Supervised Deep Learning methods for semantic image segmentation

Study carried out by the Data Science Practice
Special thanks to Ali AHMAD

canopee
empowering *ecosystem*

www.canopee-group.com

Summary

Introduction	**1**
1. What is image segmention ?	**1**
2. Supervised deep learning segmentation architectures	**1**
2.1. Backbone networks	2
2.2. Fully Convolutional Network (FCN) Models	2
2.3. Encoder-Decoder Based Models	3
2.4. Multiscale and Pyramid Network Based Models	4
2.5. Dilated Convolutional Models	5
2.6. Transformers Based Models	7
2.7. Other Models	8
3. Segmentation evaluation metrics	9
4. Deep learning segmentation loss functions	**9**
5. Use Case : Semantic segmentation of Drosophila images	**10**
5.1. Dataset preparation	10
5.2. Studied deep learning architectures and their configuration	11
5.3. Results and discussion	11
Conclusion	**12**
References	**14**

canopee
empowering *ecosystem*

www.canopee-group.com

Introduction

Image segmentation plays a fundamental role in a wide range of computer vision applications, such as medical image analysis, robotic perception, video surveillance, etc. It provides key information for the overall understanding of an image. Many traditional image segmentation methods have been proposed in the literature, including thresholding, region-based segmentation, region growing, k-means ckustering, watershed methods and edge detection segmentation, etc [1, 2]. These methods use the knowledge of image processing and mathematics to segment the image in a simple and fast way. However, their accuracy cannot be guaranteed in terms of details. In recent years, deep learning methods have given rise to a new generation of image segmentation models with remarkable improvements in the field, as they achieve a higher accuracy rate than traditional and classical methods. In this note, we present the advances of deep learning architectures in the field of image segmentation. We describe in detail the variants of deep learning models used for semantic image segmentation and compare some of the most widely used methods for a real use case of multi-class semantic segmentation of Drosophila images.

This note is organized as follows. First, we define image segmentation and its variants. Next, we review existing supervised deep learning segmentation methods following their architecture design. Then, we compare some of these deep learning architectures for multi-class image segmentation of Drosophila as a use case. Finally, we end with a conclusion that summarizes the note.

1 What is image segmentation?

Image segmentation is defined as a specific image processing technique used to divide an image into two or more meaningful regions. It is the process of assigning a label to each pixel in the image, so that pixels with the same label are linked by a visual or semantic property. There are three types of image segmentation: semantic, instance and panoptic segmentation. In semantic segmentation, each pixel is classified into one of the predefined classes so that pixels belonging to the same class belong to a single semantic entity in the image. In this mode, the result cannot differentiate or count two or more objects belonging to the same entity (see Fig. 1.b). However, in instance segmentation, where each distinct object is segmented separately, can typically handle the tasks related to countable things. It can detect each object or instance of a class present in an image and assign it a unique identifier (Fig. 1.c). Panoptic segmentation is a combination of instance segmentation and semantic segmentation. It assigns two labels to each pixel in an image - (i) a semantic label (ii) an instance identifier. Pixels with the same label are considered to belong to the same semantic class and the instance identifiers differentiate its instances (see Fig. 1.d).

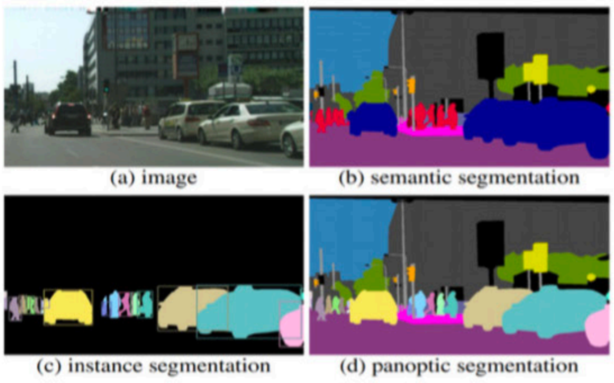

(a) image
(b) semantic segmentation
(c) instance segmentation
(d) panoptic segmentation

Figure 1: Types of image segmentation.

2 Supervised deep learning segmentation architectures

The continued success of deep learning techniques in a variety of advanced computer vision tasks, such as supervised convolutional neural networks (CNNs) for image classification, has prompted exploration of the capabilities of these networks for pixel-level labeling problems such as segmentation. Briefly, CNN is a type of neural network designed to work with images [3]. It consists of a succession of layers: an input layer (the image at the input of the network), an output layer with an output activation function (the decision of the network) and a hidden layer composed of many convolutional layers, correction layers (activation functions), pooling layers and fully connected or dense layers. Further description of the CNN architecture and its components could find in the previous note [4].

The main advantage of the deep learning techniques is their ability to learn appropriate feature representations of the data for a given problem rather than handcrafted features. This gives them an advantage over traditional methods, which require expertise, effort, and often too much fine tuning to make them work for a given scene. These advanced deep learning segmentation methods could be organized into the following categories:

- Fully Convolutional Network (FCN) Models
- Encoder-Decoder Based Models
- Multiscale and Pyramid Network Based Models
- Dilated Convolutional Models
- Transformer based Models
- Other Models

In the following, we present the popular CNN-based backbone networks and an overview of the segmentation architectures of each category.

2.1 Backbone networks

Several deep networks have made significant contributions to the field and have become standards for a wide range of applications. These networks are currently used as building blocks (backbones) for many segmentation architectures. In this section, we briefly describe a number of these networks, in particular AlexNet [5], VGG-16 [6], ResNet [7], among other CNN-based deep architectures, such as GoogLeNet [8], MobileNet [9], etc.

AlexNET it is the pioneering deep CNN network that won the ILSVRC-2012 data challenge with an accuracy of 84.6%. AlexNEt architecture described in [5] is relatively simple. It consists of five convolutional layers followed by a Rectified Linear Units (ReLUs) as non-linearities, max-pooling and three fully-connected (dense) layers. Figure 2 shows that AlexNET architecture.

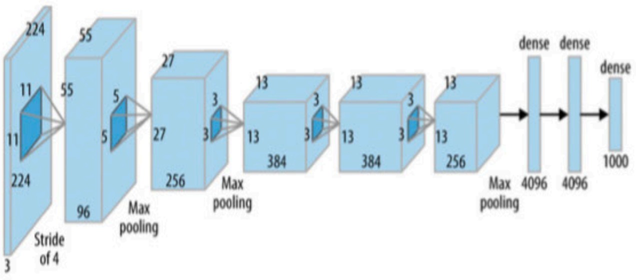


Figure 2: AlexNet Convolutional Neural Network architecture.

Visual Geometry Group (VGG-16) is a model introduced by the Visual Geometry Group (VGG) from the University of Oxford. It was among the top-5 winner of the ILSVRC-2013 Challenge with an accuracy of 92.7%. As shown in the figure 3, VGG16 architecture consists of 16 layers: 13 convolution layers divided into 5 convolutional blocks and 3 dense layers. Each convolutional layer is followed by a ReLu activation function and each convolution block is followed by a max pooling of 2×2.The main advantage of the VGG-16 architecture is that it uses a stack of convolution layers with small receptive fields in the first layers instead of a few layers with large receptive fields. This allows for fewer parameters and more non-linearities between them. Thus making the decision function more discriminative and the model easier to train.

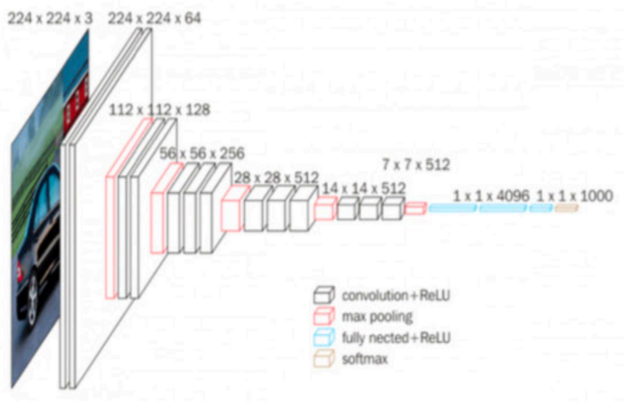


Figure 3: VGG16 architecture

Residual neural network (ResNet) It is released by Microsoft Research and won the ILSVRC-2016 challenge with an accuracy of 96.4%. The network is well known due to its deep architecture, with variants having 50 or 152 layers (ResNet50, ResNet152). The central idea of ResNet is that it introduces a residual block that skips one or more layers (see figure 4). These blocks address the problem of training a really deep architecture by introducing identity skip connections, so that layers can copy their inputs to the next layer. This ensures that the next layer learns something new and different from what the input has already encoded. In addition, this kind of connections help overcoming the vanishing gradients problem.

