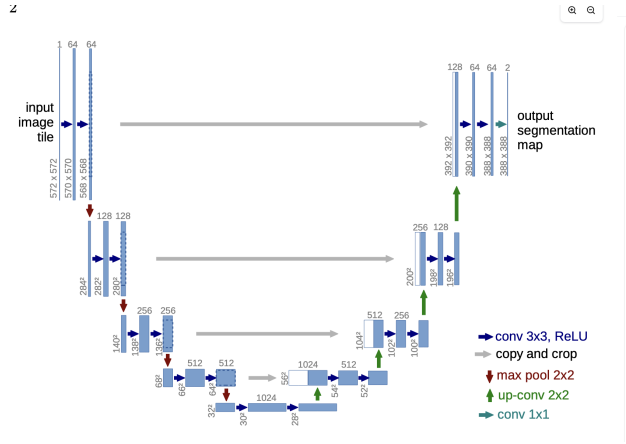
U-net

A Preprint

U-net is based on fully connected convolutional network. The main idea there is to supplement a usual contracting network by successive layers, where pooling operators are replaced by upsampling operators. To localize, high resolutional features are combined with the upsampled output. One significant modification in U-Net architecture is the inclusion of a large number of feature channels in the upsampling path. This design choice enables the network to effectively transfer contextual information to the higher-resolution layers. As a result, the expansive path becomes approximately symmetric to the contracting path, forming a characteristic U-shaped architecture. The network avoids the use of fully connected layers and operates solely on the valid region of each convolution. Consequently, the resulting segmentation map includes only those pixels for which the complete context is available in the input image. This approach supports the segmentation of arbitrarily large images through an overlap-tile strategy. To handle predictions at the image borders, the network extrapolates the missing context by mirroring the input image(Mirroring is a form of intelligent data extrapolation. It assumes that the structures just outside the image are likely to be symmetrical to the structures just inside. For many biological patterns (like cells, tissue), this is a far more reasonable assumption than assuming there is a black void. It provides the network with plausible context to make an accurate prediction even for the pixels right at the very edge.). This tiling method is essential for processing high-resolution images, as it circumvents the memory limitations of the GPU.



U-net also do not have this huge 1D vector at the end due to losing spatial information if flattening. By not having them architecture is capable of working with no fixed size and handling spatial dimensions. At the final layer a 1x1 convolution is used to map each 64 component feature vector to the desired number of classes. In total the network has 23 convolutional layers.

The energy function is computed by a pixel-wise soft-max over the final feature map combined with the cross entropy loss function:

$$p_k(x) = \frac{\exp(a_k(x))}{\sum_{i=0}^{K} \exp(a_i(x))}$$

where $a_k(x)$ enotes the activation in feature channel k at position x, K - number of classes.

Authors also pre compute weight map for each GT segmentation o compensate the different frequency of pixels from a certain class in the training data set, and to force the network to learn the small separation borders.

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

where w_c - class frequencies, d_1 - the distance to the border of the nearest cell and d_2 - the distance to the border of the second nearest cel. w and σ set as 10 and 5 respectively.

How Images Are Split and Processed: The Overlap-Tile Strategy

The U-Net architecture employs valid convolutions, meaning it does not use zero-padding. As a result, the output segmentation map is smaller than the input tile, since border pixels are lost at each convolutional layer. For example, an input tile of size 572×572 produces an output map of size 388×388.

To enable seamless segmentation of arbitrarily large images, the network adopts an overlap-tile strategy. This approach is executed as follows:

1. Tile Selection: A fixed-size input tile (e.g., 572×572 pixels) is selected from the larger source image.
2. Prediction: This tile is passed through the trained U-Net, which outputs a smaller segmentation map corresponding to the central region of the input tile where full context is available.
3. Sliding with Overlap: The next tile is chosen such that it overlaps with the previous tile, and the network processes this tile similarly.
4. Stitching Outputs: This process continues across the entire image. The final segmentation map is constructed by stitching together all valid output patches. Due to the overlap, the transitions between tiles are smooth and consistent.
5. Handling Borders: For tiles near the image boundaries, some contextual information is missing. To address this, the input image is mirrored at the edges, allowing the network to make reasonable predictions even at the borders.