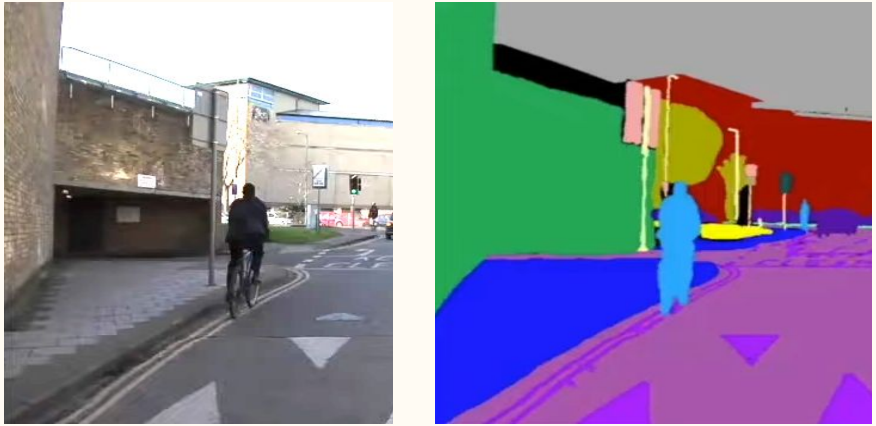
When Love is Bound by Skip Connections: The U-Net Romance

Adopted from: Ronneberger et al., 2015, U-Net

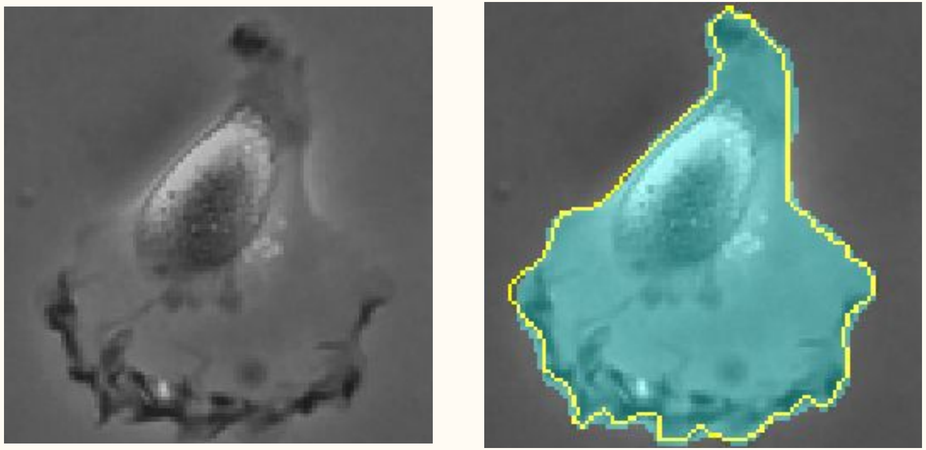
Vishagar Arunan

EN 4584 Advances in computer vision

The tale of Blurry Boundaries.



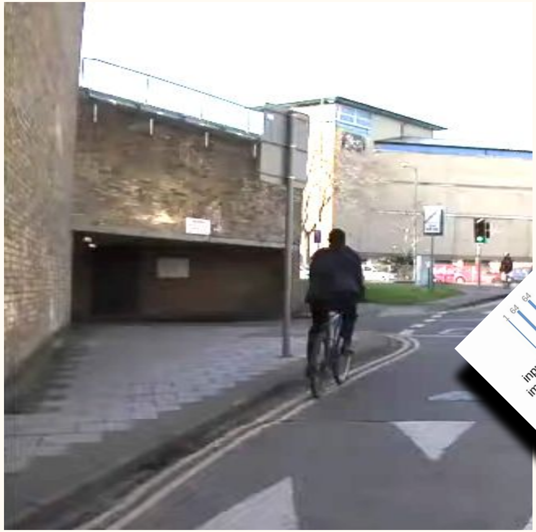
Semantic Segmentation
(Real world images)