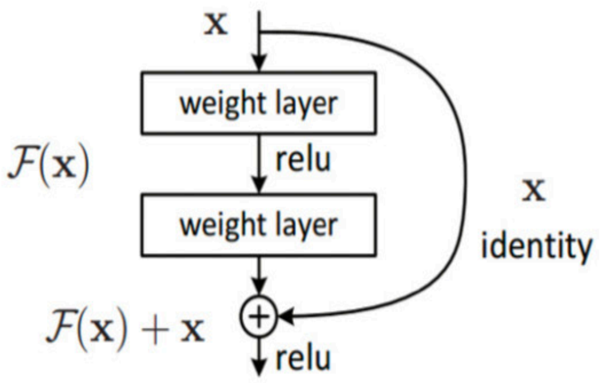

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 4: Residual block principle.

2.2 Fully Convolutional Network (FCN) Models

FCN is one of the first proposed models for end-to-end semantic segmentation [10] . It was first implemented on the PASCAL VOC 2011 segmentation dataset challenge [11] and achieved a pixel accuracy of 90.3% and a mean IOU of 62.7% (see evaluation metrics description in the section 3). In the FCN architecture, the standard image classification models such as VGG, AlexNet or GoogleNet are converted to fully convolutional by replacing all fully connected (dense) layers with a 1×1 convolution layer such that the model outputs a spatial segmentation map instead of classification scores (see Fig. 5). FCN uses the transposed convolution layers (see Fig. 6) to upsample the feature map of the last convolution layer and restore it to the same size of the input image. There are three variants of the FCN: the FCN8, the FCN16 and the FCN32. In FCN32, the output of the CNN network are directly upsampled to the size of the imput image. In FCN8 and FCN16, skip connections are used in which feature maps from the final layers of the model are upsampled and fused with feature maps of earlier layers, the model combines semantic information (from deep, coarse layers) and appearance information (from shallow, fine layers) in order to produce accurate and detailed segmentations (see Fig. 7).

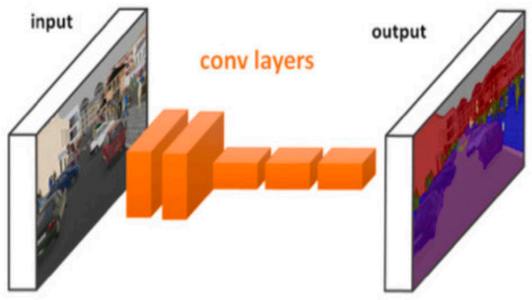


Figure 5: The structure of the fully convolutional network (FCN).

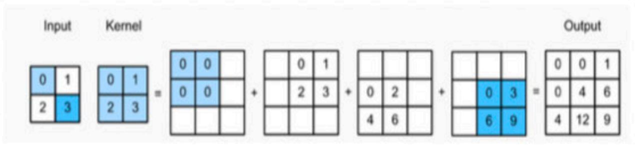


Figure 6: Principle of the transposed convolution, also known as convolution with fractional strides. The process is achieved by dilating the input space.

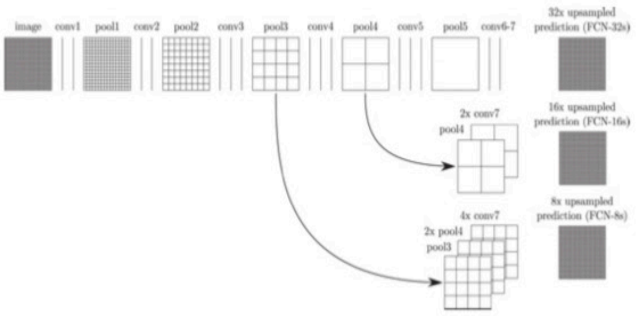


image
conv1
pool1
conv2
pool2
conv3
pool3
conv4
pool4
conv5
pool5
conv6-7
32x upsampled prediction (FCN-32s)
2x conv7
pool4
16x upsampled prediction (FCN-16s)
4x conv7
2x pool4
pool3
8x upsampled prediction (FCN-8s)

Figure 7: FCN32, FCN16 and FCN8. Skip connections combine coarse and fine information to generate detailed segmentation. Figure reproduced from [10]

Despite the success and flexibility of the FCN model, it has some limitations such as it is too computationally expensive for real-time inference, it does not account useful global context information,, and it is not easily generalizable to 3D images.

2.3 Encoder-Decoder Based Models

Encoder-decoders [12] are a family of models that learn to map data-points from an input domain to an output domain via a two-stage network (Fig. 8): the encoder takes the image as input and output a compressed representation in the latent space while the task of the decoder is to semantically project the discriminative features from the latent space (lower resolution) learnt by the encoder onto the pixel space (higher resolution) and generate pixel by pixel segmentation prediction to output a segmentation map.

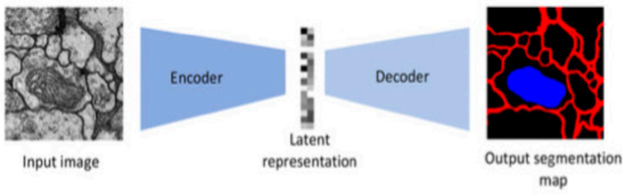


Figure 8: A simple Encoder-Decoder model architecture.

The majority of DL-based segmentation models are based on the encoder-decoder architecture. Generally speaking, all take a classification network as an encoder, such as VGG16, and remove its fully connected layers (see Fig. 3). This part of the network produces low-resolution image representations or feature maps. The challenge is to learn how to decode or map these low-resolution images into pixel-by-pixel predictions for segmentation via a decoder network. The connection of encoder-decoder parts has been the focus of researchers in recent years, leading to a wide variety of encoder-decoder based architectures available in the literature. In the following, we describe some of these architectures among other developed models.

SegNet [13], a fully convolutional encoder-decoder architecture for image segmentation (see Fig. 9)) was launched in 2015 to compete with the FCN network on complex indoor and outdoor images segmentation. It consists of a VGG16 network as an encoder, and a corresponding decoder network followed by a pixel-wise classification layer via a softmax output activation function. The key in SegNet lies in the way the decoder upsamples the low-resolution input feature maps. Specifically, it uses the pooling indices computed in the max pooling step of the corresponding encoder to perform non-linear up sampling via unpooling as shown in the figure 10.

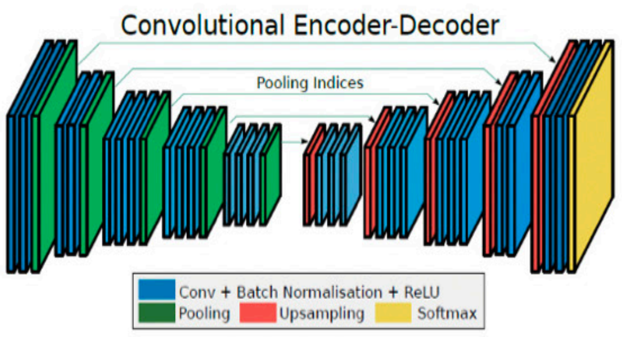

Convolutional Encoder-Decoder
Pooling Indices
Conv + Batch Normalisation + ReLU
Pooling
Upsampling
Softmax

Figure 9: SegNet architecture: the left part is the encoder and the right part is the decoder. Skip connections are used to transmit the pooling indices from the encoder to the corresponding block in the decoder part. Reproduced from [13].

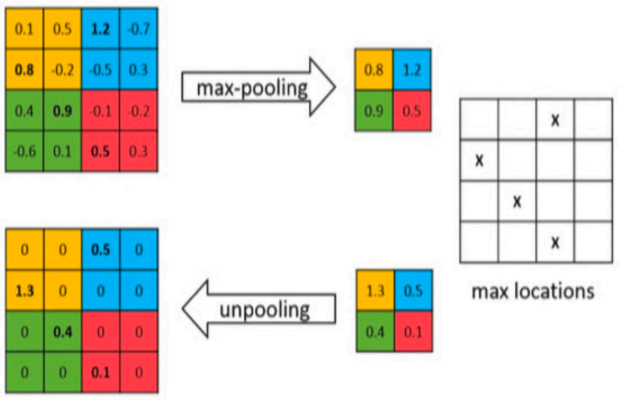


Figure 10: Upsampling process in the SegNet decoder using the 2×2 max pooling locations.

A limitation of the SegNet is the loss of fine-grained image information, due to the loss of resolution through the encoding process. This limitation were efficiently overcome by the Unet network [14], that were initially developed for medical/biomedical image segmentation, but is now also being used outside the medical domain. The winner of the 2015 ISBI cell tracking challenge is characterized by an encoder with a series of convolution and max pooling layers (5 convolution blocks each containing two convolution layers with ReLu activation and followed by 2×2 max pooling). The decoding layer contains a mirrored sequence of convolutions and transposed convolutions (see Fig. 6). It implements skip connections to copy the uncompressed feature maps from encoding blocks to be concatenated with the feature maps of their mirrored counterparts in the decoding blocks, as shown in the figure 11. These skip connections improve detail retention.

