RGB]



Ground Truth Mask Channels [RGB]



# Segmentation Models

## Performance Assessment

$$\text{Dice} = \frac{2 \times |X \cap Y|}{|X| + |Y|}$$

Dice Coefficient

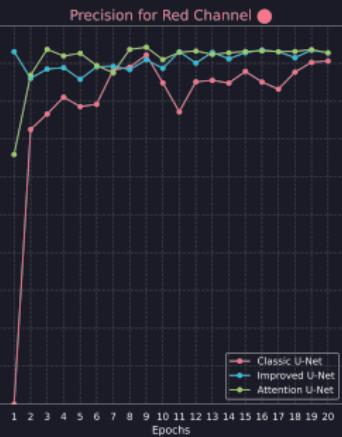
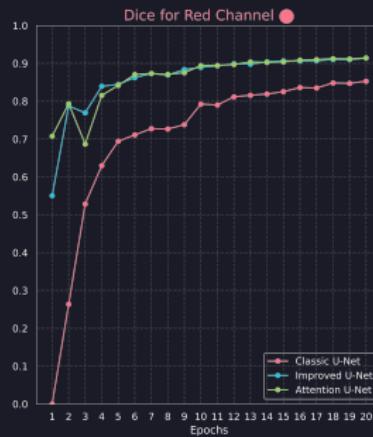
"overlap" metric

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision  
prediction quality

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall  
prediction quantity



# Segmentation Models

## Performance Assessment

$$\text{Dice} = \frac{2 \times |X \cap Y|}{|X| + |Y|}$$

Dice Coefficient

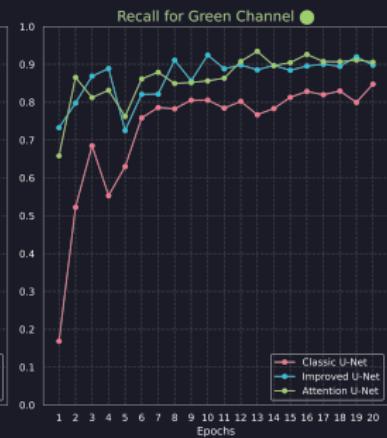
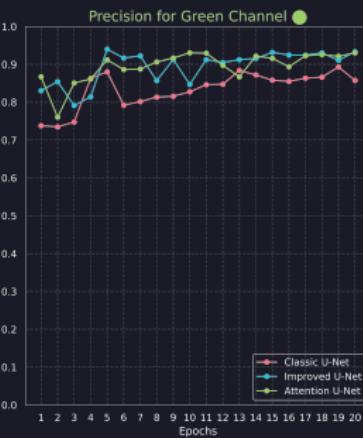
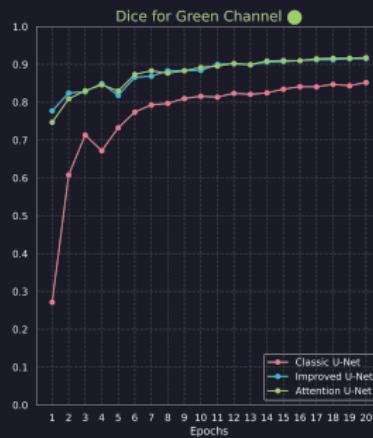
"overlap" metric

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision  
prediction quality

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall  
prediction quantity



# Segmentation Models

## Performance Assessment

$$\text{Dice} = \frac{2 \times |X \cap Y|}{|X| + |Y|}$$

Dice Coefficient

"overlap" metric

$$\text{Precision} = \frac{TP}{TP + FP}$$

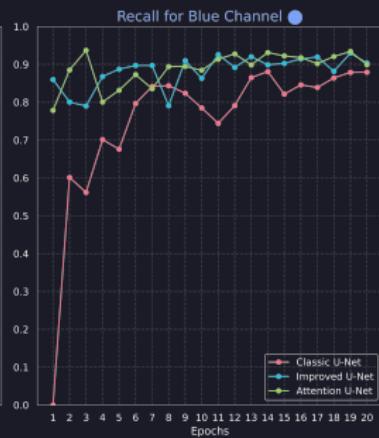
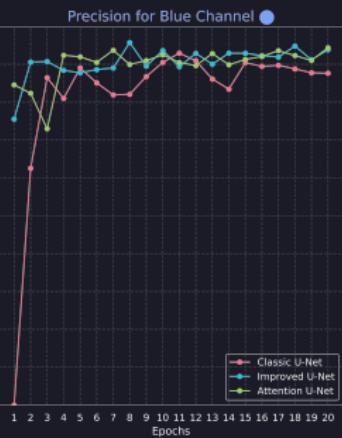
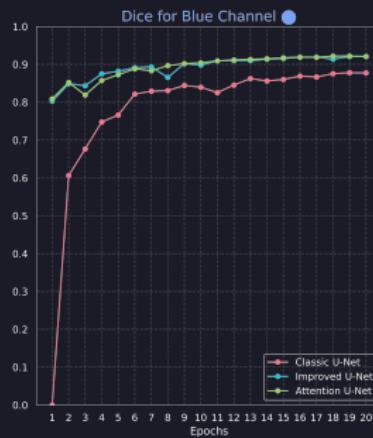
Precision

prediction quality

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall

prediction quantity



# Segmentation Models

## Performance Assessment

$$\text{Dice} = \frac{2 \times |X \cap Y|}{|X| + |Y|}$$

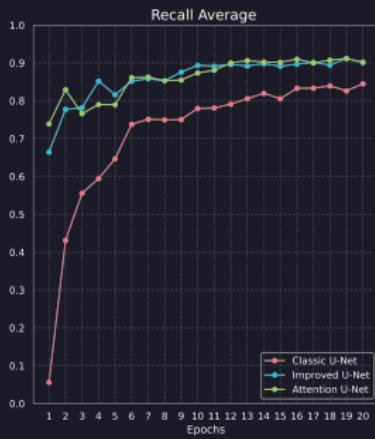
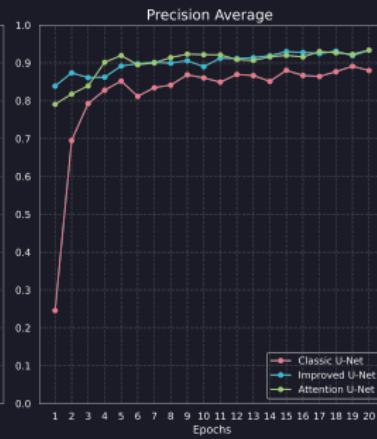
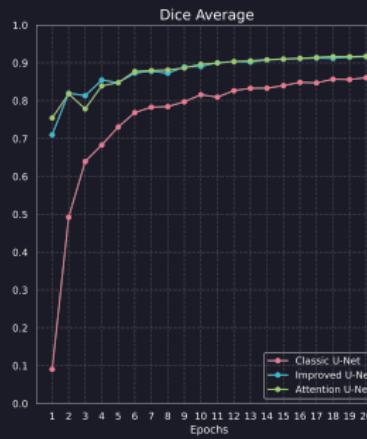
Dice Coefficient  
“overlap” metric

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision  
prediction quality

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall  
prediction quantity



## Segmentation Models

# Visualizing Attention Maps