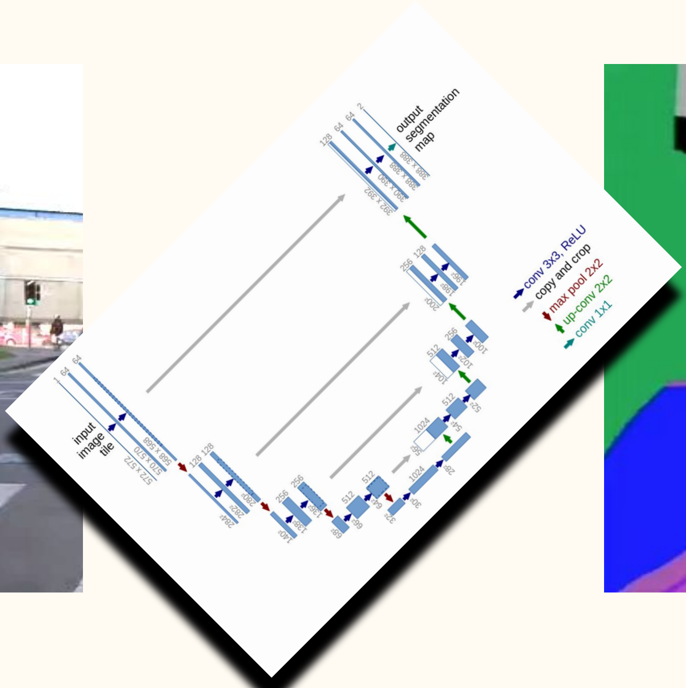


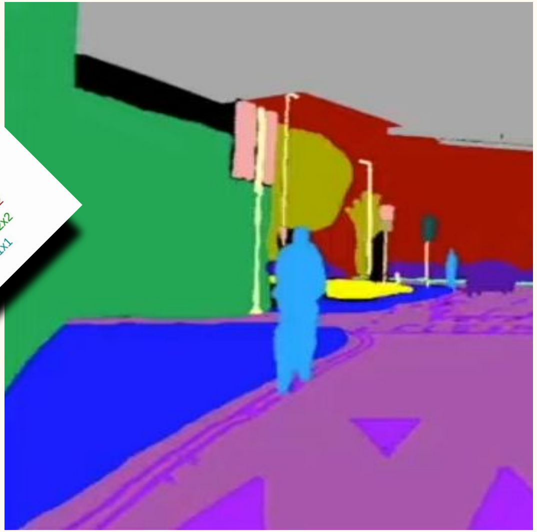
Semantic Segmentation
(Medical Images)

Source: Google Images, Ronneberger et al., 2015

The tale beyond the expectations







Semantic Segmentation

(Real world images)

Source: Google Images

The tale beyond the expectations

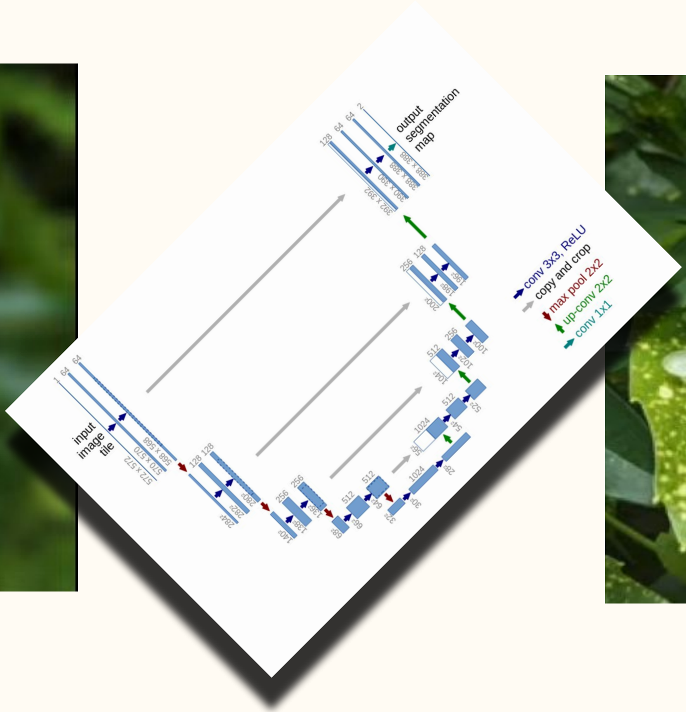
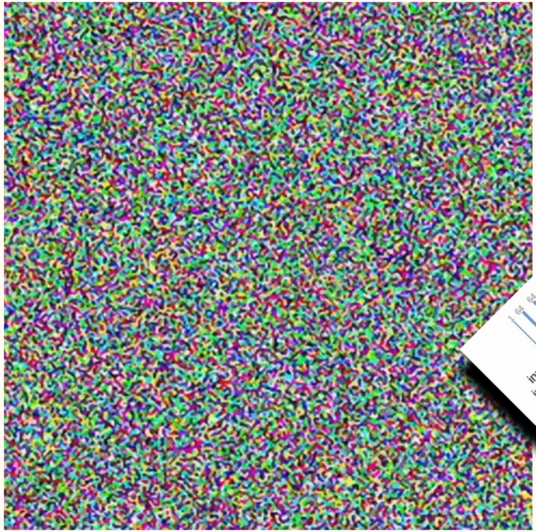