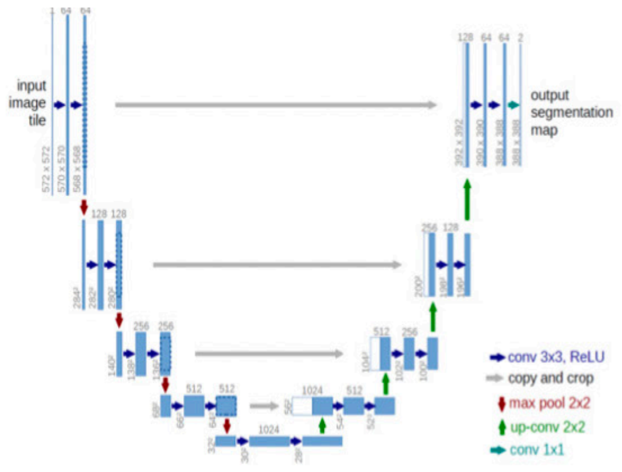


Figure 11: Unet architecture. Reproduced from [14].

Other extensions of the U-Net have been developed for different types of images and problem fields. For instance, the Unet-3D [15], a U-Net based architecture for 3D images segmentation, the Recurrent residual convolutional neural network based on U-net (R2Unet) [16], the use of residual instead of convolutional blocks in the encoder and decoder network (Linknet) [17], the Unet with attention mechanism that learns to softly weight multiscale features at each pixel location (Attention-unet) [18], etc. Other variants of Unet have also been developed for segmentation of nuclei and cellular fluorescence microscopy images, such as 2D/3D Stardist [19] and Cellpose [20].

2.4 Multiscale and Pyramid Network Based Models

Multiscale image analysis is of great interest in computer vision, especially in natural scene segmentation applications where the size of the object of interest is highly unpredictable. For example, real-world objects can vary in size, and they can appear larger or smaller depending on their location relative to the camera. In a standard CNN architecture small-scale features are captured in the early layers, while as one progresses deeper in the network, the features become more specific for larger objects. Therefore, it is often useful to extract information from feature maps at different scales to create segmentations that are generalized and scale invariant.

One of the best-known architectures that meets these conditions is the Feature Pyramid Network (FPN) [21] developed by Facebook AI Research (FAIR). FPN is originally designed for object detection and has also been applied to image segmentation [22]. The architecture of the FPN is shown in figure 12. The FPN architecture for image segmentation is composed of a bottom-up pathway, a top-down pathway and lateral connections. The bottom-up pathway (encoder) is the typical convolutional network for feature extraction and it could be used as VGG or ResNet backbone network. It is composed of many convolution blocks each has many convolution layers. As we move up, the spatial dimension is reduced by 1/2 (i.e., double the stride) and extracts the low-resolution feature map layers. In the lateral connections, a 1 × 1 convolution is applied on the featue maps of each bottom-up convolution block to reduce channel depth to 256. These are added element-wise with the upsampled feautre map layers using nearest neighbors upsampling from the top-down pathway. The latest is the decoding part, the feature maps combines these low-resolution feature map layers that has semantically strong features, with the previously upsampled image that has semantically weak features. Due to this, the feature pyramid reaches semantic features at all levels. In the third column (see Fig. 12), a 3 × 3 convolution is applied two times to the feature map layers at each stage of the top-down pathway in order to reduce the aliasing effect of upsampling. This step genarates a feature map layers of 128 channels at each level that are upsampled to 1/4 and concatenated together. As a next step, the number of output channels is reduced to K, with K is the number of classes, utilizing 1 × 1 convolutions and softmax activation function. Finally, these predictions are upsampled to the original image size using bi-linear interpolation.

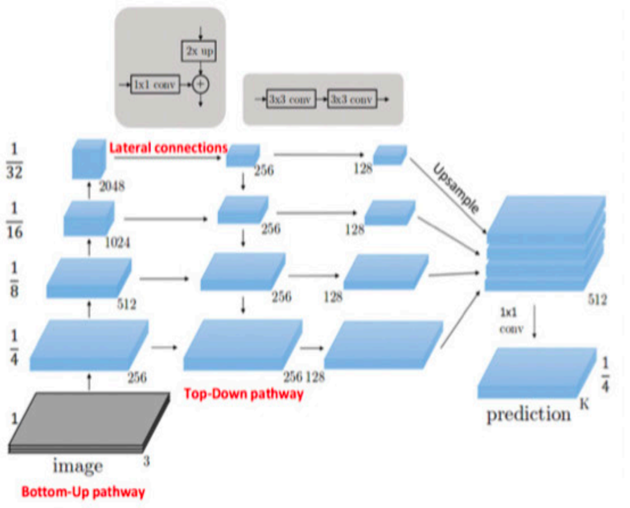


Figure 12: Illustration of the Feature Pyramid network (FPN) architecture for semantic image segmentation.

Another commonly used architecture as a multi-scale image segmentation model is the Pyramid Scene Parsing Network (PSPNet) [23]. Scene parsing is the process of segmenting and analyzing an image into various visual areas corresponding to semantic categories. The main characteristic of PSPNet is the use of the Pyramid Pooling Module (PPM). This module helps the model to capture the global context of the image, allowing it to classify pixels based on the global information. In the original design of PSPNet (see Fig. 13), multiple patterns are extracted from the input image using a ResNet (other backbone networks are also supported) as a feature extractor, with a dilated network (see next section 2.5). These feature maps are then fed into a pyramid pooling module to distinguish patterns of different scales which are 6, 3, 2, and 1 for colours green, blue, orange and red respectively. The feature maps is pooled at these scales using average pooling then they are convolved with 1 × 1 filters to reduce the feature depth. The outputs of the pyramid levels are upsampled and concatenated with the initial feature maps to capture both local and global context information. Finally, the feature layer is classified by 1 × 1 convolution kernel with a softmax, thus, the prediction result of each pixel in the image is obtained.

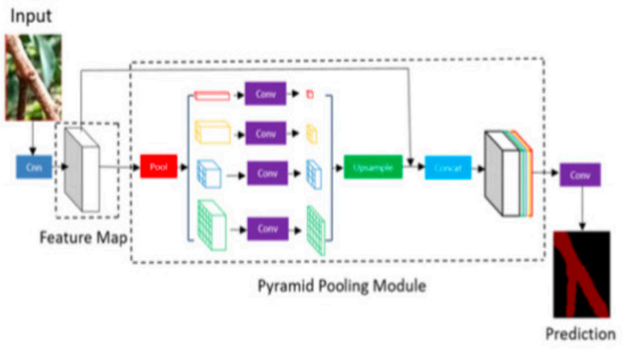


Figure 13: PSPNet architecture. Reproduced from [24].

Beside these two models, other models are also used for multiscale image segmentation, such as the RefineNet [25], the Dynamic Multiscale Filters Network (DMNet) [26], etc.

2.5 Dilated Convolutional Models

Despite the fact that pixel-wise segmentation is effective, there is still a limitation that can affect the performance of the segmentation. This concerns the small size of the kernels, which does not capture all the contextual information of the input image and focuses only on the local information. In a standard image classification pipeline, this problem can be solved by using pooling layers that increase the sensory area of the kernels by downsampling the original image. However, in segmentation, this reduces the sharpness of the segmented output. The alternative use of large kernels tends to slow down the computational process, as it greatly increases the number of model training parameters. This issue has been addressed through the dilated (a.k.a. "atrous") convolution [27]. This technique uses the dilation rate as a new parameter for the standard convolutional layer that allows contextual information to be extracted by increasing the receptive field size without increasing the number of parameters. For example, a 3 × 3 kernel with a dilation rate $r = 2$ will have the same receptive field size as a 5 × 5 kernel while using only 9 parameters instead of 25, thus enlarging the receptive field without increasing the computational cost (see Fig. 14). By the way, it is important to note that standard convolutions are only 1-dilation rate ($r = 1$).

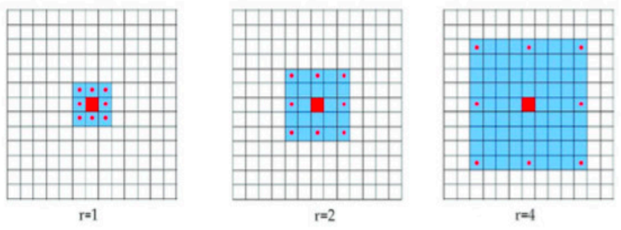


Figure 14: A 3 × 3 dilated convolution kernels with various dilation rates.