Image Super Resolution

Source: Google Images

The tale beyond the expectations



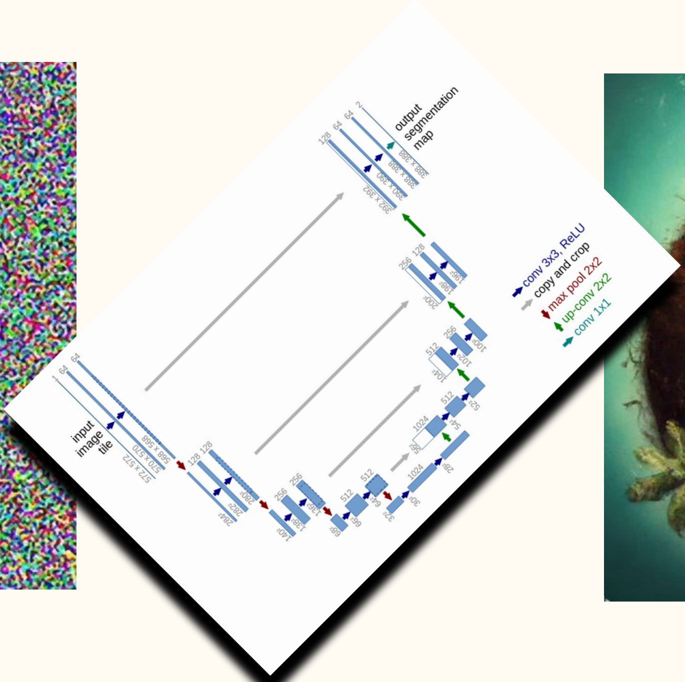


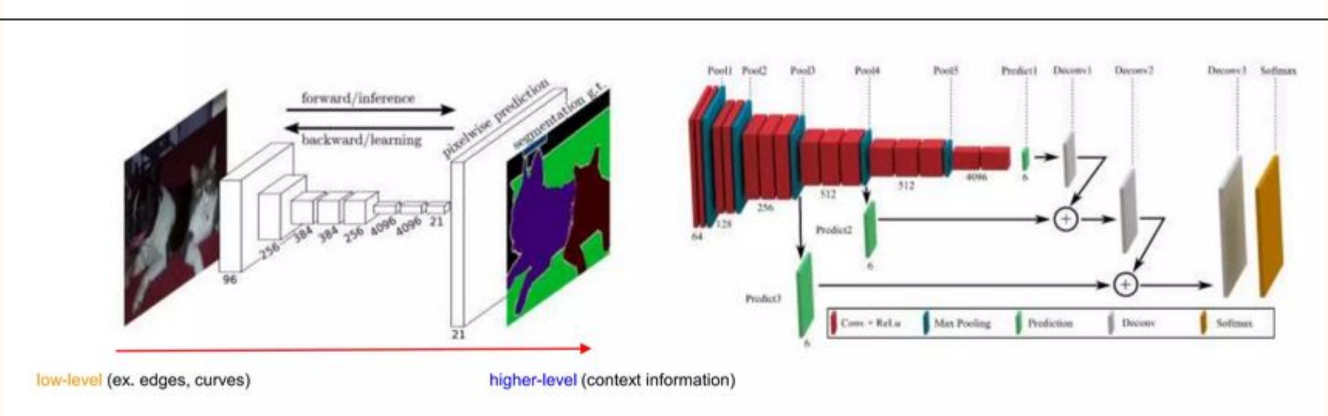
Image Generation

(DALL-E)

Source: Google Images

Where it began (FCN / CNN)

- Generally create high level feature maps as the network goes deeper.
- Can localize, yet slow.
- Trade-off between Localization Accuracy and the use of context information.
- Model variations: Utilize skip connections by summing.



The Unet Story

Mr. Encoder

Skip Connections

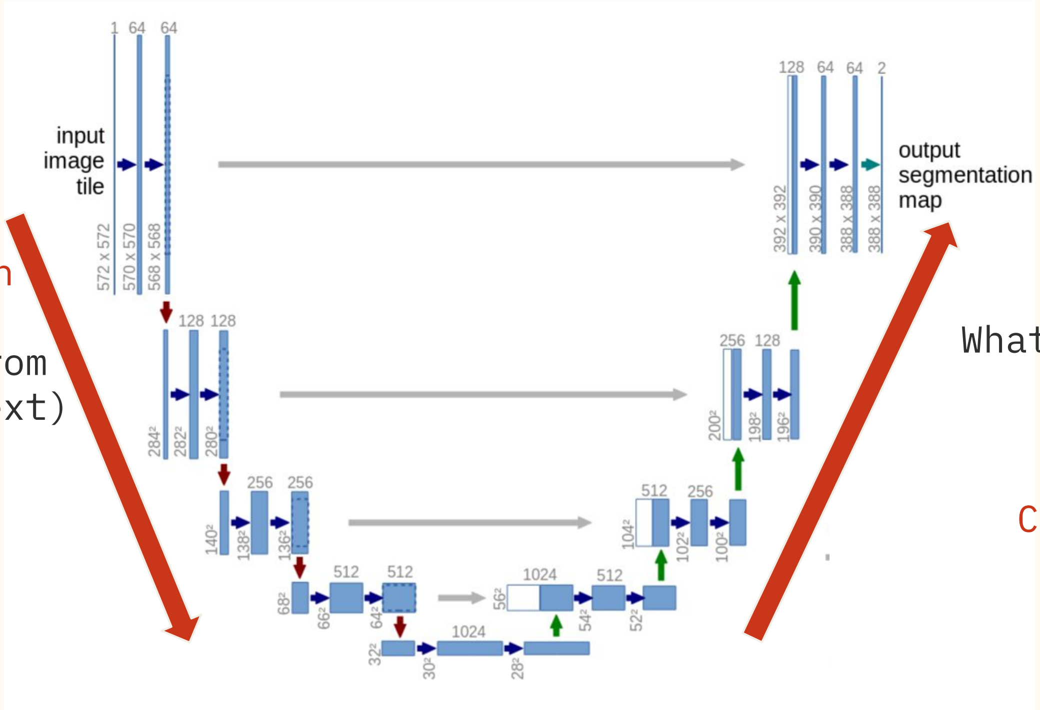
Miss. Decoder

The U-Net Story

Contracting Path

What is learnt from the image ? (context)

CNNs, Pooling



Expanding Path

What is in the image? (localization)

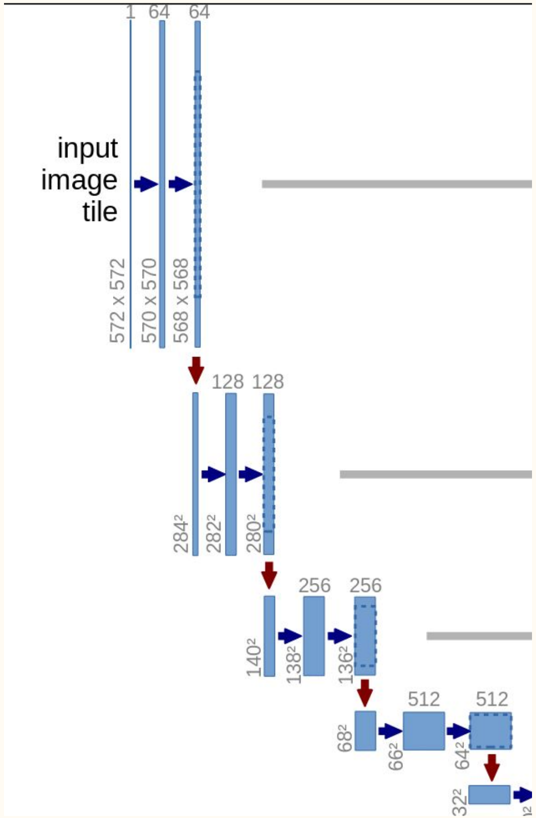
Transpose Convolution,CNNs

Source: Ronneberger et al., 2015

Mr. Encoder

- Learns the contextual information i.e. What is in the image?
- Gradually generate hierarchical features at different resolutions
- Consist,
  - 3x3 Double Valid Convolution (No padding)
  - Followed by Relu,
  - And a 2x2 Max Pool for downsampling.