Segmentation models based dilated convolutions are commonly used for real-time image segmentation applications. Some of the most important include the DeepLab family by Google [28], multiscale context aggregation [27], the Efficient Network (ENet) [29], etc. The DeepLab family is one of the most efficient segmentation models and includes 4 versions: v1, v2, v3 and v3+. The Deeplabv1 and DeepLabv2 (see Fig. 15) use dilated (atrous) convolution to address the decreasing resolution in the network caused by max-pooling and striding. They use also the fully connected Conditional Random Field (CRF) [30] to fine tune the segmentation results by considering the label of a pixel is not only dependent on its own intensity values but also the values of neighboring pixels (see Fig. 16).

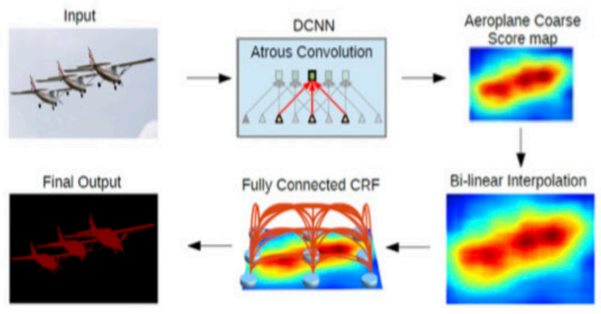


Figure 15: Deeplabv1&v2 model architecture. First, the input image goes through the network with the use of dilated convolutions. Then the output from the network is bilinearly interpolated and goes through the fully connected CRF to fine tune the result in order to obtain the final predictions.

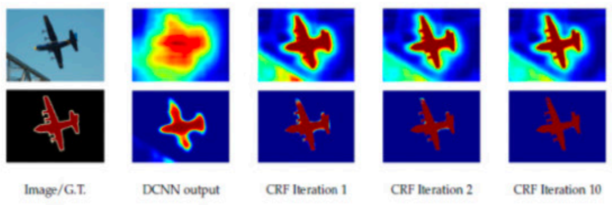
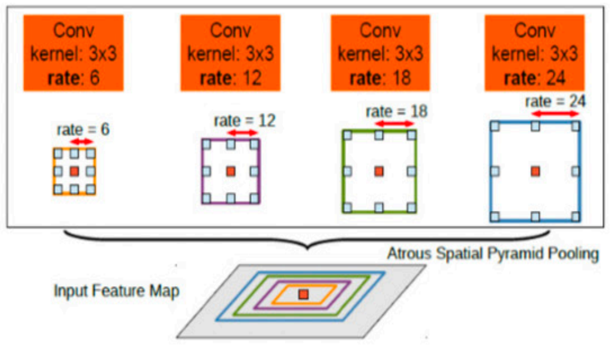


Figure 16: Impact of Fully connected CRFs to fine tune the final segmentation prediction. Top: Score map (input before softmax function), Bottom: Prediction map (output of softmax function). With 10 iteration of CRF, those small areas with different colors around the aeroplane are smoothed out successfully.

Deeplabv2 is implemented with an additional key feature, the Atrous Spatial Pyramid Pooling (ASPP), which probes an incoming convolutional feature layer with filters at multiple sampling (dilation) rates, capturing objects as well as multi-scale image context to robustly segment objects at multiple scales (see Fig. 17).



Conv kernel: 3x3 rate: 6
Conv kernel: 3x3 rate: 12
Conv kernel: 3x3 rate: 18
Conv kernel: 3x3 rate: 24
rate = 6
rate = 12
rate = 18
rate = 24
Atrous Spatial Pyramid Pooling
Input Feature Map

Figure 17: ASPP principle: parallel atrous convolution with different rate applied to the input feature map, and fused together.

For the DeepLabv3 [31], apart from using atrous convolution, it uses an improved ASPP module by including batch normalization and image-level features without the use of CRF as used in V1 and V2. In its implementation, the output from the ASPP network is passed through a 1 × 1 convolution to get the actual size of the image which will be the final segmented mask for the image after interpolation (see Fig.18). These improvements help in extracting dense feature maps for long-range contexts by increasing the receptive field exponentially without losing the spatial dimension and improves performance on segmentation tasks. However, it increases the computational time.

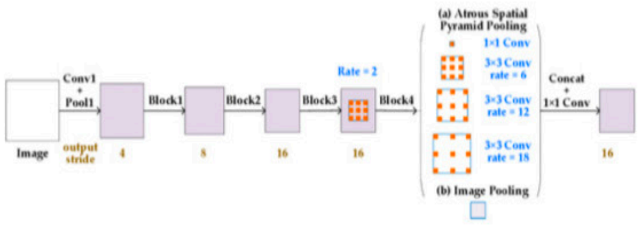


Figure 18: The structure of atrous spatial pyramid pooling module used in Deeplabv3. The module consists of two steps including (a) atrous convolution and (b) image Global average pooling (image-level features), and produces the final output by a convolution layer after concatenation of feature maps

Finally, DeepLabV3+ [32] (Fig. 19) is implemented with an encoder-decoder architecture including dilated separable convolution composed of a depthwise convolution (spatial convolution for each channel of the input) and pointwise convolution (1×1 convolution with the depthwise convolution as input). This fixes the limitation of DeepLabV3 related to the high computational time to process high-resolution images. The application of the depthwise separable convolution to both atrous spatial pyramid pooling and decoder modules results in a faster and stronger encoder-decoder network for semantic segmentation.

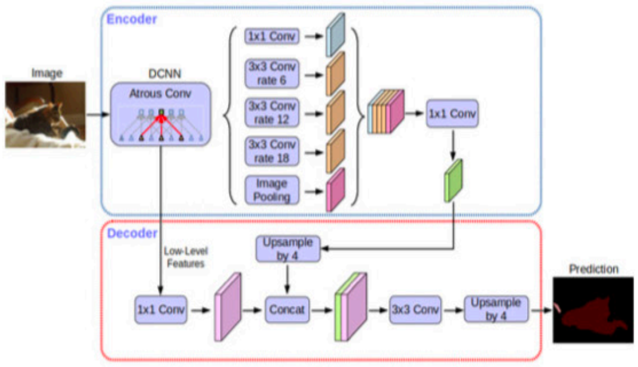


Figure 19: Deeplabv3+ model architecture. Reproduced from [32]

2.6 Transformers based Models

Transformers [33] are deep learning models that are primarily used for natural language processing (NLP) tasks [34]. The amazing results of transformers on NLP, thanks to their powerful self-attention mechanism capabilities (see this papers [35, 36] for attention mechanism description), have intrigued researchers who have explored their application to computer vision problems such as image classification and segmentation. Transformers allow the extraction of global context and long-range dependencies between input elements and support parallel processing compared to CNN methods that tend to focus on local details. Furthermore, their simple design allow multiple modalities (e.g., images, videos, text, and speech) to be processed using similar processing blocks and demonstrates excellent scalability to very large capacity networks and huge data sets compared to CNN based methods.

Many Transformers-based deep learning models have been developed in the literature for semantic image segmentation. For instance, TransUnet [37], is a U-shaped encoder-decoder architecture similar to Unet, that instead of using full CNN blocks, employs an hybrid CNN-Transformer encoder. This allows learning both the high-resolution spatial information from the CNNs and the global contextual information from the Transformers. In its design, the vision Transformers (ViT , see more details here [38] regarding the ViT architecture) are fed with the feature maps created by the CNN as input instead of the raw images. As shown in the figure 20, the CNN is firstly used as a feature extractor to generate the feature maps. For each level of the feature extractor, the output feature maps is then concatenated to the decoder path of the same level. Then, the feature maps are vectorized (tokenized) into a 2D embedding of shape (n_{patch}, D) by linear projection, and D is the total length of the embedding. After obtaining the embeddings, they are fed into 12 Transformer layers to encode less short-range and more long-range information from the image. Each layer of the Transformer uses multi-head self-attention (MSA) and multi-layer perceptron (MLP) modules. Lastly, to prepare for the up-sampling path, the output is reshaped to (D, H/16, W/16). H/16 and W/16 mean that the heights and widths by this time have been shrunk by 16 times because of the previous operations. The decoding path is similar to the Unet decoder. At the output, a softmax function is used to generate the final segmentation map.