Mr. Encoder




```python
class DoubleConv(nn.Module):
    def __init__(self, in_channels, out_channels):
        super().__init__()
        self.double_conv = nn.Sequential(
            nn.Conv2d(in_channels, out_channels, kernel_size=3, padding=1, bias=False),
            nn.BatchNorm2d(out_channels),
            nn.ReLU(inplace=True),
            nn.Conv2d(out_channels, out_channels, kernel_size=3, padding=1, bias=False),
            nn.BatchNorm2d(out_channels),
            nn.ReLU(inplace=True)
        )

    def forward(self, x):
        return self.double_conv(x)

class DownSample(nn.Module):
    def __init__(self, in_channels, out_channels):
        super().__init__()
        self.conv = DoubleConv(in_channels, out_channels)
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2)

    def forward(self, x):
        down = self.conv(x)
        pool = self.pool(down)

        return down, pool
```

```python
class UNet(nn.Module):
    def __init__(self, in_channels, num_classes):
        super().__init__()
        self.down_convolution_1 = DownSample(in_channels, 64)
        self.down_convolution_2 = DownSample(64, 128)
        self.down_convolution_3 = DownSample(128, 256)
        self.down_convolution_4 = DownSample(256, 512)
```

Miss. Decoder

- Learns the Localized information i.e. Where is in the image?
- Large number of feature channels allow network to propagate context information to higher resolution layers.
- Consist,
    - 2x2 Transpose Convolution,
    - And a 3x3 Valid Double Convolution,
    - Followed by Relu.

Miss. Decoder

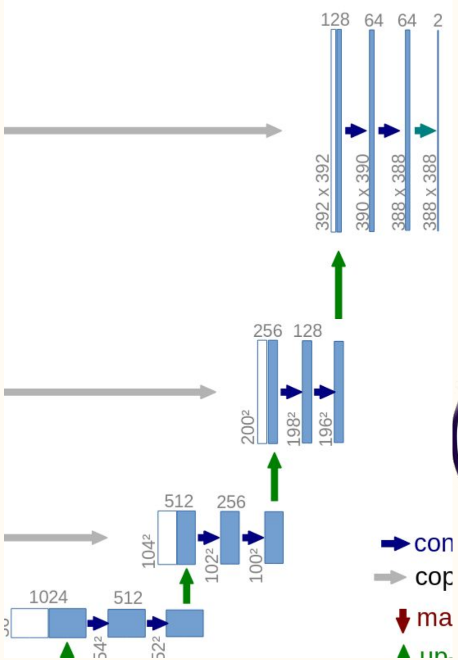
```
class UpSample(nn.Module):
    def __init__(self, in_channels, out_channels):
        super().__init__()
        self.up = nn.ConvTranspose2d(in_channels, in_channels //2 , kernel_size=2, stride=2)
        self.conv = DoubleConv(in_channels, out_channels)

    def forward(self, x1, x2):
        x1 = self.up(x1)
        diffY = x2.size()[2] - x1.size()[2]
        diffX = x2.size()[3] - x1.size()[3]
        x1 = nn.functional.pad(x1, [diffX // 2, diffX - diffX // 2, diffY // 2, diffY - diffY
        x = torch.cat([x2, x1], dim=1)

        return self.conv(x)
```

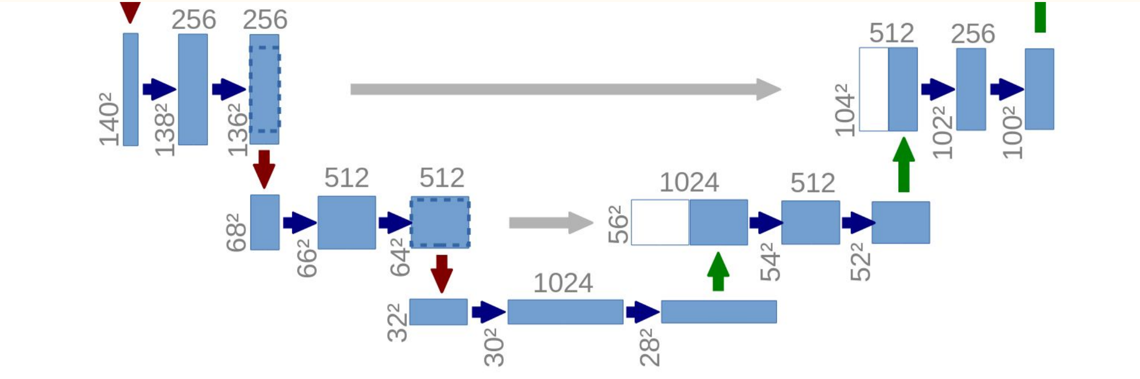
```
self.up_convolution_1 = UpSample(1024, 512)
self.up_convolution_2 = UpSample(512, 256)
self.up_convolution_3 = UpSample(256, 128)
self.up_convolution_4 = UpSample(128, 64)

self.out = nn.Conv2d(64, num_classes, kernel_size=1)
```




I will be your better half...

Skip Connections



- High resolution features from contracting path are concatenated with upsampled output.
- Propagate contextual information at similar abstraction level.
- Feature map from the contracting path is cropped to match the dimension of the expanding path.

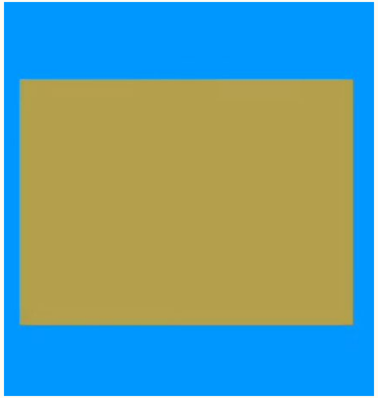
Since downsampling reduces spatial information, skip connections pass high-resolution features directly to the corresponding decoder layers. This helps recover lost details and improves segmentation accuracy.

Skip Connections

These pixels contains bike

+

This area contains bike



⇒

Combined features

⇒

Exact location of the segmentation mask


No matter how deep you contract, I know you'll bring back every lost detail for me...

More on training.

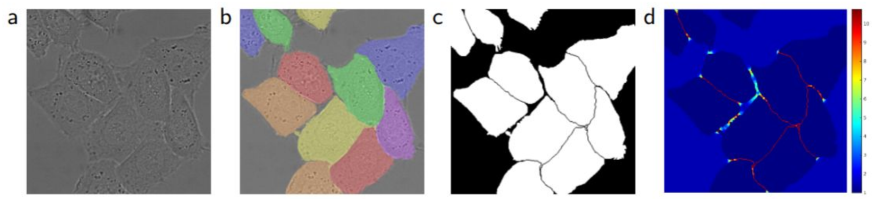
- Relies on strong use of data augmentation for,
    - Use the available annotated sample more efficiently.
    - Allows network to generalize for invariance through deformations.

- Difference in output, input image resolution due to valid convolutions.
- High momentum.
- Large tiles, Smaller Batches.
- Softmax pixel-wise energy function.
- Cross entropy loss.
- Weighted loss for separating borders.
- He Initialization.

$$p_k(\mathbf{x}) = \exp(a_k(\mathbf{x}))/\left(\sum_{k'=1}^{K} \exp(a_{k'}(\mathbf{x}))\right)$$