Unlike TransUNet, other models uses purely Transformers in their architectures such as Swin-Transformer [39], Swin-Unet [40], Segmenter [41], Segformer [42], etc. In this note we will focus on describing the Swin-Unet. More details about the other architectures could be found in the original papers. Swin-Unet, as shown in figure 21,is a pure Transformer-based U-shaped architecture that consists of encoder, bottleneck, decoder, and skip connections. They are all built based on Swin Transformer block (Hierarchical Vision Transformer using Shifted Windows) [39]). As shown in figure 22, a Swin Transformer block is composed of Layer Normalisation (LN), multi-head self attention module, residual connection and 2-layer MLP with GELU activation function (Gaussian Error Linear Unit) as non-linearity. The window based multi-head self attention (W-MSA) module and the shifted window-based multi-head self attention (SW-MSA) module are applied in the two successive Transformer blocks, respectively. The advantage of using Swin Transformers in the segmentation architecture is that the shifted windows divides the image into non-overlapping sub-windows, resulting in better computational speed and improves the performance and efficiency of Transformers by achieving scale-invariance.

Briefly, the architecture design of Swin-Unet is as the following. In the encoder part, the input image is split into non-overlapping patches with patch size of 4×4 in order to transform it into sequence embeddings. In this way, the feature dimension of each patch becomes 4 × 4 × 3 = 48 (images originally RGB with three channels). Furthermore, a linear embedding layer is applied to project feature dimension into arbitrary dimension (represented as C in Fig. 21). The transformed patch tokens pass through several Swin Transformer blocks and patch merging layers to generate the hierarchical feature representations. Specifically, patch merging layer is responsible for downsampling and increasing dimension, and Swin Transformer block is responsible for feature representation learning.

The decoder is composed of Swin Transformer blocks and patch expanding layer. The extracted context features are fused with multiscale features from encoder via skip connections to complement the loss of spatial information caused by downsampling. In contrast to patch merging layer, a patch expanding layer is specially designed to perform upsampling. The patch expanding layer reshapes feature maps of adjacent dimensions into a large feature maps with 2× upsampling. At the end, the last patch expanding layer is used to perform 4× upsampling to restore the dimension of the feature maps similar to the input image (W ×H). Then, a linear projection layer is applied on these upsampled features to output the pixel-level segmentation predictions.

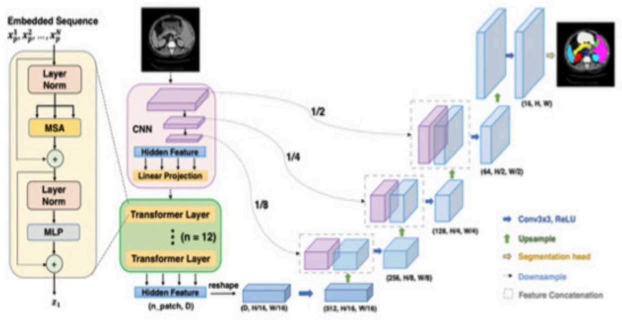
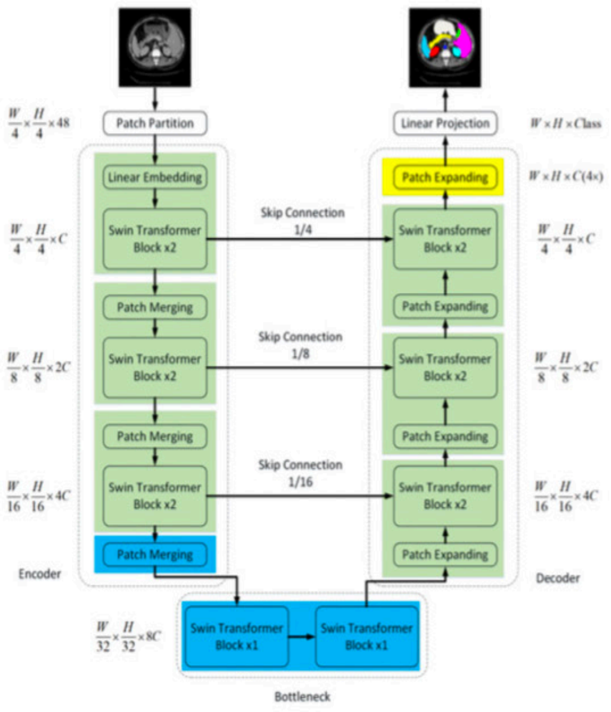


Figure 20: TransUnet architecture design. In the ViT architecture (left), MSA stands for Multi-head Self-Attention, and MLP stands for Multi-Layer Perceptron. Reproduced from [37].



Patch Partition
Linear Embedding
Swin Transformer Block x2
Patch Merging
Swin Transformer Block x2
Patch Merging
Swin Transformer Block x2
Patch Merging
Encoder
Skip Connection 1/4
Skip Connection 1/8
Skip Connection 1/16
Linear Projection
Patch Expanding
Swin Transformer Block x2
Patch Expanding
Swin Transformer Block x2
Patch Expanding
Swin Transformer Block x2
Patch Expanding
Decoder
Swin Transformer Block x1
Swin Transformer Block x1
Bottleneck
$\frac{W}{4}\times\frac{H}{4}\times 48$
$\frac{W}{4}\times\frac{H}{4}\times C$
$\frac{W}{8}\times\frac{H}{8}\times 2C$
$\frac{W}{16}\times\frac{H}{16}\times 4C$
$\frac{W}{32}\times\frac{H}{32}\times 8C$
$W\times H\times Class$
$W\times H\times C(4\times)$

Figure 21: The architecture of Swin-Unet, which is composed of encoder, bottleneck, decoder and skip connections. Encoder, bottleneck and decoder are all constructed based on swin transformer block. Reproduced from [40]

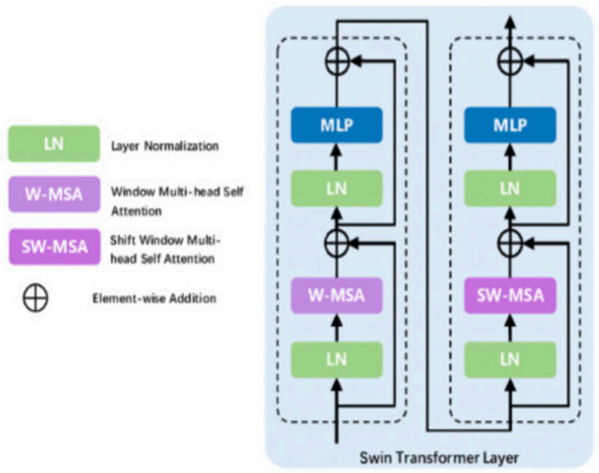


Figure 22: Architecture of Swin trasnformer block.

2.7 Other Models

Previously, we described some of the well-known categories of deep learning semantic segmentation models. Many more popular deep learning segmentation architectures are also available [43]. For example, BUT NOT LIMITED TO, R-CNN-based models that are mainly used to solve object detection and instance segmentation problems such as Mask-RCNN [44], the recurrent neural network (RNN) based models that are useful in modeling the short/long term dependencies among pixels such as Data Associated Recurrent Neural Networks (DA-RNNs) [45] , Generative Adversial Networks (GANs) based segmentation models [46] , attention-based models such as Reverse Attention Network (RAN) [47], CNN with active contour models such as Deep Active Lesion Segmentation (DALS) [48], point cloud-based models such as PointNet [49], multi-modal data fusion models such as FuseNet [50] which combines RGB and depth images (RGB-D) for semantic segmentation, as well as other derived models for panoptic segmentation such as the Panoptic Feature Pyramid Network (PFPN) [44].

3 Segmentation evaluation metrics

Deep learning segmentation models are typically evaluated in terms of quantitative accuracy and computation time. While computation speed is a valuable metric as some systems need to meet the requirements of real-time or high-throughput applications, providing the exact inference time of methods can be considered meaningless as it is highly dependent on hardware (GPU and computer) and backend implementation (Keras, Tensorflow,etc). So far, the evaluation of segmentation models has mainly focused on quantitative metrics and the most common ones are described below:

Pixel Accuracy (PA)

Is the simplest metric used to evaluate the segmentation. It is computed as the ratio of the correctly classified pixels on the total number of pixels. For a total number of classes $C+1$, the pixel accuracy is defined as follow,

$$PA = \frac{\sum_{i=0}^{C} p_{ii}}{\sum_{i=0}^{C}\sum_{j=0}^{C} p_{ij}}, \tag{1}$$

With p_{ii} represent the amount of pixels of class i correctly predicted as class i and p_{ij} is the number of pixels of class i predicted as belonging to class j.

Mean Pixel Accuracy (MPA)

is an extension of pixel accuracy (PA) in which the ratio of correct pixels is computed in a per-class basis and then averaged over the total number of classes.

$$MPA = \frac{1}{C+1}\sum_{i=0}^{C} \frac{p_{ii}}{\sum_{j=0}^{C} p_{ij}}. \tag{2}$$

Intersection over Union (IoU)

Also know as the Jaccard Index which is a standard metric used for segmentation evaluation. It is defined as the area of intersection between the predicted segmentation map A and the ground truth map B, divided by the area of the union between the two maps (see also Fig. 23).

$$IoU = J(A,B) = \frac{|A \cap B|}{|A \cup B|}. \tag{3}$$

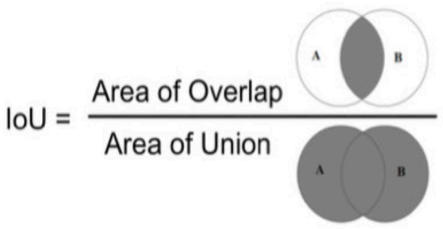


Figure 23: Principle of the Intersection over Union (IoU) metric.

Mean IoU (mIoU) The mIoU is an extension of the IoU metric. It is defined as the average IoU over all classes.

F1-score

The F1-score measure is defined as the harmonic mean between recall and precision at the pixel level and it is computed as follow,

$$\text{F1-score} = 2.\frac{Precision \times Recall}{Precision + Recall}, \tag{4}$$

with the $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$, where TP, FP and FN are the true positive, false positive and false negative respectively.

Dice coefficient

This metric is commonly used in medical image segmentation. It is defined as twice the overlap area of the predicted maps and the ground truth map divided by the total number of pixels.

$$Dice = \frac{2 \times |A \cap B|}{|A| + |B|}. \tag{5}$$

It is very similar to the IoU (Eq. 3) and when applied to binary segmentation maps (only two classes 0 and 1), with foreground as the positive class, the Dice coefficient is identical to the F1 score (Eq. 4).

$$Dice = \frac{2TP}{2TP + FP + FN} = \text{F1-score}. \tag{6}$$

4 Deep learning segmentation loss functions

The purpose of the loss function in a deep learning model is to quantify the difference between predictions and ground truths. The choice of the loss function is extremely important when designing image segmentation-based deep learning architecture, as it triggers the learning process of the algorithm. In the following, we summarize some of the semantic segmentation-based loss functions that have been commonly used and proven to provide state-of-the-art results in different domains [51].

Cross Entropy

Cross-entropy is defined as a measure of the difference between two probability distributions for a given random variable or set of events. It is widely used for classification objective and espacially for equal data distribution among classes scenarios, and as segmentation is pixel level classification it works well. We can define two different loss fonctions of the cross entropy: the binary cross entropy (BCE) or log loss, and the categorical cross entropy (CCE). The BCE is used for binary classification problems (only two classes 0 and 1), and it is defined as follow:

$$BCE(y, \hat{y}) = -[y\ log(\hat{y}) + (1 - y)\ log(1 - \hat{y})]. \quad (7)$$

with $\hat{y}$ refers to the predicted probability value and y refers to the ground truth label. This can be extended to multi-class problems, and the categorical cross entropy loss (CCE) is computed as:

$$CCE(y, p) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=0}^{C} y_{i,c} log(p_{i,c}), \quad (8)$$

where $y_{i,c}$ uses a one-hot encoding scheme of ground truth labels, $p_{i,c}$ is a matrix of predicted probability values for each class. Indices c and i iterate over all classes and pixels, respectively with N is the total number of pixels in 1D.

Cross entropy loss is based on minimising pixel-wise error, where in class imbalanced situations, leads to over-representation of larger objects in the loss, resulting in poorer quality segmentation of smaller objects. Thus other extansions of the cross entropy are developed to deal with this problem and they are explained below.

Weighted and balanced categorical cross entropy

The WCCE is a variant of CCE that is used to solve the problem of overfitting due to the class imbalance (skewed data) that speed the convergence of the loss function. It is used to increase or decrease the relative penalty of a probabilistic false negative for an individual class by using a coefficient (weights). WCCE is defined as,

$$WCCE(y, p) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=0}^{C} w_c y_{i,c} log(p_{i,c}), \quad (9)$$

with w_c is the weight of the class c. The w_c value can be used to tune false negatives and false positives. For example, $w_c > 1$ reduces the number of false negatives, similarly to decrease the number of false positives, $w_c < 1$ is used. Concerning the class-balanced Loss, the weights w_c are set in proportion to the inverse of the number of samples per class: $w_c = \frac{1}{Number\ of\ pixel\ of\ c}$. However, While these versions of loss functions balances the importance of positive/negative example of the classes, but it does not differentiate between easy/hard examples.

Focal loss (FL)

Focal loss (FL) [52] can be seen as an extension of the cross entropy. It is designed to work well for highly imbalanced classes problems. In principle, it down-weights the contribution of easy examples and enables the model to focus more on learning hard misclassified examples. Thus, ensuring that predictions on hard examples improve over time rather than becoming overly confident with easy ones. The focal loss is implemented by adding a modulating factor to the cross-entropy loss as shown below:

$$FL(y, p) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=0}^{C} \alpha_c(1 - p_{i,c})^{\gamma} y_{i,c} log(p_{i,c}), \quad (10)$$

where $\gamma > 0$ is the focusing parameter that reduces the relative loss for well-classified examples and could be set via cross validation. In the special case of $\gamma = 0$, the focal loss becomes equivalent to cross entropy. Similarly, α_c is a weighting factor that generally range from [0,1].It can be set by inverse class frequency or treated as a hyperparameter.

Dice loss

The dice loss is derived from the dice coefficient evaluation metric (Eq. 5). It is widely used in medical image segmentation tasks to address the data imbalance problem. However, it only addresses the problem of foreground/background imbalance, but neglects another imbalance between easy and difficult examples that also severely affects the training process. The dice loss is computed as : $1 - Dice$.

5 Use Case: Semantic segmentation of Drosophila images

5.1 Dataset preparation

In order to compare between the previousely presented deep learning arhitectures, we used the Drosophila neural tissue image annotated dataset [53]. It contains 20 images of 1024 × 1024 pixels and their ground truth acquired with a serial section Transmission Electron Microscopy (ssTEM). It consists of 3 classes: cytoplasm, cell membrane and mitochondria (see Fig. 24). In order to fit with the GPU size, we split the original images into 320, 256 × 256 pixels, no overlapping images. We used 230 for training, 58 for validation and 32 for testing the deep learning segmentation architectures.

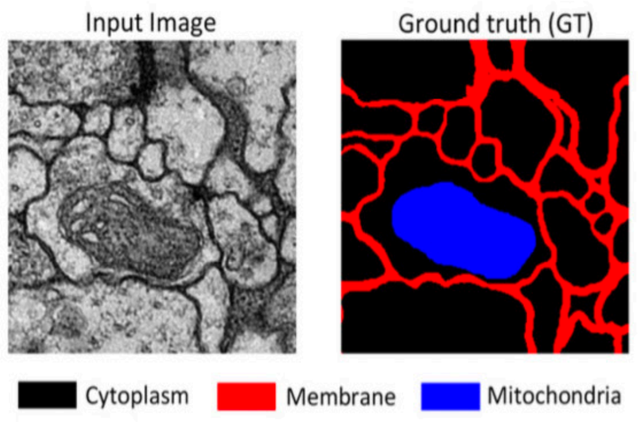


Figure 24: An example of 2D Drosophila image and its grund truth mask with the three classes to be segmented.

5.2 Studied deep learning architectures and their configuration

For the multiclass semantic segmentation of the Drosophila images and without any claim of optimality, we choosed to compare the performance of the following deep learning methods:

- FCN8 from fully convolutional models category,
- SegNet, Unet, R2Unet, Attention-Unet and Linknet from encoder-decoder models category,
- PSPNet and FPN from multiscale and pyramid network models category,
- DeepLabv3+ from dilated convolutional models category,
- TransUnet and Swin-Unet from transformers based models category,

For a fair comparision between the architectures, we trained models with the same hyperparameters, data augmentation process and loss function. The configurations are described below:

Backbone network We unified the CNN using VGG16 as the backbone for all architectures. Only transformer-based architectures, the TransUnet and Swin-Unet, were used as the original implementation described previously in the section 2.6.

Models Hyperparameters The models were trained with the following hyperparameter configurations: batch size = 4, maximum number of epochs = 300, initial learning rate (LR) = 0.001 with the introduction of the *ReduceLROnPlateau* callback function to monitor the evolution of validation loss with the following parameters: factor = 0.8, patience = 5, delta min=0.0001, and LR min=0.0001. This callback is used to reduce the learning rate when a metric has stopped improving. At the output of each model a Softmax output activation function were used to predict 3 pixel classes.

Regularization We also trained the models with the same regularisation such as early stopping by monitoring the validation loss with patient=7 and checkpoint to save the best weights of the models in order to avoid overfitting.