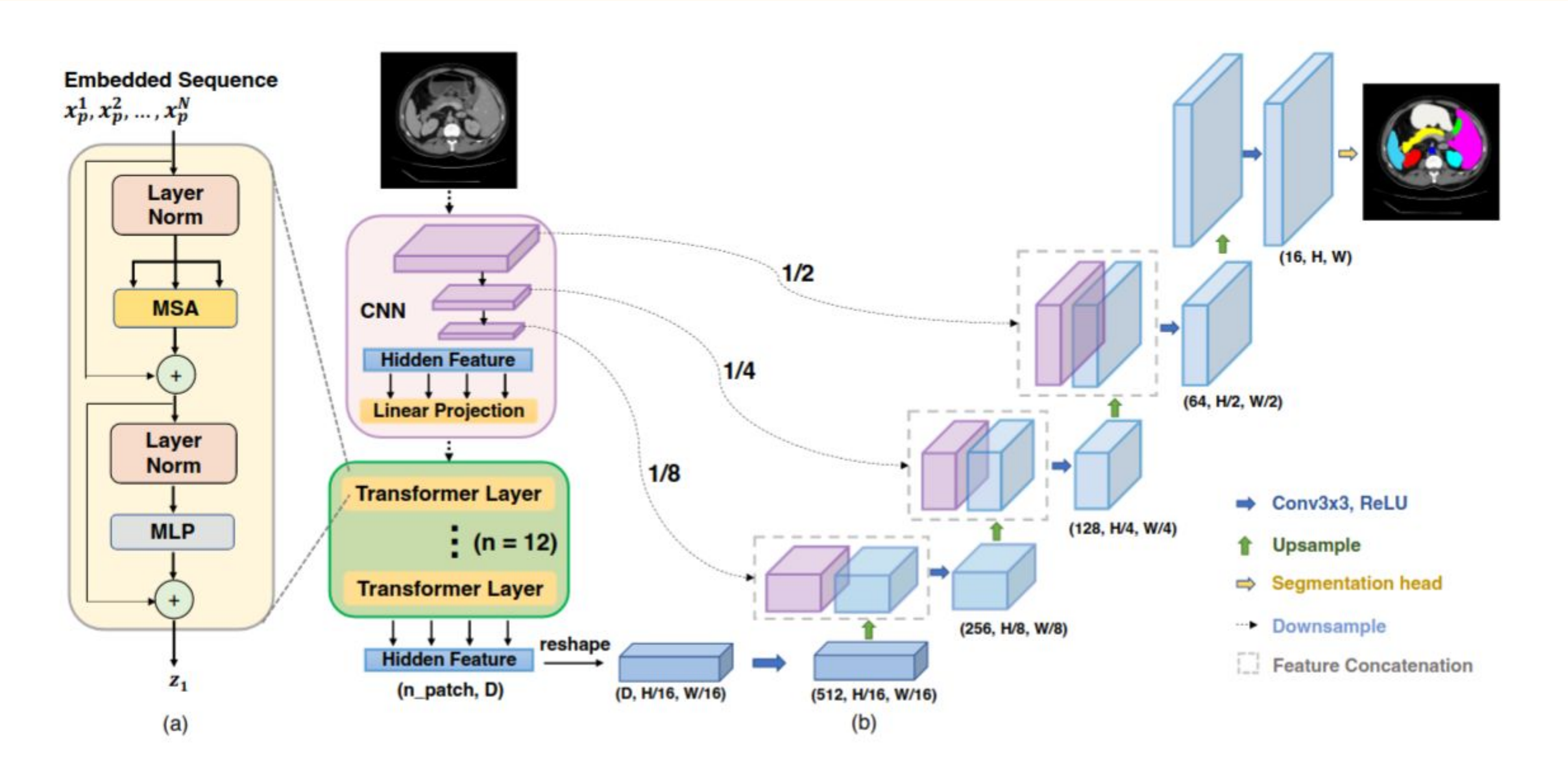
$$E = \sum_{\mathbf{x}\in\Omega} w(\mathbf{x}) \log(p_{\ell(\mathbf{x})}(\mathbf{x}))$$

$$w(\mathbf{x}) = w_c(\mathbf{x}) + w_0 \cdot \exp\left(-\frac{(d_1(\mathbf{x}) + d_2(\mathbf{x}))^2}{2\sigma^2}\right)$$



Code Implementations.

TransU-Net



Source: Cheng et al., 2021, TransUNet

More on TransU-Net

- U-Net generally demonstrates limitations in explicitly modeling long-range dependency, due to the intrinsic locality of convolution operations.
- Transformers are powerful at modeling global contexts, also demonstrate superior transferability for downstream tasks under large-scale pre-training.

ViT's Limited localization abilities.

Hybrid CNN (High level feature representation) + Transformer encoder