Data augmentation Data augmentation process were applied on the training data to generate sufficient amount of input images and their ground truth for the training. The data augmentation operation applied to this use case included horizontal and vertical flipping, random rotational transformation between 0 and 179° and zoom in up to 50%. During this step, the same transformation is applied to each input image and its corresponding ground truth mask (see Fig. 25).

Other transformations could be applied to the dataset an order to augment the amount of data such as brigthness, color or contrast modification, simulation of camera distortion such as optical distortions and blur, noise addition, etc. However, this type of data augmentation should be applied only on the input image without modifying the GT mask. In addition, it must be chosen correctly without affecting the main problem in question, e.g, if the main discrimination between image regions is based on color representation, data augmentation based on varying the brightness/illumination or colors could affect the segmentation efficiency.

Loss function The used dataset contains highly imbalaced classes with the following pixel frequencies: Cytoplasm: 76.9%, Membrane: 17.7 % , Mitochondria: 5.4 % . For this reason, we used the focal loss (Eq. 10) as a loss function to train the models. We used optimal hyperparameters of the focal loss function as reported in the original study [52] with $\alpha_c = 0.25$, $\gamma = 2$.

Evaluation metrics We used for the models evaluation the mean intersection over union (mIoU) and the intersection over union (IoU) per class (Eq. 3). The inference time were also computed for each segmentation model and they were obtained with an 11th Gen Intel Core i7-11800H 2.30 GHz processor.

5.3 Results and discussion

The quantitative evaluation of the applied segmentation models is shown in the table 1. The reported mIoU and IoU per class metrics are averaged over all 32 test images. Figure 26 shows a visual illustration of the predicted segmentation maps at the output of the models for some test images along with their ground truth. Several remarks can be made from the results:

Evalutation by models category First, the FCN8 based on a fully convolutional network (cyan colored line in the table) gives the lowest performance (*mIoU* = 0.727) among the tested networks. This is due to the fact that it does not consider useful global contextual information and is not able to retrieve fine details from microscopy images. This model may

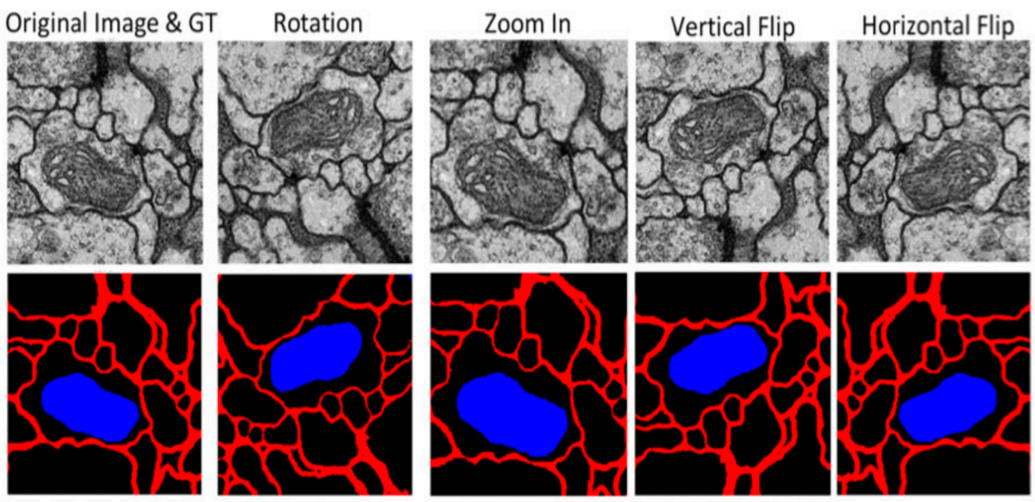


Figure 25: An exemple of data augmentation of an image and its GT mask.

be more efficient for other topics such as indoor/outdoor image segmentation where image objects contains less structured details.

Second, models based on encoder-decoder architectures (blue colored lines in the table) perform well in segmentation. In these models, the skip connection is used at all stages of the network to transfer the feature maps (low-level and high-level information) from the encoder to the decoder. With the exception of SegNet, which uses pooling indices instead of feature maps for upsampling. This explains why its $mIoU$ = 0.785 is lower than the $mIoU$ of other models from the same family.

Third, with respect to the multi-scale and pyramid network-based models (red colored lines in the table), FPN outperforms PSPNet with a mIoU of 0.833 against 0.795 for PSPNet. We infer that the arrangement of pixels in the masks contains strong semantic locality and a special features that was captured perfectly by FPN than by PSPNet.

Fourth, the DeepLabv3+ model based on dilated convolution (green colored line) leads to a good segmentation performance with a $mIoU$ = 0.821. This shows the advantage of dilated convolution to retain more global contextual information by expanding the receptive field.

Fifth, among the studied transformer-based models (orange colored lines), the Swin-Unet based on Swin transformers outperforms the TransUnet model based on the CNN and the vision Transformer (ViT) with a $mIoU$ = 0.816 against $mIoU$ = 0.778. The results show that the hierarchical representation obtained by the Swin transformers and the shifted window approach effectively transmits the information into the local window and thus improves the efficiency of the model.

Global evaluation The best computed performance belongs to R2Unet with $mIoU$ = 0.840, slightly better than Unet with $mIoU$ = 0.837 and FPN with $mIoU$ = 0.833. These results are supported by the fact that the R2Unet model uses the power of U-Net, residual networks, and recurrent convolutional neural networks (RCNNs) to preserve fine image details and provide better feature representation. In addition, R2Unet yields the most well balanced IoU per class segmentation results with $IOU_{Cytoplasm}$ = 0.926, $IOU_{Cell\ membrane}$ = 0.747, $IOU_{Mitochondria}$ = 0.848.

Evaluation based on inference speed Table 2 shows the average inference time (in seconds) computed from the Drosophila test images for each employed architecture. The fastest model is DeepLabv3+ with an average time of 0.009 second, thanks to the dilated convolution design that expands the receptive field without increasing computational costs. The DeepLabv3+ method also delivers acceptable segmentation performance with $mIoU$ = 0.821. However, when considering the balance between the per-class IoU measurements, it results in a low IoU = 0.683 for the segmentation of the cell membrane class. Thus, it cannot be considered as a trade-off for efficiency-speed. The best model in this sense is Swin-Unet, thanks to the shifted window strategy, it results in an average inference time of 0.015 second with an $mIoU$ = 0.816 and per class segmentation scores of $IOU_{Cytoplasm}$ = 0.918, $IOU_{Cell\ membrane}$ = 0.713, $IOU_{Mitochondria}$ = 0.818.

Final models selection As a consequence, this comparison allows us to consider that R2Unet is the best model for Drosophila neural tissue image segmentation with an average inference time of 0.074 seconds. By contrast, for high-throughput segmentation applications, the Swin-Unet might be a better choice with 5× faster than the R2Unet.

Conclusion

In this note, we presented an overview of deep learning segmentation methods based on their categories and architecture design. We also compared some of the described deep learning models for multi-class semantic segmentation of Drosophila microscopy images. The results of the comparative study conducted in this note favor R2Unet for offline segmentation and Swin-Unet for high-throughput applications. However, as the supervised deep learning is a data-driven process that solve problems by learning from data, using a differ-

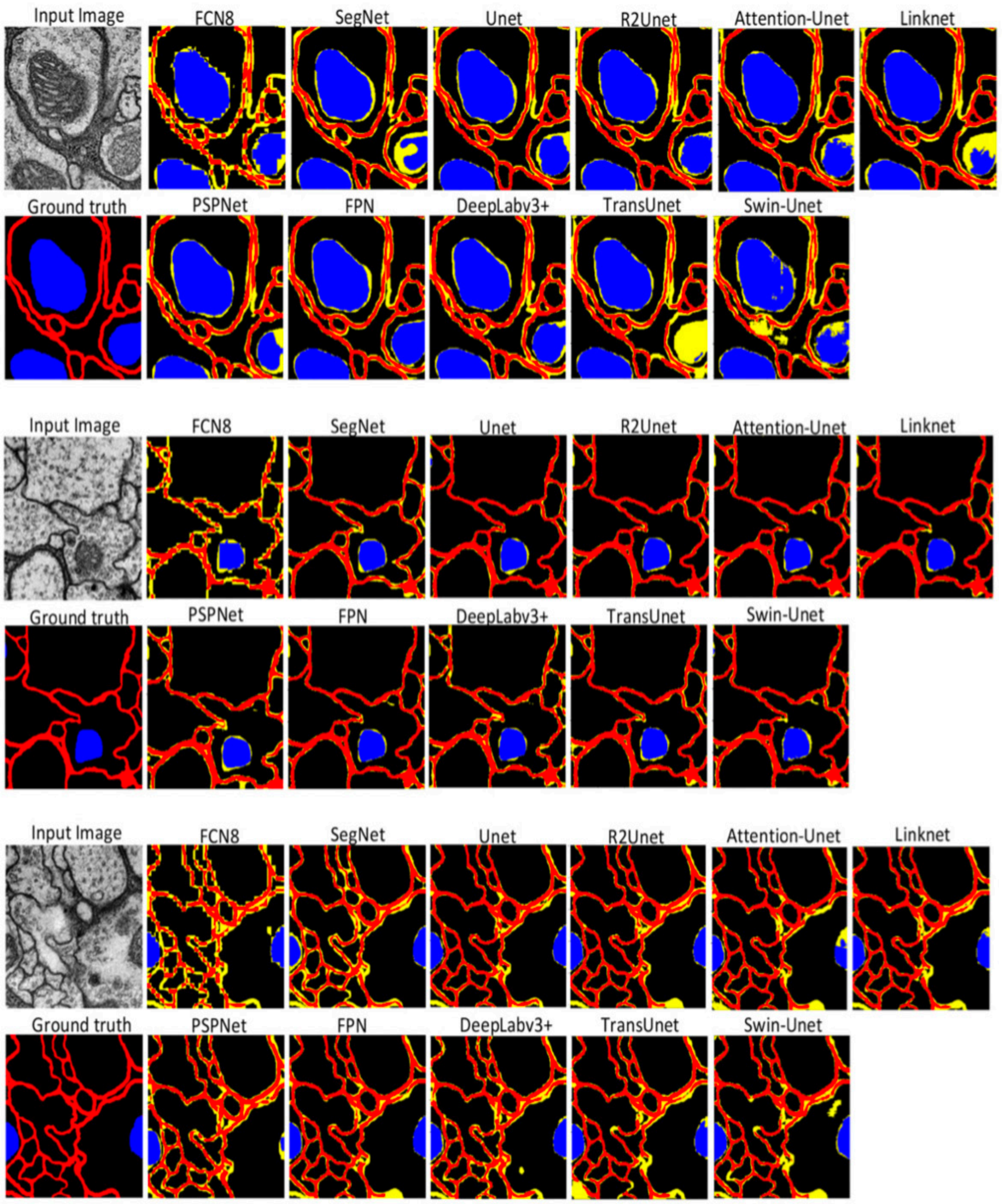


Figure 26: Illustration of the predicted segmentation maps at the output of the deep learning models. Misclassified pixels are colored in yellow for better visualization.

Table 1: Mean IoU and IoU per class metrics of the tested deep learning segmentation methods computed from Drosophila test images. Each color represent segmentation models category (cyan: fully convolutional based model, light blue: encoder-decoder based models, red: multiscale and pyramid based models, green: dilated convolutional based model, orange: Transformers based models) . Values in bold represent the best model

Methods	**mIoU**	**IOU per class**		
		Cytoplasm	*Cell membrane*	*Mitochondria*
FCN8	0.727	0.858	0.518	0.805
Segnet	0.785	0.897	0.647	0.811
Unet	0.837	0.926	0.745	0.841
R2Unet	**0.840**	**0.926**	**0.747**	**0.848**
Attention-Unet	0.827	0.924	0.731	0.825
Linknet	0.821	0.922	0.732	0.848
PSPNet	0.795	0.902	0.676	0.807
FPN	0.833	0.924	0.742	0.887
DeepLabv3+	0.821	0.909	0.683	0.870
TransUnet	0.778	0.909	0.702	0.724
Swin-Unet	0.816	0.918	0.713	0.818

Table 2: Average inference time (in seconds) computed from the Drosophila test images for the tested deep learning segmentation methods. Value in bold represent the fastest model.

Methods	FCN8	Segnet	Unet	R2Unet	Atten-Unet	Linknet	PSPNet	FPN	DeepLabv3+	TransUnet	Swin-Unet
Time	0.086	0.011	0.079	0.074	0.086	0.082	0.064	0.083	**0.009**	0.021	0.015

ent dataset, segmentation problem, or selection of hyperparameters could lead to different model selection than those obtained in this note.

Furthermore, we used the focal loss to handle class imbalance for the use case studied in this note. It would be interesting to study other loss functions and their impacts on training and testing performance. In addition, deep learning methods for object detection were not discussed in this note. These methods are applied in a wide range of applications and some extensions are also used for instance segmentation problems, so it would be interesting to revisit this topic in future work.

References

[1] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[2] Xiangbin Liu, Liping Song, Shuai Liu, and Yudong Zhang. A review of deep-learning-based medical image segmentation methods. *Sustainability*, 13(3):1224, 2021.

[3] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.

[4] Ali AHMAD. An overview of supervised machine learning and deep learning methods: Application to microscopy cell images classification. `https://coperneec.com/nos-publications/`, 2022.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[9] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[11] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[13] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[15] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.

[16] Md Zahangir Alom, Chris Yakopcic, Mahmudul Hasan, Tarek M Taha, and Vijayan K Asari. Recurrent residual u-net for medical image segmentation. *Journal of Medical Imaging*, 6(1):014006, 2019.

[17] Abhishek Chaurasia and Eugenio Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2017.

[18] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

[19] Martin Weigert, Uwe Schmidt, Robert Haase, Ko Sugawara, and Gene Myers. Star-convex polyhedra for 3d object detection and segmentation in microscopy. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3666–3673, 2020.

[20] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature methods*, 18(1):100–106, 2021.

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[22] Selim Seferbekov, Vladimir Iglovikov, Alexander Buslaev, and Alexey Shvets. Feature pyramid network for multi-class land segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 272–275, 2018.

[23] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[24] Xiaokang Qi, Jingshi Dong, Yubin Lan, and Hang Zhu. Method for identifying litchi picking position based on yolov5 and pspnet. *Remote Sensing*, 14(9):2004, 2022.

[25] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017.

[26] Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multi-scale filters for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3562–3572, 2019.

[27] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[28] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[29] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[30] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 24, 2011.

[31] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[32] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[34] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *AI Open*, 2022.

[35] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021.

[36] Abdul Mueed Hafiz, Shabir Ahmad Parah, and Rouf Ul Alam Bhat. Attention mechanisms and deep learning for machine vision: A survey of the state of the art. *arXiv preprint arXiv:2106.07550*, 2021.

[37] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

[38] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[39] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[40] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. *arXiv preprint arXiv:2105.05537*, 2021.

[41] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021.

[42] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[43] Yujian Mo, Yan Wu, Xinneng Yang, Feilin Liu, and Yujun Liao. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing*, 493:626–646, 2022.

[44] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[45] Yu Xiang and Dieter Fox. Da-rnn: Semantic mapping with data associated recurrent neural networks. *arXiv preprint arXiv:1703.03098*, 2017.

[46] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1857–1866, 2018.

[47] Qin Huang, Chunyang Xia, Chihao Wu, Siyang Li, Ye Wang, Yuhang Song, and C-C Jay Kuo. Semantic segmentation with reverse attention. *arXiv preprint arXiv:1707.06426*, 2017.

[48] Ali Hatamizadeh, Assaf Hoogi, Debleena Sengupta, Wuyue Lu, Brian Wilcox, Daniel Rubin, and Demetri Terzopoulos. Deep active lesion segmentation. In *International Workshop on Machine Learning in Medical Imaging*, pages 98–105. Springer, 2019.

[49] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[50] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian conference on computer vision*, pages 213–228. Springer, 2016.

[51] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE, 2020.

[52] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[53] Stephan Gerhard, Jan Funke, Julien Martel, Albert Cardona, and Richard Fetter. Segmented anisotropic sstem dataset of neural tissue. *figshare*, pages 0–0, 2013.

Nous sommes Canopee, un cabinet de conseil indépendant, spécialiste en Finance, DATA et transformation digitale.

Nous sommes l'*empowering ecosystem* ! Un écosystème en perpétuel mouvement.
Un écosystème qui donne le pouvoir à nos collaborateurs, nos projets et nos clients de se nourrir de l'émulation collective pour lui donner de la force.

Depuis 2009, nous intervenons dans les secteurs de la BFI de l'Asset Management, la banque privée ou de détail, les services financiers et l'assurance et cela auprès de clients tels que SG, HSBC,BNP Paribas ou encore ALLIANZ GI et AXA IM. Des écosystèmes réglementaires et spécifiques qui ont été nos premiers terrains de jeu.
C'est ici que nous avons su développer notre agilité, nos compétences, notre sens de l'engagement. Aujourd'hui, nous grandissons et continuons à nous investir et nous engager sur des écosystèmes spécifiques, techniques, mais diversifiés tels que l'industrie, le retail ou encore la pharma.
Toujours autour de nos 3 expertises que sont la finance, la data et la transformation digitale.

canopee
empowering *ecosystem*

www.canopee-group.com